Overview

♦ Look at current state of high performance computing
  ➢ Past, present and a look ahead
♦ Potential gains by exploiting lower precision devices
  ➢ GPUs, Cell, SSE2, AltaVec
♦ New performance evaluation tools
  ➢ HPCS – HPC Challenge
H. Meuer, H. Simon, E. Strohmaier, & JD

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP
  \[ Ax=b, \text{ dense problem} \]

- Updated twice a year
  SC‘xy in the States in November
  Meeting in Germany in June

- All data available from www.top500.org
Architecture/Systems Continuum

Tightly Coupled
- Custom processor with custom interconnect
  - Cray X1
  - NEC SX-8
  - IBM Regatta
  - IBM Blue Gene/L
- Commodity processor with custom interconnect
  - SGI Altix
  - Intel Itanium 2
  - Cray XT3, XD1
  - AMD Opteron
- Commodity processor with commodity interconnect
  - Clusters
    - Pentium, Itanium, Opteron, Alpha
    - GigE, Infiniband, Myrinet, Quadrics
- Loosely Coupled
  - NEC TX7
  - IBM eServer
  - Dawning

Loosely Coupled

Processor Types

Intel + IBM Power PC + AMD = 91%
### 26th List: The TOP10

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Computer</th>
<th>Rmax (TF/s)</th>
<th>Installation Site</th>
<th>Country</th>
<th>Year</th>
<th>#Proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>BlueGene/L</td>
<td>280.6</td>
<td>Lawrence Livermore Nat Lab</td>
<td>USA</td>
<td>2005</td>
<td>custom</td>
</tr>
<tr>
<td>IBM</td>
<td>eServer Blue Gene</td>
<td>91.29</td>
<td>IBM Thomas Watson Research</td>
<td>USA</td>
<td>2005</td>
<td>custom</td>
</tr>
<tr>
<td>IBM</td>
<td>ASC Purple PowerS p575</td>
<td>63.39</td>
<td>DOE Lawrence Livermore Nat Lab</td>
<td>USA</td>
<td>2005</td>
<td>custom</td>
</tr>
<tr>
<td>SGI</td>
<td>Columbia Altix, Itanium/Infiniband</td>
<td>51.87</td>
<td>NASA Ames</td>
<td>USA</td>
<td>2004</td>
<td>hybrid</td>
</tr>
<tr>
<td>Dell</td>
<td>Thunderbird Pentium/Infiniband</td>
<td>38.27</td>
<td>DOE Sandia Nat Lab</td>
<td>USA</td>
<td>2005</td>
<td>custom</td>
</tr>
<tr>
<td>Cray</td>
<td>Red Storm Cray XT3 AMD</td>
<td>36.19</td>
<td>DOE Sandia Nat Lab</td>
<td>USA</td>
<td>2005</td>
<td>hybrid</td>
</tr>
<tr>
<td>NEC</td>
<td>Earth-Simulator SX-6</td>
<td>35.86</td>
<td>Earth Simulator Center</td>
<td>Japan</td>
<td>2002</td>
<td>custom</td>
</tr>
<tr>
<td>IBM</td>
<td>MareNostrum PPC 970/Myrinet</td>
<td>27.91</td>
<td>Barcelona Supercomputer Center</td>
<td>Spain</td>
<td>2005</td>
<td>custom</td>
</tr>
<tr>
<td>IBM</td>
<td>eServer Blue Gene</td>
<td>27.45</td>
<td>ASTRON University Groningen</td>
<td>Netherlands</td>
<td>2005</td>
<td>custom</td>
</tr>
<tr>
<td>Cray</td>
<td>Jaguar Cray XT3 AMD</td>
<td>20.53</td>
<td>Oak Ridge Nat Lab</td>
<td>USA</td>
<td>2005</td>
<td>hybrid</td>
</tr>
</tbody>
</table>

### Performance Projection

- 1 Eflop/s
- 10 Pflop/s
- 100 Pflop/s
- 1 Ptflop/s
- 1 Tflop/s
- 10 Tflop/s
- 100 Tflop/s
- 1 Gflop/s
- 10 Gflop/s
- 100 Gflop/s
- 100 Mflop/s

- DARPA HPCS
- SUM
- My Laptop
- N=1
- N=500

A PetaFlop Computer by the End of the Decade

- 10 Companies working on building a Petaflop system by the end of the decade.
  - Cray
  - IBM
  - Sun
  - Dawning
  - Galactic
  - Lenovo
  - Hitachi
  - NEC
  - Fujitsu
  - Bull

Japanese

Chinese Companies

“Life Simulator” (10 Pflop/s)

Today’s CPU Architecture:

Heat becoming an unmanageable problem

Increasing the number of gates into a tight knot and decreasing the cycle time of the processor

Square relationship between the cycle time and power.
We have seen increasing number of gates on a chip and increasing clock speed.

