SIMULATION OF DISTRIBUTED OBJECT-ORIENTED SERVERS

by

Chee Khan, KWOK

December 2003

Thesis Advisor: Bill Ray
Second Reader: Shing Man Tak

Approved for public release; distribution is unlimited
**Title:** Simulation of Distributed Object Oriented Servers

**Author:** Kwok Chee Khan

**Abstract:**
Distributed object oriented (OO) computing such as RMI, COBRA and SOAP etc is fast becoming the de-facto standard for software development.

The aim of the system designer is to determine the optimal deployment strategy for the system to perform efficiently. This is an enormous task especially when multiple object servers are fielded on hardware of different specifications. The number of possible deployment strategy of object servers to hardware grows exponentially with increase number of object server and machine. For example, with 3 machines and 10 object servers there are 59049 possible deployment patterns. Eventually, the number of possible deployment makes it impossible for system designer to setup test bed to determine the optimal deployment strategy.

The main goal of the simulation model is to analyze the object server deployment, verify an existing optimization model and to determine the optimal deployment strategy that will reduce the client response time. In one of the experiment conducted with the simulation model, in an environment with 3 machine and 10 object servers, it will take 53 years to attempt all deployment patterns in the lab environment. The simulation model will take only 13 days, which is an improvement of 1480%.
SIMULATION OF DISTRIBUTED OBJECT ORIENTED SERVERS

Chee Khan, KWOK
Civilian, Ministry of Defense Singapore
B.S., National University of Singapore, 1995

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2003

Author: Chee Khan, KWOK

Approved by: Bill Ray
Thesis Advisor

Shing Man-tak
Second Reader

Peter Denning
Chairman, Department of Computer Science
ABSTRACT

Distributed object oriented (OO) computing such as RMI, COBRA and SOAP etc is fast becoming the de-facto standard for software development. Distributed OO systems can consist of multiple object servers and client application on a network computer, as oppose to a single large centralized object server.

The aim of the system designer is to determine the optimal deployment strategy for the system to perform efficiently. This is an enormous task especially when multiple object servers are fielded on hardware of different specifications. The number of possible deployment strategy of object servers to hardware grows exponentially with increase number of object server and machine. For example, with 3 machines and 10 object servers there are 59049 possible deployment patterns. Eventually, the number of possible deployment makes it impossible for system designer to setup test bed to determine the optimal deployment strategy.

The main goal of the simulation model is to analyze the object server deployment, verify an existing optimization model and to determine the optimal deployment strategy that will reduce the client response time. In one of the experiment conducted with the simulation model, in an environment with 3 machine and 10 object servers, it will take 53 years to attempt all deployment patterns in the lab environment. The simulation model will take only 13 days, which is an improvement of 1480%.
# TABLE OF CONTENTS

## I. INTRODUCTION

## II. BACKGROUND

### A. INTRODUCTION

### B. PROFILES

1. Hardware Profiles
2. Object Server Profiles
3. Client Application Profiles
4. User Profiles

### C. OPTIMIZATION MODEL

1. Objective Function
2. Processing Speed Term
3. Network Speed Term
4. RAM Limits
5. CPU Limits

### D. VALIDATION OF OPTIMIZATION MODEL

### E. CURRENT RESEARCH APPROACH

## III. OVERVIEW OF OMNET++ AND NETWORK SIMULATION

### A. BACKGROUND

### B. MODELING CONCEPTS

1. Hierarchical Nested Modules
2. Communicate with Message through Channel
3. Parameters
4. Topology Description Method

### C. PROGRAMMING THE ALGORITHM

### D. RUNNING THE SIMULATION

1. Running the Simulation
2. Analyzing The Result

## IV. OBJECT ORIENTED SERVER SIMULATION MODEL

### A. INTRODUCTION

### B. REQUIREMENT MODELING

1. Use Case Diagram
2. Actor Description
3. Use Case Description

### C. CLASS DESCRIPTION

1. Class Inheritance
2. Association Diagram

### D. NETWORK DESCRIPTION (NED) FILE

### E. MESSAGE DESCRIPTION

### F. IMPLEMENTATION DETAIL

1. Computation of Response Time
2. Random Role Execution........................................................ 31
3. RAM Limit............................................................................... 32
4. Output Result ........................................................................ 32
G. CONFIGURATION FILE .............................................................. 33
H. PATTERN GENERATOR ............................................................... 34
1. Deployment Configuration File............................................. 34
2. Role Configuration File ................................................................ 35

V. VERIFICATION EXPERIMENT 1 .................................................. 37
A. INTRODUCTION ..................................................................... 37
B. SCENARIO PROFILE .................................................................. 37
C. COMPARING SIMULATION RESULT WITH TEST BED ........... 40
1. Result for Test Case S1 ........................................................ 43
2. Result for Test Case S2 ........................................................ 45
3. Result for Test Case S3 ........................................................ 47
4. Result for Test Case S4 ........................................................ 49
5. Result for Test Case S5 ........................................................ 51
6. Result for Test Case S6 ........................................................ 53
7. Result for Test Case S7 ........................................................ 55
8. Result for Test Case S8 ........................................................ 57
9. Result for Test Case S9 ........................................................ 59
D. COMPARING SIMULATION RESULT WITH TIME SLICE .... 61
E. STUDY OF GROWTH POTENTIAL ............................................. 62
F. COMPARING SIMULATION RESULT WITH LINGO MODEL ...... 64
G. CONCLUSION ............................................................................. 64

VI. VERIFICATION EXPERIMENT 2 .................................................. 67
A. INTRODUCTION ..................................................................... 67
B. SCENARIO PROFILE .................................................................. 67
C. MODEL SETUP .......................................................................... 69
1. Simulation Model Setup ..................................................... 69
2. Lingo Model Setup .............................................................. 70
D. WEIGHED MODEL .................................................................... 70
1. Ranking .................................................................................. 70
2. Highest Priority ....................................................................... 70
3. Weighted Model ...................................................................... 70
E. COMPARING SIMULATION RESULT WITH LINGO MODEL ...... 71
1. Equal Weigh Model .............................................................. 71
2. Effect of Weigh Model .......................................................... 73
F. CONCLUSION ............................................................................. 74

VII. RECOMMENDATION AND FUTURE WORKS ............................ 75
A. MODELING OF SYSTEM USING UML ..................................... 75
B. IMPROVED NETWORK MODELING ....................................... 75
C. REFINING THE RESULT TO ROLE/METHOD CALL LEVEL ...... 76
D. ENTERPRISE RESOURCE PLANNING (ERP) ......................... 77
E. CONSTRAINTING THE DEPLOYMENT PATTERNS ................. 77
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simple and Compound Modules</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>Connecting modules</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Demonstration user interface</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Debugging window</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Plave view of response time output file.</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>User Case Diagram</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Class inheritance diagram</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>Class diagram</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>Simulation Model for S5</td>
<td>39</td>
</tr>
<tr>
<td>10</td>
<td>Graphical result of S1</td>
<td>43</td>
</tr>
<tr>
<td>11</td>
<td>CPU Utilization for Optimal Deployment Strategy Pattern 1</td>
<td>44</td>
</tr>
<tr>
<td>12</td>
<td>Role Response Time for Optimal Deployment Pattern 1</td>
<td>44</td>
</tr>
<tr>
<td>13</td>
<td>Graphical result of S2</td>
<td>45</td>
</tr>
<tr>
<td>14</td>
<td>CPU Utilization for S2</td>
<td>46</td>
</tr>
<tr>
<td>15</td>
<td>Role Response Time for S2</td>
<td>46</td>
</tr>
<tr>
<td>16</td>
<td>Graphical result of S3</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>CPU Utilization for S3</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>Response Time for S3</td>
<td>48</td>
</tr>
<tr>
<td>19</td>
<td>Graphical result of S4</td>
<td>49</td>
</tr>
<tr>
<td>20</td>
<td>CPU Utilization for S4</td>
<td>50</td>
</tr>
<tr>
<td>21</td>
<td>Response Time for S4</td>
<td>50</td>
</tr>
<tr>
<td>22</td>
<td>Graphical result of S5</td>
<td>51</td>
</tr>
<tr>
<td>23</td>
<td>CPU Utilization for S5</td>
<td>52</td>
</tr>
<tr>
<td>24</td>
<td>Response Time for S5</td>
<td>52</td>
</tr>
<tr>
<td>25</td>
<td>Graphical result of S6</td>
<td>53</td>
</tr>
<tr>
<td>26</td>
<td>CPU Utilization for S6</td>
<td>54</td>
</tr>
<tr>
<td>27</td>
<td>Graphical result of S7</td>
<td>55</td>
</tr>
<tr>
<td>28</td>
<td>CPU Utilization for S7</td>
<td>56</td>
</tr>
<tr>
<td>29</td>
<td>Response Time for S7</td>
<td>56</td>
</tr>
<tr>
<td>30</td>
<td>Graphical result of S8</td>
<td>57</td>
</tr>
<tr>
<td>31</td>
<td>CPU Utilization for S8</td>
<td>58</td>
</tr>
<tr>
<td>32</td>
<td>Response Time for S8</td>
<td>58</td>
</tr>
<tr>
<td>33</td>
<td>Graphical result of S9</td>
<td>59</td>
</tr>
<tr>
<td>34</td>
<td>Optimal Deployment Pattern S9</td>
<td>59</td>
</tr>
<tr>
<td>35</td>
<td>CPU Utilization for S9</td>
<td>60</td>
</tr>
<tr>
<td>36</td>
<td>Response Time for S9</td>
<td>60</td>
</tr>
<tr>
<td>37</td>
<td>Graphical result of Time Slice</td>
<td>61</td>
</tr>
<tr>
<td>38</td>
<td>Effect of growth on response time</td>
<td>63</td>
</tr>
<tr>
<td>39</td>
<td>Role Response Time</td>
<td>72</td>
</tr>
<tr>
<td>40</td>
<td>Result wrt to overall response time</td>
<td>73</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1. Computation of deployment pattern .......................................................... 2
Table 2. Example of hardware profile .................................................................... 6
Table 3. Example of object server profile ............................................................... 7
Table 4. Example of client application profile ....................................................... 8
Table 5. Example of user profile ........................................................................... 8
Table 6. Actor description ..................................................................................... 22
Table 7. Network Description File Parameters .................................................... 27
Table 8. Simple Module Parameters .................................................................... 29
Table 9. Message Description Table ..................................................................... 30
Table 10. Configuration file description ............................................................... 34
Table 11. Sample bit arithmetic operation .............................................................. 35
Table 12. Machine Profile .................................................................................... 37
Table 13. Object Server Performance Profile ....................................................... 37
Table 14. Complex Object Server Profile ............................................................. 37
Table 15. Role Call Pattern ................................................................................... 38
Table 16. User Interface Call Chart ...................................................................... 38
Table 17. Test Case ............................................................................................... 39
Table 18. Test Result of Test Case S1 – S4 ............................................................ 41
Table 19. Test Result of Test Case S5 – S9 ............................................................ 42
Table 20. Optimal Deployment Pattern S1 ............................................................. 43
Table 21. Optimal Deployment Pattern S2 ............................................................. 45
Table 22. Optimal Deployment Pattern S3 ............................................................. 47
Table 23. Optimal Deployment Pattern S4 ............................................................. 49
Table 24. Optimal Deployment Pattern S5 ............................................................. 51
Table 25. Optimal Deployment Pattern S6 ............................................................. 54
Table 26. Optimal Deployment Pattern S7 ............................................................. 55
Table 27. Optimal Deployment Pattern S8 ............................................................. 57
Table 28. Test Result with Time Slice ................................................................... 62
Table 29. Effect of growth on deployment pattern ............................................... 63
Table 30. Result for Test Case S1-S3 ................................................................. 64
Table 31. Result for Test Case S4-S6 ................................................................. 64
Table 32. Result for Test Case S7-S9 ................................................................. 64
Table 33. Deployment Strategy Result ................................................................. 64
Table 34. Machine Profile .................................................................................... 67
Table 35. Object Server Performance Profile ...................................................... 68
Table 36. Complex Object Server Call Profile ..................................................... 68
Table 37. Role Call Pattern .................................................................................. 69
Table 38. User Interface Call Chart ..................................................................... 69
Table 39. Example of Weighted Model ............................................................... 71
Table 40. Deployment Pattern ........................................................................... 72
Table 41. Weighted model .................................................................................. 73
THIS PAGE INTENTIONALLY LEFT BLANK
ACKNOWLEDGMENTS

I would like to thank Professor Bill Ray for his support and guidance throughout the course of my thesis work. His keen interest and knowledge in the subject area and technical expertise has been a great source of help. I thank Andras Varga, the developer of OMNet++, for the support provided in answering some of my queries related to the product. I would like to thank all the other 6 Singaporean students in the Master Defense Technology System (MDTS) Information Assurance Track for the wonderful time in learning and playing together. The time spent here will be remembered. I'm grateful for my wonderful wife Wendy and daughter Charis who provided me with their constant support, love, and understanding throughout this research. Lastly, I would like to thank God for providing me this opportunity to learn and improved myself.
I. INTRODUCTION

With the massive reduction in the cost of microprocessors and the large increase in microprocessor performance over the past few years, distributed microcomputer based systems are becoming a cost effective solution to many problems. Coupled with the advancements of object-oriented technology in the past decade, object oriented architecture is often used for the building of large scale computer based systems.

Distributed object oriented (OO) computing such as Remote Method Invocation (RMI), Common Object Request Broker Architecture (COBRA) and Simple Object Access Protocol (SOAP) etc is fast becoming the de-facto standard for software development. Distributed OO systems can consist of multiple object servers and client applications on a network of computers, as oppose to a single large centralized object server. Consequently, information processing is distributed over several computers rather than confined to a single machine.

In military environment, a set of distributed object servers could be used to support many applications aboard a ship. The distributed object servers could serve track object, intel object, logistic object and target object to a host of client applications. These applications could handle such tasks as Anti-Submarine Warfare (ASW), Anti-Surface Warfare (ASUW), Anti-Air Warfare (AAW), Electronic Warfare (EW), humanitarian missions and rescue missions.

A user’s network of computers will change frequently. Object servers, applications, hardware and user preferences will be in a constant state of change. The distribution of object servers across different hardware platform will result in multiple deployment strategies. The number of deployment strategy = $m^n$, which grows exponentially with an increase in either the number of object servers (n) or hardware platform (m).
<table>
<thead>
<tr>
<th>Object Server</th>
<th>Computer</th>
<th>Number of pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>243</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>59049</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 1. Computation of deployment pattern

As the number of deployment pattern increases, it becomes a tedious task for system engineers to setup all deployment patterns in order to find the deployment strategy that will provide the optimal system efficiency. The difficulty in selecting the best pattern increases with the need to deploy different object servers on computers with different CPU and memory capacity. Lack of better tools, the deployment strategy is based on best-guess or experience of the system engineer. The resulting deployment strategy may/may not be optimal.

As a system evolves, common hardware changes consist of adding new computers, removing old computers, upgrading CPUs, modifying RAM and modifying network bandwidth capacity. Each of these hardware changes will produce an event that would trigger the system to re-evaluate its deployment strategy. The intent of system engineers is always to find a deployment strategy that performs at peak efficiency. Some measurement of efficiency can be based on CPU resource utilization, bandwidth utilization and user response time. The user response time is affected by the following:

- The number of clients. The greater the number of client, the more object server calls will be executed and the longer is the response time.
- The frequency of function called invoked by each client. With an increase in the frequency of function call, more requests will be queued and therefore the longer will be the response time.
• The number of instruction cycle in the object method invoked. The more complex the object method, the number of instruction cycle will increase thus increasing the response time.

• The speed of the microprocessor. The faster the processor, the faster is the response time.

• The scheduling algorithm of the microprocessor. Depending on the time needed to complete an object server call, a non-preemptive and time slicing scheduling algorithm will result in different response time.

• The number of object server served by a single microprocessor. With more object server loaded in a computer, the higher the number request are queued thus increasing the response time. The number of object server to be loaded in a computer is also limited by the RAM capacity.

• The network delay due to transmission delay and data rate. The higher the network delay, the greater the response time.

By knowing ahead of time how many users will be accessing the system, each user interaction pattern and each computer configuration, a model can be developed to make recommendation of a possible optimal deployment strategy. Even a simplistic model may lead to large gains in performance.

The goal of this research is to develop a simulation model that models a distributed object-oriented environment. The model will allow system engineers to model the environment [i.e. the number of clients, the interaction patterns, computer configuration], and simulate the deployment of object server on different hardware platforms and derive the optimal deployment strategy. The criterion used to evaluate the optimal deployment strategy was to minimize the user response time.

This chapter gave a brief introduction to the problem and the motivation for the research. Chapter II gives an introduction to the dissertation work done
by Professor William J Ray [William 2001] that used a mathematical model to predict the optimal deployment patterns. Chapter III introduces the discrete event simulation tool OMNet++ that is used to model the distributed object-oriented environment. Chapter IV documents the design of the distributed object oriented simulation model, supporting tool and how the result of the simulation run can be used. Chapter V contains the comparative study on the results of the simulation model with the experimental result from Professor William J Ray’s dissertation. Chapter VI contains the comparative study of the mathematical and simulation model on a more complex scenario. Chapter VII discusses future refinement to the model. Chapter VIII contains the conclusions that can be drawn from this research.

The appendix contains the following information

- Appendix A – List of source code of the simulation model.
- Appendix B – List of source code of the supporting utility.
- Appendix C – Experiment described in Chapter V.
- Appendix D – Data collected from the test described in Chapter VI.
- Appendix E – The modified Lingo model.
II. BACKGROUND

A. INTRODUCTION

There has been little work on deployment strategies for distributed object servers. Some relevant research is in the area of load balancing and client/server performance.

The goal of load balancing is to balance the load across multiple machines. For a situation where inter object server call is prevalent, better response time may require having different object servers to run on a single machine. Therefore, to optimize the response time, previous load balancing strategies may not be useful.

The focus of current optimization technique for the client/server performance is to improve the performance of single server relationship with its clients. The need of analyzing distributed object servers architecture is to target performance when multiple servers are involved.

A research work by Professor William J. Ray describe a method that can generate distributed object oriented server deployment architectures to take advantage of hardware platform and network resources for the purpose of reducing average client response time. Average client response time was chosen over minimizing the maximum response time of one call because the method takes into account the entire usage profile. The proposed optimization model maps system characteristic (called profiles) into equations to minimize the average client response time.

The work done by Professor William J. Ray is used as a springboard for the development of this research work and the rest of this chapter will describe the approach he adopted.

B. PROFILES

A profile is an abstraction of a given characteristic of the system. The elements in the profile are the raw data that the model will use to reason about
the given characteristic. There are profiles for each machine, server, application and user type. There is also a profile that describes the network. The system then must map all of these profiles into equations to minimize the response time. The more complex the modeling of the hardware becomes the more computationally intensive the approach will become. Initially, the research explores an approach with simplistic profiles to demonstrate its capabilities.

1. **Hardware Profiles**

   The hardware profiles describe the characteristic of the hardware platforms used in the deployment. The aspects being modeled in the hardware profiles include characteristics of each computer such as CPU speed, RAM size. The hardware profile also models the network speed between each pair of computers.

   An example of hardware profile is:

<table>
<thead>
<tr>
<th>Hardware</th>
<th>RAM</th>
<th>CPU speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIX</td>
<td>64MB</td>
<td>600MHz</td>
</tr>
<tr>
<td>BR733</td>
<td>128MB</td>
<td>733MHz</td>
</tr>
<tr>
<td>GIGA</td>
<td>128MB</td>
<td>1000MHz</td>
</tr>
</tbody>
</table>

   Table 2. Example of hardware profile

2. **Object Server Profiles**

   The object server profile describes the behavior and method supported by an object server. It models the method computational time, return message size and RAM utilization of the object server. The computational time can be collected easily with a small client application that exercises each method call and records the data. Thus, actual implementation code for the application isn’t needed to estimate the object server profiles. In the optimization model, the computational time of each method call needs to be captured and normalized to specific hardware architecture. The return message size is used to compute the network
transmission time and therefore giving preference for the object server to be deployed on a hardware platform with a higher network speed. The object server RAM utilization is used to limit the number of object servers by the hardware RAM capacity.

An example of object server profile is:

<table>
<thead>
<tr>
<th>Object Server</th>
<th>RAM</th>
<th>Method</th>
<th>Return message</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>44MB</td>
<td>1</td>
<td>100kbytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>120kbytes</td>
</tr>
<tr>
<td>B</td>
<td>60MB</td>
<td>1</td>
<td>50 kbytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>200 kbytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>20 kbytes</td>
</tr>
</tbody>
</table>

Table 3. Example of object server profile

3. **Client Application Profiles**

The client application profiles describe the client interaction with the object servers. It includes modeling the object server method call and the estimated frequency of call. Profiling becomes more difficult if the application code is not available. When source code is not available, mechanism must be in place to record all the events that occur in the task. The system must allow a user to create typical scenarios and record the method calls that occur in the scenario. This could be done by simulation or monitoring calls to the object servers when the system is in a training mode.

An example of client application profile is:

<table>
<thead>
<tr>
<th>Client Application</th>
<th>Button</th>
<th>Methods Called</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client 1</td>
<td>Button 1</td>
<td>A.1</td>
</tr>
<tr>
<td></td>
<td>Button 2</td>
<td>A.2 + B.1</td>
</tr>
<tr>
<td>Client 2</td>
<td>Button 1</td>
<td>C.1 + D.3</td>
</tr>
</tbody>
</table>
4. **User Profiles**

The user profile describes the way a user interacts with a client application. The way a user interacts can be characterized, but not precisely predicted. User interaction is based on the role the user played. Different role will execute different application that would interact differently with the distributed object servers. A more refined profile could include frequency information for the tasks and calls for each task and response time goals for each task. By profiling each role, the user could choose to re-optimize his deployment to decrease the response time when user chosen roles change. Multiple roles can exist for each user. The user could then select a set of roles and have the system come up with an optimal deployment strategy to meet these criteria.

An example of user profile is:

<table>
<thead>
<tr>
<th>Role</th>
<th>Button called</th>
<th>Frequency over 20 mins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role 1</td>
<td>Client 1. Button 1</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Client 2. Button 2</td>
<td>10</td>
</tr>
<tr>
<td>Role 2</td>
<td>Client 1. Button 2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Client 2. Button 1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Client 2. Button 3</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5. Example of user profile
C. OPTIMIZATION MODEL

1. Objective Function

The objective function that needs to be minimized is the sum of all the response times for a given call pattern over a given time interval.

\[
\text{Minimize } \sum_{n=0}^{N} \sum_{m=0}^{M} \frac{a_{nm} \ast R_n \ast S_{\text{norm}}}{S_m} + \sum_{i=0}^{N} \sum_{j=0}^{N} B_{ij} \nabla
\]

Note that the optimization process ranges over all possible combinations for \( a_{nm} \) and finds the minimum based on the above objective function and constraints.

The objective function is subject to the following four constraints:

- Object servers cannot be split across machine
  \[
  A_{nm} = \begin{cases} 
  1, & \text{iff server n is running on machine m.} \\
  0, & \text{otherwise}
  \end{cases}
  \]

- Each server can run on only one machine [no multiple instances of the same server].
  \[
  \forall n \left[ \sum_{m=0}^{M} a_{nm} = 1 \right]
  \]

- RAM usage by the object servers cannot pass a set threshold on each machine.
  \[
  \forall m \left[ \sum_{n=0}^{N} a_{nm} \ast V_n \leq T_m \ast U \right]
  \]
• CPU time on a given machine cannot surpass the corresponding real time interval.

\[ \forall m \left[ \sum_{n=0}^{N} a_{nm} \ast R_n \ast S_{\text{norm}} \right] \leq C \]

where

- \( N \) = Number of object servers
- \( M \) = Number of physical machines
- \( R_n \) = Normalized machine load of server \( n \) (seconds, s)
- \( S_{\text{norm}} \) = Speed of the normalizing machine (MHz)
- \( S_m \) = Speed of machine \( m \) (MHz)
- \( B_{ij} \) = Data sent between server \( i \) to server \( j \) (bits, b)
- \( Q_{ij} \) = Network Speed between server \( i \) to server \( j \) (bps)
- \( T_m \) = Physical RAM on machine \( m \) (bits, b)
- \( n \) = Memory allocated by server \( n \) (bits, b)
- \( U \) = Multiple to limit RAM utilization \([0.1 < U < 3.0]\)
- \( C \) = Time Interval [seconds, s]

2. Processing Speed Term

This part of the function looks at all possible deployment patterns.

\[ \sum_{n=0}^{N} \sum_{m=0}^{M} a_{nm} \ast R_n \ast S_{\text{norm}} \]
\(a_{nm}\) is used to keep track of the deployments. \(a_{nm}\) is zero if SERVER \(n\) is not located on MACHINE \(m\). If SERVER \(n\) is location on MACHINE \(m\), then \(a_{nm}\) is one.

3. **Network Speed Term**

The network speed term of the objective function is:

\[
\text{Minimize} \, \sum_{i=0}^{N} \sum_{\substack{j=0 \, j \neq i}}^{N} \frac{B_{ij}}{Q_{ij}}
\]

The network speed term adds some time for each time a server-to-server method is called. The number of bits is divided by the rate of transmission.

4. **RAM Limits**

The RAM limit is to limit the amount of a machine RAM that can be used by the object servers. This constraint basically states that the total memory usage of all the object servers loaded on a machine will be less than a percentage of the memory on that machine.

5. **CPU Limits**

Since all of the processing measurements are averages and the user profiles are averages over time, we cannot exceed 100% CPU loading. Even though the CPU can queue tasks when overloads, it doesn’t have the chance to catch up if the user profiles truly reflect the user requests.

D. **VALIDATION OF OPTIMIZATION MODEL**

The optimization model is validated by experimental measurement. A test bed was created with Windows 2000 machines that match the characteristics of the profiles. Servers were created using JDK 1.3 and RMI as the middleware.
Software to simulate the three different users was also created. This simulation software was instrumented to measure the actual time the software was blocked waiting for an object server method call to response. The experiment involves 3 hardware and 3 object servers, a total of 27 different deployment patterns. All 27 different configurations were established and the average response time for each configuration was measured and recorded.

All 27 configurations were tested twice. One tested the configuration with the object servers using much less than the stated memory needs. Another tested the configuration with the object servers using all of the stated memory needs. Some configurations strained the machines memory limit. These configurations resulted in system failures in the test results.

E. CURRENT RESEARCH APPROACH

This research aims to develop a simulation model for the distributed object-oriented environment. It uses the same profile representation as the optimization model. Deployment strategy with different profiles is build using a discrete event simulation tools OMNet++.

The model will simulate user and object server interaction and compute the average response time for each user. The average response time for each deployment strategy is compiled and the minimum user response time is chosen as the optimal deployment strategy.

The experimental and optimization model of 3 object servers and 3 machines is chosen to verify the accuracy of the simulation model to the real-world implementation.
III. OVERVIEW OF OMNET++ AND NETWORK SIMULATION

A. BACKGROUND

OMNet++ is an object oriented modular discrete event simulator. The name itself stands for Objective Modular Network Test-bed in C++. The development of OMNet++ was started at the Technical University of Budapest (BME), in 1992. It has been developed mostly by Andras Varga at the Department of Telecommunication (BME-HIT).

The simulation tools can be used for modeling:

- Communication protocols
- Computer networks and traffic modeling
- Multi-processor and distributed systems.
- Administrative systems
- Any other system where the discrete event approach is suitable.

A Discrete Event System is a system where state changes (events) happen at discrete points of time, and events take zero time to happen. It is assumed that nothing (i.e. nothing interesting) happens between two consecutive events, that is, no state change takes place in the system between the events (in contrast to continuous systems where state changes are continuous).

Those systems that can be viewed as Discrete Event Systems can be modeled using Discrete Event Simulation such as OMNet++. (Continuous systems are modeled using differential equations and suchlike.) Computer networks are usually viewed as Discrete Event Systems.

B. MODELING CONCEPTS

OMNet++ provides efficient tools for the developer to describe the structure of the actual system. Some of the main features are:

- Hierarchically nested modules.
• Modules communicate with message through channels.
• Flexible module parameters.
• Topology description language.

1. Hierarchical Nested Modules

An OMNet++ model consists of hierarchically nested modules. The depth of module nesting is not limited, which allows the user to reflect the logical structure of the actual system in the model structure. The model is often referred to as networks. The top level model is the system module. The system module contains sub-modules, which can also contain sub-modules themselves. Simple modules encapsulate C++ code that generate and react to events, in other words, implement the behavior of the model. Modules can have parameters which are used for three main purposes: to customize module behavior; to create flexible model topologies and for module communication as shared variables.

Client computers, servers, and network devices can be modeled as sub-modules.

2. Communicate with Message through Channel

Gates are the input and output interfaces of the modules; messages are sent out through output gates and arrive through input gates. Each connection is created within a single level of the module hierarchy: within a compound module,
one can connect the corresponding gates of two modules, or a gate of one sub-module and a gate of the compound module.

Figure 2. Connecting modules

Three attributes can be assigned values in the body of the connection declaration, all of them optional:

- Propagation delay (sec) is the amount of time the arrival of the message is delayed when it travels through the channel.
- Bit error rate (errors/bit) has influence on the transmission of messages through the channel.
- Data rate (bits/sec) is used for transmission delay calculation.

OMNeT++ uses messages to represent events. Messages are sent from one module to another – this means that the place where the “event will occur” is the message’s destination module, and the model time when the event occurs is the arrival time of the message. Modules can send messages directly to their destination or along a predefined path, through gates and connections. The “local simulation time” of a module advances when the module receives a message. The message can arrive from another module or from the same module.

Time within the model is often called simulation time, model time or virtual time as opposed to real time or CPU time or which refers to how long the simulation program has been running or how much CPU time it has consumed.
3. Parameters

Modules can have parameters that are used for three purposes:

- To parameterize module topology
- To customize simple module behavior
- For module communication, as shared variables.

Within a compound module, parameters can define the number of sub-modules, number of gates, and the way the internal connection are made. Compound modules can pass parameters or expressions of parameters to their sub-modules.

4. Topology Description Method

The developer defines the structure of the model in the NED (Network Description) language descriptions. The NED language supports modular description of a network. This means that a network description consists of a number of component descriptions (channels, simple/compound module types). The channels, simple modules and compound modules of one network description can be used in another network description.

C. Programming the Algorithm

The simple modules of a model contain the algorithms as C++ functions. The full flexibility and power of the programming language can be used, supported by the OMNet++ simulation class library. Elements of the simulation (messages, modules, queues etc.) are represented as objects and they are designed so that they can efficiently work together, creating a powerful framework for simulation programming.

D. Running the Simulation

An OMNet++ model consists of the following parts:
• NED language topology description(s) which describe the module structure with parameters, gates etc.

• Simple modules sources that implement the behavior of each modules.

The simulation system provides the following components:

• Simulation kernel. This contains the code that manages the simulation and the simulation class library.

• User interfaces. OMNet++ user interfaces are used with simulation execution, to facilitate debugging, demonstration, or batch execution of simulations.

1. Running the Simulation

The simulation executable is a standalone program, thus it can be run on other machines without OMNet++ or the model files being present. When the program is started, it reads in a configuration file (usually called omnetpp.ini); it contains setting that control how the simulation is run, values for model parameters etc. The configuration file can also prescribe several simulations runs; in the simplest case, they will be executed by the simulation program one after another.

OMNet++ simulations can feature different user interfaces for different purposes: debugging, demonstration and batch execution. Advanced user interfaces make the inside of the model visible to the user, allow him/her to start/stop simulation execution and to intervene by changing variables/objects inside the model. User interfaces also facilitate the demonstration on how a model worked internally.

The same simulation model can be executed with different user interface without any change in the model files themselves. The user would test and debug the simulation with a powerful graphical user interface, and finally run it with a simple and fast user interface that supports batch execution.
Figure 3. Demonstration user interface

Figure 4. Debugging window
2. Analyzing The Result

The output of the simulation is written into data files: output vector files, output scalar files, and possibly the user's own output files. Data such as resource utilization, response time can be written into the output files. OMNet++ provides a GUI tool names Plove to view and plot the contents of the output vector files. The output files are text files in a format which can be read into math packages like Matlab, or imported into spreadsheets like Excel.

Figure 5. Plove view of response time output file.
THIS PAGE INTENTIONALLY LEFT BLANK
IV. OBJECT ORIENTED SERVER SIMULATION MODEL

A. INTRODUCTION

This chapter will describe the requirement, design consideration and implementation of the simulation model.

B. REQUIREMENT MODELING

1. Use Case Diagram

The block diagram below describes how the system designer interacts with the Distributed Object Oriented Server Simulation Model.

![User Case Diagram](image)

Figure 6. User Case Diagram
2. Actor Description
   The actor that interacts with the simulation model is as follows:

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Designer</td>
<td>The actor is responsible for determining the optimal deployment pattern. The actor will describe the simulated environment with hardware, object server, application and role profiles. The actor will execute the simulation model and determine the optimal deployment pattern based on the simulation result.</td>
</tr>
</tbody>
</table>

   Table 6. Actor description

3. Use Case Description
   The following is the description of the use case.

   **Use Case:** Define simulation environment
   
   **Brief Description:**
   To allow the user to define the environment to be simulated based on the machine, object server, role, and client application profile.

   **Special Requirements:**
   The simulation model should be a black box to the system designer. The redefinition of a new simulation environment should be achieved by configuration file. The definition of the simulation environment should not require the user to modify the source code of the simulation model.

   **Pre-condition**
   The environment to be simulated must already be defined. That includes the hardware, object server configuration, and the role.

   **Use Case:** Generate deployment pattern
   
   **Brief Description:**
   To allow the user to generate all the deployment pattern based on the number of machine and object server.
Special Requirements:
Nil

Pre-condition
The number of machine and object server must be known.

Use Case: Simulate environment

Brief Description:
To allow the user to define the simulation duration and execute the simulation.

Special Requirements:
Nil

Pre-condition
The simulation environment must be defined.

Use Case: Review result

Brief Description:
To allow the user to review and compare the simulation result of different deployment pattern.

Special Requirements:
Nil

Pre-condition
The simulation must be executed.
C. CLASS DESCRIPTION

1. Class Inheritance

The behaviour of each component in the simulated environment is modeled as an extension to the cSimpleModule class provided by the OMNet++ library. The class inheritance diagram is shown in the following figure:

![Class Inheritance Diagram](image)

Figure 7. Class inheritance diagram

The purpose of each class is described in the following table:

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>sObjSvr</td>
<td>sObjSvr is used to model the behavior of an object server.</td>
<td>Control Object</td>
</tr>
<tr>
<td>sMachine</td>
<td>sMachine is used to model the behavior of a computer hardware.</td>
<td>Control Object</td>
</tr>
<tr>
<td>sSwitch</td>
<td>sSwitch is used to model the behavior of a network switch.</td>
<td>Control Object</td>
</tr>
</tbody>
</table>
sRole is used to model the behavior of a role.

CAddress is used to store the address of each machine and role that is connected with the switch. The address is used by the DNS Service to provide location independent name lookup to determine the gate to send a message.

CBtnList is used to store the list of client application button and the function called when the button is invoked.

CDNSSvc implement the necessary function needed to provide location independent lookup service.

CInteractList is used to store the list of complex method call.

CObjList is used to store the list of method supported by all object servers.

CSimEvent is used to model the random invocation of application button by a role.

CStatLog is used to write the simulation result in the output file stat.log and computed.log.

2. Association Diagram

<table>
<thead>
<tr>
<th>sMachine</th>
<th>support</th>
<th>sObjSvr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0..*</td>
<td></td>
</tr>
</tbody>
</table>
D. NETWORK DESCRIPTION (NED) FILE

The Network Description (NED) language describes the network environment. The NED description file will enable the simulation model to automatically build the distributed object server environment based on the parameters supplied by the designer. The parameters used in the defining the environment is shown in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>run_name</td>
<td>The name of the simulation run. This is used to identify the output result of the different run in a batch execution. Typically the run_name is given value as run1, run2….runN.</td>
</tr>
<tr>
<td>num_machine</td>
<td>The number of machines deployed in the simulated environment. The number of machines will affect the number of connection created on the sSwitch to support switch-machine network communication.</td>
</tr>
</tbody>
</table>
Table 7. Network Description File Parameters

The Network Description (NED) language also defines the simple module used to model the object server, machine and the network switch. Each simple module definition includes parameters and gates that will be used to setup the environment and how each of the simple modules inter-connect. The actual value to be used is defined in the [Parameters] section of the omnet.ini file.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Parameters/Gates</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sRole</td>
<td>def_file</td>
<td>The filename of the configuration file used by the role.</td>
</tr>
<tr>
<td></td>
<td>r2sw_out</td>
<td>The gate used for sending message from role to switch.</td>
</tr>
<tr>
<td></td>
<td>r2sw_in</td>
<td>The gate used for receiving message from the switch.</td>
</tr>
<tr>
<td>sSwitch</td>
<td>sw2r_out[]</td>
<td>The gates used for sending message from switch to role. The number of gates is created dynamically based on the number of roles specified in num_role.</td>
</tr>
<tr>
<td></td>
<td>sw2r_in[]</td>
<td>The gates used for receiving message from switch to role.</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>switch.</td>
<td>The number of gates is created dynamically based on the number of roles specified in num_role.</td>
<td></td>
</tr>
<tr>
<td><strong>sw2m_out[]</strong></td>
<td>The gates used for sending message from switch to role. The number of gates is created dynamically based on the number of roles specified in num_machine.</td>
<td></td>
</tr>
<tr>
<td><strong>sw2m_in[]</strong></td>
<td>The gates used for receiving message from switch. The number of gates is created dynamically based on the number of machines specified in num_machine.</td>
<td></td>
</tr>
<tr>
<td>sMachine</td>
<td>num_objsvr</td>
<td>The number of object servers that was running on this machine.</td>
</tr>
<tr>
<td>objsvridx</td>
<td></td>
<td>The starting index of the first object server running on this machine. This index is used to reference to the list of object server created.</td>
</tr>
<tr>
<td>def_file</td>
<td></td>
<td>The filename of the configuration file used by the machine.</td>
</tr>
<tr>
<td><strong>m2sw_out</strong></td>
<td></td>
<td>The gate used for sending message from machine to switch.</td>
</tr>
<tr>
<td><strong>m2sw_in</strong></td>
<td></td>
<td>The gate used for receiving message from the machine.</td>
</tr>
<tr>
<td><strong>m2os_out[]</strong></td>
<td></td>
<td>The gates used for sending message from machine to object server. The number of gate is created dynamically based on the number of object servers specified in num_objsvr.</td>
</tr>
<tr>
<td><strong>m2os_in[]</strong></td>
<td></td>
<td>The gates used for receiving message from</td>
</tr>
</tbody>
</table>
object server. The number of gates is created
dynamically based on the number of
machines specified in num_objsvr.

| sObjSvr | cfg_file | The filename of the configuration file used by
the machine. |
|---------|---------|--------------------------------------------------|
| os2m_out |         | The gate used for sending message from
object server to the machine. |
| os2m_in |         | The gate used for receiving message from the
machine. |

Table 8. Simple Module Parameters

E. MESSAGE DESCRIPTION

Several messages are used in the simulation model. The following table
is a description of the messages used.

<table>
<thead>
<tr>
<th>Message</th>
<th>From/To</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME_REGISTER</td>
<td>Role/Switch</td>
<td>Use by the machine and role to register the name with the switch.</td>
</tr>
<tr>
<td></td>
<td>Machine/Switch</td>
<td></td>
</tr>
<tr>
<td>OBJSVR_REGISTER</td>
<td>ObjSvr/Machine</td>
<td>Use by the object server to register itself with the machine.</td>
</tr>
<tr>
<td>INVOKE_OBJECT_SVR_CALL</td>
<td>Role/ObjSvr</td>
<td>Use by the role to invoke an object server call.</td>
</tr>
<tr>
<td>INVOKE_OS2OS_CALL</td>
<td>ObjSvr/ ObjSvr</td>
<td>Use by an object server to call another object server. This is used to model complex method call.</td>
</tr>
<tr>
<td>MACHINE_COMPLETE_</td>
<td>Machine/ObjSvr</td>
<td>Use by the machine to inform</td>
</tr>
</tbody>
</table>
Table 9. Message Description Table

<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>the object server that the local method call has completed.</td>
</tr>
<tr>
<td>MACHINE_COMPLETE_ EXEC_WITH_REMOTE</td>
<td>Machine/ObjSvr</td>
</tr>
<tr>
<td>OBJSVR_EXECUTE_ON _CPU</td>
<td>ObjSvr/Machine</td>
</tr>
<tr>
<td>OBJSVR_RESPONSE</td>
<td>ObjSvr/Role</td>
</tr>
</tbody>
</table>

F. IMPLEMENTATION DETAIL

1. Computation of Response Time

Before the role invokes a client application button, a timestamp will be recorded. The simulation will invoke the relevant object server method calls that are associated to the application button. If there are more than one object-server method calls, a new method call will only be invoked after the role has received the response from the current method call. For example, for the application button C2.B5 described in Chapter V, the role will invoke B.2 after the reply from A.1 is received. When the response from the last method call is received, a second timestamp is recorded and the difference in time is the response time.

The response time is dependent on the following:-
- The number and complexity of method calls. The higher number of method calls has a higher tendency for a longer response time. The more complex the method calls, the longer the response time.

- Network transmission time. This includes the total transmission time to and fro from the role to machine. The network transmission time is dependent on the message size, and the bandwidth.

- The time needed to execute the object server method call. This is dependent on the CPU utilization, the CPU speed, the number of instruction cycles and the scheduling term.

The response time is consolidated over the simulation run and the average response time is computed.

2. Random Role Execution

A few random feature is added to introduce randomness in the execution of the simulation model. This is an attempt to model the random nature of the real operating environment. The random features are:-

- Random invocation of client application button by each role.

- Random allocations of CPU resource for the time slice scheduling option.

The probability of an invocation of client application button is assigned during the definition of the role profile. For example, during the training session, it is recorded there was 50 call to C1.B1, 10 call to C1.B2 and 30 call to C2.B2, the configuration file will be defined as:-

```
```

An internal table will be set up with range from 0 to 89, and a random integer based on uniform distribution will be generated to determine which method to fire. For example, if the number drawn is 45, C1.B1 will be called and if the number drawn is 81, C2.B2 will be called. Due to this random implementation,
the simulation will model closer to the real world environment if the simulation is run over sufficient period of time.

For modeling random allocation of CPU resource for time slice scheduling, a maximum CPU allocation time is defined in the hardware profile. In real life, this value represents the longest time a process can execute on the CPU without being interrupted. When a process starts to consume CPU resource, the time slice scheduling will pick a random time between 0 and the maximum CPU allocation time, this random time will be the duration (sec) a process can execute on the CPU. After the duration has passed, the current process will be suspended, a new process selected for execution and another random CPU execution time will be generated.

3. **RAM Limit**

Each machine has limited RAM capacity, with virtual memory the number of applications that can execute simultaneously on a single machine is no longer limited by the RAM capacity. In virtual memory, when a reference memory is not in RAM, a page fault will occur and the reference memory will be moved from the virtual memory (in disk) to the main memory (RAM). This I/O operation is time consuming. As more application runs on a machine, the probability of page fault will increase and thus delaying the process execution time. In the simulation model, the RAM limit is used to limit the total amount of memory used by the object server that is running on the machine.

When a new object server registered to be run on a machine, the machine will monitor the total memory used. If the (total memory used) > (Ram limit * machine total RAM) then the simulation will terminate with a memory error. When the Ram limit is set to 0.0, the simulation will assumes that the machine has unlimited memory to support the object server.

4. **Output Result**

There are three output files that are generated after each of the simulation run. The three files are vector file, stat.log and computed.log.
The vector file is generated by OMNet++. The vector file can be read by Plove application which provides a graphical display of the output data. The strength of the vector file lies in the ease of reading the result. However, when the number of output data is large, then it is difficult to extract the result from the graphical display.

The stat.log and computed.log file are textual files that was formatted in a way that is easy to manipulate using third party spreadsheet application such as Excel. Stat.log file stores the run name, the role id, the number of client applications call invoked, the maximum and minimum response time and the average response time. The stat.log file is useful for comparing the boundary of the response time. The computed.log file stores the run name, the role id and the average response time. For an environment with 4 of role2, stat.log file will contain the result of all four role2 but the computed.log will contain the computed average response time of all role2.

G. CONFIGURATION FILE

A number of configuration file is used to define the simulation model. These configuration files are used to define parameters that are beyond the scope of the OMNet++ parameter files [ometpp.ini]. The list and description of the configuration files is shown in the following table:-

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button.def</td>
<td>The file defines the list of button provided by the client application. The file also defines the list of object server method call when the application button is invoked.</td>
</tr>
<tr>
<td>Interact.def</td>
<td>The file defines the list of complex method call.</td>
</tr>
<tr>
<td>Obj.def</td>
<td>The file defines the list of method served by all object servers. It defines the list of method, the number of instruction cycles needed and the return message</td>
</tr>
</tbody>
</table>
Machine.def The file defines profile of a machine. It defines the machine name, CPU power, RAM size and limit, process and disk swap time, and the time slicing property.

ObjSvr.def The file defines profile of an object server. It defines the object server name, the object served, the ram utilization.

Role.def The file defines profile of a role. It defines the role name, the calling frequency, the object server method called and probability of call.

<table>
<thead>
<tr>
<th>Configuration File Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Machine].def</td>
</tr>
<tr>
<td>[ObjSvr].def</td>
</tr>
<tr>
<td>[Role].def</td>
</tr>
</tbody>
</table>

Table 10. Configuration file description

H. PATTERN GENERATOR

As described in Chapter 1, the number of deployment strategy = \( m^n \) (\( m \) – number of machine, \( n \) – number of object server). With 3 machine and 10 object server, there is a total of 59409 deployment strategy. Manual generation of such number of deployment strategy will be a laborious task – an automated deployment strategy generator is needed. The PatternGenerator is developed to automate the generation of configuration file. The two configuration files generated are:-

- Deployment strategy configuration file.
- Role configuration file.

1. Deployment Configuration File

The format of the deployment configuration file is shown in Appendix [William 2001]. The complexity lies in ensuring that the correct number of deployment patterns based on the environment is correctly generated. A set of bits (\( n \) bits) is used to represent the number of hardware platform such that \( 2^n > \) number of hardware. Multiple set of bits are used to represent each object
server. These bits are concatenated m times, where m is the number of object server and subsequent bit arithmetic will ensure that all patterns were represented. The resulting bit string will have n * m bits.

For example, in an environment has 3 hardware platforms, [SIX, BR733, GIGA], and 3 object server [A, B, C]. The number of bits need to represent the 3 hardware platform is 2 i.e. $2^2 = 4$. The bit representations are:

[00 – SIX]; [01 – BR733]; [10 – GIGA]; [11 – Invalid]

A resultant bit string of $2\times3 = 6$ bit length is needed to represent all the object servers. The arithmetic operation perform is shown in the following table.

<table>
<thead>
<tr>
<th>Bit pattern</th>
<th>Hardware deployment</th>
<th>Arithmetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 10 10</td>
<td>A – GIGA, B – GIGA, C – GIGA,</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>10 10 01</td>
<td>A – GIGA, B – GIGA, C – BR733</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>10 10 00</td>
<td>A – GIGA, B – GIGA, C – SIX</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>10 01 11</td>
<td>Invalid</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>10 01 10</td>
<td>A – GIGA, B – BR733, C – GIGA</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>10 01 01</td>
<td>A – GIGA, B – BR733, C – BR733</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>10 01 00</td>
<td>A – GIGA, B – BR733, C – SIX</td>
<td>Subtract bit string with 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>00 00 00</td>
<td>A – SIX, B – SIX, C – SIX</td>
<td>Complete</td>
</tr>
</tbody>
</table>

Table 11. Sample bit arithmetic operation

2. Role Configuration File

In the simulation environment, there are multiple role types and a number of roles per type. For example, an environment may have 10 of Role1, and 5 of Role2. To support the simulation, each role must have a separate configuration file. The PatternGenerator will generate all the role configuration file needed based on input value of the role type, the number of role, and the based role configuration file. In the above example, to generate the configuration file for Role1, the input value are:

Role Type : Role1
Number of Role : 10
Based File : role1.cfg

The PatternGenerator will generate 10 role1 type namely role1_0 to role1_9 and create 10 configuration file from the based file. The configuration file will contain the respective role name. The example of the based file and generated role1_0 file is given below.

Based File  role1.def

[name] ROLE1
[type] 1
[min_wait] 6
[max_wait] 7
[call] C1.B1/10
[dr_sw] 200000000
[btw_call] 0.05

Generated File  role1_0.def

[name] ROLE1_0
[type] 1
[min_wait] 6
[max_wait] 7
[call] C1.B1/10
[dr_sw] 200000000
[btw_call] 0.05
V. VERIFICATION EXPERIMENT 1

A. INTRODUCTION

The first verification experiment is based on the scenario used by the William Ray dissertation. The objective of this verification exercise is to verify the accuracy of the simulation model as compared to the real-life testbed and mathematical model.

B. SCENARIO PROFILE

The various profiles used in defining the environment is shown in the following table:

<table>
<thead>
<tr>
<th>Machine</th>
<th>RAM (MB)</th>
<th>CPU Speed (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIX</td>
<td>64</td>
<td>600</td>
</tr>
<tr>
<td>BR733</td>
<td>128</td>
<td>733</td>
</tr>
<tr>
<td>GIGA</td>
<td>128</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 12. Machine Profile

<table>
<thead>
<tr>
<th>OBJECT SERVER</th>
<th>METHOD</th>
<th>CPU TIME (s)</th>
<th>Number of instruction cycles</th>
<th>Average Size of Message (bits)</th>
<th>RAM Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0.5796</td>
<td>579600</td>
<td>11200</td>
<td>44</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>2.6203</td>
<td>2620300</td>
<td>18400</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>1.18175</td>
<td>1181750</td>
<td>44800</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>2.0264</td>
<td>2026400</td>
<td>17600</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1.76655</td>
<td>1766550</td>
<td>400000</td>
<td>60</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>3.70085</td>
<td>3700850</td>
<td>2720000</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>3.0043</td>
<td>3004300</td>
<td>32000</td>
<td>66</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4.804</td>
<td>4804000</td>
<td>4000000</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>0.48815</td>
<td>488150</td>
<td>40000</td>
<td></td>
</tr>
</tbody>
</table>

Table 13. Object Server Performance Profile

<table>
<thead>
<tr>
<th>Primary Method</th>
<th>Secondary Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.2</td>
<td>C.1</td>
</tr>
</tbody>
</table>

Table 14. Complex Object Server Profile
### Table 15. Role Call Pattern

<table>
<thead>
<tr>
<th>Roles</th>
<th>Call Pattern</th>
</tr>
</thead>
</table>

### Table 16. User Interface Call Chart

<table>
<thead>
<tr>
<th>Buttons</th>
<th>Method Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1.B1</td>
<td>A.1</td>
</tr>
<tr>
<td>C1.B2</td>
<td>A.2 + B.1</td>
</tr>
<tr>
<td>C1.B1</td>
<td>A.1</td>
</tr>
<tr>
<td>C1.B2</td>
<td>A.2 + B.1</td>
</tr>
<tr>
<td>C2.B1</td>
<td>C.1 + C.2</td>
</tr>
<tr>
<td>C2.B2</td>
<td>C.3</td>
</tr>
<tr>
<td>C2.B3</td>
<td>C.2</td>
</tr>
<tr>
<td>C2.B4</td>
<td>C.3</td>
</tr>
<tr>
<td>C2.B5</td>
<td>A.1 + B.2</td>
</tr>
<tr>
<td>C2.B6</td>
<td>B.2</td>
</tr>
<tr>
<td>C2.B7</td>
<td>A.4</td>
</tr>
<tr>
<td>C2.B8</td>
<td>C.3 + A.3</td>
</tr>
<tr>
<td>C2.B9</td>
<td>A.1 + A.2 + A.3 + B.2</td>
</tr>
<tr>
<td>C3.B1</td>
<td>C.1</td>
</tr>
<tr>
<td>C3.B2</td>
<td>B.1 + B.2</td>
</tr>
<tr>
<td>C3.B3</td>
<td>C.2</td>
</tr>
</tbody>
</table>
A total of 9 test cases is run with different number of role participation and ram limits. All test cases were simulated using the simulation model and the result is then compared with the result from the testbed and Lingo model. The 9 test cases are:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1 user (Role 1), Ram Limit 1.5</td>
</tr>
<tr>
<td>S2</td>
<td>1 user (Role 2), Ram Limit 1.5</td>
</tr>
<tr>
<td>S3</td>
<td>1 user (Role 3), Ram Limit 1.5</td>
</tr>
<tr>
<td>S4</td>
<td>4 user (Role 2), Ram Limit 1.5</td>
</tr>
<tr>
<td>S5</td>
<td>3 user (Role 3), Ram Limit 1.5</td>
</tr>
<tr>
<td>S6</td>
<td>28 user (Role 1), Ram Limit 1.5</td>
</tr>
<tr>
<td>S7</td>
<td>1 user (Role 1), Ram Limit 1.0</td>
</tr>
<tr>
<td>S8</td>
<td>1 user (Role 2), Ram Limit 1.0</td>
</tr>
<tr>
<td>S9</td>
<td>1 user (Role 3), Ram Limit 1.0</td>
</tr>
</tbody>
</table>

The simulation model was run over a simulation time of 8hrs. In the actual testbed, the time needed to execute 27 patterns x 8hrs = 216hrs. For the
simulation model, execution time for 27 patterns is about 10mins. For the Lingo model, the optimal solution is found within 4 sec.

