Architecture-aware Algorithms and Software for Peta and Exascale Computing

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
University of Manchester
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
<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Computer</th>
<th>Country</th>
<th>Cores</th>
<th>Rmax [Pflops]</th>
<th>% of Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nat. SuperComputer Center in Tianjin</td>
<td>NUDT YH Cluster, X5670 2.93Ghz 6C, NVIDIA GPU</td>
<td>China</td>
<td>186,368</td>
<td>2.57</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>DOE / ORNL Oak Ridge Nat Lab</td>
<td>Jaguar / Cray Cray XT5 sixCore 2.6 GHz</td>
<td>USA</td>
<td>224,162</td>
<td>1.76</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>Nat. Supercomputer Center in Shenzhen</td>
<td>Nebulea / Dawning / TC3600 Blade, Intel X5650, Nvidia C2050 GPU</td>
<td>China</td>
<td>120,640</td>
<td>1.27</td>
<td>43</td>
</tr>
<tr>
<td>4</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>Tusbame 2.0 HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU</td>
<td>Japan</td>
<td>73,278</td>
<td>1.19</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>DOE/SC/LBNL/NERSC</td>
<td>Hopper, Cray XE6 12-core 2.1 GHz</td>
<td>USA</td>
<td>153,408</td>
<td>1.054</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td>Commissariat a l'Energie Atomique (CEA)</td>
<td>Tera-100 Bull bullx super-node S6010/S6030</td>
<td>France</td>
<td>138,368</td>
<td>1.050</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>DOE / NNSA Los Alamos Nat Lab</td>
<td>Roadrunner / IBM BladeCenter QS22/LS21</td>
<td>USA</td>
<td>122,400</td>
<td>1.04</td>
<td>76</td>
</tr>
<tr>
<td>8</td>
<td>NSF / NICS / U of Tennessee</td>
<td>kyaken/ Cray Cray XT5 sixCore 2.6 GHz</td>
<td>USA</td>
<td>98,928</td>
<td>.831</td>
<td>81</td>
</tr>
<tr>
<td>9</td>
<td>Forschungszentrum Juelich (FZJ)</td>
<td>Jugene / IBM Blue Gene/P Solution</td>
<td>Germany</td>
<td>294,912</td>
<td>.825</td>
<td>82</td>
</tr>
<tr>
<td>10</td>
<td>DOE/ NNSA / Los Alamos Nat Lab</td>
<td>Cray XE6 8-core 2.4 GHz</td>
<td>USA</td>
<td>107,152</td>
<td>.817</td>
<td>79</td>
</tr>
</tbody>
</table>
## 36rd List: The TOP10

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nat. SuperComputer Center in Tianjin</td>
<td>NUDT YH Cluster, X5670 2.93Ghz 6C, NVIDIA GPU</td>
<td>China</td>
<td>186,368</td>
<td>2.57</td>
<td>55</td>
<td>4.04</td>
<td>636</td>
</tr>
<tr>
<td>2</td>
<td>DOE / OS Oak Ridge Nat Lab</td>
<td>Jaguar / Cray Cray XT5 sixCore 2.6 GHz</td>
<td>USA</td>
<td>224,162</td>
<td>1.76</td>
<td>75</td>
<td>7.0</td>
<td>251</td>
</tr>
<tr>
<td>3</td>
<td>Nat. Supercomputer Center in Shenzhen</td>
<td>Nebulea / Dawning / TC3600 Blade, Intel X5650, Nvidia C2050 GPU</td>
<td>China</td>
<td>120,640</td>
<td>1.27</td>
<td>43</td>
<td>2.58</td>
<td>493</td>
</tr>
<tr>
<td>4</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>Tusbame 2.0 HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU</td>
<td>Japan</td>
<td>73,278</td>
<td>1.19</td>
<td>52</td>
<td>1.40</td>
<td>850</td>
</tr>
<tr>
<td>5</td>
<td>DOE/SC/LBNL/NERSC</td>
<td>Hopper, Cray XE6 12-core 2.1 GHz</td>
<td>USA</td>
<td>153,408</td>
<td>1.054</td>
<td>82</td>
<td>2.91</td>
<td>362</td>
</tr>
<tr>
<td>6</td>
<td>Commissariat a l’Energie Atomique (CEA)</td>
<td>Tera-100 Bull bullx super-node S6010/S6030</td>
<td>France</td>
<td>138,368</td>
<td>1.050</td>
<td>84</td>
<td>4.59</td>
<td>229</td>
</tr>
<tr>
<td>7</td>
<td>DOE / NNSA Los Alamos Nat Lab</td>
<td>Roadrunner / IBM BladeCenter QS22/LS21</td>
<td>USA</td>
<td>122,400</td>
<td>1.04</td>
<td>76</td>
<td>2.35</td>
<td>446</td>
</tr>
<tr>
<td>8</td>
<td>NSF / NICS / U of Tennessee</td>
<td>kraken/ Cray Cray XT5 sixCore 2.6 GHz</td>
<td>USA</td>
<td>98,928</td>
<td>.831</td>
<td>81</td>
<td>3.09</td>
<td>269</td>
</tr>
<tr>
<td>9</td>
<td>Forschungszentrum Juelich (FZJ)</td>
<td>Jugene / IBM Blue Gene/P Solution</td>
<td>Germany</td>
<td>294,912</td>
<td>.825</td>
<td>82</td>
<td>2.26</td>
<td>365</td>
</tr>
<tr>
<td>10</td>
<td>DOE/ NNSA / Los Alamos Nat Lab</td>
<td>Cray XE6 8-core 2.4 GHz</td>
<td>USA</td>
<td>107,152</td>
<td>.817</td>
<td>79</td>
<td>2.95</td>
<td>277</td>
</tr>
</tbody>
</table>
Performance Development in Top500

