Intro To Parallel Computing

John Urbanic
Parallel Computing Scientist
Pittsburgh Supercomputing Center

Copyright 2020
Purpose of this talk

- This is the 50,000 ft. view of the parallel computing landscape. We want to orient you a bit before parachuting you down into the trenches to deal with MPI.

- This talk bookends our technical content along with the Outro to Parallel Computing talk. The Intro has a strong emphasis on hardware, as this dictates the reasons that the software has the form and function that it has. Hopefully our programming constraints will seem less arbitrary.

- The Outro talk can discuss alternative software approaches in a meaningful way because you will then have one base of knowledge against which we can compare and contrast.

- The plan is that you walk away with a knowledge of not just MPI, etc. but where it fits into the world of High Performance Computing.
FLOPS we need: Climate change analysis

<table>
<thead>
<tr>
<th>Simulations</th>
<th>Extreme data</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Cloud resolution, quantifying uncertainty, understanding tipping points, etc., will drive climate to exascale platforms</td>
<td>• “Reanalysis” projects need 100× more computing to analyze observations</td>
</tr>
<tr>
<td>• New math, models, and systems support will be needed</td>
<td>• Machine learning and other analytics are needed today for petabyte data sets</td>
</tr>
<tr>
<td></td>
<td>• Combined simulation/observation will empower policy makers and scientists</td>
</tr>
</tbody>
</table>

Courtesy Horst Simon, LBNL
Exascale combustion simulations

- Goal: 50% improvement in engine efficiency
- Center for Exascale Simulation of Combustion in Turbulence (ExaCT)
  - Combines simulation and experimentation
  - Uses new algorithms, programming models, and computer science

Courtesy Horst Simon, LBNL
Recent simulations achieve unprecedented scale of $65 \times 10^9$ neurons and $16 \times 10^{12}$ synapses.
'Nuff Said

There is an appendix with many more important exascale challenge applications at the end of our Outro To Parallel Computing talk.

And, many of you doubtless brought your own immediate research concerns. Great!
Moore's Law abandoned serial programming around 2004

![Graph showing the performance growth rate and the advent of multicore processors.](image-url)
Moore’s Law is not dead yet. Maybe.

**Intel process technology capabilities**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Size</td>
<td>90nm</td>
<td>65nm</td>
<td>45nm</td>
<td>32nm</td>
<td>22nm</td>
<td>16nm</td>
<td>14nm</td>
<td>10nm</td>
<td>7nm</td>
</tr>
<tr>
<td>Integration Capacity</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>...</td>
</tr>
</tbody>
</table>

Transistor for 90nm Process
Source: Intel

Influenza Virus
Source: CDC
At end of day, we keep using all those new transistors.

Moore’s Law – The number of transistors on integrated circuit chips (1971-2016)
Moore’s law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore’s law.
That Power and Clock Inflection Point in 2004… didn’t get better.

Fun fact: At 100+ Watts and <1V, currents are beginning to exceed 100A at the point of load!
Not a new problem, just a new scale...

Cray-2 with cooling tower in foreground, circa 1985
How to get same number of transistors to give us more performance without cranking up power?

Key is that

\[
\text{Performance} \approx \sqrt{\text{area}}
\]

Power = \(\frac{1}{4}\)

Performance = \(\frac{1}{2}\)
And how to get more performance from more transistors with the same power.

**RULE OF THUMB**

<table>
<thead>
<tr>
<th></th>
<th>Frequency Reduction</th>
<th>Power Reduction</th>
<th>Performance Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>15%</strong></td>
<td>45%</td>
<td>10%</td>
<td></td>
</tr>
</tbody>
</table>

A 15% Reduction In Voltage Yields

**SINGLE CORE**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>1</td>
</tr>
<tr>
<td>Voltage</td>
<td>1</td>
</tr>
<tr>
<td>Freq</td>
<td>1</td>
</tr>
<tr>
<td>Power</td>
<td>1</td>
</tr>
<tr>
<td>Perf</td>
<td>1</td>
</tr>
</tbody>
</table>