C. COMPARING SIMULATION RESULT WITH TEST BED

The simulation result of the S1-S9 is shown in the following two tables, highlighted cell represents the lowest average response time and thus the optimal deployment pattern.
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Sim S1</th>
<th>Sim S2</th>
<th>Sim S3</th>
<th>Sim S4</th>
<th>TB S1</th>
<th>TB S2</th>
<th>TB S3</th>
<th>TB S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>0.9558</td>
<td>5.2635</td>
<td>7.0909</td>
<td>15.0493</td>
<td>0.97633</td>
<td>5.15036</td>
<td>6.74195</td>
<td>14.60339</td>
</tr>
<tr>
<td>02</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1.0962</td>
<td>5.7279</td>
<td>8.4143</td>
<td>11.67535</td>
<td>0.89934</td>
<td>5.53033</td>
<td>8.26652</td>
<td>11.7461</td>
</tr>
<tr>
<td>03</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1.0558</td>
<td>6.371</td>
<td>7.9366</td>
<td>11.43463</td>
<td>0.96081</td>
<td>6.41717</td>
<td>7.80217</td>
<td>11.71142</td>
</tr>
<tr>
<td>05</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1.2533</td>
<td>5.8716</td>
<td>7.3726</td>
<td>11.43985</td>
<td>1.1408</td>
<td>5.95302</td>
<td>7.41334</td>
<td>11.3353</td>
</tr>
<tr>
<td>06</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1.2313</td>
<td>6.3193</td>
<td>8.6062</td>
<td>10.41733</td>
<td>1.21888</td>
<td>6.23306</td>
<td>8.50534</td>
<td>11.66662</td>
</tr>
<tr>
<td>07</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1.1837</td>
<td>6.7915</td>
<td>8.2699</td>
<td>17.17995</td>
<td>1.11909</td>
<td>6.87797</td>
<td>8.14272</td>
<td>17.06668</td>
</tr>
<tr>
<td>09</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1.1935</td>
<td>5.9121</td>
<td>9.4624</td>
<td>11.67603</td>
<td>0.99153</td>
<td>5.95855</td>
<td>9.25922</td>
<td>12.35508</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1.0857</td>
<td>7.0883</td>
<td>8.7751</td>
<td>13.61095</td>
<td>0.87878</td>
<td>7.17686</td>
<td>8.62741</td>
<td>14.30257</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>1.2209</td>
<td>7.6891</td>
<td>11.0043</td>
<td>19.2105</td>
<td>1.15777</td>
<td>7.8528</td>
<td>10.71298</td>
<td>18.37826</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1.3782</td>
<td>8.1308</td>
<td>9.3435</td>
<td>22.0154</td>
<td>1.41398</td>
<td>8.21186</td>
<td>8.972</td>
<td>20.87854</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1.3452</td>
<td>7.4566</td>
<td>10.5915</td>
<td>17.3523</td>
<td>1.19742</td>
<td>7.34209</td>
<td>10.38713</td>
<td>17.40676</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1.2877</td>
<td>7.955</td>
<td>10.3251</td>
<td>15.1899</td>
<td>1.30637</td>
<td>7.86233</td>
<td>10.36099</td>
<td>15.65908</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1.5391</td>
<td>7.9663</td>
<td>10.9063</td>
<td>13.8823</td>
<td>1.46744</td>
<td>8.03317</td>
<td>10.59013</td>
<td>15.40732</td>
</tr>
</tbody>
</table>

Table 18. Test Result of Test Case S1 – S4.
<table>
<thead>
<tr>
<th>Pat</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Sim S5</th>
<th>Sim S6</th>
<th>Sim S7</th>
<th>Sim S8</th>
<th>Sim S9</th>
<th>TB S5</th>
<th>TB S6</th>
<th>TB S7</th>
<th>TB S8</th>
<th>TB S9</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>16.6589</td>
<td>10.2673</td>
<td>1.0055</td>
<td>5.3012</td>
<td>6.9886</td>
<td>15.9786</td>
<td>0.97734</td>
<td>5.12018</td>
<td>6.77685</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>11.9086</td>
<td>5.0118</td>
<td>1.0488</td>
<td>5.6899</td>
<td>8.3268</td>
<td>13.9259</td>
<td>4.96472</td>
<td>0.94298</td>
<td>5.58044</td>
<td>8.21316</td>
</tr>
<tr>
<td>03</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>12.8099</td>
<td>7.4687</td>
<td>1.0636</td>
<td>6.3843</td>
<td>8.0593</td>
<td>13.0662</td>
<td>4.33376</td>
<td>0.88703</td>
<td>6.34986</td>
<td>7.90056</td>
</tr>
<tr>
<td>05</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>14.9085</td>
<td>7.5642</td>
<td>1.1665</td>
<td>5.8862</td>
<td>7.3435</td>
<td>14.6145</td>
<td>7.00596</td>
<td>1.14467</td>
<td>5.87464</td>
<td>7.26764</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>14.4386</td>
<td>7.6163</td>
<td>1.0827</td>
<td>7.392</td>
<td>8.776</td>
<td>14.2478</td>
<td>error</td>
<td>0.96261</td>
<td>7.28895</td>
<td>8.53298</td>
</tr>
<tr>
<td>11</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>26.8708</td>
<td>3.5508</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>25.7966</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>19.7351</td>
<td>21.8104</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>19.0296</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>15.0131</td>
<td>17.3469</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>15.8605</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
<tr>
<td>15</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>30.6386</td>
<td>26.8308</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>30.3491</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>15.9958</td>
<td>11.4163</td>
<td>1.4044</td>
<td>7.5513</td>
<td>10.6311</td>
<td>18.1436</td>
<td>error</td>
<td>1.2627</td>
<td>7.3226</td>
<td>10.5296</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>17.4819</td>
<td>15.2354</td>
<td>1.3635</td>
<td>8.145</td>
<td>10.1949</td>
<td>17.6795</td>
<td>error</td>
<td>1.43925</td>
<td>8.14897</td>
<td>10.1235</td>
</tr>
<tr>
<td>18</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>26.8316</td>
<td>7.6992</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>25.8905</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>19.6194</td>
<td>21.2661</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>19.8816</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
<tr>
<td>21</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>15.9954</td>
<td>17.1591</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>18.0537</td>
<td>error</td>
<td>error</td>
<td>error</td>
<td>Error</td>
</tr>
</tbody>
</table>

Table 19. Test Result of Test Case S5 – S9
1. Result for Test Case S1

The graphical result of the simulation model and testbed for Test Case S1 is shown in the following figure. The graph shows that there is a general trend of the two recorded response time is similar.

![Graphical result of S1](image)

**Figure 10. Graphical result of S1**

The simulation model chose a pattern that is the 4th best result from the test bed. The difference in the timing selected is 0.098 sec.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 1 (1 user), Ram Limit 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

**Table 20. Optimal Deployment Pattern S1**

The difference in the result could be attributed to the randomness of the test bed and simulation.
Figure 11. CPU Utilization for Optimal Deployment Strategy Pattern 1

Figure 12. Role Response Time for Optimal Deployment Pattern 1

The CPU utilization diagram shows that the CPU utilization of GIGA is fairly low at (0.55%) and therefore the deployment pattern is able to support a
significant growth in the number of roles. The response time range is \((8, 0.8)\) sec (max, min).

2. Result for Test Case S2

The graph shows that the simulation model modeled the test bed accurately. The highest difference between any deployment patterns is 0.2 sec.

![Simulation Result of Test Case S2](image)

**Figure 13.** Graphical result of S2

For test case 2, the simulation model chose the same optimal deployment pattern as the test bed.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 2 (1 user) , Ram Limit 1.5</th>
<th>Sim</th>
<th>Testbed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIX</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
<td>A, B, C</td>
<td></td>
</tr>
</tbody>
</table>

Table 21. Optimal Deployment Pattern S2
Figure 14. CPU Utilization for S2

Figure 15. Role Response Time for S2
The CPU utilization diagram shows that the CPU utilization of GIGA is low at (43%) and therefore the deployment pattern is able to support a higher number of roles. The response time range is (8.8, 0.8) sec (max, min).

3. Result for Test Case S3

The graph shows that simulation model modeled the test bed accurately. The highest difference between any two patterns is 0.6 sec.

![Simulation Result of Test Case S3](image)

Figure 16. Graphical result of S3

For test case 3, the simulation model chose the same optimal deployment pattern as the test bed.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 3 (1 user) , Ram Limit 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

Table 22. Optimal Deployment Pattern S3
The CPU utilization diagram shows that the CPU utilization of GIGA is low at (59%) and therefore the deployment pattern is able to support a small increase in the number roles. The response time range is (11.8, 0.5) sec (max, min).
4. **Result for Test Case S4**

The graph shows that simulation model modeled the test bed accurately. The highest difference between any two patterns is 1 sec.

![Simulation Result of Test Case S4](image)

**Figure 19.** Graphical result of S4

For test case 4, the simulation model chose the same optimal deployment pattern as the test bed.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 2 (4 user) RAM Limit 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>C</td>
</tr>
<tr>
<td>BR733</td>
<td>A</td>
</tr>
<tr>
<td>GIGA</td>
<td>B</td>
</tr>
</tbody>
</table>

**Table 23.** Optimal Deployment Pattern S4

49
The average CPU utilization is GIGA 69%, BR733 52%, SIX 40%. The CPU utilization diagram shows that the CPU utilization of GIGA is high although it has the fastest microprocessor. We can derive that for Role 2 deployment the
object server B is consuming more CPU cycle compared to other object server. Therefore, to support Role 2, the object server B should run on the fastest CPU. The deployment pattern might be able support a small increase in the number roles. However, an increase may result in GIGA being the bottleneck of the deployment server and thus further delaying the response time. The response time range is (30.0, 0.5) sec (max, min).

5. Result for Test Case S5

The graph shows that simulation model modeled the test bed accurately. The highest difference between any two patterns is 2 sec.

Table 24. Optimal Deployment Pattern S5

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 3 (3 user) RAM Limit 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>A</td>
</tr>
<tr>
<td>BR733</td>
<td>B</td>
</tr>
<tr>
<td>GIGA</td>
<td>C</td>
</tr>
</tbody>
</table>
Figure 23. CPU Utilization for S5

Figure 24. Response Time for S5
The average CPU utilization is GIGA 65%, BR733 62%, SIX 25%. The CPU utilization diagram shows that the CPU utilization of GIGA is high although it has the fastest microprocessor. We can derive that for Role 3 deployment the object server C is consuming more CPU cycle compared to other object server. Therefore, to support Role 3, the object server C should run on the fastest CPU. The deployment pattern might be able support a small increase in the number roles. However, an increase in the number of Role 3 may result in GIGA being the bottleneck of the deployment server and thus further delaying the response time. The response time range is (32.0, 0.5) sec (max, min).

6. Result for Test Case S6

The graph shows that simulation model modeled the test bed accurately. The highest difference between any two patterns is 3 sec. Deployment patterns that fail the RAM limit will be given a response time of 0. Although the total RAM is sufficient to serve the object server, the testbed registered more errors than the simulation model.

![Simulation Result of Test Case S6](image-url)
For test case 6, the simulation model chose the same optimal deployment pattern as the test bed.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 1 (28 user) RAM Limit 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>C</td>
</tr>
<tr>
<td>BR733</td>
<td>A</td>
</tr>
<tr>
<td>GIGA</td>
<td>B</td>
</tr>
</tbody>
</table>

Table 25. Optimal Deployment Pattern S6

The average CPU utilization is GIGA 10%, BR733 99%, SIX 40%. BR733 has reached optimal CPU utilization and therefore an increase in the number of role 1 will result in a significant increase in the response time.
7. Result for Test Case S7

The graph shows that simulation model modeled the test bed accurately. Deployment patterns that fail the RAM limit will be given a response time of 0. The highest difference between any two patterns is 0.1 sec.

![Simulation Result of Test Case S7](image)

Figure 27. Graphical result of S7

The simulation model chose the 4th optimal deployment pattern as the test bed. The difference in the timing selected is 0.09sec.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 1 (1 user) RAM Limit 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

Table 26. Optimal Deployment Pattern S7

The difference in the result could be attributed to the randomness of the test bed and simulation.
Figure 28. CPU Utilization for S7

Figure 29. Response Time for S7

The CPU utilization diagram shows that the CPU utilization of GIGA is fairly low at (0.55%) and therefore the deployment pattern is able to support a significant growth in the number of roles. The response time range is (8, 0.8) sec (max, min).
8. Result for Test Case S8

The graph shows that simulation model modeled the test bed accurately. Deployment patterns that fail the RAM limit will be given a response time of 0. The highest difference between any two patterns is 0.17 sec.

![Simulation Result of Test Case S8](image)

Figure 30. Graphical result of S8

For test case 8, the simulation model chose the same optimal deployment pattern as the test bed.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 2 (1 user) RAM Limit 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
</tr>
<tr>
<td>SIX</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

Table 27. Optimal Deployment Pattern S8
Figure 31. CPU Utilization for S8

Figure 32. Response Time for S8
The CPU utilization diagram shows that the CPU utilization of GIGA is low at (43%) and therefore the deployment pattern is able to support a higher number of roles. The response time range is (8.8, 0.8) sec (max, min).

9. Result for Test Case S9

The graph shows that simulation model modeled the test bed accurately. The highest difference between any two patterns is 0.3 sec.

For test case 9, the simulation model chose the same optimal deployment pattern as the test bed.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 3 (1 user)</th>
<th>RAM Limit 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sim</td>
<td>Testbed</td>
</tr>
<tr>
<td>SIX</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

Figure 34. Optimal Deployment Pattern S9
Figure 35. CPU Utilization for S9

Figure 36. Response Time for S9
The CPU utilization diagram shows that the CPU utilization of GIGA is low at (59%) and therefore the deployment pattern is able to support a small increase in the number roles. The response time range is (11.8, 0.5) sec (max, min).

D. COMPARING SIMULATION RESULT WITH TIME SLICE

The simulation model is executed for the same simulation duration with the time slice scheduling turn on. The time slice execution time is set to 0.3sec, 0.5sec, 1.0sec and 2.0sec.

The result of the simulation run is shown in the following table and graph:-

![Graphical result of Time Slice](image)

**Figure 37. Graphical result of Time Slice**

<table>
<thead>
<tr>
<th>Pattern</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>No Time Slice</th>
<th>Time Slice 0.5s</th>
<th>Time Slice 1.0s</th>
<th>Time Slice 2.0s</th>
<th>Time Slice 5.0s</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>10.2673</td>
<td>10.9358</td>
<td>10.0966</td>
<td>10.6519</td>
<td>10.6164</td>
</tr>
<tr>
<td>02</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>5.0118</td>
<td>5.3189</td>
<td>5.3019</td>
<td>5.1548</td>
<td>5.1669</td>
</tr>
<tr>
<td>03</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>7.4687</td>
<td>8.1836</td>
<td>7.6092</td>
<td>7.8479</td>
<td>7.4783</td>
</tr>
<tr>
<td>04</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>3.1608</td>
<td>3.2072</td>
<td>3.2172</td>
<td>3.0915</td>
<td>3.1004</td>
</tr>
<tr>
<td>05</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>7.5642</td>
<td>8.7225</td>
<td>8.1007</td>
<td>7.8081</td>
<td>7.6405</td>
</tr>
<tr>
<td>06</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>15.3563</td>
<td>16.7883</td>
<td>15.5095</td>
<td>15.4238</td>
<td>15.2421</td>
</tr>
<tr>
<td>07</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>11.3845</td>
<td>12.7411</td>
<td>12.1498</td>
<td>11.6686</td>
<td>11.6405</td>
</tr>
<tr>
<td>08</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>18.7025</td>
<td>20.6597</td>
<td>20.0523</td>
<td>18.9089</td>
<td>19.1652</td>
</tr>
<tr>
<td>09</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>5.0028</td>
<td>5.6524</td>
<td>5.1628</td>
<td>5.2054</td>
<td>5.0293</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>7.6163</td>
<td>8.149</td>
<td>7.6878</td>
<td>7.7994</td>
<td>7.7651</td>
</tr>
<tr>
<td>11</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>3.5508</td>
<td>3.7054</td>
<td>3.6127</td>
<td>3.5222</td>
<td>3.4656</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>17.3469</td>
<td>19.0373</td>
<td>17.9977</td>
<td>17.5824</td>
<td>17.7654</td>
</tr>
</tbody>
</table>
The time slicing scheduling has no effect on the prediction of the optimal deployment. The time slicing interrupt time has effect on the response time. If the time slice interrupt time is much smaller than the execution time of the method call, then the overall response time will increase. This is due to the additional overhead in switching the process. If the time slice interrupt time is close to the execution time, then the overall response time is similar to the result of non-time slicing scheduling.

**E. STUDY OF GROWTH POTENTIAL**

The simulation model is also useful to study the effect of growth in the number of roles. The simulation model is run with 1, 2, 4 and 8 Role 2 and the following result of the simulation is shown in the following diagram:-

Table 28. Test Result with Time Slice

<table>
<thead>
<tr>
<th></th>
<th>BR733</th>
<th>BR733</th>
<th>SIX</th>
<th>26.8308</th>
<th>29.3954</th>
<th>27.6991</th>
<th>26.661</th>
<th>26.788</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>11.4163</td>
<td>12.6777</td>
<td>12.1825</td>
<td>11.8247</td>
<td>11.816</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>15.2354</td>
<td>16.6897</td>
<td>15.9339</td>
<td>15.9211</td>
<td>15.4134</td>
</tr>
<tr>
<td>18</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>7.6992</td>
<td>8.5918</td>
<td>8.0474</td>
<td>7.7426</td>
<td>7.665</td>
</tr>
<tr>
<td>21</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>17.1591</td>
<td>18.8565</td>
<td>17.7471</td>
<td>17.8394</td>
<td>17.5215</td>
</tr>
<tr>
<td>22</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>3.0755</td>
<td>3.1661</td>
<td>3.0991</td>
<td>3.0019</td>
<td>3.0072</td>
</tr>
<tr>
<td>23</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>2.9624</td>
<td>2.9561</td>
<td>2.9226</td>
<td>2.9383</td>
<td>2.8604</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>7.646</td>
<td>8.6442</td>
<td>8.0717</td>
<td>7.7522</td>
<td>7.6896</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>7.5482</td>
<td>8.6368</td>
<td>8.1036</td>
<td>7.6793</td>
<td>7.6413</td>
</tr>
</tbody>
</table>
The simulation run reveals that the increase in the number of roles does change the optimal deployment pattern. This implies a change in the number of roles will require the system designer to redetermine the optimal deployment pattern.
F. COMPARING SIMULATION RESULT WITH LINGO MODEL

The result of deployment pattern by the Lingo model is shown in the following tables:

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 1 (1 user) S1</th>
<th>Role 2 (1 user) S2</th>
<th>Role 3 (1 user) S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim</td>
<td>Math</td>
<td>Sim</td>
<td>Math</td>
</tr>
<tr>
<td>SIX</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>A, B, C</td>
<td>A, B, C</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

Table 30. Result for Test Case S1-S3

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 1 (28 user) S4</th>
<th>Role 2 (4 user) S5</th>
<th>Role 3 (3 user) S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim</td>
<td>Math</td>
<td>Sim</td>
<td>Math</td>
</tr>
<tr>
<td>SIX</td>
<td>B</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>BR733</td>
<td>A, B, C</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>GIGA</td>
<td>C</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

Table 31. Result for Test Case S4-S6

<table>
<thead>
<tr>
<th>Roles</th>
<th>Role 1 (1 user) S7</th>
<th>Role 2 (1 user) S8</th>
<th>Role 3 (1 user) S9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim</td>
<td>Math</td>
<td>Sim</td>
<td>Math</td>
</tr>
<tr>
<td>SIX</td>
<td>C</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>BR733</td>
<td>A</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>GIGA</td>
<td>B</td>
<td>A, C</td>
<td>A, B, C</td>
</tr>
</tbody>
</table>

Table 32. Result for Test Case S7-S9

5 out of the 9 test case has predicted similar deployment pattern in both methods. For test case which the result differs, the mathematical prediction falls within the top 3 optimal deployment pattern.

G. CONCLUSION

The result of deployment pattern by the Lingo model is shown in the following table. Yes implies a correct matching in the recommended deployment pattern and No implies a mismatch in the recommended deployment pattern.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Testbed/Sim</th>
<th>Math/Sim</th>
<th>Testbed/Math</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>No (4th)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>S5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S6</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S7</td>
<td>No (4th)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>S8</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>S9</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 33. Deployment Strategy Result
From the result, the simulation model is able to model the real life test bed accurately. In fact, the accuracy of the simulation model is better than the mathematical model. The simulation model is able to model 7 out of 9 test cases whereas the mathematical model is only able to model the testbed 4 out of 9 test cases.

The simulation model also provides other useful features compare to the mathematical model:

- The simulation model will be able to rank the deployment pattern according to the response time. The mathematical model will only provide a single deployment pattern without making a reference to the rest of the deployment pattern. The ranking of deployment pattern will allow the system designer to make informed decision if other deployment pattern is chosen beside the optimal deployment pattern.

- The rank of the deployment pattern will also allow the system designer to make weighed consideration to the best deployment pattern. The weighed method in selecting best deployment pattern will be explored in the verification experiment 2.

- The minimum and maximum response time is useful information for system designer to know the upper and lower bound of the expected response time.

- The ability to execute the simulation model without considering RAM limit provide the system designer the ability to determine whether a RAM upgrade will result in better deployment pattern.

- The simulation model also provides the system designer the ability to model the effect of growth in the number of roles and the effect of microprocessor upgrade.
VI. VERIFICATION EXPERIMENT 2

A. INTRODUCTION

The second verification experiment is a new scenario with 3 machine and 10 object servers. The number of deployment patterns is 59,049. The objective of this verification exercise is to verify the robustness of the simulation model with high number of deployment patterns. The same environment profile is also used on the mathematical model and the result of the two models is compared.

If a testbed methodology is used to find the optimal deployment for this scenario, the time needed to test all 59049 deployment pattern is $59049 \times 8\text{hrs} = 472392\text{ hrs}$ (i.e. 53 years). In the simulation model, a 8hrs simulation time can be simulated within 20 secs, therefore for 59049 deployment pattern the time needed is $59049 \times 20\text{sec} = 328\text{hrs}$ (i.e. 13 days). The simulation model has reduced the time needed by 1490%.

To further reduce the number of deployment patterns to simulate, the model has bound by setting the ram limit to 1.0. With the bounded ram limit, any deployment pattern that uses more RAM than a machine possessed is rejected. This is a reasonable assumption since the memory swapped time will contribute significantly to the response time and therefore will not result in the optimal response time. For this experiment, the number of deployment pattern is reduced to 1121.

B. SCENARIO PROFILE

The various profiles used in defining the environment is shown in the following table:

<table>
<thead>
<tr>
<th>Machine</th>
<th>RAM (MB)</th>
<th>CPU Speed (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mozart</td>
<td>256</td>
<td>2000</td>
</tr>
<tr>
<td>Handel</td>
<td>512</td>
<td>2400</td>
</tr>
<tr>
<td>Beethoven</td>
<td>1000</td>
<td>3000</td>
</tr>
</tbody>
</table>

Table 34. Machine Profile
<table>
<thead>
<tr>
<th>OBJECT SERVER</th>
<th>METHOD</th>
<th>Number of instruction cycle</th>
<th>Average Size of Message (bits)</th>
<th>RAM Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1579000</td>
<td>216000</td>
<td>139</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>1140000</td>
<td>2296000</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>1599000</td>
<td>1736000</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>1394000</td>
<td>528000</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>705000</td>
<td>2752000</td>
<td>122</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1740000</td>
<td>736000</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>243000</td>
<td>1968000</td>
<td>145</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>702000</td>
<td>2824000</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>1892000</td>
<td>3728000</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>490000</td>
<td>3752000</td>
<td>153</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>1445000</td>
<td>1912000</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>1315000</td>
<td>3640000</td>
<td>231</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>1437000</td>
<td>2984000</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td>1108000</td>
<td>2616000</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1286000</td>
<td>2800000</td>
<td>130</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>1528000</td>
<td>2064000</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>367000</td>
<td>3328000</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>4</td>
<td>1750000</td>
<td>3632000</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>5</td>
<td>1802000</td>
<td>3800000</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>845000</td>
<td>2968000</td>
<td>142</td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>1437000</td>
<td>104000</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>950000</td>
<td>3848000</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>1395000</td>
<td>2552000</td>
<td>200</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
<td>1352000</td>
<td>3384000</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>2215000</td>
<td>1336000</td>
<td>189</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>1201000</td>
<td>392000</td>
<td>80</td>
</tr>
<tr>
<td>J</td>
<td>2</td>
<td>1606000</td>
<td>2624000</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>557000</td>
<td>856000</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>4</td>
<td>1101000</td>
<td>248000</td>
<td></td>
</tr>
</tbody>
</table>

Table 35. Object Server Performance Profile

<table>
<thead>
<tr>
<th>Primary Method</th>
<th>Secondary Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.2</td>
<td>A.2</td>
</tr>
<tr>
<td>J.2</td>
<td>B.1</td>
</tr>
<tr>
<td>J.3</td>
<td>I.1</td>
</tr>
<tr>
<td>E.1</td>
<td>F.5</td>
</tr>
<tr>
<td>H.1</td>
<td>G.3</td>
</tr>
</tbody>
</table>

Table 36. Complex Object Server Call Profile
### Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Call Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role 0</td>
<td>50 F1.B1 + 1 F1.B2 + 1 F2.B1 + 1 F2.B4</td>
</tr>
<tr>
<td>Role 3</td>
<td>30 F2.B3</td>
</tr>
</tbody>
</table>

Table 37. Role Call Pattern

### Buttons

<table>
<thead>
<tr>
<th>Buttons</th>
<th>Method Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1.B1</td>
<td>G.2 + C.2 + A.1</td>
</tr>
<tr>
<td>F1.B2</td>
<td>F.3</td>
</tr>
<tr>
<td>F1.B3</td>
<td>H.2</td>
</tr>
<tr>
<td>F2.B1</td>
<td>E.2 + G.2</td>
</tr>
<tr>
<td>F2.B2</td>
<td>J.4 + B.2</td>
</tr>
<tr>
<td>F2.B3</td>
<td>D.2</td>
</tr>
<tr>
<td>F2.B4</td>
<td>B.2 + E.1</td>
</tr>
<tr>
<td>F3.B1</td>
<td>J.3</td>
</tr>
<tr>
<td>F3.B2</td>
<td>F.2 + B.1 + E.2 + F.3</td>
</tr>
<tr>
<td>F4.B1</td>
<td>J.2 + A.2</td>
</tr>
<tr>
<td>F5.B1</td>
<td>H.1 + F.3 + J.3</td>
</tr>
<tr>
<td>F5.B2</td>
<td>B.1 + E.2</td>
</tr>
<tr>
<td>F5.B3</td>
<td>I.1 + J.2</td>
</tr>
<tr>
<td>F5.B4</td>
<td>D.1 + I.1</td>
</tr>
<tr>
<td>F5.B5</td>
<td>I.1</td>
</tr>
</tbody>
</table>

Table 38. User Interface Call Chart

### C. MODEL SETUP

#### 1. Simulation Model Setup

The configuration file in the simulation model is shown in Appendix D.A. A memory leak was encountered with the OMNet++ while executing the 59049 simulation model. The problem was highlighted to the author of OMNet++. To circumvent the problem, the simulation run was divided down into 6 run with 10000 runs each. A batch program was used to automate the running of the 6 runs.
2. **Lingo Model Setup**

A new Lingo model is created and is shown in Appendix D.B. The Lingo model created supports the computation based on instruction cycle instead of time.

**D. WEIGHED MODEL**

For an environment with only one role type, the average response time is used to determine the optimal deployment pattern. However, when there is more than one role type, using the average response time of a role may penalize other roles. There are a few methods to determine the optimal deployment pattern:-

1. **Ranking**

The optimal deployment pattern can be determine by ranking the average response time for each role and comparing the top \( n \) deployment pattern and determine the pattern that appears most frequent in all the roles.

2. **Highest Priority**

The optimal deployment pattern can be determine by selecting the highest priority role and select the lowest average response time. This will ensure that deployment pattern selected will favor the role with the highest priority.

3. **Weighted Model**

The third method is to assigned weighs to the role response time. Instead of selecting the role highest priority, the weighted model will assign weightage to all roles. The weights can be derived based on the criticality of the machine, and the required response time. For example, the weight of a command and control system (C2) should be higher than a manpower system simply because the time critical nature of C2 system. The weight is used to compute a model value based on the following formula:-
The deployment pattern with the smallest model value will be selected as optimal. Some examples of weight used are:

<table>
<thead>
<tr>
<th>S/No</th>
<th>Weight</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>[0.25, 0.25, 0.25, 0.25]</td>
<td>The optimal deployment pattern selected will have placed equal weight to each role.</td>
</tr>
<tr>
<td>02</td>
<td>[0.0, 0.0, 1.0, 0.0]</td>
<td>The emphasis is placed on Role3. The deployment pattern selected will have the lowest response time for Role3. The result is similar to the highest priority method.</td>
</tr>
<tr>
<td>03</td>
<td>[0.1, 0.2, 0.2, 0.5]</td>
<td>An unequal weighted scheme with emphasis placed on Role4.</td>
</tr>
</tbody>
</table>

Table 39. Example of Weighted Model

E. **COMPARING SIMULATION RESULT WITH LINGO MODEL**

1. **Equal Weigh Model**

The simulation and Lingo model was run and the equal weighted model is used to determine the optimal deployment pattern for the simulation result. The detail result of the simulation model is shown in Appendix D. The optimal deployment pattern is shown on the following table:-
Both results have spread the object server on all machines. Based on the response time generated by the simulation model, the Lingo model has picked a deployment pattern that is rank 149 positions. Although, the ranking differs significantly, the difference in the response time between the two results is small. The difference in response time in between the two patterns in shown below:

<table>
<thead>
<tr>
<th>Role</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2.5</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Table 40. Deployment Pattern
2. Effect of Weigh Model

Different scenario with different weigh model is used to compute the effect on the deployment pattern. The computation is done using an Excel spreadsheet. The result is shown in the following table:-

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.25, 0.25, 0.25, 0.25]</td>
<td>[Bee] [Moz] [Bee] [Bee] [Han] [Han] [Bee] [Bee] [Moz] [Bee] [Bee] [Moz] [Bee] [Bee] [Moz]</td>
</tr>
<tr>
<td></td>
<td>[2.6609, 1.7026, 2.4717, 1.9685]</td>
</tr>
<tr>
<td>[0.0, 1.0, 0.0, 0.0]</td>
<td>[Han] [Bee] [Bee] [Bee] [Han] [Bee] [Moz] [Han] [Bee] [Bee] [Moz]</td>
</tr>
<tr>
<td></td>
<td>[2.9501, 1.2959, 3.1182, 2.5366]</td>
</tr>
<tr>
<td>[0.1, 0.2, 0.2, 0.5]</td>
<td>[Bee] [Han] [Bee] [Bee] [Han] [Han] [Bee] [Bee] [Moz]</td>
</tr>
<tr>
<td></td>
<td>[2.9601, 1.4188, 2.9301, 1.8381]</td>
</tr>
<tr>
<td>[0.0, 0.2, 0.3, 0.5]</td>
<td>[Bee] [Han] [Moz] [Bee] [Han] [Han] [Bee] [Han] [Bee] [Bee] [Moz]</td>
</tr>
<tr>
<td></td>
<td>[3.0716, 1.9974, 2.3948, 1.8735]</td>
</tr>
</tbody>
</table>

Table 41. Weighted model
The scenario with different weigh model has shown that there is an effect on the deployment pattern in response time of the system. The weighted method is effective to determine the optimal deployment pattern based on difference priority.

F. CONCLUSION
The second verification exercise has shown that the simulation model is robust in a profile with high number of deployment patterns. The Lingo and simulation model has predicted a deployment pattern that has average response time that is close to each other. The verification exercise has also introduced and verified the use of a weighted model to determine the best optimal deployment pattern in an environment with more than one role.
VII. RECOMMENDATION AND FUTURE WORKS

Although the result from two experiments has verified that the simulation model has modeled closely the distributed object server environment, there are multiple ways that the model could be improved.

A. MODELING OF SYSTEM USING UML

The current configuration files are in textual format, and are generated manually by the system designer. This is a tedious task especially with complex environment. It will be more efficient to model the environment profiles as part of the system design process. As UML is becoming the de-facto standard for object oriented design methodology, future work in integrating the simulation model into the UML modeling will be useful. The UML deployment and class diagram can be translated into information needed in the various profiles. Tagged values can be added to the deployment diagram to represents machine profile such as CPU speed, RAM limit, network bandwidth etc. Tagged values can also be added to the class diagram to represents class profile such as complex method calls, computation time and the return message size.

There are existing tools that can convert the UML diagrams to XMI schemas or DTD (Document Type Definition). By changing the simulation model to support XML format, integration the simulation model with UML is possible.

B. IMPROVED NETWORK MODELING

The current simulation model only implements a simple network model. Future work can be done to improve the network model by implementing the different layering of network deployment. The model can be improved by modeling network protocol such as ATM, 802.11 and TCP/IP. The model can also be enhanced by allowing system designer to specify the network architecture of the system. For example, the system designer can specify the network domain, router and firewall deployment.
The current network model also assumes a symmetric and error free network between the role and the machine. However, the real-world network configuration could be asymmetric in nature. For example, DSL and broadband network has asymmetric communication link. Machine to machine communication within an enclave will potentially have a higher bandwidth than communication with host outside an enclave. Building support for asymmetric communication will improve the accuracy of the model.

Network error rate will have significant impact in the overall performance of a distributed system. A network segment with a high error rate will result in more loss packet, higher retransmission and thus longer response time. Therefore, the optimal deployment strategy may avoid deploying object server on machine that rides on a network segment with higher network error rate. To model the effect of network error, the following work must be done:

- Model the error handling mechanism by the network protocol e.g. TCP/IP error handling protocol.
- For every message send or received, model the probability of error using the network error rate.
- For message with error, retransmit or request for the message again.

C. REFINING THE RESULT TO ROLE/METHOD CALL LEVEL

The simulation model determines the optimal deployment pattern based on average response time. However, each method call may not have equal priority. A method call for real-time data update has a higher priority than for static data update. Real-time data update will have a more stringent demand for the response time. For roles with many low priority calls, using the average response time to select a deployment pattern may not be realistic. Ignoring low priority call altogether will also present an unrealistic load on the machine CPU. Therefore, there is a requirement to refine the simulation model to model timing
priority for method call and determine the optimal deployment based on high priority method call.

D. ENTERPRISE RESOURCE PLANNING (ERP)

Enterprise resource planning (ERP) capabilities can be added to the simulation model to enable the system designer to study the effect and expenses of adding memory, upgrading the CPU or adding additional machines. ERP also allows the system designer to study the effect of reducing the number of machines due to machine failure. The simulation model could advise system designer the most cost-effective upgrade that would give the most performance improvement for the least amount of money.

E. CONSTRAINTING THE DEPLOYMENT PATTERNS

Further works to explore other constraints to the deployment patterns. The aim of introducing more constraints is to reduce the time needed to execute the simulation. Besides limiting the deployment patterns based on RAM limit, the following constraints can be further explored:

- Constraint based on the CPU limit. Deployment patterns with CPU resource utilization exceeding the preset CPU limit will be rejected.
- Constraint based on probable inferior deployment. For example, deployment pattern which does not involve the most powerful CPU will likely not be the optimal provable deployment.
- Constraint based on network bandwidth. For example, with two identical CPU, the CPU with the higher network bandwidth with the role should be favored.
- Constraint based on user defined deployment pattern preference. Specific machine are used to run legacy application due to system and interface constraint. For example, application written for
Windows 95 environment will not be portable to a machine running WindowsXP.
VII. CONCLUSION

The response time predicted by the simulation model is quite close to the scenario tested. The simulation model is also able to consistently select the top few deployment patterns. Verification experiment one has shown that the simulation model prediction matches the test bed better than the mathematical model.

There are other strengths in the simulation model. First, the simulation model provides a good overview of the response time of the system. The simulation model provides information such as maximum, minimum and average response time of a role. This information is a good benchmark for system designer to set a reasonable expectation of the response time. The model also provides the response time of other deployment pattern other than the optimal deployment, thus allowing the system designer to study the tradeoff of selecting other deployment pattern. For example, if the second optimal deployment meets the operational requirement, it may be favor over the optimal pattern because of deployment preference.

The simulation model is also a valuable tool for the system designer to determine the effect of growth in the number of roles. Since the simulation run is easily repeatable, the system engineer can generate the response time based on many scenario of increasing role count. The maximum number of roles is derived when the response time is greater than the operational requirement. In a deployment environment which the number of roles is constantly changing, the optimal deployment pattern of the highest number of roles should be used to avoid the need to constantly change the deployment pattern.

System designer can use the simulation model to determine the effect of system upgrade i.e. upgrading CPU speed or RAM. For example, the result of the simulation run with no RAM limit can be used to study the effect of upgrading the RAM of the system.
Lastly the simulation model can also be used for “what-if” scenario in the event of system failure. For example, the simulation model allows the system designer to determine what the next best deployment pattern is if one of the CPU fails.

The limitation of the simulation model compared to the mathematical model is the time needed to run the simulation. For verification exercise 2, the simulation run took about 1 day to complete but the mathematical model took about 30 secs to complete. The additional time needed is justifiable with the functionalities mentioned above.
APPENDIX A - SOURCE CODE OF THE SIMULATION MODEL

A. ADDRESS.CPP

// Address.cpp: implementation of the CAddress class.

/////////////////////////////////////////////////////////////////////
#include "Address.h"
#include <string.h>

/////////////////////////////////////////////////////////////////////
// Construction/Destruction
/////////////////////////////////////////////////////////////////////

CAddress::CAddress()
{
    in_gate = -1;
    out_gate = -1;
}

CAddress::~CAddress()
{
}

void CAddress::SetAddress(int ingate, int outgate, char *svr, char *objapp)
{
    strcpy(svr_name, svr);
    strcpy(obj_name, objapp);
    in_gate = ingate;
    out_gate = outgate;
}

int CAddress::GetInGateId()
{
    return in_gate;
}

int CAddress::GetOutGateId()
{
    return out_gate;
}

char *CAddress::GetName()
{
    return obj_name;
}

B. BTNLIST.CPP

/////////////////////////////////////////////////////////////////////
// file: btnlist.cpp
```c
#include <stdio.h>
#include <string.h>
#include "omnetpp.h"
#include "global.h"
#include "simevent.h"
#include "btnlist.h"

static CBtnList *m_ButtonList = NULL;
CBtnList *CBtnList::GetInstance()
{
    if (m_ButtonList==NULL)
    {
        m_ButtonList = new CBtnList();
        m_ButtonList->initialize();
    }
    return m_ButtonList;
}

CBtnList::CBtnList()
{
}

CBtnList::~CBtnList()
{
}

bool CBtnList::initialize()
{
    buttonCount = 0;
    loadCfg(); //Loading configuration file
    return true;
}

bool CBtnList::getButton(char *name, sBUTTON &button)
{
    char tmpstr[MAX_NAME_SZ];
    bool found = false;
    int i = 0;

    while (!found && (i<buttonCount))
    {
        strcpy(tmpstr, m_Button[i].cName);
        if (strcmp(name, tmpstr) == 0)
        {
            button = m_Button[i];
            found = true;
        }
        i = i+1;
    }

    if (found)
        return true;
    else
        return true;
```
return false;
}

// Method to load the configuration file
bool CBtnList::loadCfg()
{
    char seps[] = "\n";
    char *token;

    FILE *stream;
    char line[100];
    int pos, len;
    bool endDef;

    sBUTTON tmpButton;

    /* Open for read (will fail if file "data" does not exist) */
    if( (stream = fopen( "button.def", "r" )) == NULL )
    {
        ev << "[CBtnList] >> The file 'data' was not opened\n";
        return false;
    }
    else
    {
        ev << "[CBtnList] >> The file 'data' was opened\n";

        while( fgets( line, 100, stream ) != NULL)
        {
            // checking the content of the string
            #ifdef DEBUG_FLAG_L1
            ev << "[CBtnList] >> Content " << line << '\n';
            #endif
            endDef = false;
            if (strstr( line, "[NEW_DEF]" ) != NULL)
            {
                // New definition of button, next line is name
                if ( fgets( line, 100, stream ) != NULL)
                {
                    strcpy(tmpButton.cName, line);
                    len = strlen(tmpButton.cName);
                    tmpButton.cName[len-1] = '0';
                    tmpButton.iNumCall = 0;
                }
            }
            while (!endDef)
            {
                if(fgets( line, 100, stream) == NULL)
                    return false;
                if (strstr( line, "[END_DEF]" ) != NULL)
                {
                    m_Button[buttonCount] = tmpButton;
                    buttonCount = buttonCount + 1;
                    endDef = true;
                }
            }
        }
    }
}

83
else
{

    /* Establish string and get the first token: */
    pos = 0;
    token = strtok( line, seps );

    while( token != NULL )
    {
        switch (pos)
        {
        case 0: //Getting the object server name
            ifdef DEBUG_FLAG_L1
                ev << "[CBtnList] >> ObjSvr " << token << '\n';
            endif
                strcpy(tmpButton.svcCall[tmpButton.iNumCall].cObjsvr, token);
                pos++;
                break;
            case 1: //Getting the method name
                ifdef DEBUG_FLAG_L1
                    ev << "[CBtnList] >> cMethod " << token << '\n';
                endif
                    strcpy(tmpButton.svcCall[tmpButton.iNumCall].cMethod, token);
                    pos++;
                    break;
                }
        token = strtok( NULL, seps );
    }
    tmpButton.iNumCall = tmpButton.iNumCall + 1;
}

/* Close stream */
if( fclose( stream ) )
    ev << "[CBtnList] >> The file 'data' was not closed\n";
    return true;
}

return false;

C. DNSSVC.CPP

// DnsSvc.cpp: implementation of the CDnsSvc class.
///
//=====================================================================

#include <string.h>
#include "DnsSvc.h"
CDnsSvc::CDnsSvc()
{
    counter = 0;
}

CDnsSvc::~CDnsSvc()
{
}

bool CDnsSvc::MapSvcToPort(int ingate, int outgate, char *svr, char *name)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<counter; i++)
    {
        strcpy(tmpstr, m_AddMap[i].GetName());
        if (strcmp(name, tmpstr) == 0)
        {
            //name found, there might be a shift in the ingate and outgate
            m_AddMap[i].SetAddress(ingate, outgate, svr, name);
            return true;
        }
    }
    m_AddMap[counter].SetAddress(ingate, outgate, svr, name);
    counter++;
    return true;
}

bool CDnsSvc::FindName(char *name)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<counter; i++)
    {
        strcpy(tmpstr, m_AddMap[i].GetName());
        if (strcmp(name, tmpstr) == 0)
        {
            return true;
        }
    }
    return false;
}

int CDnsSvc::FindInGateByName(char *objs vr)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<counter; i++)
    {
        strcpy(tmpstr, m_AddMap[i].GetName());
    }
if (strcmp(objsvr, tmpstr) == 0)
    return m_AddMap[i].GetInGateId();
}
return -1;

int CDnsSvc::FindOutGateByName(char *objsvr)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<counter; i++)
    {
        strcpy(tmpstr, m_AddMap[i].GetName());
        if (strcmp(objsvr, tmpstr) == 0)
            return m_AddMap[i].GetOutGateId();
    }
    return -1;
}

D. INTERACTLIST.CPP

// InteractList.cpp: implementation of the CInteractList class.
//
/////////////////////////////////////////////////////////////////////
#include <stdio.h>
#include <string.h>
#include "limits.h"
#include "omnetpp.h"
#include "InteractList.h"
/////////////////////////////////////////////////////////////////////
// Construction/Destruction
/////////////////////////////////////////////////////////////////////
static CInteractList *m_Interact = NULL;
CInteractList *CInteractList::GetInstance()
{
    if (m_Interact==NULL)
    {
        m_Interact = new CInteractList();
        m_Interact->initialize();
    }
    return m_Interact;
}
CInteractList::CInteractList()
{
}
CInteractList::~CInteractList()
//To initialize the InteractList.
bool CInteractList::initialize()
{
    m_InteractCount = 0;
    loadCfg();    //Loading configuration file
    return true;
}

//To get an interact from the object server name provided.
bool CInteractList::getInteract(char *objsvr, char *method, slInteract &interact)
{
    int index = 0;
    bool found = false;

    while ((index < m_InteractCount) && (!found))
    {
        if ((strcmp(m_InteractList[index].initCall.cObjsvr, objsvr) == 0) &&
            (strcmp(m_InteractList[index].initCall.cMethod, method) == 0))
        {
            found = true;
            interact = m_InteractList[index];
            return true;
        }
        index = index + 1;
    }
    return false;
}

//Method to load the configuration file
bool CInteractList::loadCfg()
{
    char seps[] = "/\n";
    char *token;

    FILE *file;
    char line[100];
    int pos = 0;
    bool endDef = false;

    slInteract tmpInteract;
    int tmpInt = 0;
    char tmpStr[100];
    char buf[1024 + 1];

    if( (file = fopen( "interact.def", "r+t" )) == NULL )
    {
        printf( "The file 'interact.def' was not opened\n" );
        return false;
    }

    #ifdef DEBUG_FLAG_L2
ev << "[CInteractList] >> The file 'data' was opened\n";
#endif

tmplInteract.iNumInteract = 0;
while (fgets(buf, 1024, file))
{
  if (strstr (buf, "[NEW_INTERACT]") != NULL)
  {
    strcpy(tmpStr, buf+strlen("[NEW_INTERACT]")+1);
    tmpStr[strlen(tmpStr)-1] = '\0';
    token = strtok( tmpStr, seps );
    while( token != NULL )
    {
      switch (pos)
      {
        case 0: //Getting the object server name of the method
          #ifdef DEBUG_FLAG_L1
          ev << "[CInteractList] >> ObjSvr " << token << '\n';
          #endif
          strcpy(tmpInteract.initCall.cObjsvr, token);
          pos++; break;
        case 1: //Getting the method name
          #ifdef DEBUG_FLAG_L1
          ev << "[CInteractList] >> cMethod " << token << '\n';
          #endif
          strcpy(tmpInteract.initCall.cMethod, token);
          pos++; break;
      }
      token = strtok( NULL, seps );
    }
  }
  tmpInt = 0;
  while (!endDef)
  {
    if (fgets(line, 100, file) == NULL)
      return false;
    if (strstr( line, "[ END_INTERACT]") != NULL)
    {
      tmpInteract.iNumInteract = tmpInt;
      m_InteractList[m_InteractCount] = tmpInteract;
      m_InteractCount = m_InteractCount + 1;
      endDef = true;
    }
    else if (strstr( line, "[CALL]") != NULL)
    {
      strcpy(tmpStr, line+strlen("[CALL]")+1);
      tmpStr[strlen(tmpStr)-1] = '\0';
      /* Establish string and get the first token: */
      pos = 0;
      token = strtok( tmpStr, seps );
    }
while ( token != NULL )
{
    switch (pos)
    {
        case 0: //Getting the object server name
            #ifdef DEBUG_FLAG_L1
                ev << "[CInteractList] >> ObjSvr " << token << '\n';
            #endif
            strcpy(tmpInteract.interactCall[tmpInt].cObjsvr, token);
            pos++;
            break;
        case 1: //Getting the method name
            #ifdef DEBUG_FLAG_L1
                ev << "[CInteractList] >> cMethod " << token << '\n';
            #endif
            strcpy(tmpInteract.interactCall[tmpInt].cMethod, token);
            pos++;
            break;
    }
    token = strtok( NULL, seps );
}
}

/* Close stream */
if( fclose( file ) )
    ev << "[CInteractList] >> The file 'data' was not closed\n";

    return true;
}

return false;

E. METHOD.CPP

// method.cpp: implementation of the Cmethod class.
//

/////////////////////////////////////////////////////////////////////
#include <string.h>
#include "method.h"

/////////////////////////////////////////////////////////////////////

CMethod::CMethod()
{
CMethod::~CMethod()
{
}

void CMethod::setMethod(char *str, long cpu, long msg)
{
    strcpy(name, str);
    cpuUtil = cpu;
    msgSz = msg;
}

F. OBJ.CPP

// obj.cpp: implementation of the obj class.
//
/////////////////////////////////////////////////////////////////////
#include <string.h>
#include "obj.h"
/////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////
// Construction/Destruction
/////////////////////////////////////////////////////////////////////

CObj::CObj()
{
}

CObj::~CObj()
{
}

// To retrieve the CPU utilization requirement given the object server name
long CObj::getCPU(char *name)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<num_method; i++)
    {
        strcpy(tmpstr, method[i].name);

        // if both string is the same
        if (strcmp(name, tmpstr) == 0)
            return method[i].cpuUtil;
    }
    return -1;
}
// To retrieve the message size
long CObj::getMsgSize(char *name)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<num_method; i++)
    {
        strcpy(tmpstr, method[i].name);

        // if both string is the same
        if (strcmp(name, tmpstr) == 0)
            return method[i].msgSz;
    }
    return -1;
}

G. OBJLIST.CPP

#include <stdio.h>
#include <string.h>
#include "limits.h"
#include "omnetpp.h"
#include "ObjList.h"

static CObjList *m_ObjList = NULL;
CObjList *CObjList::GetInstance()
{
    if (m_ObjList==NULL)
    {
        m_ObjList = new CObjList();
        m_ObjList->initialize();
    }
    return m_ObjList;
}

CObjList::CObjList()
{
}

CObjList::~CObjList()
{
}
// To initialize the ObjList.
bool CObjList::initialize()
{
    m_ObjCount = 0;
    loadCfg(); // Loading configuration file
    return true;
}

// To get an object using the object name
bool CObjList::getObj(char *name, CObj &obj)
{
    char tmpstr[MAX_NAME_SZ];
    bool found = false;
    int i = 0;

    while ((i<m_ObjCount) && (!found))
    {
        strcpy(tmpstr, m_Obj[i].name);
        if (strcmp(name, tmpstr) == 0)
        {
            obj = m_Obj[i];
            found = true;
        }
        i = i + 1;
    }

    if (found)
        return true;
    else
        return false;

    return true;
}

// Method to load the configuration file
bool CObjList::loadCfg()
{
    char seps[] = "\n";
    char *token;

    FILE *stream;
    char line[100];
    int pos;
    bool endDef;
    CObj tmpObj;
    long tmpLong1, tmpLong2;
    char tmpName[MAX_NAME_SZ];