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


SUM

Gordon Bell Winners

N=1

N=500
## Pflop/s Club (11 systems; Peak)

<table>
<thead>
<tr>
<th>Name</th>
<th>Peak Pflop/s</th>
<th>“Linpack” Pflop/s</th>
<th>Country</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tianhe-1A</td>
<td>4.70</td>
<td>2.57</td>
<td>China</td>
<td>NUDT: Hybrid Intel/Nvidia/Self</td>
</tr>
<tr>
<td>Nebula</td>
<td>2.98</td>
<td>1.27</td>
<td>China</td>
<td>Dawning: Hybrid Intel/Nvidia/IB</td>
</tr>
<tr>
<td>Jaguar</td>
<td>2.33</td>
<td>1.76</td>
<td>US</td>
<td>Cray: AMD/Self</td>
</tr>
<tr>
<td>Tsubame 2.0</td>
<td>2.29</td>
<td>1.19</td>
<td>Japan</td>
<td>HP: Hybrid Intel/Nvidia/IB</td>
</tr>
<tr>
<td>RoadRunner</td>
<td>1.38</td>
<td>1.04</td>
<td>US</td>
<td>IBM: Hybrid AMD/Cell/IB</td>
</tr>
<tr>
<td>Hopper</td>
<td>1.29</td>
<td>1.054</td>
<td>US</td>
<td>Cray: AMD/Self</td>
</tr>
<tr>
<td>Tera-100</td>
<td>1.25</td>
<td>1.050</td>
<td>France</td>
<td>Bull: Intel/IB</td>
</tr>
<tr>
<td>Mole-8.5</td>
<td>1.14</td>
<td>0.207</td>
<td>China</td>
<td>CAS: Hybrid Intel/Nvidia/IB</td>
</tr>
<tr>
<td>Kraken</td>
<td>1.02</td>
<td>0.831</td>
<td>US</td>
<td>Cray: AMD/Self</td>
</tr>
<tr>
<td>Cielo</td>
<td>1.02</td>
<td>0.817</td>
<td>US</td>
<td>Cray: AMD/Self</td>
</tr>
<tr>
<td>JuGene</td>
<td>1.00</td>
<td>0.825</td>
<td>Germany</td>
<td>IBM: BG-P/Self</td>
</tr>
</tbody>
</table>
Factors that Necessitate Redesign of Our Software

- Steepness of the ascent from terascale to petascale to exascale
- Extreme parallelism and hybrid design
  - Preparing for million/billion way parallelism
- Tightening memory/bandwidth bottleneck
  - Limits on power/clock speed implication on multicore
  - Reducing communication will become much more intense
  - Memory per core changes, byte-to-flop ratio will change
- Necessary Fault Tolerance
  - MTTF will drop
  - Checkpoint/restart has limitations

