Real-Time Data Warehousing and On-Line Analytical Processing at Aberdeen Test Center's Distributed Center

Mr. Michael J Reil  
SFA, Inc

Dr. Samuel F Harley  
US Army Aberdeen Test Center

Mr. T. George Bartlett  
US Army Aberdeen Test Center

UGC 2005
**Report Documentation Page**

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td></td>
<td>00-00-2005 to 00-00-2005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time Data Warehousing and On-Line Analytical Processing at Aberdeen Test Center’s Distributed Center</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5b. GRANT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5e. TASK NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Army Aberdeen Test Center, Aberdeen Proving Ground, MD, 21005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling and Simulation Conference, 2005 Dec 12-15, Las Cruces, NM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same as Report (SAR)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std Z39-18
ATC Distributed Center

Purpose

1. Provide real-time test data verification, analysis and warehousing
2. Provide OLAP tools for test data analysis and data mining

- 32 node Linux cluster (64 processors)
- 4.4 TB Panasas storage
- Hosted at ARL-MSRC – leverages existing support

UGC 2005
ATC DC Proposal

• Achieve real-time data fusion to provide real-time analytic and decision support

• Establish parallel post processing capabilities to effect knowledge extraction

• Institute a high performance data warehouse

• Real time quality control – utilizing historic data sets
ATC DC Timeline

- Oct-2003 - Proposal selected
- 4-May-2004 - System Delivered
- **28-June-2004 - System on network accepting connections**
- July-2004 - System Testing Complete
- Sept-2004 - Current data handling process (SunE10K) ported to DC
- Sept-2004 - Kerberized filters in place to allow web access to data warehouse (ARL-PET Dr. Walter Landry)

**Nov-2004 – OS Change from RHES to SuSE ES9 – Slave node NFS issues**
- Dec-2004 - Processing apps running with mpiJava
- Dec-2004 - Tomcat running in a JavaParty environment
- Nov/Dec-2004 - Army Science Conference demo of Data Warehouse
- Feb-2005 - Processing apps running with Javaparty
- April-2005 - Automated scripts to poll ATC concentrator for new data files
ATC DC Proposal

• Achieve real-time data fusion to provide real-time analytic and decision support

• Establish parallel post processing capabilities to effect knowledge extraction

• Institute a high performance data warehouse

• Real time quality control – utilizing historic data sets
Real Time Data Fusion

- Test data collected via on-board instrumentation - VxWorks based computer. Each instrument produces a continuous time history record of up to 250 parameters, up to 10KHz ea. Files closed approx. every 15-30 minutes. Single file size from 10KB to 100MB. Test item may have multiple instruments recording simultaneously.

- Must move raw data files from instrumentation to cluster for processing. Wireless or PC-Card harvesting.

- When raw data files show up on cluster – Java based conversion (raw to HDF5) process must fire automatically.

- Report applications fire, creating reports (PDF, Excel etc.) on the just processed data.

- Reports auto-published to web based Digital Library for consumption by decision makers.

- HDF5 data files registered in data warehouse.
Real Time Data Fusion (cont)

‘while(true)’ bash script to scp data files from concentrator to DC. ATC firewall allows DC ssh traffic into concentrator. SSH keys used to allow password-less (and unattended) ssh commands.

AntFlow used to start processing scripts

Polling required, as automated file transfer not achievable – increases latency and complexity

< 10 minutes
< 2 minutes
< 5 minutes
< 2 minutes

10-100Mb/Sec
100Mb/Sec

Data File Concentrator (Linux)

Aberdeen Test Center Distributed Center Linux Cluster @ARL-MSRC

UGC 2005
Real Time Data Fusion (cont)

Concentrator

Tar file(s) of gzip'ed blobs

/home/atcdata/HARVESTED_DATA/<unique>

Single text file containing name and location of tar file(s)

/ usr/people/mrel/PROCESSES/MonitorEUDB.bash

Never ending Bash script - Every 10 seconds, checks hot folder on concentrator (via passwordless ssh). When file found, scp's the corresponding tar files from concentrator to a temp dir, then creates a new hot file, and drops it into the hot folder on fasig

Tar file(s) of gzip'ed blobs

Single text file containing name and location of tar file(s)

/ usr/people/mrel/AntFlow/...

