HPCS HPCchallenge Benchmark Suite

David Koester
The MITRE Corporation
Email Address: dkoester@mitre.org

Jack Dongarra and Piotr Luszczek
ICL/UT
Email Addresses: {dongarra, luszczek}@cs.utk.edu

Abstract
The Defense Advanced Research Projects Agency (DARPA) High Productivity Computing Systems (HPCS) HPCchallenge Benchmarks examine the performance of High Performance Computing (HPC) architectures using kernels with more challenging memory access patterns than just the High Performance LINPACK (HPL) benchmark used in the Top500 list. The HPCchallenge Benchmarks build on the HPL framework and augment the Top500 list by providing benchmarks that bound the performance of many real applications as a function of memory access locality characteristics. The real utility of the HPCchallenge benchmarks are that architectures can be described with a wider range of metrics than just Flop/s from HPL. Even a small percentage of random memory accesses in real applications can significantly affect the overall performance of that application on architectures not designed to minimize or hide memory latency. The HPCchallenge Benchmarks includes a new metric — Giga UPdates per Second — and a new benchmark — RandomAccess — to measure the ability of an architecture to access memory randomly, i.e., with no locality. When looking only at HPL performance and the Top500 List, inexpensive build-your-own clusters appear to be much more cost effective than more sophisticated HPC architectures. HPCchallenge Benchmarks provide users with additional information to justify policy and purchasing decisions. We will compare the measured HPCchallenge Benchmark performance on various HPC architectures — from Cray X1s to Beowulf clusters — in the presentation and paper. Additional information on the HPCchallenge Benchmarks can be found at http://icl.cs.utk.edu/hpcc/

Introduction
At SC2003 in Phoenix (15-21 November 2003), Jack Dongarra (ICL/UT) announced the release of a new benchmark suite — the HPCchallenge Benchmarks — that examine the performance of HPC architectures using kernels with more challenging memory access patterns than High Performance Linpack (HPL) used in the Top500 list. The HPCchallenge Benchmarks are being designed to complement the Top500 list and provide benchmarks that bound the performance of many real applications as a function of memory access characteristics — e.g., spatial and temporal locality. Development of the HPCchallenge Benchmarks is being funded by the Defense Advanced Research Projects Agency (DARPA) High Productivity Computing Systems (HPCS) Program.

The HPCchallenge Benchmark Kernels

<table>
<thead>
<tr>
<th>Local</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGEMM (matrix x matrix multiply)</td>
<td>High Performance LINPACK (HPL)</td>
</tr>
<tr>
<td>STREAM</td>
<td>PTRANS — parallel matrix transpose</td>
</tr>
<tr>
<td>• COPY</td>
<td></td>
</tr>
<tr>
<td>• SCALE</td>
<td></td>
</tr>
<tr>
<td>• ADD</td>
<td></td>
</tr>
<tr>
<td>• TRIADD</td>
<td></td>
</tr>
<tr>
<td>RandomAccess</td>
<td>(MPI)RandomAccess</td>
</tr>
<tr>
<td>1D FFT</td>
<td>1D FFT</td>
</tr>
</tbody>
</table>

I/O

b_eff — effective bandwidth benchmark
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 FEB 2005</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPCS HPCchallenge Benchmark Suite</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>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>The MITRE Corporation</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>

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release, distribution unlimited

**13. SUPPLEMENTARY NOTES**
See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.

**14. ABSTRACT**

**15. SUBJECT TERMS**

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
<td>unclassified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UU</td>
<td>28</td>
<td></td>
</tr>
</tbody>
</table>
Additional information on the HPCchallenge Benchmarks can be found at http://icl.cs.utk.edu/hpcc/.

