

# **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.

# 1<sup>st</sup> Theme

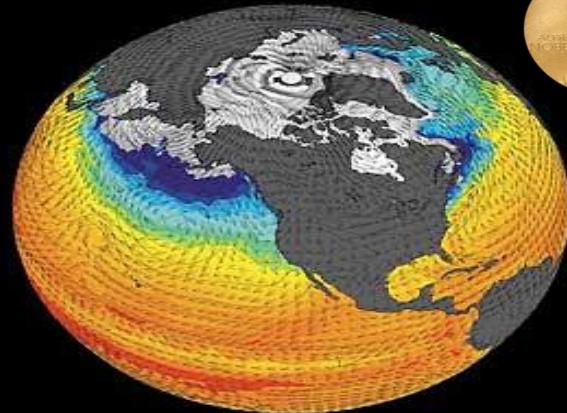
We need Exascale computing

We aren't getting to Exascale without parallel

What does parallel computing look like

Where is this going

# FLOPS we need: Climate change analysis



---

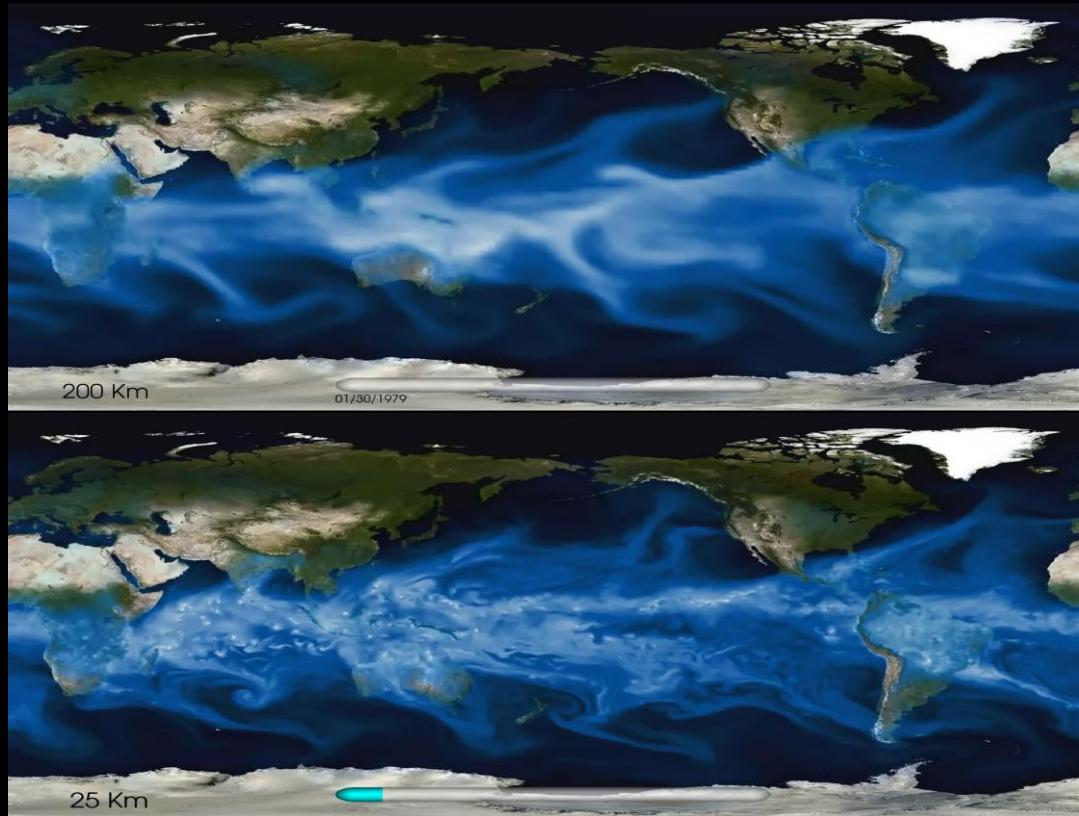
## Simulations

- 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× 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

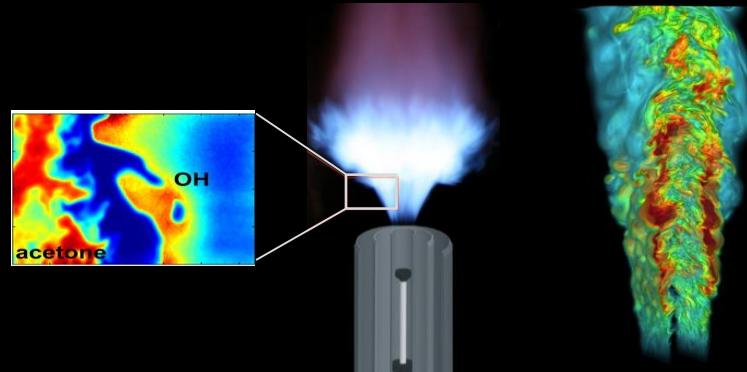
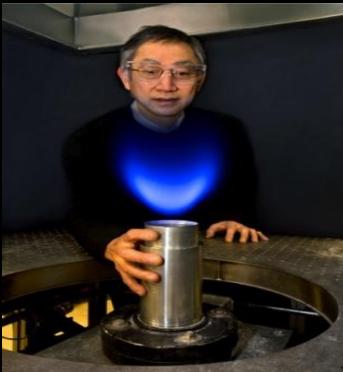
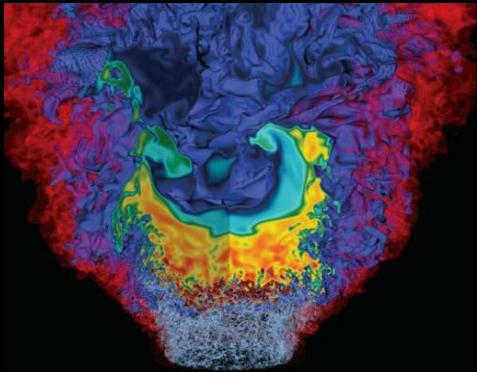
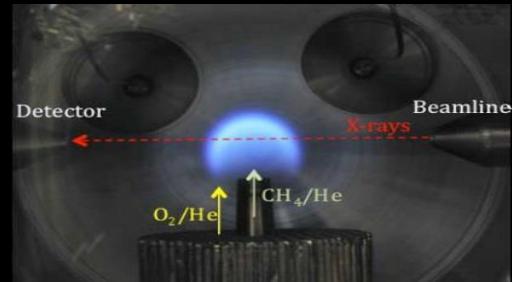
# Qualitative Improvement of Simulation with Higher Resolution (2011)



Michael Wehner, Prabhat, Chris Algieri, Fuyu Li, Bill Collins, Lawrence Berkeley National Laboratory; Kevin Reed, University of Michigan; Andrew Gettelman, Julio Bacmeister, Richard Neale, National Center for Atmospheric Research

# 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



# Modha Group at IBM Almaden



Mouse



Rat



Cat



Monkey



Human

N:  $16 \times 10^6$

N:  $56 \times 10^6$

N:  $763 \times 10^6$

N:  $2 \times 10^9$

N:  $22 \times 10^9$

S:  $128 \times 10^9$

S:  $448 \times 10^9$

S:  $6.1 \times 10^{12}$

S:  $20 \times 10^{12}$

S:  $220 \times 10^{12}$



Almaden

BG/L

December, 2006



Watson

BG/L

April, 2007



WatsonShaheen

BG/P

March, 2009



LLNL Dawn

BG/P

May, 2009

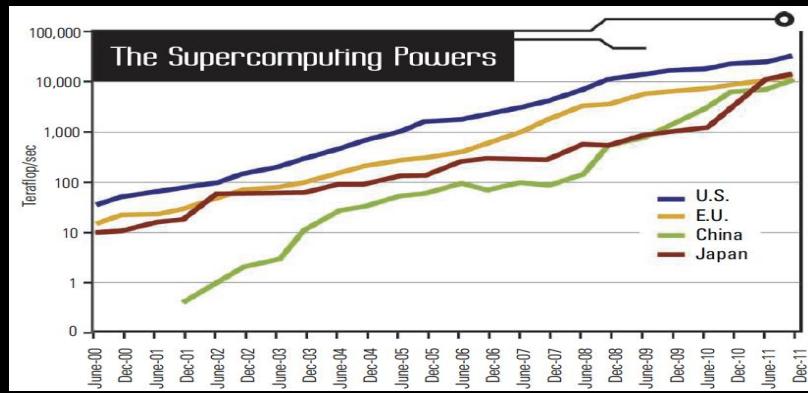
Recent simulations achieve  
unprecedented scale of  
 $65 \times 10^9$  neurons and  $16 \times 10^{12}$  synapses