    /* Open for read (will fail if file "data" does not exist) */
    if( (stream = fopen( "obj.def", "r" )) == NULL )
    {
        ev << "[CObjList] >> The file 'data' was not opened\n";
        return false;
    }
    else


```c
{ 
    ev << "[CObjList] >> The file 'data' was opened
"

    while ( fgets( line, 100, stream ) != NULL) 
    { 
        // checking the content of the string 

    #ifdef DEBUG_FLAG_L1
        ev << "[CObjList] >> Content " << line << '"';
    #endif

        endDef = false;
        if (strstr( line, "[NEW_OBJ_DEF]" ) != NULL) 
        { 
            // New definition of button, next line is name 
            if ( fgets( line, 100, stream ) != NULL) 
            { 
                pos = 0;
                tmpObj.num_method = 0;
                token = strtok( line, seps );

                while ( token != NULL ) 
                { 
                    switch ( pos ) 
                    { 
                    case 0: // Getting the object server name 
    #ifdef DEBUG_FLAG_L1
                        ev << "[CObjList] >> ObjSvr " << token << '"';
    #endif
    strcpy(tmpObj.name, line);
                pos++;
                break;

                    case 1: // Getting the method name 
    #ifdef DEBUG_FLAG_L1
                        ev << "[CObjList] >> cMethod " << token << '"';
    #endif
    tmpObj.ram_sz = atoi(token);
                pos++;
                break;

                    } 
                    token = strtok( NULL, seps );
                } 

            while (!endDef) 
            { 
                if (fgets( line, 100, stream) == NULL) 
                    return false;

                if (strstr( line, "[END_OBJ_DEF]" ) != NULL) 
                { 
                    m_Obj[m_ObjCount] = tmpObj;
                    m_ObjCount = m_ObjCount + 1;
                    endDef = true;
                } 
                else 
                { 
```
/* Establish string and get the first token: */
pos = 0;
token = strtok(line, seps);

while (token != NULL)
{
    switch (pos)
    {
    case 0: //Getting the object server name
        #ifdef DEBUG_FLAG_L1
        ev << "[CObjList] >> ObjSvr " << token << "\n";
        #endif
        strcpy(tmpName, token);
pos++;
break;
    case 1: //Getting the method name
        #ifdef DEBUG_FLAG_L1
        ev << "[CObjList] >> cMethod " << token << "\n";
        #endif
        if (atol(token) <= LONG_MAX)
            tmpLong1 = atol(token);
        else
            { tmpLong1 =
                ev << "[CObjList] >> larger than expected value received! \n";
                pos++;
break;
        case 2: //Getting the method name
        #ifdef DEBUG_FLAG_L1
        ev << "[CObjList] >> cMethod " << token << "\n";
        #endif
        tmpLong2 = atol(token);
break;
        }
    token = strtok(NULL, seps);
}
tmpObj.method[tmpObj.num_method].setMethod(tmpName, tmpLong1, tmpLong2);
tmpObj.num_method = tmpObj.num_method + 1;
}
/* Close stream */
if (fclose(stream))
    ev << "[CObjList] >> The file 'data' was not closed\n";
delete(token);
return true;
```cpp
return false;
}

H.  SIMEVENT.CPP

////////////////////////////////////////////////////////////////////////
// file: SimEvent.cpp
////////////////////////////////////////////////////////////////////////
#include "omnetpp.h"
#include "global.h"
#include "simevent.h"

CSimEvent::CSimEvent()
{
    numEvent = 0;
    maxPro = 0;
}

CSimEvent::~CSimEvent()
{
}

// Each added event will be indexed and the probability is stored. 
// The maxPro is computed upon each additional event. This maxPro
// is used for the random uniform distribution number picked in the
// getNextEvent()
void CSimEvent::addEvent(int index, int pro)
{
    prob[index] = pro;
    maxPro = pro + maxPro;
    numEvent = numEvent + 1;
}

// Is to get the next event. The index return will be used in order
// to determine which button is pressed.
int CSimEvent::getNextEvent()
{
    int state = 0;

    // Generate a random discrete number
    int num = intuniform(0, maxPro);

    for (int i=0; i<numEvent; i++)
    {
        state = state + prob[i];
    }

    #ifdef DEBUG_FLAG_L2
    ev << "[CSimEvent] state >> " << state << "\n";
    ev << "[CSimEvent] num >> " << num << "\n";
    #endif

    if (num <= state)
    {
```
I. SMACHINE.CPP

/////////////////////////////////////////////////////////////////////
// file: sMachine.cpp
/////////////////////////////////////////////////////////////////////

#include <stdio.h>
#include <string.h>
#include "omnetpp.h"
#include "interactlist.h"
#include "global.h"
#include "statlog.h"

class sMachine : public cSimpleModule
{
  Module_Class_Members(sMachine,cSimpleModule,16384)
  virtual void finish();
  virtual void activity();
  virtual void initialize();

protected:
  bool isSwitchGateId(int id);
  bool isObjSvrGateId(int id);
  void InitObjSvrName();
  void AddObjSvrName(char *objsvr, int index);
  int FindObjSvrName(char *objsvr);

  bool initProcess();
  bool addNewProcess(cMessage *msg);
  int findProcessByRequestId(int id);
  bool execNextProcess();
  bool deleteProcess(int id);

  bool initRemoteRequestList();
  int addNewRemoteRequest(int instIndex);
  int findRemoteRequest(int index);
  bool recRemoteResponse(int index);

  double getWaitTime(double cpu, int osid);
  double getWaitTime();
  bool writeData(char *str);
  bool loadDef();

  bool isTimeSlice;
  long cpuPow, ramSz, dataRate2Sw;
  int prevObjSvrlid;
Define_Module(sMachine);

void sMachine::initialize()
{
    //initialize the list gateid
    m2osin_gt = gate("m2os_in");
    m2swin_gt = gate("m2sw_in");
    m2swout_gt = gate("m2sw_out");

    m2osin_sz = m2osin_gt->size();
    if (m2osin_sz > MAX_GATE_SZ)
    {
        ev << "[!! Server] >> Maximum number of gate defined (m2osin_sz) " << m2osin_sz << '\n';
        m2osin_sz = MAX_GATE_SZ;
    }

    for (int i=0; i<m2osin_sz; i++)
    {
        cGate *tmp = gate("m2os_in", i);
        m2osin_id[i] = tmp->id();
    }
}
#endif DEBUG_FLAG_L1
ev << "[sMachine] >> ObjSvr (i) " << i << " (gateid) " << tmp->id() << "\n";
#endif

//initializing the list of server to switch gate id
m2swin_sz = m2swin_gt->size();
if (m2swin_sz > MAX_GATE_SZ)
{
    ev << "[!! sMachine] >> Maximum number of gate defined (m2swin_sz) " << m2swin_sz << "\n";
    m2swin_sz = MAX_GATE_SZ;
}
for (i=0; i<m2swin_sz; i++)
{
    cGate *tmp = gate("m2sw_in", i);
    m2swin_id[i] = tmp->id();
#endif DEBUG_FLAG_L1
    ev << "[sMachine] >> Switch (i) " << i << " (gateid) " << tmp->id() << "\n";
#endif

cpuPow = 0;
ramSz = 0;
ramUsed = 0;
processSwapTime = 0.0;
diskSwapTime = 0.0;
prevObjSvrId = -1;

InitObjSvrName();

//Getting the configuration file name and loading the configuration
cPar &tmpPar = par("def_file");
strcpy(defFile, tmpPar.stringValue ());
isTimeSlice = false;

tmpPar = parentModule()->par("run_name");
strcpy(runName, tmpPar.stringValue ());
ev << "[sRole] >> Initializing " << runName << "\n";
loadDef();

//Getting the configuration file name
setName(machineName);

//Initialize the process handling list.
initProcess();

//To set the data rate.
tmpPar.setLongValue (dataRate2Sw);
m2swin_gt->setDataRate(&tmpPar);
m2swout_gt->setDataRate(&tmpPar);

//Initialize the interact list
CInteractList::GetInstance();

98
// To initialize the request list.
initRemoteRequestList();
}

// To handle the finish event
void sMachine::finish()
{
    delete(m2osin_gt);
    delete(m2swin_gt);
    delete(m2swout_gt);
}

void sMachine::activity()
{
    double clock_speed = 1.0;
    double process_time = 0.0;
    double avg_utilization = 0.0;

    int type, objsvr_id, msgSz, requestId, index, tmpRam;
    bool newMsg = false;

    cPar tmpPar;
    cPar cpObjSvrName, cpMethodName, cpRequestorName;
    cMessage *svr_resp, *m_resp, *obj_msg;
    cGate *rcv_gate;
    cOutVector resp_v("CPU");

    // Waiting code
    process_time = uniform(1.0, 3.0);
    wait(process_time);

    char tmpStr[MAX_NAME_SZ];
    for(;;)
    {
        // Listen for incoming message
        newMsg = false;
        cMessage *rcv_msg = receive(0.01);

        if (rcv_msg != NULL)
        {
            newMsg = true;
            type = rcv_msg->kind(); // Message kind
            rcv_gate = rcv_msg->arrivalGate();

            #ifdef DEBUG_FLAG_L2
            ev << "[sMachine] >> receiving msg type " << type << "\n";
            #endif

            // Message is from the m2sw_in port.
            if (isObjSvrGateId(rcv_gate->id()))
            {
                switch (type)
                {
                    case OBJSVR_REGISTER:
server

//OBJSVR register message to be sent over to the DNS

NAME_REGISTER);

// Forward the information to the server
svr_resp = new cMessage("NAME_REGISTER", NAME_REGISTER);

svr_resp->addPar("add_machine") = machineName;
svr_resp->addPar("add_name") = rcv_msg->par("add_name");
svr_resp->addPar("add_obj") = rcv_msg->par("add_obj");
tmpRam = (int)rcv_msg->par("ram_total");

//if the ramLimit is set to 0.0, we will compute the
//best pattern without considering the ram size.
if (ramLimit != 0.0)
{
    ramUsed = ramUsed + tmpRam;
    totalRamLimit = ramLimit*ramSz;

    // checking for the amount of ram used.
    if (ramUsed > totalRamLimit)
    {
        // the amount of ram used is higher than
        // what is currently
        CStatLog::GetInstance()
            ->writeLogWithRamError(runName, totalRamLimit, ramUsed);
        endSimulation();
    }
}

send(svr_resp, "m2sw_out");

#ifdef DEBUG_FLAG_L2
    ev << ", [sMachine] >> Sending NAME_REGISTER response\n" << \n;
#endif

#ifdef DEBUG_FLAG_L2
    ev << ", [sMachine] >> Registering which port is the object server from.\n"
#endif

tmpPar = rcv_msg->par("add_name");
strcpy(tmpStr, tmpPar.stringValue ());
AddObjSvrName(tmpStr, rcv_gate->index());

#ifdef DEBUG_FLAG_L2
    ev << ", [sMachine] >> Received object server dns request " << tmpPar << \n";
#endif

// deleting the received message
delete rcv_msg;

break;

//OBJSVR request to execute on CPU

case OBJSVR_EXECUTE_ON_CPU:
#ifdef DEBUG_FLAG_L2
    ev << ", [sMachine] >> Received object server request to execute. \n";
#endif

100
if (addNewProcess(rcv_msg))  
{  
  //delete the rcv_msg  
  delete rcv_msg;  
  #ifdef DEBUG_FLAG_L2  
  ev << "[sMachine] >> request saved on instControlBlock! \n";  
  #endif  
  }  
else  
{  
  //we might want to inform the sender.  
  ev << "[sMachine] >> instControlBlock full, message discarded! \n";  
}  
break;  

//OBJSVR send the response back to the role, or remote object server  
#endif DEBUG_FLAG_L2  
case OBJSVR_RESPONSE:  
  #ifdef DEBUG_FLAG_L2  
  ev << "[sMachine] >> Received object server response. \n";  
  #endif  
  //We have to check whether the address is the to an object server running on  
  //on the current machine.  
  tmpPar = rcv_msg->par("add_request");  
  strcpy(tmpStr, tmpPar.stringValue());  
  objsvr_id = FindObjSvrName(tmpStr);  
  ev << "[sMachine] >> Received object server response to requestor " << tmpPar << " "  
<< objsvr_id << "\n";  
  if (objsvr_id == -1)  
  {  
    //The object server is not running on the current machine  
    msgSz = rcv_msg->par("msgSz");  
    m_res = new cMessage("OBJSVR_RESPONSE", OBJSVR_RESPONSE);  
    m_res->addPar("add_request") = rcv_msg->par("add_request");  
    m_res->addPar("request_id") = rcv_msg->par("request_id");  
    m_res->addPar("msgSz") = msgSz;  
    m_res->setLength(msgSz);  
    send( m_res, "m2sw_out");  
    delete(rcv_msg);  
  }  
else  
{  
  //The object server is running on the current machine.  
  //Process the request at the machine
requestId = rcv_msg->par("request_id");
index = findProcessByRequestId(requestId);
if (index == -1)
    break;
else
{
    recRemoteResponse(index);
}
ev << "[sMachine] >> Receiving a remote response id " << requestId <<'\n';
delete(rcv_msg);
break;
}
else if (isSwitchGateId(rcv_gate->id()))
{
    switch( type )
    {
    case OJJSVR_RESPONSE:
#ifdef DEBUG_FLAG_L2
    ev << "[sMachine] >> Received object server response. \n";
#endif
    //receiving a response from the remote object server.
    requestId = rcv_msg->par("request_id");
    index = findProcessByRequestId(requestId);
    if (index == -1)
        break;
    else
    {
        recRemoteResponse(index);
    }
    ev << "[sMachine] >> Receiving a remote response id " << requestId <<'\n';
    delete(rcv_msg);
    break;
    }
}
else if (isSwitchGateId(rcv_gate->id()))
{
    switch( type )
    {
    case INVOKE_OS2OS_CALL:
        cpObjSvrName = rcv_msg->par("add_os"); //Object
        //Method name
        cpMethodName = rcv_msg->par("add_method");
        //Method name
        cpRequestorName = rcv_msg->par("add_request");
        requestId = rcv_msg->par("request_id"); //Method name
        delete rcv_msg;

        //Invoking object server call
        obj_msg = new cMessage("INVOKE_OS2OS_CALL", INVOKE_OS2OS_CALL);
        obj_msg->addPar("add_os") = cpObjSvrName;
        obj_msg->addPar("add_method") = cpMethodName;
        obj_msg->addPar("add_request") = cpRequestorName;
        obj_msg->addPar("request_id") = requestId;

        strcpy(tmpStr, cpObjSvrName.stringValue());

0
objsvr_id = FindObjSvrName(tmpStr);

// Need to check with port to send out to.
send( obj_msg, "m2os_out", objsvr_id);

#ifdef DEBUG_FLAG_L2
    ev << "[Server] >> Invoking object server call on objsvr " << objsvr_id
    <<"\n";
#endif
    break;

case INVOKE_OBJECT_SVR_CALL:
    // Receive message from the switch
    cpObjSvrName = rcv_msg->par("add_os");  // Object server name
    cpMethodName = rcv_msg->par("add_method");  // Method name
    cpRequestorName = rcv_msg->par("add_request");  // Method name
    requestId = rcv_msg->par("request_id");
    delete rcv_msg;

    // Invoking object server call
    obj_msg = new cMessage("INVOKE_OBJECT_SVR_CALL", INVOKE_OBJECT_SVR_CALL);
    obj_msg->addPar("add_os") = cpObjSvrName;
    obj_msg->addPar("add_method") = cpMethodName;
    obj_msg->addPar("add_request") = cpRequestorName;
    obj_msg->addPar("request_id") = requestId;
    strcpy(tmpStr, cpObjSvrName.stringValue());
    objsvr_id = FindObjSvrName(tmpStr);

    // Need to check with port to send out to.
    send( obj_msg, "m2os_out", objsvr_id);
    ev << "[sMachine] >> " << machineName << " receive from " << cpRequestorName << "\n";

    #ifdef DEBUG_FLAG_L2
        ev << "[Server] >> Invoking object server call on objsvr " << objsvr_id
        <<"\n";
    #endif
    break;
    }
    }
    if (!newMsg)
        execNextProcess();
    }

//********************/
//* Protected Method
//********************/
// Method to initialize the request list. The request list is used to
// store the list of id that will be used to identify the remote request that was
// forward other object server. This is needed because the machine may process a
//few message from the same object server and the completion of the message may
//not be in order.
bool sMachine::initRemoteRequestList()
{
    for (int i=0; i<MAX_REMOTE_REQUEST_PER_MACHINE; i++)
    {
        remoteRequestIdFlag[i] = -1;
    }

    return true;
}

//Method to add a new remote request in the request list.
//The method will find a remoteRequestIdFlag that is -1 and
//use that position to store the corresponding instIndex.
int sMachine::addNewRemoteRequest(int instIndex)
{
    bool found = false;
    int tmpInt = 0;

    //Search the requestIdFlag to find an id that was not in use.
    while ((tmpInt<MAX_REMOTE_REQUEST_PER_MACHINE) && (!found))
    {
        if (remoteRequestIdFlag[tmpInt] == -1)
        {
            remoteRequestIdFlag[tmpInt] = instIndex;
            found = true;
            return tmpInt;
        }
        tmpInt++;
    }

    return -1;
}

//Method to return the instruction index of the remote request response.
int sMachine::findRemoteRequest(int index)
{
    return remoteRequestIdFlag[index];
}

//Method to add a new process in queue
bool sMachine::initProcess()
{
    for (int i=0; i<MAX_PROCESS_PER_MACHINE; i++)
    {
        instControlBlock[i].used = false;
    }

    curProcessExecuted = 0;
    curProcessAddPosition = 0;

    return true;
}
Method to add a new process in queue
int sMachine::findProcessByRequestId(int id)
{
    bool found = false;
    int tmpInt = 0;

    while ((tmpInt < MAX_PROCESS_PER_MACHINE) && (!found))
    {
        if ((instControlBlock[tmpInt].used == true) &&
            (instControlBlock[tmpInt].remoteRequestId == id))
        {
            return tmpInt;
        }
        else
        {
            tmpInt++;
        }
    }
    return -1;
}

Method to add a new process in queue
bool sMachine::recRemoteResponse(int index)
{
    int remoteCallIndex = 0;
    int request_id, objsvr_id;
    cMessage *m_resp;

    //reset the remote request id to indicate that the response was received
    remoteRequestIdFlag[instControlBlock[index].remoteRequestId] = -1;

    //There is some remote instruction. Format message to send to the remote objsvr.
    if (instControlBlock[index].curRemoteInstruction !=
        instControlBlock[index].maxRemoteInstruction)
    {
        //if there is still some other remote call, process the next one.
        instControlBlock[index].remoteRequestId = addNewRemoteRequest(index);

        remoteCallIndex = instControlBlock[index].curRemoteInstruction;
        instControlBlock[index].curRemoteInstruction++;

        cMessage *call_msg = new cMessage( "INVOKE_OS2OS_CALL",
            INVOKE_OS2OS_CALL );
        call_msg->addPar("add_os") =
            instControlBlock[index].interactList.interactCall[remoteCallIndex].cObjsvr;
        call_msg->addPar("add_method") =
            instControlBlock[index].interactList.interactCall[remoteCallIndex].cMethod;
        call_msg->addPar("add_request") = instControlBlock[index].cObjSvrName;
        call_msg->addPar("request_id") = instControlBlock[index].remoteRequestId;

        int port =
            FindObjSvrName(instControlBlock[index].interactList.interactCall[remoteCallIndex].cObjsvr);
        if ( port == -1)
        {
            //...
//Object server is not running on the current machine, forward to the
switch
    send( call_msg, "m2sw_out");
} else {
    //Object server is not running on the current machine, forward to the
correct object server
    int result = send( call_msg, "m2os_out", port);
}

instControlBlock[index].wait = true;
} else {
    instControlBlock[index].used = false;
    instControlBlock[index].wait = false;

    request_id = instControlBlock[index].requestId;
    objsvr_id = FindObjSvrName(instControlBlock[index].cObjSvrName);

    // Forward the information to the server
    m_resp = new cMessage( "MACHINE_COMPLETE_EXEC_WITH_REMOTE",
MACHINE_COMPLETE_EXEC_WITH_REMOTE );
    m_resp->addPar("request_id") = request_id;

    //set the remote request id to -1.
    remoteRequestIdFlag[instControlBlock[index].remoteRequestId] = -1;

#endif
    ev << "[sMachine] >> Sending MACHINE_COMPLETE_EXEC response " <<
    instControlBlock[index].cObjSvrName << 
; //endif
    send( m_resp, "m2os_out", objsvr_id);
}

return true;

//Method to add a new process in queue
bool sMachine::addNewProcess(cMessage *msg)
{
    bool found = false;
    int index;
    int tmpInt = 0;
    cPar cpObjSvrName, cpMethodName, cpOrigRequestName;

    //search for the first empty slot.
    while ((tmpInt<MAX_PROCESS_PER_MACHINE) && (!found))
    {
        if (instControlBlock[tmpInt].used == false)
        {
            //found new used position
            found = true;
            index = tmpInt;
        }
    }
```
else {
    tmpInt++; 
}
}

if (!found) 
    return false;

int cpu = msg->par("cpu");

//both cpu and cpuPow is defined with unit 1000Hz
double process_time = (double)(cpu)*((double)1/cpuPow);
ev << "[sMachine] (init) >> index " << index << 
    "\n;  
ev << "[sMachine] (init) >> process time " << process_time << 
    "\n;  
ev << "[sMachine] (init) >> cpu " << cpu << "\n;  
ev << "[sMachine] (init) >> cpuPow " << cpuPow << "\n;  

//initializing the value of the instruction control block
  cpObjSvrName = msg->par("add_os"); //Object server name
  strcpy(instControlBlock[index].cObjSvrName, cpObjSvrName.stringValue ());
  cpMethodName = msg->par("add_method"); //Method name
  strcpy(instControlBlock[index].cMethodName, cpMethodName.stringValue ());
  cpOrigRequestName = msg->par("orig_request"); //Method name
  strcpy(instControlBlock[index].cOrigRequest, cpOrigRequestName.stringValue ());

  //wait is used to determine whether the process is waiting for response from
  //a remote object server.
  instControlBlock[index].wait = false;

  //setting the current indexed instControlBlock as used.
  instControlBlock[index].used = true;

  //Setting the request id
  instControlBlock[index].requestId = msg->par("request_id");

  //Setting the process time.
  instControlBlock[index].processTime = process_time;

  //Setting the information for remote instruction.
  sInteract interact;
  found = false;
  found = CInteractList::GetInstance()->getInteract(instControlBlock[index].cObjSvrName, 
    instControlBlock[index].cMethodName, interact);

  //if there is an interaction list found for this method implies that remote objsvr call is
  //required.
  if (found)
    {
    instControlBlock[index].curRemoteInstruction = -1;
    instControlBlock[index].maxRemoteInstruction = interact.iNumInteract;
    instControlBlock[index].remoteRequestId = -1;
    instControlBlock[index].interactList = interact;
    }
else 
    {
```
// Method to return the next process for processing
bool sMachine::execNextProcess()
{
    bool found = false;
    bool completeExec = false;
    bool execRemote = false;
    int index;

    int objsvr_id, request_id, port;
    int remoteCallIndex;
    cPar cpObjSvrName, cpMethodName;
    cMessage *m_resp;
    int tmpInt = 0;

    // An attempt to make the behaviour of the execution process more random.
    // old code
    double ranExecTime = uniform(-1*ranExec, ranExec);

    double ranExecTime = uniform(0.0, execTime);
    double ranDelayTime = uniform(0.0, processSwapTime);

    // To find the next instruction to execute
    tmpInt = curProcessExecuted;
    while (tmpInt<MAX_PROCESS_PER_MACHINE) && (!found))
    {
        if ((instControlBlock[tmpInt].used == true) &&
            (instControlBlock[tmpInt].wait == false))
        {
            // found new used position
            found = true;
            index = tmpInt;
        }
        else
        {
            tmpInt++;
        }
    }

    if (!found)
    {
        tmpInt = 0;
        while ((tmpInt<=(curProcessExecuted-1)) && (!found))
        {
            if ((instControlBlock[tmpInt].used == true) &&
                (instControlBlock[tmpInt].wait == false))
            {
                // found new used position
                found = true;
            }
        }
    }

    return true;
}
index = tmpInt;
}
else
{
    tmpInt++;
}
}
//No instruction found. CPU remains idle
if (!found)
    return false;
if (instControlBlock[index].processTime == 0.0)
{
    //if processTime is 0.0, this could be the scenario when the machine is
    //waiting for the response from remote system. If wait is true, then it
    //is time to execute the next remote call.
    if (instControlBlock[index].wait == true)
    {
        //instruction still waiting for the remote object server to response.
        //ie do nothing for now.
        return true;
    }
    else
    {
        //instruction not waiting for the response
        //execute next remote instruction call if any.
        if (instControlBlock[index].curRemoteInstruction ==
            instControlBlock[index].maxRemoteInstruction)
        {
            //all instruction completed.
            completeExec = true;
            execRemote = false;
        }
    }
}
else
{
    if (isTimeSlice)
    {
        // old
        if (instControlBlock[index].processTime < (execTime+ranExecTime))
            if (instControlBlock[index].processTime < (ranExecTime))
            {
                wait(instControlBlock[index].processTime+ranDelayTime);
                instControlBlock[index].processTime = 0.0;
                execRemote = true;
            }
        else
        {
            // old
            instControlBlock[index].processTime =
                instControlBlock[index].processTime-(execTime+ranExecTime);
            // old
            wait(execTime+ranExecTime+ranDelayTime);
            instControlBlock[index].processTime =
                instControlBlock[index].processTime-(ranExecTime);
            wait(ranExecTime+ranDelayTime);
        }
    }
}
else {
    ev << "[sMachine] >> index " << index << "n;";
    ev << "[sMachine] >> random delay time " << ranDelayTime << "n;";
    ev << "[sMachine] >> process time " << instControlBlock[index].processTime << "n;";
    ev << "[sMachine] >> wait time " << instControlBlock[index].processTime+ranDelayTime << "n;";
    
    wait(instControlBlock[index].processTime+ranDelayTime);
    instControlBlock[index].processTime = 0.0;
    execRemote = true;
}

if (execRemote)
{
    //The current implementation will requires the machine to execute all the
    //needed for the local object server before requesting for remote call.

    //After setting the processTime to 0.0, we need to check whether there
    //is any remote obj server call.
    if (instControlBlock[index].maxRemoteInstruction == 0)
    {
        //There is no remote instruction
        completeExec = true;
    }
    else
    {
        completeExec = false;

        //There is some remote instruction. Format message to send to the
        remote objsvr.
        instControlBlock[index].remoteRequestId = addNewRemoteRequest(index);
        
        if (instControlBlock[index].curRemoteInstruction == -1)
        {
            instControlBlock[index].curRemoteInstruction = 0;
            remoteCallIndex = instControlBlock[index].curRemoteInstruction;
        }
        else
        {
            remoteCallIndex = instControlBlock[index].curRemoteInstruction;
        }

        instControlBlock[index].curRemoteInstruction++;

        cMessage *call_msg = new cMessage( "INVOKE_OS2OS_CALL", INVOKE_OS2OS_CALL );
        call_msg->addPar("add_os") = instControlBlock[index].interactList.interactCall[remoteCallIndex].cObjsvr;
        call_msg->addPar("add_method") = instControlBlock[index].interactList.interactCall[remoteCallIndex].cMethod;
call_msg->addPar("add_request") = instControlBlock[index].cObjSvrName;
call_msg->addPar("request_id") = instControlBlock[index].remoteRequestId;
// call_msg->addPar("request_id") = remoteCallIndex;

port = FindObjSvrName(instControlBlock[index].interactList.interactCall[remoteCallIndex].cObjsvr);
if ( port == -1)
{
    //Object server is not running on the current machine, forward to
    the switch
    send( call_msg, "m2sw_out");
}
else
{
    //Object server is not running on the current machine, forward to
    the correct object server
    int result = send( call_msg, "m2os_out", port);
}

instControlBlock[index].wait = true;
#if defined DEBUG_FLAG_L2
    ev << "[sMachine] >> Sending remote objsvr call to " << remoteCallIndex << " " << instControlBlock[index].interactList.interactCall[remoteCallIndex].cObjsvr << " " << instControlBlock[index].interactList.interactCall[remoteCallIndex].cMethod << '\n';
#endif

if (completeExec)
{
    instControlBlock[index].used = false;
    request_id = instControlBlock[index].requestId;
    objsvr_id = FindObjSvrName(instControlBlock[index].cObjSvrName);

    // Forward the information to the serevr
    m_resp = new cMessage( "MACHINE_COMPLETE_EXEC",
                          MACHINE_COMPLETE_EXEC );
    m_resp->addPar("request_id") = request_id;
#endif
    ev << "[sMachine] >> Sending response" << '\n';
#endif
    send( m_resp, "m2os_out", objsvr_id);
}

curProcessExecuted = index++;
return true;

//Method to delete the process
bool sMachine::deleteProcess(int id)
{
    return true;
}
// Method to check whether the gate id is a switch gate id.
bool sMachine::isSwitchGateId(int id)
{
    for (int i=0; i<m2swin_sz; i++)
    {
        if (m2swin_id[i] == id)
            return true;
    }
    return false;
}

// To return the index of the gate.
void sMachine::InitObjSvrName()
{
    char tmpStr[MAX_NAME_SZ] = "0";
    for (int i=0; i<MAX_OBJSVR_PER_MACHINE; i++)
    {
        strcpy(objSvrName[i], tmpStr);
    }
}

// To save the object server name and the index of the gate,
// information will be used for the reply message
void sMachine::AddObjSvrName(char *objsvr, int index)
{
    strcpy(objSvrName[index], objsvr);
}

// To return the index of the gate.
int sMachine::FindObjSvrName(char *objsvr)
{
    char tmpstr[MAX_NAME_SZ];
    for (int i=0; i<MAX_OBJSVR_PER_MACHINE; i++)
    {
        strcpy(tmpstr, objSvrName[i]);
        if (strcmp(objsvr, tmpstr) == 0)
            return i;
    }
    return -1;
}

// Method to check whether the gate id is a server gate id.
bool sMachine::isObjSvrGateId(int id)
{
    for (int i=0; i<m2osin_sz; i++)
    {
        if (m2osin_id[i] == id)
            return true;
    }
    return false;
}
double sMachine::getWaitTime(double cpu, int osid)
{
    double totTime = cpu;
    double waitTime = 0.0;

    if (osid != prevObjSvrId)
    {
        totTime = totTime + processSwapTime;
        prevObjSvrId = osid;
    }

    waitTime = uniform(totTime-0.05, totTime+0.05);
    return waitTime;
}

double sMachine::getWaitTime()
{
    double waitTime = processCallTime;
    return waitTime;
}

bool sMachine::writeData(char *str)
{
    //The logging function can be turn off at compilation time.
    #ifdef DEBUG_DETAIL_LOG
        FILE *stream;
        if( (stream = fopen( "test.txt", "a" )) == NULL )
            printf( "The file 'test.txt' was not opened\n" );

        //fwrite( str, sizeof( char ), 100, stream );
        fprintf(stream, str);

        /* Close stream */
        if( fclose( stream ) )
            printf( "The file 'data' was not closed\n" );
    #endif

    return true;
}

bool sMachine::loadDef()
{
    char seps[] = "/n";
    char tmpStr[50];
    char buf[1024 + 1];
    FILE *file;

    if( (file = fopen( defFile, "r+t" )) == NULL )
    {
printf( "The file '%s' was not opened\n", defFile );

if (!file)
{
    ev << "[sMachine] >> The file 'data' was not opened\n";
    return false;
}

//some changes were made in the way we read from the file because
//there is some conflict between omnet++ 2.3 with ifstream.
//The using-declaration generate a error during compilation.
//There is a conflict with including "ifstream.h" and "strstrea.h"
//
//After copying from the buf string to tmpStr, we
//need to set the last character to NULL so that any new-line character
//will not affect the string read.
while (fgets(buf, 1024, file))
{
    if (strstr(buf, "[name]") != NULL)
    {
        strcpy(tmpStr, buf+strlen("[name]")+1);
        tmpStr[strlen(tmpStr)-1] = '\0';
        strcpy(machineName, tmpStr);
        #ifdef DEBUG_FLAG_L1
        ev << "[sMachine] >> Object server name " << machineName << '\n';
        #endif
    }
    else if (strstr (buf, "[cpu_pow]") != NULL)
    {
        strcpy(tmpStr, buf+len("[cpu_pow]")+1);
        tmpStr[strlen(tmpStr)-1] = '\0';
        cpuPow = atoi(tmpStr);
        #ifdef DEBUG_FLAG_L1
        ev << "[sMachine] >> CPU power " << cpuPow << '\n';
        #endif
    }
    else if (strstr (buf, "[exec_time]") != NULL)
    {
        strcpy(tmpStr, buf+len("[exec_time]")+1);
        tmpStr[strlen(tmpStr)-1] = '\0';
        execTime = strtotime(tmpStr, '0');
        #ifdef DEBUG_FLAG_L1
        ev << "[sMachine] >> Execution time " << execTime << '\n';
        #endif
    }
    else if (strstr (buf, "[ran_exec]") != NULL)
    {
        strcpy(tmpStr, buf+len("[ran_exec]")+1);
        tmpStr[strlen(tmpStr)-1] = '\0';
        ranExec = strtotime(tmpStr, '0');
        #ifdef DEBUG_FLAG_L1
        ev << "[sMachine] >> Random Execution time " << ranExec << '\n';
        #endif
    }
    else if (strstr (buf, "[ram_sz]") != NULL)
    {
        strcpy(tmpStr, buf+len("[ram_sz]")+1);
        tmpStr[strlen(tmpStr)-1] = '\0';
        #ifdef DEBUG_FLAG_L1
        ev << "[sMachine] >> RAM size " << ranExec << '\n';
        #endif
    }
}
```c
strcpy(tmpStr, buf+strlen("[ram_sz]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
ramSz = atoi(tmpStr);
#endif DEBUG_FLAG_L1
  ev << "[sMachine] >> Ram size " << ramSz << 'n';
#endif
else if (strstr (buf, "[ram_limit]") != NULL)
{
  strcpy(tmpStr, buf+strlen("[ram_limit]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
ramLimit = strtod(tmpStr, ""');
#endif DEBUG_FLAG_L1
  ev << "[sMachine] >> Ram limit " << ramSz << 'n';
#endif
else if (strstr (buf, "[dr_sw]") != NULL)
{
  strcpy(tmpStr, buf +strlen("[dr_sw]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
dataRate2Sw = atoi(tmpStr);
#endif DEBUG_FLAG_L1
  ev << "[sMachine] >> Data rate to the switch " << dataRate2Sw << 'n';
#endif
else if (strstr (buf, "[er_sw]") != NULL)
{
  strcpy(tmpStr, buf +strlen("[er_sw]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
errorRate2Sw = strtod(tmpStr, ""');
#endif DEBUG_FLAG_L1
  ev << "[sMachine] >> Error rate to the switch " << errorRate2Sw << 'n';
#endif
else if (strstr (buf, "[process_call]") != NULL)
{
  strcpy(tmpStr, buf+strlen("[process_call]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
processCallTime = strtod(tmpStr, ""');
#endif DEBUG_FLAG_L1
  ev << "[sMachine] >> Process call time " << processCallTime << 'n';
#endif
else if (strstr (buf, "[process_swap]") != NULL)
{
  strcpy(tmpStr, buf+strlen("[process_swap]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
processSwapTime = strtod(tmpStr, ""');
#endif DEBUG_FLAG_L1
  ev << "[sMachine] >> Process swap time " << processSwapTime << 'n';
#endif
else if (strstr (buf, "[disk_swap]") != NULL)
{
  strcpy(tmpStr, buf+strlen("[disk_swap]")+1);
tmpStr[strlen(tmpStr)-1] = '"';
```


diskSwapTime = strtof(tmpStr, "\0");
#ifdef DEBUG_FLAG_L1
   ev << "[sMachine] >> Disk swap time " << diskSwapTime << "\n";
#endif
}
else if (strstr (buf, "[time_slice]") != NULL)
{
   strcpy(tmpStr, buf+strlen("[time_slice]")+1);
   tmpStr[strlen(tmpStr)-1] = '0';
   if (strcmp("no", tmpStr)==0)
      isTimeSlice = false;
   else
      isTimeSlice = true;
#endif
   ev << "[sMachine] >> isTimeSlice " << isTimeSlice << "\n";
}
}
if( fclose( file ) )
   printf( "The file '%s' was not closed\n", defFile);

return true;

}
bool loadDef();

cGate *os2min_gt;
int os2min_id[MAX_GATE_SZ];
int os2min_sz;

char defFile[MAX_NAME_SZ];

int ramUtil;    //ram usage of the object server without the consideration of the object
int numObj;
int ramTotal;
int machineId;

char objSvrName[MAX_NAME_SZ];
char objName[MAX_NAME_SZ];
CObj objSvc;

cMessage *requestList[MAX_PROCESS_PER_MACHINE];
bool requestIdFlag[MAX_PROCESS_PER_MACHINE];

};

Define_Module(sObjSvr);

//initialize method called when the object is first initialized.
void sObjSvr::initialize()
{
    os2min_gt = gate("os2m_in");

    //Initializing the value of the os2m_in gate id
    os2min_sz = os2min_gt->size();
    if (os2min_sz > MAX_GATE_SZ)
    {
        ev << "[! sObjSvr] >> Maximum number of gate defined (os2min_sz) " << os2min_sz << "\n";
        os2min_sz = MAX_GATE_SZ;
    }

    for (int i=0; i<os2min_sz; i++)
    {
        cGate *tmp = gate("os2m_in", i);
        os2min_id[i] = tmp->id();
    }

    #ifdef DEBUG_FLAG_L1
    ev << "\n[sObjSvr] >> Server (i) " << i << " (gateid) " << tmp->id() << "\n";
    #endif

    #ifdef DEBUG_FLAG_L1
    ev << "************ [sObjSvr] ************ " << "\n";
    #endif

    //initialize the value
    ramUtil = 0;
    numObj = 1;
    ramTotal = 0;
// Loading the configuration file.
cPar& filename = par("cfg_file");
    strcpy(defFile, filename.stringValue ());
loadDef();  // Loading definition file

// Setting the object server name
setName(objSvrName);

// Getting the object server machine id
// machineld = (int)par("machine_id");

CObjList::GetInstance()->getObj(objName, objSvc);
ramTotal = ramUtil;  // + numObj*objSvc.ram_sz;

// To initialize the request list.
initRequestList();
}

// To handle the finish event
void sObjSvr::finish()
{
    delete(os2min_gt);
    clearRequestList();
}

// Method to handle activity event
void sObjSvr::activity()
{
    int type, requestId;
    int cpu, msgSz;
    bool found;

cGate *rcv_gate;
cPar cpMethodName, cpMachineName, cpRequestName, cpObjSvr;
cMessage *objsvr_resp, *m_resp;
char tmpStr[MAX_NAME_SZ];

double process_time = uniform(1.0,3.0);
writeln(process_time);

    // Registering the
    cMessage *dns_reg_msg = new cMessage("OBJSVR_REGISTER", OBJSVR_REGISTER);
    dns_reg_msg->addPar("add_name") = objSvrName;
    dns_reg_msg->addPar("add_obj") = objName;
    dns_reg_msg->addPar("ram_total") = ramTotal;

    send( dns_reg_msg, "os2m_out");
#endif DEBUG_FLAG_L2
    ev << "[ObjSvr] >> Registering object service with DNS" << 
#endif
for(;;)
{
    // Receive message from the server
cMessage *rcv_msg = receive();
    type = rcv_msg->kind();
    rcv_gate = rcv_msg->arrivalGate();

#ifdef DEBUG_FLAG_L2
    ev << "[ObjSvrApp] >> Received message (type) " << type << " from (gate_id) " << rcv_gate->id() <<'
';
#endif

//Message is from the machine port.
if (isMachineGateId(rcv_gate->id()))
{
    switch( type )
    {
        case INVOKE_OS2OS_CALL:
            //A call to invoke the object server.
            cpObjSvr = rcv_msg->par("add_os"); //Object server name
            cpMethodName = rcv_msg->par("add_method"); //Method name
            cpRequestName = rcv_msg->par("add_request");
            strcpy(tmpStr, cpRequestName.stringValue());
            strcpy(tmpStr, cpMethodName.stringValue());
            cpu = objSvc.getCPU(tmpStr);
            requestId = addNewRequest(rcv_msg);

            // Forward the information to the server
            objsvr_resp = new cMessage( "OBJSVR_EXECUTE_ON_CPU", OBJSVR_EXECUTE_ON_CPU);
            objsvr_resp->addPar("request_id") = requestId;
            objsvr_resp->addPar("add_os") = rcv_msg->par("add_os");
            objsvr_resp->addPar("add_method") = rcv_msg-
            >par("add_method");
            objsvr_resp->addPar("orig_request") = rcv_msg-
            >par("add_request");
            objsvr_resp->addPar("cpu") = cpu;
            send( objsvr_resp, "os2m_out");
            break;

        case INVOKE_OBJECT_SVR_CALL:
            //A call to invoke the object server.
            cpObjSvr = rcv_msg->par("add_os"); //Object server name
            cpMethodName = rcv_msg->par("add_method"); //Method name
            strcpy(tmpStr, cpMethodName.stringValue());
            cpu = objSvc.getCPU(tmpStr);

            requestId = addNewRequest(rcv_msg);

        #ifdef DEBUG_FLAG_L2
        ev << "[ObjSvrApp] >> receiving INVOKE_OBJECT_SVR_CALL (method) " << method_id <<'
';
        #endif
    }
}
// Forward the information to the server
objsvr_resp = new cMessage("OBJSVR_EXECUTE_ON_CPU",
OBJSVR_EXECUTE_ON_CPU);
    objsvr_resp->addPar("request_id") = requestId;
    objsvr_resp->addPar("add_os") = rcv_msg->par("add_os");
    objsvr_resp->addPar("add_method") = rcv_msg->par("add_method");
    objsvr_resp->addPar("orig_request") = rcv_msg->par("add_request");
    objsvr_resp->addPar("cpu") = cpu;
    send( objsvr_resp, "os2m_out");

#ifdef DEBUG_FLAG_L2
    ev << "[ObjSvr] >> Sending response (cpu) " << cpu << " (ram) " << ram << "\n";
#endif
break;

case MACHINE_COMPLETE_EXEC:
    // A return call from the machine to indicate that the execution has completed.
    // Retrieve the requestId to verify that the id is one that was sent to from the object server.
    //the object server.
    requestId = rcv_msg->par("request_id");
    found = completeRequest(requestId);
#endif DEBUG_FLAG_L2
    ev << "[ObjSvrApp] >> receiving MACHINE_COMPLETE_EXEC replied to " << requestList[requestId]->par("add_request") << "\n";
#endif DEBUG_FLAG_L2
    //If the request id is found.
    if (found == true)
    {
        m_resp = new cMessage("OBJSVR_RESPONSE",
OBJSVR_RESPONSE);
        cpRequestName = requestList[requestId]->par("add_request");
        m_resp->addPar("add_request") = requestList[requestId]->par("add_request");
        m_resp->addPar("request_id") = requestList[requestId]->par("request_id");
        cpObjSvr = requestList[requestId]->par("add_os");
        cpMethodName = requestList[requestId]->par("add_method");
        //Method name
        strcpy(tmpStr, cpMethodName.stringValue());
        msgSz = objSvc.getMsgSize(tmpStr);
        m_resp->addPar("msgSz") = msgSz;
        send( m_resp, "os2m_out");
ev << "[ObjSvrApp] >> sending OBJSVR_RESPONSE from " << objSvrName << " to (requestor) " << cpRequestName << '
';

//Deleting the message
delete requestList[requestId];
delete rcv_msg;
}
break;

case MACHINE_COMPLETE_EXEC_WITH_REMOTE:
//A return call from the machine to indicate that the execution has completed.
//Retrieve the requestId to verify that the id is one that was sent to from
//the object server.
requestId = rcv_msg->par("request_id");
found = completeRequest(requestId);

ev << "[ObjSvrApp] >> receiving MACHINE_COMPLETE_EXEC_WITH_REMOTE replied to " << requestList[requestId]->par("add_request") << '
';

for (int j=0; j<MAX_REQUEST_PER_OBJSVR; j++)
{
    if (requestIdFlag[j] == true)
    {
        cpRequestName = requestList[j]->par("add_request");
        strcpy(tmpStr, cpRequestName.stringValue());
        ev << "[ObjSvrApp] >> request id " << j << " (request) "
<< cpRequestName << "\n";
    }
}

//ifdef DEBUG_FLAG_L2
//endif

//If the request id is found.
if (found == true)
{
    m_resp = new cMessage("OBJSVR_RESPONSE",
>par("add_request");
strcpy(tmpStr, cpRequestName.stringValue());
    m_resp->addPar("add_request") = requestList[requestId]->par("add_request");
    m_resp->addPar("request_id") = requestList[requestId]->par("request_id");

    //Object server name
    cpObjSvr = requestList[requestId]->par("add_os");

    //Method name
    cpMethodName = requestList[requestId]->par("add_method");
    strcpy(tmpStr, cpMethodName.stringValue());
    msgSz = objSvc.getMsgSize(tmpStr);
    m_resp->addPar("msgSz") = msgSz;
}
send( m_resp, "os2m_out");
ev << "[ObjSvrApp] >> sending OBJSVR_RESPONSE from " << objSvrName << " to (requestor) " << cpRequestName <<"\n";

//Deleting the message
delete requestList[requestId];
delete rcv_msg;
} break;
}
}

/********************/
/* Protected Method */
/********************/
//Method to initialize the request list. The request list is used to
//store the list of id that will be used to identify the request that was
//forward to the machine. This is needed because the machine may process a
//few message from the same object server and the completion of the message may
//not be in order.
bool sObjSvr::initRequestList()
{
    for (int i=0; i<MAX_REQUEST_PER_OBJSVR; i++)
    {
        requestIdFlag[i] = false;
    }
    return true;
}

bool sObjSvr::clearRequestList()
{
    for (int i=0; i<MAX_REQUEST_PER_OBJSVR; i++)
    {
        if (requestIdFlag[i] == true)
        {
            delete requestList[i];
        }
    }
    return true;
}

//Method to add a new request in the request list. The message is stored
//in case future work require the message.
int sObjSvr::addNewRequest(cMessage *msg)
{
    bool found = false;
    int index;
    int tmpInt = 0;

    //Search the requestIdFlag to find an id that was not in use.
    while ((tmpInt<MAX_REQUEST_PER_OBJSVR) && (!found))
    {
        if (requestIdFlag[tmpInt] == false)
// If an available id is used.
if (found)
{
    requestIdFlag[tmpInt] = true;
    return tmpInt;
}
else
{
    tmpInt++;
}

// If an available id is used.
if (found)
{
    requestList[index] = msg;
    return index;
}
else
{
    // return -1 if nothing is found.
    return -1;
}

// Method to handle a completion of request.
bool sObjSvr::completeRequest(int id)
{
    if (id > MAX_REQUEST_PER_OBJSVR)
    {
        return false;
    }
    if (requestIdFlag[id] == true)
    {
        requestIdFlag[id] = false;
        return true;
    }
    return false;
}

// Method to check whether the gate id is a server gate id.
bool sObjSvr::isMachineGateId(int id)
{
    for (int i=0; i<os2min_sz; i++)
    {
        if (os2min_id[i] == id)
        {
            return true;
        }
    }
    return false;
}

// Method to load the configuration file
bool sObjSvr::loadDef()
{
    char seps[] = "/n";
    char tmpStr[50];
    char buf[1024 + 1];

    FILE *file;
    if ( (file = fopen( defFile, "r+t" )) == NULL )
    {
        printf( "The file '%s' was not opened\n", defFile );
        ev << "[sObjSvr] >> The file 'data' was not opened\n";
        return false;
    }

    #ifdef DEBUG_FLAG_L2
    ev << "[sObjSvr] >> The file 'data' was opened\n";
    #endif

    //some changes were made in the way we read from the file because
    //there is some conflict between omnet++ 2.3 with ifstream.
    //The using-declaration generate a error during compilation.
    //There is a conflict with including "ifstream.h" and "strstrea.h"
    //After copying from the buf string to tmpStr, we
    //need to set the last character to NULL so that any new-line character
    //will not affect the string read.
    while (fgets(buf, 1024, file))
    {
        if (strstr(buf, "[name"]") != NULL)
        {
            strcpy(tmpStr, buf + strlen("[name"]") + 1);
            tmpStr[strlen(tmpStr) - 1] = '\0';
            strcpy(objSvrName, tmpStr);
            #ifdef DEBUG_FLAG_L1
            ev << "[sObjSvr] >> Object server name " << objSvrName << 'n';
            #endif
        }
        else if (strstr(buf, "[ram_util"]") != NULL)
        {
            strcpy(tmpStr, buf + strlen("[ram_util"]") + 1);
            tmpStr[strlen(tmpStr) - 1] = '\0';
            ramUtil = atoi(tmpStr);
            #ifdef DEBUG_FLAG_L1
            ev << "[sObjSvr] >> Ram Util " << ramUtil << 'n';
            #endif
        }
        else if (strstr(buf, "[object"]") != NULL)
        {
            strcpy(tmpStr, buf + strlen("[object"]") + 1);
            tmpStr[strlen(tmpStr) - 1] = '\0';
            strcpy(objName, tmpStr);
            #ifdef DEBUG_FLAG_L1
            ev << "[sObjSvr] >> Object Supported " << objName << 'n';
            #endif
    
}
```c++.

if( strcmp(buf, "[num_obj]") != NULL )
{
    strcpy(tmpStr, buf+strlen("[num_obj]")+1);
    tmpStr[strlen(tmpStr)-1] = '0';
    numObj = atoi(tmpStr);
}
#endif
    ev << "[sObjSvr] >> Number of object " << numObj << '"\n';
#endif
}

if( fclose( file ) )
    printf( "The file '%s' was not closed\n", defFile );
return true;

K. SROLE.CPP

/********************************************************************************
// file: sRole.cc
/********************************************************************************
#ifndef __ROLE
#define __ROLE

#include <stdio.h>
#include <string.h>
#include "omnetpp.h"
#include "global.h"
#include "simevent.h"
#include "btnlist.h"
#include "statlog.h"

class sRole : public cSimpleModule
{
    Module_Class_Members(sRole,cSimpleModule,16384)
    virtual void finish();
    virtual void activity();
    virtual void initialize();

protected:
    bool isSwitchGateId(int id);
    bool loadDef();
    bool writeLog(char *str);

    double betweenCallTime;

    int r2swin_sz;
    int r2swin_id[MAX_GATE_SZ];
    cGate *r2swin_gt;
    cGate *r2swout_gt;

```
char defFile[MAX_NAME_SZ];
char logFile[MAX_NAME_SZ];
char runName[MAX_NAME_SZ];
char roleName[MAX_NAME_SZ];

int callPatternCount;
int minWait, maxWait, roleId, dataRate2Sw, roleType;
sCallPattern callPattern[MAX_BUTTON_PERROLE];

CSimEvent simEvent;
CBtnList btnList;

CStdDev stat;

};

Define_Module(sRole);

void sRole::initialize()
{
    //initialize the list gateid
    r2swin_gt = gate("r2sw_in");
    r2swout_gt = gate("r2sw_out");

    //initializing the list of switch to client gate id
    r2swin_sz = r2swin_gt->size();
    if (r2swin_sz > MAX_GATE_SZ)
    {
        ev << "[!! sRole] >> Maximum number of gate defined (r2swin_sz) " << r2swin_sz
            << "\n";
        r2swin_sz = MAX_GATE_SZ;
    }

    for (int i=0; i<r2swin_sz; i++)
    {
        cGate *tmp = gate("r2sw_in", i);
        r2swin_id[i] = tmp->id();
        #ifdef DEBUG_FLAG_L1
            ev << [sRole] >> sRole (i) " " << i " " (gateid) " " << tmp->id() " " << 'n';
        #endif
    }

    //Initializing all the parameters
    callPatternCount = 0;

    //Getting the configuration file name and loading the configuration file
    cPar& tmpPar = par("def_file");
    strcpy(defFile, tmpPar.stringValue ());
    loadDef();

    //Setting the role name
    setName(roleName);

    //Setting the run name
    tmpPar = parentModule()->par("run_name");
    strcpy(runName, tmpPar.stringValue ());
ev << "{sRole} >> Initializing " << runName << \"\n\";

//The index will return roleid
roleid = index();
sprintf(logFile, "%s_log%d.txt", runName, roleid);

callPatternCount = 0;  //Setting the number of button to 0
CBtnList::GetInstance();
CStatLog::GetInstance()->registerRoleType(roleType);

//Detail logging can be turn off at compilation time.
#ifdef DEBUGDETAIL_LOG
    FILE *stream;
    if( (stream = fopen( logFile, "w+" )) == NULL )
        printf( \"The file \"%s\" was not opened\n\", logFile );
    else
        fclose( stream );
#endif

//Setting for the date rate.
tmpPar.setLongValue (dataRate2Sw);
r2swin_gt->setDataRate(&tmpPar);
r2swout_gt->setDataRate(&tmpPar);
}

//To handle the finish event
void sRole::finish()
{
    char logStr[200];
    long num_samples = stat.samples();
    double smallest = stat.min();
    double largest = stat.max();
    double mean = stat.mean();
    double dev = stat.stddev();
    double var = stat.variance();

    //recording the sample size, minimum, maximum and average value
    CStatLog::GetInstance()->writeLog(runName, roleName, roleType, num_samples, smallest, largest, mean);

