Design of Parallel and High-Performance Computing

Fall 2017

Lecture: Introduction

Instructor: Torsten Hoefler & Markus Püschel

TA: Salvatore Di Girolamo
Goals of this lecture

- Motivate you!
- Trends
- High performance computing
- Programming models
- Course overview
Trends
What doubles ...?
How to increase the compute power?

Clock Speed!

Power Density (W/cm²)

- Sun’s Surface
- Rocket Nozzle
- Nuclear Reactor
- Hot Plate
- Pentium® Processors

Source: Intel®
How to increase the compute power?

Not an option anymore!
Clock Speed!

Source: Intel®

Power Density (W/cm²)


Source: Intel®
Evolutions of Processors (Intel)

CPU Frequency [GHz]

- Pentium
- Pentium Pro
- Pentium II
- Pentium III
- Pentium 4
- Core
- Nehalem
- Sandy Bridge
- Haswell

Evolutions of Processors (Intel)

Floating point peak performance [Gflop/s]
CPU Frequency [GHz]

Evolutions of Processors (Intel)

Floating point peak performance [Gflop/s]
CPU Frequency [GHz]

A more complete view

Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond
Exponential Growth of Computing
Twentieth through twenty first century

Can we do this today?
High-Performance Computing
High-Performance Computing (HPC)

- a.k.a. “Supercomputing”
- Question: define “Supercomputer”!
High-Performance Computing (HPC)

- a.k.a. “Supercomputing”

- **Question: define “Supercomputer”!**
  - “A supercomputer is a computer at the frontline of contemporary processing capacity--particularly speed of calculation.” (Wikipedia)
  - Usually quite expensive ($s and kWh) and big (space)

- **HPC is a quickly growing niche market**
  - Not all “supercomputers”, wide base
  - Important enough for vendors to specialize
  - Very important in research settings (up to 40% of university spending)
  
    - “Goodyear Puts the Rubber to the Road with High Performance Computing”
    - “Procter & Gamble: Supercomputers and the Secret Life of Coffee”
    - “Motorola: Driving the Cellular Revolution With the Help of High Performance Computing”
    - “Microsoft: Delivering High Performance Computing to the Masses”
Growth in Supercomputer Power

Logarithmic Plot

Required for Human Brain Neural Simulation for Uploading (2025)

1 Exaflop! ~2023?

TaihuLight, ~125 PF (2016)

Blue Waters, ~13 PF (2012)

Flops (floating point operations)

10^1 10^2 10^3 10^4 10^5 10^6 10^7 10^8 10^9 10^10 10^11 10^12 10^13 10^14 10^15 10^16 10^17 10^18 10^19 10^20 10^21

Year

Doubling time = 1.2 years

Source: www.singularity.com
Blue Waters in 2012
Simulating 1 second of human brain activity takes 82,944 processors

By Ryan Whitwam on August 5, 2013 at 1:34 pm  

21 Comments

Forschungszentrum Jülich in Germany have managed to simulate a single second of human brain activity in a very, very powerful computer.

The brain is a deviously complex biological computing device that even the fastest supercomputers in the world fail to emulate. Well, that's not entirely true anymore. Researchers at the Okinawa Institute of Technology Graduate University in Japan and
The Top500 List

- A benchmark, solve $Ax=b$
  - As fast as possible! $\rightarrow$ as big as possible 😊
  - Reflects some applications, not all, not even many
  - Very good historic data!

- Speed comparison for computing centers, states, countries, nations, continents 😞
  - Politicized (sometimes good, sometimes bad)
  - Yet, fun to watch
## The Top500 List (June 2017)