LLNL Sequoia

BG/Q

June, 2012

# Also important: We aren't going to be left behind.



# **2<sup>nd</sup> Theme**

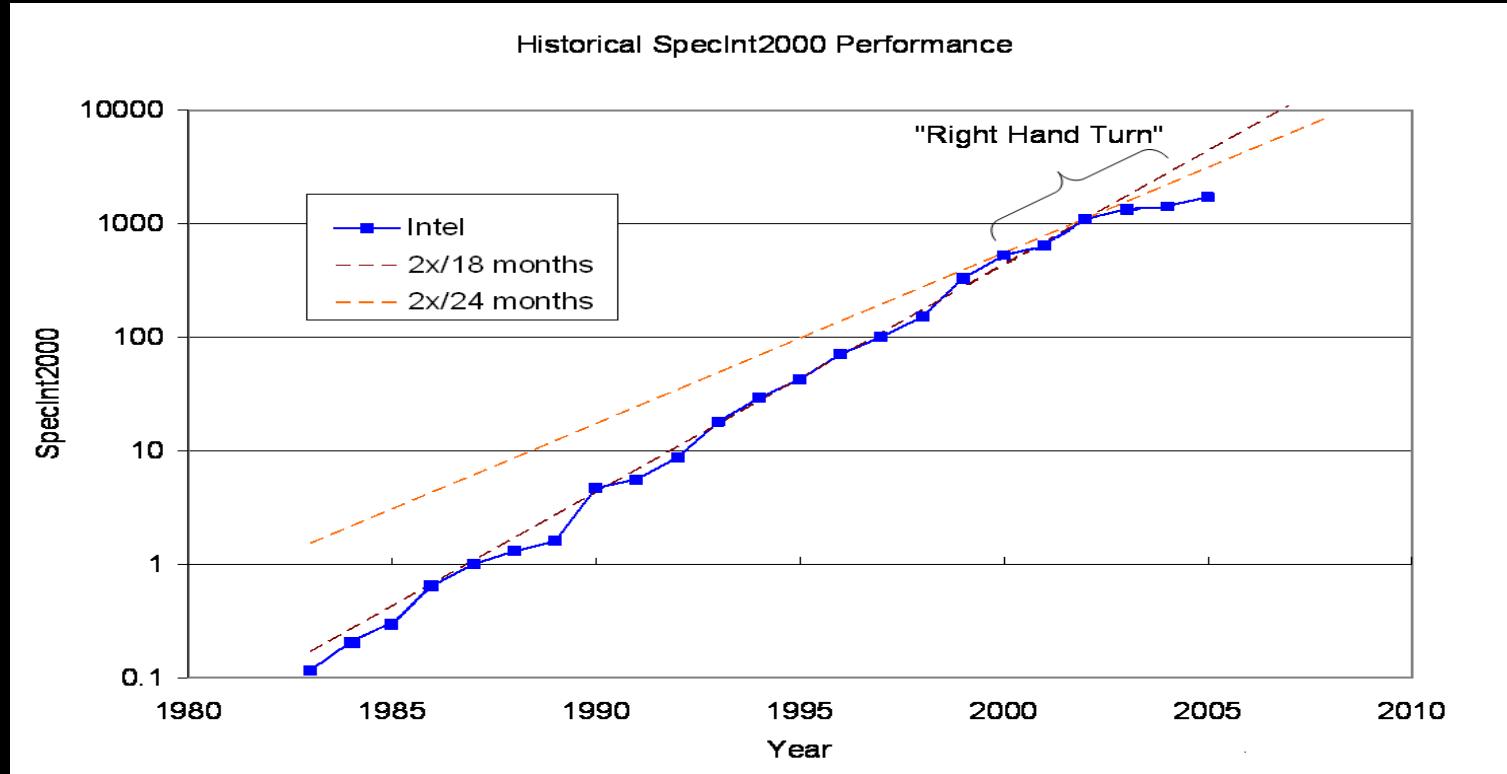
**We will have Exascale computing**

**You aren't getting to Exascale without going very parallel**

**What does parallel computing look like**

**Where is this going**

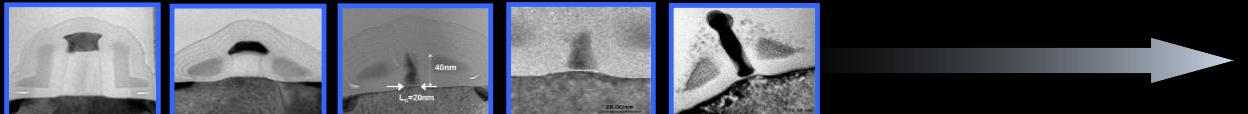
# Waiting for Moore's Law to save your serial code started getting bleak in 2004



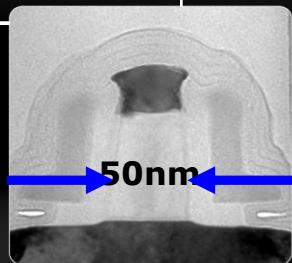
Source: published SPECInt data

# Moore's Law is not at all dead...

## Intel process technology capabilities

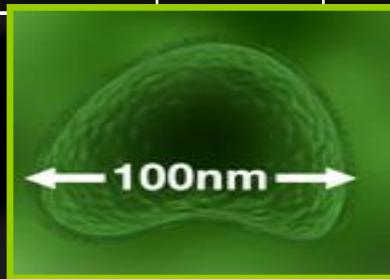


High Volume Manufacturing	2004	2006	2008	2010	2012	2014	2016	2018
Feature Size	90nm	65nm	45nm	32nm	22nm	16nm	11nm	8nm
Integration Capacity (Billions of Transistors)	2	4	8	16	32	64	128	256



Transistor for  
90nm Process

Source: Intel



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:

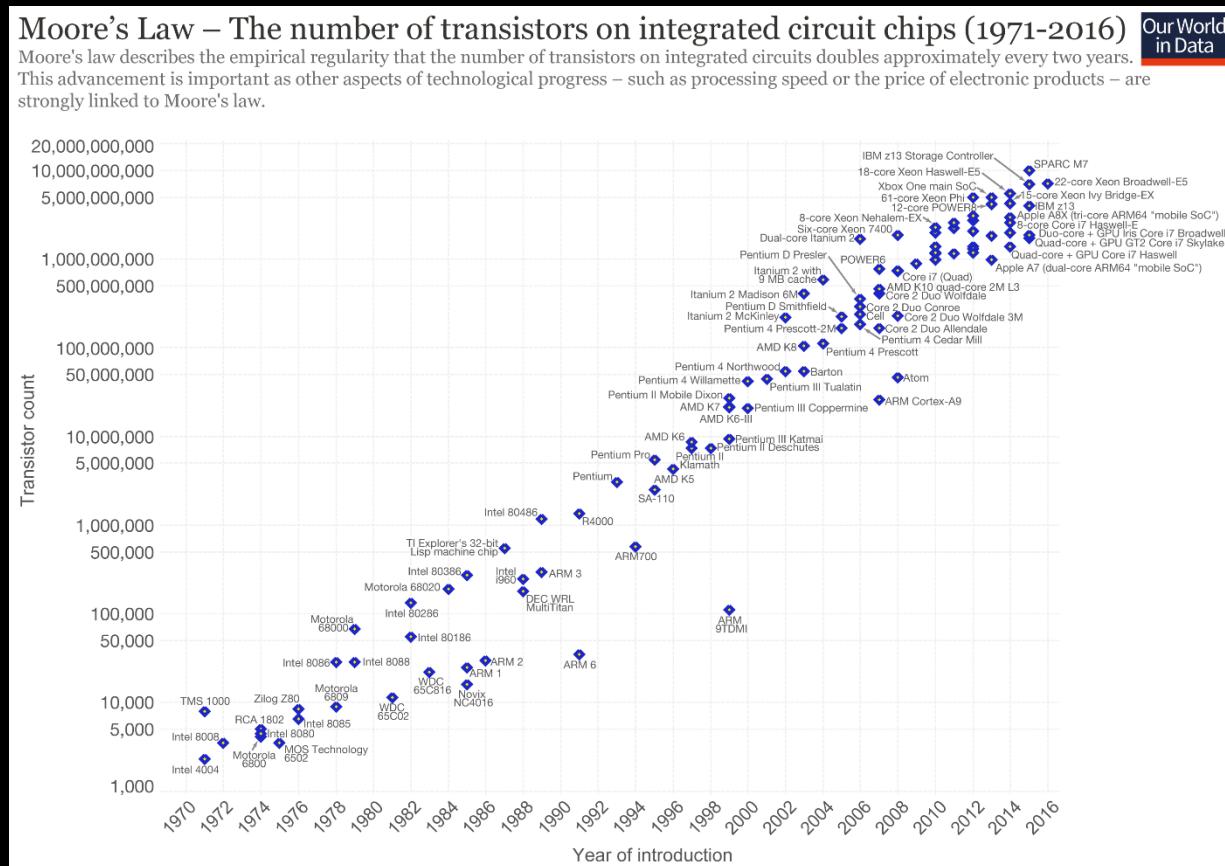
- error-correction circuitry
- redundancy
- read- and write-boosting circuitry for failing static RAM cells
- circuits to track and adapt to performance variations
- complicated memory hierarchies to handle multicore architectures.

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.**

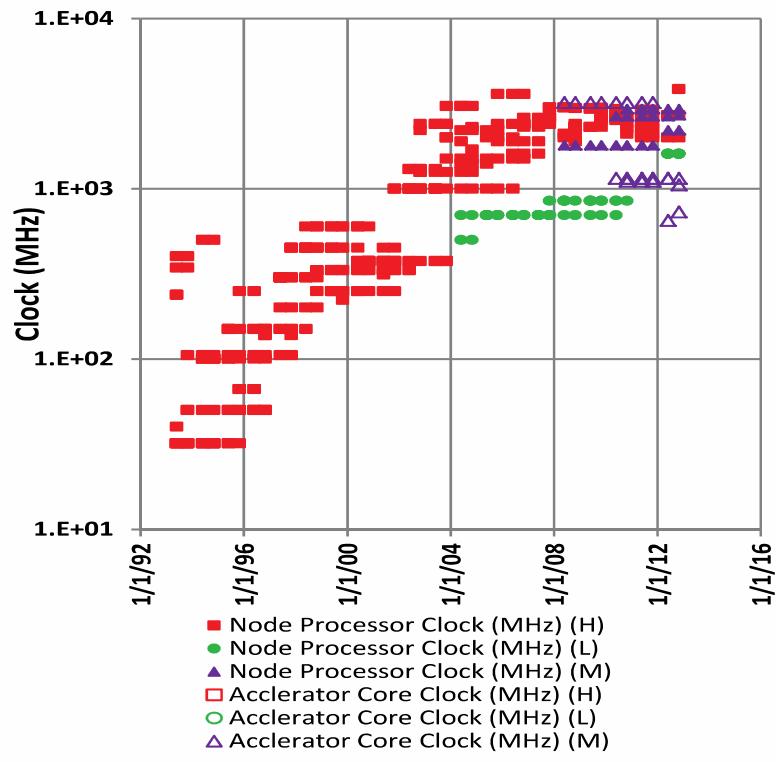
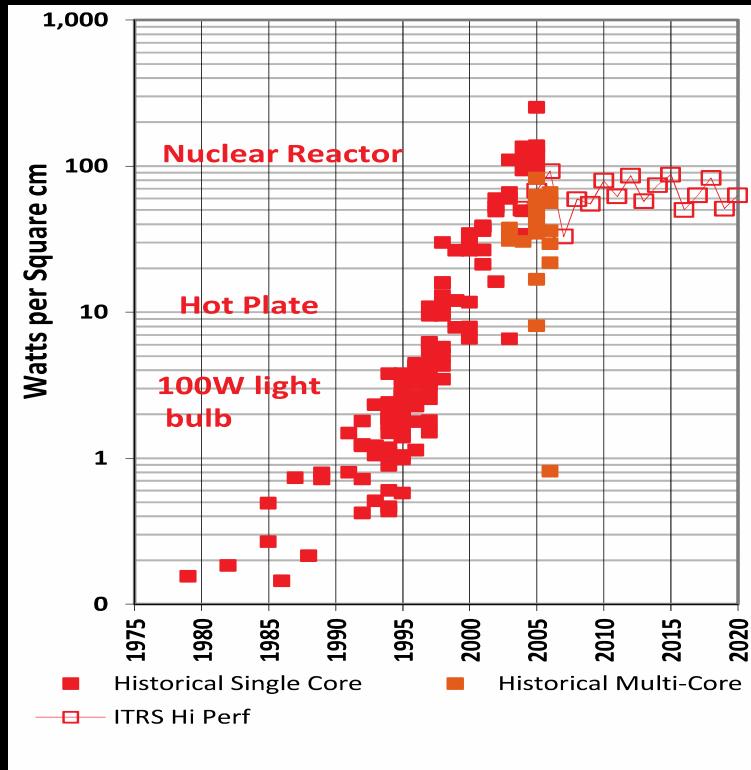


Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))

The data visualization is available at [OurWorldInData.org](http://OurWorldInData.org). There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Rose

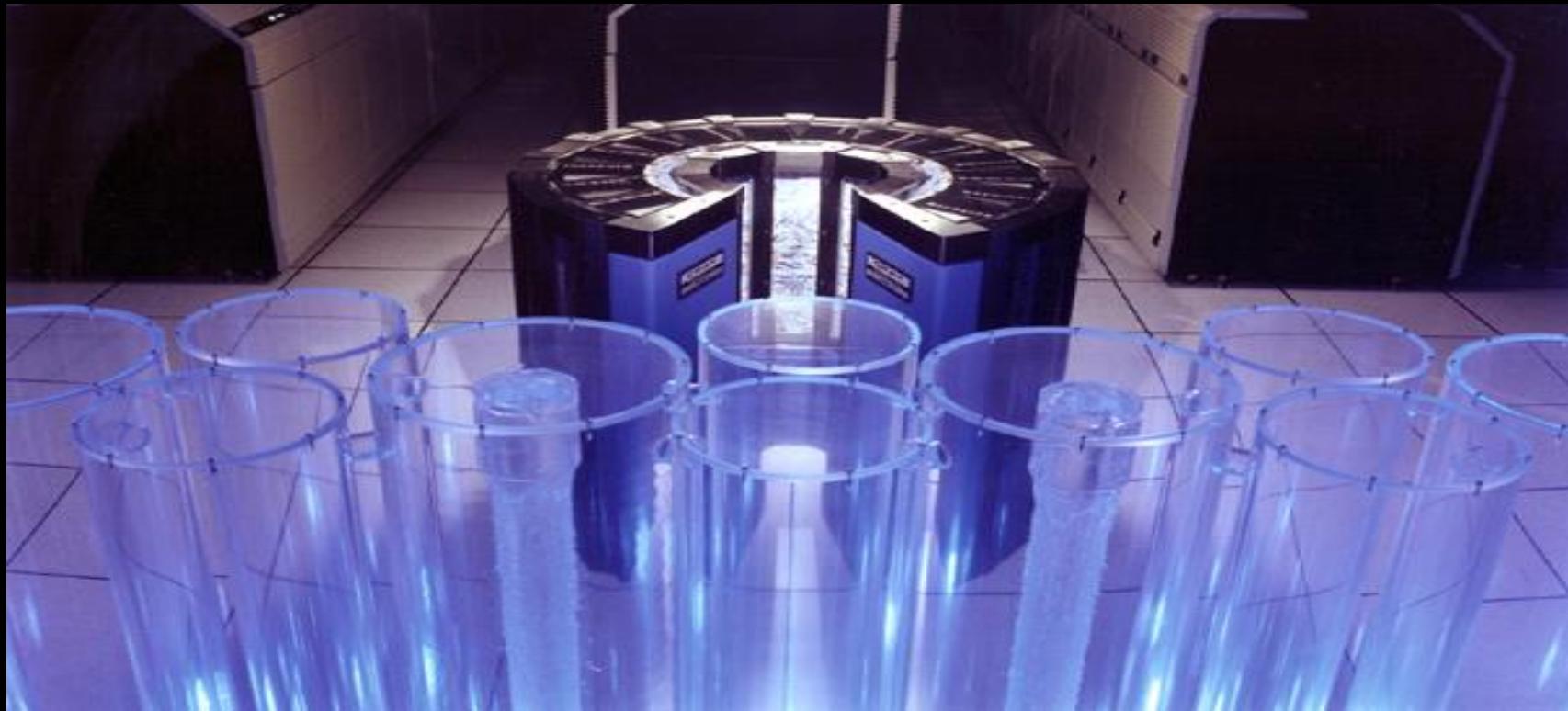
# That Power and Clock Inflection Point in 2004... didn't get better.



Fun fact: At 100+ Watts and <1V, currents are beginning to exceed 100A at the point of load!

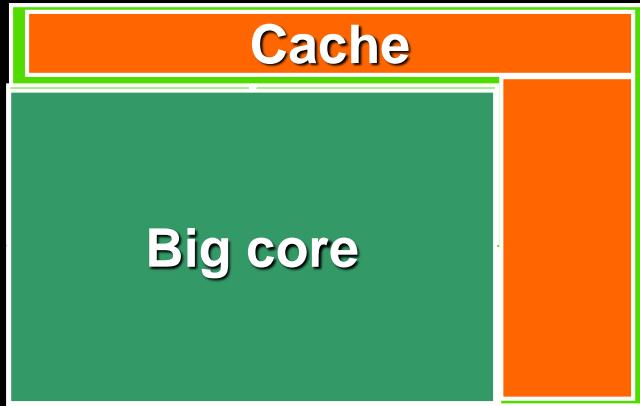
Courtesy Horst Simon, LBNL

# Not a new problem, just a new scale...



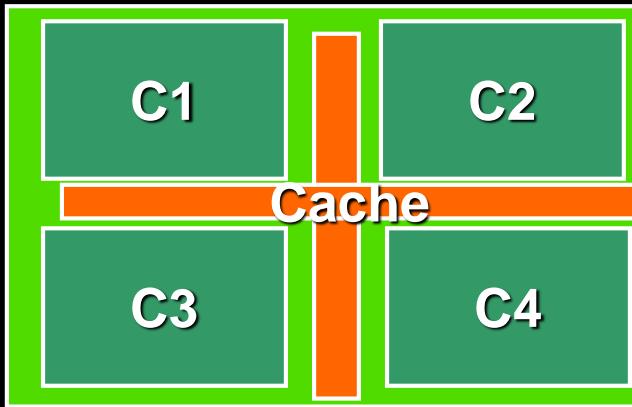
Cray-2 with cooling tower in foreground, circa 1985

# How to get same number of transistors to give us more performance without cranking up power?



Key is that

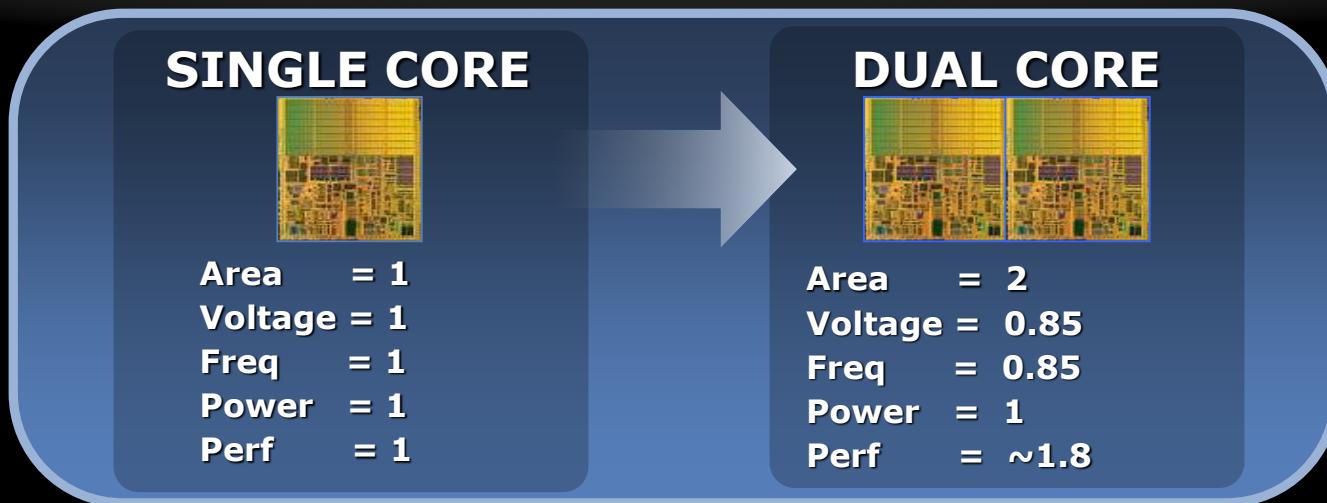
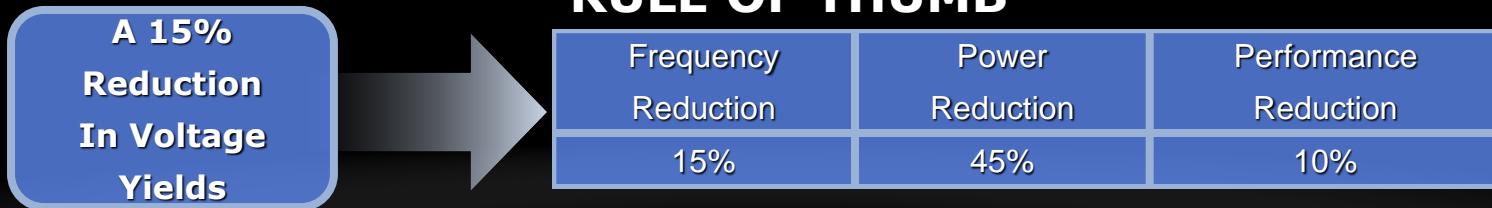
$\text{Performance} \approx \sqrt{\text{area}}$



