November 2011: The TOP10

<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>
<th>Power [MW]</th>
<th>MFlops/Watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RIKEN Advanced Inst for Comp Sci</td>
<td>K computer Fujitsu SPARC64 VIIIfx + custom</td>
<td>Japan</td>
<td>705,024</td>
<td>10.5</td>
<td>93</td>
<td>12.7</td>
<td>830</td>
</tr>
<tr>
<td>2</td>
<td>Nat. Supercomputer Center in Tianjin</td>
<td>Tianhe-1A, NUDT Intel + Nvidia GPU + custom</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>3</td>
<td>DOE / OS Oak Ridge Nat Lab</td>
<td>Jaguar, Cray AMD + custom</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>4</td>
<td>Nat. Supercomputer Center in Shenzhen</td>
<td>Nebulae, Dawning Intel + Nvidia GPU + IB</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>5</td>
<td>GSIC Center, Tokyo Institute of Technology</td>
<td>Tsuname 2.0, HP Intel + Nvidia GPU + IB</td>
<td>Japan</td>
<td>73,278</td>
<td>1.19</td>
<td>52</td>
<td>1.40</td>
<td>865</td>
</tr>
<tr>
<td>6</td>
<td>DOE / NNSA LANL &amp; SNL</td>
<td>Golo, Cray AMD + custom</td>
<td>USA</td>
<td>142,272</td>
<td>1.11</td>
<td>81</td>
<td>3.98</td>
<td>279</td>
</tr>
<tr>
<td>7</td>
<td>NASA Ames Research Center/NAS</td>
<td>Pleiades SGI Altix ICE 8200EX/8400EX + IB</td>
<td>USA</td>
<td>111,104</td>
<td>1.09</td>
<td>83</td>
<td>4.10</td>
<td>265</td>
</tr>
<tr>
<td>8</td>
<td>DOE / OS Lawrence Berkeley Nat Lab</td>
<td>Hopper, Cray AMD + custom</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>9</td>
<td>Commissariat à l’Energie Atomique (CEA)</td>
<td>Tera-10, Bull Intel + IB</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>10</td>
<td>DOE / NNSA Los Alamos Nat Lab</td>
<td>Roadrunner, IBM AMD + Cell GPU + IB</td>
<td>USA</td>
<td>122,400</td>
<td>1.04</td>
<td>76</td>
<td>2.35</td>
<td>446</td>
</tr>
</tbody>
</table>

500 IT Service IBM Cluster, Intel + GigE USA 7,236 .051 53
Japanese K Computer

K computer Specifications

<table>
<thead>
<tr>
<th>CPU</th>
<th>Node</th>
<th>8 cores (2.0GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Performance</td>
<td>126GFlop/s</td>
</tr>
<tr>
<td></td>
<td>Architecture</td>
<td>SPARC V9 + HPC extension</td>
</tr>
<tr>
<td>Cache</td>
<td>L1(4KB) Cache</td>
<td>32KB/32KB</td>
</tr>
<tr>
<td></td>
<td>L2 Cache</td>
<td>16MB</td>
</tr>
<tr>
<td>Power</td>
<td>56W (typ. 30 C)</td>
<td></td>
</tr>
<tr>
<td>Mem. bandwidth</td>
<td>64GB/s</td>
<td></td>
</tr>
<tr>
<td>Config</td>
<td>1 CPU / Node</td>
<td></td>
</tr>
<tr>
<td>Memory Capacity</td>
<td>16GB (2GB/core)</td>
<td></td>
</tr>
<tr>
<td>System board(SB)</td>
<td>No. of nodes 4 nodes /SB</td>
<td></td>
</tr>
<tr>
<td>Rack</td>
<td>No. of SB</td>
<td>24 SB/track</td>
</tr>
<tr>
<td>System</td>
<td>Nodes/system</td>
<td>&gt; 80,000</td>
</tr>
</tbody>
</table>

CPU 128GFlops SPARC64 VIIIfx 8 Cores@2.0GHz
Node 128 GFlops 16GB Memory 64GB/s Memory bandwidth
System Board 512 GFlops 64 GB memory
Interconnect 6U Mesh/Torus
Performance 500GF/s for each link
No. of link 16 links/node
Additional feature HW barrier, reduction
Architecture Routing chip structure (no network switch box)
Cooling CPU, I/O7 Direct water cooling
Other parts Air cooling

New Linpack run with 705,024 cores at 10.51 Pflop/s (88,128 CPUs), 12.7 MW; 29.5 hours
Fujitsu to have a 100 Pflop/s system in 2014

