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**

**Advanced Design & Prototypes**

**Concept Study**

**Phase 1** (2003-2005)

**Phase 2**

**Phase 3** (2006-2010)

**Half-Way Point**

**Phase 2**

**Vendors**

**Validated Procurement Evaluation Methodology**

**Technology Assessment Review**

**Test Evaluation Framework**

**New Evaluation Framework**

**Productivity Team**

**Petascale/s Systems**

**MITRE**

**ICL/UTK**

**CRAY**

**IBM**

**Sun**

**sgi**
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"

• 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
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

\[ A \mathbf{x} = \mathbf{b} \]

Solve this linear equation for the vector \( \mathbf{x} \), where \( A \) is a known matrix, and \( \mathbf{b} \) 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

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

\( a(i), b(i), \) 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

\[ T \]

\[ a_i \]

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

64 bits

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

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

Bit-Level Exclusive Or

\[ \oplus \]

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

Data Stream

\[ \{A_i\} \]

Length \( 2^{n+2} \)

Data-Driven Memory Access

Acceptable Error — 1%

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

HPCS Productivity Design Points

Spatial Locality

Low

High

Temporal Locality

High

Low

Mission Partner Applications

FFT

RandomAccess

PTRANS

STREAM

HPL
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
Web Site
http://icl.cs.utk.edu/hpcc/

• Home
• Rules
• News
• Download
• FAQ
• Links
• Collaborators
• Sponsors
• Upload
• Results
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>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>System</th>
<th>EP-STREAM TRIAD Tflop/s</th>
<th>HPL TFlop/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray X1 X1 124 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cray X1 X1 124 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AlphaServer SC45 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM 655 Power4+ 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGI Altix Itanium 2 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SunFire 15K 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltaire Cluster Xeon 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP zx6000 Itanium 2 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMD 128 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STREAM TRIAD</td>
<td>a(i) = b(i) + q * c(i)</td>
<td></td>
</tr>
<tr>
<td>HPL</td>
<td>A x = b</td>
<td></td>
</tr>
</tbody>
</table>

a(i) = b(i) + q * c(i)

A x = b
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>Cray X1 252 procs</td>
<td></td>
<td>A x = b</td>
</tr>
<tr>
<td>IBM p690 Power4+ 256 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGI Origin 23900 R16K 256 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP AlphaServer SC45 484 procs</td>
<td>a(i) = b(i) + q *c(i)</td>
<td></td>
</tr>
<tr>
<td>IBM SP Power3 512 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cray T3E 1024 procs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Equations:**
- STREAM TRIAD: \( a(i) = b(i) + q \cdot c(i) \)
- HPL: \( A \cdot x = b \)
STREAM ADD vs PTRANS
60-128 Processors

**Basic Performance**
60-128 Processors

<table>
<thead>
<tr>
<th>STREAM ADD</th>
<th>$a(i) = b(i) + c(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTRANS</td>
<td>$a = a + b^T$</td>
</tr>
</tbody>
</table>
STREAM ADD vs PTRANS

>252 Processors

Basic Performance

>=252 Processors

<table>
<thead>
<tr>
<th>System</th>
<th>STREAM ADD</th>
<th>PTRANS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray X1 252 procs</td>
<td>a(i) = b(i) + c(i)</td>
<td>a = a + b^T</td>
</tr>
<tr>
<td>IBM p690 Power4 504 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGI Origin 23900 R/16K 236 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP AlphaServer SC45 484 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM SP Power3 512 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cray T3E 1024 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux Network Xeon 256 procs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM 655 Power4 256 procs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GB/s

0.1 1.0 10.0 100.0 1,000.0 10,000.0
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?”

• HPCchallenge Benchmarks — http://icl.cs.utk.edu/hpcc/
  – Peruse the results!
  – Contribute!