Power =  $\frac{1}{4}$

Performance =  $\frac{1}{2}$

**And how to get more performance from more transistors with the same power.**



# Single Socket Parallelism

Processor	Year	Vector	Bits	SP FLOPs / core / cycle	Cores	FLOPs/cycle
Pentium III	1999	SSE	128	3	1	3
Pentium IV	2001	SSE2	128	4	1	4
Core	2006	SSE3	128	8	2	16
Nehalem	2008	SSE4	128	8	10	80
Sandybridge	2011	AVX	256	16	12	192
Haswell	2013	AVX2	256	32	18	576
KNC	2012	AVX512	512	32	64	2048
KNL	2016	AVX512	512	64	72	4608
Skylake	2017	AVX512	512	96	28	2688

# **3<sup>rd</sup> Theme**

We will have Exascale computing

You will get there by going very parallel

**What does parallel computing look like?**

Where is this going

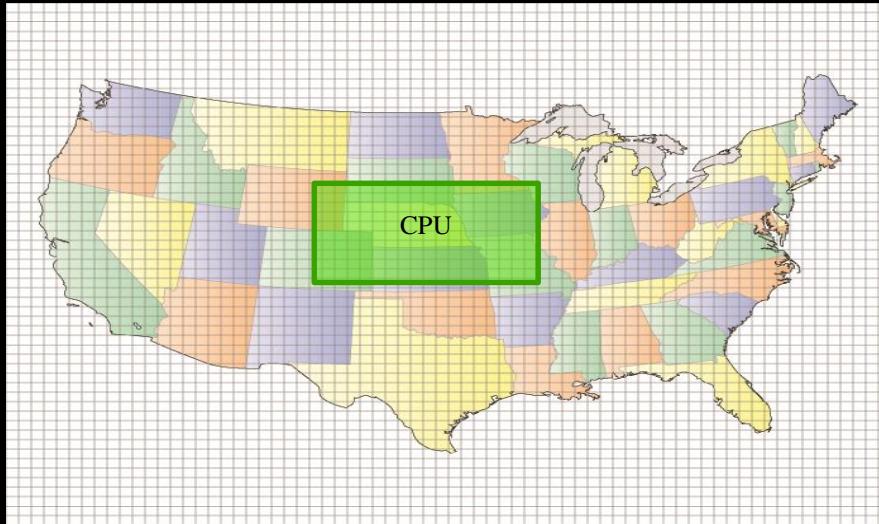
# Parallel Computing

**One woman can make a baby in 9 months.**

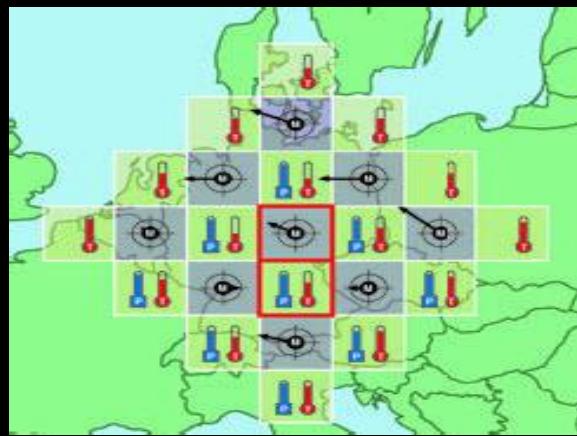
**Can 9 women make a baby in 1 month?**

**But 9 women can make 9 babies in 9 months.**

# Prototypical Application: Serial Weather Model

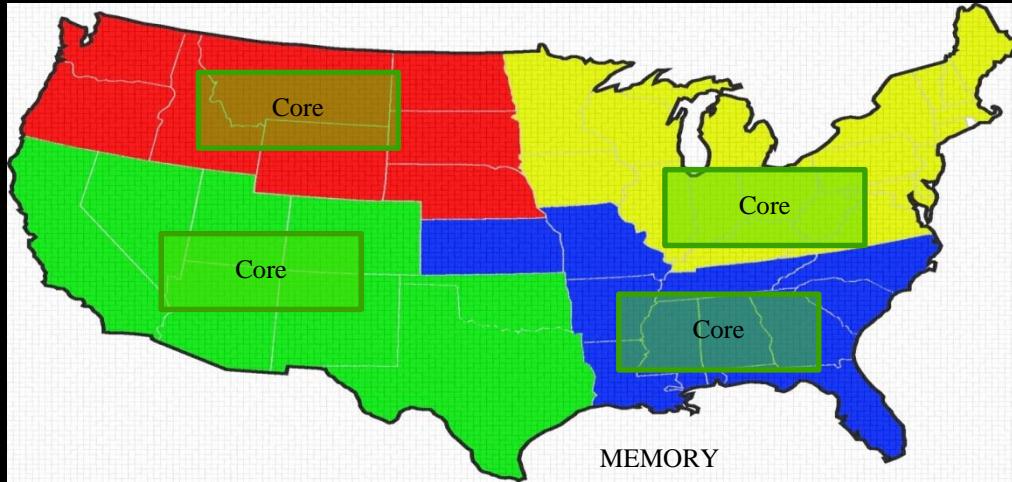


# First Parallel Weather Modeling Algorithm: Richardson in 1917



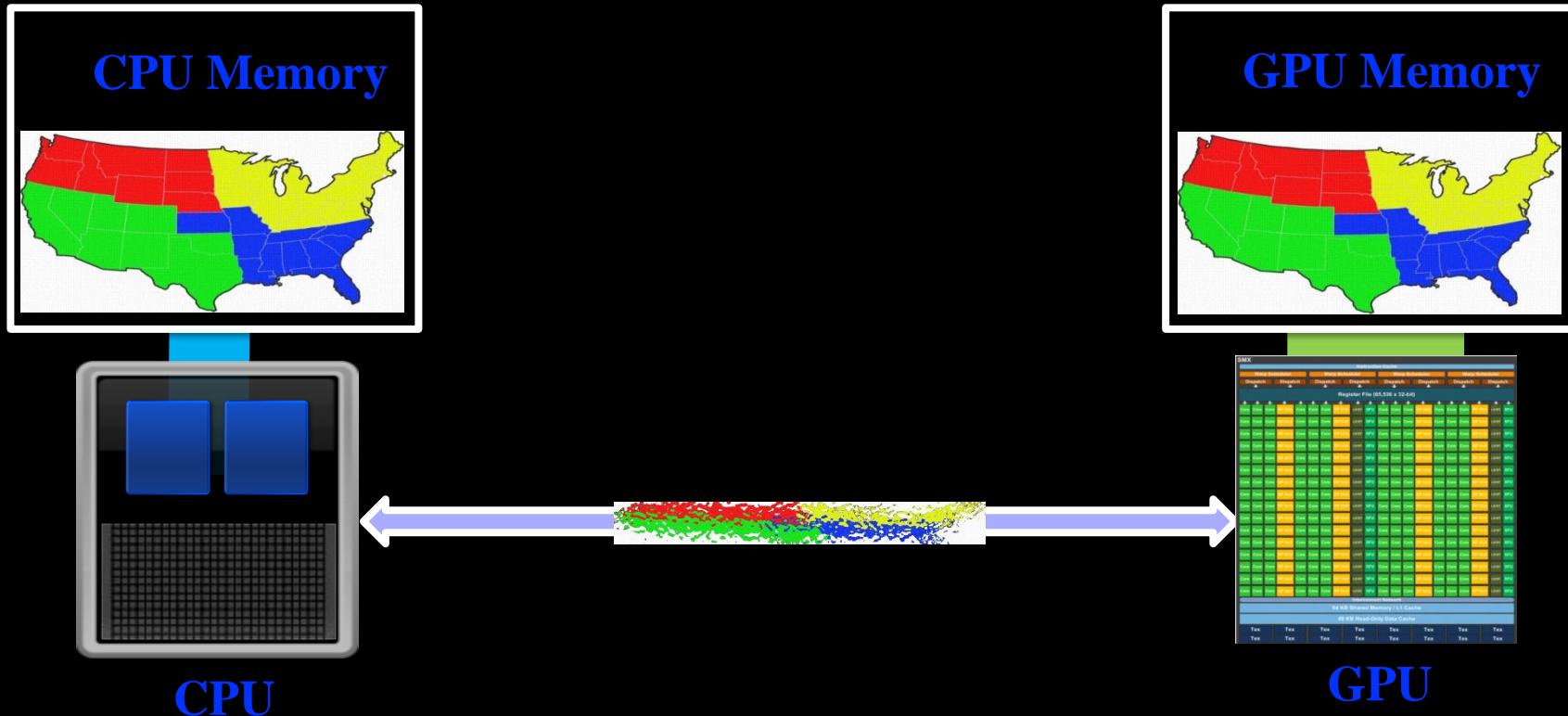
*Courtesy John Burkhardt, Virginia Tech*