China

First Chinese Supercomputer to use a Chinese Processor
- Sunway BlueLight MPP
- ShenWei SW1600 processor, 16 core, 65 nm, fabbed in China
- 125 Gflop/s peak
- #14 with 139,364 cores, .796 Pflop/s & 1.07 Pflop/s Peak
- Power Efficiency 741 Mflops/W

Coming soon, Loongson (Godson) processor
- 8-core, 65nm Loongson 3B processor runs at 1.05 GHz, with a peak performance of 128 Gflop/s
### 10+ Pflop/s Systems Planned in the States

- **DOE Funded, Titan at ORNL, Based on Cray design w/AMD & Nvidia accelerators,**
  - 20 Pflop/s, 2012
- **DOE Funded, Sequoia at Lawrence Livermore Nat. Lab, Based on IBM’s BG/Q,**
  - 20 Pflop/s, 2012
- **DOE Funded, BG/Q at Argonne National Lab, Based on IBM’s BG/Q,**
  - 10 Pflop/s, 2012
- **NSF Funded, Blue Waters at University of Illinois UC, Based Cray XE6/XK6 hybrid,**
  - 11.5 Pflop/s, 2012
- **NSF Funded, U of Texas, Austin, Based on Dell/Intel MIC,**
  - 10 Pflop/s, 2013

### Tianhe-1A

**Main configuration of TH-1A system**

- 7,168 compute nodes (YH-X5670-FEP)
  - 2 six-core CPU and 1 GPU per node
  - CPU: Xeon X5670 (Westmere)
    - Processor speed - 2.93GHz
  - GPU: nVIDIA M2050
    - Connected with GPU by PCI-E
  - 32GB memory per node
  - 2U height

\[
\text{Total: } 4,701,061 \text{ GFlops}
\]

\[
7168(\text{nodes}) \times 2(\text{CPU}) \times 2.93(\text{GHz}) \times 6(\text{Core}) \times 4
= 1.008 \text{PFlops}
\]

\[
7168(\text{nodes}) \times 1(\text{GPU}) \times 1.15(\text{GHz})^*448(\text{CUDA Cores})
= 3.892 \text{PFlops}
\]
#3 ORNL’s Jaguar System - Cray XT5

2.3 Pflop/s system with more than 224K processor cores using AMD’s 6 Core chip.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak performance</td>
<td>2.3 PF</td>
</tr>
<tr>
<td>System memory</td>
<td>300 TB</td>
</tr>
<tr>
<td>Disk space</td>
<td>10 PB</td>
</tr>
<tr>
<td>Disk bandwidth</td>
<td>240+ GB/s</td>
</tr>
<tr>
<td>Interconnect bandwidth</td>
<td>374 TB/s</td>
</tr>
</tbody>
</table>

• Upgrade of existing Jaguar Cray XT5
• Cray Linux Environment operating system
• Gemini interconnect
  • 3-D Torus
  • Globally addressable memory
  • Advanced synchronization features
• AMD Opteron 6200 processor (Interlagos)
• New accelerated node design using NVIDIA multi-core accelerators
  • 2011: 960 NVIDIA M2090 “Fermi” GPUs
  • 2012: 10-20 PF NVIDIA “Kepler” GPUs
• 10-20 PFlops peak performance
  • Performance based on available funds
• 600 TB DDR3 memory (2x that of Jaguar)

ORNL’s “Titan” System

- Titan Specs

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Nodes</td>
<td>18,688</td>
</tr>
<tr>
<td>Login &amp; I/O Nodes</td>
<td>512</td>
</tr>
<tr>
<td>Memory per node</td>
<td>32 GB + 6 GB</td>
</tr>
<tr>
<td>NVIDIA “Fermi”</td>
<td>665 GFlops</td>
</tr>
<tr>
<td># of Fermi chips</td>
<td>960</td>
</tr>
<tr>
<td>NVIDIA “Kepler”</td>
<td>&gt;1 TFlops</td>
</tr>
<tr>
<td>Opteron</td>
<td>2.2 GHz</td>
</tr>
<tr>
<td>Opteron performance</td>
<td>141 GFlops</td>
</tr>
<tr>
<td>Total Opteron Flops</td>
<td>2.6 PFlops</td>
</tr>
<tr>
<td>Disk Bandwidth</td>
<td>~ 1 TB/s</td>
</tr>
</tbody>
</table>

- Titan Specs

- 6-8 GB/s direct power...
**NSF Supercomputing Centers**

University of Illinois - Blue Waters will be the powerhouse of the National Science Foundation’s strategy to support supercomputers for scientists nationwide.