**DUAL CORE**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>2</td>
</tr>
<tr>
<td>Voltage</td>
<td>0.85</td>
</tr>
<tr>
<td>Freq</td>
<td>0.85</td>
</tr>
<tr>
<td>Power</td>
<td>1</td>
</tr>
<tr>
<td>Perf</td>
<td>~1.8</td>
</tr>
<tr>
<td>Processor</td>
<td>Year</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>Pentium III</td>
<td>1999</td>
</tr>
<tr>
<td>Pentium IV</td>
<td>2001</td>
</tr>
<tr>
<td>Core</td>
<td>2006</td>
</tr>
<tr>
<td>Nehalem</td>
<td>2008</td>
</tr>
<tr>
<td>Sandybridge</td>
<td>2011</td>
</tr>
<tr>
<td>Haswell</td>
<td>2013</td>
</tr>
<tr>
<td>KNC</td>
<td>2012</td>
</tr>
<tr>
<td>KNL</td>
<td>2016</td>
</tr>
<tr>
<td>Skylake</td>
<td>2017</td>
</tr>
</tbody>
</table>
Parallel Computing

One woman can make a baby in 9 months.

Can 9 women make a baby in 1 month?

But 9 women can make 9 babies in 9 months.

First two bullets are Brook’s Law. From *The Mythical Man-Month.*
Prototypical Application: Serial Weather Model
First Parallel Weather Modeling Algorithm: Richardson in 1917

Courtesy John Burkhardt, Virginia Tech
Weather Model: Shared Memory (OpenMP)

Four meteorologists in the same room sharing the map.

Fortran:

```fortran
!$omp parallel do
do i = 1, n
   a(i) = b(i) + c(i)
enddo
```

C/C++:

```c
#pragma omp parallel for
for(i=1; i<=n; i++)
   a[i] = b[i] + c[i];
```
V100 GPU and SM

Volta GV100 GPU with 85 Streaming Multiprocessor (SM) units

Volta GV100 SM

Rapid evolution continues with:

Turing
Ampere
Hopper
Weather Model: Accelerator (OpenACC)

1 meteorologists coordinating 1000 math savants using tin cans and a string.

```c
#pragma acc kernels
for (i=0; i<N; i++)  {
  double t = (double)((i+0.05)/N);
  pi += 4.0/(1.0+t*t);
}

__global__ void saxpy_kernel( float a, float* x, float* y, int n ){
  int i;
  i = blockIdx.x*blockDim.x + threadIdx.x;
  if( i <= n ) x[i] = a*x[i] + y[i];
}
```
call MPI_Send( numbertosend, 1, MPI_INTEGER, index, 10, MPI_COMM_WORLD, errcode)

call MPI_Recv( numb妥receive, 1, MPI_INTEGER, 0, 10, MPI_COMM_WORLD, status, errcode)

call MPI_Barrier(MPI_COMM_WORLD, errcode)

50 meteorologists using a telegraph.
The pieces fit like this…
Cores, Nodes, Processors, PEs?

- A "core" can run an independent thread of code. Hence the temptation to refer to it as a processor.

- “Processors” refer to a physical chip. Today these almost always have more than one core.

- “Nodes” is used to refer to an actual physical unit with a network connection; usually a circuit board or "blade" in a cabinet. These often have multiple processors.

- To avoid ambiguity, it is precise to refer to the smallest useful computing device as a Processing Element, or PE. On normal processors this corresponds to a core.

I will try to use the term PE consistently myself, but I may slip up. Get used to it as you will quite often hear all of the above terms used interchangeably where they shouldn’t be. Context usually makes it clear.
Many Levels and Types of Parallelism

- Vector (SIMD)
- Instruction Level (ILP)
  - Instruction pipelining
  - Superscaler (multiple instruction units)
  - Out-of-order
  - Register renaming
  - Speculative execution
  - Branch prediction
- Multi-Core (Threads)
- SMP/Multi-socket
- Accelerators: GPU & MIC
- Clusters
- MPPs

Also Important
- ASIC/FPGA/DSP
- RAID/IO

Compiler (not your problem)

OpenMP 4/5 can help!
MPPs (Massively Parallel Processors)

Distributed memory at largest scale. Shared memory at lower level.