# Weather Model: Shared Memory (OpenMP)



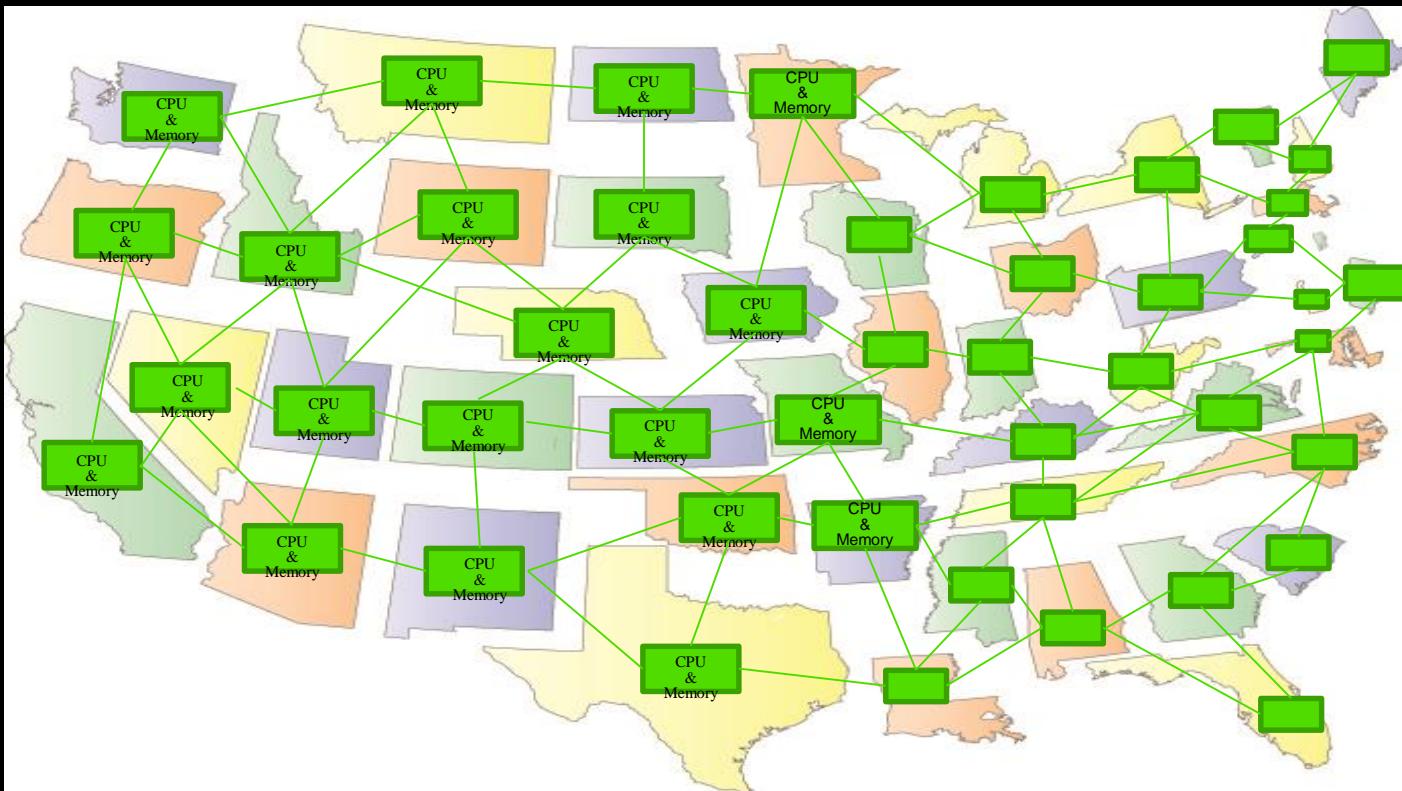
*Four meteorologists in the same room sharing the map.*

# Weather Model: Accelerator (OpenACC)



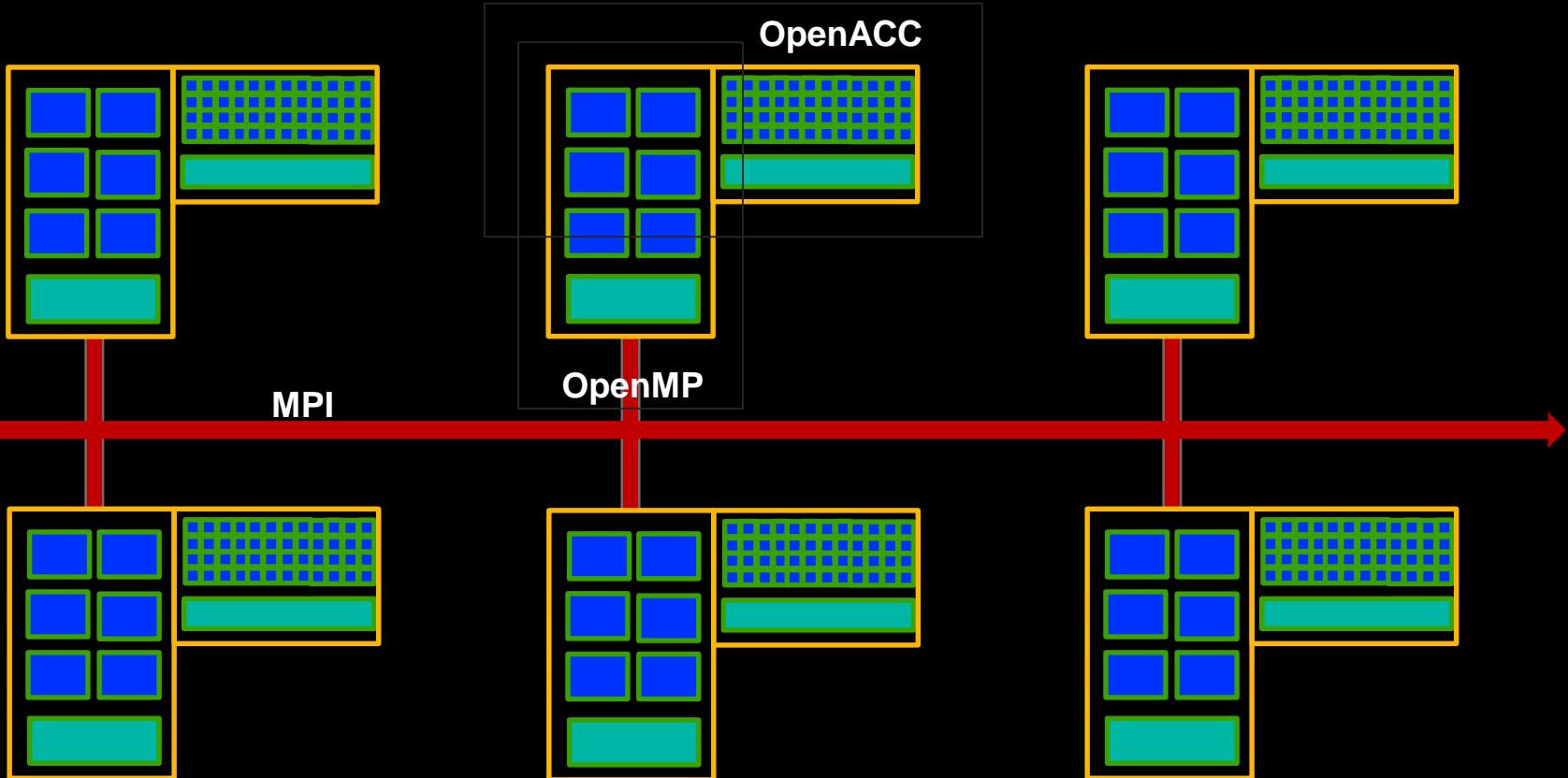
*1 meteorologist coordinating 1000 math savants using tin cans and a string.*