AntFlow task - When new file appears, unzips the corresponding tar file(s) into the /data/BLOBS/NEW_ANT directory, removes the tar file(s), the tar directory and the text file, gunzip's the blobs then runs the blob processing script. This task must ensure that there is only one instance of itself running at any time (crude job scheduling)

Single directory of new blobs

/data/BLOBS/NEW_ANT

Blob Processing Script

/ usr/people/mrel/PROCESSES/...
ATC DC Proposal

• Achieve real-time data fusion to provide real-time analytic and decision support

• Establish parallel post processing capabilities to effect knowledge extraction

• Institute a high performance data warehouse

• Real time quality control – utilizing historic data sets
Establish post processing capabilities to effect knowledge extraction

- Raw data files are converted to a common format – HDF5 chosen. ([http://hdf.ncsa.uiuc.edu/HDF5](http://hdf.ncsa.uiuc.edu/HDF5))

- Existing library of java classes and *nix scripts to convert raw data files to HDF5. Originally single threaded java code, extended to utilize multiple java threads. Worked well on SMP machines (Sun E10K), but not on distributed processor/memory systems (Linux cluster). Processing is easy to parallelize. Each thread gets one data file to convert. Java classes used lots of memory – object oriented nature of code contributed to this – each data point was a java object. Garbage collection times also large.

- mpiJava – thin java wrapper around MPICH. Created java app that distributed processing of data files via message passing (MPI). Worked well, but required knowledge of the MPI framework and library. Also dependent on availability of MPICH for your OS/disto.
Establish post processing capabilities to effect knowledge extraction (cont)

- JavaParty – [http://www.ipd.uka.de/JavaParty/features.html](http://www.ipd.uka.de/JavaParty/features.html) - *allows easy port of multi-threaded Java programs to distributed environments such as clusters. Regular Java already supports parallel applications with threads and synchronization mechanisms. While multi-threaded Java programs are limited to a single address space, JavaParty extends the capabilities of Java to distributed computing environments.*

*From the JavaParty Web Site*
public class ConvertToHDF5 extends Thread {
    ...
    ...
    ConvertToHDF5 worker = new ConvertToHDF5(...);
    worker.start();
    ...
    ...
    ...

    public void run() {
        ...
        ...
    }
}

Each thread is mapped to a physical processor by the JVM (Java Virtual Machine) – for SMP machines only!
mpiJava

Works well on SMP or Distributed machines – it is just MPI!
public remote class HelloJP {
    public void hello() {
        // Print on the console of the virtual machine where the object lives
        System.out.println("Hello JavaParty!");
    }
    public static void main(String[] args) {
        for (int n = 0; n < 10; n++) {
            // Create a remote object
            HelloJP world = new HelloJP();
            // Remotely invoke a method
            world.hello();
        }
    }
}

Each new ‘remote’ object is created on a slave processor. User can control which processor with the /** @i */ construct in code.
JavaParty

• Uses ssh to spawn JVMs on slave nodes of cluster (similar to MPI)
• One JVM per slave processor.
• Controlled via .jp-nodefile (similar to ‘machines’ file used with MPI).
• Pure java implementation – no native libraries required.
• Uses RMI to serialize java objects between JVMs.
• High performance RMI engine supplied (KaRMI).
• Possible to use without ‘breaking’ java source code – extend ‘RemoteThread’ class instead of using ‘remote’ keyword.

• This is the framework that we are now using.

• Regular java – invoke application:
  • java <classname>

• JavaParty – invoke application:
  • jpinvite <classname>
Establish post processing capabilities to effect knowledge extraction (cont)

Data Ingestion

Histogram Processing

Number of Processors

Number of Files

Time to Compute (sec)

Time (minutes)

Single Processor
Cluster with 32 Processors
Cluster with 64 Processors

UGC 2005
ATC DC Proposal

• Achieve real-time data fusion to provide real-time analytic and decision support

• Establish parallel post processing capabilities to effect knowledge extraction

• Institute a high performance data warehouse

• Real time quality control – utilizing historic data sets
What is OLAP?

• Online Analytical Processing
• Software that enables decision support via rapid queries to large databases that store corporate data in multidimensional hierarchies and views.
Institute a high performance data warehouse

• PostgreSQL 7.4 installed on dedicated filesystem (500 MB RAID5 JBOD) on head node.

• Java based web application ported to JavaParty. Allows data set queries submitted by the web app user to be run on all nodes of the cluster in parallel (for aggregate operations). Tomcat started via ‘javaparty’ rather than the standard ‘java’. This allows servlets to create remote objects, which run on the remote nodes.

• Kerberos/SecureID authentication module written by PET IMT – Dr. Walter Landry @ ARL. Uses J2EE servlet filter framework and cookies to authenticate each HTTP request.