Software infrastructure does not exist today
Commodity plus Accelerators

**Commodity**

Intel Xeon
- 8 cores
- 3 GHz
- 8*4 ops/cycle
- 96 Gflop/s (DP)

**Accelerator (GPU)**

Nvidia C2050 “Fermi”
- 448 “Cuda cores”
- 1.15 GHz
- 448 ops/cycle
- 515 Gflop/s (DP)

Interconnect
- PCI Express
- 512 MB/s to 32GB/s
- 8 MW – 512 MW
Major Changes to Software

- **Must rethink the design of our software**
  - Another disruptive technology
    - Similar to what happened with cluster computing and message passing
  - Rethink and rewrite the applications, algorithms, and software

- **Numerical libraries for example will change**
  - For example, both LAPACK and ScaLAPACK will undergo major changes to accommodate this
Five Important Software Features to Consider When Computing at Scale

1. Effective Use of Many-Core and Hybrid architectures
   - Break fork-join parallelism
   - Dynamic Data Driven Execution
   - Block Data Layout

2. Exploiting Mixed Precision in the Algorithms
   - Single Precision is 2X faster than Double Precision
   - With GP-GPUs 10x
   - Power saving issues

3. Self Adapting / Auto Tuning of Software
   - Too hard to do by hand

4. Fault Tolerant Algorithms
   - With 1,000,000’s of cores things will fail

5. Communication Reducing Algorithms
   - For dense computations from $O(n \log p)$ to $O(\log p)$ communications
   - Asynchronous iterations
   - GMRES k-step compute ($x, Ax, A^2x, \ldots A^kx$)
• Fork-join, bulk synchronous processing
Parallel Tasks in LU/LLᵀ/QR

- Break into smaller tasks and remove dependencies

* LU does block pair wise pivoting
Objectives

- High utilization of each core
- Scaling to large number of cores
- Shared or distributed memory

Methodology

- Dynamic DAG scheduling
- Explicit parallelism
- Implicit communication
- Fine granularity / block data layout

Arbitrary DAG with dynamic scheduling

PLASMA: Parallel Linear Algebra s/w for Multicore Architectures
LU - Intel64 - 16 cores

DGETRF - Intel64 Xeon quad-socket quad-core (16 cores)
theoretical peak 153.6 Gflop/s

Gflop/s

Matrix size

DGEMM
PLASMA
LAPACK
PLASMA Performance (QR, 48 cores)

ISTANBUL AMD 8 socket 6 core (48 cores) @2.8GHz
Challenges of using GPUs

- High levels of parallelism
  Many GPU cores
  [ e.g. Tesla C2050 (Fermi) has 448 CUDA cores ]

- Hybrid/heterogeneous architectures
  Match algorithmic requirements to architectural strengths
  [ e.g. small, non-parallelizable tasks to run on CPU, large and parallelizable on GPU ]

- Compute vs communication gap
  Exponentially growing gap; persistent challenge
  [ Processor speed improves 59%, memory bandwidth 23%, latency 5.5% ]
  [ on all levels, e.g. a GPU Tesla C1070 (4 x C1060) has compute power of $O(1,000)$ Gflop/s but GPUs communicate through the CPU using $O(1)$ GB/s connection ]
Matrix Algebra on GPU and Multicore Architectures

- **MAGMA**: a new generation linear algebra (LA) libraries to achieve the fastest possible time to an accurate solution on hybrid/heterogeneous architectures, starting with current multicore+MultiGPU systems
  
  **Homepage**: [http://icl.cs.utk.edu/magma/](http://icl.cs.utk.edu/magma/)

- **MAGMA & LAPACK**
  - **MAGMA** - based on LAPACK and extended for hybrid systems (multi-GPUs + multicore systems);
  - **MAGMA** - designed to be similar to LAPACK in functionality, data storage and interface, in order to allow scientists to effortlessly port any of their LAPACK-relying software components to take advantage of the new architectures
  - **MAGMA** - to leverage years of experience in developing open source LA software packages and systems like LAPACK, ScaLAPACK, BLAS, ATLAS as well as the newest LA developments (e.g. communication avoiding algorithms) and experiences on homogeneous multicores (e.g. PLASMA)

- **Support**
  - NSF, Microsoft, NVIDIA [CUDA Center of Excellence at UTK on the development of Linear Algebra Libraries for CUDA-based Hybrid Architectures]

