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 brough your own immediate research concerns. Great!
in very little time. Performing a billion operations, on the other hand, could take minutes or hours, though it’s still possible provided you are patient. Performing a trillion operations, however, will basically take forever. So a fair rule of thumb is that the calculations we can perform on a computer are ones that can be done with about a billion operations or less.
Welcome to The Year of Exascale!

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

64-bit precision floating point operations per second
Where are those 10 or 12 orders of magnitude?

How do we get there from here?

BTW, that's a bigger gap than
Moore's Law abandoned serial programming around 2004.
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
But, at end of day we keep using getting more transistors.
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.

Courtesy Horst Simon, LBNL
Not a new problem, just a new scale...

Cray-2 with cooling tower in foreground, circa 1985
And how to get more performance from more transistors with the same power.

A 15% Reduction In Voltage Yields

**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>

**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

A 15% Reduction In Voltage Yields

**A 15% Reduction In Voltage Yields**
## Single Socket Parallelism

<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

The graph shows the trend of various computer performance metrics from 1970 to 2020, including:
- Transistors (thousands)
- Single-Thread Performance (SpecINT x 10^3)
- Frequency (MHz)
- Typical Power (Watts)
- Number of Logical Cores

The data up to 2010 was collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. The new plot and data for 2010-2017 were collected by K. Rupp.
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];
```
Rapid evolution continues with:

- **Turing**
- **Ampere**
- **Hopper**

Volta GV100 GPU with 85 Streaming Multiprocessor (SM) units

Volta GV100 SM
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);
}
```

```c
__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.
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
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

OpenMP 4/5 can help!
The pieces fit like this…

OpenMP

OpenACC

MPI
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.
<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>RIKEN Center for Computational Science Japan</td>
<td>Fujitsu</td>
<td>Fugaku</td>
<td>ARM 8.2A+ 48C 2.2GHz Torus Fusion Interconnect</td>
<td>7,299,072</td>
<td>442,010</td>
<td>537,212</td>
<td>29.8</td>
</tr>
<tr>
<td>2</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>3</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>4</td>
<td>National Super Computer Center in Wuxi China</td>
<td>NRCP</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>5</td>
<td>DOE/LBNL/NERSC United States</td>
<td>HPE</td>
<td>Perlmutter</td>
<td>EPYC 64C 2.45 GHz Slingshot NVIDIA A100</td>
<td>706,304</td>
<td>64,590</td>
<td>89,794</td>
<td>2.5</td>
</tr>
<tr>
<td>6</td>
<td>NVIDIA Corp. United States</td>
<td>NVIDIA</td>
<td>Selene</td>
<td>EPYC 64C 2.25 GHz, Infiniband HDR NVIDIA A100</td>
<td>555,520</td>
<td>63,460</td>
<td>79,215</td>
<td>2.6</td>
</tr>
<tr>
<td>7</td>
<td>National Super Computer Center in Guangzhou China</td>
<td>NUDT</td>
<td>Tianhe-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>8</td>
<td>Forschungszentrum Juelich Germany</td>
<td>Bull</td>
<td>Juwels</td>
<td>EPYC 24C 2.8GHz, Infiniband HDR NVIDIA A100</td>
<td>449,280</td>