**Flop/s**

The Flop/s metric from HPL has been the de facto standard for comparing High Performance Computers for many years. HPL works well on all architectures — even cache-based, distributed memory multiprocessors — and the measured performance may not be representative of a wide range of real user applications like adaptive multi-physics simulations used in weapons and vehicle design and weather, climate models, and defense applications. HPL is more compute friendly than these applications because it has more extensive memory reuse in the Level 3 BLAS-based calculations.

**Memory Performance**

There is a need for benchmarks that test memory performance. When looking only at HPL performance and the Top500 List, inexpensive build-your-own clusters appear to be much more cost effective than more sophisticated HPC architectures. HPL has high spatial and temporal locality — characteristics shared by few real user applications. HPCchallenge benchmarks provide users with additional information to justify policy and purchasing decisions.

Not only does the Japanese Earth Simulator outperform the top American systems on the HPL benchmark (Tflop/s), the differences in bandwidth performance on John McCalpin’s STREAM TRIAD benchmark (Level 1 BLAS) shows even greater performance disparity. The Earth Simulator outperforms the ASCI Q by a factor of 4.64 on HPL. Meanwhile, the higher bandwidth memory and interconnect systems of the Earth Simulator are clearly evident as it outperforms ASCI Q by a factor of 36.25 on STREAM TRIAD. In the presentation and paper, we will compare the measured HPCchallenge Benchmark performance on various HPC architectures — from Cray X1s to Beowulf clusters — using the updated results at http://icl.cs.utk.edu/hpcc/hpcc_results.cgi

Even a small percentage of random memory accesses in real applications can significantly affect the overall performance of that application on architectures not designed to minimize or hide memory latency. Memory latency has not kept up with Moore’s Law. Moore’s Law hypothesizes a 60% compound growth rate per year for microprocessor “performance”, while memory latency has been improving at a compound rate of only 7% per year. The memory-processor performance gap has been growing at a rate of over 50% per year since 1980. The HPCchallenge Benchmarks includes a new metric — Giga UPdates per Second — and a new benchmark — RandomAccess — to measure the ability of an architecture to access memory randomly, i.e., with no locality.

GUPS is calculated by identifying the number of memory locations that can be randomly updated in one second, divided by 1 billion (1e9). The term “randomly” means that there is little relationship between one address to be updated and the next, except that they occur in the space of ½ the total system memory. An update is a read-modify-write operation on a table of 64-bit words. An address is generated, the value at that address read from memory, modified by an integer operation (add, and, or, xor) with a literal value, and that new value is written back to memory.
HPCS HPCchallenge Benchmark Suite

David Koester, Ph.D. (MITRE)
Jack Dongarra (UTK)
Piotr Luszczek (ICL/UTK)

28 September 2004
Outline

• Brief DARPA HPCS Overview
• Architecture/Application Characterization
• HPCchallenge Benchmarks
• Preliminary Results
• Summary
High Productivity Computing Systems

Create a new generation of economically viable computing systems and a procurement methodology for the security/industrial community (2007 – 2010)

Impact:

- **Performance** (time-to-solution): speedup critical national security applications by a factor of 10X to 40X
- **Programmability** (idea-to-first-solution): reduce cost and time of developing application solutions
- **Portability** (transparency): insulate research and operational application software from system
- **Robustness** (reliability): apply all known techniques to protect against outside attacks, hardware faults, & programming errors

Applications:

- Intelligence/surveillance, reconnaissance, cryptanalysis, weapons analysis, airborne contaminant modeling and biotechnology

Fill the Critical Technology and Capability Gap

Today (late 80’s HPC technology)…..to…..Future (Quantum/Bio Computing)
Create a new generation of **economically viable computing systems** and a **procurement methodology** for the security/industrial community (2007 – 2010)

- **Full Scale Development**
  - **Concept Study**
  - **Advanced Design & Prototypes**
  - **Half-Way Point Phase 2**
  - **Full Scale Development**
- **Petascale/s Systems**
  - **Vendors**
  - **Validated Procurement Evaluation Methodology**
  - **Test Evaluation Framework**
  - **New Evaluation Framework**
- **Productivity Team**
- **Phase 1** (2003-2005)
- **Phase 2** (2006-2010)
HPCS Program Goals‡

