Parallel Computing
- CENG577 – Spring 2019

Murat Manguoğlu
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
Parallel Computing

Sequential computing

Parallel computing
Motivations for Parallel Computing

- Faster time to solution
- Solving large problems
- New Architectures are increasingly parallel
Why do we care about history of science?

- *Die Geschichte der Wissenschaft is die Wissenschaft selbst* (The history of science is science itself) – Johann Wolfgang von Goethe 1749-1832

- In fact, if you read any scientific paper/thesis/work, you will notice it starts with the review of the literature (i.e. History) of the earlier work

- Important to learn from past mistakes and not repeat them
History of Parallel Computing/Processing

• Before Computers?

Image source: http://s3-eu-west-1.amazonaws.com/lookandlearn-preview/XM/XM10/XM10005/XM10005807.jpg
In Nature …

Image source:
More recently

1870 – Accountants

Burroughs Corp. D825 – 1962

4 processors + multiple memory modules using a crossbar switch\(^1\) - Multiple Instruction Multiple Data (MIMD)

\(^1\)https://wiki.cc.gatech.edu/folklore/index.php/Some_Burroughs_Transistor_Computers
ILLIAC IV – 1966~74

256 Processors – Single Instruction Multiple Data (SIMD)
Flynn’s Taxonomy of Computers
Michael J. Flynn - 1966

- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Single Data (MISD)
- Multiple Instruction Multiple Data (MIMD)
Trends in CPU transistor counts – Moore’s Law

source: http://en.wikipedia.org/wiki/Moore’s_law
Trends in CPU clock rate

Moore’s law →
Number of cores double every two years

Good days ... are gone

Programming models and another classification based on the address space

- **Process** based programming model, communicate via messages and can use
  - Shared address space (UMA and NUMA)
  - Distributed address space
  - or Shared/Distributed address space platforms
- **Thread** based programming model communicate via shared variables and can use
  - Usually shared address space platforms only
Limiting (hardware) factors in parallel computing

- Processor
- Memory
- Datapath
Difficulties in Parallel Computing

• Automatic parallelization is not efficient and developing/implementing parallel algorithms is not easy
  • Usually «the best sequential algorithm» ≠ «best parallel algorithm»
  • How to partition the problem? Load balancing? Communication/Synchronization? How to debug parallel code?

• Highly platform dependent → often low level

• Finding enough parallelism in a problem is not always possible

→ Amdhal’s Law and Gustafson’s Law
CENG577 - Parallel Computing

Scientific/Numerical

• Applications
• Algorithms and Kernels
• Programming Languages/models
• Tools (Debuggers, profilers...)
• Hardware

CENG478 - Introduction to Parallel Computing
Applications/Problems/Algorithms

- Embarrassingly parallel
- Fine grained
- Coarse grained
Applications/Problems/Algorithms

- CPU bound
- Memory bound
Levels of Parallelism

- Bit Level
- Instruction Level
- Data Level
- Task Level

Easier for the compiler/processor to handle

Explicit

Implicit
Applications of Parallel Computing

• It used to be the largest problems in the past
• Today the applications are everywhere, but still the largest problems have more impact and more challenging

example: George Smoot who won the Nobel Prize says:

«...Computing has played a big role from the very beginning of what is a long series of experiments, and the computing requirements have grown with each experiment...»

«...For five years, or probably 10 or 15 years, cosmology is going to stress large-scale computing in a serious way..»

Source: http://scidacreview.org/0704/html/interview.html
Applications of Parallel Computing

Uncovering Alzheimer's complex genetic networks

Two CyberShake hazard models for the Los Angeles region calculated on Blue Waters using a simple 1D earth model (left) and a more realistic 3D earth model (right). Seismic hazard estimates produced using the 3D earth model show lower near-fault intensities due to 3D scattering (1), much higher intensities in near-fault basins (2), higher intensities in the Los Angeles basins (3), and lower intensities in hard-rock areas (4).