Heat becoming an unmanageable problem, Intel Processors > 100 Watts

We will not see the dramatic increases in clock speeds in the future.

However, the number of gates on a chip will continue to increase.

No Free Lunch For Traditional Software
(Without highly concurrent software it won’t get any faster!)

From Craig Mundie, Microsoft
CPU Desktop Trends – Change is Coming

- Relative processing power will continue to double every 18 months
- 256 logical processors per chip in late 2010

Commodity Processor Trends
Bandwidth/Latency is the Critical Issue, not FLOPS

<table>
<thead>
<tr>
<th></th>
<th>Annual increase</th>
<th>Typical value in 2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-chip floating-point performance</td>
<td>59%</td>
<td>4 GFLOP/s</td>
</tr>
<tr>
<td>Front-side bus bandwidth</td>
<td>23%</td>
<td>1 GWord/s</td>
</tr>
<tr>
<td>DRAM latency</td>
<td>(5.5%)</td>
<td>70 ns</td>
</tr>
</tbody>
</table>

Got Bandwidth?

That Was the Good News

♦ Bad news: the effect of the hardware change on the existing software base
♦ Must rethink the design of our software
  ➢ Another disruptive technology
  ➢ Rethink and rewrite the applications, algorithms, and software

LAPACK - ScaLAPACK

♦ Numerical libraries for linear algebra
♦ LAPACK
  ➢ Late 1980’s
  ➢ Sequential and SMPs
♦ ScaLAPACK
  ➢ Early 1990’s
  ➢ Message passing systems
Right-Looking LU factorization (LAPACK)

- DGETF2 – Unblocked LU
- DLSWP – row swaps
- DLSWP
- DTRSM – triangular solve with many right-hand sides
- DGEMM – matrix-matrix multiply

Steps in the LAPACK LU

- DGETF2
- DLSWP
- DLSWP
- DTRSM
- DGEMM
LU Timing Profile

LAPACK + BLAS threads

Time for each component

1D decomposition and SGI Origin

In this case the performance difference comes from parallelizing row exchanges (DLASWP) and threads in the LU algorithm.

1D decomposition and SGI Origin
Right-Looking LU Factorization

Right-Looking LU with a Lookahead
Pivot Rearrangement and Lookahead

4 Processor runs

Pivot Rearrangement and Lookahead

16 SMP runs
Motivated by...

- The PlayStation 3's CPU based on a chip codenamed "Cell"
- Each Cell contains 8 APUs.
  - An APU is a self-contained vector processor which acts independently from the others.
  - 4 floating point units capable of a total of 32 Gflop/s (8 Gflop/s each)
  - 256 Gflop/s peak 32-bit floating point; 64 bit floating point at 25 Gflop/s.
  - IEEE format, but only rounds toward zero in 32 bit, overflow set to largest

According to IBM, the SPE’s double precision unit is fully IEEE854 compliant.

---

**GPU Performance**

<table>
<thead>
<tr>
<th>GPU Vendor</th>
<th>NVIDIA 6800Ultra</th>
<th>NVIDIA 7800GTX</th>
<th>NVIDIA X1900XTX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2004</td>
<td>2005</td>
<td>2006</td>
</tr>
<tr>
<td>32-bit Performance</td>
<td>60 GFLOPS</td>
<td>200 GFLOPS</td>
<td>400 GFLOPS</td>
</tr>
<tr>
<td>64-bit Performance</td>
<td>must be emulated in software</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Idea Something Like This...

- Exploit 32 bit floating point as much as possible.
  - Especially for the bulk of the computation
- Correct or update the solution with selective use of 64 bit floating point to provide a refined results
- Intuitively:
  - Compute a 32 bit result,
  - Calculate a correction to 32 bit result using selected higher precision and,
  - Perform the update of the 32 bit results with the correction using high precision.