<table>
<thead>
<tr>
<th>T1</th>
<th>Blue Waters</th>
<th>NCSA/Illinois</th>
<th>10 Pflop/s peak; 1 Pflop/s sustained per second in 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>Kraken</td>
<td>NICS/U of Tennessee</td>
<td>1 Pflop/s peak per second</td>
</tr>
<tr>
<td></td>
<td>Ranger</td>
<td>TACC/U of Texas</td>
<td>504 Tflop/s peak per second</td>
</tr>
<tr>
<td>T3</td>
<td>Campuses across the U.S.</td>
<td>Several sites</td>
<td>50-100 Tflop/s peak per second</td>
</tr>
</tbody>
</table>

**Today’s Multicores**

99% of Top500 Systems Are Based on Multicore

- Sun Niagara2 (8 cores)
- IBM Power 7 (8 cores)
- IBM Cell (9 cores)
- Intel Westmere (10 cores)
- Intel Polaris [experimental] (80 cores)
- Fujitsu Venus (16 cores)
- IBM BG/Q (18 cores)
- AMD Interlagos (16 cores)
General Purpose CPU
Concurrency Trends

• Today
  ▪ Typical server node chip ~ 8 cores
  ▪ 1k node cluster ➔ 8,000 cores
  ▪ Laptop ~ 2 cores (low power)

• By 2020
  ▪ Typical server node chip ~400 cores
  ▪ 1k node cluster ➔ 400,000 cores
  ▪ Laptop ~ 100 cores (low power)

Assuming continuation of 58% historical density improvement per year

Future Computer Systems

• Most likely be a hybrid design
• Think standard multicore chips and accelerator (GPUs)
  • Today accelerators are attached
  • Next generation more integrated
  • Intel’s “Knights Corner” and “Knights Ferry” to come.
    ▪ 48 x86 cores
• AMD’s Fusion in 2011 - 2013
  ▪ Multicore with embedded graphics ATI
• Nvidia’s Project Denver plans to develop an integrated chip using ARM architecture in 2013.
Evolution of GPUs

**GPUs**: excelling in graphics rendering

| Scene model | Streams of data | Graphics pipelined computation | Final image |

Repeated fast over and over: e.g. TV refresh rate is 30 fps; limit is 60 fps

This type of computation:
- Requires **enormous computational power**
- Allows for **high parallelism**
- Needs **high bandwidth vs low latency**

(As low latencies can be compensated with deep graphics pipeline)

Obviously, this pattern of computation is common with many other applications

Challenges of using GPUs

- **High levels of parallelism**
  - Many GPU cores, serial kernel execution
    - e.g. 240 in the Nvidia Tesla; up to 512 in Fermi - to have concurrent kernel execution

- **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
Moore’s Law is Alive and Well

![Graph showing exponential growth in transistors over time.]

Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanović

Slide from Kathy Yelick

But Clock Frequency Scaling Replaced by Scaling Cores / Chip

![Graph showing a decrease in frequency growth rate.]

15 Years of exponential growth ~2x year has ended

Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanović

Slide from Kathy Yelick
Performance Has Also Slowed, Along with Power

A hardware issue just became a software problem

Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanović
Slide from Kathy Yelick

Power Cost of Frequency

- Power $\propto$ Voltage$^2 \times$ Frequency (V$^2$F)
- Frequency $\propto$ Voltage
- Power $\propto$ Frequency$^3$

<table>
<thead>
<tr>
<th>Cores</th>
<th>V</th>
<th>Freq</th>
<th>Perf</th>
<th>Power</th>
<th>PE (Relative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superscalar</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>&quot;New&quot; Superscalar</td>
<td>1X</td>
<td>1.5X</td>
<td>1.5X</td>
<td>3.3X</td>
<td>0.45X</td>
</tr>
</tbody>
</table>
Power Cost of Frequency

- Power $\propto$ Voltage$^2 \times$ Frequency  \( (V^2F) \)
- Frequency $\propto$ Voltage
- Power $\propto$ Frequency$^3$

<table>
<thead>
<tr>
<th></th>
<th>Cores</th>
<th>V</th>
<th>Freq</th>
<th>Perf</th>
<th>Power</th>
<th>PE (Thousand)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superscalar</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>&quot;New&quot; Superscalar</td>
<td>1X</td>
<td>1.5X</td>
<td>1.5X</td>
<td>3.3X</td>
<td>0.45X</td>
<td></td>
</tr>
<tr>
<td>Multicore</td>
<td>2X</td>
<td>0.75X</td>
<td>0.75X</td>
<td>1.5X</td>
<td>0.8X</td>
<td>1.88X</td>
</tr>
</tbody>
</table>

(Bigger # is better)