<table>
<thead>
<tr>
<th>Rank</th>
<th>System</th>
<th>Cores</th>
<th>Rmax  (TFlop/s)</th>
<th>Rpeak (TFlop/s)</th>
<th>Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRPC</td>
<td>10,649,600</td>
<td>93,014.6</td>
<td>125,435.9</td>
<td>15,371</td>
</tr>
<tr>
<td>2</td>
<td>Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P, NUDT</td>
<td>3,120,000</td>
<td>33,862.7</td>
<td>54,902.4</td>
<td>17,808</td>
</tr>
<tr>
<td>3</td>
<td>Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect, NVIDIA Tesla P100, Cray Inc.</td>
<td>361,760</td>
<td>19,590.0</td>
<td>25,326.3</td>
<td>2,272</td>
</tr>
<tr>
<td>4</td>
<td>Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x, Cray Inc.</td>
<td>560,640</td>
<td>17,590.0</td>
<td>27,112.5</td>
<td>8,209</td>
</tr>
<tr>
<td>5</td>
<td>Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom, IBM</td>
<td>1,572,864</td>
<td>17,173.2</td>
<td>20,132.7</td>
<td>7,890</td>
</tr>
<tr>
<td>6</td>
<td>Cori - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect, Cray Inc.</td>
<td>622,336</td>
<td>14,014.7</td>
<td>27,880.7</td>
<td>3,939</td>
</tr>
<tr>
<td>7</td>
<td>Oakforest-PACS - PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path, Fujitsu</td>
<td>556,104</td>
<td>13,554.6</td>
<td>24,913.5</td>
<td>2,719</td>
</tr>
<tr>
<td>8</td>
<td>K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect, Fujitsu</td>
<td>705,024</td>
<td>10,510.0</td>
<td>11,280.4</td>
<td>12,660</td>
</tr>
</tbody>
</table>
Top500: Trends

Source: Jack Dongarra
# Green Top500 List (June 2017)

<table>
<thead>
<tr>
<th>Rank</th>
<th>TOP500 Rank</th>
<th>System</th>
<th>Cores</th>
<th>Rmax (TFlop/s)</th>
<th>Power (kW)</th>
<th>Power Efficiency (GFlops/watts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61</td>
<td><strong>TSUBAME3.0</strong> - SGI ICE XA, IP139-SXM2, Xeon E5-2680v4 14C 2.4GHz, Intel Omni-Path, NVIDIA Tesla P100 SXM2, HPE GSIC Center, Tokyo Institute of Technology Japan</td>
<td>36,286</td>
<td>1,998.0</td>
<td>142</td>
<td>14.110</td>
</tr>
<tr>
<td>2</td>
<td>465</td>
<td><strong>kukai</strong> - ZettaScaler-1.6 GPGPU system, Xeon E5-2650Lv4 14C 1.7GHz, Infiniband FDR, NVIDIA Tesla P100, ExaScaler, Yahoo Japan Corporation Japan</td>
<td>10,080</td>
<td>460.7</td>
<td>33</td>
<td>14.046</td>
</tr>
<tr>
<td>3</td>
<td>148</td>
<td><strong>AIST AI Cloud</strong> - NEC 4U-8GPU Server, Xeon E5-2630Lv4 10C 1.8GHz, Infiniband EDR, NVIDIA Tesla P100 SXM2, NEC National Institute of Advanced Industrial Science and Technology Japan</td>
<td>23,400</td>
<td>961.0</td>
<td>76</td>
<td>12.681</td>
</tr>
<tr>
<td>4</td>
<td>305</td>
<td><strong>RAIDEN GPU subsystem</strong> - NVIDIA DGX-1, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100, Fujitsu Center for Advanced Intelligence Project, RIKEN Japan</td>
<td>11,712</td>
<td>635.1</td>
<td>60</td>
<td>10.603</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td><strong>Wilkes-2</strong> - Dell C4130, Xeon E5-2650v4 12C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100, Dell University of Cambridge United Kingdom</td>
<td>21,240</td>
<td>1,193.0</td>
<td>114</td>
<td>10.428</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td><strong>Piz Daint</strong> - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect, NVIDIA Tesla P100, Cray Inc., Swiss National Supercomputing Centre (CSCS) Switzerland</td>
<td>361,760</td>
<td>19,590.0</td>
<td>2,272</td>
<td>10.398</td>
</tr>
</tbody>
</table>
Piz Daint @ CSCS

HPC Applications: Scientific Computing

- **Most natural sciences are simulation driven or are moving towards simulation**
  - Theoretical physics (solving the Schrödinger equation, QCD)
  - Biology (Gene sequencing)
  - Chemistry (Material science)
  - Astronomy (Colliding black holes)
  - Medicine (Protein folding for drug discovery)
  - Meteorology (Storm/Tornado prediction)
  - Geology (Oil reservoir management, oil exploration)
  - and many more ... (even Pringles uses HPC)
HPC Applications: Commercial Computing

