Intro To Parallel Computing

John Urbanic
Pittsburgh Supercomputing Center
Parallel Computing Scientist
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

Simulations
- Cloud resolution, quantifying uncertainty, understanding tipping points, etc., will drive climate to exascale platforms
- New math, models, and systems support will be needed

Extreme data
- “Reanalysis” projects need $100\times$ more computing to analyze observations
- Machine learning and other analytics are needed today for petabyte data sets
- Combined simulation/observation will empower policy makers and scientists

Courtesy Horst Simon, LBNL
Qualitative Improvement of Simulation with Higher Resolution (2011)
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
Warhead assessment and certification of a smaller nuclear stockpile

Predicting with confidence requires precise understanding of aging and individual life-extended weapons

- High accuracy in individual weapons calculations:
  - Advanced physical models
  - 3-D to resolve features and phenomena
  - Extreme resolution

- Uncertainty quantification:
  - Massive numbers of high-resolution 3-D simulations to explore impacts of small variations in individual devices

- Assessment process: All aspects will require exascale computing to support commitment never to return to underground testing

Modeling molten tantalum: 10M atoms required; modeling molten plutonium: 1000× more computing power

Self-steering (a data analysis challenge): Essential to minimize number of simulations

Courtesy Horst Simon, LBNL
Recent simulations achieve unprecedented scale of 65*10^9 neurons and 16*10^{12} synapses.

Courtesy Horst Simon, LBNL
Also important: We aren’t going to be left behind.
Waiting for Moore’s Law to save your serial code started getting bleak in 2004

Source: published SPECInt data
Moore’s Law is not at all dead…

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>
</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>11nm</td>
<td>8nm</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>
</tr>
</tbody>
</table>

Source: Intel

Transistor for 90nm Process

Source: Intel

Influenza Virus

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

Courtesy Horst Simon, LBNL
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

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

**A 15% Reduction In Voltage Yields**

**SINGLE CORE**

- Area = 1
- Voltage = 1
- Freq = 1
- Power = 1
- Perf = 1

**DUAL CORE**

- Area = 2
- Voltage = 0.85
- Freq = 0.85
- Power = 1
- Perf = ~1.8

Frequency Reduction: 15%
Power Reduction: 45%
Performance Reduction: 10%
Parallel Computing

One woman can make a baby in 9 months.

Can 9 woman 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 meterologists in the same room sharing the map.
Weather Model: Accelerator (OpenACC)

1 meterologist coordinating 1000 savants using tin cans and a string.
Weather Model: Distributed Memory (MPI)

50 meterologists using telegraphs.
The pieces fit like this...

OpenMP

OpenACC

MPI
Cores, Nodes, Processors, PEs?

- The most unambiguous way to refer to the smallest useful computing device is as a Processing Element, or PE.
- This is usually the same as a single core.
- "Processors" usually have more than one core – as per the previous list.
- "Nodes" is commonly used to refer to an actual physical unit, most commonly a circuit board or blade with a network connection. These often have multiple processors.

I will try to use the term PE consistently here, but I may slip up myself. Get used to it as you will quite often hear all of the above terms used interchangeably where they shouldn’t be.
Multi-socket Motherboards

- Dual and Quad socket boards are very common in the enterprise and HPC world.
- Less desirable in consumer world.
Shared-Memory Processing

Each processor can access the entire data space

- **Pros**
  - Easier to program
  - Amenable to automatic parallelism
  - Can be used to run large memory serial programs

- **Cons**
  - Expensive for a lot of cores
  - Difficult to implement on the hardware level
  - Processor count limited by contention/coherency (currently around 512)
  - Watch out for “NU” part of “NUMA”
Shared-Memory Processing at Extreme Scale

- Programming
  - OpenMP, Pthreads, Shmem

- Examples
  - All multi-socket motherboards
  - SGI UV (Blacklight!)
    - Intel Xeon 8 dual core processors linked by the UV interconnect
    - 4096 cores sharing 32 TB of memory
  - As big as it gets right now
Distributed Memory Machines

- Each node in the computer has a locally addressable memory space
- The computers are connected together via some high-speed network
  - Infiniband, Myrinet, Giganet, etc..

