HPC Challenge Benchmark Suite and the Path Towards Usable Petascale Computing

Dr. Jeremy Kepner
MIT Lincoln Laboratory

This work is sponsored by the Defense Advanced Research Projects Administration under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.
Outline

• Introduction

• Evolution of Supercomputing
  • Program Goals
  • Architecture Challenges

• HPC Challenge
• Competition Results
• Towards Petascale
• Evaluating Productivity
• Summary
**Evolution of Supercomputing**

**Killer Apps:**
- **1980s:** Weapons Design, Cryptanalysis
- **1990s:** Internet, Biotech
- **2000s:** Finance, Animation
- **2010s:** Biotech, Entertainment

**Processors:**
- **1980s:** ~10
- **1990s:** ~1000
- **2000s:** ~10,000
- **2010s:** ~100,000

**% who can use:**
- **1980s:** Most
- **1990s:** Most
- **2000s:** Most
- **2010s:** Very Few

**Key Points:**
- 10^9 increase in peak performance
- *Not* focused on DoD applications
- Extremely difficult to program

**Legend:**
- Cray Vector Machines
- Massively Parallel IBM SP1 Thinking Machines
- Japanese Vector Machines
- Linux/Intel Clusters
Goal:
- Provide a new generation of economically viable high productivity computing systems for the national security and industrial user community (2010)

Focus on:
- Real (not peak) performance of critical national security applications
  - Intelligence/surveillance
  - Reconnaissance
  - Cryptanalysis
  - Weapons analysis
  - Airborne contaminant modeling
  - Biotechnology
- Programmability: reduce cost and time of developing applications
- Software portability and system robustness
• Standard architecture produces a “steep” multi-layered memory hierarchy
  – Programmer must manage this hierarchy to get good performance
• HPCS technical goal
  – Produce a system with a “flatter” memory hierarchy that is easier to program
HPCS Performance Targets

HPC Challenge Benchmark
- Top500: solves a system
  \[ Ax = b \]
- STREAM: vector operations
  \[ A = B + s \times C \]
- FFT: 1D Fast Fourier Transform
  \[ Z = \text{FFT}(X) \]
- RandomAccess: random updates
  \[ T(i) = \text{XOR}(T(i), r) \]

Corresponding Memory Hierarchy

- Registers
- Cache
- Local Memory
- Remote Memory
- Disk

HPCS Targets (improvement)

- 2 Petaflops (8x)
- 6.5 Petabyte/s (40x)
- 0.5 Petaflops (200x)
- 64,000 GUPS (2000x)

- HPCS program has developed a new suite of benchmarks (HPC Challenge)
- Each benchmark focuses on a different part of the memory hierarchy
- HPCS program performance targets will flatten the memory hierarchy, improve real application performance, and make programming easier
HPCS Roadmap

- 5 vendors in phase 1; 3 vendors in phase 2; 1+ vendors in phase 3
- MIT Lincoln Laboratory leading measurement and evaluation team

**Full Scale Development**

**Advanced Design & Prototypes**

**Concept Study**

- Phase 1: $20M (2002)
- Phase 3: TBD (2006-2010)

- MITRE team
- New Evaluation Framework
- Validated Procurement Evaluation Methodology
- Test Evaluation Framework
- Petascale Systems
Today’s Talk

8 HPCchallenge Benchmarks

HPCS Benchmark Spectrum

Spectrum of benchmarks provide different views of system
- HPCchallenge pushes spatial and temporal boundaries; sets performance bounds
- Applications drive system issues; set legacy code performance bounds
- Kernels and Compact Apps for deeper analysis of execution and development time
Outline

- Introduction
- HPC Challenge
  - Benchmark Details
  - Connecting to Real Apps
- Competition Results
- Toward Petascale
- Summary
HPL “Top500” Benchmark

- High Performance Linpack (HPL) solves a system $Ax = b$
- Core operation is a LU factorization of a large $M \times M$ matrix
- Results are reported in floating point operations per second (flops)

Parallel Algorithm

- Linear system solver (requires all-to-all communication)
- Stresses local matrix multiply performance
- DARPA HPCS goal: 2 Petaflops (8x over current best)
STREAM Benchmark

- Performs scalar multiply and add
- Results are reported in bytes/second

Parallel Algorithm

\[ A = B + s \times C \]

- Basic operations on large vectors (requires no communication)
- Stresses local processor to memory bandwidth
- DARPA HPCS goal: 6.5 Petabytes/second (40x over current best)
FFT Benchmark

- 1D Fast Fourier Transforms an N element complex vector
- Typically done as a parallel 2D FFT
- Results are reported in floating point operations per second (flops)

Parallel Algorithm

- FFT a large complex vector (requires all-to-all communication)
- Stresses interprocessor communication of large messages
- DARPA HPCS goal: 0.5 Petaflops (200x over current best)
RandomAccess Benchmark

- Randomly updates N element table of unsigned integers
- Each processor generates indices, sends to all other processors, performs XOR
- Results are reported in Giga Updates Per Second (GUPS)

Parallel Algorithm

- Generate random indices
- Send, XOR, Update
- Send, XOR, Update

Registers

Cache

Local Memory

Remote Memory

Disk

Instr. Operands

Blocks

Messages

Pages

Table

0 1 . . Np-1

0 1 . . NP-1

- Randomly updates memory (requires all-to-all communication)
- Stresses interprocessor communication of *small* messages
- DARPA HPCS goal: 64,000 GUPS (*2000x over current best*)
Outline

