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.
1st Theme

We need Exascale computing

We aren’t getting to Exascale without parallel

What does parallel computing look like

Where is this going
FLOPS we need: Climate change analysis

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

- “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
Recent simulations achieve unprecedented scale of $65 \times 10^9$ neurons and $16 \times 10^{12}$ synapses.
2nd Theme

We will have Exascale computing

You aren’t getting to Exascale without going very parallel

What does parallel computing look like

Where is this going
Moore's Law abandoned serial programming around 2004

**Diagram Description:**
- **SPECFINT Performance (log scale):** This axis represents the performance growth rate of SPECINT benchmarks over time.
- **Historic Performance Growth Rate:** This line shows the historical growth rate of performance improvement.
- **Advent of Multicore:** This denotes the point when multicore processors became prevalent.
- **12x:** This indicates a 12-fold increase in performance.
- **7 Years Behind:** This suggests a lag in performance growth compared to historical trends.

**Legend:**
- CPU92
- CPU95
- CPU2000
- CPU2006
- CPU2017

*Courtesy Liberty Computer Architecture Research Group*
Moore’s Law is not quite 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

50nm Transistor for 90nm Process

Source: Intel

100nm Influenza Virus

Source: CDC
But tradeoffs are stealing these gains.

The density and power levels on a state-of-the-art chip have forced designers to compensate by adding:

- error-correction circuitry
- redundancy
- read- and write-boosting circuitry for failing static RAM cells
- circuits to track and adapt to performance variations
- complicated memory hierarchies to handle multicore architectures.

All of those extra circuits add area. Some analysts have concluded that when you factor those circuits in, chips are no longer twice as dense from generation to generation. One such analysis suggests, the density improvement over the past three generations, from 2007 on, has been closer to 1.6 than 2.

And cost per transistor has gone up for the first time ever:

- 2012  20M  28nm transistors/dollar
- 2015  19M  16nm transistors/dollar
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.

The data visualization is available at OurWorldInData.org. There you find more visualizations and research on this topic. Licensed under CC-BY-SA by the author Max Ford.
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 ≈ \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>Frequency Reduction</th>
<th>Power Reduction</th>
<th>Performance Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>15%</td>
<td>45%</td>
<td>10%</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%
<table>
<thead>
<tr>
<th>Processor</th>
<th>Year</th>
<th>Vector</th>
<th>Bits</th>
<th>SP FLOPs / core / cycle</th>
<th>Cores</th>
<th>FLOPs/cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium III</td>
<td>1999</td>
<td>SSE</td>
<td>128</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Pentium IV</td>
<td>2001</td>
<td>SSE2</td>
<td>128</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Core</td>
<td>2006</td>
<td>SSE3</td>
<td>128</td>
<td>8</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Nehalem</td>
<td>2008</td>
<td>SSE4</td>
<td>128</td>
<td>8</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>Sandybridge</td>
<td>2011</td>
<td>AVX</td>
<td>256</td>
<td>16</td>
<td>12</td>
<td>192</td>
</tr>
<tr>
<td>Haswell</td>
<td>2013</td>
<td>AVX2</td>
<td>256</td>
<td>32</td>
<td>18</td>
<td>576</td>
</tr>
<tr>
<td>KNC</td>
<td>2012</td>
<td>AVX512</td>
<td>512</td>
<td>32</td>
<td>64</td>
<td>2048</td>
</tr>
<tr>
<td>KNL</td>
<td>2016</td>
<td>AVX512</td>
<td>512</td>
<td>64</td>
<td>72</td>
<td>4608</td>
</tr>
<tr>
<td>Skylake</td>
<td>2017</td>
<td>AVX512</td>
<td>512</td>
<td>96</td>
<td>28</td>
<td>2688</td>
</tr>
</tbody>
</table>
Putting It All Together

- Transistors (thousands)
- Single-Thread Performance (SpecINT x 10^3)
- Frequency (MHz)
- Typical Power (Watts)
- Number of Logical Cores

We will have Exascale computing

You will get there by going very parallel

What does parallel computing look like?