- Pros
  - Really large machines
  - Cheaper to build and run
  - Size limited only by gross physical considerations:
    - Room size
    - Cable lengths (10’s of meters)
    - Power/cooling capacity
    - Money!

- Cons
  - Harder to program
  - Data Locality
Clusters

System X (Virginia Tech)
• 1100 Dual 2.3 GHz PowerPC 970FX processors
• 4 GB ECC DDR400 (PC3200) RAM
• 80 GB S-ATA hard disk drive
• One Mellanox Cougar InfiniBand 4x HCA*
• Running Mac OS X

Thunderbird (Sandia National Labs)
• Dell PowerEdge Series Capacity Cluster
• 4096 dual 3.6 Ghz Intel Xeon processors
• 6 GB DDR-2 RAM per node
• 4x InfiniBand interconnect
MPPs (Massively Parallel Processors)

Distributed memory at largest scale. Often shared memory at lower level.

- **Sequoia (LLNL)**
  - 16.32475 petaflops Rmax and 20.13266 petaflops Rpeak
  - IBM Blue Gene/Q
  - 98,304 compute nodes
  - 1.6 million processor cores
  - 1.6 PB of memory

- **Titan (ORNL)**
  - AMD Opteron 6274 processors (Interlagos)
  - 560,640 cores
  - Gemini interconnect (3-D Torus)
  - Accelerated node design using NVIDIA multi-core accelerators
  - 20+ PFlops peak system performance

<table>
<thead>
<tr>
<th>Compute Nodes</th>
<th>18,688</th>
</tr>
</thead>
<tbody>
<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># of Fermi chips</td>
<td>960</td>
</tr>
<tr>
<td># of NVIDIA “Kepler” (2013)</td>
<td>14,592</td>
</tr>
<tr>
<td>Total System Memory</td>
<td>688 TB</td>
</tr>
<tr>
<td>Total System Peak Performance</td>
<td>20+ Petaflops</td>
</tr>
<tr>
<td>Liquid cooling at the cabinet level</td>
<td>Cray EcoPHLex</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
Fat Tree

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

From Torsten Hoefler's Network Topology Repository at http://www.unixer.de/research/topologies/
XT3 has Global Addressing hardware, and this helps to simulate shared memory. Torus means that “ends” are connected. This means A is really connected to B and the cube has no real boundary.
GPU Architecture - GK110 Kepler

From a document you should read if you are interested in this:
Intel’s MIC Approach

Since the days of RISC vs. CISC, Intel has mastered the art of figuring out what is important about a new processing technology, and saying “why can’t we do this in x86?”

The Intel Many Integrated Core (MIC) architecture is about large die, simpler circuit, much more parallelism, in the x86 line.
<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>National Super Computer Center in Guangzhou China</td>
<td>NRCPC</td>
<td>Sunway TaihuLight</td>
<td>Sunway SW26010 260C 1.45GHz</td>
<td>10,649,600</td>
<td>93,014</td>
<td>125,435</td>
<td>15.3</td>
</tr>
<tr>
<td>2</td>
<td>National Super Computer Center in Guangzhou China</td>
<td>NUDT</td>
<td>Tianhe-2 (MilkyWay)</td>
<td>Intel Xeon E5-2692 2.2 GHz TH Express-2 Intel Xeon Phi 3151P</td>
<td>3,120,000</td>
<td>33,862</td>
<td>54,902</td>
<td>17.8</td>
</tr>
<tr>
<td>3</td>
<td>DOE/SC/Oak Ridge National Laboratory United States</td>
<td>Cray</td>
<td>Titan Cray XK7</td>
<td>Opteron 6274 2.2 GHz Gemini NVIDIA K20x</td>
<td>560,640</td>
<td>17,590</td>
<td>27,112</td>
<td>8.2</td>
</tr>
<tr>
<td>4</td>
<td>DOE/NNSA/LLNL United States</td>
<td>IBM</td>
<td>Sequoia BlueGene/Q</td>
<td>Power BQC 1.6 GHz Custom</td>
<td>1,572,864</td>
<td>17,173</td>
<td>20,132</td>
<td>7.8</td>
</tr>
<tr>
<td>5</td>
<td>DOE/SC/LBNL/NERSC United States</td>
<td>Cray</td>
<td>Cori Cray XC40</td>
<td>Aries Intel Xeon Phi 7250</td>
<td>622,336</td>
<td>14,014</td>
<td>27,880</td>
<td>3.9</td>
</tr>
<tr>
<td>6</td>
<td>Joint Center for Advanced High Performance Computing Japan</td>
<td>Fujitsu</td>
<td>Oakforest Primergy</td>
<td>Intel OPA Intel Xeon Phi 7250</td>
<td>556,105</td>
<td>13,554</td>
<td>24,913</td>
<td>2.7</td>
</tr>
<tr>
<td>5</td>
<td>RIKEN Advanced Institute for Computational Science (AICS) Japan</td>
<td>Fujitsu</td>
<td>K Computer</td>
<td>SPARC64 VIIIfx 2.0 GHz Tofu</td>
<td>705,024</td>
<td>10,510</td>
<td>11,280</td>
<td>12.6</td>
</tr>
<tr>
<td>8</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>206,720</td>
<td>9,779</td>
<td>15,988</td>
<td>1.3</td>
</tr>
<tr>
<td>9</td>
<td>DOE/SC/Argonne National Laboratory United States</td>
<td>IBM</td>
<td>Mira BlueGene/Q</td>
<td>Power BQC 1.6 GHz Custom</td>
<td>786,432</td>
<td>8,586</td>
<td>10,066</td>
<td>3.9</td>
</tr>
<tr>
<td>10</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</td>
<td>301,056</td>
<td>8,100</td>
<td>11,078</td>
<td>4.2</td>
</tr>
</tbody>
</table>