50% more performance with 20% less power
Preferable to use multiple slower devices, than one superfast device

Moore’s Law Reinterpreted

- Number of cores per chip doubles every 2 year, while clock speed decreases (not increases).
  - Need to deal with systems with millions of concurrent threads
    - Future generation will have billions of threads!
  - Need to be able to easily replace inter-chip parallelism with intro-chip parallelism
- Number of threads of execution doubles every 2 year
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

Exascale Computing

- Exascale systems are likely feasible by 2017±2
- 10-100 Million processing elements (cores or mini-cores) with chips perhaps as dense as 1,000 cores per socket, clock rates will grow more slowly
- 3D packaging likely
- Large-scale optics based interconnects
- 10-100 PB of aggregate memory
- Hardware and software based fault management
- Heterogeneous cores
- Performance per watt — stretch goal 100 GF/watt of sustained performance $\Rightarrow \gg 10 - 100$ MW Exascale system
- Power, area and capital costs will be significantly higher than for today’s fastest systems

Google: exascale computing study
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  www.exascale.org

Conclusions

• For the last decade or more, the research investment strategy has been overwhelmingly biased in favor of hardware.

• This strategy needs to be rebalanced - barriers to progress are increasingly on the software side.

• Moreover, the return on investment is more favorable to software.
  ▪ Hardware has a half-life measured in years, while software has a half-life measured in decades.

• High Performance Ecosystem out of balance
  ▪ Hardware, OS, Compilers, Software, Algorithms, Applications
    ▪ No Moore’s Law for software, algorithms and applications
Why Fast Machines Run Slow

- **Latency**
  - Waiting for access to memory or other parts of the system

- **Overhead**
  - Extra work that has to be done to manage program concurrency and parallel resources the real work you want to perform

- **Starvation**
  - Not enough work to do due to insufficient parallelism or poor load balancing among distributed resources

- **Contention**
  - Delays due to fighting over what task gets to use a shared resource next. Network bandwidth is a major constraint.

Latency in a Single System

![Graph showing latency in a single system](image)

**THE WALL**
Memory hierarchy

- Typical latencies for today’s technology

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Processor clocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>1</td>
</tr>
<tr>
<td>L1 cache</td>
<td>2-3</td>
</tr>
<tr>
<td>L2 cache</td>
<td>6-12</td>
</tr>
<tr>
<td>L3 cache</td>
<td>14-40</td>
</tr>
<tr>
<td>Near memory</td>
<td>100-300</td>
</tr>
<tr>
<td>Far memory</td>
<td>300-900</td>
</tr>
<tr>
<td>Remote memory</td>
<td>$O(10^3)$</td>
</tr>
<tr>
<td>Message-passing</td>
<td>$O(10^3)$-$O(10^4)$</td>
</tr>
</tbody>
</table>

Most programs have a high degree of locality in their accesses:
- spatial locality: accessing things nearby previous accesses
- temporal locality: reusing an item that was previously accessed

Memory hierarchy tries to exploit locality

<table>
<thead>
<tr>
<th>Speed</th>
<th>1ns</th>
<th>10ns</th>
<th>100ns</th>
<th>10ms</th>
<th>10sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>B</td>
<td>KB</td>
<td>MB</td>
<td>GB</td>
<td>TB</td>
</tr>
</tbody>
</table>
Percentage of peak

- A rule of thumb that often applies
  - A contemporary RISC processor, for a spectrum of applications, delivers (i.e., sustains) 10% of peak performance

- There are exceptions to this rule, in both directions

- Why such low efficiency?

- There are two primary reasons behind the disappointing percentage of peak
  - IPC (in)efficiency
  - Memory (in)efficiency

Different Architectures

- **Parallel computing**: single systems with many processors working on same problem
- **Distributed computing**: many systems loosely coupled by a scheduler to work on related problems
- **Grid Computing**: many systems tightly coupled by software, perhaps geographically distributed, to work together on single problems or on related problems
Types of Parallel Computers

- The simplest and most useful way to classify modern parallel computers is by their memory model:
  - shared memory
  - distributed memory

Shared vs. Distributed Memory

Shared memory - single address space. All processors have access to a pool of shared memory. (Ex: SGI Origin, Sun E10000)

Distributed memory - each processor has its own local memory. Must do message passing to exchange data between processors. (Ex: CRAY T3E, IBM SP, clusters)
Uniform memory access (UMA): Each processor has uniform access to memory. Also known as symmetric multiprocessors (Sun E10000)

Non-uniform memory access (NUMA): Time for memory access depends on location of data. Local access is faster than non-local access. Easier to scale than SMPs (SGI Origin)