**Summit (ORNL)**
- 122 PFlops Rmax and 187 PFlops Rpeak
- IBM Power 9, 22 core, 3GHz CPUs
- 2,282,544 cores
- NVIDIA Volta GPUs
- EDR Infiniband

**Sunway TaihuLight (NSC, China)**
- 93 PFlops Rmax and 125 PFlops Rpeak
- Sunway SW26010 260 core, 1.45GHz CPU
- 10,649,600 cores
- Sunway interconnect
<table>
<thead>
<tr>
<th>#</th>
<th>Site</th>
<th>Manufacturer</th>
<th>Computer</th>
<th>CPU Interconnect [Accelerator]</th>
<th>Cores</th>
<th>Rmax (Tflops)</th>
<th>Rpeak (Tflops)</th>
<th>Power (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DOE/SC/ORNL United States</td>
<td>IBM</td>
<td>Summit</td>
<td>Power9 22C 3.0 GHz Dual-rail Infiniband EDR NVIDIA V100</td>
<td>2,414,592</td>
<td>148,600</td>
<td>200,794</td>
<td>10.1</td>
</tr>
<tr>
<td>2</td>
<td>DOE/NNSA/LLNL United States</td>
<td>IBM</td>
<td>Sierra</td>
<td>Power9 3.1 GHz 22C Infiniband EDR NVIDIA V100</td>
<td>1,572,480</td>
<td>94,640</td>
<td>125,712</td>
<td>7.4</td>
</tr>
<tr>
<td>3</td>
<td>National Super Computer Center in Wuxi China</td>
<td>NRCPC</td>
<td>Sunway TaihuLight</td>
<td>Sunway SW26010 260C 1.45GHz</td>
<td>10,649,600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>National Super Computer Center in Guangzhou United States</td>
<td>NUDT</td>
<td>Tianhe-2 (MilkyWay-2)</td>
<td>Intel Xeon E5-2692 2.2 GHz TH Express-2 Intel Xeon Phi 31S1P</td>
<td>4,981,760</td>
<td>61,444</td>
<td>100,678</td>
<td>18.4</td>
</tr>
<tr>
<td>5</td>
<td>Texas Advanced Computing Center/Univ. of Texas United States</td>
<td>Dell</td>
<td>Frontera</td>
<td>Intel Xeon 8280 28C 2.7 GHz InfiniBand HDR</td>
<td>448,448</td>
<td>23,516</td>
<td>38,745</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Swiss National Supercomputing Centre (CSCS) Switzerland</td>
<td>Cray</td>
<td>Piz Daint Cray XC50</td>
<td>Xeon E5-2690 2.6 GHz Aries NVIDIA P100</td>
<td>387,872</td>
<td>21,230</td>
<td>27,154</td>
<td>2.4</td>
</tr>
<tr>
<td>7</td>
<td>DOE/NNSA/LANL/SNL United States</td>
<td>Cray</td>
<td>Trinity Cray XC40</td>
<td>Xeon E5-2698v3 2.3 GHz Aries Intel Xeon Phi 7250</td>
<td>979,072</td>
<td>20,158</td>
<td>41,461</td>
<td>7.6</td>
</tr>
<tr>
<td>8</td>
<td>ABCI Japan</td>
<td>Fujitsu</td>
<td>AI Bridging Cloud Primergy</td>
<td>Xeon 6148 20C 2.4GHz InfiniBand EDR NVIDIA V100</td>
<td>391,680</td>
<td>19,880</td>
<td>32,576</td>
<td>1.6</td>
</tr>
<tr>
<td>9</td>
<td>Leibniz Rechenzentrum Germany</td>
<td>Lenovo</td>
<td>SuperMUC-NG</td>
<td>Xeon 8174 24C 3.1GHz Intel Omni-Path NVIDIA V100</td>
<td>305,856</td>
<td>19,476</td>
<td>26,873</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DOE/NNSA/LLNL United States</td>
<td>IBM</td>
<td>Lassen</td>
<td>Power9 22C 3.1 GHz InfiniBand EDR NVIDIA V100</td>
<td>288,288</td>
<td>18,200</td>
<td>23,047</td>
<td></td>
</tr>
</tbody>
</table>
Networks