    /*
    char tmpStr[MAX_NAME_SZ];
    sprintf(tmpStr, "stat_\%s", logFile);
    FILE *stream;
    if( (stream = fopen( tmpStr, "w+" )) == NULL )
        printf( \"The file \‘data2\’ was not opened\n\" );
    else
        { stat.saveToFile(stream);
          fclose( stream );
        }
    */
stat.clearResult();

//Storing all information.
sprintf(logStr, "%s_role%d (Min) %.4f (Max) %.4f (Ave) %.4f", runName, roleid, smallest, largest, mean);
writeLog(logStr);
}

//To handle the activity event
void sRole::activity()
{

    int index;

    int current_server = 0;
cOutVector resp_v("Role RT");

    //Time waited to ensure that DNS service is all register before sending.
double process_time = 0.0;
double response_time = 0.0;
double cpuTime = 0.0;
double startTime = 0.0;
double sendTime = 0.0;
double rcvTime = 0.0;
double curSimTime = 0.0;
double rdelay = 0.0;
double roleTime = 0.0;

    char logStr[200];
cMessage *dummy_msg = new cMessage();

    //wait for all the module to stablize before sending the register message
    process_time = uniform(1.0,3.0);
    wait(process_time);

cMessage *dns_reg_msg = new cMessage("NAME_REGISTER", NAME_REGISTER );
dns_reg_msg->addPar("add_name") = roleName;
dns_reg_msg->addPar("add_machine") = roleName;
send( dns_reg_msg, "r2sw_out");
ev << "[*****] registering name " << roleName << "\n";

    for(;;)
    {
        ev << "[*****] looping name " << minWait << " " << maxWait << roleName << "\n";
        //Listen for incoming message
        process_time = uniform(minWait,maxWait);
        wait(process_time);

        //This will return the next event to execute.
        index = simEvent.getNextEvent();
        if (index == -1) //Do nothing
            continue;

        //To record the startTime of a button call
        startTime = simTime();
        ev << "[*****] name " << roleName << " " << index << "\n";

    }
/For all the call, send a request message to all the object server.
for (int j=0; j<callPattern[index].button.iNumCall; j++)
{
    cMessage *call_msg = new cMessage("INVOKE_OBJECT_SVR_CALL", INVOKE_OBJECT_SVR_CALL);
    call_msg->addPar("add_os") = callPattern[index].button.svcCall[j].cObjsvr;
    call_msg->addPar("add_method") = callPattern[index].button.svcCall[j].cMethod;
    call_msg->addPar("add_request") = roleName;
    call_msg->addPar("request_id") = -1;
    call_msg->addPar("roleTime") = simTime();

    int result = send( call_msg, "r2sw_out");
    ev << "[*****] name " << roleName << " send result " << result << " dest objsvr " << callPattern[index].button.svcCall[j].cObjsvr << " dest method " << callPattern[index].button.svcCall[j].cMethod << "\n";

    //The message receive will be in order of the message sent. So there is no
    //need to store a request id.
    cMessage *done = receiveOn("r2sw_in");
delete done;
}

//Completion of sending all the message and compute the time spent.
curSimTime = simTime();
response_time = curSimTime - startTime;
resp_v.record(response_time);
stat.collect(response_time);

sprintf(logStr, "Role%d Total Response \t%d\t%.4f\n", roleId, index, response_time);
writeLog(logStr);
"
}
return true;
}

//Method to check whether the gate id is a server gate id.
bool sRole::isSwitchGateId(int id)
{
    for (int i=0; i<r2swin_sz; i++)
    {
        if (r2swin_id[i] == id)
            return true;
    }
    return false;
}

//Method to load the configuration file
bool sRole::loadDef()
{
    char seps[] = "/\n";
    char *token;
    char tmpStr[50];
    char buf[1024 + 1];
    int pos;
    sCallPattern tmpPattern;
    FILE *file;
    if( (file = fopen( defFile, "r+t" )) == NULL )
    {
        printf( "The file '%s' was not opened\n", defFile );
        return false;
    }
    
    #ifdef DEBUG_FLAG_L2
    ev << "[sRole] >> The file 'data' was opened\n";
    #endif

    //some changes were made in the way we read from the file because
    //there is some conflict between omnet++ 2.3 with ifstream.
    //The using-declaration generate a error during compilation.
    //There is a conflict with including "ifstream.h" and "strtstrea.h"
    //After copying from the buf string to tmpStr, we
    //need to set the last character to NULL so that any new-line character
    //will not affect the string read.
    while (fgets(buf, 1024, file))
    {
        if (strstr (buf, "[call]" ) != NULL)
        {
            strcpy(tmpStr, buf + strlen("[call]" )+1);
            tmpStr[strlen(tmpStr)-1] = "\0";
            //...
pos = 0;
token = strtok(tmpStr, seps);
while( token != NULL )
{
    switch (pos)
    {
        case 0: //Getting the name of the button
            ifdef DEBUG_FLAG_L1
                ev << "[sRole] >> Button Name " << token << \
"n;"
            endif
            strcpy(tmpPattern.cBtnName, token);
pos++;
break;
    case 1: //Getting the probability of call
            ifdef DEBUG_FLAG_L1
                ev << "[sRole] >> iProbCall " << token << \
"n;"
            endif
            tmpPattern.iProbCall = atoi(token);
break;
    }
    token = strtok(NULL, seps);
}
if (CBtnList::GetInstance()->
    >getButton(tmpPattern.cBtnName,tmpPattern.button))
{
    callPattern[callPatternCount] = tmpPattern;
simEvent.addEvent(callPatternCount, tmpPattern.iProbCall);
callPatternCount = callPatternCount + 1;
}
else if (strstr(buf, "[type]") != NULL)
{
    strcpy(tmpStr, buf+strlen("[type]")+1);
tmpStr[strlen(tmpStr)-1] = \0;
roleType = atoi(tmpStr);
ifdef DEBUG_FLAG_L1
    ev << "[sRole] >> Minimum wait " << minWait << \
"n;"
endif
}
else if (strstr(buf, "[min_wait]") != NULL)
{
    strcpy(tmpStr, buf+strlen("[min_wait]")+1);
tmpStr[strlen(tmpStr)-1] = \0;
minWait = atoi(tmpStr);
ifdef DEBUG_FLAG_L1
    ev << "[sRole] >> Minimum wait " << minWait << \
"n;"
endif
}
else if (strstr(buf, "[max_wait]") != NULL)
{
    strcpy(tmpStr, buf+strlen("[max_wait]")+1);
tmpStr[strlen(tmpStr)-1] = \0;
maxWait = atoi(tmpStr);
ifdef DEBUG_FLAG_L1
    ev << "[sRole] >> Maximum wait " << maxWait << \
"n;"
endif
}
#endif
}  
else if (strstr (buf, "[dr_sw]") != NULL)
{
    strcpy(tmpStr, buf + strlen("[dr_sw]")+1);
    tmpStr[strlen(tmpStr)-1] = '0';
    dataRate2Sw = atoi(tmpStr);
#endif DEBUG_FLAG_L1
ev << "[sRole] >> Data rate to switch " << dataRate2Sw << 'n';
#endif

} else if (strstr (buf, "[btw_call]") != NULL)
{
    strcpy(tmpStr, buf + strlen("[btw_call]")+1);
    tmpStr[strlen(tmpStr)-1] = '0';
    betweenCallTime = atoi(tmpStr);
#endif DEBUG_FLAG_L1
ev << "[sRole] >> Time between call " << betweenCallTime << 'n';
#endif

} else if (strstr (buf, "[name]") != NULL)
{
    strcpy(tmpStr, buf + strlen("[name]")+1);
    tmpStr[strlen(tmpStr)-1] = '0';
    strcpy(roleName, tmpStr);
#endif DEBUG_FLAG_L1
ev << "[sRole] >> Role name " << roleName << 'n';
#endif

}

if( fclose( file ) )
    printf( "The file '%s' was not closed\n", defFile);

return false;
#endif

L. SSWITCH.CPP

/////////////////////////////////////////////////////////////////
// file: switch.cpp
/////////////////////////////////////////////////////////////////

#include <string.h>
#include "omnetpp.h"
#include "global.h"
#include "dnssvc.h"

class sSwitch : public cSimpleModule
{
    Module_Class_Members(sSwitch,cSimpleModule,16384)
    virtual void finish();
}
virtual void activity();
virtual void initialize();

protected:
    bool isMachineGateId(int id);
    bool isRoleGateId(int id);

    int sw2min_sz;
    int sw2min_id[MAX_GATE_SZ];
    cGate *sw2min_gt;

    int sw2rin_id[MAX_GATE_SZ];
    int sw2rin_sz;
    cGate *sw2rin_gt;

    cGate *sw2mout_gt;

    CDnsSvc dnsObjSvc;
    CDnsSvc dnsRoleSvc;
};

Define_Module( sSwitch );

//initialize method called when the object is first initialized.
void sSwitch::initialize()
{
    //initialize the list of gateid
    sw2min_gt = gate("sw2m_in");
    sw2rin_gt = gate("sw2r_in");
    sw2mout_gt = gate("sw2m_out");

    int sw2mout_sz;

    //initializing the list of switch to server gate id
    sw2min_sz = sw2min_gt->size();
    if (sw2min_sz > MAX_GATE_SZ)
    {
        ev << "[!! sSwitch] >> Maximum number of gate defined (sw2min_sz) " << sw2min_sz << 'n';
        sw2min_sz = MAX_GATE_SZ;
    }

    for (int i=0; i<sw2min_sz; i++)
    {
        cGate *tmp = gate("sw2m_in", i);
        sw2min_id[i] = tmp->id();
        #ifdef DEBUG_FLAG_L1
        ev << "[sSwitch] >> Server (i) " << i << " (gateid) " << tmp->id() << 'n';
        #endif
    }

    //initializing the list of switch to client gate id
    sw2rin_sz = sw2rin_gt->size();
    if (sw2rin_sz > MAX_GATE_SZ)
    {

ev << "[!! sSwitch] >> Maximum number of gate defined (sz_sw2r) " << sw2rin_sz << '
';
    sw2rin_sz = MAX_GATE_SZ;
}

for (i=0; i<sw2rin_sz; i++)
{
    cGate *tmp = gate("sw2r_in", i);
    sw2rin_id[i] = tmp->id();
#ifdef DEBUG_FLAG_L1
    ev << "[sSwitch] >> Client (i) " << i << " (gateid) " << tmp->id() << '
';
#endif
}

//initializing the list of switch to client out gate id
sw2mout_sz = sw2mout_gt->size();
if (sw2mout_sz > MAX_GATE_SZ)
{
    ev << "[!! sSwitch] >> Maximum number of gate defined (sw2mout_sz) " << sw2mout_sz << '
';
    sw2mout_sz = MAX_GATE_SZ;
}

for (i=0; i<sw2mout_sz; i++)
{
    cGate *tmp = gate("sw2m_out", i);
#ifdef DEBUG_FLAG_L1
    ev << "[sSwitch] >> Server out (i) " << i << " (gateid) " << tmp->id() << '
';
#endif
}

#ifdef DEBUG_FLAG_L1
    ev << "************ [sSwitch] ************" << '
';
#endif
}

//To handle the finish event
void sSwitch::finish()
{
    delete(sw2min_gt);
    delete(sw2rin_gt);
    delete(sw2mout_gt);
}

//Method to handle activity event call
void sSwitch::activity()
{
    int type;
    cGate *rcv_gate;

cPar cpObjSvrName, cpMethodName, cpMachineName, cpRequestorName, cpName;
cMessage *sw_msg;
    int ingate, outgate, index, requestId;
    int inport, outport, msgSz;
    double roleTime, tmpDouble;
for(;;)
{
    //Listen for incoming message
    cMessage *rcv_msg = receive();

    type = rcv_msg->kind();          //Determine the type of the message.
    rcv_gate = rcv_msg->arrivalGate(); //Determine the gate type

    //If the message is from a machine gate.
    if (isMachineGateId(rcv_gate->id()))
    {
        //if the message is from a server_in gate id.
        #ifdef DEBUG_FLAG_L2
        ev << "[sSwitch] >> Received server message (type) " << type << " from sw2s_in" << 'n';
        #endif
        switch( type )
        {
        case NAME_REGISTER:
            //DNS Register message to be sent over to the DNS
            cpMachineName = rcv_msg->par("add_machine");
            //Server machine name
            cpName = rcv_msg->par("add_name"); //Object server machine name
            delete rcv_msg;
            //Extract the string varlu of the machine and object server name
            strcpy(tmpStr1, cpMachineName.stringValue());
            strcpy(tmpStr2, cpName.stringValue());
            //Extract the gate id
            ingate = rcv_gate->id();
            index = rcv_gate->index();
            outgate = gate("sw2m_out", index)->index();
            //Mapping the port to the machine and object server name.
            dnsObjSvrSvc.MapSvcToPort(inigate, outgate, tmpStr1, tmpStr2);
            inport = dnsObjSvrSvc.FindInGateByName(tmpStr2);
            outport = dnsObjSvrSvc.FindOutGateByName(tmpStr2);
            break;
        //Routing the server response
        case OBJSVR_RESPONSE:
            //The requestor address
            cpRequestorName = rcv_msg->par("add_request");
            requestId = rcv_msg->par("request_id");
            msgSz = rcv_msg->par("msgSz"); //Msg Size
            delete rcv_msg;
            ev << "[sSwitch] SERVER_RESPONSE to " << cpRequestorName << 'n';
        } //End of switch
    } //End if (isMachineGateId)
} //End for(;;)
```c
#ifdef DEBUG_FLAG_L2
    ev << "[sSwitch] SERVER_RESPONSE (client_id) " << client_id << "\n";
#endif

strcpy(tmpStr1, cpRequestorName.stringValue());

sw_msg = new cMessage("OBJSVR_RESPONSE");

sw_msg->addPar("add_request") = cpRequestorName;
sw_msg->addPar("request_id") = requestId;
sw_msg->setLength(msgSz);

//check whether the destination is in the role id
outport = dnsRoleSvc.FindOutGateByName(tmpStr1);
if (outport != -1)
{
   //destination belongs to the role
   send(sw_msg, "sw2r_out", outport);
}

outport = dnsObjSvrSvc.FindOutGateByName(tmpStr1);
if (outport != -1)
{
   //destination belongs to the role
   send(sw_msg, "sw2m_out", outport);
}
break;

case INVOKE_OS2OS_CALL:
    cpObjSvrName = rcv_msg->par("add_os");
    cpMethodName = rcv_msg->par("add_method");

#ifdef DEBUG_FLAG_L2
    ev << "[sSwitch] INVOKE_OS2OS_CALL (obj) " << cpObjSvrName << " (method) " <<
        cpMethodName << "\n";
#endif

    //Extract the string value of the machine and object
    server name
    strcpy(tmpStr1, cpObjSvrName.stringValue());
    strcpy(tmpStr2, cpMethodName.stringValue);

    outport = dnsObjSvrSvc.FindOutGateByName(tmpStr1);

    //Send the port to the relevant out port.
    if (outport != -1)
        send(rcv_msg, "sw2m_out", outport);
    break;

//If the message is from a role gate.
else if (isRoleGateId(rcv_gate->id()))
{
#ifdef DEBUG_FLAG_L2
    ev << "[sSwitch] >> Received client message (type) " << type << " from sw2s_in" << "\n";
#endif
```
switch (type)
{
    // Invoke object server call message received.
    case INVOKE_OBJECT_SVR_CALL:
        cpObjSvrName = rcv_msg->par("add_os");
        cpMethodName = rcv_msg->par("add_method");
        cpRequestorName = rcv_msg->par("add_request");
        roleTime = rcv_msg->par("roleTime"); // Msg Size
        tmpDouble = simTime() - roleTime;
        ev << "[sSwitch] receive from " << cpRequestorName << "\n";

        #ifdef DEBUG_FLAG_L2
            ev << "[sSwitch] (obj) " << objSvrName << " (method) " << methodName << "\n";
        #endif

        // Extract the string varlu of the machine and object server name
        strcpy(tmpStr1, cpObjSvrName.stringValue());
        strcpy(tmpStr2, cpMethodName.stringValue());

        // Send the port to the relevant out port.
        if (outport != -1)
            send (rcv_msg, "sw2m_out", outport);
        break;
}

    case NAME_REGISTER:
        // DNS Register message to be sent over to the DNS server
        cpMachineName = rcv_msg->par("add_machine");
        // Server machine name
        cpName = rcv_msg->par("add_name"); // Object server name
        delete rcv_msg;

        // Extract the string varlu of the machine and object server name
        strcpy(tmpStr1, cpMachineName.stringValue());
        strcpy(tmpStr2, cpName.stringValue());

        // Extract the gate id
        ingate = rcv_gate->id();
        index = rcv_gate->index();
        outgate = gate("sw2r_out", index)->index();

        // Mapping the port to the machine and object server name.
        dnsRoleSvc.MapSvcToPort(inGate, outgate, tmpStr1, tmpStr2);
        inport = dnsRoleSvc.FindInGateByName(tmpStr2);
        outport = dnsRoleSvc.FindOutGateByName(tmpStr2);
        break;
// Method to check whether the gate id is a server gate id.
bool isServerGateId(int id) {
    for (int i=0; i<sw2min_sz; i++)
        if (sw2min_id[i] == id)
            return true;
    return false;
}

// Method to check whether the gate id is a client gate id.
bool isRoleGateId(int id) {
    for (int i=0; i<sw2rin_sz; i++)
        if (sw2rin_id[i] == id)
            return true;
    return false;
}

M. STATLOG.CPP

// StatLog.cpp: implementation of the CStatLog class.
//
_WAKE_KEYWORDS_*/
#include <stdio.h>
#include <string.h>
#include "omnetpp.h"
#include "StatLog.h"

static CStatLog *m_StatLog = NULL;
CStatLog *CStatLog::GetInstance()
{
    if (m_StatLog==NULL)
    {
        m_StatLog = new CStatLog();
        m_StatLog->initialize();
    }
CStatLog::CStatLog()
{
}
CStatLog::~CStatLog()
{
}

// To initialize the StatLog.
bool CStatLog::initialize()
{
    for (int i = 0; i < MAX_ROLETYPE; i++)
    {
        meanRoleType[i] = 0.0;
        countRoleType[i] = 0;
        maxRoleType[i] = 0;
    }
    return true;
}

// To get an object using the object name
bool CStatLog::registerRoleType(int type)
{
    maxRoleType[type] = maxRoleType[type] + 1;
    return true;
}

// To get an object using the object name
bool CStatLog::writeLog(char *name, char *role, int type, double sz, double smallest, double largest, double mean)
{
    char str[500];
    FILE *stream;

    if( (stream = fopen( "stat.log", "a+" )) == NULL )
        printf( "The file 'stat.log' was not opened\n" );

    sprintf(str, "[Sz\Min\Max\Ave]\t%%.4f \t%%.4f \t%%.4f \t%%.4f\n", name, role, sz, smallest, largest, mean);
    fprintf(stream, str);
    /* Close stream */
    if( fclose( stream ) )
        printf( "The file 'data' was not closed\n" );
    countRoleType[type] = countRoleType[type] + 1;
meanRoleType[type] = meanRoleType[type] + mean;

if (countRoleType[type] == maxRoleType[type])
{
    if( (stream = fopen( "computed.log", "a+" )) == NULL )
        printf( "The file 'computed.log' was not opened\n" );

    double computedMean = (double)(meanRoleType[type]/countRoleType[type]);

    sprintf(str, "[Type\nSz\nTotal Mean\nMean] %s %s %d %.4f %.4f\n", name, role, countRoleType[type], meanRoleType[type], computedMean);
    fprintf(stream, str);

    /* Close stream */
    if( fclose( stream ) )
        printf( "The file 'computed.log' was not closed\n" );

    meanRoleType[type] = 0.0;
    countRoleType[type] = 0;
    maxRoleType[type] = 0;
}
return true;

bool CStatLog::writeLog(char *str)
{
    FILE *stream;

    if( (stream = fopen( "stat.log", "a+" )) == NULL )
        printf( "The file 'stat.log' was not opened\n" );

    fprintf(stream, str);

    /* Close stream */
    if( fclose( stream ) )
        printf( "The file 'stat.log' was not closed\n" );

    return true;
}

bool CStatLog::writeLogWithRamError(char *name, double sz, double cur)
{
    FILE *stream;
    char str[500];

    if( (stream = fopen( "computed.log", "a+" )) == NULL )
        printf( "The file 'computed.log' was not opened\n" );

    sprintf(str, "[Type\nSz\nTotal Mean\nMean] %s -- RAM ERROR (Sz)%2f (Cur)%2f \n", name, sz, cur);
    fprintf(stream, str);

    /* Close stream */
    if( fclose( stream ) )
        printf( "The file 'computed.log' was not closed\n" );

    return true;
}
return true;
}

N. DISOBJSVRSM.NED NETWORK DESCRIPTION FILE

//-------------------------------------------------------------
// file: DisObjSvrSim.ned
//-------------------------------------------------------------

/***************************************************************
// Hardware Modelled
/***************************************************************
// Role --
// A client computer which periodically connects to the
// server for data exchange. In future, role based client should
// be implemented for the thesis.
//
simple sRole
    parameters:
        def_file : string;
    gates:
        //switch connection port
        out: r2sw_out;
        in: r2sw_in;
endsimple

// Switch --
// A very simple module which models the network between
// the servers and clients
//
simple sSwitch
    gates:
        out: sw2r_out[];
        in: sw2r_in[];
        out: sw2m_out[];
        in: sw2m_in[];
        in: sw2d_in;
endsimple

// Server --
// Models a server which accepts connections from the
// client computers. It serves multiple connections of
// object servers.
//
simple sMachine
    parameters:
        num_objsvr : numeric,
        objsvridx : numeric,
        def_file : string;
    gates:
/switch connection port to the s
out: m2sw_out;
in: m2sw_in;

//object server connection port to the object server app
out: m2os_out[];
in: m2os_in[];
endsimple

// ObjSvrApp --
//
// Models a client application
//
simple sObjSvr
parameters:
    cfg_file : string;
gates:
    //object server application connection port to the server
    out: os2m_out;
in: os2m_in;
endsimple

// DisObjSvrSim --
//
//
module DisObjSvrSim
//***************************************************************
// Parameters
//***************************************************************
parameters:
    run_name : string,      //The name of the run
    num_machine : numeric,  //Number of server
    num_role : numeric,     //Total number of roles in the system
    num_objsvr : numeric,   //Total number of object server in the system
    max_objsvr : numeric;   //Max number of object server per server

//***************************************************************
// Submodules
//***************************************************************
submodules:
    smachine: sMachine[num_machine];
gatesizes:
    m2os_out[max_objsvr],
m2os_in[max_objsvr];
display: "p=95,250,row,50;i=proc1;b=36,32";

role: sRole[num_role];
display: "p=95,450,row,50;i=proc1;b=36,32";

switch: sSwitch;
gatesizes:
    sw2m_out[num_machine],
    sw2m_in[num_machine],
sw2r_out[num_role],
sw2r_in[num_role],
sw2d_in;
display: "p=155,350,row,50;i=cloud;b=32,32";

objsvr: sObjSvr[num_objsvr];
display: "p=51,150,row,50;i=proc2;b=32,32";

//***************************************************************
// Connections
//***************************************************************
connections nocheck:
  //Connections between server and switch
  for i=0..num_role-1 do
    role[i].r2sw_out --> delay 10ms --> switch.sw2r_in[i];
    role[i].r2sw_in <-- delay 10ms <-- switch.sw2r_out[i];
  endfor;

  for i=0..num_machine-1 do
    smachine[i].m2sw_out --> delay 10ms --> switch.sw2m_in[i];
    smachine[i].m2sw_in <-- delay 10ms <-- switch.sw2m_out[i];
  endfor;

  for i=0..num_machine-1, j=smachine[i].objsvridx+0..smachine[i].objsvridx+smachine[i].num_objsvr-1 do
    smachine[i].m2os_out[i-j=smachine[i].objsvridx] --> delay 10ms --> objsvr[j].os2m_in;
    smachine[i].m2os_in[j-smachine[i].objsvridx] <-- delay 10ms <-- objsvr[j].os2m_out;
  endfor;
endmodule

// theDisObjSvrSim --
//
// Instantiates the DisObjSvrSim network
//
//network theDisObjSvrSim : DisObjSvrSim
network model : DisObjSvrSim
parameters:
  // ... (parameter assignments)
endnetwork
APPENDIX B - SOURCE CODE OF PATTERN GENERATOR

A. PATTERNGENERATOR.CPP

// PatternGenerator.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

const int MAX_MACHINEINFO = 20;
const int MAX_ROLE = 20;
const int MAX_OBJSVR = 20;
const int MAX_PATTERN = 1000;
const int MAX_STRING_SZ = 100;

char newBaseFile[MAX_STRING_SZ];
char baseFile[MAX_STRING_SZ];
char machineName[MAX_MACHINEINFO][MAX_STRING_SZ];
char objsvrName[MAX_OBJSVR][MAX_STRING_SZ];

int numRoleType = 0;
char roleName[MAX_ROLE][MAX_STRING_SZ];
char roleBaseFile[MAX_ROLE][MAX_STRING_SZ];
int roleCount[MAX_ROLE];

int patternCount = 0;
int fileCount = 0;

struct sMACHINFO
{
    char name[MAX_STRING_SZ];
    int numOfObj;
    int objIndex[MAX_OBJSVR];
    int indexOfObj;
} machinfo;

sMACHINFO machineInfo[MAX_MACHINEINFO];
int machineInfoCount;
int objSvrIndex;

//To initialize the machine info array for future storage.
//The maximum number of machine info is set to 20.
void initMachineInfo()
{
    for (int i=0; i<MAX_MACHINEINFO; i++)
    {
        strcpy(machineInfo[i].name, "");
        machineInfo[i].numOfObj = 0;
        machineInfo[i].indexOfObj = 0;
    }
}
for (int j=0; j<MAX_OBJSVR; j++)
    machineInfo[i].objIndex[j] = -1;
}
machineInfoCount = 0;
}

//To get the number of bits needed to represent the machine.
//For example, if there is 5 machines, we will need 101 i.e. 3 bits
//to represent 5 machines.
int getNumOfBits(int machine)
{
    unsigned int tmpNum1 = 0;
    unsigned int tmpNum2 = 0;
    int counter = 0;

    tmpNum1 = (unsigned int)machine;

    //To shift the bit pattern until the number
    //becomes zero. The counter will represent the number of bits
    //needed.
    while (tmpNum1 != 0)
    {
        tmpNum1 = tmpNum1 >> 1;
        counter = counter + 1;
    }

    return counter;
}

//To get the filter needed for the bits pattern.
//For example, if for 5 machines, base=3 and the filter pattern
//should be 111.
unsigned int getFilter(int base)
{
    unsigned int tmpNum1 = 0;

    //For the number of bit, compute the filter.
    for (int i=0; i<base; i++)
    {
        tmpNum1 = tmpNum1 + pow(2, i);
    }
    return tmpNum1;
}

//To get the bits string to represent the machine and objsvr
unsigned int getBitsString(int base, int machine, int objsvr)
{
    unsigned int tmpNum1 = 0;
    unsigned int tmpNum2 = 0;
    int counter = 0;

    tmpNum1 = (unsigned int)machine;

    while (counter != objsvr)
{  
    //shift the bit to the left by base length.
    //For example if base=2, tmpNum2=0, the result will be 11
    tmpNum2 = tmpNum2 << base;

    //Forcing an OR string will map the value to the appropriate value
    //For example, if the machine is 5(101), ie base=3. Having the initial
    //value as 111 will introduce additional processing time that is not necessary
    //Forcing it to 101 will help to reduce unnecessary processing.
    tmpNum2 = tmpNum2 | tmpNum1;
    counter = counter + 1;
}

return tmpNum2;
}

//To get the bits string to represent the machine and objsvr
unsigned int getTerminateString(int base, int machine, int objsvr)
{
    unsigned int tmpNum1 = 0;
    unsigned int tmpNum2 = 0;
    int counter = 0;

    tmpNum1 = (unsigned int)machine;

    while (counter != objsvr)
    {
        //shift the bit to the left by base length.
        //For example if base=2, tmpNum2=0, the result will be 11
        tmpNum2 = tmpNum2 << base;
        counter = counter + 1;
    }

return tmpNum2;
}

//To check whether the terminate string is reached.
bool checkFinalString(unsigned int terminateString, unsigned int string, unsigned int filter)
{
    if (terminateString > string)
        return true;
    return false;
}

//To check whether the string is valid. This method will break up the
//string pattern into blocks of bit in the same size as the base.
//For example, 11111111 is broken in 4 blocks of 2 bits each.
bool checkString(unsigned int string, int base, int machine, int objsvr, unsigned int filter)
{
    unsigned int tmpNum1 = 0;
    unsigned int tmpNum2 = 0;
    unsigned int tmpNum3 = 0;
    int counter = 0;

    tmpNum1 = string;

    while (counter < base)
tmpNum3 = (unsigned int)machine;

// check the length first.

while (counter != objsvr)
{
    // Perform an AND operation.
    // For example 11101010 AND 11 will return 10.
    tmpNum2 = tmpNum1 & filter;

    // if the extracted block is 0 or greater than the number of machine
    // then it is an infeasible pattern. Ignore the pattern.
    if ((tmpNum2 == 0) || (tmpNum2 > machine))
    {
        return false;
    }

    // shift right by base bit.
    tmpNum1 = tmpNum1 >> base;
    counter = counter + 1;
}

return true;

// To print the role file content.
// Example: simR1(2).ini
// [Run 1]       #0
// model.run_name = "pat1"     #1
// model.num_machine = 3     #2
// model.num_objsvr = 3      #3
// model.max_objsvr = 3      #4
// model.smachine[2].def_file = "SIX.def"    #5
// model.smachine[2].num_objsvr = 3    #6
// model.smachine[2].objsvridx = 0    #7
// model.objsvr[0].cfg_file = "A.def"     #8
// model.objsvr[1].cfg_file = "B.def"     #9
// model.objsvr[2].cfg_file = "C.def"     #10
bool printPatternString(unsigned int string, int count, int base, int machine, int objsvr, unsigned int filter)
{
    FILE *in_stream;
    FILE *out_stream;

    unsigned int tmpNum1 = 0;
    unsigned int tmpNum2 = 0;
    unsigned int tmpNum3 = 0;
    int counter = 0;

    tmpNum1 = string;
    tmpNum3 = (unsigned int)machine;
    char tmpStr2[200] = "";
    char tmpStr1[20] = "";
    char tmpStr3[200] = "";

    bool found = false;
bool breakFile = false;
int maxNumObjSvr = 0;

//The purpose of these code is to break the initiation file
//into block of 1000 call pattern each. These will allow
//smaller file for easy management. To enable the features
//set the flag breakFile to true.
if ((patternCount > MAX_PATTERN) && (breakFile))
{
    fileCount = fileCount + 1;
    sprintf(newBaseFile, "%s_%d.cfg", baseFile, fileCount);
}
else
{
    sprintf(newBaseFile, "%s.cfg", baseFile);
}

//open the pattern file for writing.
if( (out_stream = fopen( newBaseFile, "a+" )) == NULL )
    return 0;

//initialize the machine info block
initMachineInfo();

//Initialize the machine information with the machine name.
for (int j=0; j<machine; j++)
{
    strcpy(machineInfo[j].name, &machineName[j][0]);
    machineInfo[j].numOfObj = 0;
    machineInfo[j].indexOfObj = 0;
}

//Write the information for each object server.
while (counter != objsvr)
{
    found = false;
    tmpNum2 = tmpNum1 & filter;

    //getting the name of the machine
    sprintf(tmpStr1, "%s", &machineName[tmpNum2-1][0]);
    for (int i=0; i<machine; i++)
    {
        //compare the machine name is it is equal than
        //increase the numOfObj by one.
        if (strcmp(machineInfo[i].name, tmpStr1) == 0)
        {
            //found matching string that is already in the machineInfo
            //store the index which is the counter.
            machineInfo[i].objIndex[machineInfo[i].numOfObj] = counter;
            machineInfo[i].numOfObj = machineInfo[i].numOfObj + 1;

            //to determine the max number of object server allocated to any
            //machine.
            if (maxNumObjSvr < machineInfo[i].numOfObj)
                maxNumObjSvr = machineInfo[i].numOfObj;
    }
found = true;
}
}
sprintf(tmpStr3, "[%s] ", tmpStr1);
strcat(tmpStr2, tmpStr3);
tmpNum1 = tmpNum1 >> base;
counter = counter + 1;
}

objSvrIndex = 0;

//write comment line
sprintf(tmpStr3, "## pat%d %s ##
", count+1, tmpStr2);
fprintf(out_stream, tmpStr3);
printf(tmpStr3);

//write line #0
sprintf(tmpStr3, "[Run %d]\n", count+1);
fprintf(out_stream, tmpStr3);

//write line #1
sprintf(tmpStr2, "model.run_name = \"pat%d\"
", count+1);
fprintf(out_stream, tmpStr2);

//write line #2
sprintf(tmpStr2, "model.num_machine = %d\n", machine);
fprintf(out_stream, tmpStr2);

//write line #3
sprintf(tmpStr2, "model.num_objsvr = %d\n", objsvr);
fprintf(out_stream, tmpStr2);

//write line #4
sprintf(tmpStr2, "model.max_objsvr = %d\n", maxNumObjSvr);
fprintf(out_stream, tmpStr2);

//write line #5-10
for(int k=0; k<machine; k++)
{
    //write line #5
    sprintf(tmpStr2, "model.smachine[%d].def_file = \"%s.def\"
", k, machineInfo[k].name);
    fprintf(out_stream, tmpStr2);

    //write line #6
    sprintf(tmpStr2, "model.smachine[%d].num_objsvr = %d\n", k, machineInfo[k].numOfObj);
    fprintf(out_stream, tmpStr2);

    //write line #7
    if (machineInfo[k].numOfObj != 0)
    {
        sprintf(tmpStr2, "model.smachine[%d].objsvridx = %d\n", k, objSvrIndex);
        fprintf(out_stream, tmpStr2);
    }
}
else
{
    sprintf(tmpStr2, "model.smachine[%d].objsvidx = %d\n", k, 0);
    fprintf(out_stream, tmpStr2);
}

//write line #8-10
for (int m=0; m<(machineInfo[k].numOfObj; m++)
{
    sprintf(tmpStr2, "model.objsvr[%d].cfg_file = "s.def\n", m+objSvrIndex, objsvrName[machineInfo[k].objIndex[m]]);
    fprintf(out_stream, tmpStr2);
}

objSvrIndex = objSvrIndex + machineInfo[k].numOfObj;
fprintf(out_stream, "\n");
fclose(out_stream);
patternCount = patternCount + 1;
return true;
}

bool printRoleFile()
{
    FILE *in_stream;
    FILE *out_stream, *out_stream2;

    int roleIndex = 0;
    int totalRole = 0;

    char tmpStr[200] = "";
    char curFile[200] = "";
    char roleUName[200] = "";

    //To open the baseFile.
if( (out_stream = fopen( baseFile, "a+" )) == NULL )
    return 0;

//printing the [Parameters] header, this is the format
//needed by the omnet.ini file.
sprintf(tmpStr, "[Parameters]\n");
fprintf(out_stream, tmpStr);

//Generate the file for each role type
while (roleIndex != numRoleType)
{
    //opening the role base file for duplicating
    if( (in_stream = fopen( roleBaseFile[roleIndex], "r+" )) == NULL )
        return 0;

    //generate a role file for each
    for (int i=0; i<roleCount[roleIndex]; i++)
    {
        //creating the role configuration file.
        sprintf(tmpStr, "model.role[%d].def_file = "%s_%d.def\n", totalRole, 
                roleName[roleIndex], i);  
        fprintf(out_stream, tmpStr);
        totalRole = totalRole + 1;

        //creating a configuration file for each role defined.
        fseek(in_stream, 0, SEEK_SET);
        fgets(tmpStr, 100, in_stream);

        //printing a unique role name for each of the role defined.
        //the role name is of the format <RoleName>_<Index of this role>
        strcpy(roleUName, roleName[roleIndex]);
        _strupr(roleUName);

        //creating a filename roleA_1.def
        sprintf(curFile, "%s_%d.def", roleName[roleIndex], i);
        if( (out_stream2 = fopen( curFile, "w+" )) == NULL )
            return 0;
        printf("Writing %s...
", curFile);

        //print line #1
        sprintf(tmpStr, "[name] %s_%d\n", roleUName, i);
        fprintf(out_stream2, tmpStr);

        //copying from the base file to the new role file name.
        while (fgets(tmpStr, 100, in_stream) != NULL)
        {
            //copying line #2-9 from the role base file into the new file
            fprintf(out_stream2, tmpStr);
        }

        //close the new file
        fclose(out_stream2);
    }
    //close the file only the number of files are generated.
fclose(in_stream);
roleIndex = roleIndex + 1;
}

//printing the line #2 for role.ini
sprintf(tmpStr, "model.num_role = %d\n", totalRole);
fprintf(out_stream, tmpStr);

//close the role file.
fclose(out_stream);

return true;
}

void generatePatternFile()
{
    char tmpStr[100];
    char curFile[100];
    char roleUName[100];

    int numMachine = 0;
    int numObjSvr = 0;
    int numBaseBit = 0;
    int numValidString = 0;

    unsigned int bitsString = 0;
    unsigned int oriString = 0;
    unsigned int filter = 0;
    unsigned int terminateString = 0;

    bool valid = false;
    bool final = false;

    //Enter the new pattern file name
    printf("Please enter the base file > ");
    gets( baseFile );
    sprintf(newBaseFile, "%s_%d.ini", baseFile, fileCount);

    //Enter the number of machine
    printf("Please enter the number of machine > ");
    gets( tmpStr );
    numMachine = atoi(tmpStr);

    //For each of the machine, prompt for the machine name.
    for (int i=0; i<numMachine; i++)
    {
        printf("Please enter the name of machine > ");
        gets( tmpStr );
        strcpy(&machineName[i][0], tmpStr);
    }

    //Enter the number of object server
    printf("Please enter the number of object server > ");
    gets( tmpStr );
numObjSvr = atoi(tmpStr);

// For each of the object server, prompt for the object server name
for (int j=0; j<numObjSvr; j++)
{
    printf("Please enter the name of object server > ");
    gets( tmpStr );
    strcpy(&objsvrName[j][0], tmpStr);
}

// For example the scenario of 3 machines[A,B,C] and 4 objsvrs.
// Each machine can be represented as two bits i.e. 11. A bit string is created such as
// 11111111
// So 11111111 will means that the 4 objsvrs are run in the 3rd machine.
// For each iteration, the bitstring is subtract by 1. And the deployment pattern is computed again.
// 1st pass : 11111111 > {C}{C}{C}{C}
// 2nd pass : 11111110 > {C}{C}{C}{B}
// 3rd pass : 11111101 > {C}{C}{C}{A}
// 4th pass : 11111100 > No feasible, ignore.
// .......
// nth pass : 00111111 > Terminating string, since any decrease will have no effect.

// To find the number of bits needed to represent the machine
numBaseBit = getNumOfBits(numMachine);

// To get the filter pattern with the bits pattern.
filter = getFilter(numBaseBit);

// To find the bits string to represent all the machine.
bitsString = getBitsString(numBaseBit, numMachine, numObjSvr);

// To find the determining the terminating string. For example, if there is 4 objsvr, the terminate String is 00111111 = 63.
terminateString = getTerminateString(numBaseBit, numMachine, numObjSvr-1);

// Copy the string to oristring.
oriString = bitsString;

while ((bitsString != 0) && (!final))
{
    // To check whether the final string has occurred.
    final = checkFinalString(terminateString, bitsString, filter);

    // If the value has not reach the terminating string.
    if (!final)
    {
        // Check whether the string is a valid string, in the previous example
        // 0 is a non feasible solution and it will be ignore.
        valid = checkString(bitsString, numBaseBit, numMachine, numObjSvr, filter);

        if (valid)
        {

        }

}
//try printing out the value
printPatternString(bitsString, numValidString, numBaseBit,
numMachine, numObjSvr, filter);
numValidString = numValidString + 1;

//subtract the value by one.
bitsString = bitsString - 1;

printf("Number of Valid String >> %d\n", numValidString);
}

//a. To prompt the user for input that is needed to generate the role file
//b. Generate the role file
void generateRoleFile()
{
    char tmpStr[100];

    //Enter the role configuration file name
    printf("Please enter the new role configuration filename > ");
    gets( baseFile );

    //Enter the number of role type.  E.g. if there is
    //RoleA, RoleB, RoleC, the number of role type is 3
    printf("Please enter the number of role type > ");
    gets( tmpStr );
    numRoleType = atoi(tmpStr);

    //For each role type.
    for (int k=0; k<numRoleType; k++)
    {
        //Enter the name of the role.
        printf("Please enter the name of role %d > ", k);
        gets( tmpStr );
        strcpy(&roleName[k][0], tmpStr);

        //Enter the base filename.  The base filename will be
        //used to generate the role configuration file.
        //For example, if there is 10 RoleA, then there will
        //be 10 files generated from roleA_0.def to roleA_9.def
        printf("Please enter the base file name > ");
        gets( tmpStr );
        strcpy(&roleBaseFile[k][0], tmpStr);

        //Enter the number of this role type.
        printf("Please enter the number of role > ");
        gets( tmpStr );
        roleCount[k] = atoi(tmpStr);
    }

    //To print the role file.
    printRoleFile();
}
int main(int argc, char* argv[]) {
    bool finish = false;
    char tmpStr[10];
    int choice = 9;

    // Prompt for a choice and if choice is 9, exit the program.
    while (!finish)
    {
        printf("Please select file to generate > \n");
        printf("1. Generate call pattern file. \n");
        printf("2. Generate role configuration file. \n");
        printf("3. Reserved. \n");
        printf("9. Exit \n");
        gets(tmpStr);

        choice = atoi(tmpStr);
        switch (choice)
        {
            // To generate the pattern file.
            case 1: generatePatternFile(); break;

            // To generate the role file.
            case 2: generateRoleFile(); break;

            // To exit the program.
            case 9: finish = true; break;
        }
    }

    return 0;
}
APPENDIX C – VERIFICATION EXPERIMENT 1

A. CONFIGURATION FILES

Object server A – A.def
[name] A
[ram_util] 44000
[object] A
[num_obj] 1

Object server B – B.def
[name] B
[ram_util] 60000
[object] B
[num_obj] 1

Object server C – C.def
[name] C
[ram_util] 66000
[object] C
[num_obj] 1

Role0 Configuration File – role0.def
[name] ROLE0
[type] 0
[min_wait] 15
[max_wait] 16
[call] C1.B2/1
[call] C2.B1/1
[call] C2.B6/1
[dr_sw] 200000000
[btw_call] 0.05

Role1 Configuration File – role1.def
[name] ROLE1
[type] 1
[min_wait] 6
[max_wait] 7
[call] C1.B1/10
[dr_sw] 200000000
[btw_call] 0.05

Role2 Configuration File – role2.def
[name] ROLE2
<table>
<thead>
<tr>
<th>Machine</th>
<th>CPU Power</th>
<th>RAM Size</th>
<th>RAM Limit</th>
<th>DR SW</th>
<th>ER SW</th>
<th>Process Call</th>
<th>Process Swap</th>
<th>Disk Swap</th>
<th>Execution Time</th>
<th>Random Executive</th>
<th>Time Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIX</td>
<td>600000</td>
<td>64000</td>
<td>0.0</td>
<td>200000000</td>
<td>0.000000001</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>5.0</td>
<td>0.05</td>
<td>yes</td>
</tr>
<tr>
<td>BR733</td>
<td>733000</td>
<td>128000</td>
<td>0.0</td>
<td>200000000</td>
<td>0.000000001</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>5.0</td>
<td>0.05</td>
<td>yes</td>
</tr>
<tr>
<td>GIGA</td>
<td>1000000</td>
<td>128000</td>
<td>0.0</td>
<td>200000000</td>
<td>0.000000001</td>
<td>0.005</td>
<td>0.005</td>
<td>0.005</td>
<td>5.0</td>
<td>0.05</td>
<td>yes</td>
</tr>
</tbody>
</table>
Button Configuration File – *button.def*

```
[NEW_DEF]
C1.B1
A/1
[END_DEF]
[NEW_DEF]
C1.B2
A/2
B/1
[END_DEF]
[NEW_DEF]
C2.B1
C/1
C/2
[END_DEF]
[NEW_DEF]
C2.B2
C/3
[END_DEF]
[NEW_DEF]
C2.B3
C/2
[END_DEF]
[NEW_DEF]
C2.B4
C/3
[END_DEF]
[NEW_DEF]
C2.B5
A/1
B/2
```

Interaction Configuration File – *interact.def*

```
[NEW_INTERACT] B/2
[CALL] C/1
[END_INTERACT]
```

Object Configuration File – *obj.def*

```
[NEW_OBJ_DEF]
A/10000
1/579600/112000
2/2620300/18400
3/1181750/44800
4/2026400/176000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
B/10000
1/1766550/4000000
2/3700850/2720000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
C/10000
```
Role Configuration File for Simulation1

[Parameters]
model.role[0].def_file = "role0.def"
model.num_role = 1

Role Configuration File for Simulation2

[Parameters]
model.role[0].def_file = "role1.def"
model.num_role = 1

Role Configuration File for Simulation3

[Parameters]
model.role[0].def_file = "role2.def"
model.num_role = 1

Role Configuration File for Simulation4

[Parameters]
model.role[0].def_file = "ROLE1_0.def"
model.role[1].def_file = "ROLE1_1.def"
model.role[2].def_file = "ROLE1_2.def"
model.role[3].def_file = "ROLE1_3.def"
model.num_role = 4

Role Configuration File for Simulation5

[Parameters]
model.role[0].def_file = "ROLE2_0.def"
model.role[1].def_file = "ROLE2_1.def"
model.role[2].def_file = "ROLE2_2.def"
model.num_role = 3

Role Configuration File for Simulation6

[Parameters]
theDisObjSvrSim.role[0].def_file = "role0_0.def"
theDisObjSvrSim.role[1].def_file = "role0_1.def"
theDisObjSvrSim.role[2].def_file = "role0_2.def"
theDisObjSvrSim.role[3].def_file = "role0_3.def"
theDisObjSvrSim.role[4].def_file = "role0_4.def"
theDisObjSvrSim.role[5].def_file = "role0_5.def"
theDisObjSvrSim.role[6].def_file = "role0_6.def"
theDisObjSvrSim.role[7].def_file = "role0_7.def"
theDisObjSvrSim.role[8].def_file = "role0_8.def"
theDisObjSvrSim.role[9].def_file = "role0_9.def"
theDisObjSvrSim.role[10].def_file = "role0_10.def"
theDisObjSvrSim.role[11].def_file = "role0_11.def"
theDisObjSvrSim.role[12].def_file = "role0_12.def"
theDisObjSvrSim.role[13].def_file = "role0_13.def"
theDisObjSvrSim.role[14].def_file = "role0_14.def"
theDisObjSvrSim.role[15].def_file = "role0_15.def"
theDisObjSvrSim.role[16].def_file = "role0_16.def"
theDisObjSvrSim.role[17].def_file = "role0_17.def"
theDisObjSvrSim.role[18].def_file = "role0_18.def"
theDisObjSvrSim.role[19].def_file = "role0_19.def"
theDisObjSvrSim.role[20].def_file = "role0_20.def"
theDisObjSvrSim.role[21].def_file = "role0_21.def"
theDisObjSvrSim.role[22].def_file = "role0_22.def"
theDisObjSvrSim.role[23].def_file = "role0_23.def"
theDisObjSvrSim.role[24].def_file = "role0_24.def"
theDisObjSvrSim.role[25].def_file = "role0_25.def"
theDisObjSvrSim.role[26].def_file = "role0_26.def"
theDisObjSvrSim.role[27].def_file = "role0_27.def"
model.num_role = 28

Pattern Configuration File for

## pat1 [GIGA] [GIGA] [GIGA] ##
[Run 1]
model.run_name = "pat1"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 3
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 3
model.smachine[2].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "B.def"
model.objsvr[2].cfg_file = "C.def"

## pat2 [SIX] [GIGA] [GIGA] ##
[Run 2]
model.run_name = "pat2"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[1].cfg_file = "B.def"
model.objsvr[2].cfg_file = "C.def"

## pat3 [BR733] [GIGA] [GIGA] ##
[Run 3]
model.run_name = "pat3"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[1].cfg_file = "B.def"
model.objsvr[2].cfg_file = "C.def"

## pat4 [GIGA] [SIX] [GIGA] ##
[Run 4]
model.run_name = "pat4"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.objsvr[2].cfg_file = "C.def"

## pat5 [SIX] [SIX] [GIGA] ##
[Run 5]
model.run_name = "pat5"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 2
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "B.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "C.def"

## pat6 [BR733] [SIX] [GIGA] ##
[Run 6]
model.run_name = "pat6"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 1
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "B.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[2].cfg_file = "C.def"

## pat7 [GIGA] [BR733] [GIGA] ##
[Run 7]
model.run_name = "pat7"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.objsvr[2].cfg_file = "C.def"
## pat8 [SIX] [BR733] [GIGA] ##

[Run 8]
model.run_name = "pat8"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 1
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "C.def"

## pat9 [BR733] [BR733] [GIGA] ##

[Run 9]
model.run_name = "pat9"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 2
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "B.def"
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "C.def"

## pat10 [GIGA] [GIGA] [SIX] ##

[Run 10]
model.run_name = "pat10"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "C.def"
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.objsvr[2].cfg_file = "B.def"

## pat11 [SIX] [GIGA] [SIX] ##
[Run 11]
model.run_name = "pat11"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 2
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "C.def"
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "B.def"

## pat12 [BR733] [GIGA] [SIX] ##
[Run 12]
model.run_name = "pat12"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 1
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "C.def"
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "B.def"

## pat13 [GIGA] [SIX] [SIX] ##
[Run 13]
model.run_name = "pat13"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 2
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"
model.objsvr[1].cfg_file = "C.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "A.def"

## pat14 [SIX] [SIX] [SIX] ##
[Run 14]
model.run_name = "pat14"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 3
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 0
model.smachine[0].objsvridx = 0

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 3
model.smachine[1].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "B.def"
model.objsvr[2].cfg_file = "C.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0

## pat15 [BR733] [SIX] [SIX] ##
[Run 15]
model.run_name = "pat15"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 2
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "B.def"
model.objsvr[2].cfg_file = "C.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0
## pat16 [GIGA] [BR733] [SIX] ##
[Run 16]
model.run_name = "pat16"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 1
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "C.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0

## pat17 [SIX] [BR733] [SIX] ##
[Run 17]
model.run_name = "pat17"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 2
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.objsvr[2].cfg_file = "C.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0

## pat18 [BR733] [BR733] [SIX] ##
[Run 18]
model.run_name = "pat18"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 2
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "B.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 2
model.objsvr[2].cfg_file = "C.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0

## pat19 [GIGA] [GIGA] [BR733] ##
[Run 19]
model.run_name = "pat19"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "C.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 2
model.smachine[2].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.objsvr[2].cfg_file = "B.def"

## pat20 [SIX] [GIGA] [BR733] ##
[Run 20]
model.run_name = "pat20"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 1
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "C.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "B.def"

## pat21 [BR733] [GIGA] [BR733] ##
[Run 21]
model.run_name = "pat21"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 2
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "C.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "A.def"

## pat22 [GIGA] [SIX] [BR733] ##
[Run 22]
model.run_name = "pat22"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 1
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "C.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "B.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0
model.objsvr[2].cfg_file = "A.def"

## pat23 [SIX] [SIX] [BR733] ##
[Run 23]
model.run_name = "pat23"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 1
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "C.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 2
model.smachine[1].objsvridx = 1
model.objsvr[1].cfg_file = "A.def"
model.objsvr[2].cfg_file = "B.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0

169
## pat24 [BR733] [SIX] [BR733] ##

[Run 24]
model.run_name = "pat24"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 2
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "A.def"
model.objsvr[0].cfg_file = "C.def"
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 2
model.objsvr[2].cfg_file = "B.def"
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 0

## pat25 [GIGA] [BR733] [BR733] ##

[Run 25]
model.run_name = "pat25"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 2
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"
model.objsvr[1].cfg_file = "C.def"
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0
model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 1
model.smachine[2].objsvridx = 2
model.objsvr[2].cfg_file = "A.def"

## pat26 [SIX] [BR733] [BR733] ##

[Run 26]
model.run_name = "pat26"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 2
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 2
model.smachine[0].objsvridx = 0
model.objsvr[0].cfg_file = "B.def"
model.objsvr[1].cfg_file = "C.def"
model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 1
model.smachine[1].objsvridx = 2
model.objsvr[2].cfg_file = "A.def"