• Introduction

• HPC Challenge
  • Benchmark Details
  • Connecting to Real Apps

• Competition Results

• Towards Petascale

• Evaluating Productivity

• Summary
Example SAR Application

**Front-End Sensor Processing**
- Adaptive Beamforming
- Image Formation

**Back-End Detection and ID**
- Change Detection
- Target ID
  - Many small correlations on random pieces of large image

**Random Access**
- Top500
  - Solve linear systems

**FFT STREAM**
- Pulse compression
- Polar Interpolation
- FFT, IFFT (corner turn)

**HPC Challenge benchmarks are similar to pieces of real apps**
**Real applications are an average of many different operations**
**How do we correlate HPC Challenge with application performance?**
Spatial and Temporal Locality

- Programs can be decomposed into memory reference patterns
- Stride is the distance between memory references
  - Programs with small strides have high “Spatial Locality”
- Reuse is the number of operations performed on each reference
  - Programs with large reuse have high “Temporal Locality”
- Can measure in real programs and correlate with HPC Challenge
• HPC Challenge bounds real applications
  – Allows us to map between applications and benchmarks
• How do we get HPC Challenge run on the biggest systems?
Outline

• Introduction

• HPC Challenge

• Competition Results

• Towards Petascale

• Summary

• HPC Challenge Award
• Performance Results
• Programming Results
HPC Challenge Award Competition

• Class 1: Best Performance (4 awards)
  – Best performance on a run submitted to the website
    HPL
    RandomAccess
    STREAM
    FFT
  – The prize will be $500 plus a certificate for each benchmark

• Class 2: Most Productivity
  – Most "elegant" implementation of at least two benchmarks
  – 50% on performance
  – 50% on code elegance, clarity, and size
  – The prize will be $1500 plus a certificate for this award

• Awards presented at the Supercomputing 2005 conference

• Co-chairs: Jack Dongarra (UTK) and Jeremy Kepner (MIT LL)

Prizes sponsored by HPCWire
Competitors

• Some Notable Class 1 Competitors
  - SGI (NASA) “Columbia” 10,000 CPUs
  - NEC (HLRS) SX-8 512 CPUs
  - IBM (DOE LLNL) BG/L 131,072 CPUs “Purple” 10,240 CPUs
  - CRAY (DOE ORNL) X1 1008 CPUs “Jaguar” XT3 5200 CPUs
  - DELL (MIT LL) 300 CPUs “LLGrid”
  - CRAY (DOD ERDC) XT3 4096 CPUs “Sapphire”

• Class 2: 11 Submissions / 5 Finalists
  - B. Kuszmaul (MIT CSAIL) Cilk on Sun Ultrasparc
  - C. Cascaval (IBM) UPC on Blue Gene/L
  - J. Feo (Cray) pragmas on MultiThreaded Architecture (MTA)
  - N. Wichmann (CRAY) UPC on X1E
  - C. Moler (The Mathworks) Parallel Matlab Prototype on Cray XD1
HPC Challenge Performance Results

- All results in words/second
- Highlights memory hierarchy
- Clusters
  - Hierarchy steepens
- HPC systems
  - Hierarchy constant
- HPCS Goals
  - Hierarchy flattens
  - Easier to program

Effective Bandwidth (words/second)

- Top500 (words/s)
- STREAM (words/s)
- FFT (words/s)
- RandomAccess (words/s)

Systems (in Top500 order)

Clusters ~10^6

HPC ~10^4

HPCS ~10^2
Outline

• Introduction
• HPC Challenge
• **Competition Results**
  • HPC Challenge Award
  • Performance Results
  • *Productivity Results*
• Towards Petascale
• Evaluating Productivity
• Summary
### Programming Models and Languages

<table>
<thead>
<tr>
<th>Memory Model / Architecture</th>
<th>Programming Languages Studied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial</td>
<td>C/C++</td>
</tr>
<tr>
<td></td>
<td>Fortran</td>
</tr>
<tr>
<td></td>
<td>Java</td>
</tr>
<tr>
<td></td>
<td>Matlab</td>
</tr>
<tr>
<td>Shared Memory</td>
<td>C/Fortran + OpenMP</td>
</tr>
<tr>
<td></td>
<td>High Performance Fortran (HPF)</td>
</tr>
<tr>
<td></td>
<td>Unified Parallel C (UPC)</td>
</tr>
<tr>
<td></td>
<td>Cilk</td>
</tr>
<tr>
<td>Distributed Memory</td>
<td>C/Fortran + MPI</td>
</tr>
<tr>
<td></td>
<td>Matlab*P</td>
</tr>
<tr>
<td></td>
<td>pMatlab</td>
</tr>
</tbody>
</table>

- HPCS Program is making a significant investment in new programming languages and programming models
- HPC Challenge Class 2 Award is designed to highlight this work
• Class 2 Award
  – 50% Performance
  – 50% Elegance
• 30 Codes submitted by 11 teams
• Speedup relative to serial C on workstation
• Code size relative to serial C
• Results show there are better parallel programming approaches
  – 27 of 30 smaller than C+MPI Ref; 15 smaller than serial
  – 24 of 30 faster than serial; 15 in HPCS quadrant (includes all winners)
Summary

• HPCS Goals
  – Provide a new generation of economically viable high productivity computing systems for the national security and industrial user community (2010)

• HPSS Productivity Team goal is to develop an acquisition quality framework for HPC systems that includes
  – Development time
  – Execution time

• HPC Challenge is a powerful tool for evaluating system performance and HPCS goals
  – Class 1 results highlights benefits relative to current HPC systems (e.g. flatter memory hierarchy)
  – Class 2 awards demonstrates that there are many “better” programming approaches than C+MPI