• HPCS overall productivity goals:
  – Execution (sustained performance)
    ▪ 1 Petaflop/sec (scalable to greater than 4 Petaflop/sec)
    ▪ Reference: Production workflow
  – Development
    ▪ 10X over today’s systems
    ▪ Reference: Lone researcher and Enterprise workflows

• Productivity Framework
  – Base lined for today’s systems
  – Successfully used to evaluate the vendors emerging productivity techniques
  – Provide a solid reference for evaluation of vendor’s proposed Phase III designs.

• Subsystem Performance Indicators
  1) 2+ PF/s LINPACK
  2) 6.5 PB/sec data STREAM bandwidth
  3) 3.2 PB/sec bisection bandwidth
  4) 64,000 GUPS

‡Bob Graybill (DARPA/IPTO) (Emphasis added)
Outline

• Brief DARPA HPCS Overview
• Architecture/Application Characterization
• HPCchallenge Benchmarks
• Preliminary Results
• Summary
Processor-Memory Performance Gap

“Moore’s Law”

- Processor-Memory Performance Gap: (grows 50% / year)
- CPU 60%/yr.
- DRAM 7%/yr.

- Alpha 21264 full cache miss / instructions executed: 180 ns/1.7 ns = 108 clks x 4 or 432 instructions
- Caches in Pentium Pro: 64% area, 88% transistors

*Taken from Patterson-Keeton Talk to SigMod
Processing vs. Memory Access

• Doesn’t cache solve this problem?
  – It depends. With small amounts of contiguous data, usually. With large amounts of non-contiguous data, usually not
  – In most computers the programmer has no control over cache
  – Often “a few” Bytes/FLOP is considered OK

• However, consider operations on the transpose of a matrix (e.g., for adjunct problems)
  – $Xa = b$  $X^Ta = b$
  – If $X$ is big enough, 100% cache misses are guaranteed, and we need at least 8 Bytes/FLOP (assuming $a$ and $b$ can be held in cache)

• Latency and limited bandwidth of processor-memory and node-node communications are major limiters of performance for scientific computation
Consider another benchmark: Linpack

\[ A \times x = b \]

Solve this linear equation for the vector \( x \), where \( A \) is a known matrix, and \( b \) is a known vector. Linpack uses the BLAS routines, which divide \( A \) into blocks.

On the average Linpack requires 1 memory reference for every 2 FLOPs, or 4Bytes/Flop.

Many of these can be cache references.
Consider the simple benchmark: STREAM TRIAD

\[ a(i) = b(i) + q \cdot c(i) \]

\( a(i), b(i), \) and \( c(i) \) are vectors; \( q \) is a scalar
Vector length is chosen to be much longer than cache size

Each execution includes
2 memory loads + 1 memory store
2 FLOPs
12 Bytes/FLOP (assuming 32 bit precision)

No computer has enough memory bandwidth to reference
12 Bytes for each FLOP!
Processing vs. Memory Access

Random Access

Tables

\[ T \]

\[ 2^n \]

\[ 1/2 \text{ Memory} \]

Define Addresses

Sequences of bits within \( a_i \)

Data Stream

\[ \{A_i\} \]

\[ \text{Length} \ 2^{n+2} \]

Data-Driven Memory Access

\[ a_i \]

\[ 64 \text{ bits} \]

\[ \text{Bit-Level Exclusive Or} \]

\[ \oplus \]

The Commutative and Associative nature of \( \oplus \) allows processing in any order

\[ k = [a_i <63, 64-n>] \]

The expected value of the number of accesses per memory location \( T[k] \)

\[ E[T[k]] = \frac{2^{n+2}}{2^n} = 4 \]