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0

## pat27 [BR733] [BR733] [BR733] ##
[Run 27]
model.run_name = "pat27"
model.num_machine = 3
model.num_objsvr = 3
model.max_objsvr = 3
model.smachine[0].def_file = "BR733.def"
model.smachine[0].num_objsvr = 3
model.objsvr[0].cfg_file = "A.def"
model.objsvr[1].cfg_file = "B.def"
model.objsvr[2].cfg_file = "C.def"

model.smachine[1].def_file = "SIX.def"
model.smachine[1].num_objsvr = 0
model.smachine[1].objsvridx = 0

model.smachine[2].def_file = "GIGA.def"
model.smachine[2].num_objsvr = 0
model.smachine[2].objsvridx = 0
B. DETAILED RESULT

1. Simulation Run 1 (1 role 1, Ram Limit 0.0)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
<td>Min</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1750</td>
<td>0.9558</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1734</td>
<td>1.0962</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1738</td>
<td>1.0558</td>
</tr>
<tr>
<td>4</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1718</td>
<td>1.2533</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1720</td>
<td>1.2313</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1727</td>
<td>1.1837</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>1716</td>
<td>1.2751</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>1723</td>
<td>1.1935</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1736</td>
<td>1.0857</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1721</td>
<td>1.2209</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>1711</td>
<td>1.3226</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1696</td>
<td>1.4807</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1706</td>
<td>1.3782</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1693</td>
<td>1.4893</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>1709</td>
<td>1.3452</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1715</td>
<td>1.2877</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1697</td>
<td>1.4664</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>1689</td>
<td>1.5399</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1690</td>
<td>1.5391</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1693</td>
<td>1.5078</td>
</tr>
<tr>
<td>21</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1724</td>
<td>1.2014</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1739</td>
<td>1.058</td>
</tr>
<tr>
<td>23</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1714</td>
<td>1.2905</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1717</td>
<td>1.2798</td>
</tr>
<tr>
<td>25</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1692</td>
<td>1.519</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1701</td>
<td>1.4169</td>
</tr>
</tbody>
</table>
## 2. Simulation Run 2 (1 role 2, Ram Limit 0.0)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
<td>Min</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2449</td>
<td>5.2635</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>2356</td>
<td>5.7279</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>2238</td>
<td>6.371</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>2179</td>
<td>6.7169</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2328</td>
<td>5.8716</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>2246</td>
<td>6.3193</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>2166</td>
<td>6.7915</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>2118</td>
<td>7.107</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>2321</td>
<td>5.9121</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>2119</td>
<td>7.0883</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>2029</td>
<td>7.6891</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2243</td>
<td>6.3535</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>2156</td>
<td>6.8569</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1967</td>
<td>8.1308</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>1890</td>
<td>8.7288</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>2063</td>
<td>7.4566</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1990</td>
<td>7.955</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>1966</td>
<td>8.141</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>2062</td>
<td>7.4566</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1990</td>
<td>7.9663</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1929</td>
<td>8.4218</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>2131</td>
<td>7.0065</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>2074</td>
<td>7.4031</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>2216</td>
<td>6.4926</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>2029</td>
<td>7.681</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>2179</td>
<td>6.7215</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>2083</td>
<td>7.318</td>
</tr>
</tbody>
</table>
### Simulation Run 3 (1 role 3, Ram Limit 0.0)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave</td>
<td>Min</td>
<td>Max</td>
<td>Ave</td>
<td>Min</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>260</td>
<td>9.284</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>227</td>
<td>11.2949</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>245</td>
<td>10.1469</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>212</td>
<td>12.4098</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>256</td>
<td>9.4766</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>229</td>
<td>11.1748</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>241</td>
<td>10.3449</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>208</td>
<td>12.7509</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>212</td>
<td>12.5131</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>235</td>
<td>10.7928</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>196</td>
<td>13.8507</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>256</td>
<td>9.5256</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>203</td>
<td>13.2308</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>229</td>
<td>11.1858</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>183</td>
<td>15.0968</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>194</td>
<td>13.9055</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>205</td>
<td>12.9896</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>190</td>
<td>14.4321</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>211</td>
<td>12.4835</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>191</td>
<td>14.2588</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>200</td>
<td>13.3633</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>197</td>
<td>13.7414</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>205</td>
<td>12.9471</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>200</td>
<td>13.4877</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>239</td>
<td>10.5546</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>229</td>
<td>11.1704</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>236</td>
<td>10.703</td>
</tr>
</tbody>
</table>
### Simulation Run 4 (4 role 2, Ram Limit 0.0)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
<th>Ave</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1336</td>
<td></td>
<td>15.0493</td>
<td>0.661525</td>
<td>36.60915</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1584</td>
<td></td>
<td>11.67535</td>
<td>0.661275</td>
<td>25.8519</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1605</td>
<td></td>
<td>11.43463</td>
<td>0.6612</td>
<td>31.58019</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1307</td>
<td></td>
<td>15.53168</td>
<td>0.66125</td>
<td>48.36535</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1605</td>
<td></td>
<td>11.43985</td>
<td>0.872325</td>
<td>33.33738</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1702</td>
<td></td>
<td>10.41733</td>
<td>0.8723</td>
<td>30.30368</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1216</td>
<td></td>
<td>17.17995</td>
<td>0.872475</td>
<td>33.82055</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>995</td>
<td></td>
<td>22.42225</td>
<td>0.8726</td>
<td>50.05655</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1584</td>
<td></td>
<td>11.67603</td>
<td>0.661225</td>
<td>28.12353</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1431</td>
<td></td>
<td>13.61095</td>
<td>0.661225</td>
<td>40.11288</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>1120</td>
<td></td>
<td>19.21055</td>
<td>0.6612</td>
<td>58.8928</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1548</td>
<td></td>
<td>12.10203</td>
<td>1.0476</td>
<td>34.61813</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1540</td>
<td></td>
<td>12.19353</td>
<td>1.047625</td>
<td>32.2734</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1010</td>
<td></td>
<td>22.01543</td>
<td>1.04835</td>
<td>39.9369</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>836</td>
<td></td>
<td>27.93548</td>
<td>1.0483</td>
<td>62.90595</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1207</td>
<td></td>
<td>17.35225</td>
<td>0.8727</td>
<td>34.48555</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1327</td>
<td></td>
<td>15.18999</td>
<td>0.8723</td>
<td>41.12468</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>1108</td>
<td></td>
<td>19.47055</td>
<td>0.872325</td>
<td>61.46175</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1277</td>
<td></td>
<td>16.03185</td>
<td>1.047625</td>
<td>48.69093</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1413</td>
<td></td>
<td>13.88233</td>
<td>1.047675</td>
<td>38.25873</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1010</td>
<td></td>
<td>21.99245</td>
<td>1.04785</td>
<td>41.59348</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1633</td>
<td></td>
<td>11.12345</td>
<td>0.6612</td>
<td>34.45988</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1437</td>
<td></td>
<td>13.5477</td>
<td>0.6612</td>
<td>39.30033</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1845</td>
<td></td>
<td>9.10485</td>
<td>0.872325</td>
<td>25.15523</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1399</td>
<td></td>
<td>14.08285</td>
<td>0.872325</td>
<td>39.62575</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1750</td>
<td></td>
<td>9.950025</td>
<td>1.047625</td>
<td>24.67665</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1544</td>
<td></td>
<td>12.15075</td>
<td>1.0476</td>
<td>32.21468</td>
</tr>
</tbody>
</table>
## 5. Simulation Run 5 (3 role 3, Ram Limit 0.0)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1361</td>
<td>16.6589</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1755</td>
<td>11.9086</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1662</td>
<td>12.8099</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1121</td>
<td>21.1609</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1483</td>
<td>14.9085</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1430</td>
<td>15.6307</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1678</td>
<td>12.6598</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>1000</td>
<td>24.2953</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1480</td>
<td>14.9543</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1520</td>
<td>14.4386</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>917</td>
<td>26.8708</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1461</td>
<td>15.2147</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1187</td>
<td>19.7351</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1475</td>
<td>15.0132</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>819</td>
<td>30.6386</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1405</td>
<td>15.9958</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1309</td>
<td>17.4819</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>918</td>
<td>26.831633</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1111</td>
<td>21.419166</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1193</td>
<td>19.6194</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1405</td>
<td>15.995433</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1468</td>
<td>15.115366</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1540</td>
<td>14.191833</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1487</td>
<td>14.8577</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1636</td>
<td>13.0968</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1711</td>
<td>12.327733</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1774</td>
<td>11.7289</td>
</tr>
</tbody>
</table>
### 6. Simulation Run 6 (28 role 1, Ram Limit 1.5)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1117</td>
<td>10.2673</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1404</td>
<td>5.011779</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1253</td>
<td>7.468704</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1543</td>
<td>3.160807</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1248</td>
<td>7.5642</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>933</td>
<td>15.3563</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1071</td>
<td>11.38449</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>842</td>
<td>18.70253</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1404</td>
<td>5.002846</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1245</td>
<td>7.616296</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1022</td>
<td>12.67358</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1069</td>
<td>11.41627</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>936</td>
<td>15.23539</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1020</td>
<td>12.72855</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1550</td>
<td>3.075521</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1559</td>
<td>2.962407</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1244</td>
<td>7.646021</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1249</td>
<td>7.548154</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1023</td>
<td>12.64221</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1029</td>
<td>12.47938</td>
</tr>
</tbody>
</table>
7. Simulation Run 7 (1 role 1, Ram Limit 1.5)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1744</td>
<td>0.6702</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1739</td>
<td>0.6702</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1737</td>
<td>0.6702</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1731</td>
<td>0.6702</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1727</td>
<td>0.8813</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1709</td>
<td>0.8813</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1723</td>
<td>0.8813</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>1717</td>
<td>0.8813</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1729</td>
<td>0.6702</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1735</td>
<td>0.6702</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1699</td>
<td>1.0566</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1703</td>
<td>0.8813</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1708</td>
<td>0.8813</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1689</td>
<td>1.0566</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1742</td>
<td>0.6702</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1729</td>
<td>0.6702</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1712</td>
<td>0.8813</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1722</td>
<td>0.8813</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1707</td>
<td>1.0566</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1698</td>
<td>1.0566</td>
</tr>
</tbody>
</table>
### Simulation Run 8 (1 role 2, Ram Limit 1.5)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2440</td>
<td>0.6702</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>2362</td>
<td>0.6702</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>2235</td>
<td>0.6702</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>2197</td>
<td>0.6702</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2325</td>
<td>0.8813</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>2264</td>
<td>0.8813</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>2151</td>
<td>0.8813</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>2091</td>
<td>0.8813</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>2312</td>
<td>0.6702</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>2073</td>
<td>0.6702</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2258</td>
<td>1.0566</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>2050</td>
<td>0.8813</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1965</td>
<td>0.8813</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>2014</td>
<td>1.0566</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>2138</td>
<td>0.6702</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>2069</td>
<td>0.6702</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>2220</td>
<td>0.8813</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>2019</td>
<td>0.8813</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>2182</td>
<td>1.0566</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>2084</td>
<td>1.0566</td>
</tr>
</tbody>
</table>
# Simulation Run 9 (1 role 3, Ram Limit 1.5)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
<td>Min</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2507</td>
<td>0.5802</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>2245</td>
<td>0.758</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>2293</td>
<td>0.5806</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>2089</td>
<td>0.7589</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2431</td>
<td>0.5806</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>2185</td>
<td>0.7582</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>2271</td>
<td>0.5803</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>2051</td>
<td>0.758</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>2059</td>
<td>0.9057</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>2170</td>
<td>0.5802</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>2381</td>
<td>0.5802</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1903</td>
<td>0.9058</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1959</td>
<td>0.7583</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>2020</td>
<td>0.758</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>error</td>
<td>error</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1941</td>
<td>0.9059</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1967</td>
<td>0.7583</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>2017</td>
<td>0.906</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>2115</td>
<td>0.5805</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>2160</td>
<td>0.7583</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>2218</td>
<td>0.5803</td>
</tr>
</tbody>
</table>
## 10. Simulation Run 5 with Time Slice 0.3 sec

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1296</td>
<td>0.7865</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1694</td>
<td>0.812967</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1676</td>
<td>0.599133</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1068</td>
<td>2.7078</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1443</td>
<td>1.2133</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1402</td>
<td>0.873833</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1665</td>
<td>0.6028</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>950</td>
<td>2.3349</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1459</td>
<td>2.354367</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1546</td>
<td>0.817167</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>875</td>
<td>1.941167</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1437</td>
<td>0.6167</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1174</td>
<td>1.4006</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1449</td>
<td>0.602533</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>771</td>
<td>1.909433</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1375</td>
<td>1.1941</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1301</td>
<td>0.845133</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>879</td>
<td>2.5394</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1062</td>
<td>2.161067</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1172</td>
<td>1.3686</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1372</td>
<td>0.8039</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1437</td>
<td>1.352033</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1519</td>
<td>0.815633</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1432</td>
<td>1.3278</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1657</td>
<td>0.602733</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1661</td>
<td>0.7996</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1786</td>
<td>0.620633</td>
</tr>
</tbody>
</table>
## 11. Simulation Run 5 with Time Slice 0.5 sec

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time Ave</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1318</td>
<td>2.0484</td>
<td>44.17643</td>
<td>17.33753</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1739</td>
<td>0.7884</td>
<td>28.29127</td>
<td>12.0512</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1719</td>
<td>0.5932</td>
<td>32.17673</td>
<td>12.2479</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1102</td>
<td>2.284333</td>
<td>45.7475</td>
<td>21.61923</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1489</td>
<td>1.881233</td>
<td>33.63357</td>
<td>14.834</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1427</td>
<td>0.837733</td>
<td>48.7212</td>
<td>15.67527</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1702</td>
<td>0.593333</td>
<td>38.26807</td>
<td>12.4122</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>977</td>
<td>1.992567</td>
<td>64.39967</td>
<td>24.9665</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1507</td>
<td>1.257933</td>
<td>32.03333</td>
<td>14.59767</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1587</td>
<td>0.585567</td>
<td>35.26557</td>
<td>13.64613</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>909</td>
<td>2.675267</td>
<td>60.22457</td>
<td>27.1556</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1468</td>
<td>1.200933</td>
<td>33.7923</td>
<td>15.1225</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1203</td>
<td>1.574133</td>
<td>60.17807</td>
<td>19.43113</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1515</td>
<td>0.592933</td>
<td>44.8829</td>
<td>14.50497</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>801</td>
<td>1.937767</td>
<td>84.2609</td>
<td>31.42537</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1403</td>
<td>1.058833</td>
<td>40.1938</td>
<td>16.02557</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1350</td>
<td>0.785733</td>
<td>46.17747</td>
<td>16.82987</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>904</td>
<td>3.8977</td>
<td>60.2902</td>
<td>27.32653</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1098</td>
<td>3.553667</td>
<td>46.7826</td>
<td>21.7159</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1197</td>
<td>1.749233</td>
<td>56.48117</td>
<td>19.56097</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1395</td>
<td>0.781233</td>
<td>50.98</td>
<td>16.13057</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1471</td>
<td>1.246533</td>
<td>33.96027</td>
<td>15.0774</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1549</td>
<td>0.787467</td>
<td>34.27813</td>
<td>14.07673</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1483</td>
<td>1.208467</td>
<td>33.32643</td>
<td>14.90627</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1700</td>
<td>0.590367</td>
<td>30.50503</td>
<td>12.43387</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1705</td>
<td>0.775633</td>
<td>31.50473</td>
<td>12.38327</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1823</td>
<td>0.589767</td>
<td>29.79467</td>
<td>11.29057</td>
</tr>
</tbody>
</table>
### Simulation Run 5 with Time Slice 1.0 sec

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ave</td>
</tr>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1357</td>
<td>1.505133</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1759</td>
<td>0.763</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1727</td>
<td>0.580767</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1132</td>
<td>1.316267</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1515</td>
<td>0.590167</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1455</td>
<td>0.763533</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1724</td>
<td>0.582167</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>1002</td>
<td>2.175533</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1521</td>
<td>1.179</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1615</td>
<td>0.580767</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>929</td>
<td>4.623</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1498</td>
<td>1.309533</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1217</td>
<td>1.237833</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1528</td>
<td>0.581633</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>824</td>
<td>1.798433</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1427</td>
<td>1.2559</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1366</td>
<td>0.788967</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>925</td>
<td>5.0801</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1121</td>
<td>2.239667</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1218</td>
<td>1.3768</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1411</td>
<td>0.7601</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1496</td>
<td>1.0366</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1564</td>
<td>0.770967</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1511</td>
<td>0.9428</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1709</td>
<td>0.5815</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1718</td>
<td>0.760133</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1817</td>
<td>0.582533</td>
</tr>
</tbody>
</table>
13. Simulation Run 5 with Time Slice 2.0 sec

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
<th>Number of Call</th>
<th>Average Response Time</th>
<th>Ave</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GIGA</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1381</td>
<td></td>
<td>1.4063</td>
<td>40.20587</td>
<td>16.34963</td>
</tr>
<tr>
<td>2</td>
<td>GIGA</td>
<td>GIGA</td>
<td>BR733</td>
<td>1752</td>
<td></td>
<td>0.759233</td>
<td>39.85037</td>
<td>11.93463</td>
</tr>
<tr>
<td>3</td>
<td>GIGA</td>
<td>BR733</td>
<td>GIGA</td>
<td>1704</td>
<td></td>
<td>0.5805</td>
<td>33.02707</td>
<td>12.39853</td>
</tr>
<tr>
<td>4</td>
<td>GIGA</td>
<td>BR733</td>
<td>BR733</td>
<td>1146</td>
<td></td>
<td>1.198833</td>
<td>43.55523</td>
<td>20.61667</td>
</tr>
<tr>
<td>5</td>
<td>BR733</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1527</td>
<td></td>
<td>1.643933</td>
<td>32.5286</td>
<td>14.36297</td>
</tr>
<tr>
<td>6</td>
<td>BR733</td>
<td>GIGA</td>
<td>BR733</td>
<td>1460</td>
<td></td>
<td>0.759467</td>
<td>46.77287</td>
<td>15.21773</td>
</tr>
<tr>
<td>7</td>
<td>BR733</td>
<td>BR733</td>
<td>GIGA</td>
<td>1707</td>
<td></td>
<td>0.581233</td>
<td>35.58317</td>
<td>12.3638</td>
</tr>
<tr>
<td>8</td>
<td>BR733</td>
<td>BR733</td>
<td>BR733</td>
<td>1012</td>
<td></td>
<td>3.217367</td>
<td>61.37557</td>
<td>23.9637</td>
</tr>
<tr>
<td>9</td>
<td>GIGA</td>
<td>GIGA</td>
<td>SIX</td>
<td>1514</td>
<td></td>
<td>1.980367</td>
<td>34.20853</td>
<td>14.51863</td>
</tr>
<tr>
<td>10</td>
<td>GIGA</td>
<td>SIX</td>
<td>GIGA</td>
<td>1612</td>
<td></td>
<td>0.580833</td>
<td>33.20963</td>
<td>13.35987</td>
</tr>
<tr>
<td>11</td>
<td>GIGA</td>
<td>SIX</td>
<td>SIX</td>
<td>936</td>
<td></td>
<td>4.023167</td>
<td>56.5411</td>
<td>26.261</td>
</tr>
<tr>
<td>12</td>
<td>SIX</td>
<td>GIGA</td>
<td>GIGA</td>
<td>1504</td>
<td></td>
<td>0.5862</td>
<td>34.12003</td>
<td>14.64633</td>
</tr>
<tr>
<td>13</td>
<td>SIX</td>
<td>GIGA</td>
<td>SIX</td>
<td>1229</td>
<td></td>
<td>2.014333</td>
<td>55.57123</td>
<td>18.9297</td>
</tr>
<tr>
<td>14</td>
<td>SIX</td>
<td>SIX</td>
<td>GIGA</td>
<td>1506</td>
<td></td>
<td>0.580667</td>
<td>43.11107</td>
<td>14.62017</td>
</tr>
<tr>
<td>15</td>
<td>SIX</td>
<td>SIX</td>
<td>SIX</td>
<td>836</td>
<td></td>
<td>1.8724</td>
<td>75.9428</td>
<td>29.8945</td>
</tr>
<tr>
<td>16</td>
<td>BR733</td>
<td>BR733</td>
<td>SIX</td>
<td>1433</td>
<td></td>
<td>0.911367</td>
<td>37.33553</td>
<td>15.59267</td>
</tr>
<tr>
<td>17</td>
<td>BR733</td>
<td>SIX</td>
<td>BR733</td>
<td>1362</td>
<td></td>
<td>0.758833</td>
<td>44.30693</td>
<td>16.6259</td>
</tr>
<tr>
<td>18</td>
<td>BR733</td>
<td>SIX</td>
<td>SIX</td>
<td>935</td>
<td></td>
<td>4.141867</td>
<td>57.2442</td>
<td>26.29647</td>
</tr>
<tr>
<td>19</td>
<td>SIX</td>
<td>BR733</td>
<td>BR733</td>
<td>1131</td>
<td></td>
<td>2.122667</td>
<td>45.03013</td>
<td>20.9292</td>
</tr>
<tr>
<td>20</td>
<td>SIX</td>
<td>BR733</td>
<td>SIX</td>
<td>1221</td>
<td></td>
<td>0.909767</td>
<td>56.9247</td>
<td>19.0871</td>
</tr>
<tr>
<td>21</td>
<td>SIX</td>
<td>SIX</td>
<td>BR733</td>
<td>1400</td>
<td></td>
<td>0.758867</td>
<td>45.0065</td>
<td>16.06243</td>
</tr>
<tr>
<td>22</td>
<td>GIGA</td>
<td>BR733</td>
<td>SIX</td>
<td>1491</td>
<td></td>
<td>1.128633</td>
<td>41.00177</td>
<td>14.8088</td>
</tr>
<tr>
<td>23</td>
<td>GIGA</td>
<td>SIX</td>
<td>BR733</td>
<td>1562</td>
<td></td>
<td>0.759833</td>
<td>39.01757</td>
<td>13.92577</td>
</tr>
<tr>
<td>24</td>
<td>BR733</td>
<td>GIGA</td>
<td>SIX</td>
<td>1505</td>
<td></td>
<td>1.323267</td>
<td>34.0198</td>
<td>14.62723</td>
</tr>
<tr>
<td>25</td>
<td>BR733</td>
<td>SIX</td>
<td>GIGA</td>
<td>1692</td>
<td></td>
<td>0.581067</td>
<td>31.5452</td>
<td>12.51057</td>
</tr>
<tr>
<td>26</td>
<td>SIX</td>
<td>GIGA</td>
<td>BR733</td>
<td>1716</td>
<td></td>
<td>0.7604</td>
<td>33.4506</td>
<td>12.283</td>
</tr>
<tr>
<td>27</td>
<td>SIX</td>
<td>BR733</td>
<td>GIGA</td>
<td>1795</td>
<td></td>
<td>0.580667</td>
<td>34.9834</td>
<td>11.53953</td>
</tr>
</tbody>
</table>
APPENDIX D – VERIFICATION EXPERIMENT 2

A. CONFIGURATION FILES

Object server A – A.def
[name] A
[ram_util] 139000
[object] A
[num_obj] 1

Object server B – B.def
[name] B
[ram_util] 122000
[object] B
[num_obj] 1

Object server C – C.def
[name] C
[ram_util] 145000
[object] C
[num_obj] 1

Object server D – D.def
[name] D
[ram_util] 153000
[object] D
[num_obj] 1

Object server E – E.def
[name] E
[ram_util] 231000
[object] E
[num_obj] 1

Object server F – F.def
[name] F
[ram_util] 130000
[object] F
[num_obj] 1

Object server G – G.def
[name] G
[ram_util] 142000
[object] G
[num_obj] 1
Object server H – H.def
[name] H
[ram_util] 200000
[object] H
[num_obj] 1

Object server I – I.def
[name] I
[ram_util] 189000
[object] I
[num_obj] 1

Object server J – J.def
[name] J
[ram_util] 80000
[object] J
[num_obj] 1

Role0 Configuration File – role0.def
[name] ROLE0
[type] 0
[min_wait] 15
[max_wait] 16
[call] F2.B1/10
[dr_sw] 200000000
[btw_call] 0.05

Role1 Configuration File – role1.def
[name] ROLE1_0
[type] 1
[min_wait] 15
[max_wait] 16
[dr_sw] 200000000
[btw_call] 0.05

Role2 Configuration File – role2.def
[name] ROLE2_0
[type] 2
[min_wait] 12
[max_wait] 13
Role3 Configuration File – role3.def

[name] ROLE3_0
[type] 3
[min_wait] 39
[max_wait] 40
[call] F2.B3/30
[dr_sw] 200000000
[btw_call] 0.05

MACHINE Handel – Handel.def

[name] HANDEL
[cpu_pow] 2400000
[ram_sz] 512000
[ram_limit] 1.0
[dr_sw] 200000000
[er_sw] 0.000000001
[process_call] 0.005
[process_swap] 0.005
[exec_time] 0.5
[ran_exec] 0.05
[time_slice] no

MACHINE MOZART – Mozart.def

[name] MOZART
[cpu_pow] 2000000
[ram_sz] 256000
[ram_limit] 1.0
[dr_sw] 200000000
[er_sw] 0.000000001
[process_call] 0.005
[process_swap] 0.005
[exec_time] 0.5
[ran_exec] 0.05
[time_slice] no

MACHINE BEETHOVAN – BEE.def

[name] BEETHOVEN
[cpu_pow] 3000000
[ram_sz] 1000000
[ram_limit] 1.0
[dr_sw] 200000000
[er_sw] 0.000000001
[process_call] 0.005
[process_swap] 0.005
[disk_swap] 0.005
[exec_time] 0.5
[ran_exec] 0.05
[time_slice] no

**Button Configuration File – button.def**

```
[NEW_DEF]
F1.B1
G/2
C/2
A/1
[END_DEF]

[NEW_DEF]
F1.B2
F/3
[END_DEF]

[NEW_DEF]
F1.B3
H/2
[END_DEF]

[NEW_DEF]
F2.B1
E/2
G/2
[END_DEF]

[NEW_DEF]
F2.B2
J/4
B/2
[END_DEF]

[NEW_DEF]
F2.B3
D/2
[END_DEF]

[NEW_DEF]
F2.B4
B/2
E/1
[END_DEF]

[NEW_DEF]
F3.B1
J/3
[END_DEF]

[NEW_DEF]
F3.B2
F/2
B/1
E/2
F/3
[END_DEF]
```
Interaction Configuration File – *interact.def*

```
[NEW_DEF]
F4.B1
J/2
A/2
[END_DEF]

[NEW_DEF]
F5.B1
H/1
F/3
J/3
[END_DEF]

[NEW_DEF]
F5.B2
B/1
E/2
[END_DEF]

[NEW_DEF]
F5.B3
I/1
J/2
[END_DEF]

[NEW_DEF]
F5.B4
D/1
I/1
[END_DEF]

[NEW_DEF]
F5.B5
I/1
[END_DEF]

[NEW_INTERACT] D/2
[CALL] A/2
[END_INTERACT]

[NEW_INTERACT] J/2
[CALL] B/1
[END_INTERACT]

[NEW_INTERACT] J/3
[CALL] I/1
[END_INTERACT]

[NEW_INTERACT] E/1
[CALL] F/5
[END_INTERACT]

[NEW_INTERACT] H/1
[CALL] G/3
[END_INTERACT]
```
Object Configuration File – obj.def

[NEW_OBJ_DEF]
A/10000
1/2345600/216000
2/1318100/2296000
3/1963200/1736000
4/1718500/528000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
B/10000
1/2587600/2752000
2/686200/736000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
C/10000
1/1482700/1968000
2/1944700/2824000
3/1087000/3728000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
D/10000
1/975000/3752000
2/2354300/1912000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
E/10000
1/1678500/3640000
2/1002900/2984000
3/2106100/2616000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
F/10000
1/2398600/2800000
2/2402200/2064000
3/2434300/3328000
4/562300/3632000
5/2028300/3800000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
G/10000
1/2065800/2968000
2/1724300/104000
3/1138200/3848000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
H/10000
1/580500/2552000
2/1111500/3384000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
I/10000
1/1131400/1336000
[END_OBJ_DEF]
[NEW_OBJ_DEF]
J/10000
Role Configuration File

[Parameters]
model.role[0].def_file = "role0_0.def"
model.role[1].def_file = "role0_1.def"
model.role[2].def_file = "role0_2.def"
model.role[3].def_file = "role0_3.def"
model.role[4].def_file = "role0_4.def"
model.role[5].def_file = "role0_5.def"
model.role[6].def_file = "role0_6.def"
model.role[7].def_file = "role0_7.def"
model.role[8].def_file = "role0_8.def"
model.role[9].def_file = "role0_9.def"
model.role[10].def_file = "role0_10.def"
model.role[11].def_file = "role0_11.def"
model.role[12].def_file = "role0_12.def"
model.role[13].def_file = "role0_13.def"
model.role[14].def_file = "role0_14.def"
model.role[15].def_file = "role0_15.def"
model.role[16].def_file = "role0_16.def"
model.role[17].def_file = "role0_17.def"
model.role[18].def_file = "role0_18.def"
model.role[19].def_file = "role0_19.def"
model.role[20].def_file = "role1_0.def"
model.role[21].def_file = "role1_1.def"
model.role[22].def_file = "role1_2.def"
model.role[23].def_file = "role1_3.def"
model.role[24].def_file = "role1_4.def"
model.role[25].def_file = "role2_0.def"
model.role[26].def_file = "role2_1.def"
model.role[27].def_file = "role2_2.def"
model.role[28].def_file = "role2_3.def"
model.role[29].def_file = "role2_4.def"
model.role[30].def_file = "role2_5.def"
model.role[31].def_file = "role2_6.def"
model.role[32].def_file = "role2_7.def"
model.role[33].def_file = "role3_0.def"
model.role[34].def_file = "role3_1.def"
model.role[35].def_file = "role3_2.def"
model.role[36].def_file = "role3_3.def"
model.role[37].def_file = "role3_4.def"
model.role[38].def_file = "role3_5.def"
model.role[39].def_file = "role3_6.def"
model.role[40].def_file = "role3_7.def"
model.role[41].def_file = "role3_8.def"
model.role[42].def_file = "role3_9.def"
model.num_role = 43
B. LINGO MODEL

MODEL:

SETS:
  MACHINE / Mozart Handel Beethoven /
  MEMORY, SPEED;
  SERVER / A B C D E F G H I J /
  MULTIPLIER, MEMORYUSE;
  NET_SPD (SERVER, SERVER): U;
  DEPLOYMENT (MACHINE, SERVER): V;
  MEM_USED (MACHINE): T;
  CPU_USED (MACHINE): Q;
  CYCLE_AVAIL (MACHINE) : CPU_AVAIL;
ENDSETS

DATA:
  MEMORY SPEED =
    256    2000000
    512    2400000
    1000   3000000;
  MULTIPLIER MEMORYUSE =
    1733150000  139
    901800000   122
    456300000   145
    433500000   153
    591340000   231
    356360000   130
    1297450000  142
    111600000   200
    230360000   189
    981352000   80;
  MEM_LIMIT = 1;
  NET_BW = 100000000;
  CPU_TIME = 1200;
ENDDATA

MIN = PROC_SPEED + NET_SPEED;

! PROC_SPEED = @SUM( DEPLOYMENT( I, J ) :
!    V ( I, J ) * MULTIPLIER ( J ) * NORM_SPEED / SPEED( I ) );
!
PROC_SPEED = @SUM( DEPLOYMENT( I, J ) :
    V ( I, J ) * MULTIPLIER ( J ) );

! Inter-Server communications function. Ignore Client/Server Comms
! because they always exist and we are letting the Client location
! be the free variable. NOTE: ASSUME LOCAL TWICE AS FAST AS REMOTE
!
NET_SPEED = 1653120000/(U(@INDEX(D),@INDEX(A))*NET_BW) +
3434496000/(U(@INDEX(J),@INDEX(B))*NET_BW) +
352704000/(U(@INDEX(J),@INDEX(I))*NET_BW) +
456000000/(U(@INDEX(E),@INDEX(F))*NET_BW) +
923520000/(U(@INDEX(H),@INDEX(G))*NET_BW);

NET_SPEED = 68880000/(U(@INDEX(D),@INDEX(A))*NET_BW) +
143104000/(U(@INDEX(J),@INDEX(B))*NET_BW) +
14696000/(U(@INDEX(J),@INDEX(I))*NET_BW) +
19000000/(U(@INDEX(E),@INDEX(F))*NET_BW) +
38480000/(U(@INDEX(H),@INDEX(G))*NET_BW);

@FOR (SERVER(K):
  @FOR (SERVER(L):
    U(K,L) = @SUM ( MACHINE(R): V(R,K)*V(R,L)) + 1;
  );
); A server cannot be split over multiple machines
@FOR (DEPLOYMENT: @BIN(V));

Each server can only run on one machine.
@FOR (SERVER(K):
  @SUM ( MACHINE(R): V(R, K)) = 1;
 );

Constraint for limiting the RAM load on a single machine.
@FOR (MACHINE(R):
  T(R) = @SUM ( SERVER(K): V(R, K)*MEMORYUSE(K));
  T(R) < MEMORY(R)*MEM_LIMIT;
 );

Computing the number of instruction cycle available.
@FOR (MACHINE(R):
  CPU_AVAIL(R) = CPU_TIME * SPEED(R);
 );

Constraint for limiting the CPU load on a single machine.
@FOR (MACHINE(R):
  Q(R) = @SUM (SERVER(K): V(R, K)*MULTIPLIER(K));
  (Q(R)/1000) < (CPU_AVAIL(R)/1000);
 );
### C. DETAILED RESULT

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Role1</th>
<th>Role2</th>
<th>Role3</th>
<th>Role4</th>
<th>Model Value (Equal Weight)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pat108</td>
<td>3.5795</td>
<td>2.1258</td>
<td>3.1042</td>
<td>3.2595</td>
<td>3.01725</td>
</tr>
<tr>
<td>pat10944</td>
<td>4.3416</td>
<td>2.8207</td>
<td>3.5431</td>
<td>3.5956</td>
<td>3.57525</td>
</tr>
<tr>
<td>pat10954</td>
<td>13.5533</td>
<td>10.75</td>
<td>14.3082</td>
<td>12.5766</td>
<td>12.797025</td>
</tr>
<tr>
<td>pat10956</td>
<td>4.1629</td>
<td>3.2762</td>
<td>4.1032</td>
<td>3.8242</td>
<td>3.841625</td>
</tr>
<tr>
<td>pat10990</td>
<td>14.8603</td>
<td>10.2881</td>
<td>12.2111</td>
<td>7.1023</td>
<td>11.11545</td>
</tr>
<tr>
<td>pat10992</td>
<td>4.7059</td>
<td>3.2482</td>
<td>4.0163</td>
<td>2.5143</td>
<td>3.621175</td>
</tr>
<tr>
<td>pat10996</td>
<td>11.1938</td>
<td>6.5015</td>
<td>8.1772</td>
<td>5.8929</td>
<td>7.94135</td>
</tr>
<tr>
<td>pat11008</td>
<td>10.5391</td>
<td>8.364</td>
<td>11.1136</td>
<td>6.6134</td>
<td>9.157525</td>
</tr>
<tr>
<td>pat11098</td>
<td>14.0368</td>
<td>8.6305</td>
<td>12.6567</td>
<td>11.0602</td>
<td>11.59605</td>
</tr>
<tr>
<td>pat11424</td>
<td>5.7975</td>
<td>3.4991</td>
<td>4.9229</td>
<td>4.422</td>
<td>4.660375</td>
</tr>
<tr>
<td>pat11428</td>
<td>13.5752</td>
<td>5.2758</td>
<td>9.996</td>
<td>11.2167</td>
<td>10.015925</td>
</tr>
<tr>
<td>pat11440</td>
<td>12.3339</td>
<td>7.0866</td>
<td>13.615</td>
<td>12.6974</td>
<td>11.433225</td>
</tr>
<tr>
<td>pat1224</td>
<td>2.8819</td>
<td>2.0639</td>
<td>2.5412</td>
<td>2.6198</td>
<td>2.5267</td>
</tr>
<tr>
<td>pat1236</td>
<td>2.8283</td>
<td>1.7879</td>
<td>2.4133</td>
<td>2.5902</td>
<td>2.404925</td>
</tr>
<tr>
<td>pat12394</td>
<td>8.6625</td>
<td>7.2423</td>
<td>9.6251</td>
<td>8.2768</td>
<td>8.451675</td>
</tr>
<tr>
<td>pat12396</td>
<td>5.0022</td>
<td>2.4136</td>
<td>3.5997</td>
<td>4.2395</td>
<td>3.81375</td>
</tr>
<tr>
<td>pat1240</td>
<td>3.2182</td>
<td>1.8142</td>
<td>4.1397</td>
<td>3.757</td>
<td>3.232275</td>
</tr>
<tr>
<td>pat12400</td>
<td>5.688</td>
<td>4.0441</td>
<td>5.6607</td>
<td>5.101</td>
<td>5.12345</td>
</tr>
<tr>
<td>pat12412</td>
<td>5.2784</td>
<td>5.2051</td>
<td>7.3662</td>
<td>6.0677</td>
<td>5.97935</td>
</tr>
<tr>
<td>pat12448</td>
<td>6.1746</td>
<td>5.011</td>
<td>6.5349</td>
<td>4.578</td>
<td>5.574625</td>
</tr>
<tr>
<td>pat1272</td>
<td>2.8749</td>
<td>1.8383</td>
<td>2.4398</td>
<td>2.6132</td>
<td>2.44155</td>
</tr>
<tr>
<td>pat1276</td>
<td>3.7648</td>
<td>1.9074</td>
<td>3.7915</td>
<td>3.0979</td>
<td>3.1404</td>
</tr>
<tr>
<td>pat1288</td>
<td>3.4078</td>
<td>2.3473</td>
<td>4.7078</td>
<td>3.2267</td>
<td>3.4224</td>
</tr>
<tr>
<td>pat12880</td>
<td>7.5732</td>
<td>4.7586</td>
<td>8.7326</td>
<td>7.897</td>
<td>7.24035</td>
</tr>
<tr>
<td>pat13140</td>
<td>3.0679</td>
<td>2.2002</td>
<td>3.2347</td>
<td>3.3456</td>
<td>2.9621</td>
</tr>
<tr>
<td>pat13146</td>
<td>3.1361</td>
<td>2.0079</td>
<td>2.9086</td>
<td>3.0134</td>
<td>2.7665</td>
</tr>
<tr>
<td>pat13148</td>
<td>3.032</td>
<td>1.9967</td>
<td>2.9524</td>
<td>3.2541</td>
<td>2.8088</td>
</tr>
<tr>
<td>pat13158</td>
<td>3.6358</td>
<td>2.3152</td>
<td>3.148</td>
<td>2.4812</td>
<td>2.89505</td>
</tr>
<tr>
<td>pat13170</td>
<td>3.4413</td>
<td>2.4605</td>
<td>3.3098</td>
<td>2.1045</td>
<td>2.829025</td>
</tr>
<tr>
<td>pat13174</td>
<td>7.8464</td>
<td>5.1445</td>
<td>7.8744</td>
<td>5.3383</td>
<td>6.5509</td>
</tr>
<tr>
<td>pat13182</td>
<td>3.5516</td>
<td>2.1038</td>
<td>2.9658</td>
<td>2.5099</td>
<td>2.782775</td>
</tr>
<tr>
<td>pat13184</td>
<td>3.5653</td>
<td>2.0948</td>
<td>2.9171</td>
<td>2.3795</td>
<td>2.739175</td>
</tr>
<tr>
<td>pat13188</td>
<td>3.3345</td>
<td>2.475</td>
<td>3.4901</td>
<td>2.483</td>
<td>2.94565</td>
</tr>
<tr>
<td>pat13192</td>
<td>7.8764</td>
<td>5.2529</td>
<td>7.9102</td>
<td>5.2186</td>
<td>6.564525</td>
</tr>
<tr>
<td>pat13196</td>
<td>3.2878</td>
<td>2.4284</td>
<td>3.4394</td>
<td>1.98</td>
<td>2.7839</td>
</tr>
<tr>
<td>pat13198</td>
<td>7.7695</td>
<td>5.2079</td>
<td>7.8671</td>
<td>5.5274</td>
<td>6.529725</td>
</tr>
<tr>
<td>pat13206</td>
<td>4.2644</td>
<td>2.9201</td>
<td>3.9232</td>
<td>3.4144</td>
<td>3.630525</td>
</tr>
<tr>
<td>pat13210</td>
<td>10.7688</td>
<td>5.5876</td>
<td>9.2245</td>
<td>8.924</td>
<td>8.626225</td>
</tr>
<tr>
<td>pat13212</td>
<td>3.3381</td>
<td>2.0834</td>
<td>2.9937</td>
<td>3.0151</td>
<td>2.857575</td>
</tr>
<tr>
<td>pat13222</td>
<td>9.7177</td>
<td>7.1188</td>
<td>12.1816</td>
<td>10.147</td>
<td>9.791275</td>
</tr>
<tr>
<td>pat13224</td>
<td>2.9887</td>
<td>2.1052</td>
<td>3.0723</td>
<td>3.0139</td>
<td>2.795025</td>
</tr>
<tr>
<td>pat13228</td>
<td>6.8521</td>
<td>4.0454</td>
<td>7.604</td>
<td>7.0227</td>
<td>6.38105</td>
</tr>
<tr>
<td>pat13258</td>
<td>11.1808</td>
<td>7.0986</td>
<td>10.4839</td>
<td>6.11</td>
<td>8.718325</td>
</tr>
<tr>
<td>pat13260</td>
<td>3.4998</td>
<td>2.1963</td>
<td>3.1493</td>
<td>2.5141</td>
<td>2.839875</td>
</tr>
<tr>
<td>pat13264</td>
<td>8.0042</td>
<td>4.1647</td>
<td>6.8043</td>
<td>4.8121</td>
<td>5.946325</td>
</tr>
<tr>
<td>pat13276</td>
<td>7.2493</td>
<td>5.2505</td>
<td>8.9064</td>
<td>5.2937</td>
<td>6.674975</td>
</tr>
<tr>
<td>pat13286</td>
<td>4.3458</td>
<td>2.868</td>
<td>3.9823</td>
<td>3.6757</td>
<td>3.71795</td>
</tr>
<tr>
<td>pat13288</td>
<td>10.8444</td>
<td>5.7067</td>
<td>9.3505</td>
<td>8.9438</td>
<td>8.72135</td>
</tr>
<tr>
<td>pat13294</td>
<td>9.8104</td>
<td>7.2201</td>
<td>12.3638</td>
<td>10.1651</td>
<td>9.88985</td>
</tr>
<tr>
<td>pat13312</td>
<td>11.1185</td>
<td>6.9168</td>
<td>10.6</td>
<td>6.2161</td>
<td>8.71285</td>
</tr>
<tr>
<td>pat13371</td>
<td>3.9224</td>
<td>1.9031</td>
<td>3.1671</td>
<td>3.1904</td>
<td>3.04575</td>
</tr>
<tr>
<td>pat13374</td>
<td>4.0699</td>
<td>2.0324</td>
<td>3.4009</td>
<td>3.5704</td>
<td>3.2684</td>
</tr>
<tr>
<td>pat13386</td>
<td>3.7786</td>
<td>2.2597</td>
<td>3.9557</td>
<td>3.744</td>
<td>3.4345</td>
</tr>
<tr>
<td>pat13389</td>
<td>2.9411</td>
<td>1.5639</td>
<td>2.7241</td>
<td>2.9576</td>
<td>2.546675</td>
</tr>
<tr>
<td>pat13422</td>
<td>4.4283</td>
<td>2.3985</td>
<td>3.9178</td>
<td>2.4421</td>
<td>3.296675</td>
</tr>
<tr>
<td>pat13423</td>
<td>10.8526</td>
<td>4.4688</td>
<td>8.5637</td>
<td>5.8321</td>
<td>7.4293</td>
</tr>
<tr>
<td>pat13425</td>
<td>3.2926</td>
<td>1.5845</td>
<td>2.7699</td>
<td>2.5443</td>
<td>2.547825</td>
</tr>
<tr>
<td>pat13426</td>
<td>10.7027</td>
<td>4.3629</td>
<td>8.4299</td>
<td>6.0954</td>
<td>7.397725</td>
</tr>
<tr>
<td>pat13438</td>
<td>9.8699</td>
<td>5.635</td>
<td>11.187</td>
<td>6.7319</td>
<td>8.35595</td>
</tr>
<tr>
<td>pat13441</td>
<td>6.8727</td>
<td>3.106</td>
<td>7.1324</td>
<td>5.0535</td>
<td>5.5366</td>
</tr>
<tr>
<td>pat13528</td>
<td>13.0035</td>
<td>5.4377</td>
<td>12.8541</td>
<td>11.546</td>
<td>10.710325</td>
</tr>
<tr>
<td>pat13531</td>
<td>9.6654</td>
<td>2.9758</td>
<td>8.4663</td>
<td>8.6409</td>
<td>7.4371</td>
</tr>
<tr>
<td>pat13614</td>
<td>3.99</td>
<td>2.0747</td>
<td>3.1872</td>
<td>3.322</td>
<td>3.143475</td>
</tr>
<tr>
<td>pat13616</td>
<td>4.0145</td>
<td>1.8651</td>
<td>3.2231</td>
<td>3.4803</td>
<td>3.14575</td>
</tr>
<tr>
<td>pat13620</td>
<td>3.5676</td>
<td>2.4582</td>
<td>4.0351</td>
<td>3.7714</td>
<td>3.458075</td>
</tr>
<tr>
<td>pat13624</td>
<td>9.6723</td>
<td>4.2041</td>
<td>9.9011</td>
<td>10.0964</td>
<td>8.468475</td>
</tr>
<tr>
<td>pat13628</td>
<td>3.639</td>
<td>2.2701</td>
<td>4.0952</td>
<td>4.0082</td>
<td>3.503125</td>
</tr>
<tr>
<td>pat13638</td>
<td>4.3352</td>
<td>2.571</td>
<td>3.8156</td>
<td>1.9789</td>
<td>3.175175</td>
</tr>
<tr>
<td>pat13642</td>
<td>10.5105</td>
<td>4.1575</td>
<td>8.2166</td>
<td>6.0561</td>
<td>7.235175</td>
</tr>
<tr>
<td>pat13654</td>
<td>9.9396</td>
<td>5.7567</td>
<td>11.2542</td>
<td>6.7201</td>
<td>8.41765</td>
</tr>
<tr>
<td>pat13664</td>
<td>4.4949</td>
<td>2.4397</td>
<td>3.8947</td>
<td>2.1908</td>
<td>3.255025</td>
</tr>
<tr>
<td>pat13666</td>
<td>10.7243</td>
<td>4.2903</td>
<td>8.4026</td>
<td>5.894</td>
<td>7.3278</td>
</tr>
<tr>
<td>pat13672</td>
<td>9.9028</td>
<td>5.7276</td>
<td>11.3366</td>
<td>6.6091</td>
<td>8.394025</td>
</tr>
<tr>
<td>pat13690</td>
<td>12.8876</td>
<td>5.4194</td>
<td>12.7288</td>
<td>11.4747</td>
<td>10.627625</td>
</tr>
<tr>
<td>pat13692</td>
<td>3.8737</td>
<td>2.1691</td>
<td>3.5267</td>
<td>3.3098</td>
<td>3.219825</td>
</tr>
<tr>
<td>pat13696</td>
<td>9.4261</td>
<td>2.8875</td>
<td>8.3214</td>
<td>8.4901</td>
<td>7.281275</td>
</tr>
<tr>
<td>pat13708</td>
<td>8.4951</td>
<td>4.0538</td>
<td>11.1896</td>
<td>9.7961</td>
<td>8.38365</td>
</tr>
<tr>
<td>pat13744</td>
<td>10.2151</td>
<td>4.5178</td>
<td>9.7463</td>
<td>5.9337</td>
<td>7.603225</td>
</tr>
<tr>
<td>pat1380</td>
<td>2.895</td>
<td>1.9861</td>
<td>2.3461</td>
<td>2.5888</td>
<td>2.454</td>
</tr>
<tr>
<td>pat1384</td>
<td>3.4386</td>
<td>1.5707</td>
<td>3.6481</td>
<td>3.3058</td>
<td>2.9908</td>
</tr>
<tr>
<td>pat13860</td>
<td>2.7584</td>
<td>1.7042</td>
<td>2.4495</td>
<td>2.577</td>
<td>2.372275</td>
</tr>
<tr>
<td>pat13872</td>
<td>2.7687</td>
<td>1.7123</td>
<td>2.4298</td>
<td>2.5715</td>
<td>2.370575</td>
</tr>
<tr>
<td>pat13876</td>
<td>3.5536</td>
<td>2.7636</td>
<td>4.4413</td>
<td>3.8938</td>
<td>3.663075</td>
</tr>
<tr>
<td>pat13908</td>
<td>2.7665</td>
<td>1.6437</td>
<td>2.381</td>
<td>2.5439</td>
<td>2.338775</td>
</tr>
<tr>
<td>pat13912</td>
<td>4.207</td>
<td>2.8181</td>
<td>4.1255</td>
<td>3.1873</td>
<td>3.584475</td>
</tr>
<tr>
<td>pat13924</td>
<td>3.8802</td>
<td>3.4845</td>
<td>5.1973</td>
<td>3.4061</td>
<td>3.992025</td>
</tr>
<tr>
<td>pat1396</td>
<td>2.9047</td>
<td>1.7777</td>
<td>4.343</td>
<td>3.6777</td>
<td>3.175775</td>
</tr>
<tr>
<td>pat14014</td>
<td>5.6467</td>
<td>4.2153</td>
<td>6.9418</td>
<td>5.6563</td>
<td>5.615025</td>
</tr>
<tr>
<td>pat14340</td>
<td>2.7893</td>
<td>1.7842</td>
<td>2.482</td>
<td>2.4706</td>
<td>2.381525</td>
</tr>
<tr>
<td>pat14344</td>
<td>5.3071</td>
<td>2.5937</td>
<td>5.4657</td>
<td>5.2158</td>
<td>4.645575</td>
</tr>
<tr>
<td>pat14356</td>
<td>4.5016</td>
<td>3.2952</td>
<td>7.0793</td>
<td>5.9846</td>
<td>5.215175</td>
</tr>
<tr>
<td>pat14392</td>
<td>5.6981</td>
<td>3.4945</td>
<td>6.404</td>
<td>4.2016</td>
<td>4.94955</td>
</tr>
<tr>
<td>pat144</td>
<td>3.3682</td>
<td>2.1748</td>
<td>3.3162</td>
<td>3.8459</td>
<td>3.176275</td>
</tr>
<tr>
<td>pat14586</td>
<td>4.4204</td>
<td>1.8924</td>
<td>2.9661</td>
<td>3.7064</td>
<td>3.246325</td>
</tr>
<tr>
<td>pat14588</td>
<td>2.7148</td>
<td>1.5782</td>
<td>2.3195</td>
<td>2.4164</td>
<td>2.257225</td>
</tr>
<tr>
<td>pat14592</td>
<td>4.4171</td>
<td>2.1129</td>
<td>3.1842</td>
<td>4.0388</td>
<td>3.43825</td>
</tr>
<tr>
<td>pat14596</td>
<td>3.5626</td>
<td>2.8762</td>
<td>4.5955</td>
<td>3.9041</td>
<td>3.7346</td>
</tr>
<tr>
<td>pat14600</td>
<td>2.8517</td>
<td>1.7216</td>
<td>2.5498</td>
<td>2.8669</td>
<td>2.4975</td>
</tr>
<tr>
<td>pat14602</td>
<td>3.7035</td>
<td>2.8313</td>
<td>4.5282</td>
<td>3.8943</td>
<td>3.739325</td>
</tr>
<tr>
<td>pat14610</td>
<td>4.6162</td>
<td>2.0794</td>
<td>3.1079</td>
<td>3.6964</td>
<td>3.374975</td>
</tr>
<tr>
<td>pat14614</td>
<td>4.1713</td>
<td>2.85</td>
<td>4.203</td>
<td>3.1424</td>
<td>3.591675</td>
</tr>
<tr>
<td>pat14626</td>
<td>4.0397</td>
<td>3.589</td>
<td>5.1995</td>
<td>3.5506</td>
<td>4.0947</td>
</tr>
<tr>
<td>pat14636</td>
<td>2.7851</td>
<td>1.6579</td>
<td>2.4118</td>
<td>2.7922</td>
<td>2.41175</td>
</tr>
</tbody>
</table>