3 characteristics sum up the network:

• **Latency**
  
The time to send a 0 byte packet of data on the network

• **Bandwidth**
  
The rate at which a very large packet of information can be sent

• **Topology**
  
The configuration of the network that determines how processing units are directly connected.
Ethernet with Workstations
Complete Connectivity
Crossbar
Binary Tree
Fat Tree

http://www.unixer.de/research/fat-tree/
Other Fat Trees

Big Red @ IU

Jaguar @ ORNL

Odin @ IU

Atlas @ LLNL

Tsubame @ Tokyo Inst. of Tech

From Torsten Hoefler's Network Topology Repository at http://www.unixer.de/research/topologies/
A newer innovation in network design is the dragonfly topology, which benefits from advanced hardware capabilities like:

- High-Radix Switches
- Adaptive Routing
- Optical Links

Various 42 node Dragonfly configurations.

Purple links are optical, and blue are electrical.

Graphic from the excellent paper Design space exploration of the Dragonfly topology by Yee, Wilke, Bergman and Rumley.
Torus simply means that “ends” are connected. This means A is really connected to B and the cube has no real boundary.
Parallel IO (RAID…)

- There are increasing numbers of applications for which many PB of data need to be written.
- Checkpointing is also becoming very important due to MTBF issues (a whole ‘nother talk).
- Build a large, fast, reliable filesystem from a collection of smaller drives.
- Supposed to be transparent to the programmer.
- Increasingly mixing in SSD.
The Future Is Now!

Exascale Computing and you.
Today

- Pflops computing fully established with more than 500 machines
- The field is thriving
- Interest in supercomputing is now worldwide, and growing in many new markets
- Exascale projects in many countries and regions
Exascale?

$$\text{exa} = 10^{18} = 1,000,000,000,000,000,000 = \text{quintillion}$$

23,800 X

Cray Red Storm
2004
42 Tflops

or

833,000 X

NVIDIA K40
1.2 Tflops
Sustaining Performance Improvements

The diagram illustrates the sustained performance improvements over time. It shows a steady increase in performance from 1995 to 2020, with data points for the sum, the top-ranked system (#1), and the top 500 systems (net 500). The performance metrics are measured in MFlop/s, GFlop/s, TFlop/s, PFlop/s, and EFlop/s, with the graph showing a linear increase over the years.
USA: ECP by the Numbers

A seven-year, $1.7 B R&D effort that launched in 2016

- Six core DOE National Laboratories: Argonne, Lawrence Berkeley, Lawrence Livermore, Oak Ridge, Sandia, Los Alamos
  - Staff from most of the 17 DOE national laboratories take part in the project

- Three technical focus areas: Hardware and Integration, Software Technology, Application Development supported by a Project Management Office

- More than 100 top-notch R&D teams

- Hundreds of consequential milestones delivered on schedule and within budget since project inception