# Weather Model: Distributed Memory (MPI)

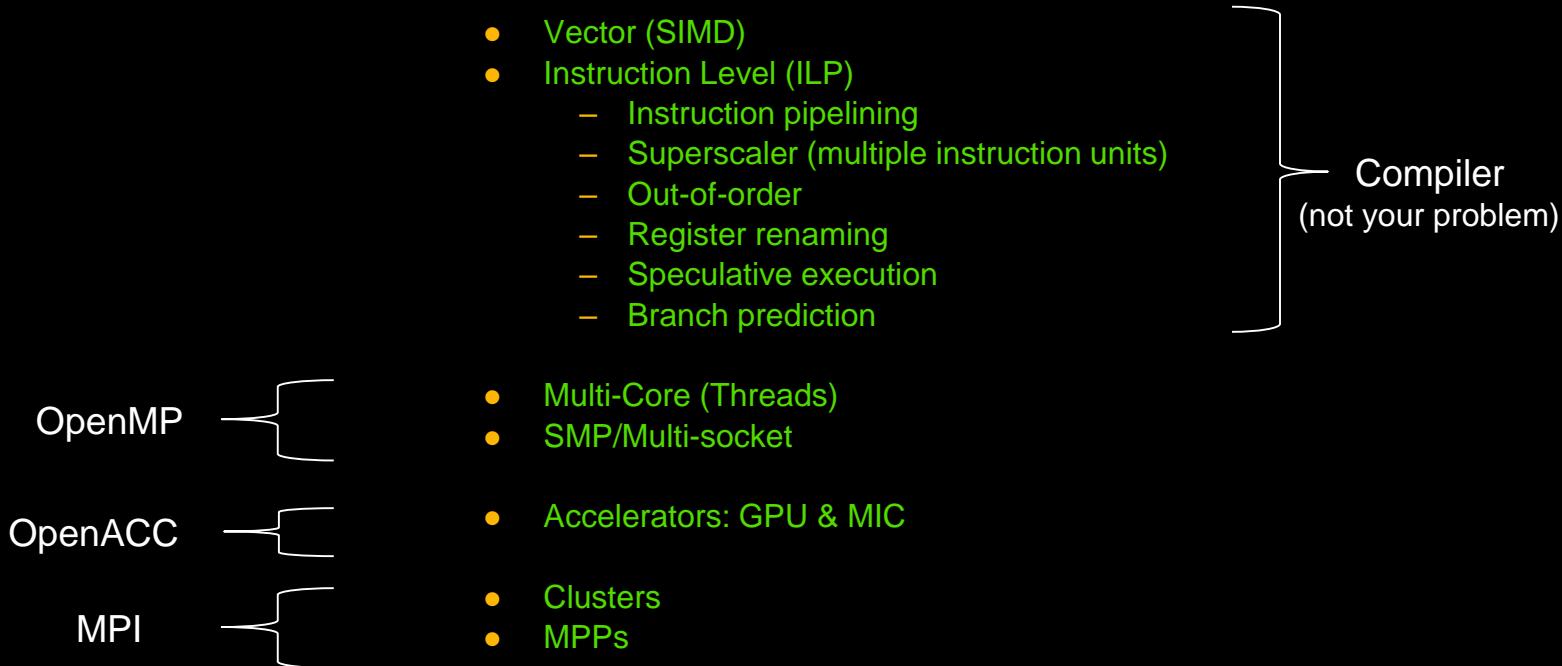


*50 meteorologists using telegraphs.*

# The pieces fit like this...



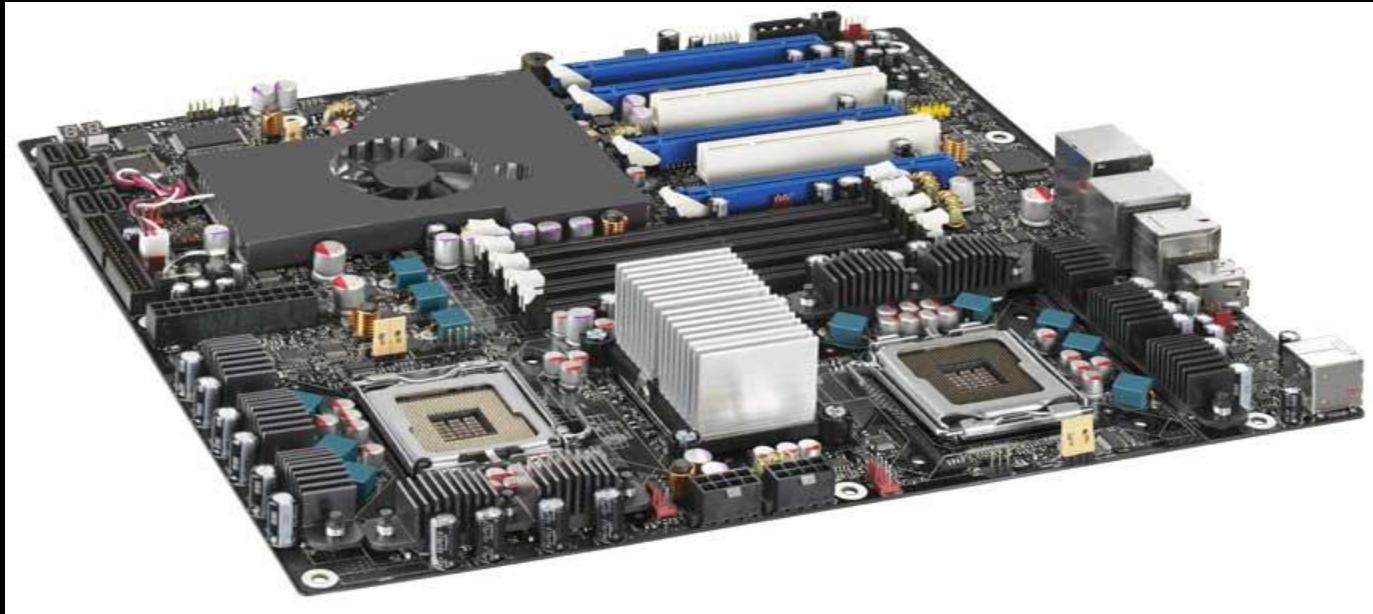
# Many Levels and Types of Parallelism



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. 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



Compute Nodes	18,688
Login & I/O Nodes	512
Memory per node	32 GB + 6 GB
# of Fermi chips (2012)	960
# of NVIDIA "Kepler" (2013)	14,592
Total System Memory	688 TB
Total System Peak Performance	20+ Petaflops
Liquid cooling at the cabinet level	Cray EcoPHLex

# Cores, Nodes, Processors, PEs?

- “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 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

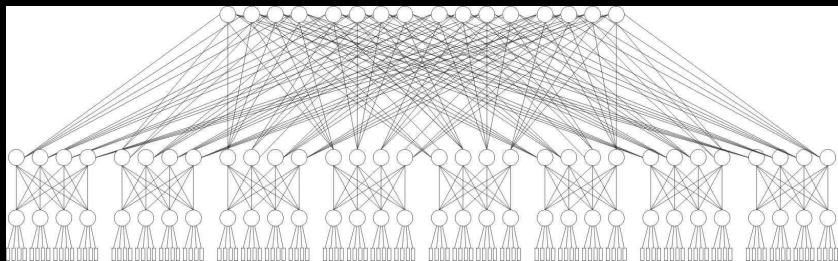
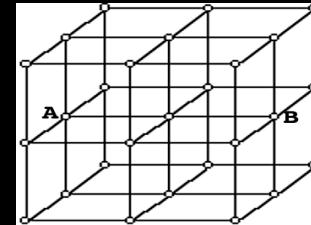
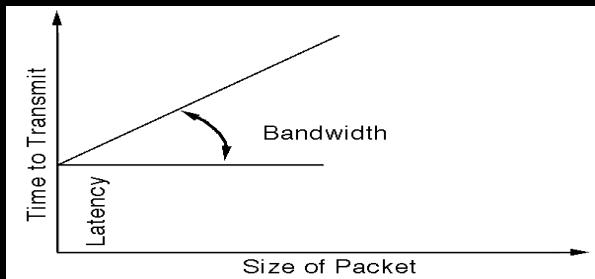
3 characteristics sum up the network:

- **Latency**

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

- **Bandwidth**

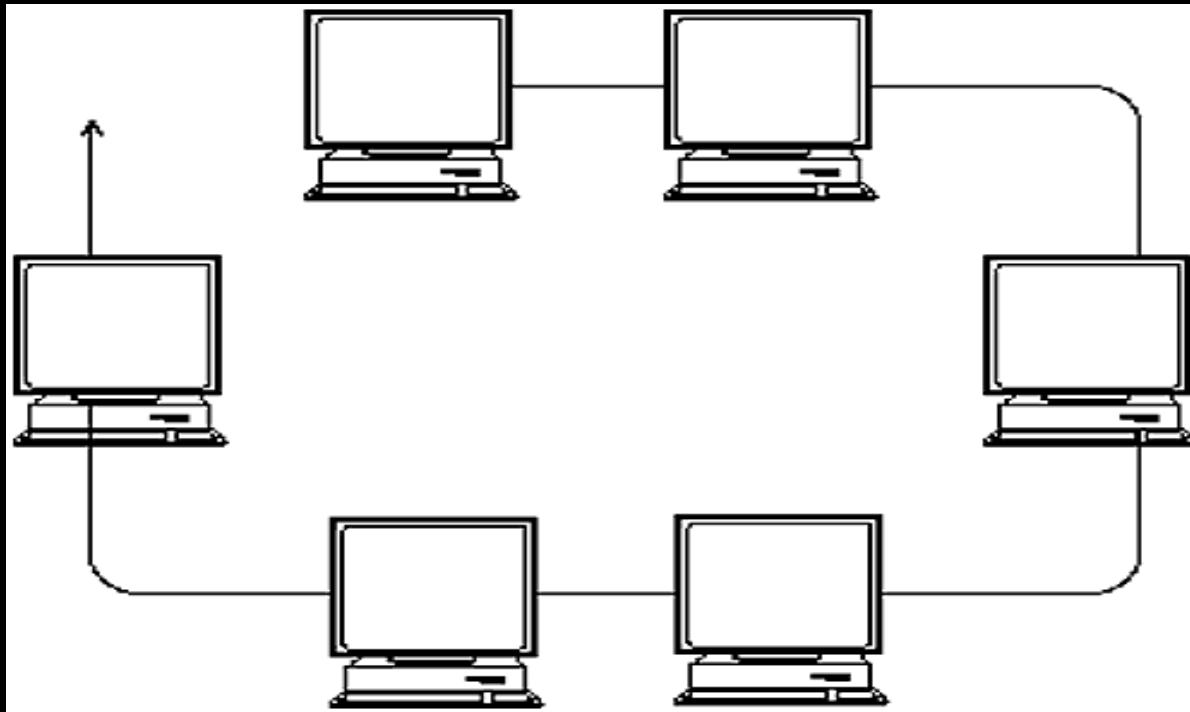
The rate at which a very large packet of information can be sent



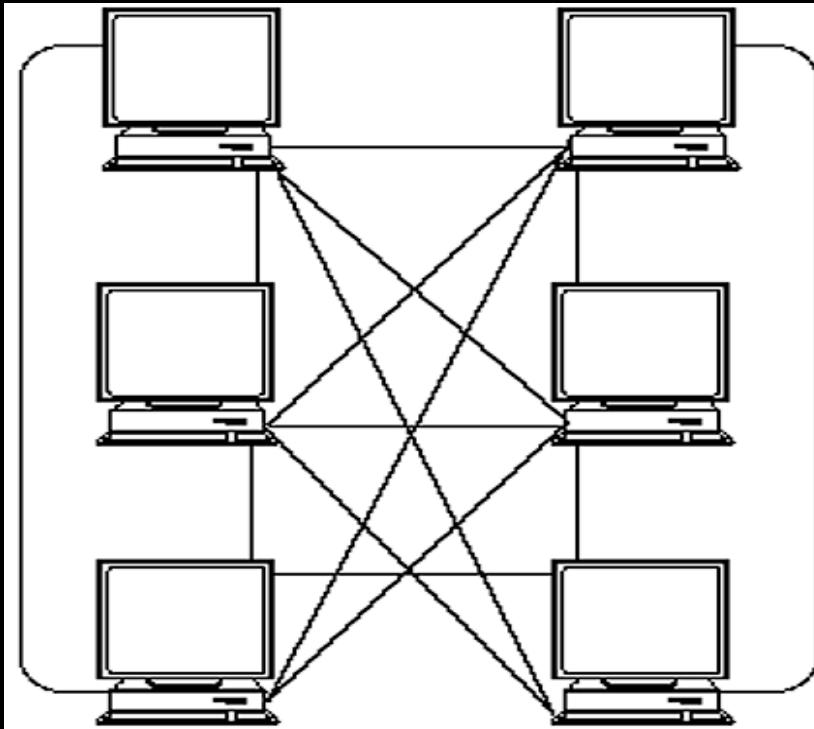
- **Topology**

The configuration of the network that determines how processing units are directly connected.