Acceptable Error — 1%

Look ahead and Storage — 1024 per “node”
Bounding Mission Partner Applications

HPCS Productivity Design Points

Spatial Locality

High

Low

Temporal Locality

High

Low

Mission Partner Applications

FFT

HPL

RandomAccess

STREAM

PTRANS
Outline

• Brief DARPA HPCS Overview
• Architecture/Application Characterization
• HPCchallenge Benchmarks
• Preliminary Results
• Summary
HPCS HPCchallenge Benchmarks

• HPCScHallenge Benchmarks
  – Being developed by Jack Dongarra (ICL/UT)
  – Funded by the DARPA High Productivity Computing Systems (HPCS) program (Bob Graybill (DARPA/IPTO))

To examine the performance of High Performance Computer (HPC) architectures using kernels with more challenging memory access patterns than High Performance Linpack (HPL)
HPCchallenge Goals

• To examine the performance of HPC architectures using kernels with more challenging memory access patterns than HPL
  – HPL works well on all architectures — even cache-based, distributed memory multiprocessors due to
    1. Extensive memory reuse
    2. Scalable with respect to the amount of computation
    3. Scalable with respect to the communication volume
    4. Extensive optimization of the software

• To complement the Top500 list

• To provide benchmarks that bound the performance of many real applications as a function of memory access characteristics — e.g., spatial and temporal locality
HPCchallenge Benchmarks

**Local**
- DGEMM (matrix x matrix multiply)
- STREAM
  - COPY
  - SCALE
  - ADD
  - TRIADD
- EP-RandomAccess
- 1D FFT

**Global**
- High Performance LINPACK (HPL)
- PTRANS — parallel matrix transpose
- G-RandomAccess
- 1D FFT
- b_eff — interprocessor bandwidth and latency

HPCchallenge pushes spatial and temporal boundaries; sets performance bounds
HPC Challenge Benchmark

The HPC Challenge benchmark consists of basically 7 benchmarks:

1. **SMT** - the Sparse Matrix benchmark which measures the floating point rate of execution for solving a linear system of equations.
2. **DFT** - measures the floating point rate of execution of a double precision real matrix multiply.
3. **STREAM** - a simple synthetic benchmark program that measures sustainable memory bandwidth (in GFLOPS) and the corresponding computation rate for simple vector load.
4. **FTN95** (parallel matrix transpose) - exercises the communications where pairs of processes communicate with each other simultaneously. It is a useful test of the total communications capability of the network.
5. **RandomAccess** - measures the rate of integer random updates of memory (in MFLOPS).
6. **FFLOPS** - measures the floating point rate of execution of double precision complex one-dimensional Fast Fourier Transform (FFT).
7. **tr4** (effective bandwidth benchmark) - a set of tests to measure latency and bandwidth of a number of simultaneous communication patterns

**HPC Challenge Home Page**

- Home
- Rules
- News
- Download
- FAQ
- Links
- Collaborators
- Sponsors
- Upload
- Results

**MITRE**

HPC Challenge Benchmarks

**ICL/UTK**

Web Site

http://icl.cs.utk.edu/hpcc/
Outline

- Brief DARPA HPCS Overview
- Architecture/Application Characterization
- HPCchallenge Benchmarks
- Preliminary Results
- Summary
## Preliminary Results