- 7 YEARS
- $1.7B
- 6 CORE DOE LABS
- 3 FOCUS AREAS
- 100 R&D TEAMS
- 1000 RESEARCHERS
The Plan

Pre-Exascale Systems

- **2012**: TITAN, Cray/AMD/NVIDIA
- **2016**: CORI, LBNL, Cray/Intel
- **2018**: SUMMIT, ORNL, IBM/NVIDIA

Future Exascale Systems

- **2020**: PERLMUTTER, LBNL, Cray/AMD/NVIDIA
- **2021–2023**: FRONTIER, ORNL, Cray/AMD

- **2012**: MIRA, ANL, IBM BG/Q
- **2016**: THETA, ANL, Intel/Cray
- **2018**: SIERRA, LLNL, IBM/NVIDIA
- **2020**: CRUSADES, LANL, Cray
- **2021–2023**: COLUMBUS, LANL, Intel/Cray
- **2021–2023**: EL CAPITAN, LLNL, Cray
<table>
<thead>
<tr>
<th>System</th>
<th>Performance</th>
<th>Power</th>
<th>Interconnect</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aurora (ANL)</td>
<td>&gt; 1 EF</td>
<td></td>
<td>100 GB/s Cray Slingshot Dragonfly</td>
<td>2 Intel Xeon CPU + 6 Intel Xe GPUs</td>
</tr>
<tr>
<td>El Capitan (LLNL)</td>
<td>&gt; 1.5 EF</td>
<td>30-40 MW</td>
<td>100 GB/s Cray Slingshot Dragonfly</td>
<td>AMD Epyc CPU + 4 Radeon GPUs</td>
</tr>
<tr>
<td>Frontier (ORNL)</td>
<td>&gt; 1.5 EF</td>
<td></td>
<td>100 GB/s Cray Slingshot Dragonfly</td>
<td>AMD Epyc CPU + 4 Radeon GPUs</td>
</tr>
<tr>
<td>Perlmutter (LBNL)</td>
<td></td>
<td></td>
<td>Cray Slingshot Dragonfly</td>
<td>2 AMD Epyc CPU + 4 Volta GPUs</td>
</tr>
</tbody>
</table>
Two Additional Boosts to Improve Flops/Watt and Reach Exascale Target

First boost: many-core/accelerator

Second Boost: 3D (2016 – 2020)

Third Boost: SiPh (2020 – 2024)

• We will be able to reach usable Exaflops for ~30 MW by 2021

Will any of the other technologies give additional boosts after 2025?

Courtesy Horst Simon, LBNL
It is not just “exaflops” – we are changing the whole computational model
Current programming systems have WRONG optimization targets

Old Constraints
• Peak clock frequency as primary limiter for performance improvement
• Cost: FLOPs are biggest cost for system: optimize for compute
• Concurrency: Modest growth of parallelism by adding nodes
• Memory scaling: maintain byte per flop capacity and bandwidth
• Locality: MPI+X model (uniform costs within node & between nodes)
• Uniformity: Assume uniform system performance
• Reliability: It’s the hardware’s problem

New Constraints
• Power is primary design constraint for future HPC system design
• Cost: Data movement dominates: optimize to minimize data movement
• Concurrency: Exponential growth of parallelism within chips
• Memory Scaling: Compute growing 2x faster than capacity or bandwidth
• Locality: must reason about data locality and possibly topology
• Heterogeneity: Architectural and performance non-uniformity increase
• Reliability: Cannot count on hardware protection alone

Fundamentally breaks our current programming paradigm and computing ecosystem

Adapted from John Shalf
End of Moore’s Law Will Lead to New Architectures

Non-von Neumann

ARCHITECTURE

von Neumann

NEUROMORPHIC

Cerebras WSE
1.2 Trillion transistors
46,225 mm² silicon

Largest GPU
21.1 Billion transistors
815 mm² silicon

Beyond CMOS

Courtesy Horst Simon, LBNL
It would only be the 6th paradigm.
We can do better. We have a role model.

- Straightforward extrapolation results in a real-time human brain scale simulation at about 1 - 10 Exaflop/s with 4 PB of memory.

- Current predictions envision Exascale computers in 2022+ with a power consumption of at best 20 - 30 MW.

- The human brain takes 20W.

- Even under best assumptions in 2020 our brain will still be a million times more power efficient.

Courtesy Horst Simon, LBNL
Why you should be (extra) motivated.

- This parallel computing thing is no fad.
- The laws of physics are drawing this roadmap.
- If you get on board (the right bus), you can ride this trend for a long, exciting trip.

Let’s learn how to use these things!
In Conclusion…

OpenMP

OpenACC

MPI