- **MAGMA developers**
  - University of Tennessee, Knoxville; University of California, Berkeley; University of Colorado, Denver
Hybridization methodology

- **MAGMA** uses **HYBRIDIZATION** methodology based on
  - Representing linear algebra algorithms as collections of TASKS and DATA DEPENDENCIES among them
  - Properly SCHEDULING the tasks' execution over the multicore and the GPU hardware components

- Successfully applied to fundamental linear algebra algorithms
  - One and two-sided factorizations and solvers
  - Iterative linear and eigen-solvers

- Faster, cheaper, better?
  - High-level
  - Leveraging prior developments
  - Exceeding in performance homogeneous solutions

Hybrid CPU+GPU algorithms (small tasks for multicores and large tasks for GPUs)
Linear solvers on Fermi

MAGMA LU-based solvers on Fermi (C2050)

FERMI
Tesla C2050: 448 CUDA cores @ 1.15GHz
SP/DP peak is 1030 / 515 GFlop/s

- Direct solvers
  - Factor and solve in working precision
- Mixed Precision Iterative Refinement
  - Factor in single (i.e. the bulk of the computation in fast arithmetic) and use it as preconditioner in simple double precision iteration, e.g.
  \[ x_{i+1} = x_i + (L U_{SP})^{-1} P (b - A x_i) \]

- Similar results for Cholesky & QR
## New Release for SC2010 PLASMA 2.3

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear systems and least squares</td>
<td>LU, Cholesky, QR &amp; LQ</td>
</tr>
<tr>
<td>Mixed-precision linear systems</td>
<td>LU, Cholesky, QR</td>
</tr>
<tr>
<td><strong>Tall and skinny</strong> factorization</td>
<td>QR</td>
</tr>
<tr>
<td>Generation of the Q matrix</td>
<td>QR, LQ, tall and skinny QR</td>
</tr>
<tr>
<td>Explicit matrix inversion</td>
<td>Cholesky</td>
</tr>
<tr>
<td>Level 3 BLAS</td>
<td>GEMM, HEMM, HER2K, HERK, SYMM, SYR2K, SYRK, TRMM, TRSM (complete set)</td>
</tr>
<tr>
<td>In-place layout translations</td>
<td>CM, RM, CCRB, CRRB, RCRB, RRRB (all combinations)</td>
</tr>
</tbody>
</table>

### Features
- Covering four precisions: Z, C, D, S (and mixed-precision: ZC, DS)
- Static scheduling and dynamic scheduling with QUARK
- Support for Linux, MS Windows, Mac OS and AIX
# New Release for SC2010 MAGMA 1.0

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear systems and least squares</td>
<td>LU, Cholesky, QR &amp; LQ</td>
</tr>
<tr>
<td>Mixed-precision linear systems</td>
<td>LU, Cholesky, QR</td>
</tr>
<tr>
<td><strong>Eigenvalue and singular value problems</strong></td>
<td>Reductions to upper Hessenberg, bidiagonal, and tridiagonal forms</td>
</tr>
<tr>
<td>Generation of the Q matrix</td>
<td>QR, LQ, Hessenberg, bidiagonalization, and tridiagonalization</td>
</tr>
<tr>
<td>MAGMA BLAS</td>
<td>Subset of BLAS, critical for MAGMA performance for Tesla and Fermi</td>
</tr>
</tbody>
</table>

## Features

- Covering four precisions: Z, C, D, S (and mixed-precision: ZC, DS)
- Support for multicore and one NVIDIA GPU
- CPU and GPU interfaces
- Support for Linux and Mac OS
Summary

• **Major Challenges are ahead for extreme computing**
  - Parallelism
  - Hybrid
  - Fault Tolerance
  - Power
  - ... and many others not discussed here

• **We will need completely new approaches and technologies to reach the Exascale level**

• **This opens up many new opportunities for applied mathematicians and computer scientists**
Collaborators / Support

- **MAGMA** [Matrix Algebra on GPU and Multicore Architectures] team
  http://icl.cs.utk.edu/magma/

- **PLASMA** [Parallel Linear Algebra for Scalable Multicore Architectures] team
  http://icl.cs.utk.edu/plasma

- Collaborating partners

  University of Tennessee, Knoxville
  University of California, Berkeley
  University of Colorado, Denver

  University of Coimbra, Portugal
  INRIA, France (StarPU team)