### Machine List (1 of 2)

<table>
<thead>
<tr>
<th>Affiliation</th>
<th>Manufacturer</th>
<th>System</th>
<th>ProcessorType</th>
<th>Procs</th>
</tr>
</thead>
<tbody>
<tr>
<td>U Tenn</td>
<td>Atipa Cluster AMD 128 procs</td>
<td>Conquest cluster</td>
<td>AMD Opteron</td>
<td>128</td>
</tr>
<tr>
<td>AHPCRC</td>
<td>Cray X1 124 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>124</td>
</tr>
<tr>
<td>AHPCRC</td>
<td>Cray X1 124 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>124</td>
</tr>
<tr>
<td>AHPCRC</td>
<td>Cray X1 124 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>124</td>
</tr>
<tr>
<td>ERDC</td>
<td>Cray X1 60 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>60</td>
</tr>
<tr>
<td>ERDC</td>
<td>Cray X1 60 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>60</td>
</tr>
<tr>
<td>ORNL</td>
<td>Cray X1 252 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>252</td>
</tr>
<tr>
<td>ORNL</td>
<td>Cray X1 252 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>252</td>
</tr>
<tr>
<td>AHPCRC</td>
<td>Cray X1 120 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>120</td>
</tr>
<tr>
<td>ORNL</td>
<td>Cray X1 64 procs</td>
<td>X1</td>
<td>Cray X1 MSP</td>
<td>64</td>
</tr>
<tr>
<td>AHPCRC</td>
<td>Cray T3E 1024 procs</td>
<td>T3E</td>
<td>Alpha 21164</td>
<td>1024</td>
</tr>
<tr>
<td>ORNL</td>
<td>HP zx6000 Itanium 2 128 procs</td>
<td>Integrity zx6000</td>
<td>Intel Itanium 2</td>
<td>128</td>
</tr>
<tr>
<td>PSC</td>
<td>HP AlphaServer SC45 128 procs</td>
<td>AlphaServer SC45</td>
<td>Alpha 21264B</td>
<td>128</td>
</tr>
<tr>
<td>ERDC</td>
<td>HP AlphaServer SC45 484 procs</td>
<td>AlphaServer SC45</td>
<td>Alpha 21264B</td>
<td>484</td>
</tr>
</tbody>
</table>
### Preliminary Results
### Machine List (2 of 2)

<table>
<thead>
<tr>
<th>Affiliation</th>
<th>Manufacturer</th>
<th>System</th>
<th>ProcessorType</th>
<th>Procs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>IBM 655 Power4+ 64 procs</td>
<td>eServer pSeries 655</td>
<td>IBM Power 4+</td>
<td>64</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM 655 Power4+ 128 procs</td>
<td>eServer pSeries 655</td>
<td>IBM Power 4+</td>
<td>128</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM 655 Power4+ 256 procs</td>
<td>eServer pSeries 655</td>
<td>IBM Power 4+</td>
<td>256</td>
</tr>
<tr>
<td>NAVO</td>
<td>IBM p690 Power4 504 procs</td>
<td>p690</td>
<td>IBM Power 4</td>
<td>504</td>
</tr>
<tr>
<td>ARL</td>
<td>IBM SP Power3 512 procs</td>
<td>RS/6000 SP</td>
<td>IBM Power 3</td>
<td>512</td>
</tr>
<tr>
<td>ORNL</td>
<td>IBM p690 Power4 256 procs</td>
<td>p690</td>
<td>IBM Power 4</td>
<td>256</td>
</tr>
<tr>
<td>ORNL</td>
<td>IBM p690 Power4 64 procs</td>
<td>p690</td>
<td>IBM Power 4</td>
<td>64</td>
</tr>
<tr>
<td>ARL</td>
<td>Linux Networx Xeon 256 procs</td>
<td>Powell</td>
<td>Intel Xeon</td>
<td>256</td>
</tr>
<tr>
<td>U Manchester</td>
<td>SGI Altix Itanium 2 32 procs</td>
<td>Altix 3700</td>
<td>Intel Itanium 2</td>
<td>32</td>
</tr>
<tr>
<td>ORNL</td>
<td>SGI Altix Itanium 2 128 procs</td>
<td>Altix</td>
<td>Intel Itanium 2</td>
<td>128</td>
</tr>
<tr>
<td>U Tenn</td>
<td>SGI Altix Itanium 2 32 procs</td>
<td>Altix</td>
<td>Intel Itanium 2</td>
<td>32</td>
</tr>
<tr>
<td>U Tenn</td>
<td>SGI Altix Itanium 2 32 procs</td>
<td>Altix</td>
<td>Intel Itanium 2</td>
<td>32</td>
</tr>
<tr>
<td>U Tenn</td>
<td>SGI Altix Itanium 2 32 procs</td>
<td>Altix</td>
<td>Intel Itanium 2</td>
<td>32</td>
</tr>
<tr>
<td>U Tenn</td>
<td>SGI Altix Itanium 2 32 procs</td>
<td>Altix</td>
<td>Intel Itanium 2</td>
<td>32</td>
</tr>
<tr>
<td>U Tenn</td>
<td>SGI Altix Itanium 2 32 procs</td>
<td>Altix</td>
<td>Intel Itanium 2</td>
<td>32</td>
</tr>
<tr>
<td>NASA ASC</td>
<td>SGI Origin 23900 R16K 256 procs</td>
<td>Origin 3900</td>
<td>SGI MIPS R16000</td>
<td>256</td>
</tr>
<tr>
<td>U Aachen/RWTH</td>
<td>SunFire 15K 128 procs</td>
<td>Sun Fire 15k/6800 SMP-Cluster</td>
<td>Sun UltraSparc III</td>
<td>128</td>
</tr>
<tr>
<td>OSC</td>
<td>Voltaire Cluster Xeon 128 procs</td>
<td>Pinnacle 2X200 Cluster</td>
<td>Intel Xeon</td>
<td>128</td>
</tr>
</tbody>
</table>
STREAM TRIAD vs HPL
120-128 Processors