Where is this going
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
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];
```
1 meteorologists coordinating 1000 math savants using tin cans and a string.

#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( numbertoreceive, 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...

OpenMP

OpenACC

MPI
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

Compiler (not your problem)

OpenMP

OpenACC

MPI

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

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

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

To avoid this 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.
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/topologies/
Other Fat Trees

From Torsten Hoefler's Network Topology Repository at http://www.unixer.de/research/topologies/
3-D Torus (T3D – XT7…)

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>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>2,397,824</td>
<td>143,500</td>
<td>200,794</td>
<td>9.7</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</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</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>Swiss National Supercomputing Centre (CSCS) Switzerland</td>
<td>Cray</td>
<td>Piz Daint Cray X50</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>6</td>
<td>DOE/NNSA/LANL/SNL United States</td>
<td>Cray</td>
<td>Trinity Cray X40</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>7</td>
<td>AIST Japan</td>
<td>Fujitsu</td>
<td>AI Bridging Cloud Primergy</td>
<td>Xeon 6148 2.4GHz Intel Omni-Path</td>
<td>305,856</td>
<td>19,476</td>
<td>26,873</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Leibniz Rechenzentrum Germany</td>
<td>Lenovo</td>
<td>SuperMUC-NG</td>
<td>Xeon 8174 3.1GHz Infiniband EDR NVIDIA V100</td>
<td>1,572,864</td>
<td>17,173</td>
<td>20,132</td>
<td>7.8</td>
</tr>
<tr>
<td>9</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>10</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>
</tbody>
</table>
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.
4th Theme

We will have Exascale computing

You will get there by going very parallel

What does parallel computing look like?

Where is this going?
Today

- Pflops computing fully established with more than 270 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

![Graph showing sustained performance improvements over time](image-url)
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
One of the many groups established to enable this outcome (the Advanced Scientific Computing Advisory Committee) puts forward this list of 10 technical challenges:

- Energy efficient circuit, power and cooling technologies.
- High performance interconnect technologies.
- Advanced memory technologies to dramatically improve capacity and bandwidth.
- Scalable system software that is power and resilience aware.
- Data management software that can handle the volume, velocity and diversity of data.
- Programming environments to express massive parallelism, data locality, and resilience.
- Reformulating science problems and refactoring solution algorithms for exascale.
- Ensuring correctness in the face of faults, reproducibility, and algorithm verification.
- Mathematical optimization and uncertainty quantification for discovery, design, and decision.
- Software engineering and supporting structures to enable scientific productivity.
It is not just “exaflops” – we are changing the whole computational model
Current programming systems have WRONG optimization targets

<table>
<thead>
<tr>
<th>Old Constraints</th>
<th>New Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Peak clock frequency as primary limiter for performance improvement</td>
<td>• Power is primary design constraint for future HPC system design</td>
</tr>
<tr>
<td>• Cost: FLOPs are biggest cost for system: optimize for compute</td>
<td>• Cost: Data movement dominates: optimize to minimize data movement</td>
</tr>
<tr>
<td>• Concurrency: Modest growth of parallelism by adding nodes</td>
<td>• Concurrency: Exponential growth of parallelism within chips</td>
</tr>
<tr>
<td>• Memory scaling: maintain byte per flop capacity and bandwidth</td>
<td>• Memory Scaling: Compute growing 2x faster than capacity or bandwidth</td>
</tr>
<tr>
<td>• Locality: MPI+X model (uniform costs within node &amp; between nodes)</td>
<td>• Locality: must reason about data locality and possibly topology</td>
</tr>
<tr>
<td>• Uniformity: Assume uniform system performance</td>
<td>• Heterogeneity: Architectural and performance non-uniformity increase</td>
</tr>
<tr>
<td>• Reliability: It’s the hardware’s problem</td>
<td>• Reliability: Cannot count on hardware protection alone</td>
</tr>
</tbody>
</table>

Fundamentally breaks our current programming paradigm and computing ecosystem

Adapted from John Shalf
Power Issues by 2018

FLOPs will cost less than on-chip data movement!

Adapted from John Shalf

Courtesy Horst Simon, LBNL
End of Moore’s Law Will Lead to New Architectures

Non-von Neumann

ARCHITECTURE

von Neumann

NEUROMORPHIC

QUANTUM COMPUTING

TODAY

BEYOND CMOS

CMOS

TECHNOLOGY

Beyond CMOS
It would only be the 6th paradigm.
As a last resort, we could will learn to program again.

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!
In Conclusion…

OpenMP

OpenACC

MPI
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)