195
<p>| pat14638 | 4.1459 | 2.7905 | 4.1197 | 3.4289 | 3.62125 |
| pat14644 | 3.8575 | 3.4842 | 5.1961 | 3.8367 | 4.093625 |
| pat14662 | 5.697 | 4.4219 | 7.1305 | 5.855 | 5.7761 |
| pat14664 | 4.5 | 1.9565 | 3.0077 | 3.7962 | 3.3151 |
| pat14668 | 3.8351 | 2.4603 | 4.0912 | 3.5295 | 3.479025 |
| pat14680 | 3.4756 | 2.9271 | 4.8757 | 3.9364 | 3.8037 |
| pat14716 | 4.0355 | 2.9509 | 4.4853 | 3.4887 | 3.7401 |
| pat1476 | 7.7857 | 3.9016 | 5.3856 | 5.9908 | 5.765925 |
| pat1482 | 7.321 | 2.4913 | 3.4371 | 4.8747 | 4.531025 |
| pat14826 | 4.6476 | 1.9679 | 3.3178 | 4.0419 | 3.4938 |
| pat14827 | 5.1274 | 2.5937 | 5.5471 | 5.2055 | 4.618425 |
| pat14829 | 4.4903 | 1.8129 | 2.9341 | 3.8047 | 3.2605 |
| pat14830 | 5.24 | 2.5646 | 5.5353 | 5.1902 | 4.632525 |
| pat1484 | 3.2733 | 1.772 | 2.3737 | 2.4658 | 2.4712 |
| pat14842 | 4.6184 | 3.3412 | 7.0699 | 6.0626 | 5.273025 |
| pat14845 | 3.3082 | 1.869 | 4.0488 | 3.643 | 3.21725 |
| pat14878 | 5.6795 | 3.4485 | 6.4335 | 4.6191 | 5.04515 |
| pat14881 | 3.6353 | 1.9002 | 3.6448 | 3.2431 | 3.10585 |
| pat1494 | 7.8185 | 3.9373 | 5.4363 | 6.15 | 5.835525 |
| pat1506 | 7.5275 | 2.6841 | 3.4782 | 4.8754 | 4.6413 |
| pat15068 | 2.8165 | 1.6504 | 2.6726 | 2.7714 | 2.477725 |
| pat15070 | 5.1679 | 2.6423 | 5.0037 | 5.1488 | 4.615675 |
| pat15076 | 4.4466 | 3.5246 | 7.2301 | 6.0746 | 5.318975 |
| pat15094 | 5.6303 | 3.5805 | 6.4917 | 4.3614 | 5.015975 |
| pat1510 | 3.5563 | 2.439 | 3.5041 | 2.6548 | 3.03855 |
| pat15148 | 4.7662 | 2.6916 | 6.1155 | 5.2366 | 4.702475 |
| pat1518 | 6.5779 | 2.2539 | 3.6932 | 7.4933 | 5.004575 |
| pat1520 | 2.8094 | 1.5625 | 2.2916 | 3.2481 | 2.4779 |
| pat1524 | 6.6221 | 2.3717 | 3.731 | 7.5351 | 5.069475 |
| pat1528 | 3.1348 | 2.3442 | 3.5904 | 3.4467 | 3.108775 |
| pat15318 | 4.3152 | 2.8171 | 3.4336 | 3.5379 | 3.52595 |
| pat1532 | 3.0335 | 1.5829 | 2.2393 | 2.8383 | 2.4235 |
| pat15330 | 4.1117 | 3.3071 | 4.0497 | 3.7546 | 3.805775 |
| pat1534 | 3.3829 | 2.2044 | 3.305 | 3.0736 | 2.991475 |
| pat15364 | 14.8447 | 10.118 | 12.0764 | 7.08 | 11.029775 |
| pat15366 | 4.7182 | 3.2928 | 3.9784 | 2.4591 | 3.612125 |
| pat15370 | 11.2386 | 6.484 | 8.2083 | 5.8608 | 7.947925 |
| pat15382 | 10.6562 | 8.3974 | 11.1358 | 6.5852 | 9.19365 |
| pat15472 | 13.8221 | 8.6782 | 12.5881 | 11.2462 | 11.58365 |
| pat1548 | 7.7495 | 3.894 | 5.333 | 5.9286 | 5.726275 |
| pat156 | 3.3451 | 1.9258 | 2.8641 | 2.9661 | 2.775275 |
| pat1560 | 7.4005 | 2.5922 | 3.4223 | 4.9527 | 4.591925 |
| pat1564 | 3.4057 | 2.1539 | 3.3805 | 2.7468 | 2.921725 |
| pat15798 | 5.7397 | 3.4449 | 4.8305 | 4.3351 | 4.58755 |
| pat15802 | 13.586 | 5.4637 | 10.005 | 11.0663 | 10.03025 |
| pat15814 | 12.2319 | 7.1733 | 13.5804 | 12.5588 | 11.3861 |
| pat15850 | 13.7729 | 7.0554 | 11.4195 | 7.1069 | 9.838675 |
| pat1596 | 6.6825 | 2.3856 | 3.7351 | 7.6333 | 5.109125 |
| pat160 | 5.0558 | 3.0601 | 5.4449 | 4.0029 | 4.390925 |
| pat1600 | 3.294 | 2.0922 | 3.1119 | 3.2926 | 2.947675 |
| pat1612 | 3.3136 | 2.2865 | 3.4058 | 3.1196 | 3.031375 |
| pat1626 | 7.7213 | 3.5355 | 3.7838 | 5.6136 | 5.16355 |
| pat1628 | 3.174 | 1.9262 | 2.2535 | 2.1597 | 2.37835 |
| pat1632 | 7.6988 | 3.5521 | 3.7957 | 5.5824 | 5.15725 |
| pat1636 | 3.2771 | 2.3086 | 3.477 | 2.7524 | 2.953775 |
| pat1650 | 8.0164 | 3.5702 | 3.8432 | 5.7376 | 5.29185 |
| pat1654 | 3.6276 | 2.3724 | 3.1958 | 2.4612 | 2.91425 |
| pat16768 | 8.5875 | 7.2245 | 9.5565 | 8.272 | 8.410125 |
| pat16770 | 4.6742 | 2.327 | 3.2805 | 3.9651 | 3.5617 |
| pat16774 | 5.7704 | 4.0571 | 5.6398 | 5.1906 | 5.164475 |
| pat16786 | 5.2585 | 5.0983 | 7.2933 | 6.2203 | 5.9676 |
| pat16822 | 6.1913 | 5.0226 | 6.5003 | 4.608 | 5.58055 |
| pat1707 | 4.3434 | 1.7685 | 2.8668 | 3.6072 | 3.146475 |
| pat1710 | 7.7248 | 3.8234 | 5.4584 | 6.1043 | 5.777725 |
| pat1722 | 7.1719 | 2.4855 | 3.5616 | 5.0241 | 4.560775 |
| pat1723 | 3.3585 | 1.878 | 4.2009 | 3.7987 | 3.309025 |
| pat1725 | 7.3785 | 2.4021 | 3.3928 | 4.8423 | 4.503925 |
| pat17254 | 7.5823 | 4.7771 | 8.7033 | 7.7167 | 7.19485 |
| pat1726 | 3.7788 | 2.0577 | 4.2406 | 3.6301 | 3.4268 |
| pat17504 | 4.2592 | 2.5776 | 3.1894 | 3.437 | 3.3658 |
| pat17524 | 14.7181 | 10.0056 | 11.984 | 7.1173 | 10.95625 |
| pat17530 | 11.2874 | 6.6633 | 8.368 | 6.0156 | 8.083575 |
| pat17578 | 13.9013 | 8.5135 | 12.6175 | 11.2769 | 11.5773 |
| pat1758 | 6.7627 | 2.3142 | 3.8511 | 7.4952 | 5.1058 |
| pat17584 | 10.2806 | 5.1706 | 8.3044 | 8.4595 | 8.053775 |
| pat1759 | 3.7124 | 1.9081 | 3.7548 | 3.2999 | 3.1688 |
| pat1761 | 6.5825 | 2.1666 | 3.6467 | 7.664 | 5.01495 |
| pat1762 | 3.9354 | 2.0855 | 3.8675 | 3.7236 | 3.403 |
| pat1774 | 3.9252 | 2.4592 | 4.6923 | 3.8492 | 3.731475 |
| pat17743 | 13.8006 | 5.5444 | 10.0936 | 11.0438 | 10.1206 |
| pat17746 | 13.9178 | 5.383 | 10.069 | 11.2166 | 10.1466 |
| pat1777 | 3.2106 | 1.6474 | 3.0364 | 3.0407 | 2.733775 |
| pat180 | 3.617 | 2.7418 | 3.5357 | 3.8488 | 3.435825 |
| pat18226 | 8.8087 | 7.3286 | 9.6398 | 8.2347 | 8.50295 |
| pat18232 | 5.8855 | 4.0806 | 5.6406 | 5.3084 | 5.228775 |
| pat1866 | 7.6481 | 3.3689 | 3.8244 | 5.6034 | 5.1112 |
| pat1867 | 3.449 | 1.5951 | 3.6053 | 3.257 | 2.9766 |
| pat1869 | 7.781 | 3.467 | 3.7533 | 5.4846 | 5.121475 |
| pat1870 | 3.9911 | 1.9487 | 3.8256 | 3.306 | 3.26785 |
| pat188 | 2.6922 | 1.6393 | 2.5854 | 2.849 | 2.441475 |
| pat1950 | 5.1283 | 3.0897 | 3.297 | 3.9658 | 3.8702 |
| pat1952 | 2.7798 | 1.6913 | 2.3417 | 2.3873 | 2.300025 |
| pat1956 | 5.0525 | 3.1236 | 3.441 | 4.1941 | 3.9528 |
| pat1964 | 2.8736 | 1.6439 | 2.478 | 2.748 | 2.435875 |
| pat1966 | 3.4003 | 1.998 | 4.142 | 3.7032 | 3.310875 |
| pat19737 | 3.4404 | 1.9702 | 3.068 | 3.3447 | 2.955825 |
| pat1974 | 5.189 | 3.124 | 3.3606 | 3.8971 | 3.892675 |
| pat19755 | 3.014 | 1.9352 | 3.1299 | 3.9569 | 3.009 |
| pat19761 | 3.1596 | 1.5916 | 2.8045 | 3.1685 | 2.68105 |
| pat19763 | 3.2908 | 1.5734 | 5.2952 | 4.0533 | 3.553175 |
| pat1978 | 3.8286 | 2.2148 | 3.8444 | 2.9674 | 3.2138 |
| pat198 | 3.9455 | 2.8758 | 3.4996 | 3.4924 | 3.453325 |
| pat19851 | 2.7343 | 1.7562 | 2.6074 | 2.5003 | 2.39955 |
| pat19853 | 3.5025 | 1.6941 | 5.2807 | 4.125 | 3.650575 |
| pat19854 | 3.7547 | 2.7352 | 3.3421 | 3.6411 | 3.368275 |
| pat19857 | 2.7368 | 1.7895 | 2.4003 | 2.7511 | 2.419425 |
| pat19861 | 4.1532 | 2.4608 | 3.1396 | 3.9063 | 3.414975 |
| pat19863 | 3.514 | 2.5236 | 3.3337 | 3.5783 | 3.2374 |
| pat19865 | 3.2764 | 1.6996 | 5.2991 | 4.3601 | 3.6588 |
| pat19867 | 4.078 | 2.5801 | 3.4868 | 3.9586 | 3.525875 |
| pat19870 | 4.2538 | 2.5806 | 3.1328 | 3.914 | 3.4703 |
| pat19871 | 3.2569 | 1.593 | 5.1875 | 4.0356 | 3.51825 |
| pat19875 | 2.8533 | 1.7975 | 2.5454 | 2.8699 | 2.516525 |
| pat19879 | 4.51 | 2.5213 | 3.0632 | 3.4737 | 3.39205 |
| pat19881 | 3.486 | 2.5672 | 3.4963 | 3.8442 | 3.348425 |
| pat19891 | 4.3553 | 3.1226 | 3.8494 | 3.8606 | 3.796975 |
| pat19897 | 3.1485 | 1.9646 | 2.7267 | 2.8878 | 2.6819 |
| pat1990 | 3.7002 | 2.6012 | 4.7942 | 3.4 | 3.6289 |
| pat19901 | 3.533 | 1.7534 | 5.3432 | 4.0339 | 3.665875 |
| pat19903 | 4.4338 | 2.5659 | 3.2486 | 3.0197 | 3.317 |
| pat19906 | 4.629 | 2.6413 | 2.9642 | 3.4483 | 3.4207 |
| pat19907 | 3.3334 | 1.6324 | 5.3102 | 4.3328 | 3.6522 |
| pat19909 | 4.5287 | 3.1761 | 3.7831 | 3.4587 | 3.73665 |
| pat19915 | 3.2223 | 2.0171 | 2.6366 | 2.891 | 2.69175 |
| pat19953 | 3.5127 | 2.068 | 2.9033 | 3.2422 | 2.93155 |
| pat19989 | 3.1359 | 1.9791 | 3.1578 | 3.9645 | 3.059325 |
| pat2000 | 2.7494 | 1.5809 | 2.3485 | 2.7551 | 2.358475 |
| pat20001 | 3.3027 | 1.6325 | 2.4877 | 3.1275 | 2.6376 |
| pat20005 | 3.9582 | 2.0564 | 2.806 | 3.2325 | 3.013275 |
| pat2002 | 3.7566 | 2.0681 | 3.809 | 3.2775 | 3.2278 |
| pat2008 | 3.4798 | 2.58 | 4.8008 | 3.7349 | 3.648875 |
| pat20091 | 3.0745 | 1.7693 | 2.4023 | 2.8532 | 2.524825 |
| pat20095 | 5.7881 | 2.0884 | 3.3094 | 4.8971 | 4.02075 |
| pat20097 | 3.6756 | 2.6157 | 3.299 | 3.5675 | 3.28945 |
| pat20107 | 5.6346 | 2.8365 | 4.6028 | 5.971 | 4.761225 |
| pat20113 | 3.6505 | 1.6807 | 2.7581 | 3.468 | 2.889325 |
| pat20143 | 6.421 | 3.0189 | 4.1575 | 4.145 | 4.4356 |
| pat20149 | 4.0624 | 1.8457 | 2.7192 | 3.2596 | 2.971725 |
| pat20187 | 2.87 | 2.1091 | 2.6386 | 2.851 | 2.617175 |
| pat20193 | 2.7219 | 1.7742 | 2.5678 | 2.5034 | 2.391825 |
| pat20195 | 3.288 | 1.5084 | 5.2101 | 4.2937 | 3.57505 |
| pat20205 | 2.8484 | 2.1592 | 2.7808 | 2.9538 | 2.68555 |
| pat20217 | 2.8526 | 1.7946 | 2.4778 | 2.6599 | 2.446225 |
| pat20221 | 3.8834 | 1.9666 | 2.9525 | 3.5562 | 3.089675 |
| pat20229 | 2.802 | 1.9633 | 2.7425 | 2.7744 | 2.57055 |
| pat20231 | 3.5075 | 1.6372 | 5.3038 | 4.1146 | 3.640775 |
| pat20235 | 2.8149 | 1.8327 | 2.4469 | 2.6109 | 2.42635 |
| pat20239 | 3.9882 | 1.9916 | 2.8418 | 3.2106 | 3.00805 |
| pat20243 | 3.2982 | 1.5683 | 5.2024 | 3.769 | 3.459475 |
| pat20245 | 3.9306 | 2.0432 | 3.1085 | 3.0729 | 3.0388 |
| pat2028 | 5.0585 | 3.0412 | 3.2501 | 4.0259 | 3.843925 |
| pat2032 | 3.5285 | 1.9209 | 3.7468 | 3.265 | 3.1153 |
| pat20333 | 3.7056 | 1.6251 | 5.243 | 4.3401 | 3.72845 |
| pat20334 | 3.277 | 2.2324 | 2.5236 | 2.9254 | 2.7396 |
| pat20335 | 5.7396 | 2.0843 | 3.6039 | 4.9369 | 4.091175 |
| pat20337 | 2.7986 | 2.0297 | 2.7038 | 2.5219 | 2.5135 |
| pat20338 | 5.6971 | 2.1159 | 3.2748 | 4.8519 | 3.984925 |
| pat20339 | 3.4444 | 1.5987 | 5.2058 | 4.0845 | 3.58335 |
| pat20341 | 5.6814 | 2.9089 | 4.7275 | 6.009 | 4.8317 |
| pat20343 | 2.8236 | 1.9957 | 2.4378 | 2.7844 | 2.510375 |
| pat20347 | 3.5858 | 1.5682 | 2.752 | 3.4678 | 2.843675 |
| pat20350 | 5.5958 | 2.8684 | 4.5765 | 5.933 | 4.7585 |
| pat20351 | 3.1862 | 1.4773 | 5.2091 | 4.2713 | 3.535975 |
| pat20353 | 3.5066 | 1.5638 | 3.0551 | 3.5707 | 2.92405 |
| pat20359 | 6.3868 | 2.9166 | 4.3174 | 4.5213 | 4.535525 |
| pat20361 | 2.8795 | 2.0763 | 2.5934 | 2.9368 | 2.6215 |
| pat20365 | 4.0097 | 1.7501 | 2.8399 | 3.3782 | 2.994475 |
| pat20377 | 3.8037 | 1.975 | 3.4254 | 3.651 | 3.213775 |
| pat20439 | 3.7374 | 1.8176 | 3.51 | 2.6603 | 2.931325 |
| pat2044 | 3.181 | 2.0186 | 4.1249 | 3.6392 | 3.312925 |
| pat20475 | 3.262 | 1.7966 | 3.7426 | 3.6859 | 3.121775 |
| pat20487 | 3.5221 | 1.4425 | 3.1198 | 2.7828 | 2.7168 |
| pat20491 | 2.8507 | 1.535 | 2.9443 | 1.9373 | 2.316825 |
| pat20577 | 3.0183 | 1.5157 | 2.975 | 2.0917 | 2.400175 |
| pat20581 | 3.0359 | 1.5904 | 3.0081 | 1.9476 | 2.3955 |
| pat20583 | 3.8348 | 2.3953 | 3.8922 | 3.0619 | 3.29605 |
| pat20593 | 2.9684 | 1.8656 | 3.2813 | 2.3749 | 2.62255 |
| pat20599 | 2.919 | 1.5655 | 2.9527 | 1.796 | 2.3083 |
| pat20629 | 3.0745 | 1.7654 | 3.0874 | 2.1374 | 2.516175 |
| pat20635 | 2.7757 | 1.4834 | 2.8617 | 2.1156 | 2.3091 |
| pat2080 | 3.6514 | 2.104 | 4.0861 | 3.4001 | 3.3104 |
| pat20907 | 3.0676 | 1.982 | 3.2952 | 2.2838 | 2.65715 |
| pat20919 | 3.109 | 1.6255 | 2.9538 | 2.1165 | 2.4512 |
| pat20923 | 2.9173 | 1.4368 | 2.9906 | 2.2123 | 2.38925 |
| pat20955 | 3.0361 | 1.6637 | 3.0487 | 2.4777 | 2.55655 |
| pat20959 | 2.9098 | 1.4441 | 2.8663 | 2.0476 | 2.31695 |
| pat20971 | 3.0053 | 1.6465 | 3.1456 | 2.099 | 2.4741 |
| pat21061 | 3.4035 | 1.6582 | 3.3319 | 2.652 | 2.7614 |
| pat21063 | 3.1854 | 1.9178 | 3.0701 | 2.1701 | 2.58585 |
| pat21067 | 2.9601 | 1.4188 | 2.9301 | 1.8381 | 2.286775 |
| pat21079 | 2.8109 | 1.4617 | 3.1116 | 2.2344 | 2.40465 |
| pat2108 | 2.838 | 1.6787 | 2.2614 | 2.4506 | 2.307175 |
| pat2110 | 3.5863 | 1.7928 | 3.6921 | 3.2914 | 3.09065 |
| pat21159 | 7.7339 | 3.8258 | 5.6988 | 5.8391 | 5.7744 |
| pat2116 | 3.0998 | 2.1521 | 4.4376 | 3.7021 | 3.3479 |
| pat21165 | 7.2402 | 2.5407 | 3.8379 | 4.6266 | 4.56135 |
| pat21167 | 3.8731 | 1.6914 | 5.2717 | 3.7965 | 3.658175 |
| pat21177 | 7.5469 | 3.7367 | 5.62 | 6.3888 | 5.8231 |
| pat21189 | 7.2909 | 2.33 | 3.5229 | 5.2038 | 4.5869 |
| pat21193 | 3.03 | 1.7989 | 2.4504 | 2.405 | 2.421075 |
| pat21201 | 6.4623 | 2.1353 | 4.1147 | 7.6029 | 5.0788 |
| pat21203 | 3.4664 | 1.6025 | 5.3473 | 4.8297 | 3.811475 |
| pat21207 | 6.5718 | 2.1532 | 3.8591 | 7.5235 | 5.0269 |
| pat2111 | 2.7546 | 1.7269 | 2.3485 | 2.8945 | 2.431125 |
| pat21215 | 3.7012 | 1.5328 | 4.9912 | 4.3483 | 3.643375 |
| pat21217 | 2.845 | 1.6922 | 2.3986 | 2.4949 | 2.357675 |
| pat21305 | 3.3798 | 1.5949 | 5.1055 | 3.8766 | 3.4892 |
| pat21306 | 7.6557 | 3.4065 | 3.8294 | 5.4682 | 5.08995 |
| pat21307 | 2.7148 | 1.7641 | 2.5422 | 2.1156 | 2.284175 |
| pat21309 | 7.6234 | 3.3853 | 4.1933 | 5.2732 | 5.1188 |
| pat21310 | 3.3078 | 2.1037 | 2.4685 | 2.0816 | 2.4904 |
| pat21311 | 3.8148 | 1.8843 | 5.3189 | 3.6814 | 3.67485 |
| pat21313 | 2.796 | 1.9624 | 2.5091 | 2.4651 | 2.43315 |
| pat21315 | 7.8944 | 3.4811 | 3.9795 | 5.485 | 5.21 |
| pat21319 | 3.0716 | 1.9974 | 2.3948 | 1.8735 | 2.334325 |
| pat21331 | 2.8027 | 1.8589 | 2.4855 | 2.5976 | 2.436175 |
| pat21333 | 7.6513 | 3.324 | 3.9966 | 6.0353 | 5.2518 |
| pat21337 | 2.9048 | 1.8851 | 2.3916 | 2.2096 | 2.347775 |
| pat2134 | 3.7398 | 2.2198 | 4.0662 | 3.0659 | 3.272925 |
| pat21393 | 7.7337 | 3.8189 | 5.4904 | 5.9196 | 5.74065 |
| pat214 | 5.0513 | 3.0394 | 5.4181 | 3.8662 | 4.34375 |
| pat21405 | 7.1472 | 2.4773 | 3.4148 | 4.8221 | 4.46535 |
| pat21409 | 3.2236 | 1.8104 | 2.4061 | 2.3051 | 2.4363 |
| pat21441 | 6.6544 | 2.2425 | 3.8096 | 7.6135 | 5.08 |
| pat21445 | 2.7608 | 1.6547 | 2.2893 | 2.8874 | 2.39805 |
| pat21457 | 3.0985 | 1.782 | 2.3357 | 2.8378 | 2.5135 |
| pat21547 | 3.2485 | 1.8608 | 2.6459 | 2.897 | 2.66305 |
| pat21549 | 7.6652 | 3.4727 | 3.8223 | 5.5118 | 5.118 |
| pat21553 | 3.1333 | 1.9204 | 2.3376 | 1.9653 | 2.33915 |
| pat21633 | 4.9638 | 3.1567 | 3.6415 | 3.8385 | 3.900125 |
| pat21635 | 3.3581 | 1.7539 | 5.2599 | 3.7421 | 3.5285 |
| pat21639 | 5.1926 | 3.1425 | 3.365 | 3.9773 | 3.91935 |
| pat21643 | 2.7367 | 1.7689 | 2.395 | 2.2316 | 2.28305 |
| pat21647 | 3.5885 | 1.6808 | 5.2479 | 4.0688 | 3.6465 |
| pat21649 | 2.7632 | 1.7214 | 2.5754 | 2.3231 | 2.345775 |
| pat21657 | 5.0204 | 3.0095 | 3.4096 | 4.3394 | 3.944725 |
| pat21661 | 2.6035 | 1.6982 | 2.3898 | 2.451 | 2.285625 |
| pat21673 | 2.8457 | 1.8245 | 2.5647 | 2.7127 | 2.4869 |
| pat21683 | 3.3458 | 1.6027 | 5.1786 | 4.3895 | 3.62915 |
| pat21685 | 2.5122 | 1.6097 | 2.4298 | 2.4386 | 2.247575 |
| pat21691 | 2.7613 | 1.8198 | 2.4205 | 2.7409 | 2.435625 |
| pat21790 | 3.2893 | 1.9774 | 2.646 | 2.8115 | 2.68105 |
| pat21791 | 3.4169 | 1.7158 | 5.1223 | 3.7778 | 3.5082 |
| pat21793 | 2.6609 | 1.7026 | 2.4717 | 1.9685 | 2.200925 |
| pat21799 | 2.7443 | 1.7577 | 2.3734 | 2.389 | 2.3161 |
| pat21817 | 2.6792 | 1.6953 | 2.3458 | 2.4682 | 2.297125 |
| pat2214 | 3.9309 | 2.4728 | 3.3504 | 3.5694 | 3.330875 |
| pat224 | 2.9329 | 1.7023 | 2.5299 | 2.654 | 2.454775 |
| pat2250 | 3.9354 | 2.5882 | 3.541 | 3.7126 | 3.4443 |
| pat2262 | 3.9229 | 2.5474 | 3.3632 | 2.8731 | 3.17665 |
| pat2266 | 7.657 | 5.1012 | 7.2575 | 5.2275 | 6.3108 |
| pat232 | 4.9909 | 3.0288 | 5.5693 | 4.1856 | 4.44365 |
| pat2352 | 4.3573 | 2.9682 | 3.7405 | 3.5701 | 3.659025 |
| pat2356 | 10.4314 | 5.2709 | 8.6274 | 8.5455 | 8.2188 |
| pat2358 | 4.1721 | 2.9181 | 3.5681 | 3.8305 | 3.6222 |
| pat2374 | 6.7734 | 4.0124 | 7.1662 | 6.9159 | 6.216975 |
| pat24063 | 2.8676 | 1.8374 | 2.4528 | 2.4496 | 2.40185 |
| pat24065 | 3.7622 | 1.8195 | 5.0075 | 4.2264 | 3.7039 |
| pat24066 | 3.1549 | 2.0212 | 2.5755 | 3.0142 | 2.69145 |
| pat24069 | 2.8893 | 1.9774 | 2.3268 | 2.7917 | 2.4963 |
| pat24073 | 6.2125 | 4.0764 | 3.6119 | 5.5136 | 4.8536 |
| pat24075 | 2.7437 | 1.7595 | 2.582 | 2.8479 | 2.483275 |
| pat24077 | 3.6017 | 2.0701 | 5.2494 | 4.5761 | 3.874325 |
| pat24078 | 3.1198 | 2.0785 | 2.4116 | 2.9232 | 2.633275 |
| pat24079 | 6.0741 | 4.155 | 3.8654 | 5.0501 | 4.8999 |
| pat24081 | 2.6709 | 1.5908 | 2.5683 | 2.5133 | 2.335825 |
| pat24082 | 6.1965 | 4.1213 | 3.5823 | 5.5312 | 4.857825 |
| pat24083 | 3.3053 | 1.6241 | 5.2061 | 4.2509 | 3.5966 |
| pat24087 | 3.0847 | 1.997 | 2.4908 | 2.4576 | 2.507525 |
| pat24091 | 6.6414 | 3.9684 | 3.434 | 4.3282 | 4.593 |
| pat24093 | 2.7288 | 1.757 | 2.7736 | 2.9076 | 2.54175 |
| pat2410 | 7.5367 | 4.0765 | 6.2259 | 4.8597 | 5.6747 |
| pat24103 | 6.9399 | 5.417 | 4.8708 | 5.0106 | 5.559575 |</p>
<table>
<thead>
<tr>
<th>pat24105</th>
<th>2.776</th>
<th>1.6476</th>
<th>2.4661</th>
<th>2.6222</th>
<th>2.377975</th>
</tr>
</thead>
<tbody>
<tr>
<td>pat24109</td>
<td>4.0962</td>
<td>2.7259</td>
<td>2.9732</td>
<td>3.5599</td>
<td>3.3388</td>
</tr>
<tr>
<td>pat24113</td>
<td>3.9624</td>
<td>2.1101</td>
<td>5.2505</td>
<td>3.7783</td>
<td>3.775325</td>
</tr>
<tr>
<td>pat24114</td>
<td>3.2971</td>
<td>2.0518</td>
<td>2.5577</td>
<td>2.7067</td>
<td>2.653325</td>
</tr>
<tr>
<td>pat24115</td>
<td>6.5323</td>
<td>3.9642</td>
<td>3.6316</td>
<td>3.7664</td>
<td>4.473625</td>
</tr>
<tr>
<td>pat24117</td>
<td>2.6967</td>
<td>1.5919</td>
<td>2.7155</td>
<td>2.674</td>
<td>2.419525</td>
</tr>
<tr>
<td>pat24118</td>
<td>6.7029</td>
<td>4.0389</td>
<td>3.3069</td>
<td>4.0489</td>
<td>4.5244</td>
</tr>
<tr>
<td>pat24119</td>
<td>3.4641</td>
<td>1.6199</td>
<td>5.2234</td>
<td>4.128</td>
<td>3.60885</td>
</tr>
<tr>
<td>pat24121</td>
<td>6.9719</td>
<td>5.4565</td>
<td>4.7846</td>
<td>4.521</td>
<td>5.4335</td>
</tr>
<tr>
<td>pat24123</td>
<td>2.7778</td>
<td>1.6417</td>
<td>2.451</td>
<td>2.5996</td>
<td>2.367525</td>
</tr>
<tr>
<td>pat24127</td>
<td>4.2092</td>
<td>2.7024</td>
<td>2.8797</td>
<td>3.208</td>
<td>3.249825</td>
</tr>
<tr>
<td>pat24130</td>
<td>7.0081</td>
<td>5.4453</td>
<td>4.6774</td>
<td>4.6802</td>
<td>5.45275</td>
</tr>
<tr>
<td>pat24131</td>
<td>3.4349</td>
<td>1.7675</td>
<td>5.2786</td>
<td>3.8491</td>
<td>3.582525</td>
</tr>
<tr>
<td>pat24133</td>
<td>4.1169</td>
<td>2.8743</td>
<td>3.1228</td>
<td>3.0704</td>
<td>3.2961</td>
</tr>
<tr>
<td>pat24221</td>
<td>3.7567</td>
<td>1.9194</td>
<td>5.2285</td>
<td>4.2885</td>
<td>3.798275</td>
</tr>
<tr>
<td>pat24223</td>
<td>6.0268</td>
<td>3.3675</td>
<td>3.6392</td>
<td>4.817</td>
<td>4.462625</td>
</tr>
<tr>
<td>pat24229</td>
<td>6.2615</td>
<td>4.4463</td>
<td>4.738</td>
<td>6.0449</td>
<td>5.372675</td>
</tr>
<tr>
<td>pat24247</td>
<td>6.8311</td>
<td>4.3032</td>
<td>4.3875</td>
<td>4.5492</td>
<td>5.01775</td>
</tr>
<tr>
<td>pat24303</td>
<td>3.5684</td>
<td>2.0373</td>
<td>2.3781</td>
<td>3.0201</td>
<td>2.750975</td>
</tr>
<tr>
<td>pat24307</td>
<td>8.652</td>
<td>3.5465</td>
<td>3.589</td>
<td>6.8238</td>
<td>5.652825</td>
</tr>
<tr>
<td>pat24309</td>
<td>2.9957</td>
<td>1.7932</td>
<td>2.5654</td>
<td>2.9469</td>
<td>2.5753</td>
</tr>
<tr>
<td>pat24319</td>
<td>8.9699</td>
<td>5.2888</td>
<td>5.7278</td>
<td>8.8939</td>
<td>7.2201</td>
</tr>
<tr>
<td>pat24321</td>
<td>2.9616</td>
<td>1.6859</td>
<td>2.3262</td>
<td>2.8918</td>
<td>2.466375</td>
</tr>
<tr>
<td>pat24325</td>
<td>5.271</td>
<td>2.5427</td>
<td>3.1057</td>
<td>5.0868</td>
<td>4.00155</td>
</tr>
<tr>
<td>pat24355</td>
<td>9.4942</td>
<td>4.9669</td>
<td>4.9394</td>
<td>5.2559</td>
<td>6.1641</td>
</tr>
<tr>
<td>pat24357</td>
<td>3.0883</td>
<td>1.6836</td>
<td>2.4952</td>
<td>2.7142</td>
<td>2.495325</td>
</tr>
<tr>
<td>pat24361</td>
<td>5.7643</td>
<td>2.6292</td>
<td>2.9432</td>
<td>3.8173</td>
<td>3.7885</td>
</tr>
<tr>
<td>pat24373</td>
<td>5.9928</td>
<td>3.7686</td>
<td>4.1391</td>
<td>4.3967</td>
<td>4.5743</td>
</tr>
<tr>
<td>pat24463</td>
<td>8.87</td>
<td>3.9658</td>
<td>5.0217</td>
<td>7.4142</td>
<td>6.317925</td>
</tr>
<tr>
<td>pat24545</td>
<td>4.3108</td>
<td>2.1332</td>
<td>5.2149</td>
<td>4.6632</td>
<td>4.080525</td>
</tr>
<tr>
<td>pat24546</td>
<td>3.593</td>
<td>2.2564</td>
<td>2.4281</td>
<td>3.0441</td>
<td>2.8304</td>
</tr>
<tr>
<td>pat24547</td>
<td>8.5347</td>
<td>3.5412</td>
<td>3.9512</td>
<td>6.7296</td>
<td>5.689175</td>
</tr>
<tr>
<td>pat24549</td>
<td>2.7672</td>
<td>1.7374</td>
<td>2.5116</td>
<td>2.428</td>
<td>2.36105</td>
</tr>
<tr>
<td>pat24550</td>
<td>8.5416</td>
<td>3.5164</td>
<td>3.622</td>
<td>6.7776</td>
<td>5.6144</td>
</tr>
<tr>
<td>pat24551</td>
<td>3.6264</td>
<td>1.5556</td>
<td>5.0623</td>
<td>4.2661</td>
<td>3.6276</td>
</tr>
<tr>
<td>pat24553</td>
<td>9.079</td>
<td>5.2641</td>
<td>5.8468</td>
<td>8.8492</td>
<td>7.259775</td>
</tr>
<tr>
<td>pat24555</td>
<td>2.807</td>
<td>1.7634</td>
<td>2.3401</td>
<td>2.7703</td>
<td>2.4202</td>
</tr>
<tr>
<td>pat24559</td>
<td>5.3748</td>
<td>2.5075</td>
<td>3.166</td>
<td>5.1118</td>
<td>4.040025</td>
</tr>
<tr>
<td>pat24562</td>
<td>9.0103</td>
<td>5.2308</td>
<td>5.6983</td>
<td>8.8198</td>
<td>7.1898</td>
</tr>
<tr>
<td>pat24563</td>
<td>3.4827</td>
<td>1.6609</td>
<td>5.2812</td>
<td>4.5401</td>
<td>3.741225</td>
</tr>
<tr>
<td>pat24565</td>
<td>5.2386</td>
<td>2.495</td>
<td>3.4689</td>
<td>5.083</td>
<td>4.071375</td>
</tr>
<tr>
<td>pat24571</td>
<td>9.694</td>
<td>5.1284</td>
<td>5.1637</td>
<td>5.7236</td>
<td>6.427425</td>
</tr>
<tr>
<td>pat24573</td>
<td>2.9007</td>
<td>1.808</td>
<td>2.4646</td>
<td>2.4923</td>
<td>2.4164</td>
</tr>
<tr>
<td>pat24577</td>
<td>5.5462</td>
<td>2.4091</td>
<td>3.0144</td>
<td>4.03</td>
<td>3.749925</td>
</tr>
<tr>
<td>pat24589</td>
<td>5.9488</td>
<td>3.4479</td>
<td>4.2809</td>
<td>4.7005</td>
<td>4.594525</td>
</tr>
<tr>
<td>pat24598</td>
<td>9.4974</td>
<td>4.9122</td>
<td>4.9366</td>
<td>5.2195</td>
<td>6.141425</td>
</tr>
<tr>
<td>pat24599</td>
<td>3.7611</td>
<td>1.7823</td>
<td>5.3019</td>
<td>3.8774</td>
<td>3.680675</td>
</tr>
<tr>
<td>pat24601</td>
<td>5.8041</td>
<td>2.6976</td>
<td>3.2832</td>
<td>3.6039</td>
<td>3.8472</td>
</tr>
<tr>
<td>pat24607</td>
<td>6.1599</td>
<td>3.6266</td>
<td>4.3025</td>
<td>4.2965</td>
<td>4.596375</td>
</tr>
<tr>
<td>pat24789</td>
<td>2.8744</td>
<td>1.448</td>
<td>2.899</td>
<td>1.9223</td>
<td>2.285925</td>
</tr>
<tr>
<td>pat24793</td>
<td>3.5612</td>
<td>2.062</td>
<td>3.0871</td>
<td>2.4918</td>
<td>2.800525</td>
</tr>
<tr>
<td>pat24795</td>
<td>2.8997</td>
<td>1.5068</td>
<td>3.3149</td>
<td>2.2264</td>
<td>2.48695</td>
</tr>
<tr>
<td>pat24805</td>
<td>3.6031</td>
<td>2.6876</td>
<td>3.6313</td>
<td>3.137</td>
<td>3.26475</td>
</tr>
<tr>
<td>pat24807</td>
<td>3.0448</td>
<td>1.3793</td>
<td>2.9614</td>
<td>2.1107</td>
<td>2.37405</td>
</tr>
<tr>
<td>pat24811</td>
<td>2.9899</td>
<td>1.7307</td>
<td>2.9814</td>
<td>2.1915</td>
<td>2.473375</td>
</tr>
<tr>
<td>pat24841</td>
<td>3.8472</td>
<td>2.5231</td>
<td>3.4225</td>
<td>2.4728</td>
<td>3.0664</td>
</tr>
<tr>
<td>pat24843</td>
<td>2.9501</td>
<td>1.2959</td>
<td>3.1182</td>
<td>2.5366</td>
<td>2.4752</td>
</tr>
<tr>
<td>pat24847</td>
<td>2.9681</td>
<td>1.6239</td>
<td>2.9043</td>
<td>2.1073</td>
<td>2.4009</td>
</tr>
<tr>
<td>pat24859</td>
<td>3.1044</td>
<td>1.9932</td>
<td>3.1434</td>
<td>2.0814</td>
<td>2.5806</td>
</tr>
<tr>
<td>pat24949</td>
<td>3.6187</td>
<td>2.214</td>
<td>3.3905</td>
<td>2.6476</td>
<td>2.9677</td>
</tr>
<tr>
<td>pat25273</td>
<td>4.7546</td>
<td>2.6368</td>
<td>3.9309</td>
<td>3.8871</td>
<td>3.80235</td>
</tr>
<tr>
<td>pat25275</td>
<td>2.9104</td>
<td>1.5452</td>
<td>2.9688</td>
<td>1.994</td>
<td>2.3546</td>
</tr>
<tr>
<td>pat25279</td>
<td>3.2631</td>
<td>1.5104</td>
<td>2.9428</td>
<td>2.3099</td>
<td>2.50655</td>
</tr>
<tr>
<td>pat25291</td>
<td>3.3483</td>
<td>1.9765</td>
<td>3.4674</td>
<td>3.034</td>
<td>2.95655</td>
</tr>
<tr>
<td>pat25327</td>
<td>3.5521</td>
<td>1.8799</td>
<td>3.2602</td>
<td>2.3607</td>
<td>2.763225</td>
</tr>
<tr>
<td>pat25517</td>
<td>3.3129</td>
<td>1.575</td>
<td>5.1829</td>
<td>4.0087</td>
<td>3.519875</td>
</tr>
<tr>
<td>pat25518</td>
<td>4.7263</td>
<td>1.9174</td>
<td>3.2285</td>
<td>3.9363</td>
<td>3.452125</td>
</tr>
<tr>
<td>pat25519</td>
<td>3.163</td>
<td>2.2178</td>
<td>2.7561</td>
<td>2.6832</td>
<td>2.705025</td>
</tr>
<tr>
<td>pat25521</td>
<td>4.5436</td>
<td>1.8259</td>
<td>3.6008</td>
<td>3.7187</td>
<td>3.42225</td>
</tr>
<tr>
<td>pat25522</td>
<td>3.3038</td>
<td>2.213</td>
<td>2.5341</td>
<td>2.633</td>
<td>2.670975</td>
</tr>
<tr>
<td>pat25523</td>
<td>3.2671</td>
<td>1.5306</td>
<td>5.2429</td>
<td>3.7318</td>
<td>3.4431</td>
</tr>
<tr>
<td>pat25525</td>
<td>3.3061</td>
<td>2.7724</td>
<td>3.047</td>
<td>3.349</td>
<td>3.118625</td>
</tr>
<tr>
<td>pat25527</td>
<td>4.6592</td>
<td>1.7996</td>
<td>3.2963</td>
<td>3.8756</td>
<td>3.407675</td>
</tr>
<tr>
<td>pat25531</td>
<td>2.7785</td>
<td>1.8604</td>
<td>2.4196</td>
<td>2.2953</td>
<td>2.33845</td>
</tr>
<tr>
<td>pat25534</td>
<td>3.5516</td>
<td>2.8909</td>
<td>3.0358</td>
<td>3.4463</td>
<td>3.23115</td>
</tr>
<tr>
<td>pat25535</td>
<td>3.5067</td>
<td>1.5333</td>
<td>5.1875</td>
<td>4.0006</td>
<td>3.557025</td>
</tr>
<tr>
<td>pat25537</td>
<td>2.8068</td>
<td>1.8723</td>
<td>2.5931</td>
<td>2.3093</td>
<td>2.395375</td>
</tr>
<tr>
<td>pat25543</td>
<td>3.4632</td>
<td>2.5732</td>
<td>2.9472</td>
<td>3.0999</td>
<td>3.020875</td>
</tr>
<tr>
<td>pat25545</td>
<td>4.5434</td>
<td>1.7271</td>
<td>3.855</td>
<td>4.287</td>
<td>3.48575</td>
</tr>
<tr>
<td>pat25549</td>
<td>2.6191</td>
<td>1.7201</td>
<td>2.3984</td>
<td>2.4373</td>
<td>2.293725</td>
</tr>
<tr>
<td>pat25561</td>
<td>2.8761</td>
<td>2.0554</td>
<td>2.601</td>
<td>2.7925</td>
<td>2.58125</td>
</tr>
<tr>
<td>pat25570</td>
<td>3.6297</td>
<td>2.6162</td>
<td>2.8004</td>
<td>3.0971</td>
<td>3.03585</td>
</tr>
<tr>
<td>pat25571</td>
<td>3.3022</td>
<td>1.4115</td>
<td>5.0947</td>
<td>4.3819</td>
<td>3.547575</td>
</tr>
<tr>
<td>pat25573</td>
<td>2.5556</td>
<td>1.6772</td>
<td>2.47</td>
<td>2.463</td>
<td>2.29145</td>
</tr>
<tr>
<td>pat25579</td>
<td>2.8133</td>
<td>1.9965</td>
<td>2.4352</td>
<td>2.7528</td>
<td>2.49945</td>
</tr>
<tr>
<td>pat25579</td>
<td>4.4765</td>
<td>2.7968</td>
<td>3.3582</td>
<td>4.2541</td>
<td>3.7214</td>
</tr>
<tr>
<td>pat25761</td>
<td>4.6043</td>
<td>1.7856</td>
<td>3.2402</td>
<td>3.9079</td>
<td>3.3845</td>
</tr>
<tr>
<td>pat25765</td>
<td>3.0443</td>
<td>1.7446</td>
<td>2.3785</td>
<td>2.4894</td>
<td>2.4142</td>
</tr>
<tr>
<td>pat25777</td>
<td>3.2033</td>
<td>2.0566</td>
<td>2.7657</td>
<td>3.1887</td>
<td>2.803575</td>
</tr>
<tr>
<td>pat25813</td>
<td>3.2135</td>
<td>1.9683</td>
<td>2.551</td>
<td>2.945</td>
<td>2.66945</td>
</tr>
<tr>
<td>pat26002</td>
<td>4.5136</td>
<td>2.8436</td>
<td>3.3956</td>
<td>4.2614</td>
<td>3.75355</td>
</tr>
<tr>
<td>pat26003</td>
<td>3.3622</td>
<td>1.5651</td>
<td>5.2757</td>
<td>4.0061</td>
<td>3.543275</td>
</tr>
<tr>
<td>pat26005</td>
<td>2.8864</td>
<td>1.7138</td>
<td>2.612</td>
<td>2.5176</td>
<td>2.43245</td>
</tr>
<tr>
<td>pat26011</td>
<td>3.0158</td>
<td>2.044</td>
<td>2.7972</td>
<td>3.158</td>
<td>2.75375</td>
</tr>
<tr>
<td>pat26029</td>
<td>3.1179</td>
<td>1.9895</td>
<td>2.751</td>
<td>2.9861</td>
<td>2.711125</td>
</tr>
<tr>
<td>pat267</td>
<td>2.9456</td>
<td>1.5609</td>
<td>2.7547</td>
<td>2.9459</td>
<td>2.551775</td>
</tr>
<tr>
<td>pat2682</td>
<td>4.1725</td>
<td>2.5907</td>
<td>3.364</td>
<td>3.5919</td>
<td>3.429775</td>
</tr>
<tr>
<td>pat2694</td>
<td>3.8861</td>
<td>2.6087</td>
<td>3.8285</td>
<td>3.7577</td>
<td>3.52025</td>
</tr>
<tr>
<td>pat270</td>
<td>3.8754</td>
<td>2.0803</td>
<td>3.42</td>
<td>3.6779</td>
<td>3.2634</td>
</tr>
<tr>
<td>pat2730</td>
<td>4.4115</td>
<td>2.7385</td>
<td>3.7481</td>
<td>2.5316</td>
<td>3.357425</td>
</tr>
<tr>
<td>pat2734</td>
<td>10.454</td>
<td>4.2505</td>
<td>7.787</td>
<td>5.8862</td>
<td>7.094425</td>
</tr>
<tr>
<td>pat2746</td>
<td>9.7407</td>
<td>5.7785</td>
<td>10.6118</td>
<td>6.7558</td>
<td>8.2217</td>
</tr>
<tr>
<td>pat2836</td>
<td>13.0596</td>
<td>5.5004</td>
<td>12.3716</td>
<td>11.3236</td>
<td>10.5638</td>
</tr>
<tr>
<td>pat2838</td>
<td>4.1073</td>
<td>2.518</td>
<td>3.5058</td>
<td>3.563</td>
<td>3.423525</td>
</tr>
<tr>
<td>pat2842</td>
<td>9.2639</td>
<td>2.9206</td>
<td>7.7793</td>
<td>8.2314</td>
<td>7.0488</td>
</tr>
<tr>
<td>pat2854</td>
<td>8.4289</td>
<td>3.9174</td>
<td>10.6646</td>
<td>9.5271</td>
<td>8.1345</td>
</tr>
<tr>
<td>pat303</td>
<td>3.4</td>
<td>1.6676</td>
<td>2.8382</td>
<td>2.4933</td>
<td>2.600275</td>
</tr>
<tr>
<td>pat306</td>
<td>3.7497</td>
<td>2.0865</td>
<td>3.5659</td>
<td>3.7864</td>
<td>3.297125</td>
</tr>
<tr>
<td>pat318</td>
<td>3.7465</td>
<td>1.9293</td>
<td>3.3142</td>
<td>2.9722</td>
<td>2.99055</td>
</tr>
<tr>
<td>pat319</td>
<td>6.9112</td>
<td>3.1443</td>
<td>7.2025</td>
<td>5.1338</td>
<td>5.59795</td>
</tr>
<tr>
<td>pat321</td>
<td>3.3059</td>
<td>1.5453</td>
<td>2.6659</td>
<td>3.0269</td>
<td>2.636</td>
</tr>
<tr>
<td>pat32879</td>
<td>2.6396</td>
<td>1.6058</td>
<td>2.5344</td>
<td>2.4693</td>
<td>5.215175</td>
</tr>
<tr>
<td>pat32830</td>
<td>6.5802</td>
<td>4.3848</td>
<td>3.9837</td>
<td>5.912</td>
<td>6.536975</td>
</tr>
<tr>
<td>pat32831</td>
<td>3.3901</td>
<td>1.646</td>
<td>5.3037</td>
<td>4.3704</td>
<td>3.67755</td>
</tr>
<tr>
<td>pat32835</td>
<td>3.1585</td>
<td>2.0846</td>
<td>2.5087</td>
<td>2.3969</td>
<td>2.537175</td>
</tr>
<tr>
<td>pat32839</td>
<td>7.0125</td>
<td>4.2015</td>
<td>3.7321</td>
<td>4.4767</td>
<td>4.8557</td>
</tr>
<tr>
<td>pat32841</td>
<td>2.7548</td>
<td>1.7719</td>
<td>2.7267</td>
<td>2.926</td>
<td>2.54485</td>
</tr>
<tr>
<td>pat32851</td>
<td>7.1844</td>
<td>5.6379</td>
<td>5.3276</td>
<td>5.0998</td>
<td>5.39525</td>
</tr>
<tr>
<td>pat32853</td>
<td>2.7543</td>
<td>1.6677</td>
<td>2.442</td>
<td>2.5733</td>
<td>2.359325</td>
</tr>
<tr>
<td>pat32857</td>
<td>4.4093</td>
<td>2.9127</td>
<td>3.2175</td>
<td>3.7163</td>
<td>3.56395</td>
</tr>
<tr>
<td>pat32861</td>
<td>4.0306</td>
<td>2.2433</td>
<td>5.3</td>
<td>3.7373</td>
<td>3.8353</td>
</tr>
<tr>
<td>pat32862</td>
<td>3.3631</td>
<td>2.0886</td>
<td>2.5525</td>
<td>2.6813</td>
<td>2.671375</td>
</tr>
<tr>
<td>pat32863</td>
<td>6.8589</td>
<td>4.2358</td>
<td>3.9357</td>
<td>3.928</td>
<td>4.7396</td>
</tr>
<tr>
<td>pat32865</td>
<td>2.7166</td>
<td>1.6092</td>
<td>2.6826</td>
<td>2.6459</td>
<td>2.413575</td>
</tr>
<tr>
<td>pat32867</td>
<td>3.4861</td>
<td>1.6676</td>
<td>5.2295</td>
<td>4.0553</td>
<td>3.609625</td>
</tr>
<tr>
<td>pat32869</td>
<td>7.3349</td>
<td>5.6996</td>
<td>5.2341</td>
<td>4.7176</td>
<td>5.74655</td>
</tr>
<tr>
<td>pat32871</td>
<td>2.7744</td>
<td>1.6865</td>
<td>2.4432</td>
<td>2.5562</td>
<td>2.365075</td>
</tr>
<tr>
<td>pat32875</td>
<td>4.4979</td>
<td>2.9599</td>
<td>3.1487</td>
<td>3.3294</td>
<td>3.483975</td>
</tr>
<tr>
<td>pat32878</td>
<td>7.3162</td>
<td>5.7342</td>
<td>5.148</td>
<td>4.8367</td>
<td>5.758775</td>
</tr>
<tr>
<td>pat32879</td>
<td>3.4124</td>
<td>1.8192</td>
<td>5.3168</td>
<td>3.8187</td>
<td>3.591775</td>
</tr>
<tr>
<td>pat32881</td>
<td>4.2981</td>
<td>3.0114</td>
<td>3.3808</td>
<td>3.1595</td>
<td>3.46245</td>
</tr>
<tr>
<td>pat32969</td>
<td>3.8232</td>
<td>1.9478</td>
<td>5.2872</td>
<td>4.349</td>
<td>3.8518</td>
</tr>
<tr>
<td>pat32971</td>
<td>6.355</td>
<td>3.4979</td>
<td>3.9022</td>
<td>5.0606</td>
<td>4.703925</td>
</tr>
<tr>
<td>pat32977</td>
<td>6.5279</td>
<td>4.7683</td>
<td>5.2477</td>
<td>6.3134</td>
<td>5.714325</td>
</tr>
<tr>
<td>pat32995</td>
<td>7.1343</td>
<td>4.4646</td>
<td>4.7413</td>
<td>4.6086</td>
<td>5.2372</td>
</tr>
<tr>
<td>pat33051</td>
<td>3.6576</td>
<td>2.0583</td>
<td>2.4444</td>
<td>3.0537</td>
<td>2.8035</td>
</tr>
<tr>
<td>pat3357</td>
<td>3.0404</td>
<td>1.7723</td>
<td>2.5127</td>
<td>2.9472</td>
<td>2.56815</td>
</tr>
<tr>
<td>pat33067</td>
<td>9.4468</td>
<td>5.6524</td>
<td>6.3302</td>
<td>9.2297</td>
<td>7.664775</td>
</tr>
<tr>
<td>pat33069</td>
<td>2.9606</td>
<td>1.678</td>
<td>2.3203</td>
<td>2.8669</td>
<td>2.45645</td>
</tr>
<tr>
<td>pat33073</td>
<td>5.6352</td>
<td>2.6555</td>
<td>3.4436</td>
<td>5.0006</td>
<td>4.311225</td>
</tr>
<tr>
<td>pat33103</td>
<td>10.2275</td>
<td>5.3559</td>
<td>5.5175</td>
<td>5.3725</td>
<td>6.61835</td>
</tr>
<tr>
<td>pat33105</td>
<td>3.1374</td>