OpenACC is a first class API!
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.
Today

- Pflops computing fully established with more than 50 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?

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
Trends with ends.

Source: Kogge and Shalf, IEEE CISE

Courtesy Horst Simon, LBNL
Two Additional Boosts to Improve Flops/Watt and Reach Exascale Target

First boost: many-core/accelerator

Second Boost: 3D (2016 – 2020)
- We will be able to reach usable Exaflops for ~20 MW by 2024
- But at what cost?
- Will any of the other technologies give additional boosts after 2025?

Third Boost: SiPh (2020 – 2024)

• We will be able to reach usable Exaflops for ~20 MW by 2024
• But at what cost?
• Will any of the other technologies give additional boosts after 2025?

Courtesy Horst Simon, LBNL
Power Issues by 2018

FLOPs will cost less than on-chip data movement!

Adapted from John Shalf

Courtesy Horst Simon, LBNL
Flops are free?

At exascale, >99% of power is consumed by moving operands across machine.

Does it make sense to focus on flops, or should we optimize around data movement?

To those that say the future will simply be Big Data:

“All science is either physics or stamp collecting.”

- Ernest Rutherford
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

TODAY

CMOS

TECHNOLOGY

Beyond CMOS

von Neumann

NEUROMORPHIC

QUANTUM COMPUTING

BEYOND CMOS

Beyond CMOS

Courtesy Horst Simon, LBNL
It would only be the 6th paradigm.
It has become a mantra of contemporary programming philosophy that developer hours are so much more valuable than hardware, that the best design compromise is to throw more hardware at slower code.

This might well be valid for some Java dashboard app used twice a week by the CEO. But this has spread and results in...

The common observation that a modern PC (or phone) seems to be more laggy than one from a few generations ago that had literally one thousandth the processing power.

Moore’s Law has been the biggest enabler (or more accurately rationalization) for this trend. If Moore’s Law does indeed end, then progress will require good programming.

No more garbage collecting, script languages. I am looking at you, Java, Python, Matlab.
We can do better. We have a role model.

- Straight forward 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!
Credits

- Horst Simon of LBNL
  - His many beautiful graphics are a result of his insightful perspectives
  - He puts his money where his mouth is: $2000 bet in 2012 that Exascale machine would not exist by end of decade

- Intel
  - Many datapoints flirting with NDA territory

- Top500.org
  - Data and tools

- Supporting cast:
  - Erich Strohmaier (LBNL)
  - Jack Dongarra (UTK)
  - Rob Leland (Sandia)
  - John Shalf (LBNL)
  - Scott Aronson (MIT)
  - Bob Lucas (USC-ISI)
  - John Kubiatowicz (UC Berkeley)
  - Dharmendra Modha and team (IBM)
  - Karlheinz Meier (Univ. Heidelberg)