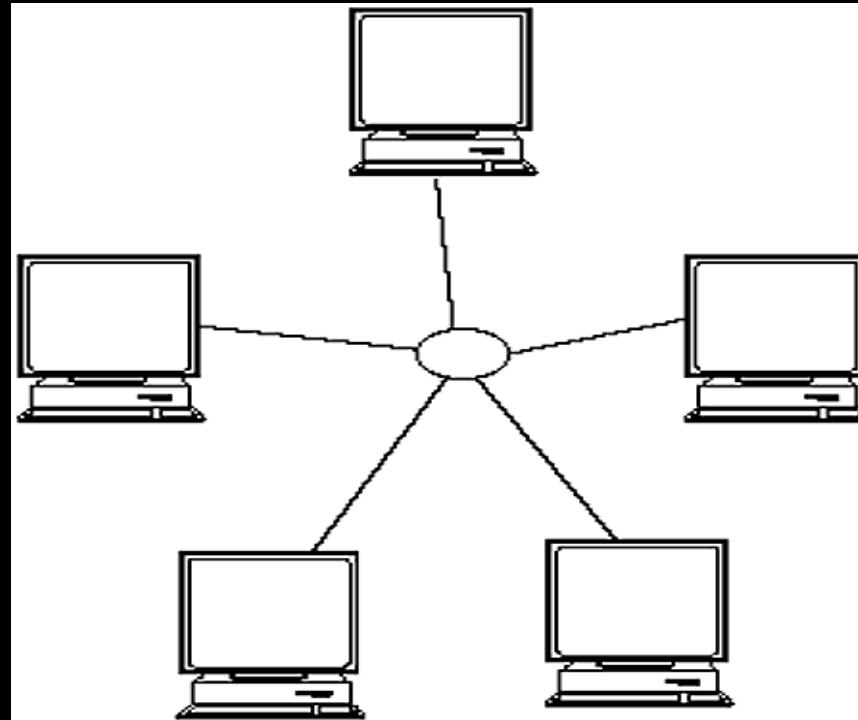
# Ethernet with Workstations



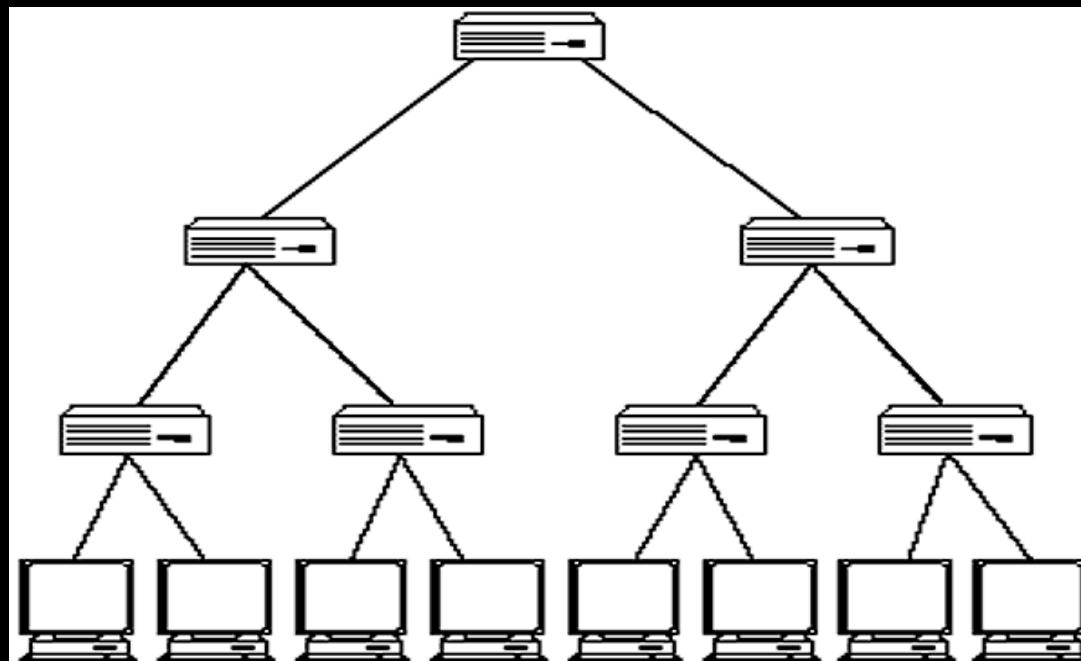
# Complete Connectivity



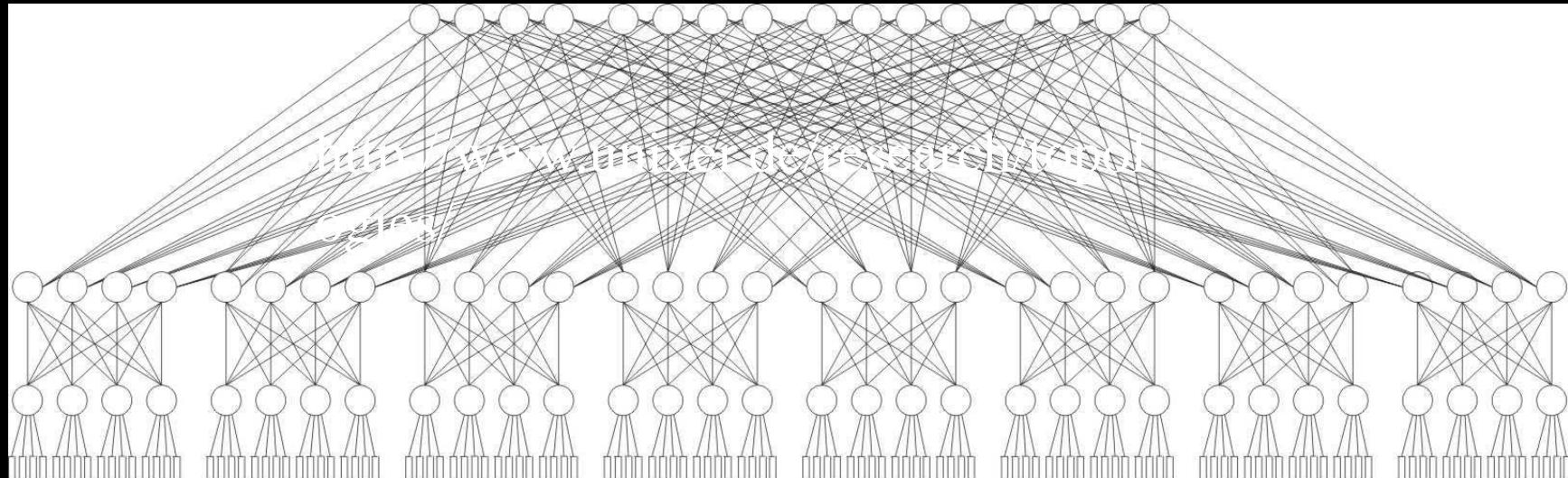
# Crossbar



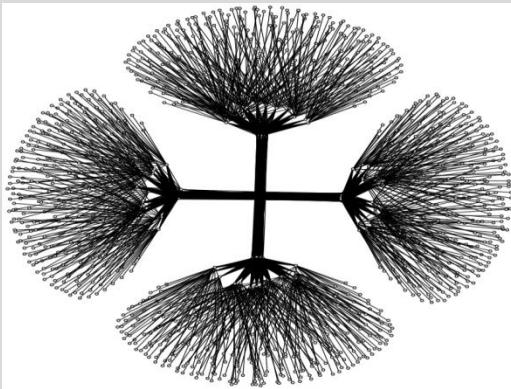
# Binary Tree



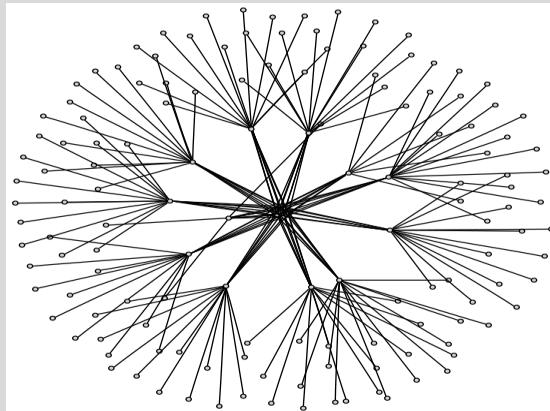
# Fat Tree



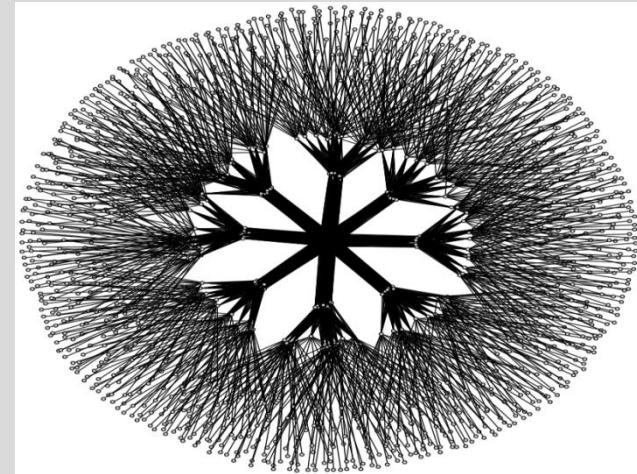
# Other Fat Trees



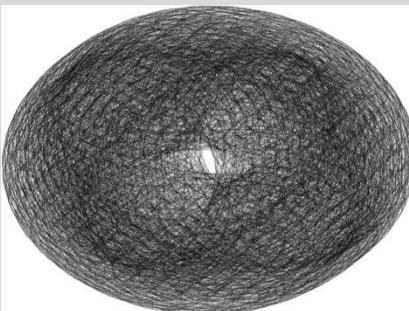
Big Red @ IU



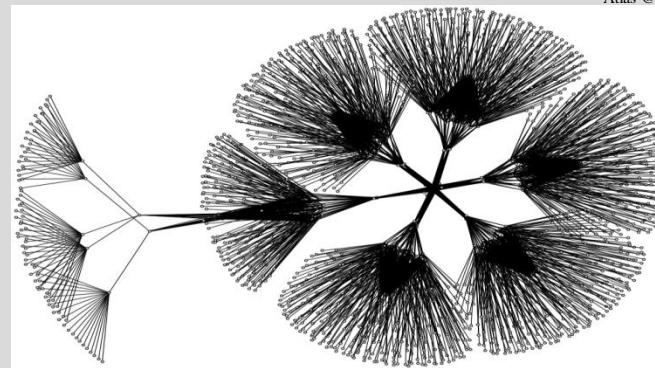
Odin @ IU



Atlas @ LLNL



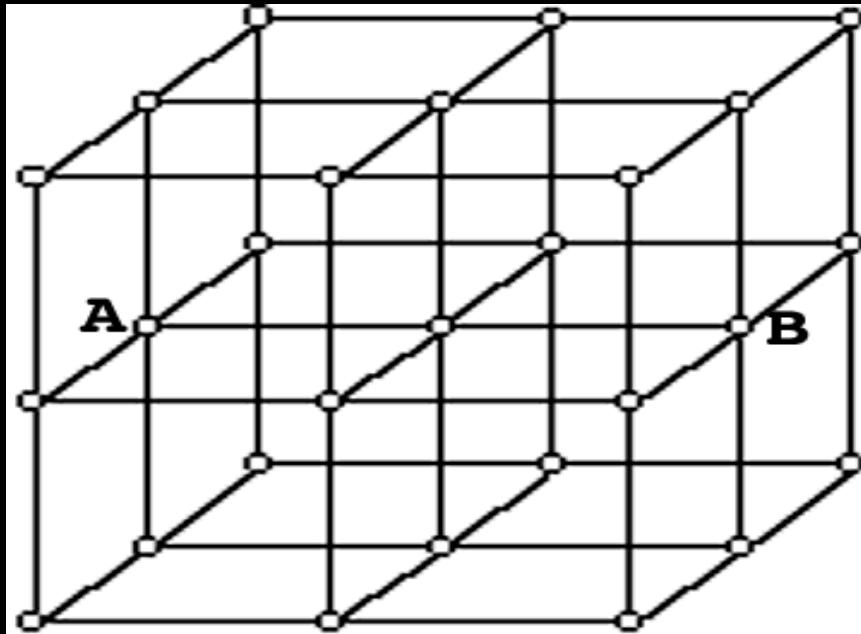
Jaguar @ ORNL



Tsubame @ Tokyo Inst. of Tech

From Torsten Hoefler's Network Topology Repository at  
<http://www.unixer.de/research/topologies/>