<td>1.7107</td>
<td>2.4631</td>
<td>2.6008</td>
<td>2.478</td>
</tr>
<tr>
<td>pat33109</td>
<td>6.0617</td>
<td>2.7412</td>
<td>3.1945</td>
<td>3.8687</td>
<td>3.966525</td>
</tr>
<tr>
<td>pat33121</td>
<td>6.4065</td>
<td>3.8497</td>
<td>4.6038</td>
<td>4.6721</td>
<td>4.883025</td>
</tr>
<tr>
<td>pat33211</td>
<td>9.2682</td>
<td>4.1712</td>
<td>5.5366</td>
<td>7.7953</td>
<td>6.692825</td>
</tr>
<tr>
<td>pat33293</td>
<td>4.3872</td>
<td>2.0801</td>
<td>5.2419</td>
<td>4.6886</td>
<td>4.09945</td>
</tr>
<tr>
<td>pat33294</td>
<td>3.6592</td>
<td>2.2286</td>
<td>2.4428</td>
<td>3.0963</td>
<td>2.856725</td>
</tr>
<tr>
<td>pat33297</td>
<td>2.7948</td>
<td>1.7213</td>
<td>2.43</td>
<td>2.4207</td>
<td>2.3417</td>
</tr>
<tr>
<td>pat33299</td>
<td>3.7066</td>
<td>1.5756</td>
<td>5.1531</td>
<td>4.3852</td>
<td>3.705125</td>
</tr>
<tr>
<td>pat33301</td>
<td>9.3488</td>
<td>5.4427</td>
<td>6.421</td>
<td>9.2135</td>
<td>7.6065</td>
</tr>
<tr>
<td>pat33303</td>
<td>2.8342</td>
<td>1.8128</td>
<td>2.3342</td>
<td>2.771</td>
<td>2.43085</td>
</tr>
<tr>
<td>pat33307</td>
<td>5.8486</td>
<td>2.7207</td>
<td>3.5699</td>
<td>5.6791</td>
<td>4.454575</td>
</tr>
<tr>
<td>pat33310</td>
<td>9.2698</td>
<td>5.567</td>
<td>6.3093</td>
<td>9.3019</td>
<td>7.612</td>
</tr>
<tr>
<td>pat33311</td>
<td>3.5473</td>
<td>1.6658</td>
<td>5.3465</td>
<td>4.6335</td>
<td>3.798275</td>
</tr>
<tr>
<td>pat33313</td>
<td>5.5837</td>
<td>2.5845</td>
<td>3.7915</td>
<td>5.4237</td>
<td>4.34685</td>
</tr>
<tr>
<td>pat33319</td>
<td>9.8971</td>
<td>5.0789</td>
<td>5.6012</td>
<td>5.7859</td>
<td>6.590775</td>
</tr>
<tr>
<td>pat33321</td>
<td>2.95</td>
<td>1.8493</td>
<td>2.4642</td>
<td>2.4781</td>
<td>2.4354</td>
</tr>
<tr>
<td>pat33325</td>
<td>6.1234</td>
<td>2.716</td>
<td>3.3524</td>
<td>4.2871</td>
<td>4.119725</td>
</tr>
<tr>
<td>pat33337</td>
<td>6.37</td>
<td>3.7183</td>
<td>4.7505</td>
<td>4.9479</td>
<td>4.946675</td>
</tr>
<tr>
<td>pat33346</td>
<td>10.1253</td>
<td>5.1865</td>
<td>5.4133</td>
<td>5.4125</td>
<td>6.5344</td>
</tr>
<tr>
<td>pat33347</td>
<td>3.791</td>
<td>1.777</td>
<td>5.2405</td>
<td>3.831</td>
<td>3.659875</td>
</tr>
<tr>
<td>pat33349</td>
<td>5.9405</td>
<td>2.6902</td>
<td>3.5143</td>
<td>3.7487</td>
<td>3.973425</td>
</tr>
<tr>
<td>pat33355</td>
<td>6.3449</td>
<td>3.7313</td>
<td>4.6605</td>
<td>4.4394</td>
<td>4.794025</td>
</tr>
<tr>
<td>pat33537</td>
<td>2.8908</td>
<td>1.469</td>
<td>2.8958</td>
<td>1.9345</td>
<td>2.297525</td>
</tr>
<tr>
<td>pat33541</td>
<td>3.7818</td>
<td>2.1646</td>
<td>3.2852</td>
<td>2.6175</td>
<td>2.962275</td>
</tr>
<tr>
<td>pat33543</td>
<td>2.8865</td>
<td>1.4911</td>
<td>3.1915</td>
<td>2.1397</td>
<td>2.4272</td>
</tr>
<tr>
<td>pat33553</td>
<td>3.7019</td>
<td>2.8425</td>
<td>3.902</td>
<td>3.2589</td>
<td>3.426325</td>
</tr>
<tr>
<td>pat33555</td>
<td>3.0061</td>
<td>1.4287</td>
<td>2.9253</td>
<td>2.1022</td>
<td>3.665575</td>
</tr>
<tr>
<td>pat33559</td>
<td>3.0279</td>
<td>1.758</td>
<td>3.0883</td>
<td>2.2351</td>
<td>2.527325</td>
</tr>
<tr>
<td>pat33589</td>
<td>4.0671</td>
<td>2.7053</td>
<td>3.6632</td>
<td>2.5674</td>
<td>3.25075</td>
</tr>
<tr>
<td>pat33591</td>
<td>2.9248</td>
<td>1.348</td>
<td>3.0336</td>
<td>2.4163</td>
<td>2.430675</td>
</tr>
<tr>
<td>pat33595</td>
<td>3.0495</td>
<td>1.6868</td>
<td>2.9942</td>
<td>2.1052</td>
<td>2.458925</td>
</tr>
<tr>
<td>pat33607</td>
<td>3.1513</td>
<td>2.0388</td>
<td>3.2889</td>
<td>2.1553</td>
<td>2.658575</td>
</tr>
<tr>
<td>pat33697</td>
<td>3.7024</td>
<td>2.3216</td>
<td>3.5785</td>
<td>2.7461</td>
<td>3.08715</td>
</tr>
<tr>
<td>pat34021</td>
<td>4.9255</td>
<td>2.8206</td>
<td>4.1909</td>
<td>4.0608</td>
<td>3.99945</td>
</tr>
<tr>
<td>pat34023</td>
<td>2.8803</td>
<td>1.5176</td>
<td>2.8847</td>
<td>1.9613</td>
<td>2.310975</td>
</tr>
<tr>
<td>pat34027</td>
<td>3.4358</td>
<td>1.5921</td>
<td>3.0593</td>
<td>2.4455</td>
<td>2.633175</td>
</tr>
<tr>
<td>pat34039</td>
<td>3.3668</td>
<td>2.0053</td>
<td>3.5964</td>
<td>3.1019</td>
<td>3.0176</td>
</tr>
<tr>
<td>pat34075</td>
<td>3.7108</td>
<td>1.9489</td>
<td>3.3895</td>
<td>2.4448</td>
<td>2.8735</td>
</tr>
<tr>
<td>pat34265</td>
<td>3.3648</td>
<td>1.6017</td>
<td>5.3223</td>
<td>4.0674</td>
<td>3.58905</td>
</tr>
<tr>
<td>pat34266</td>
<td>4.4395</td>
<td>1.9368</td>
<td>2.9866</td>
<td>3.7343</td>
<td>3.2743</td>
</tr>
<tr>
<td>pat34267</td>
<td>3.2644</td>
<td>2.287</td>
<td>2.9047</td>
<td>2.7962</td>
<td>2.813075</td>
</tr>
<tr>
<td>pat34269</td>
<td>4.3031</td>
<td>1.7614</td>
<td>3.3201</td>
<td>3.5479</td>
<td>3.233125</td>
</tr>
<tr>
<td>pat34270</td>
<td>3.4256</td>
<td>2.3166</td>
<td>2.617</td>
<td>2.7846</td>
<td>2.78595</td>
</tr>
<tr>
<td>pat34271</td>
<td>3.2947</td>
<td>1.5358</td>
<td>5.3042</td>
<td>3.9055</td>
<td>3.51005</td>
</tr>
<tr>
<td>pat34273</td>
<td>3.3937</td>
<td>2.9104</td>
<td>3.2629</td>
<td>3.5172</td>
<td>3.27105</td>
</tr>
<tr>
<td>pat34275</td>
<td>4.4705</td>
<td>1.7422</td>
<td>3.0164</td>
<td>3.7057</td>
<td>3.2337</td>
</tr>
<tr>
<td>pat34279</td>
<td>2.7634</td>
<td>1.8817</td>
<td>2.488</td>
<td>2.3628</td>
<td>2.368975</td>
</tr>
<tr>
<td>pat34282</td>
<td>3.5905</td>
<td>2.979</td>
<td>3.2204</td>
<td>3.5319</td>
<td>3.33045</td>
</tr>
<tr>
<td>pat34283</td>
<td>3.5299</td>
<td>1.5405</td>
<td>5.2304</td>
<td>4.1125</td>
<td>3.603325</td>
</tr>
<tr>
<td>pat34285</td>
<td>2.7964</td>
<td>1.9532</td>
<td>2.6814</td>
<td>2.3529</td>
<td>2.445975</td>
</tr>
<tr>
<td>pat34291</td>
<td>3.629</td>
<td>2.7869</td>
<td>3.143</td>
<td>3.1833</td>
<td>3.18555</td>
</tr>
<tr>
<td>pat34293</td>
<td>4.4817</td>
<td>1.7356</td>
<td>3.1176</td>
<td>4.1671</td>
<td>3.3755</td>
</tr>
<tr>
<td>pat34297</td>
<td>2.6809</td>
<td>1.7912</td>
<td>2.4625</td>
<td>2.5481</td>
<td>2.370675</td>
</tr>
<tr>
<td>pat34309</td>
<td>2.9802</td>
<td>2.122</td>
<td>2.7065</td>
<td>2.8222</td>
<td>2.639725</td>
</tr>
<tr>
<td>pat34318</td>
<td>3.7687</td>
<td>2.8144</td>
<td>2.9997</td>
<td>3.1385</td>
<td>3.180325</td>
</tr>
<tr>
<td>pat34319</td>
<td>3.2821</td>
<td>1.4451</td>
<td>5.1405</td>
<td>4.4025</td>
<td>3.56755</td>
</tr>
<tr>
<td>pat34321</td>
<td>2.6235</td>
<td>1.7214</td>
<td>2.5121</td>
<td>2.444</td>
<td>2.32525</td>
</tr>
<tr>
<td>pat34327</td>
<td>2.8635</td>
<td>2.0458</td>
<td>2.5519</td>
<td>2.7169</td>
<td>2.544525</td>
</tr>
<tr>
<td>pat34507</td>
<td>4.6445</td>
<td>2.9535</td>
<td>3.6474</td>
<td>4.4527</td>
<td>3.924525</td>
</tr>
<tr>
<td>pat34509</td>
<td>4.397</td>
<td>1.785</td>
<td>2.9104</td>
<td>3.6792</td>
<td>3.1929</td>
</tr>
<tr>
<td>pat34513</td>
<td>3.1086</td>
<td>1.785</td>
<td>2.4575</td>
<td>2.6083</td>
<td>2.48985</td>
</tr>
<tr>
<td>pat34525</td>
<td>3.229</td>
<td>2.1304</td>
<td>2.9195</td>
<td>3.3285</td>
<td>2.90185</td>
</tr>
<tr>
<td>pat34561</td>
<td>3.3208</td>
<td>1.9982</td>
<td>2.651</td>
<td>2.9661</td>
<td>2.734025</td>
</tr>
<tr>
<td>pat34750</td>
<td>4.6993</td>
<td>2.974</td>
<td>3.6499</td>
<td>4.5374</td>
<td>3.96515</td>
</tr>
<tr>
<td>pat34751</td>
<td>3.2875</td>
<td>1.5378</td>
<td>5.208</td>
<td>3.9972</td>
<td>3.507625</td>
</tr>
<tr>
<td>pat34753</td>
<td>2.9386</td>
<td>1.7466</td>
<td>2.687</td>
<td>2.6338</td>
<td>2.5015</td>
</tr>
<tr>
<td>pat34759</td>
<td>3.0361</td>
<td>2.1113</td>
<td>2.9223</td>
<td>3.289</td>
<td>2.839675</td>
</tr>
<tr>
<td>pat34777</td>
<td>3.2214</td>
<td>2.0207</td>
<td>2.8204</td>
<td>2.9957</td>
<td>2.76455</td>
</tr>
<tr>
<td>pat3654</td>
<td>7.7559</td>
<td>3.9189</td>
<td>5.4889</td>
<td>6.0254</td>
<td>5.797275</td>
</tr>
<tr>
<td>pat3666</td>
<td>7.4489</td>
<td>2.7469</td>
<td>3.5913</td>
<td>5.0741</td>
<td>4.7153</td>
</tr>
<tr>
<td>pat3670</td>
<td>4.049</td>
<td>3.0174</td>
<td>4.3594</td>
<td>3.7269</td>
<td>3.799425</td>
</tr>
<tr>
<td>pat3702</td>
<td>6.7189</td>
<td>2.4559</td>
<td>3.8677</td>
<td>7.4025</td>
<td>5.11125</td>
</tr>
<tr>
<td>pat3706</td>
<td>4.2053</td>
<td>2.8774</td>
<td>3.8928</td>
<td>3.7256</td>
<td>3.675275</td>
</tr>
<tr>
<td>pat3718</td>
<td>4.2062</td>
<td>3.4797</td>
<td>4.7293</td>
<td>3.7972</td>
<td>4.0531</td>
</tr>
<tr>
<td>pat37181</td>
<td>4.5384</td>
<td>2.6707</td>
<td>5.2082</td>
<td>4.6311</td>
<td>4.2621</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>pat37183</td>
<td>9.4403</td>
<td>5.6434</td>
<td>4.322</td>
<td>6.952</td>
<td>6.589425</td>
</tr>
<tr>
<td>pat37186</td>
<td>9.675</td>
<td>5.7027</td>
<td>4.0166</td>
<td>7.134</td>
<td>6.632075</td>
</tr>
<tr>
<td>pat37187</td>
<td>3.7636</td>
<td>1.884</td>
<td>5.0238</td>
<td>4.268</td>
<td>3.73485</td>
</tr>
<tr>
<td>pat37189</td>
<td>9.819</td>
<td>7.8034</td>
<td>6.2625</td>
<td>8.8664</td>
<td>8.192825</td>
</tr>
<tr>
<td>pat37195</td>
<td>6.1566</td>
<td>4.0233</td>
<td>3.5459</td>
<td>5.4321</td>
<td>4.789475</td>
</tr>
<tr>
<td>pat37207</td>
<td>10.3148</td>
<td>7.0599</td>
<td>5.4838</td>
<td>5.7224</td>
<td>7.145225</td>
</tr>
<tr>
<td>pat37213</td>
<td>6.4325</td>
<td>3.8135</td>
<td>3.3828</td>
<td>4.2662</td>
<td>4.47375</td>
</tr>
<tr>
<td>pat37429</td>
<td>8.3758</td>
<td>3.5959</td>
<td>3.4815</td>
<td>6.6076</td>
<td>5.5152</td>
</tr>
<tr>
<td>pat37909</td>
<td>5.3779</td>
<td>3.9113</td>
<td>4.2561</td>
<td>4.1583</td>
<td>4.4259</td>
</tr>
<tr>
<td>pat37915</td>
<td>3.578</td>
<td>2.0476</td>
<td>3.0945</td>
<td>2.4402</td>
<td>2.790075</td>
</tr>
<tr>
<td>pat3808</td>
<td>5.5446</td>
<td>4.2009</td>
<td>6.4271</td>
<td>5.4319</td>
<td>5.401125</td>
</tr>
<tr>
<td>pat3810</td>
<td>7.7573</td>
<td>3.6308</td>
<td>3.8604</td>
<td>5.6764</td>
<td>5.231225</td>
</tr>
<tr>
<td>pat3814</td>
<td>4.1884</td>
<td>2.7035</td>
<td>3.8706</td>
<td>3.2452</td>
<td>3.501925</td>
</tr>
<tr>
<td>pat39420</td>
<td>6.161</td>
<td>3.3343</td>
<td>8.147</td>
<td>5.4106</td>
<td>5.763225</td>
</tr>
<tr>
<td>pat39438</td>
<td>4.4008</td>
<td>3.0232</td>
<td>8.9026</td>
<td>7.9416</td>
<td>6.06705</td>
</tr>
<tr>
<td>pat39446</td>
<td>2.7496</td>
<td>1.5628</td>
<td>2.7609</td>
<td>2.9006</td>
<td>2.493475</td>
</tr>
<tr>
<td>pat39456</td>
<td>3.4878</td>
<td>2.29</td>
<td>5.4787</td>
<td>3.975</td>
<td>3.807875</td>
</tr>
<tr>
<td>pat39468</td>
<td>3.9818</td>
<td>1.9475</td>
<td>4.406</td>
<td>3.8009</td>
<td>3.53405</td>
</tr>
<tr>
<td>pat39472</td>
<td>4.2711</td>
<td>2.6178</td>
<td>3.2911</td>
<td>3.8459</td>
<td>3.506475</td>
</tr>
<tr>
<td>pat39474</td>
<td>6.188</td>
<td>3.35</td>
<td>8.1599</td>
<td>5.3853</td>
<td>5.7708</td>
</tr>
<tr>
<td>pat39504</td>
<td>3.3754</td>
<td>1.7013</td>
<td>4.8088</td>
<td>4.5338</td>
<td>3.604825</td>
</tr>
<tr>
<td>pat39508</td>
<td>4.5936</td>
<td>2.6657</td>
<td>3.2065</td>
<td>3.5512</td>
<td>3.50425</td>
</tr>
<tr>
<td>pat39510</td>
<td>4.4745</td>
<td>3.1129</td>
<td>8.9822</td>
<td>8.0577</td>
<td>6.156825</td>
</tr>
<tr>
<td>pat39520</td>
<td>4.5549</td>
<td>3.2029</td>
<td>3.742</td>
<td>3.6629</td>
<td>3.790675</td>
</tr>
<tr>
<td>pat39526</td>
<td>3.3328</td>
<td>2.0382</td>
<td>3.0287</td>
<td>3.0768</td>
<td>2.869125</td>
</tr>
<tr>
<td>pat39534</td>
<td>3.7791</td>
<td>2.2614</td>
<td>4.9511</td>
<td>3.917</td>
<td>3.72715</td>
</tr>
<tr>
<td>pat39536</td>
<td>2.7691</td>
<td>1.6988</td>
<td>2.6805</td>
<td>2.4791</td>
<td>2.406875</td>
</tr>
<tr>
<td>pat39540</td>
<td>3.6862</td>
<td>2.3387</td>
<td>5.0831</td>
<td>4.1404</td>
<td>3.8121</td>
</tr>
<tr>
<td>pat39544</td>
<td>4.119</td>
<td>2.6711</td>
<td>3.4303</td>
<td>3.8631</td>
<td>3.520875</td>
</tr>
<tr>
<td>pat39548</td>
<td>2.6934</td>
<td>1.6219</td>
<td>2.3867</td>
<td>2.7469</td>
<td>2.362225</td>
</tr>
<tr>
<td>pat39550</td>
<td>4.2491</td>
<td>2.5334</td>
<td>3.179</td>
<td>3.9245</td>
<td>3.4715</td>
</tr>
<tr>
<td>pat39558</td>
<td>3.9313</td>
<td>2.3705</td>
<td>5.0739</td>
<td>3.859</td>
<td>3.808675</td>
</tr>
<tr>
<td>pat39562</td>
<td>4.5951</td>
<td>2.7009</td>
<td>3.2589</td>
<td>3.0562</td>
<td>3.402775</td>
</tr>
<tr>
<td>pat39574</td>
<td>4.5934</td>
<td>3.2374</td>
<td>3.7631</td>
<td>3.4918</td>
<td>3.771425</td>
</tr>
<tr>
<td>pat39584</td>
<td>2.8226</td>
<td>1.6691</td>
<td>2.5649</td>
<td>2.824</td>
<td>2.47015</td>
</tr>
<tr>
<td>pat39586</td>
<td>4.5783</td>
<td>2.5679</td>
<td>3.0851</td>
<td>3.4711</td>
<td>3.4256</td>
</tr>
<tr>
<td>pat39592</td>
<td>4.4912</td>
<td>3.2519</td>
<td>3.8595</td>
<td>3.7642</td>
<td>3.8417</td>
</tr>
<tr>
<td>pat39633</td>
<td>3.9981</td>
<td>1.6866</td>
<td>4.3717</td>
<td>3.75</td>
<td>3.4516</td>
</tr>
<tr>
<td>pat39636</td>
<td>6.461</td>
<td>3.4516</td>
<td>8.2667</td>
<td>5.6014</td>
<td>5.945175</td>
</tr>
<tr>
<td>pat39669</td>
<td>3.2299</td>
<td>1.5191</td>
<td>4.7159</td>
<td>4.5681</td>
<td>3.50825</td>
</tr>
<tr>
<td>pat39672</td>
<td>4.6128</td>
<td>3.0144</td>
<td>8.8956</td>
<td>7.8925</td>
<td>6.103825</td>
</tr>
<tr>
<td>pat39685</td>
<td>3.966</td>
<td>1.9419</td>
<td>2.8112</td>
<td>3.2883</td>
<td>3.00185</td>
</tr>
<tr>
<td>pat39688</td>
<td>4.1414</td>
<td>2.1161</td>
<td>3.2833</td>
<td>3.6883</td>
<td>3.307275</td>
</tr>
<tr>
<td>pat39774</td>
<td>4.117</td>
<td>2.3067</td>
<td>5.0418</td>
<td>4.1397</td>
<td>3.9013</td>
</tr>
<tr>
<td>pat39775</td>
<td>5.7156</td>
<td>2.156</td>
<td>3.2526</td>
<td>4.8876</td>
<td>4.00295</td>
</tr>
<tr>
<td>pat39777</td>
<td>3.7481</td>
<td>2.166</td>
<td>5.0016</td>
<td>3.9036</td>
<td>3.704825</td>
</tr>
<tr>
<td>pat39778</td>
<td>5.6511</td>
<td>2.1676</td>
<td>3.5406</td>
<td>4.7705</td>
<td>4.03245</td>
</tr>
<tr>
<td>pat39790</td>
<td>5.8282</td>
<td>2.9109</td>
<td>4.6925</td>
<td>6.0885</td>
<td>4.880025</td>
</tr>
<tr>
<td>pat39793</td>
<td>3.6801</td>
<td>1.618</td>
<td>2.8058</td>
<td>3.4662</td>
<td>2.892525</td>
</tr>
<tr>
<td>pat39826</td>
<td>6.3085</td>
<td>2.9152</td>
<td>4.2627</td>
<td>4.3518</td>
<td>4.45955</td>
</tr>
<tr>
<td>pat39829</td>
<td>3.98</td>
<td>1.7508</td>
<td>2.8142</td>
<td>3.2833</td>
<td>2.957075</td>
</tr>
<tr>
<td>pat39870</td>
<td>3.9766</td>
<td>3.7559</td>
<td>6.3361</td>
<td>4.4379</td>
<td>4.626625</td>
</tr>
<tr>
<td>pat39876</td>
<td>4.678</td>
<td>2.9647</td>
<td>4.9763</td>
<td>3.965</td>
<td>4.146</td>
</tr>
<tr>
<td>pat39878</td>
<td>2.7212</td>
<td>1.6554</td>
<td>2.4415</td>
<td>2.6299</td>
<td>2.362</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>pat39888</td>
<td>4.0514</td>
<td>3.7216</td>
<td>6.3156</td>
<td>4.1867</td>
<td>4.568825</td>
</tr>
<tr>
<td>pat39900</td>
<td>4.6958</td>
<td>3.0676</td>
<td>4.9904</td>
<td>3.7988</td>
<td>4.13815</td>
</tr>
<tr>
<td>pat39904</td>
<td>4.1066</td>
<td>2.2543</td>
<td>2.9839</td>
<td>3.0482</td>
<td>3.09825</td>
</tr>
<tr>
<td>pat39912</td>
<td>3.7431</td>
<td>2.8125</td>
<td>5.3983</td>
<td>4.9722</td>
<td>4.231525</td>
</tr>
<tr>
<td>pat39914</td>
<td>2.6516</td>
<td>1.7524</td>
<td>2.6173</td>
<td>2.8834</td>
<td>2.476175</td>
</tr>
<tr>
<td>pat39918</td>
<td>3.7022</td>
<td>2.806</td>
<td>5.4048</td>
<td>5.02</td>
<td>4.23325</td>
</tr>
<tr>
<td>pat39922</td>
<td>3.9492</td>
<td>2.3106</td>
<td>3.1479</td>
<td>3.5627</td>
<td>3.2426</td>
</tr>
<tr>
<td>pat39926</td>
<td>2.7269</td>
<td>1.6206</td>
<td>2.2863</td>
<td>2.3982</td>
<td>2.258</td>
</tr>
<tr>
<td>pat39928</td>
<td>4.0823</td>
<td>2.1303</td>
<td>2.8985</td>
<td>3.374</td>
<td>3.121275</td>
</tr>
<tr>
<td>pat39936</td>
<td>3.3469</td>
<td>2.2461</td>
<td>3.7466</td>
<td>3.2945</td>
<td>3.158525</td>
</tr>
<tr>
<td>pat39940</td>
<td>5.962</td>
<td>2.3656</td>
<td>3.4095</td>
<td>5.0152</td>
<td>4.180075</td>
</tr>
<tr>
<td>pat39942</td>
<td>4.0176</td>
<td>3.6992</td>
<td>6.324</td>
<td>4.2378</td>
<td>4.56966</td>
</tr>
<tr>
<td>pat39952</td>
<td>5.8367</td>
<td>3.0198</td>
<td>4.5799</td>
<td>6.1809</td>
<td>4.904325</td>
</tr>
<tr>
<td>pat39954</td>
<td>4.6355</td>
<td>2.9326</td>
<td>4.9631</td>
<td>4.0069</td>
<td>4.134525</td>
</tr>
<tr>
<td>pat39958</td>
<td>3.7373</td>
<td>1.8599</td>
<td>2.9501</td>
<td>3.4649</td>
<td>3.00305</td>
</tr>
<tr>
<td>pat39988</td>
<td>6.3806</td>
<td>3.0138</td>
<td>4.0667</td>
<td>4.1696</td>
<td>4.407675</td>
</tr>
<tr>
<td>pat39990</td>
<td>3.8018</td>
<td>2.8427</td>
<td>5.3918</td>
<td>5.0481</td>
<td>4.2711</td>
</tr>
<tr>
<td>pat39994</td>
<td>3.9729</td>
<td>1.9906</td>
<td>2.9644</td>
<td>3.356</td>
<td>3.070975</td>
</tr>
<tr>
<td>pat40006</td>
<td>3.9772</td>
<td>2.1748</td>
<td>3.3055</td>
<td>3.4676</td>
<td>3.231275</td>
</tr>
<tr>
<td>pat40016</td>
<td>3.0103</td>
<td>1.6725</td>
<td>2.4118</td>
<td>2.8354</td>
<td>2.4825</td>
</tr>
<tr>
<td>pat40018</td>
<td>5.6701</td>
<td>2.1166</td>
<td>3.2399</td>
<td>4.8298</td>
<td>3.9641</td>
</tr>
<tr>
<td>pat40024</td>
<td>5.8483</td>
<td>2.972</td>
<td>4.655</td>
<td>6.2031</td>
<td>4.9196</td>
</tr>
<tr>
<td>pat40042</td>
<td>6.4961</td>
<td>3.1176</td>
<td>4.114</td>
<td>4.1943</td>
<td>4.4805</td>
</tr>
<tr>
<td>pat40122</td>
<td>6.415</td>
<td>3.4496</td>
<td>8.1829</td>
<td>5.196</td>
<td>5.810875</td>
</tr>
<tr>
<td>pat40158</td>
<td>4.5257</td>
<td>2.9453</td>
<td>9.0106</td>
<td>8.077</td>
<td>6.13965</td>
</tr>
<tr>
<td>pat40174</td>
<td>2.677</td>
<td>1.6752</td>
<td>2.6246</td>
<td>2.5654</td>
<td>2.38554</td>
</tr>
<tr>
<td>pat40260</td>
<td>3.7838</td>
<td>2.1712</td>
<td>5.1034</td>
<td>3.7465</td>
<td>3.701225</td>
</tr>
<tr>
<td>pat40264</td>
<td>2.7712</td>
<td>1.7991</td>
<td>2.6125</td>
<td>2.0782</td>
<td>2.31525</td>
</tr>
<tr>
<td>pat40276</td>
<td>2.8309</td>
<td>1.936</td>
<td>2.5438</td>
<td>2.4586</td>
<td>2.443225</td>
</tr>
<tr>
<td>pat40312</td>
<td>2.7573</td>
<td>1.8408</td>
<td>2.4926</td>
<td>2.6149</td>
<td>2.4264</td>
</tr>
<tr>
<td>pat40590</td>
<td>3.9869</td>
<td>3.6929</td>
<td>6.3401</td>
<td>4.0913</td>
<td>4.5278</td>
</tr>
<tr>
<td>pat40602</td>
<td>4.6934</td>
<td>2.8577</td>
<td>4.8872</td>
<td>3.8327</td>
<td>4.06775</td>
</tr>
<tr>
<td>pat40606</td>
<td>2.7294</td>
<td>1.7267</td>
<td>2.4242</td>
<td>2.2524</td>
<td>2.283175</td>
</tr>
<tr>
<td>pat40638</td>
<td>3.7327</td>
<td>2.7426</td>
<td>5.3676</td>
<td>5.1816</td>
<td>4.256125</td>
</tr>
<tr>
<td>pat40642</td>
<td>2.5345</td>
<td>1.7409</td>
<td>2.4262</td>
<td>2.5495</td>
<td>2.312775</td>
</tr>
<tr>
<td>pat40654</td>
<td>2.7596</td>
<td>1.7731</td>
<td>2.3658</td>
<td>2.5048</td>
<td>2.350825</td>
</tr>
<tr>
<td>pat40744</td>
<td>3.1794</td>
<td>1.818</td>
<td>2.6511</td>
<td>2.8742</td>
<td>2.630675</td>
</tr>
<tr>
<td>pat40842</td>
<td>11.6586</td>
<td>5.4009</td>
<td>14.9932</td>
<td>8.6782</td>
<td>10.182725</td>
</tr>
<tr>
<td>pat40848</td>
<td>12.7206</td>
<td>3.7472</td>
<td>10.5555</td>
<td>7.6055</td>
<td>8.6572</td>
</tr>
<tr>
<td>pat40850</td>
<td>4.7072</td>
<td>2.3108</td>
<td>4.2948</td>
<td>2.0566</td>
<td>3.34235</td>
</tr>
<tr>
<td>pat40860</td>
<td>11.6011</td>
<td>5.4846</td>
<td>15.1287</td>
<td>8.9431</td>
<td>10.289375</td>
</tr>
<tr>
<td>pat40872</td>
<td>12.9315</td>
<td>3.6814</td>
<td>10.7221</td>
<td>7.9919</td>
<td>8.831725</td>
</tr>
<tr>
<td>pat40876</td>
<td>4.6328</td>
<td>2.508</td>
<td>4.2534</td>
<td>1.9108</td>
<td>3.32625</td>
</tr>
<tr>
<td>pat40886</td>
<td>3.4955</td>
<td>1.9792</td>
<td>4.5357</td>
<td>4.4434</td>
<td>3.61345</td>
</tr>
<tr>
<td>pat40894</td>
<td>3.4628</td>
<td>2.1268</td>
<td>4.4333</td>
<td>4.0987</td>
<td>3.5304</td>
</tr>
<tr>
<td>pat40908</td>
<td>7.9626</td>
<td>2.4622</td>
<td>8.9146</td>
<td>6.5306</td>
<td>6.4675</td>
</tr>
<tr>
<td>pat40912</td>
<td>3.1398</td>
<td>1.9563</td>
<td>3.3185</td>
<td>1.9899</td>
<td>2.601125</td>
</tr>
<tr>
<td>pat40914</td>
<td>11.4766</td>
<td>5.4867</td>
<td>14.9135</td>
<td>8.6899</td>
<td>10.141675</td>
</tr>
<tr>
<td>pat40924</td>
<td>3.4947</td>
<td>2.0498</td>
<td>3.0753</td>
<td>2.4054</td>
<td>2.7563</td>
</tr>
<tr>
<td>pat40926</td>
<td>12.8402</td>
<td>3.7686</td>
<td>10.3258</td>
<td>7.5698</td>
<td>8.6261</td>
</tr>
<tr>
<td>pat40930</td>
<td>4.6387</td>
<td>2.5204</td>
<td>4.2377</td>
<td>1.7981</td>
<td>3.298725</td>
</tr>
<tr>
<td>pat40960</td>
<td>2.949</td>
<td>1.7561</td>
<td>2.948</td>
<td>3.1168</td>
<td>2.692475</td>
</tr>
<tr>
<td>pat40986</td>
<td>3.5341</td>
<td>2.1965</td>
<td>4.4447</td>
<td>4.0275</td>
<td>3.5507</td>
</tr>
<tr>
<td>pat40988</td>
<td>3.1408</td>
<td>1.657</td>
<td>2.9705</td>
<td>2.0853</td>
<td>2.4634</td>
</tr>
<tr>
<td>pat40990</td>
<td>3.3259</td>
<td>1.93</td>
<td>3.031</td>
<td>1.9933</td>
<td>2.57005</td>
</tr>
<tr>
<td>pat40996</td>
<td>3.2464</td>
<td>2.1922</td>
<td>3.3277</td>
<td>2.4201</td>
<td>2.7966</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>pat41014</td>
<td>3.3975</td>
<td>2.1132</td>
<td>3.1108</td>
<td>2.125</td>
<td>2.686625</td>
</tr>
<tr>
<td>pat41073</td>
<td>7.8856</td>
<td>2.4959</td>
<td>8.9416</td>
<td>6.5131</td>
<td>6.45905</td>
</tr>
<tr>
<td>pat41076</td>
<td>11.5459</td>
<td>5.3846</td>
<td>14.9779</td>
<td>8.654</td>
<td>10.1406</td>
</tr>
<tr>
<td>pat41088</td>
<td>13.0374</td>
<td>3.8494</td>
<td>10.6674</td>
<td>7.8271</td>
<td>8.845325</td>
</tr>
<tr>
<td>pat41089</td>
<td>3.4246</td>
<td>1.6181</td>
<td>2.8408</td>
<td>2.2261</td>
<td>2.5274</td>
</tr>
<tr>
<td>pat41091</td>
<td>13.0271</td>
<td>3.7763</td>
<td>10.6207</td>
<td>7.7478</td>
<td>8.792975</td>
</tr>
<tr>
<td>pat41092</td>
<td>4.6909</td>
<td>2.3715</td>
<td>4.2641</td>
<td>2.0834</td>
<td>3.352475</td>
</tr>
<tr>
<td>pat4111</td>
<td>3.2222</td>
<td>1.6309</td>
<td>2.6909</td>
<td>2.9304</td>
<td>2.6186</td>
</tr>
<tr>
<td>pat41125</td>
<td>2.7715</td>
<td>1.3986</td>
<td>2.7382</td>
<td>3.0227</td>
<td>2.48275</td>
</tr>
<tr>
<td>pat41128</td>
<td>3.6295</td>
<td>2.0851</td>
<td>4.5029</td>
<td>4.2091</td>
<td>3.60665</td>
</tr>
<tr>
<td>pat41129</td>
<td>3.7006</td>
<td>1.9635</td>
<td>3.3491</td>
<td>2.6186</td>
<td>2.90795</td>
</tr>
<tr>
<td>pat41133</td>
<td>3.1782</td>
<td>1.6585</td>
<td>2.9477</td>
<td>1.8681</td>
<td>2.413125</td>
</tr>
<tr>
<td>pat41136</td>
<td>9.3482</td>
<td>5.354</td>
<td>9.9933</td>
<td>6.8731</td>
<td>7.89215</td>
</tr>
<tr>
<td>pat41138</td>
<td>3.1791</td>
<td>2.3491</td>
<td>3.5435</td>
<td>1.9541</td>
<td>2.75645</td>
</tr>
<tr>
<td>pat41122</td>
<td>9.1421</td>
<td>5.3354</td>
<td>9.8824</td>
<td>6.7643</td>
<td>7.87105</td>
</tr>
<tr>
<td>pat41126</td>
<td>3.2159</td>
<td>2.5051</td>
<td>3.5376</td>
<td>2.1654</td>
<td>2.856</td>
</tr>
<tr>
<td>pat41130</td>
<td>3.6946</td>
<td>2.0226</td>
<td>2.9565</td>
<td>2.3003</td>
<td>2.7435</td>
</tr>
<tr>
<td>pat41132</td>
<td>3.7536</td>
<td>2.1911</td>
<td>3.0659</td>
<td>2.2367</td>
<td>2.811825</td>
</tr>
<tr>
<td>pat41134</td>
<td>5.2229</td>
<td>3.2217</td>
<td>3.5134</td>
<td>4.146</td>
<td>4.026</td>
</tr>
<tr>
<td>pat41136</td>
<td>9.3787</td>
<td>5.463</td>
<td>10.0407</td>
<td>7.0469</td>
<td>7.982325</td>
</tr>
<tr>
<td>pat41134</td>
<td>3.2338</td>
<td>2.4527</td>
<td>3.4424</td>
<td>1.951</td>
<td>2.769975</td>
</tr>
<tr>
<td>pat41156</td>
<td>3.8117</td>
<td>2.309</td>
<td>3.1332</td>
<td>2.2622</td>
<td>2.879025</td>
</tr>
<tr>
<td>pat41166</td>
<td>2.964</td>
<td>1.8819</td>
<td>2.9675</td>
<td>3.4382</td>
<td>2.8129</td>
</tr>
<tr>
<td>pat41168</td>
<td>2.9932</td>
<td>1.9646</td>
<td>2.9358</td>
<td>3.1542</td>
<td>2.76195</td>
</tr>
<tr>
<td>pat41174</td>
<td>3.0625</td>
<td>2.1817</td>
<td>3.1867</td>
<td>3.4211</td>
<td>2.963</td>
</tr>
<tr>
<td>pat41138</td>
<td>5.0653</td>
<td>2.6596</td>
<td>5.1154</td>
<td>4.7822</td>
<td>4.405625</td>
</tr>
<tr>
<td>pat41192</td>
<td>3.2439</td>
<td>1.9794</td>
<td>2.9634</td>
<td>2.7021</td>
<td>2.7222</td>
</tr>
<tr>
<td>pat41194</td>
<td>9.4038</td>
<td>5.4968</td>
<td>9.9462</td>
<td>6.8201</td>
<td>7.916725</td>
</tr>
<tr>
<td>pat41198</td>
<td>3.2677</td>
<td>2.4297</td>
<td>3.4704</td>
<td>1.8168</td>
<td>2.74615</td>
</tr>
<tr>
<td>pat414</td>
<td>4.0633</td>
<td>2.5222</td>
<td>3.6063</td>
<td>3.858</td>
<td>3.51245</td>
</tr>
<tr>
<td>pat41410</td>
<td>3.6568</td>
<td>2.1941</td>
<td>3.1256</td>
<td>2.2366</td>
<td>2.803275</td>
</tr>
<tr>
<td>pat41446</td>
<td>2.979</td>
<td>1.9959</td>
<td>2.9498</td>
<td>3.1371</td>
<td>2.76545</td>
</tr>
<tr>
<td>pat4150</td>
<td>4.6396</td>
<td>3.3975</td>
<td>6.5853</td>
<td>5.6527</td>
<td>5.068775</td>
</tr>
<tr>
<td>pat41562</td>
<td>3.8823</td>
<td>2.5734</td>
<td>5.3866</td>
<td>4.1167</td>
<td>3.98975</td>
</tr>
<tr>
<td>pat41574</td>
<td>4.2854</td>
<td>2.2453</td>
<td>4.3444</td>
<td>3.9642</td>
<td>3.709825</td>
</tr>
<tr>
<td>pat41578</td>
<td>6.2465</td>
<td>4.1724</td>
<td>3.7635</td>
<td>5.4124</td>
<td>4.8987</td>
</tr>
<tr>
<td>pat41580</td>
<td>6.3928</td>
<td>3.5036</td>
<td>8.146</td>
<td>5.5012</td>
<td>5.8859</td>
</tr>
<tr>
<td>pat41610</td>
<td>3.8222</td>
<td>2.11</td>
<td>4.7023</td>
<td>4.3951</td>
<td>3.7574</td>
</tr>
<tr>
<td>pat41614</td>
<td>6.5485</td>
<td>3.9984</td>
<td>3.5638</td>
<td>4.2976</td>
<td>4.602075</td>
</tr>
<tr>
<td>pat41616</td>
<td>4.6731</td>
<td>3.1787</td>
<td>8.8931</td>
<td>7.9659</td>
<td>6.1777</td>
</tr>
<tr>
<td>pat41626</td>
<td>6.906</td>
<td>5.3998</td>
<td>4.6909</td>
<td>4.8135</td>
<td>5.45255</td>
</tr>
<tr>
<td>pat41632</td>
<td>4.33</td>
<td>2.9081</td>
<td>3.2899</td>
<td>3.6693</td>
<td>3.549325</td>
</tr>
<tr>
<td>pat41718</td>
<td>4.2046</td>
<td>2.6665</td>
<td>5.0839</td>
<td>4.2087</td>
<td>4.040925</td>
</tr>
<tr>
<td>pat41722</td>
<td>6.1345</td>
<td>3.4511</td>
<td>3.5983</td>
<td>4.7365</td>
<td>4.4801</td>
</tr>
<tr>
<td>pat41734</td>
<td>6.2166</td>
<td>4.5002</td>
<td>4.6524</td>
<td>5.8875</td>
<td>5.314175</td>
</tr>
<tr>
<td>pat41770</td>
<td>6.7206</td>
<td>4.3115</td>
<td>4.294</td>
<td>4.3619</td>
<td>4.922</td>
</tr>
<tr>
<td>pat4186</td>
<td>5.5441</td>
<td>3.5934</td>
<td>5.991</td>
<td>4.4504</td>
<td>4.894725</td>
</tr>
<tr>
<td>pat42042</td>
<td>4.002</td>
<td>2.6669</td>
<td>3.7345</td>
<td>3.6676</td>
<td>3.51775</td>
</tr>
<tr>
<td>pat42046</td>
<td>8.5812</td>
<td>3.73</td>
<td>3.6698</td>
<td>6.7028</td>
<td>5.67095</td>
</tr>
<tr>
<td>pat42048</td>
<td>4.2616</td>
<td>3.804</td>
<td>6.1468</td>
<td>4.3287</td>
<td>4.635275</td>
</tr>
<tr>
<td>pat42058</td>
<td>8.9777</td>
<td>5.4544</td>
<td>5.7478</td>
<td>8.993</td>
<td>7.932225</td>
</tr>
<tr>
<td>pat42060</td>
<td>4.8449</td>
<td>3.0763</td>
<td>4.9151</td>
<td>4.1894</td>
<td>4.256425</td>
</tr>
<tr>
<td>pat42064</td>
<td>5.4113</td>
<td>2.7574</td>
<td>3.344</td>
<td>4.9176</td>
<td>4.107575</td>
</tr>
<tr>
<td>pat42094</td>
<td>9.7151</td>
<td>5.1909</td>
<td>4.9399</td>
<td>5.4574</td>
<td>6.325825</td>
</tr>
<tr>
<td>pat42096</td>
<td>4.1186</td>
<td>3.0649</td>
<td>5.4544</td>
<td>5.0211</td>
<td>4.41475</td>
</tr>
<tr>
<td>pat42100</td>
<td>5.932</td>
<td>3.0433</td>
<td>3.3047</td>
<td>4.1722</td>
<td>4.11305</td>
</tr>
<tr>
<td>pat42112</td>
<td>6.1795</td>
<td>3.9132</td>
<td>4.2462</td>
<td>4.6781</td>
<td>4.75425</td>
</tr>
<tr>
<td>pat42202</td>
<td>8.7352</td>
<td>3.9544</td>
<td>4.9214</td>
<td>7.2428</td>
<td>6.21345</td>
</tr>
<tr>
<td>pat427</td>
<td>6.0923</td>
<td>2.1102</td>
<td>7.172</td>
<td>6.9118</td>
<td>5.571575</td>
</tr>
<tr>
<td>pat4294</td>
<td>4.8874</td>
<td>2.7787</td>
<td>5.6612</td>
<td>4.9673</td>
<td>4.57365</td>
</tr>
<tr>
<td>pat430</td>
<td>6.13</td>
<td>2.088</td>
<td>7.0508</td>
<td>6.8085</td>
<td>5.519325</td>
</tr>
<tr>
<td>pat43014</td>
<td>8.0224</td>
<td>2.5739</td>
<td>9.013</td>
<td>6.5941</td>
<td>6.55085</td>
</tr>
<tr>
<td>pat43018</td>
<td>3.6725</td>
<td>2.433</td>
<td>3.4006</td>
<td>2.4933</td>
<td>2.99985</td>
</tr>
<tr>
<td>pat43020</td>
<td>11.5592</td>
<td>5.5035</td>
<td>14.8823</td>
<td>8.782</td>
<td>10.18175</td>
</tr>
<tr>
<td>pat43030</td>
<td>4.1376</td>
<td>2.8906</td>
<td>3.4723</td>
<td>3.2321</td>
<td>3.43315</td>
</tr>
<tr>
<td>pat43032</td>
<td>13.0115</td>
<td>3.928</td>
<td>10.5232</td>
<td>7.7298</td>
<td>8.798125</td>
</tr>
<tr>
<td>pat43036</td>
<td>4.7624</td>
<td>2.7189</td>
<td>4.2322</td>
<td>2.0934</td>
<td>3.451725</td>
</tr>
<tr>
<td>pat43066</td>
<td>3.7157</td>
<td>2.4529</td>
<td>3.2684</td>
<td>3.4897</td>
<td>3.231675</td>
</tr>
<tr>
<td>pat43072</td>
<td>3.6682</td>
<td>2.3003</td>
<td>4.4738</td>
<td>4.2048</td>
<td>3.661775</td>
</tr>
<tr>
<td>pat43174</td>
<td>3.8776</td>
<td>2.5449</td>
<td>3.3939</td>
<td>2.67</td>
<td>3.1216</td>
</tr>
<tr>
<td>pat43498</td>
<td>4.5993</td>
<td>2.9709</td>
<td>3.6046</td>
<td>4.0129</td>
<td>3.796925</td>
</tr>
<tr>
<td>pat43500</td>
<td>9.3325</td>
<td>5.4761</td>
<td>9.9609</td>
<td>6.9171</td>
<td>7.92165</td>
</tr>
<tr>
<td>pat43504</td>
<td>3.6552</td>
<td>2.6031</td>
<td>3.5512</td>
<td>2.2588</td>
<td>3.017075</td>
</tr>
<tr>
<td>pat43516</td>
<td>4.1652</td>
<td>2.6748</td>
<td>3.4476</td>
<td>2.9405</td>
<td>3.307025</td>
</tr>
<tr>
<td>pat43552</td>
<td>3.5768</td>
<td>2.4279</td>
<td>3.2585</td>
<td>3.5378</td>
<td>3.20025</td>
</tr>
<tr>
<td>pat43746</td>
<td>3.2418</td>
<td>1.7787</td>
<td>3.5712</td>
<td>3.1548</td>
<td>2.936625</td>
</tr>
<tr>
<td>pat43748</td>
<td>3.04</td>
<td>1.7932</td>
<td>2.3322</td>
<td>2.7774</td>
<td>2.4857</td>
</tr>
<tr>
<td>pat43752</td>
<td>3.0739</td>
<td>2.0103</td>
<td>3.7457</td>
<td>3.516</td>
<td>3.086475</td>
</tr>
<tr>
<td>pat43756</td>
<td>6.2441</td>
<td>4.226</td>
<td>3.6779</td>
<td>5.6444</td>
<td>4.9481</td>
</tr>
<tr>
<td>pat43760</td>
<td>2.9711</td>
<td>2.0097</td>
<td>2.2874</td>
<td>3.141</td>
<td>2.6023</td>
</tr>
<tr>
<td>pat43762</td>
<td>6.1693</td>
<td>4.0984</td>
<td>3.535</td>
<td>5.5216</td>
<td>4.831075</td>
</tr>
<tr>
<td>pat43770</td>
<td>3.4725</td>
<td>2.0749</td>
<td>3.7635</td>
<td>2.812</td>
<td>3.030725</td>
</tr>
<tr>
<td>pat43774</td>
<td>6.5951</td>
<td>3.9669</td>
<td>3.3476</td>
<td>3.8943</td>
<td>4.450975</td>
</tr>
<tr>
<td>pat43786</td>
<td>7.0643</td>
<td>5.5337</td>
<td>4.683</td>
<td>4.7361</td>
<td>5.504275</td>
</tr>
<tr>
<td>pat43796</td>
<td>3.2525</td>
<td>2.0568</td>
<td>2.446</td>
<td>2.538</td>
<td>2.573325</td>
</tr>
<tr>
<td>pat43798</td>
<td>6.653</td>
<td>4.0875</td>
<td>3.3525</td>
<td>4.0747</td>
<td>4.541925</td>
</tr>
<tr>
<td>pat43804</td>
<td>6.9741</td>
<td>5.362</td>
<td>4.7229</td>
<td>4.7606</td>
<td>5.4549</td>
</tr>
<tr>
<td>pat43824</td>
<td>3.2372</td>
<td>1.8429</td>
<td>3.6871</td>
<td>3.2593</td>
<td>3.006625</td>
</tr>
<tr>
<td>pat43828</td>
<td>6.0542</td>
<td>3.3302</td>
<td>3.3411</td>
<td>4.8557</td>
<td>4.3953</td>
</tr>
<tr>
<td>pat43840</td>
<td>6.2392</td>
<td>4.5047</td>
<td>4.5565</td>
<td>6.0456</td>
<td>5.3365</td>
</tr>
<tr>
<td>pat43876</td>
<td>6.8306</td>
<td>4.3766</td>
<td>4.1534</td>
<td>4.2561</td>
<td>4.904175</td>
</tr>
<tr>
<td>pat43904</td>
<td>3.0844</td>
<td>1.8447</td>
<td>2.4444</td>
<td>2.7944</td>
<td>2.541975</td>
</tr>
<tr>
<td>pat43906</td>
<td>6.1803</td>
<td>3.3178</td>
<td>3.3341</td>
<td>4.8757</td>
<td>4.426975</td>
</tr>
<tr>
<td>pat43912</td>
<td>6.2231</td>
<td>4.538</td>
<td>4.6494</td>
<td>6.0924</td>
<td>5.375725</td>
</tr>
<tr>
<td>pat4392</td>
<td>3.0315</td>
<td>2.2093</td>
<td>3.1966</td>
<td>3.377</td>
<td>2.9536</td>
</tr>
<tr>
<td>pat43930</td>
<td>6.7335</td>
<td>4.2795</td>
<td>4.1236</td>
<td>4.1595</td>
<td>4.824025</td>
</tr>
<tr>
<td>pat4398</td>
<td>3.1618</td>
<td>1.9872</td>
<td>2.8161</td>
<td>3.0308</td>
<td>2.748975</td>
</tr>
<tr>
<td>pat43986</td>
<td>3.8041</td>
<td>1.9315</td>
<td>3.7436</td>
<td>3.6454</td>
<td>3.28115</td>
</tr>
<tr>
<td>pat43987</td>
<td>8.8323</td>
<td>3.7007</td>
<td>3.5412</td>
<td>6.9475</td>
<td>5.755425</td>
</tr>
<tr>
<td>pat43989</td>
<td>3.0689</td>
<td>1.5</td>
<td>3.4954</td>
<td>3.0951</td>
<td>2.78985</td>
</tr>
<tr>
<td>pat43990</td>
<td>8.6614</td>
<td>3.6915</td>
<td>3.6922</td>
<td>6.828</td>
<td>5.718275</td>
</tr>
<tr>
<td>pat4400</td>