Basic Performance
120-128 Processors

<table>
<thead>
<tr>
<th>STREAM TRIAD</th>
<th>a(i) = b(i) + q * c(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPL</td>
<td>A x = b</td>
</tr>
</tbody>
</table>
STREAM TRIAD vs HPL
>252 Processors

Basic Performance
>=252 Processors

<table>
<thead>
<tr>
<th>System</th>
<th>STREAM TRIAD</th>
<th>HPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP-STREAM TRIAD Tflop/s</td>
<td>$a(i) = b(i) + q \cdot c(i)$</td>
<td>$A \cdot x = b$</td>
</tr>
<tr>
<td>Cray X1 252 proc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM p690 Power4 256 proc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux Network Xeon 256 proc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGI Origin 2390 R16K 256 proc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP AlphaServer SC45 484 proc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM SP Power3 512 proc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cray T3E 1024 proc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TFlop/s

0.0
0.5
1.0
1.5
2.0
2.5

()/op/s
STREAM ADD vs PTRANS
60-128 Processors

\[
\begin{align*}
\text{STREAM ADD} & \quad a(i) = b(i) + c(i) \\
\text{PTRANS} & \quad a = a + b^T
\end{align*}
\]
STREAM ADD vs PTRANS
>252 Processors

Basic Performance
>=252 Processors

| STREAM ADD | a(i) = b(i) + c(i) |
| PTRANS     | a = a + b^T        |

<table>
<thead>
<tr>
<th>GB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000.0</td>
</tr>
<tr>
<td>1,000.0</td>
</tr>
<tr>
<td>100.0</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.1</td>
</tr>
</tbody>
</table>

- **STREAM ADD**: `a(i) = b(i) + c(i)`
- **PTRANS**: `a = a + b^T`
Outline

• Brief DARPA HPCS Overview
• Architecture/Application Characterization
• HPCchallenge Benchmarks
• Preliminary Results
• Summary
Summary

- DARPA HPCS Subsystem Performance Indicators
  - 2+ PF/s LINPACK
  - 6.5 PB/sec data STREAM bandwidth
  - 3.2 PB/sec bisection bandwidth
  - 64,000 GUPS

- Important to understand architecture/application characterization
  - Where did all the lost “Moore’s Law performance go?”

  - Peruse the results!
  - Contribute!

![HPCS Productivity Design Points Diagram](image-url)