**Title goes here**

**Trends in High Performance Computing and Using Numerical Libraries on Clusters**

Jack Dongarra  
University of Tennessee

---

**Outline**

- Look at Clusters in the context of:
  - Top500 Supercomputers (Snapshot from June 2002)
  - Top100 Clusters (Based on Theoretical Peak)

- Self Adapting Numerical Software (SANS) effort
  - Automatic Translation for Linear Algebra Software (ATLAS)
  - LAPACK for Clusters (LFC)
  - Self-Adaptive Linear Solver Architecture (SALSA)

---

**TOP500**

H. Meuer, H. Simon, E. Strohmaier, & JD

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP  
  \( Ax=b \), dense problem

- Updated twice a year
  SC’xy in the States in November
  Meeting in Mannheim, Germany in June
- All data available from [www.top500.org](http://www.top500.org)

---

**Fastest Computer Over Time**

![Graph showing the evolution of the fastest computer over time](image)

---

**Fastest Computer Over Time**

![Graph showing the evolution of the fastest computer over time](image)
Title goes here

Fastest Computer Over Time

Japanese Earth Simulator NEC 5104

Performance Extrapolation

80 Clusters on the Top500

- A total of 42 Intel based and 8 AMD based PC clusters are in the TOP500.
  - 31 of these Intel based cluster are IBM Netfinity systems delivered by IBM.
- A substantial part of these are installed at industrial customers especially in the oil-industry.
  - Including 5 Sun and 5 Alpha based clusters and 21 HP AlphaServer.
- 14 of these clusters are labeled as ‘Self-Made’.

Top10 of the Top500

<table>
<thead>
<tr>
<th>Rank</th>
<th>Manufacturer</th>
<th>Computer</th>
<th>Rating (TF)</th>
<th>Installation Site</th>
<th>Country</th>
<th>Top500</th>
<th>Area of Activity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NEC</td>
<td>Earth Simulator</td>
<td>1,106</td>
<td>Earth Simulator Center</td>
<td>Japan</td>
<td>2004</td>
<td>Research</td>
<td>2,000</td>
</tr>
<tr>
<td>2</td>
<td>IBM</td>
<td>ASCI White</td>
<td>932</td>
<td>Lawrence Livermore National Laboratory</td>
<td>USA</td>
<td>2000</td>
<td>Research</td>
<td>1,300</td>
</tr>
<tr>
<td>3</td>
<td>IBM</td>
<td>ASCI White</td>
<td>932</td>
<td>Lawrence Livermore National Laboratory</td>
<td>USA</td>
<td>2000</td>
<td>Research</td>
<td>1,300</td>
</tr>
<tr>
<td>4</td>
<td>HP</td>
<td>Alphastream 5</td>
<td>4,600</td>
<td>Peking High</td>
<td>China</td>
<td>2002</td>
<td>Academics</td>
<td>3,000</td>
</tr>
<tr>
<td>5</td>
<td>HP</td>
<td>Alphastream 5</td>
<td>4,600</td>
<td>Peking High</td>
<td>China</td>
<td>2002</td>
<td>Academics</td>
<td>3,000</td>
</tr>
<tr>
<td>6</td>
<td>HP</td>
<td>Alphastream 5</td>
<td>4,600</td>
<td>Peking High</td>
<td>China</td>
<td>2002</td>
<td>Academics</td>
<td>3,000</td>
</tr>
<tr>
<td>7</td>
<td>IBM</td>
<td>SP Extended</td>
<td>1,600</td>
<td>NCSA (IL)</td>
<td>USA</td>
<td>2001</td>
<td>Research</td>
<td>1,500</td>
</tr>
<tr>
<td>8</td>
<td>IBM</td>
<td>SP Extended</td>
<td>1,600</td>
<td>NCSA (IL)</td>
<td>USA</td>
<td>2001</td>
<td>Research</td>
<td>1,500</td>
</tr>
<tr>
<td>9</td>
<td>Intel</td>
<td>ASCI Red</td>
<td>2400</td>
<td>Sandia National Laboratory</td>
<td>USA</td>
<td>2002</td>
<td>Research</td>
<td>1,800</td>
</tr>
<tr>
<td>10</td>
<td>IBM</td>
<td>ASCI Red</td>
<td>2400</td>
<td>Sandia National Laboratory</td>
<td>USA</td>
<td>2002</td>
<td>Research</td>
<td>1,800</td>
</tr>
</tbody>
</table>

Architectures

80 Clusters on the Top500

Cluster on the Top500
Title goes here
Title goes here

Software Generation Strategy - ATLAS BLAS

- Parameter study of the hardware
- Generate multiple versions of code, with different values of key performance parameters
- Run and measure the performance for various versions
- Pick best and generate library
- Level 1 cache multiply optimization for:
  - TLB access
  - L1 cache reuse
  - FP unit usage
  - Memory fetch
  - Regular reuse
  - Loop overhead minimization
- Takes ~ 20 minutes to run, generates Level 1, 2, 3 BLAS
- "New" model of high performance programming where critical code is machine generated using parameter optimization
- Designed for RISC arch
  - Super Scalar
  - Need reasonable C compiler
- Today ATLAS is in use within various ASCI and SciDAC activities and by Matlab, Mathematica, Octave, Maple, Debian, Scylla Beowulf, SuSE...