- **Databases, data mining, search**
  - Amazon, Facebook, Google

- **Transaction processing**
  - Visa, Mastercard

- **Decision support**
  - Stock markets, Wall Street, Military applications

- **Parallelism in high-end systems and back-ends**
  - Often throughput-oriented
  - Used equipment varies from COTS (Google) to high-end redundant mainframes (banks)
HPC Applications: Industrial Computing

- Aeronautics (airflow, engine, structural mechanics, electromagnetism)
- Automotive (crash, combustion, airflow)
- Computer-aided design (CAD)
- Pharmaceuticals (molecular modeling, protein folding, drug design)
- Petroleum (Reservoir analysis)
- Visualization (all of the above, movies, 3d)
What can faster computers do for us?

- **Solving bigger problems than we could solve before!**
  - E.g., Gene sequencing and search, simulation of whole cells, mathematics of the brain, ...
  - The size of the problem grows with the machine power
    - \( \rightarrow \) **Weak Scaling**

- **Solve today’s problems faster!**
  - E.g., large (combinatorial) searches, mechanical simulations (aircrafts, cars, weapons, ...)
  - The machine power grows with constant problem size
    - \( \rightarrow \) **Strong Scaling**
Towards the age of massive parallelism

- **Everything goes parallel**
  - Desktop computers get more cores
    - 2, 4, 8, soon dozens, hundreds?
  - Supercomputers get more PEs (cores, nodes)
    - > 10 million today
    - > 50 million on the horizon
    - 1 billion in a couple of years (after 2020)

- **Parallel Computing is inevitable!**

---

*Parallel vs. Concurrent computing*

Concurrent activities *may* be executed in parallel

Example:
- A1 starts at T1, ends at T2; A2 starts at T3, ends at T4
  - Intervals (T1, T2) and (T3, T4) may overlap!

Parallel activities:
- A1 is executed *while* A2 is running
  - Usually requires separate resources!
Programming Models
## Flynn’s Taxonomy

<table>
<thead>
<tr>
<th>SISD</th>
<th>SIMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Serial Computer (nearly extinct)</td>
<td>Vector Machines or Extensions (very common)</td>
</tr>
<tr>
<td>MISD</td>
<td>MIMD</td>
</tr>
<tr>
<td>Redundant Execution (fault tolerance)</td>
<td>Multicore (ubiquitous)</td>
</tr>
</tbody>
</table>
## Parallel Resources and Programming

<table>
<thead>
<tr>
<th>Parallel Resource</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction-level parallelism</td>
<td>Compiler</td>
</tr>
<tr>
<td>Pipelining</td>
<td>(inline assembly)</td>
</tr>
<tr>
<td>VLIW</td>
<td>Hardware scheduling</td>
</tr>
<tr>
<td>Superscalar</td>
<td>Compiler (inline assembly)</td>
</tr>
<tr>
<td>SIMD operations</td>
<td>Intrinsics</td>
</tr>
<tr>
<td>Vector operations</td>
<td>Libraries</td>
</tr>
<tr>
<td>Instruction sequences</td>
<td>Compilers (very limited)</td>
</tr>
<tr>
<td>Multiprocessors</td>
<td>Expert programmers</td>
</tr>
<tr>
<td>Multicores</td>
<td>Parallel languages</td>
</tr>
<tr>
<td>Multithreading</td>
<td>Parallel libraries</td>
</tr>
<tr>
<td></td>
<td>Hints</td>
</tr>
</tbody>
</table>
Historic Architecture Examples

- **Systolic Array**
  - Data-stream driven (data counters)
  - Multiple streams for parallelism
  - Specialized for applications (reconfigurable)

- **Dataflow Architectures**
  - No program counter, execute instructions when all input arguments are available
  - Fine-grained, high overheads
    
    *Example: compute $f = (a+b) \cdot (c+d)$*

- Both come-back in FPGA computing
  - Interesting research opportunities!