• GUI is java applet, which runs in users browser. GUI presents metadata to user, who selects filter settings, and applet then submits SQL statement on users behalf to data warehouse. List of data sets is returned – user can then request composite routines be run on the set of data files – these are run on the entire cluster in the JavaParty environment.
Institute a high performance data warehouse

- Filter re-directs any incoming HTTP request that does not have the proper credentials to a login page.
- User supplies HPCMP credentials, filter performs kinit, attaches encrypted token to HTTP response.
- Each subsequent HTTP request goes through the filter performed by container, checks for this token. If found, filter lets request pass, else redirects to login page.
Institute a high performance data warehouse
Screenshots Of OLAP GUI
UGC 2005
<table>
<thead>
<tr>
<th>Label</th>
<th>Units</th>
<th>UUID</th>
<th>AVG</th>
<th>MAX</th>
<th>MIN</th>
<th>NUMPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Side bottom frame</td>
<td>microstrain</td>
<td>1E7297D56304DE02A2F5-6.099</td>
<td>74.2888</td>
<td>-136.4038</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Axle=2 Vertical</td>
<td>g's</td>
<td>222508DF87419A65F-0.0179</td>
<td>1.8136</td>
<td>-2.4805</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Right Front Accel V</td>
<td>g's</td>
<td>269F7925468030000000</td>
<td>2.8533</td>
<td>-2.4897</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Axle=2 Long Knuckle</td>
<td>microstrain</td>
<td>35DBB5CAB7485242A65E1-0.0595</td>
<td>162.5941</td>
<td>-34.6593</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>JT708 Road Speed</td>
<td>MPH</td>
<td>3852DCC23CA9F3F4F0B3123706</td>
<td>40.5</td>
<td>0</td>
<td>100124</td>
<td></td>
</tr>
<tr>
<td>Axle=3 Long Knuckle</td>
<td>microstrain</td>
<td>2DCCB87F1904B7590142</td>
<td>114.5358</td>
<td>-88.2665</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Hydraulic Oil Temp</td>
<td>degrees</td>
<td>6BA1F8B913A2432AC5092332</td>
<td>60.0461</td>
<td>59.5304</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Axle=2 Displacement</td>
<td>inches</td>
<td>6BEDF00240C9F4B12833-0.4578</td>
<td>1.1474</td>
<td>-1.3408</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Right Side fish plate</td>
<td>transition</td>
<td>8CD1258E7A9F2438B8A4</td>
<td>47.4641</td>
<td>-267.0275</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Forward fish plate on Top</td>
<td>microstrain</td>
<td>A1BF5CC519A4A2B812C0275124</td>
<td>288.4886</td>
<td>-81.7579</td>
<td>180124</td>
<td></td>
</tr>
<tr>
<td>Bottom Frame front Rear</td>
<td>transition</td>
<td>A4AD8D95C67E4A3F3AA1575258</td>
<td>317.5180</td>
<td>-5.2526</td>
<td>180124</td>
<td></td>
</tr>
</tbody>
</table>

UGC 2005
Results compiled on DC and presented to analyst in under 1 second!
Results computed on DC and presented to analyst in under 1 second!
Results computed on DC and presented to analyst in under 1 second!
Institute a high performance data warehouse

Over 80 projects using Data Warehouse

Data Warehouse Volume By Project - March 4, 2005

PROJECT

- Zipped BLOB (260GB)
- HDF (570GB)

GB

0.00
20.00
40.00
60.00
80.00
100.00
120.00
140.00
160.00

Projects: Stryker APG RAM, EFV Camp Pend., FMTV, PVT, M1A2 SEP, FMTV, Dump Truck, M2K3 SW Upgrade, FMTV A1 PVT (IDM167), AA-AV P3x River, FMTV Handling System, HMMWV M1685 IOT, N2A3 BFST, PVT LT T#191, LOSAT, Mobility, Durability, ARL, Aramco, SOCEM M1114, TFFT (#186), O’Gara, Head Armor, SOCOM M999, Leaf Spring, M1082, Leaf Spring, FMTV, Light Armor, HEMMA Armor, FMTV, Acoustic Noise, HEMMA1 Armor, M1A2 LV 100, Engine Kit, HEMMA Armor, ARL, Aramco, Misc Projects.
ATC DC Proposal

• Achieve real-time data fusion to provide real-time analytic and decision support

• Establish parallel post processing capabilities to effect knowledge extraction

• Institute a high performance data warehouse

• Real time quality control – utilizing historic data sets
Real time quality control – utilizing historic data sets

• New data sets compared with warehoused data from the same channel/test item for anomaly detection.

• Future Work
Summary

• Parallel java applications are running very well on cluster.
• Polling vs. interrupt (event) driven processing not ideal – but workable.
• ARL MSRC administering the system is ideal.
• Data warehouse access requiring kerberos/secureID does not fit well with our current Digital Library project based authentication. ATC customers must obtain HPCMP account in order to use data warehouse (they don’t even know they are using HPCMP assets).
• Special thanks to Tom Kendall, Chris Slaughter and Ryan Baxter at ARL-MSRC for assistance every step of the way!
Partnering For Success

UGC 2005