Source: http://www.ncsa.illinois.edu/news/story/do_the_wave

And others: nanotechnology, environmental engineering, quantum computing, physics, biology, chemistry, mechanical engineering, social sciences, .....
**Motif/Dwarf: Common Computational Methods**

*Red Hot → Blue Cool*

<table>
<thead>
<tr>
<th></th>
<th>Embed</th>
<th>SPEC</th>
<th>DB</th>
<th>Games</th>
<th>ML</th>
<th>HPC</th>
<th>Health</th>
<th>Image</th>
<th>Speech</th>
<th>Music</th>
<th>Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Finite State Mach.</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>2</td>
<td>Combinational</td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>3</td>
<td>Graph Traversal</td>
<td></td>
<td>✔️</td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>4</td>
<td>Structured Grid</td>
<td></td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>5</td>
<td>Dense Matrix</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>6</td>
<td>Sparse Matrix</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>7</td>
<td>Spectral (FFT)</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>8</td>
<td>Dynamic Prog</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>9</td>
<td>N-Body</td>
<td></td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>10</td>
<td>MapReduce</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>11</td>
<td>Backtrack/ B&amp;B</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>12</td>
<td>Graphical Models</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>13</td>
<td>Unstructured Grid</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

**What do applications have in common?**

Source: [http://www.cs.berkeley.edu/~demmel/cs267_Spr12/](http://www.cs.berkeley.edu/~demmel/cs267_Spr12/)
# Applications and Kernels/Algorithms

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lattice gauge (QCD)</td>
<td>quantum mechanics</td>
<td>weather</td>
<td>CFD</td>
<td>geodsy</td>
<td>inverse problems</td>
<td>structures</td>
<td>device simulation</td>
<td>circuit simulation</td>
<td>electromagnetics</td>
</tr>
<tr>
<td>2</td>
<td>1 linear systems</td>
<td>2 least squares</td>
<td>3 nonlinear systems</td>
<td>4 eigenvalues/SDVs</td>
<td>5 fast transforms</td>
<td>6 rapid elliptic solvers</td>
<td>7 multigrid</td>
<td>8 stiff ODE</td>
<td>9 Monte Carlo</td>
<td>10 integral transforms</td>
</tr>
</tbody>
</table>

Figure 5. Intel’s RMS and how it maps down to functions that are more primitive. Of the five categories at the top of the figure, Computer Vision is classified as Recognition, Data Mining is Mining, and Rendering, Physical Simulation, and Financial Analytics are Synthesis. [Chen 2006]
FLOP/s as a metric

High Performance Computing (HPC) units are:
  Flop: floating point operation, usually double precision unless noted
  Flop/s: floating point operations per second
  Bytes: size of data (a double precision floating point number is 8)

Typical sizes are millions, billions, trillions…

<table>
<thead>
<tr>
<th>Unit</th>
<th>Flop/s</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mega</td>
<td>(10^6) flop/sec</td>
<td>(2^{20}) (\approx) (10^6) bytes</td>
</tr>
<tr>
<td>Giga</td>
<td>(10^9) flop/sec</td>
<td>(2^{30}) (\approx) (10^9) bytes</td>
</tr>
<tr>
<td>Tera</td>
<td>(10^{12}) flop/sec</td>
<td>(2^{40}) (\approx) (10^{12}) bytes</td>
</tr>
<tr>
<td>Peta</td>
<td>(10^{15}) flop/sec</td>
<td>(2^{50}) (\approx) (10^{15}) bytes</td>
</tr>
<tr>
<td>Exa</td>
<td>(10^{18}) flop/sec</td>
<td>(2^{60}) (\approx) (10^{18}) bytes</td>
</tr>
<tr>
<td>Zetta</td>
<td>(10^{21}) flop/sec</td>
<td>(2^{70}) (\approx) (10^{21}) bytes</td>
</tr>
<tr>
<td>Yotta</td>
<td>(10^{24}) flop/sec</td>
<td>(2^{80}) (\approx) (10^{24}) bytes</td>
</tr>
</tbody>
</table>

Current fastest (public) machine \(\sim 27\) Pflop/s

Up-to-date list at www.top500.org

This slide is from: http://www.cs.berkeley.edu/~demmel/cs267_Spr13/
The Top500 List

Listing the 500 most powerful computers in the world

Yardstick: Rmax of Linpack

- Solve Ax=b, dense problem, matrix is random
- Dominated by dense matrix-matrix multiply

Update twice a year:

- ISC’xy in June in Germany
- SCxy in November in the U.S.