![Diagram](Source: ni.com)
Parallel Architectures 101

- Uniform memory access
  - Today’s laptops
    - Core
    - Cache
    - Memory

- Non-uniform memory access
  - Today’s servers
    - Core
    - Cache
    - Memory

- Time-division multiplexing
  - Yesterday’s clusters
    - Core
    - Cache
    - Memory
    - Network

- Remote direct-memory access
  - Today’s clusters
    - Core
    - Memory

- ... and mixtures of those
Programming Models

- **Shared Memory Programming (SM)**
  - Shared address space
  - Implicit communication
  - Hardware for cache-coherent remote memory access
  - Cache-coherent Non Uniform Memory Access (cc NUMA)
  - Pthreads, OpenMP

- **(Partitioned) Global Address Space (PGAS)**
  - Remote Memory Access
  - Remote vs. local memory (cf. ncc-NUMA)

- **Distributed Memory Programming (DM)**
  - Explicit communication (typically messages)
  - Message Passing
MPI: de-facto large-scale prog. standard

Using MPI
Portable Parallel Programming with the Message-Passing Interface
third edition

William Gropp
Ewing Lusk
Anthony Skjellum

Using Advanced MPI
Modern Features of the Message-Passing Interface

William Gropp
Torsten Hoefler
Rajeev Thakur
Ewing Lusk

Basic MPI
Advanced MPI, including MPI-3
DPHPC Overview

- Concepts & Techniques
  - Locality
    - Caches
    - Memory hierarchy
  - Parallelism
    - Vector ISA
    - Shared memory
    - Distributed memory
  - Cache coherency
  - Memory models
    - Locks
    - Lock free
    - Wait free
    - Linearizability
  - Distributed algorithms
  - Group communications

- Models
  - Amdahl's and Gustafson's law
  - Memory
    - \( \alpha - \beta \)
  - PRAM
  - LogP
  - I/O complexity
  - Balance principles I
  - Little's Law
  - Balance principles II
  - Scheduling
# Schedule of Last Year

<table>
<thead>
<tr>
<th>k</th>
<th>Monday</th>
<th>Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>09/19: no lecture</td>
<td>09/22: [MPI Tutorial (white bg)]</td>
</tr>
<tr>
<td>1</td>
<td>09/26: Organization - Introduction (1pp) (6pp)</td>
<td>09/29: [Projects - Advanced MPI Tutorial]</td>
</tr>
<tr>
<td>2</td>
<td>10/03: Cache Coherence &amp; Memory Models (1pp) (6pp)</td>
<td>10/06: [Cache Organization - Introduction to OpenMP]</td>
</tr>
<tr>
<td>3</td>
<td>10/10: Memory Models (1pp) (6pp)</td>
<td>10/13: [Sequential Consistency + OpenMP Synchronization]</td>
</tr>
<tr>
<td>4</td>
<td>10/17: Linearizability (1pp) (6pp)</td>
<td>10/20: Linearizability</td>
</tr>
<tr>
<td>5</td>
<td>10/24: Languages and Locks (1pp) (6pp)</td>
<td>10/27: Locks</td>
</tr>
<tr>
<td>6</td>
<td>10/31: Amdahl's Law (1pp) (6pp) - Notes</td>
<td>11/03: Amdahl's Law</td>
</tr>
<tr>
<td>7</td>
<td>11/07: Project presentations</td>
<td>11/10: No recitation session</td>
</tr>
<tr>
<td>8</td>
<td>11/14: Roofline Model (1pp) (6pp) - Notes</td>
<td>11/17: Roofline Model</td>
</tr>
<tr>
<td>10</td>
<td>11/28: Locks and Lock-Free (1pp) (6pp)</td>
<td>12/01: SPIN Tutorial</td>
</tr>
<tr>
<td>11</td>
<td>12/05: Lock-Free and distributed memory (1pp) (6pp)</td>
<td>12/08: [Benchmarking - (paper)]</td>
</tr>
<tr>
<td>12</td>
<td>12/12: Guest lecture - Dr. Tobias Grosser</td>
<td>12/15: Network Models</td>
</tr>
<tr>
<td>13</td>
<td>12/19: Final Presentations</td>
<td></td>
</tr>
</tbody>
</table>
Related classes in the SE/PL focus

- **263-2910-00L Program Analysis**
  
  
  Spring 2017
  
  Lecturer: Martin Vechev

- **263-2300-00L How to Write Fast Numerical Code**
  
  
  Spring 2017
  
  Lecturer: Markus Püschel

- This list is not exhaustive!