<td>41,120</td>
<td>70,980</td>
<td>1.8</td>
</tr>
<tr>
<td>9</td>
<td>Eni S.p.A Italy</td>
<td>Dell</td>
<td>HPC5</td>
<td>Xeon 24C 2.1 GHz Infiniband HDR NVIDIA V100</td>
<td>669,760</td>
<td>35,450</td>
<td>51,720</td>
<td>2.2</td>
</tr>
<tr>
<td>10</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>
</tbody>
</table>
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
# System Designs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>System performance (TF)</td>
<td>&gt; 15.6 PF</td>
<td>&gt; 30 PF</td>
<td>200 PF</td>
<td>&gt; 120PF</td>
<td>35 – 45PF</td>
<td>&gt; 1.5 EF</td>
<td>≥ 1 EF DP sustained</td>
</tr>
<tr>
<td>Total system memory (GB/s)</td>
<td>&lt; 2.1</td>
<td>&lt; 3.7</td>
<td>10</td>
<td>6</td>
<td>&lt; 2</td>
<td>29</td>
<td>&gt; 60</td>
</tr>
<tr>
<td>Total system memory (TB)</td>
<td>847 TB DDR4 + 70 TB HBM + 7.5 TB GPU memory</td>
<td>~1 PB DDR4 + High Bandwidth Memory (HBM) + 1.5PB persistent memory</td>
<td>2.4 PB DDR4 + 0.4 PB HBM + 7.4 PB persistent memory</td>
<td>1.92 PB DDR4 + 240TB HBM</td>
<td>&lt; 2</td>
<td>4.6 PB DDR4 + 4.6 PB HMB2e + 36 PB persistent memory</td>
<td>&gt; 10 PB</td>
</tr>
<tr>
<td>Node performance (TF)</td>
<td>2.7 TF (KNL node) and 166.4 TF (GPU node)</td>
<td>&gt; 3</td>
<td>43</td>
<td>&gt; 70 (GPU)</td>
<td>&gt; 4 (CPU)</td>
<td>&gt; 70 TF</td>
<td>TBD</td>
</tr>
<tr>
<td>Node processors</td>
<td>Intel Xeon Phi 7220 64-core CPUs (KNL) and many core CPUs coupled with 2 AMD Epyc 64-core CPUs</td>
<td>Intel Knights Landing many core CPUs</td>
<td>Intel Haswell CPU in data partition</td>
<td>2 IBM Power9 CPUs + 5 Nvidia Volta GPUs</td>
<td>CPU only nodes: AMD Epyc Milan CPUs; GPU-CPU nodes: AMD Epyc Milan with NVIDIA A100 GPUs</td>
<td>1 CPU; 4 GPUs</td>
<td>1 HPC and AI optimized AMD Epyc CPU and 4 AMD Radeon Instinct GPUs</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>4,392 KNL nodes and 24 DGX-A100 nodes</td>
<td>9,300 nodes in data partition</td>
<td>4608 nodes</td>
<td>&gt; 1,500 (GPU)</td>
<td>&gt; 3,000 (CPU)</td>
<td>&gt; 500</td>
<td>&gt; 9,000 nodes</td>
</tr>
<tr>
<td>CPU-GPU Interconnect</td>
<td>NVLINK on GPU nodes</td>
<td>N/A</td>
<td>NVLINK Coherent memory across node</td>
<td>PCIe</td>
<td>AMD Infinity Fabric Coherent memory across the node</td>
<td>Unified memory architecture, RAMBO</td>
<td>Unified memory architecture, RAMBO</td>
</tr>
<tr>
<td>Node-to-node Interconnect</td>
<td>Aries (KNL nodes) and HDR200 (GPU nodes)</td>
<td>Aries</td>
<td>Dual Rail EDR-IB</td>
<td>HPE Slingshot NIC</td>
<td>HPE Slingshot NIC</td>
<td>HPE Slingshot</td>
<td>HPE Slingshot</td>
</tr>
<tr>
<td>File System</td>
<td>200 PB, 1.3 TB/s Lustre</td>
<td>10 PB, 210 GB/s Lustre</td>
<td>28 PB, 744 GB/s Lustre</td>
<td>250 PB, 2.5 TB/s GPFS</td>
<td>35 PB All Flash, Lustre</td>
<td>N/A</td>
<td>695 PB + 10 PB Flash performance tier, Lustre</td>
</tr>
</tbody>
</table>

[ASCR Computing Upgrades At-a-Glance](https://www.ornl.gov/ascr)  
November 24, 2020
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/
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

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.
Sustaining Performance Improvements
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

NEUROMORPHIC

ARCHITECTURE

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.

- Straight forward extrapolation results in a real-time human brain scale simulation at about 1 - 10 Exaflop/s with 4 PB of memory
- Exascale computers in 2021 will have a power consumption of at 20 - 30 MW
- The human brain takes 20W
- Even under best assumptions in 2021 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