All information available from the TOP500 web site at: www.top500.org

Green500: https://www.top500.org/green500/ (most energy efficient)

HPCG: https://www.top500.org/hpcg/ (instead of Linpack, uses Conjugate Gradient algorithm for solving sparse linear systems)

This slide is from: http://www.cs.berkeley.edu/~demmel/cs267_Spr13/
### Top 500 Supercomputers

<table>
<thead>
<tr>
<th>Rank</th>
<th>Supercomputer</th>
<th>Site</th>
<th>Country</th>
<th>Cores</th>
<th>Peak Performance</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Summit</td>
<td>DOE/SC/ORNL, USA</td>
<td>USA</td>
<td>2,282,544</td>
<td>143.5</td>
<td>11.1</td>
</tr>
<tr>
<td>2</td>
<td>Sierra</td>
<td>DOE/NNSA/LLNL, USA</td>
<td>USA</td>
<td>1,572,480</td>
<td>94.6</td>
<td>7.44</td>
</tr>
<tr>
<td>3</td>
<td>Sunway TaihuLight</td>
<td>NSCC in Wuxi, China</td>
<td>China</td>
<td>10,649,600</td>
<td>93.0</td>
<td>15.4</td>
</tr>
<tr>
<td>4</td>
<td>Tianhe-2A (Milkyway-2A)</td>
<td>NSCC Guangzhou, China</td>
<td></td>
<td>4,981,760</td>
<td>61.4</td>
<td>18.5</td>
</tr>
<tr>
<td>5</td>
<td>Piz Daint</td>
<td>CSCS, Switzerland</td>
<td>Switzerland</td>
<td>319,424</td>
<td>21.2</td>
<td>2.38</td>
</tr>
</tbody>
</table>

#### Performance Development

![Graph showing performance development over time](image_url)
What is the difference between Parallel and Distributed Computing?

They are highly related, the main difference is in distributed computing the processing unit/data is located physically quite away from each other hence the communication has high latency/low bandwidth and there is no shared clock. Therefore, the applications that work well differs significantly.
Course outline

• Introduction and a review of the architectural features of parallel processors, memory hierarchy, programming models, task decomposition and design of parallel algorithms

• Basic communication operations and their implementation on various network topologies and performance analysis of parallel algorithms

• Dense parallel matrix computations/algorithms

• Sparse parallel matrix computations/algorithms

• Graphs and graph partitioning

• Case studies: various real world applications
Requirements

• Some experience in parallel programming.

• 4 programming homework assignments using our departmental HPC platform
  (http://ceng.metu.edu.tr/slurm)

• Reports using LaTeX
Requirements

• Attendance, I will check the attendance randomly

• Grading:
  Homeworks ($\times 4$) 60%
  Midterm 20%
  Final 20%
  Total 100%
Textbooks

Introduction to Parallel Computing, by Grama, Gupta, Kumar, and Karypis, Addison Wesley. 2003

Introduction to High Performance Computing for Scientists and Engineers, by Hager and Wellein, Chapman & Hall/CRC Computational Science. 2010

The Sourcebook of Parallel Computing, Dongarra, Foster, Fox, and Gropp, Kaufmann. 2002


Homework policy

Discussion of ideas or concepts are allowed. However, when writing your code or solution you should not look at other people’s work.

If you use a source (online or offline) you are expected to cite it and state how you used it.
That's all Folks!