32 and 64 Bit Floating Point Arithmetic

- Iterative refinement for dense systems can work this way.
  - Solve $Ax = b$ in lower precision,
    - save the factorization $(L^*U = A^*P)$; $O(n^3)$
  - Compute in higher precision $r = b - A^*x$; $O(n^2)$
    - Requires the original data $A$ (stored in high precision)
  - Solve $Az = r$; using the lower precision factorization; $O(n^2)$
  - Update solution $x_+ = x + z$ using high precision; $O(n)$
  - Iterate until converged.

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- We can show using this approach that we can compute the solution to 64-bit floating point precision.

Requires extra storage, total is 1.5 times normal;
$O(n^3)$ work is done in lower precision
$O(n^2)$ work is done in high precision
Problems if the matrix is ill-conditioned in sp: $O(10^9)$
On the Way to Understanding How to Use the Cell Something Else Happened ...

- Realized have the similar situation on our commodity processors.
  - That is, SP is 2X as fast as DP on many systems.

- The Intel Pentium and AMD Opteron have SSE2.
  - 2 flops/cycle DP
  - 4 flops/cycle SP

- IBM PowerPC has AltiVec.
  - 8 flops/cycle SP
  - 4 flops/cycle DP
  - No DP on AltiVec

### Processor and BLAS Library

<table>
<thead>
<tr>
<th>Processor</th>
<th>SGEMM (GFlop/s)</th>
<th>DGMEM (GFlop/s)</th>
<th>Speedup SP/DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium III Katmai (0.6GHz)</td>
<td>0.98</td>
<td>0.46</td>
<td>2.13</td>
</tr>
<tr>
<td>Pentium III Coppermine (0.9GHz)</td>
<td>1.59</td>
<td>0.79</td>
<td>2.01</td>
</tr>
<tr>
<td>Pentium Xeon Northwood (2.4GHz)</td>
<td>7.68</td>
<td>3.88</td>
<td>1.98</td>
</tr>
<tr>
<td>Pentium Xeon Prescott (3.2GHz)</td>
<td>10.54</td>
<td>5.15</td>
<td>2.05</td>
</tr>
<tr>
<td>Pentium IV Prescott (3.4GHz)</td>
<td>11.09</td>
<td>5.61</td>
<td>1.98</td>
</tr>
<tr>
<td>AMD Opteron 240 (1.4GHz)</td>
<td>4.89</td>
<td>2.48</td>
<td>1.97</td>
</tr>
<tr>
<td>PowerPC G5 (2.7GHz) AltiVec</td>
<td>18.28</td>
<td>9.98</td>
<td>1.83</td>
</tr>
</tbody>
</table>

Performance of single precision and double precision matrix multiply (SGEMM and DGMEM) with n=m=k=1000

### Speedups (Ratio of Times)

<table>
<thead>
<tr>
<th>Architecture (BLAS)</th>
<th>n</th>
<th>DGEMM / SGEMM</th>
<th>DP Solve / SP Solve</th>
<th>DP Solve / Iter Ref</th>
<th># iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium IV-M Northwood</td>
<td>4000</td>
<td>2.02</td>
<td>1.98</td>
<td>1.54</td>
<td>5</td>
</tr>
<tr>
<td>Intel Pentium III Katmai</td>
<td>3000</td>
<td>2.12</td>
<td>2.11</td>
<td>1.79</td>
<td>4</td>
</tr>
<tr>
<td>Intel Pentium III Coppermine</td>
<td>3500</td>
<td>2.10</td>
<td>2.24</td>
<td>1.92</td>
<td>4</td>
</tr>
<tr>
<td>Intel Pentium IV Prescott</td>
<td>4000</td>
<td>2.00</td>
<td>1.86</td>
<td>1.57</td>
<td>5</td>
</tr>
<tr>
<td>AMD Opteron (Goto)</td>
<td>4000</td>
<td>1.98</td>
<td>1.93</td>
<td>1.53</td>
<td>5</td>
</tr>
<tr>
<td>Sun UltraSPARC Ie (Sunperf)</td>
<td>3000</td>
<td>1.45</td>
<td>1.79</td>
<td>1.58</td>
<td>4</td>
</tr>
<tr>
<td>IBM Power PC G5 (2.7 GHz)</td>
<td>5000</td>
<td>2.29</td>
<td>2.05</td>
<td>1.24</td>
<td>5</td>
</tr>
<tr>
<td>Cray X1 (ilbci)</td>
<td>4000</td>
<td>1.68</td>
<td>1.57</td>
<td>1.32</td>
<td>7</td>
</tr>
<tr>
<td>Compaq Alpha EV6 (CXML)</td>
<td>3000</td>
<td>0.99</td>
<td>1.08</td>
<td>1.01</td>
<td>4</td>
</tr>
<tr>
<td>IBM SP Power3 (ESSL)</td>
<td>3000</td>
<td>1.03</td>
<td>1.13</td>
<td>1.00</td>
<td>3</td>
</tr>
<tr>
<td>SGI Octane (ATLAS)</td>
<td>2000</td>
<td>1.08</td>
<td>1.13</td>
<td>0.91</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Architecture (BLAS-MPI)</th>
<th># procs</th>
<th>n</th>
<th>DP Solve / SP Solve</th>
<th>DP Solve / Iter Ref</th>
<th># iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD Opteron (Goto – OpenMPI MX)</td>
<td>32</td>
<td>22627</td>
<td>1.85</td>
<td>1.79</td>
<td>6</td>
</tr>
<tr>
<td>AMD Opteron (Goto – OpenMPI MX)</td>
<td>64</td>
<td>32000</td>
<td>1.90</td>
<td>1.83</td>
<td>6</td>
</tr>
</tbody>
</table>
Refinement Technique Using Single/Double Precision