<td>2.9635</td>
<td>1.9212</td>
<td>2.8056</td>
<td>3.2212</td>
<td>2.727875</td>
</tr>
<tr>
<td>pat44002</td>
<td>9.1263</td>
<td>5.2974</td>
<td>5.7152</td>
<td>9.0423</td>
<td>7.2953</td>
</tr>
<tr>
<td>pat44005</td>
<td>5.401</td>
<td>2.5428</td>
<td>3.1135</td>
<td>5.0989</td>
<td>4.03905</td>
</tr>
<tr>
<td>pat44038</td>
<td>9.6698</td>
<td>5.0119</td>
<td>4.9961</td>
<td>5.5007</td>
<td>6.294625</td>
</tr>
<tr>
<td>pat44041</td>
<td>5.7765</td>
<td>2.6209</td>
<td>2.956</td>
<td>3.8242</td>
<td>3.7944</td>
</tr>
<tr>
<td>pat4410</td>
<td>3.5878</td>
<td>2.299</td>
<td>3.1133</td>
<td>2.4953</td>
<td>2.87385</td>
</tr>
<tr>
<td>pat44146</td>
<td>8.7055</td>
<td>3.8999</td>
<td>4.94</td>
<td>7.3184</td>
<td>6.21595</td>
</tr>
<tr>
<td>pat44149</td>
<td>5.1827</td>
<td>2.0432</td>
<td>2.9177</td>
<td>4.4556</td>
<td>3.6498</td>
</tr>
<tr>
<td>pat4422</td>
<td>3.4212</td>
<td>2.3968</td>
<td>3.2005</td>
<td>2.1526</td>
<td>2.792775</td>
</tr>
<tr>
<td>pat44228</td>
<td>3.6972</td>
<td>2.0187</td>
<td>2.3647</td>
<td>3.323</td>
<td>2.8509</td>
</tr>
<tr>
<td>pat44230</td>
<td>8.7066</td>
<td>3.6826</td>
<td>3.5567</td>
<td>6.8731</td>
<td>5.70475</td>
</tr>
<tr>
<td>pat44236</td>
<td>9.1665</td>
<td>5.3756</td>
<td>5.6578</td>
<td>9.0525</td>
<td>7.3131</td>
</tr>
<tr>
<td>pat44254</td>
<td>9.7477</td>
<td>4.9181</td>
<td>4.8116</td>
<td>5.3726</td>
<td>6.2125</td>
</tr>
<tr>
<td>pat4426</td>
<td>7.5894</td>
<td>5.0677</td>
<td>7.1877</td>
<td>5.1661</td>
<td>6.252725</td>
</tr>
<tr>
<td>pat44308</td>
<td>8.8169</td>
<td>3.9195</td>
<td>4.8594</td>
<td>7.399</td>
<td>6.2487</td>
</tr>
<tr>
<td>pat4434</td>
<td>3.4896</td>
<td>2.0643</td>
<td>2.9275</td>
<td>2.6062</td>
<td>2.7719</td>
</tr>
<tr>
<td>pat4436</td>
<td>3.4689</td>
<td>2.06</td>
<td>2.7802</td>
<td>2.3987</td>
<td>2.67695</td>
</tr>
<tr>
<td>pat4440</td>
<td>3.2888</td>
<td>2.388</td>
<td>3.3702</td>
<td>2.5951</td>
<td>2.910525</td>
</tr>
<tr>
<td>pat4444</td>
<td>7.5262</td>
<td>4.9614</td>
<td>7.2633</td>
<td>5.0886</td>
<td>6.203125</td>
</tr>
<tr>
<td>pat44472</td>
<td>2.8346</td>
<td>1.4593</td>
<td>3.5816</td>
<td>2.8544</td>
<td>2.682475</td>
</tr>
<tr>
<td>pat44476</td>
<td>3.2488</td>
<td>2.158</td>
<td>2.4683</td>
<td>2.6809</td>
<td>2.639</td>
</tr>
<tr>
<td>pat4448</td>
<td>3.2546</td>
<td>2.3598</td>
<td>3.2374</td>
<td>2.0474</td>
<td>2.7248</td>
</tr>
<tr>
<td>pat44488</td>
<td>3.3923</td>
<td>2.7772</td>
<td>2.8827</td>
<td>3.3758</td>
<td>3.107</td>
</tr>
<tr>
<td>pat4450</td>
<td>7.5456</td>
<td>5.0149</td>
<td>7.2473</td>
<td>5.0507</td>
<td>6.214625</td>
</tr>
<tr>
<td>pat44524</td>
<td>3.6583</td>
<td>2.6243</td>
<td>2.7541</td>
<td>2.8682</td>
<td>2.976225</td>
</tr>
<tr>
<td>pat4458</td>
<td>4.1805</td>
<td>2.8382</td>
<td>3.6999</td>
<td>3.3469</td>
<td>3.516375</td>
</tr>
<tr>
<td>pat4462</td>
<td>10.5771</td>
<td>5.4989</td>
<td>8.6098</td>
<td>8.777</td>
<td>8.3657</td>
</tr>
<tr>
<td>pat44632</td>
<td>3.3977</td>
<td>2.3769</td>
<td>2.7458</td>
<td>2.9561</td>
<td>2.869125</td>
</tr>
<tr>
<td>pat4464</td>
<td>3.2974</td>
<td>2.0523</td>
<td>2.9876</td>
<td>3.083</td>
<td>2.855075</td>
</tr>
<tr>
<td>pat4476</td>
<td>3.0059</td>
<td>2.073</td>
<td>3.0063</td>
<td>3.0278</td>
<td>2.77825</td>
</tr>
<tr>
<td>pat4480</td>
<td>6.7876</td>
<td>3.9773</td>
<td>7.2334</td>
<td>6.988</td>
<td>6.246575</td>
</tr>
<tr>
<td>pat44956</td>
<td>4.5979</td>
<td>2.7785</td>
<td>3.2203</td>
<td>4.2701</td>
<td>3.7167</td>
</tr>
<tr>
<td>pat4510</td>
<td>10.771</td>
<td>6.7117</td>
<td>9.8387</td>
<td>5.8599</td>
<td>8.295325</td>
</tr>
<tr>
<td>pat4512</td>
<td>3.4981</td>
<td>2.1496</td>
<td>3.096</td>
<td>2.6415</td>
<td>2.8463</td>
</tr>
<tr>
<td>pat4516</td>
<td>7.6076</td>
<td>4.074</td>
<td>6.2518</td>
<td>4.6898</td>
<td>5.6558</td>
</tr>
<tr>
<td>pat45200</td>
<td>2.7121</td>
<td>1.4425</td>
<td>2.6645</td>
<td>2.3022</td>
<td>2.280325</td>
</tr>
<tr>
<td>pat45202</td>
<td>3.385</td>
<td>2.1014</td>
<td>2.8116</td>
<td>2.6524</td>
<td>2.7376</td>
</tr>
<tr>
<td>pat45208</td>
<td>3.389</td>
<td>2.7302</td>
<td>3.3725</td>
<td>3.2839</td>
<td>3.1939</td>
</tr>
<tr>
<td>pat45226</td>
<td>3.6095</td>
<td>2.5688</td>
<td>3.1151</td>
<td>2.6707</td>
<td>2.991025</td>
</tr>
<tr>
<td>pat4528</td>
<td>7.0493</td>
<td>5.1503</td>
<td>8.3165</td>
<td>5.1544</td>
<td>6.417625</td>
</tr>
<tr>
<td>pat45280</td>
<td>3.3903</td>
<td>2.2378</td>
<td>3.0483</td>
<td>2.7641</td>
<td>2.860125</td>
</tr>
<tr>
<td>pat4538</td>
<td>4.3271</td>
<td>2.8273</td>
<td>3.8303</td>
<td>3.6541</td>
<td>3.6597</td>
</tr>
<tr>
<td>pat4540</td>
<td>10.4129</td>
<td>5.3675</td>
<td>8.6347</td>
<td>8.7837</td>
<td>8.2997</td>
</tr>
<tr>
<td>pat4542</td>
<td>4.5773</td>
<td>2.6247</td>
<td>3.6617</td>
<td>4.0752</td>
<td>3.734725</td>
</tr>
<tr>
<td>pat45445</td>
<td>3.0669</td>
<td>1.6133</td>
<td>2.6518</td>
<td>2.4204</td>
<td>2.4381</td>
</tr>
<tr>
<td>pat4564</td>
<td>11.0118</td>
<td>6.8814</td>
<td>10.0127</td>
<td>6.1789</td>
<td>8.5212</td>
</tr>
<tr>
<td>pat45948</td>
<td>4.4015</td>
<td>2.3667</td>
<td>4.4974</td>
<td>4.0419</td>
<td>3.826875</td>
</tr>
<tr>
<td>pat45952</td>
<td>6.7272</td>
<td>4.1797</td>
<td>4.1518</td>
<td>5.8124</td>
<td>5.277325</td>
</tr>
<tr>
<td>pat45954</td>
<td>6.3065</td>
<td>3.5335</td>
<td>8.1101</td>
<td>5.4668</td>
<td>5.854225</td>
</tr>
<tr>
<td>pat45984</td>
<td>3.8831</td>
<td>2.1877</td>
<td>4.9506</td>
<td>4.4262</td>
<td>3.8619</td>
</tr>
<tr>
<td>pat45988</td>
<td>6.8922</td>
<td>4.2234</td>
<td>3.8445</td>
<td>4.3965</td>
<td>4.83915</td>
</tr>
<tr>
<td>pat45990</td>
<td>4.6492</td>
<td>3.1667</td>
<td>8.7466</td>
<td>7.8044</td>
<td>6.091725</td>
</tr>
<tr>
<td>pat46000</td>
<td>7.2733</td>
<td>5.6623</td>
<td>5.1946</td>
<td>5.0704</td>
<td>5.80015</td>
</tr>
<tr>
<td>pat46006</td>
<td>4.5282</td>
<td>3.0086</td>
<td>3.5124</td>
<td>3.7447</td>
<td>3.698475</td>
</tr>
<tr>
<td>pat46092</td>
<td>4.1376</td>
<td>2.5738</td>
<td>5.0577</td>
<td>4.0942</td>
<td>3.965825</td>
</tr>
<tr>
<td>pat46096</td>
<td>6.4513</td>
<td>3.6044</td>
<td>3.8892</td>
<td>5.0569</td>
<td>4.75045</td>
</tr>
<tr>
<td>pat46108</td>
<td>6.5893</td>
<td>4.7442</td>
<td>5.1842</td>
<td>6.3073</td>
<td>5.70625</td>
</tr>
<tr>
<td>pat46144</td>
<td>7.1807</td>
<td>4.5815</td>
<td>4.6963</td>
<td>4.5945</td>
<td>5.26325</td>
</tr>
<tr>
<td>pat4623</td>
<td>3.8061</td>
<td>1.9121</td>
<td>3.0007</td>
<td>3.1651</td>
<td>2.971</td>
</tr>
<tr>
<td>pat4626</td>
<td>4.011</td>
<td>2.0249</td>
<td>3.3428</td>
<td>3.6006</td>
<td>3.244825</td>
</tr>
<tr>
<td>pat463</td>
<td>6.9677</td>
<td>2.3435</td>
<td>6.2072</td>
<td>4.6137</td>
<td>5.033025</td>
</tr>
<tr>
<td>pat4638</td>
<td>3.7963</td>
<td>2.3003</td>
<td>3.8623</td>
<td>3.7796</td>
<td>3.43625</td>
</tr>
<tr>
<td>pat4641</td>
<td>2.9651</td>
<td>1.5579</td>
<td>2.6897</td>
<td>2.9249</td>
<td>2.5344</td>
</tr>
<tr>
<td>pat46416</td>
<td>4.1256</td>
<td>2.7143</td>
<td>3.8468</td>
<td>3.7152</td>
<td>3.600475</td>
</tr>
<tr>
<td>pat46422</td>
<td>4.3383</td>
<td>3.8308</td>
<td>6.2238</td>
<td>4.4278</td>
<td>4.705175</td>
</tr>
<tr>
<td>pat46432</td>
<td>9.3881</td>
<td>5.4431</td>
<td>6.2966</td>
<td>9.2318</td>
<td>7.5899</td>
</tr>
<tr>
<td>pat46434</td>
<td>4.85</td>
<td>3.1529</td>
<td>4.9655</td>
<td>4.2117</td>
<td>4.295025</td>
</tr>
<tr>
<td>pat46438</td>
<td>5.7574</td>
<td>2.8659</td>
<td>3.6742</td>
<td>5.3298</td>
<td>4.406825</td>
</tr>
<tr>
<td>pat46468</td>
<td>10.0819</td>
<td>5.3633</td>
<td>5.4579</td>
<td>5.6723</td>
<td>6.64385</td>
</tr>
<tr>
<td>pat46470</td>
<td>4.1658</td>
<td>3.0336</td>
<td>5.4649</td>
<td>4.8926</td>
<td>4.389225</td>
</tr>
<tr>
<td>pat46474</td>
<td>6.0376</td>
<td>3.065</td>
<td>3.5277</td>
<td>4.2616</td>
<td>4.222975</td>
</tr>
<tr>
<td>pat46486</td>
<td>6.4776</td>
<td>3.9924</td>
<td>4.6528</td>
<td>4.8788</td>
<td>5.0004</td>
</tr>
<tr>
<td>pat466</td>
<td>7.1038</td>
<td>2.4374</td>
<td>6.2997</td>
<td>4.9121</td>
<td>5.18825</td>
</tr>
<tr>
<td>pat4674</td>
<td>4.3063</td>
<td>2.2535</td>
<td>3.739</td>
<td>2.5349</td>
<td>3.208425</td>
</tr>
<tr>
<td>pat4675</td>
<td>10.29</td>
<td>4.127</td>
<td>7.8169</td>
<td>5.7643</td>
<td>6.99955</td>
</tr>
<tr>
<td>pat4677</td>
<td>3.3131</td>
<td>1.6362</td>
<td>2.7939</td>
<td>2.5961</td>
<td>2.584825</td>
</tr>
<tr>
<td>pat4678</td>
<td>10.576</td>
<td>4.3052</td>
<td>7.9323</td>
<td>5.9397</td>
<td>7.1883</td>
</tr>
<tr>
<td>pat4690</td>
<td>9.6977</td>
<td>5.6989</td>
<td>10.5637</td>
<td>6.6592</td>
<td>8.154875</td>
</tr>
<tr>
<td>pat4693</td>
<td>6.5779</td>
<td>2.9846</td>
<td>6.5681</td>
<td>4.9066</td>
<td>5.2593</td>
</tr>
<tr>
<td>pat47388</td>
<td>8.0303</td>
<td>2.6807</td>
<td>8.9245</td>
<td>6.5836</td>
<td>6.554775</td>
</tr>
<tr>
<td>pat47392</td>
<td>3.7622</td>
<td>2.4856</td>
<td>3.5503</td>
<td>2.6296</td>
<td>3.106925</td>
</tr>
<tr>
<td>pat47394</td>
<td>11.7795</td>
<td>5.4742</td>
<td>15.1284</td>
<td>8.781</td>
<td>10.290775</td>
</tr>
<tr>
<td>pat47404</td>
<td>4.2086</td>
<td>3.0551</td>
<td>3.73</td>
<td>3.3575</td>
<td>3.5878</td>
</tr>
<tr>
<td>pat47406</td>
<td>13.1004</td>
<td>3.9934</td>
<td>10.6908</td>
<td>7.7965</td>
<td>8.895275</td>
</tr>
<tr>
<td>pat47410</td>
<td>4.8013</td>
<td>2.754</td>
<td>4.372</td>
<td>2.1333</td>
<td>3.51515</td>
</tr>
<tr>
<td>pat47440</td>
<td>3.9106</td>
<td>2.6835</td>
<td>3.5216</td>
<td>3.6409</td>
<td>3.43915</td>
</tr>
<tr>
<td>pat47446</td>
<td>3.7313</td>
<td>2.3421</td>
<td>4.5229</td>
<td>4.2036</td>
<td>3.699975</td>
</tr>
<tr>
<td>pat47548</td>
<td>3.9972</td>
<td>2.6457</td>
<td>3.5745</td>
<td>2.7618</td>
<td>3.2448</td>
</tr>
<tr>
<td>pat4780</td>
<td>12.7741</td>
<td>5.2478</td>
<td>12.1866</td>
<td>11.2613</td>
<td>10.36745</td>
</tr>
<tr>
<td>pat4783</td>
<td>9.618</td>
<td>3.0195</td>
<td>8.0245</td>
<td>8.3682</td>
<td>7.25755</td>
</tr>
<tr>
<td>pat47872</td>
<td>4.8773</td>
<td>3.1414</td>
<td>3.9367</td>
<td>4.3028</td>
<td>4.06455</td>
</tr>
<tr>
<td>pat47874</td>
<td>9.4493</td>
<td>5.4744</td>
<td>10.0337</td>
<td>6.9259</td>
<td>7.970825</td>
</tr>
<tr>
<td>pat47878</td>
<td>3.6842</td>
<td>2.6589</td>
<td>3.6301</td>
<td>2.3069</td>
<td>3.070025</td>
</tr>
<tr>
<td>pat47890</td>
<td>4.2116</td>
<td>2.6862</td>
<td>3.5547</td>
<td>3.0136</td>
<td>3.366525</td>
</tr>
<tr>
<td>pat47926</td>
<td>3.6955</td>
<td>2.474</td>
<td>3.3895</td>
<td>3.5791</td>
<td>3.284525</td>
</tr>
<tr>
<td>pat4866</td>
<td>3.8666</td>
<td>2.0603</td>
<td>3.0827</td>
<td>3.2039</td>
<td>3.053375</td>
</tr>
<tr>
<td>pat4868</td>
<td>3.9909</td>
<td>1.8646</td>
<td>3.0749</td>
<td>3.4303</td>
<td>3.090175</td>
</tr>
<tr>
<td>pat4872</td>
<td>3.5481</td>
<td>2.4817</td>
<td>3.9009</td>
<td>3.7562</td>
<td>3.421725</td>
</tr>
<tr>
<td>pat4880</td>
<td>3.5881</td>
<td>2.26</td>
<td>3.9127</td>
<td>3.9662</td>
<td>3.43175</td>
</tr>
<tr>
<td>pat4890</td>
<td>4.2002</td>
<td>2.5317</td>
<td>3.6489</td>
<td>2.0933</td>
<td>3.105025</td>
</tr>
<tr>
<td>pat4894</td>
<td>10.4551</td>
<td>4.2226</td>
<td>7.7313</td>
<td>5.9337</td>
<td>7.085675</td>
</tr>
<tr>
<td>pat4906</td>
<td>9.5944</td>
<td>5.5552</td>
<td>10.599</td>
<td>6.5963</td>
<td>8.086225</td>
</tr>
<tr>
<td>pat4916</td>
<td>4.2563</td>
<td>2.2815</td>
<td>3.6207</td>
<td>2.1971</td>
<td>3.0889</td>
</tr>
<tr>
<td>pat4918</td>
<td>10.4365</td>
<td>4.2961</td>
<td>7.8795</td>
<td>5.8449</td>
<td>7.11425</td>
</tr>
<tr>
<td>pat4924</td>
<td>9.7343</td>
<td>5.886</td>
<td>10.7047</td>
<td>6.5594</td>
<td>8.2211</td>
</tr>
<tr>
<td>pat4942</td>
<td>13.0765</td>
<td>5.5842</td>
<td>12.214</td>
<td>11.3352</td>
<td>10.552475</td>
</tr>
<tr>
<td>pat4944</td>
<td>3.7799</td>
<td>2.0962</td>
<td>3.4037</td>
<td>3.2958</td>
<td>3.1439</td>
</tr>
<tr>
<td>pat4948</td>
<td>9.4441</td>
<td>2.9874</td>
<td>7.8823</td>
<td>8.44</td>
<td>7.18845</td>
</tr>
<tr>
<td>pat4960</td>
<td>8.4792</td>
<td>4.0081</td>
<td>10.7225</td>
<td>9.5019</td>
<td>8.177925</td>
</tr>
<tr>
<td>pat4996</td>
<td>9.7687</td>
<td>4.3038</td>
<td>9.1211</td>
<td>5.7699</td>
<td>7.240875</td>
</tr>
<tr>
<td>pat50304</td>
<td>4.1085</td>
<td>2.6297</td>
<td>3.8013</td>
<td>3.6939</td>
<td>3.55835</td>
</tr>
<tr>
<td>pat50308</td>
<td>9.4722</td>
<td>5.539</td>
<td>4.0684</td>
<td>7.0829</td>
<td>6.540625</td>
</tr>
<tr>
<td>pat50320</td>
<td>9.9583</td>
<td>7.876</td>
<td>6.145</td>
<td>8.8518</td>
<td>8.207775</td>
</tr>
<tr>
<td>pat50356</td>
<td>10.5668</td>
<td>7.2546</td>
<td>5.4236</td>
<td>5.5577</td>
<td>7.200675</td>
</tr>
<tr>
<td>pat504</td>
<td>3.0262</td>
<td>2.1525</td>
<td>3.1675</td>
<td>3.344</td>
<td>2.92255</td>
</tr>
<tr>
<td>pat50464</td>
<td>9.8373</td>
<td>6.0813</td>
<td>5.4434</td>
<td>7.622</td>
<td>7.246</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>pat510</td>
<td>3.104</td>
<td>1.8349</td>
<td>2.8388</td>
<td>3.005</td>
<td>2.695675</td>
</tr>
<tr>
<td>pat5112</td>
<td>2.7701</td>
<td>1.7773</td>
<td>2.5136</td>
<td>2.6047</td>
<td>2.416425</td>
</tr>
<tr>
<td>pat512</td>
<td>2.877</td>
<td>1.4885</td>
<td>2.8191</td>
<td>3.2042</td>
<td>2.5972</td>
</tr>
<tr>
<td>pat5124</td>
<td>2.8312</td>
<td>1.706</td>
<td>2.4077</td>
<td>2.5724</td>
<td>2.379325</td>
</tr>
<tr>
<td>pat5128</td>
<td>3.5389</td>
<td>2.706</td>
<td>4.2219</td>
<td>3.7881</td>
<td>3.563725</td>
</tr>
<tr>
<td>pat5160</td>
<td>2.8182</td>
<td>1.6194</td>
<td>2.4792</td>
<td>2.7021</td>
<td>2.404725</td>
</tr>
<tr>
<td>pat5164</td>
<td>4.0754</td>
<td>2.7472</td>
<td>3.8362</td>
<td>3.1335</td>
<td>3.448075</td>
</tr>
<tr>
<td>pat5176</td>
<td>3.8392</td>
<td>3.4918</td>
<td>4.9292</td>
<td>3.3882</td>
<td>3.9121</td>
</tr>
<tr>
<td>pat51760</td>
<td>5.1635</td>
<td>3.945</td>
<td>3.9668</td>
<td>4.276</td>
<td>4.337825</td>
</tr>
<tr>
<td>pat522</td>
<td>3.5496</td>
<td>2.421</td>
<td>3.0813</td>
<td>2.572</td>
<td>2.905975</td>
</tr>
<tr>
<td>pat52500</td>
<td>3.109</td>
<td>2.0961</td>
<td>3.7056</td>
<td>3.6016</td>
<td>3.128075</td>
</tr>
<tr>
<td>pat52504</td>
<td>6.5824</td>
<td>4.4431</td>
<td>4.0627</td>
<td>6.0358</td>
<td>5.281</td>
</tr>
<tr>
<td>pat52508</td>
<td>3.067</td>
<td>2.0785</td>
<td>2.4012</td>
<td>3.1682</td>
<td>2.678725</td>
</tr>
<tr>
<td>pat52510</td>
<td>6.645</td>
<td>4.4055</td>
<td>3.9343</td>
<td>6.0397</td>
<td>5.256125</td>
</tr>
<tr>
<td>pat52518</td>
<td>3.4779</td>
<td>2.091</td>
<td>3.6447</td>
<td>2.7274</td>
<td>2.98525</td>
</tr>
<tr>
<td>pat52522</td>
<td>6.8266</td>
<td>4.1513</td>
<td>3.5903</td>
<td>4.0491</td>
<td>4.654325</td>
</tr>
<tr>
<td>pat52534</td>
<td>7.3535</td>
<td>5.7561</td>
<td>5.0993</td>
<td>4.8738</td>
<td>5.770675</td>
</tr>
<tr>
<td>pat52544</td>
<td>3.799</td>
<td>2.11</td>
<td>2.5044</td>
<td>2.4701</td>
<td>2.6161</td>
</tr>
<tr>
<td>pat52546</td>
<td>7.0071</td>
<td>4.1549</td>
<td>3.6194</td>
<td>4.1946</td>
<td>4.744</td>
</tr>
<tr>
<td>pat52552</td>
<td>7.3169</td>
<td>5.6913</td>
<td>5.1195</td>
<td>4.9062</td>
<td>5.758475</td>
</tr>
<tr>
<td>pat52572</td>
<td>3.325</td>
<td>1.9344</td>
<td>3.6204</td>
<td>3.252</td>
<td>3.03295</td>
</tr>
<tr>
<td>pat52576</td>
<td>6.5445</td>
<td>3.5852</td>
<td>3.7122</td>
<td>5.2148</td>
<td>4.764175</td>
</tr>
<tr>
<td>pat52588</td>
<td>6.5056</td>
<td>4.7258</td>
<td>5.125</td>
<td>6.3313</td>
<td>5.671925</td>
</tr>
<tr>
<td>pat52624</td>
<td>7.0733</td>
<td>4.4399</td>
<td>4.4672</td>
<td>4.361</td>
<td>5.07635</td>
</tr>
<tr>
<td>pat52652</td>
<td>3.1452</td>
<td>1.9322</td>
<td>2.5499</td>
<td>2.9147</td>
<td>2.6355</td>
</tr>
<tr>
<td>pat52654</td>
<td>6.4831</td>
<td>3.4191</td>
<td>3.6334</td>
<td>5.1462</td>
<td>4.67045</td>
</tr>
<tr>
<td>pat5266</td>
<td>5.5625</td>
<td>4.205</td>
<td>6.5158</td>
<td>5.5517</td>
<td>5.45875</td>
</tr>
<tr>
<td>pat52660</td>
<td>6.5096</td>
<td>4.7013</td>
<td>5.117</td>
<td>6.4456</td>
<td>5.693375</td>
</tr>
<tr>
<td>pat52678</td>
<td>7.1553</td>
<td>4.4385</td>
<td>4.524</td>
<td>4.3178</td>
<td>5.1089</td>
</tr>
<tr>
<td>pat52734</td>
<td>3.9542</td>
<td>2.0652</td>
<td>3.6119</td>
<td>3.7181</td>
<td>3.33735</td>
</tr>
<tr>
<td>pat52737</td>
<td>3.1323</td>
<td>1.55</td>
<td>3.4147</td>
<td>3.1315</td>
<td>2.807125</td>
</tr>
<tr>
<td>pat52753</td>
<td>5.6003</td>
<td>2.6103</td>
<td>3.3891</td>
<td>5.4021</td>
<td>4.25045</td>
</tr>
<tr>
<td>pat52786</td>
<td>10.1362</td>
<td>5.1945</td>
<td>5.4181</td>
<td>5.6954</td>
<td>6.61105</td>
</tr>
<tr>
<td>pat52789</td>
<td>6.1033</td>
<td>2.7506</td>
<td>3.2001</td>
<td>3.9961</td>
<td>4.019275</td>
</tr>
<tr>
<td>pat52894</td>
<td>9.1577</td>
<td>3.9331</td>
<td>5.4815</td>
<td>7.7096</td>
<td>6.570475</td>
</tr>
<tr>
<td>pat52897</td>
<td>5.589</td>
<td>2.1527</td>
<td>3.1896</td>
<td>4.6809</td>
<td>3.93005</td>
</tr>
<tr>
<td>pat52976</td>
<td>3.8372</td>
<td>2.1193</td>
<td>2.4613</td>
<td>3.362</td>
<td>2.94496</td>
</tr>
<tr>
<td>pat52984</td>
<td>9.5747</td>
<td>5.5944</td>
<td>6.26</td>
<td>9.4811</td>
<td>7.72755</td>
</tr>
<tr>
<td>pat53002</td>
<td>10.221</td>
<td>5.2544</td>
<td>5.3767</td>
<td>5.6603</td>
<td>6.6281</td>
</tr>
<tr>
<td>pat53056</td>
<td>9.3584</td>
<td>4.0948</td>
<td>5.3804</td>
<td>7.9214</td>
<td>6.68875</td>
</tr>
<tr>
<td>pat53220</td>
<td>2.7812</td>
<td>1.5059</td>
<td>3.3857</td>
<td>2.8124</td>
<td>2.6213</td>
</tr>
<tr>
<td>pat53224</td>
<td>3.3684</td>
<td>2.2423</td>
<td>2.5717</td>
<td>2.7838</td>
<td>2.74155</td>
</tr>
<tr>
<td>pat53236</td>
<td>3.5009</td>
<td>2.9659</td>
<td>3.1283</td>
<td>3.5428</td>
<td>3.284475</td>
</tr>
<tr>
<td>pat53272</td>
<td>3.8269</td>
<td>2.8654</td>
<td>2.9885</td>
<td>2.9811</td>
<td>3.165475</td>
</tr>
<tr>
<td>pat53380</td>
<td>3.5083</td>
<td>2.4474</td>
<td>2.9106</td>
<td>3.0641</td>
<td>2.9826</td>
</tr>
<tr>
<td>pat534</td>
<td>3.4009</td>
<td>2.1782</td>
<td>3.1936</td>
<td>2.2055</td>
<td>2.74455</td>
</tr>
<tr>
<td>pat53704</td>
<td>4.8952</td>
<td>2.9788</td>
<td>3.5408</td>
<td>4.4413</td>
<td>3.964025</td>
</tr>
<tr>
<td>pat538</td>
<td>7.056</td>
<td>3.1734</td>
<td>7.3025</td>
<td>5.2182</td>
<td>5.687525</td>
</tr>
<tr>
<td>pat53948</td>
<td>2.6872</td>
<td>1.4855</td>
<td>2.6044</td>
<td>2.3633</td>
<td>2.2851</td>
</tr>
<tr>
<td>pat53950</td>
<td>3.4205</td>
<td>2.1743</td>
<td>2.8634</td>
<td>2.6927</td>
<td>2.787725</td>
</tr>
<tr>
<td>pat53956</td>
<td>3.4533</td>
<td>2.8983</td>
<td>3.5173</td>
<td>3.4392</td>
<td>3.327025</td>
</tr>
<tr>
<td>pat53974</td>
<td>3.8192</td>
<td>2.7564</td>
<td>3.2395</td>
<td>2.8168</td>
<td>3.157975</td>
</tr>
<tr>
<td>pat54</td>
<td>3.7669</td>
<td>2.2659</td>
<td>3.2129</td>
<td>2.931</td>
<td>3.044175</td>
</tr>
<tr>
<td>pat54028</td>
<td>3.52</td>
<td>2.3838</td>
<td>3.1846</td>
<td>2.9347</td>
<td>3.005775</td>
</tr>
<tr>
<td>pat54190</td>
<td>4.7371</td>
<td>2.8422</td>
<td>3.8549</td>
<td>4.3222</td>
<td>3.9391</td>
</tr>
<tr>
<td>pat54193</td>
<td>3.1456</td>
<td>1.6427</td>
<td>2.6723</td>
<td>2.5278</td>
<td>2.4971</td>
</tr>
<tr>
<td>pat546</td>
<td>3.475</td>
<td>2.0088</td>
<td>2.8933</td>
<td>2.5375</td>
<td>2.72865</td>
</tr>
<tr>
<td>pat54678</td>
<td>4.0786</td>
<td>2.5956</td>
<td>3.6624</td>
<td>3.6676</td>
<td>3.50105</td>
</tr>
<tr>
<td>pat54694</td>
<td>9.9323</td>
<td>7.8896</td>
<td>6.1728</td>
<td>8.923</td>
<td>8.229425</td>
</tr>
<tr>
<td>pat54730</td>
<td>10.571</td>
<td>7.3419</td>
<td>5.4032</td>
<td>5.578</td>
<td>7.223525</td>
</tr>
<tr>
<td>pat548</td>
<td>3.3239</td>
<td>1.6517</td>
<td>2.7586</td>
<td>2.4028</td>
<td>2.53425</td>
</tr>
<tr>
<td>pat54838</td>
<td>9.6548</td>
<td>5.9281</td>
<td>5.3358</td>
<td>7.5822</td>
<td>7.125225</td>
</tr>
<tr>
<td>pat552</td>
<td>3.1924</td>
<td>2.2467</td>
<td>3.3586</td>
<td>2.5677</td>
<td>2.84135</td>
</tr>
<tr>
<td>pat556</td>
<td>7.1772</td>
<td>3.2464</td>
<td>7.3522</td>
<td>5.224</td>
<td>5.74995</td>
</tr>
<tr>
<td>pat5592</td>
<td>2.8169</td>
<td>1.758</td>
<td>2.5397</td>
<td>2.5004</td>
<td>2.40375</td>
</tr>
<tr>
<td>pat5596</td>
<td>5.0395</td>
<td>2.4242</td>
<td>5.0449</td>
<td>4.9444</td>
<td>4.36325</td>
</tr>
<tr>
<td>pat560</td>
<td>3.0299</td>
<td>1.7887</td>
<td>3.1836</td>
<td>2.0404</td>
<td>2.51065</td>
</tr>
<tr>
<td>pat5608</td>
<td>4.3821</td>
<td>3.1947</td>
<td>6.6102</td>
<td>5.7651</td>
<td>4.988025</td>
</tr>
<tr>
<td>pat56134</td>
<td>5.1627</td>
<td>4.0466</td>
<td>3.932</td>
<td>4.3916</td>
<td>4.383225</td>
</tr>
<tr>
<td>pat562</td>
<td>7.1009</td>
<td>3.1702</td>
<td>7.2744</td>
<td>5.2217</td>
<td>5.6918</td>
</tr>
<tr>
<td>pat5644</td>
<td>5.5735</td>
<td>3.4124</td>
<td>6.0343</td>
<td>4.1393</td>
<td>4.789875</td>
</tr>
<tr>
<td>pat56864</td>
<td>4.0368</td>
<td>2.6872</td>
<td>2.4768</td>
<td>3.3998</td>
<td>3.15015</td>
</tr>
<tr>
<td>pat56866</td>
<td>9.6353</td>
<td>5.6374</td>
<td>3.8732</td>
<td>7.0958</td>
<td>6.560425</td>
</tr>
<tr>
<td>pat56872</td>
<td>9.9616</td>
<td>7.872</td>
<td>6.0447</td>
<td>9.0265</td>
<td>8.2262</td>
</tr>
<tr>
<td>pat56890</td>
<td>10.5114</td>
<td>7.093</td>
<td>5.2877</td>
<td>5.4404</td>
<td>7.083125</td>
</tr>
<tr>
<td>pat56944</td>
<td>9.8177</td>
<td>6.1764</td>
<td>5.3451</td>
<td>7.6526</td>
<td>7.24795</td>
</tr>
<tr>
<td>pat57106</td>
<td>13.0767</td>
<td>6.5914</td>
<td>6.0935</td>
<td>10.2031</td>
<td>8.991175</td>
</tr>
<tr>
<td>pat57109</td>
<td>8.2887</td>
<td>3.5879</td>
<td>3.403</td>
<td>6.4902</td>
<td>5.44245</td>
</tr>
<tr>
<td>pat57592</td>
<td>5.2293</td>
<td>4.0975</td>
<td>3.5717</td>
<td>4.4776</td>
<td>4.344025</td>
</tr>
<tr>
<td>pat576</td>
<td>3.3207</td>
<td>2.1347</td>
<td>2.9484</td>
<td>3.0955</td>
<td>2.875825</td>
</tr>
<tr>
<td>pat5838</td>
<td>4.7737</td>
<td>1.9875</td>
<td>3.2514</td>
<td>3.9228</td>
<td>3.48385</td>
</tr>
<tr>
<td>pat5840</td>
<td>2.7111</td>
<td>1.5987</td>
<td>2.3068</td>
<td>2.3879</td>
<td>2.251125</td>
</tr>
<tr>
<td>pat5844</td>
<td>4.564</td>
<td>2.0886</td>
<td>3.3744</td>
<td>4.0373</td>
<td>3.516075</td>
</tr>
<tr>
<td>pat5848</td>
<td>3.5046</td>
<td>2.8322</td>
<td>4.3531</td>
<td>3.745</td>
<td>3.608725</td>
</tr>
<tr>
<td>pat5852</td>
<td>2.9014</td>
<td>1.7291</td>
<td>2.5183</td>
<td>2.7893</td>
<td>2.484525</td>
</tr>
<tr>
<td>pat5854</td>
<td>3.7094</td>
<td>2.7815</td>
<td>4.2614</td>
<td>3.8075</td>
<td>3.63995</td>
</tr>
<tr>
<td>pat5862</td>
<td>4.8192</td>
<td>2.1033</td>
<td>3.3392</td>
<td>3.8627</td>
<td>3.5311</td>
</tr>
<tr>
<td>pat5866</td>
<td>4.1041</td>
<td>2.8384</td>
<td>3.9656</td>
<td>3.0402</td>
<td>3.487075</td>
</tr>
<tr>
<td>pat5878</td>
<td>3.9946</td>
<td>3.4578</td>
<td>4.902</td>
<td>3.4813</td>
<td>3.958925</td>
</tr>
<tr>
<td>pat588</td>
<td>2.987</td>
<td>1.8606</td>
<td>2.954</td>
<td>3.0345</td>
<td>2.709025</td>
</tr>
<tr>
<td>pat5888</td>
<td>2.7619</td>
<td>1.6018</td>
<td>2.3695</td>
<td>2.827</td>
<td>2.39005</td>
</tr>
<tr>
<td>pat5890</td>
<td>4.0305</td>
<td>2.6612</td>
<td>3.8542</td>
<td>3.3936</td>
<td>3.471375</td>
</tr>
<tr>
<td>pat5896</td>
<td>3.7511</td>
<td>3.3662</td>
<td>4.8199</td>
<td>3.7517</td>
<td>3.922225</td>
</tr>
<tr>
<td>pat5914</td>
<td>5.4568</td>
<td>4.1964</td>
<td>6.5554</td>
<td>5.5131</td>
<td>5.430425</td>
</tr>
<tr>
<td>pat5916</td>
<td>4.6403</td>
<td>1.9804</td>
<td>3.2702</td>
<td>3.937</td>
<td>3.456975</td>
</tr>
<tr>
<td>pat592</td>
<td>5.887</td>
<td>1.9668</td>
<td>6.8904</td>
<td>6.6493</td>
<td>5.348375</td>
</tr>
<tr>
<td>pat5920</td>
<td>3.7634</td>
<td>2.4253</td>
<td>3.8633</td>
<td>3.4153</td>
<td>3.366825</td>
</tr>
<tr>
<td>pat5932</td>
<td>3.4892</td>
<td>2.9073</td>
<td>4.6024</td>
<td>3.8454</td>
<td>3.711075</td>
</tr>
<tr>
<td>pat5968</td>
<td>3.9412</td>
<td>2.7847</td>
<td>4.2403</td>
<td>3.47</td>
<td>3.60905</td>
</tr>
<tr>
<td>pat6078</td>
<td>4.7571</td>
<td>2.0116</td>
<td>3.5347</td>
<td>4.219</td>
<td>3.6306</td>
</tr>
<tr>
<td>pat6079</td>
<td>4.9302</td>
<td>2.4901</td>
<td>5.0903</td>
<td>5.0469</td>
<td>4.389375</td>
</tr>
<tr>
<td>pat6081</td>
<td>4.7729</td>
<td>1.8819</td>
<td>3.2608</td>
<td>4.0223</td>
<td>3.484475</td>
</tr>
<tr>
<td>pat6082</td>
<td>5.0334</td>
<td>2.5337</td>
<td>5.1178</td>
<td>4.8552</td>
<td>4.385025</td>
</tr>
<tr>
<td>pat6094</td>
<td>4.5865</td>
<td>2.3662</td>
<td>6.6566</td>
<td>5.7679</td>
<td>5.0693</td>
</tr>
<tr>
<td>pat6097</td>
<td>3.3536</td>
<td>1.893</td>
<td>3.9366</td>
<td>3.6579</td>
<td>3.210275</td>
</tr>
<tr>
<td>pat6130</td>
<td>5.4584</td>
<td>3.2504</td>
<td>5.9567</td>
<td>4.4678</td>
<td>4.783325</td>
</tr>
<tr>
<td>pat6133</td>
<td>3.546</td>
<td>1.8265</td>
<td>3.4772</td>
<td>3.2201</td>
<td>3.01745</td>
</tr>
<tr>
<td>pat624</td>
<td>3.4046</td>
<td>2.0979</td>
<td>3.0529</td>
<td>2.5528</td>
<td>2.77705</td>
</tr>
<tr>
<td>pat628</td>
<td>6.9585</td>
<td>2.3175</td>
<td>6.1479</td>
<td>4.6777</td>
<td>5.0254</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>pat6320</td>
<td>2.8172</td>
<td>1.6481</td>
<td>2.6148</td>
<td>2.7694</td>
<td>2.462375</td>
</tr>
<tr>
<td>pat6322</td>
<td>4.9224</td>
<td>2.5224</td>
<td>5.122</td>
<td>4.9164</td>
<td>4.3708</td>
</tr>
<tr>
<td>pat6328</td>
<td>4.3949</td>
<td>3.4586</td>
<td>6.8582</td>
<td>5.9065</td>
<td>5.15455</td>
</tr>
<tr>
<td>pat6346</td>
<td>5.5282</td>
<td>3.5447</td>
<td>6.1355</td>
<td>4.2204</td>
<td>4.8572</td>
</tr>
<tr>
<td>pat640</td>
<td>6.4125</td>
<td>3.1851</td>
<td>8.1894</td>
<td>5.1357</td>
<td>5.730675</td>
</tr>
<tr>
<td>pat6400</td>
<td>4.6045</td>
<td>2.5854</td>
<td>5.0709</td>
<td>4.9709</td>
<td>4.467175</td>
</tr>
<tr>
<td>pat654</td>
<td>3.4052</td>
<td>2.0617</td>
<td>2.783</td>
<td>3.0077</td>
<td>2.8144</td>
</tr>
<tr>
<td>pat656</td>
<td>3.1662</td>
<td>1.5162</td>
<td>2.6974</td>
<td>2.9054</td>
<td>2.5713</td>
</tr>
<tr>
<td>pat6588</td>
<td>3.9885</td>
<td>2.615</td>
<td>3.5118</td>
<td>3.688</td>
<td>3.451075</td>
</tr>
<tr>
<td>pat660</td>
<td>2.9407</td>
<td>2.1232</td>
<td>3.1128</td>
<td>3.3166</td>
<td>2.873325</td>
</tr>
<tr>
<td>pat6624</td>
<td>4.0088</td>
<td>2.6123</td>
<td>3.7038</td>
<td>3.7067</td>
<td>3.5079</td>
</tr>
<tr>
<td>pat6636</td>
<td>3.9017</td>
<td>2.5421</td>
<td>3.4858</td>
<td>2.9035</td>
<td>3.286275</td>
</tr>
<tr>
<td>pat664</td>
<td>6.0445</td>
<td>2.1788</td>
<td>7.1183</td>
<td>6.8642</td>
<td>5.55145</td>
</tr>
<tr>
<td>pat6660</td>
<td>7.8723</td>
<td>5.1891</td>
<td>7.776</td>
<td>5.4397</td>
<td>6.596275</td>
</tr>
<tr>
<td>pat668</td>
<td>2.701</td>
<td>1.5019</td>
<td>3.0713</td>
<td>3.3031</td>
<td>2.644325</td>
</tr>
<tr>
<td>pat670</td>
<td>5.9882</td>
<td>2.0295</td>
<td>6.994</td>
<td>6.8396</td>
<td>5.462825</td>
</tr>
<tr>
<td>pat6726</td>
<td>4.4338</td>
<td>3.0015</td>
<td>3.9775</td>
<td>3.6392</td>
<td>3.763</td>
</tr>
<tr>
<td>pat6730</td>
<td>11.0234</td>
<td>5.6589</td>
<td>9.3809</td>
<td>9.0411</td>
<td>8.776075</td>
</tr>
<tr>
<td>pat6732</td>
<td>4.2126</td>
<td>2.9564</td>
<td>3.6713</td>
<td>3.8583</td>
<td>3.67465</td>
</tr>
<tr>
<td>pat6748</td>
<td>6.8857</td>
<td>4.0973</td>
<td>7.665</td>
<td>7</td>
<td>6.412</td>
</tr>
<tr>
<td>pat6778</td>
<td>10.8992</td>
<td>6.678</td>
<td>10.3844</td>
<td>5.9888</td>
<td>8.4906</td>
</tr>
<tr>
<td>pat6780</td>
<td>3.5417</td>
<td>2.3259</td>
<td>3.0442</td>
<td>2.4351</td>
<td>2.836725</td>
</tr>
<tr>
<td>pat6784</td>
<td>7.8549</td>
<td>4.3442</td>
<td>6.7063</td>
<td>5.0026</td>
<td>5.977</td>
</tr>
<tr>
<td>pat682</td>
<td>7.0343</td>
<td>2.4227</td>
<td>6.1609</td>
<td>4.6559</td>
<td>5.06845</td>
</tr>
<tr>
<td>pat694</td>
<td>6.366</td>
<td>3.1065</td>
<td>8.137</td>
<td>5.1195</td>
<td>5.68225</td>
</tr>
<tr>
<td>pat7056</td>
<td>4.2796</td>
<td>2.5438</td>
<td>3.5159</td>
<td>3.6176</td>
<td>3.492225</td>
</tr>
<tr>
<td>pat7068</td>
<td>3.9369</td>
<td>2.6693</td>
<td>4.0527</td>
<td>3.8294</td>
<td>3.622075</td>
</tr>
<tr>
<td>pat7072</td>
<td>9.8225</td>
<td>4.2459</td>
<td>10.0166</td>
<td>10.0164</td>
<td>8.515275</td>
</tr>
<tr>
<td>pat7104</td>
<td>4.579</td>
<td>2.8507</td>
<td>3.9656</td>
<td>2.4935</td>
<td>3.4722</td>
</tr>
<tr>
<td>pat7108</td>
<td>10.62</td>
<td>4.2427</td>
<td>8.4322</td>
<td>6.048</td>
<td>7.335725</td>
</tr>
<tr>
<td>pat7120</td>
<td>9.9701</td>
<td>5.8716</td>
<td>11.3949</td>
<td>6.7586</td>
<td>8.4988</td>
</tr>
<tr>
<td>pat72</td>
<td>3.1754</td>
<td>2.2288</td>
<td>3.4604</td>
<td>3.8354</td>
<td>3.175</td>
</tr>
<tr>
<td>pat7210</td>
<td>13.15</td>
<td>5.6521</td>
<td>12.8406</td>
<td>11.4566</td>
<td>10.774825</td>
</tr>
<tr>
<td>pat7212</td>
<td>4.1236</td>
<td>2.5321</td>
<td>3.6835</td>
<td>3.561</td>
<td>3.47505</td>
</tr>
<tr>
<td>pat7216</td>
<td>9.7017</td>
<td>3.0073</td>
<td>8.4077</td>
<td>8.5949</td>
<td>7.4279</td>
</tr>
<tr>
<td>pat7228</td>
<td>8.5784</td>
<td>3.9564</td>
<td>11.2714</td>
<td>9.8225</td>
<td>8.407175</td>
</tr>
<tr>
<td>pat756</td>
<td>3.6629</td>
<td>2.0988</td>
<td>2.8708</td>
<td>3.1094</td>
<td>2.935475</td>
</tr>
<tr>
<td>pat78</td>
<td>3.4421</td>
<td>1.8432</td>
<td>2.8027</td>
<td>3.0462</td>
<td>2.78365</td>
</tr>
<tr>
<td>pat792</td>
<td>3.1187</td>
<td>1.932</td>
<td>2.9855</td>
<td>4.0165</td>
<td>3.013175</td>
</tr>
<tr>
<td>pat80</td>
<td>2.7834</td>
<td>1.7213</td>
<td>2.6495</td>
<td>2.2968</td>
<td>2.36275</td>
</tr>
<tr>
<td>pat8028</td>
<td>7.6769</td>
<td>3.8743</td>
<td>5.4541</td>
<td>5.9608</td>
<td>5.741525</td>
</tr>
<tr>
<td>pat804</td>
<td>3.3347</td>
<td>1.6755</td>
<td>2.4616</td>
<td>3.1263</td>
<td>2.649525</td>
</tr>
<tr>
<td>pat8040</td>
<td>7.3833</td>
<td>2.9023</td>
<td>3.6344</td>
<td>5.0713</td>
<td>4.747825</td>
</tr>
<tr>
<td>pat8044</td>
<td>4.154</td>
<td>3.0735</td>
<td>4.5638</td>
<td>3.8214</td>
<td>3.903175</td>
</tr>
<tr>
<td>pat8076</td>
<td>6.6787</td>
<td>2.5138</td>
<td>3.9109</td>
<td>7.34</td>
<td>5.11085</td>
</tr>
<tr>
<td>pat808</td>
<td>2.9929</td>
<td>2.1295</td>
<td>3.3387</td>
<td>2.7</td>
<td>2.790275</td>
</tr>
<tr>
<td>pat8080</td>
<td>4.4006</td>
<td>3.0016</td>
<td>4.2706</td>
<td>3.8143</td>
<td>3.871775</td>
</tr>
<tr>
<td>pat8092</td>
<td>4.3204</td>
<td>3.579</td>
<td>5.0403</td>
<td>3.89</td>
<td>4.207425</td>
</tr>
<tr>
<td>pat8182</td>
<td>5.7264</td>
<td>4.1242</td>
<td>6.9375</td>
<td>5.6941</td>
<td>5.6926</td>
</tr>
<tr>
<td>pat8184</td>
<td>7.7851</td>
<td>3.6416</td>
<td>3.9263</td>
<td>5.6279</td>
<td>5.245225</td>
</tr>
<tr>
<td>pat8188</td>
<td>4.3601</td>
<td>2.7873</td>
<td>4.1618</td>
<td>3.4464</td>
<td>3.6889</td>
</tr>
<tr>
<td>pat8508</td>
<td>5.1957</td>
<td>3.2504</td>
<td>3.5379</td>
<td>4.1631</td>
<td>4.036775</td>
</tr>
<tr>
<td>pat8512</td>
<td>5.2784</td>
<td>2.851</td>
<td>5.5432</td>
<td>5.092</td>
<td>4.691115</td>
</tr>
<tr>
<td>pat8524</td>
<td>4.6827</td>
<td>3.4891</td>
<td>7.0319</td>
<td>5.9039</td>
<td>5.2769</td>
</tr>
<tr>
<td></td>
<td>pat8560</td>
<td>pat8668</td>
<td>pat900</td>
<td>pat916</td>
<td>pat952</td>
</tr>
<tr>
<td>----</td>
<td>---------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>5.6956</td>
<td>4.9297</td>
<td>3.7486</td>
<td>2.8527</td>
<td>3.1052</td>
</tr>
<tr>
<td>2</td>
<td>3.6628</td>
<td>2.879</td>
<td>2.6658</td>
<td>1.9411</td>
<td>1.9917</td>
</tr>
<tr>
<td>4</td>
<td>4.5749</td>
<td>5.0284</td>
<td>3.4618</td>
<td>2.7832</td>
<td>2.8312</td>
</tr>
<tr>
<td>5</td>
<td>5.094625</td>
<td>4.712675</td>
<td>3.268575</td>
<td>2.709425</td>
<td>2.7353</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


[Hassan] Hassan Gomaa, Designing Concurrent, Distributed, and Real Time Applications with UML.


THIS PAGE INTENTIONALLY LEFT BLANK
INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Tan Lai Poh
   Temasek Defense System Institute
   Singapore

4. Seah Siew Hwee, Human Resource Department
   Defence Science Technology Agency
   Singapore