ATLAS Matrix Multiply

Intel Pentium 4 at 2.53GHz – using SSE2

To Use ScaLAPACK a User Must:

- Download the package and auxiliary packages (like PBLAS, BLAS, BLACS, & MPI) to the machines.
- Write a SPMD program which
  - Sets up the logical 2-D process grid
  - Places the data on the logical process grid
  - Calls the numerical library routine in a SPMD fashion
  - Collects the solution after the library routine finishes
- The user must allocate the processors and decide the number of processes the application will run on
- The user must start the application
  - "mpirun -np N user_appl"
    - Note: the number of processors is fixed by the user before the run, if problem size changes dynamically...
- Upon completion, return the processors to the pool of resources

Cluster Numerical Library

- Want to relieve the user of some of the tasks
- Make decisions on which machines to use based on the user’s problem and the state of the system
  - Determine set of procs that should be used
  - Optimize for the best time to solution
  - Distribute the data on the processors and collections of results
  - Start the SPMD library routine on all the platforms
  - Check to see if the computation is proceeding as planned
    - If not perhaps migrate application

Needs an expert to do the tuning:

- Number of processors
- Grid aspect ratio for runs
- Blocksize

ScaLAPACK
Title goes here

**LAPACK For Clusters**
- Developing middleware which couples cluster system information with the specifics of a user problem to launch cluster based applications on the "best" set of resource available.
- Using ScalAPACK as the prototype software, but developing a framework.

**User Interface/Middleware**
- User has problem to solve (e.g. Ax = b)
  - Natural Data (A,b)
  - Middleware
  - Structured Data (A' ,b')
  - Application Library (e.g. LAPACK, ScalAPACK, PETSc,...)
  - Structured Answer (x')

**LAPACK For Clusters**
- software design
  - LFC data collection - a daemon
  - LFC user interface
  - LFC middleware
  - LFC end-ware
- Like Wolski's NWS but for Clusters

**Typical LFC Run**
- User creates data package in a file
- User calls LFC routine
- Available memory
- Execution environment
- Performance monitor
- Solution

**Sample Of The Predictive Power Of The Adhoc Modeler:**

**Experimental LFC on a 64 processor Pentium 4 - 2.4 GHz Xeon Cluster at UTK**
Run-Time Adaptivity for Linear Systems

- Many possible methods: Nature of data is prime consideration in choice
- Dense systems: fairly cut and dry, only adapt to infrastructure
- Sparse systems: a mess. Direct and iterative methods, multigrid, different preconditioners. No one algorithm best for sparse system.

Intelligent Component

- System to mediate between user application and multiple possible libraries
- Self-Adaptivity and Learning Behavior
  - Heuristics are tuned based on data
  - The system can educate the user
- User Interaction
  - User can guide the system by providing further information
  - System teaches user about properties of the data

Future SANS Effort

- Intelligent Component
  - Automates method selection based on data, algorithm, and system attributes
- System component
  - Provides intelligent management of and access to clusters and computational grids
- History database
  - Records relevant info generated by the IC and maintains past performance data
- Fault Tolerant Aspect
  - Transparently detect and recover from failure
  - Algorithmic Fault Tolerance

Collaborators

- TOP500
  - H. Mauer, Mannheim U
  - H. Simon, NERSC
  - E. Strohmaier, NERSC
- SANS-Effort
  - Jeffrey Chen, UTk
  - Jun Ding, UTk
  - Tom Eideen, 3CASE
  - Victor Ejikhome, UTk
  - Piotr Luczczek, UTk
  - Kenny Roche, UTk
  - Sathish Vadvaniyar, UTk
- HPL and ATLAS
  - Antoine Petitet, Sun
  - Clint Whaley, FSU

Availability

- Top500: http://www.top500.org/
- ATLAS: http://icl.cs.utk.edu/atlas/
- LFC: 5 drivers from ScALAPACK 2000/2001
- Algorithm Fault Tolerance: www.icl.utk.edu/~jeftek/papers/AFT.pdf