- **Linear Systems**
  - LU (dense and sparse)
  - Cholesky
  - QR Factorization

- **Eigenvalue**
  - Symmetric eigenvalue problem
  - SVD
  - Same idea as with dense systems,
    - Reduce to tridiagonal/bi-diagonal in lower precision, retain original data and improve with iterative technique using the lower precision to solve systems and use higher precision to calculate residual with original data.
  - $O(n^3)$ per value/vector

- **Iterative Linear System**
  - Relaxed GMRES
  - Inner/outer scheme

LAPACK Working Note

Motivation for Additional Benchmarks

- **Linpack Benchmark**
  - **Good**
    - One number
    - Simple to define & easy to rank
    - Allows problem size to change with machine and over time
  - **Bad**
    - Emphasizes only “peak” CPU speed and number of CPUs
    - Does not stress local bandwidth
    - Does not stress the network
    - Does not test gather/scatter
    - Ignores Amdahl’s Law (Only does weak scaling)
    - …
  - **Ugly**
    - Benchmarketeering hype

- **From Linpack Benchmark and Top500: “no single number can reflect overall performance”**

- **Clearly need something more than Linpack**

- **HPC Challenge Benchmark**
  - Test suite stresses not only the processors, but the memory system and the interconnect.
  - The real utility of the HPCC benchmarks are that architectures can be described with a wider range of metrics than just Flop/s from Linpack.
**DARPA’s High Productivity Computing Systems**

Full Scale Development

Advanced Design & Prototypes

Concept Study

Phase 1

Phase 2 (2003-2005)

Phase 3 (2006-2010)

$10M

$50M

$100M

Half-Way Point

Phase 2

Technology Assessment Review

Vendors

Petascale/s Systems

Test Evaluation Framework

Validated Procurement Evaluation Methodology

New Evaluation Framework

Productivity Team

Goals HPC Challenge Benchmark

- Stress CPU, memory system, interconnect
- 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
- Allow for optimizations
  - Record effort needed for tuning
  - Base run requires MPI and BLAS
- Provide verification of results
- Archive results
Tests on Single Processor and System

- **Local** - only a single processor is performing computations.
- **Embarrassingly Parallel** - each processor in the entire system is performing computations but they do no communicate with each other explicitly.
- **Global** - all processors in the system are performing computations and they explicitly communicate with each other.

HPC Challenge Benchmark

Consists of basically 7 benchmarks:

1. **HPL (LINPACK)** — MPI Global \((Ax = b)\)
2. **STREAM** — Local; single CPU
   *STREAM* — Embarrassingly parallel
3. **PTRANS \((A \leftarrow A + B^T)\)** — MPI Global
4. **RandomAccess** — Local; single CPU
   *RandomAccess* — Embarrassingly parallel
   RandomAccess — MPI Global
5. **BW and Latency** — MPI
6. **FFT** — Global, single CPU, and EP
7. **Matrix Multiply** — single CPU and EP

Random integer read; update; & write
Computational Resources and HPC Challenge Benchmarks

- CPU computational speed
- Memory bandwidth
- Node Interconnect bandwidth

HPL Matrix Multiply

- CPU computational speed
- Computational resources

STREAM

Random & Natural Ring Bandwidth & Latency
**HPC Challenge Benchmark**

The HPC Challenge benchmark consists of basically 7 benchmarks:

1. **HPL** - the Linpack TPB benchmark which measures the floating point rate of execution for solving a linear system of equations.
2. **DGEMM** - measures the floating point rate of execution of double precision real matrix-matrix multiplication.
3. **STREAM** - a simple synthetic benchmark program that measures sustained memory bandwidth (in GB/s) and the corresponding computation rate for simple vector kernel.
4. **STRAW (parallel matrix transpose)** - exercises the communications where pairs of processors communicate with each other simultaneously. It is a useful test of the total communications capacity of the network.
5. **Bandwidth** - measures the rate of integer random updates of memory (OLP). 
6. **EPI** - measures the floating point rate of execution of double precision complex one-dimensional Discrete Fourier Transform (DFT).
7. Communication bandwidth and latency - a set of tests to measure latency and bandwidth of a number of simultaneous communication patterns based on send (effective bandwidth benchmark).
HPC Challenge Benchmark

The HPC Challenge benchmark consists of basically 7 benchmarks:

1. **HPL** - the Lapack TPF benchmark which measures the floating point rate of execution for solving a linear system of equations.

2. **OGEMM** - measures the floating point rate of execution of double precision real matrix-matrix multiplication.

3. **STREAM** - a simple synthetic benchmark program that measures sustainable memory bandwidth (in GB/s) and the corresponding computation rate for simple vector kernels.

4. **STREAM (parallel matrix transpose)** - exercises the communications where pairs of processors communicate with each other simultaneously. It is useful test of the total communications capacity of the network.

5. **Bandwidth** - measures the rate of integer random updates of memory (ULPS).

6. **SFFT** - measures the floating point rate of execution of double precision complex one-dimensional Discrete Fourier Transform (DFT).

7. **Communication bandwidth and latency** - a set of tests to measure latency and bandwidth of a number of simultaneous communication patterns based on *sfl* (effective bandwidth benchmark).
Summary of Current Unmet Needs

- **Performance / Portability**
- **Fault tolerance**
- **Memory bandwidth/Latency**
- **Adaptability**: Some degree of autonomy to self optimize, test, or monitor.
  - Able to change mode of operation: static or dynamic
- **Better programming models**
  - Global shared address space
  - Visible locality
- **Maybe coming soon** (incremental, yet offering real benefits):
  - Global Address Space (GAS) languages: UPC, Co-Array Fortran, Titanium, Chapel, X10, Fortress
    - "Minor" extensions to existing languages
    - More convenient than MPI
    - Have performance transparency via explicit remote memory references
- **What's needed is a long-term, balanced investment in** hardware, software, algorithms and applications in the HPC Ecosystem.

Real Crisis With HPC Is With The Software

- **Our ability to configure a hardware system capable of** 1 PetaFlop ($10^{15}$ ops/s) is without question just a matter of time and $\$$.
- **A supercomputer application and software** are usually much more long-lived than a hardware
  - Hardware life typically five years at most... Apps 20-30 years
  - Fortran and C are the main programming models (still!!)
- **The REAL CHALLENGE is Software**
  - Programming hasn't changed since the 70's
  - HUGE manpower investment
    - MPI... is that all there is?
  - Often requires HERO programming
  - Investments in the entire software stack is required (OS, libs, etc.)
- **Software is a major cost component of modern technologies.**
  - The tradition in HPC system procurement is to assume that the software is free... SOFTWARE COSTS (over and over)
- **What's needed is a long-term, balanced investment in the HPC Ecosystem: hardware, software, algorithms and applications.**
Collaborators / Support

♦ Top500 Team
  ➢ Erich Strohmaier, NERSC
  ➢ Hans Meuer, Mannheim
  ➢ Horst Simon, NERSC

♦ Sca/LAPACK
  ➢ Julien Langou
  ➢ Jakub Kurzak
  ➢ Piotr Luszczek
  ➢ Stan Tomov
  ➢ Julie Langou