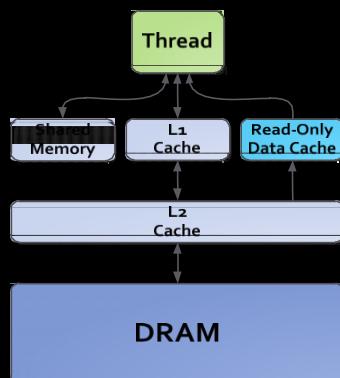
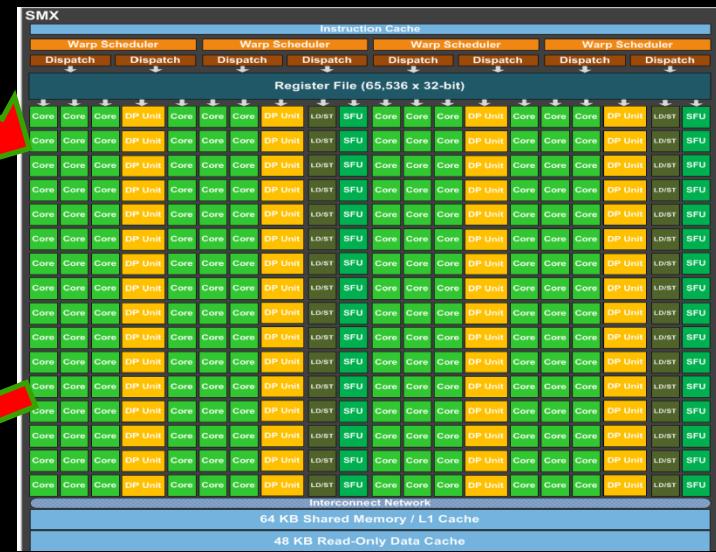
# 3-D Torus (T3D – XT7...)



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:

<http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf>

# 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.



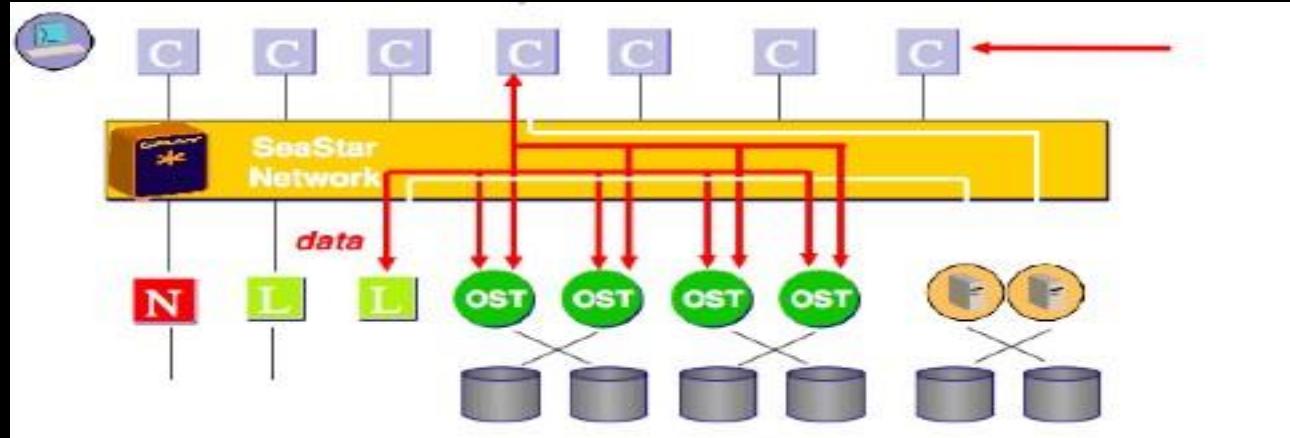
# Top 10 Systems as of November 2017

#	Site	Manufacturer	Computer	CPU Interconnect [Accelerator]	Cores	Rmax (Tflops)	Rpeak (Tflops)	Power (MW)
1	National Super Computer Center in Guangzhou China	NRCPC	Sunway TaihuLight	Sunway SW26010 260C 1.45GHz	10,649,600	33,014	125,455	15.3
2	National Super Computer Center in Guangzhou China	NUDT	Tianhe-2 (MilkyWay-2)	Intel Xeon E5-2692 2.2 GHz TH Express-2 Intel Xeon Phi 31S1P	3,120,000	33,862	54,902	17.8
3	Swiss National Supercomputing Centre (CSCS) Switzerland	Cray	Piz Daint Cray XC50	Xeon E5-2690 2.6 GHz Aries NVIDIA P100	361,760	19,590	25,326	2.2
4	Japan Agency for Marine-Earth Science Japan	ExaScaler	Gyoukou	Xeon D-1571 1.3GHz Infiniband EDR	19,860,000	19,135	28,192	1.3
5	DOE/SC/Oak Ridge National Laboratory United States	Cray	Titan Cray XK7	Opteron 6274 2.2 GHz Gemini NVIDIA K20x	560,640	17,590	27,112	8.2
6	DOE/NNSA/LLNL United States	IBM	Sequoia BlueGene/Q	Power BQC 1.6 GHz Custom	1,572,864	17,173	20,132	7.8
7	DOE/NNSA/LANL/SNL United States	Cray	Trinity Cray XC40	Xeon E5-2698v3 2.3 GHz Aries Intel Xeon Phi 7250	979,968	17,173	20,132	7.8
8	DOE/SC/LBNL/NERSC United States	Cray	Cori Cray XC40	Aries Intel Xeon Phi 7250	622,336	14,014	27,880	3.9
9	Joint Center for Advanced High Performance Computing Japan	Fujitsu	Oakforest Primergy	Intel OPA Intel Xeon Phi 7250	556,104	13,554	24,913	2.7
10	RIKEN Advanced Institute for Computational Science (AICS)	Fujitsu	K Computer	SPARC64 VIIIfx 2.0 GHz Tofu	705,024	10,510	11,280	12.6

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.



# **4<sup>th</sup> 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 120 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 = quintillion**

23,800 X



833,000 X

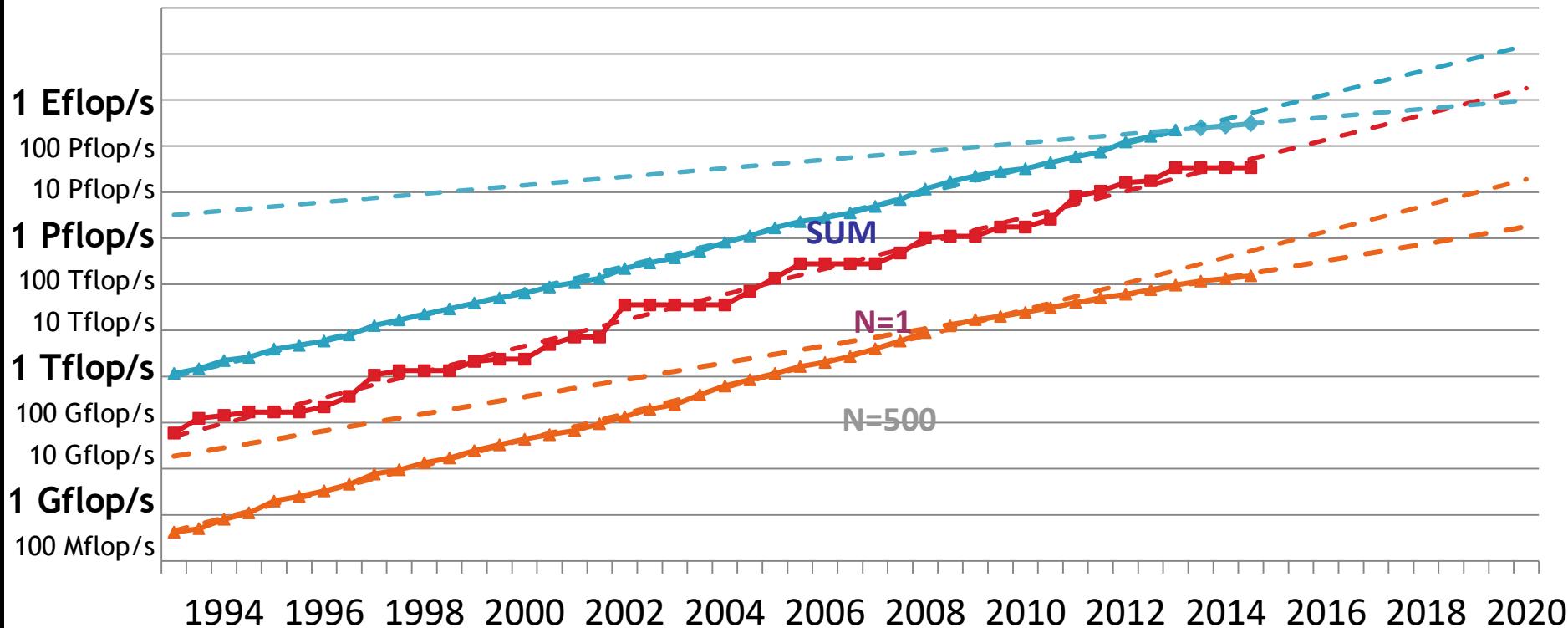


or

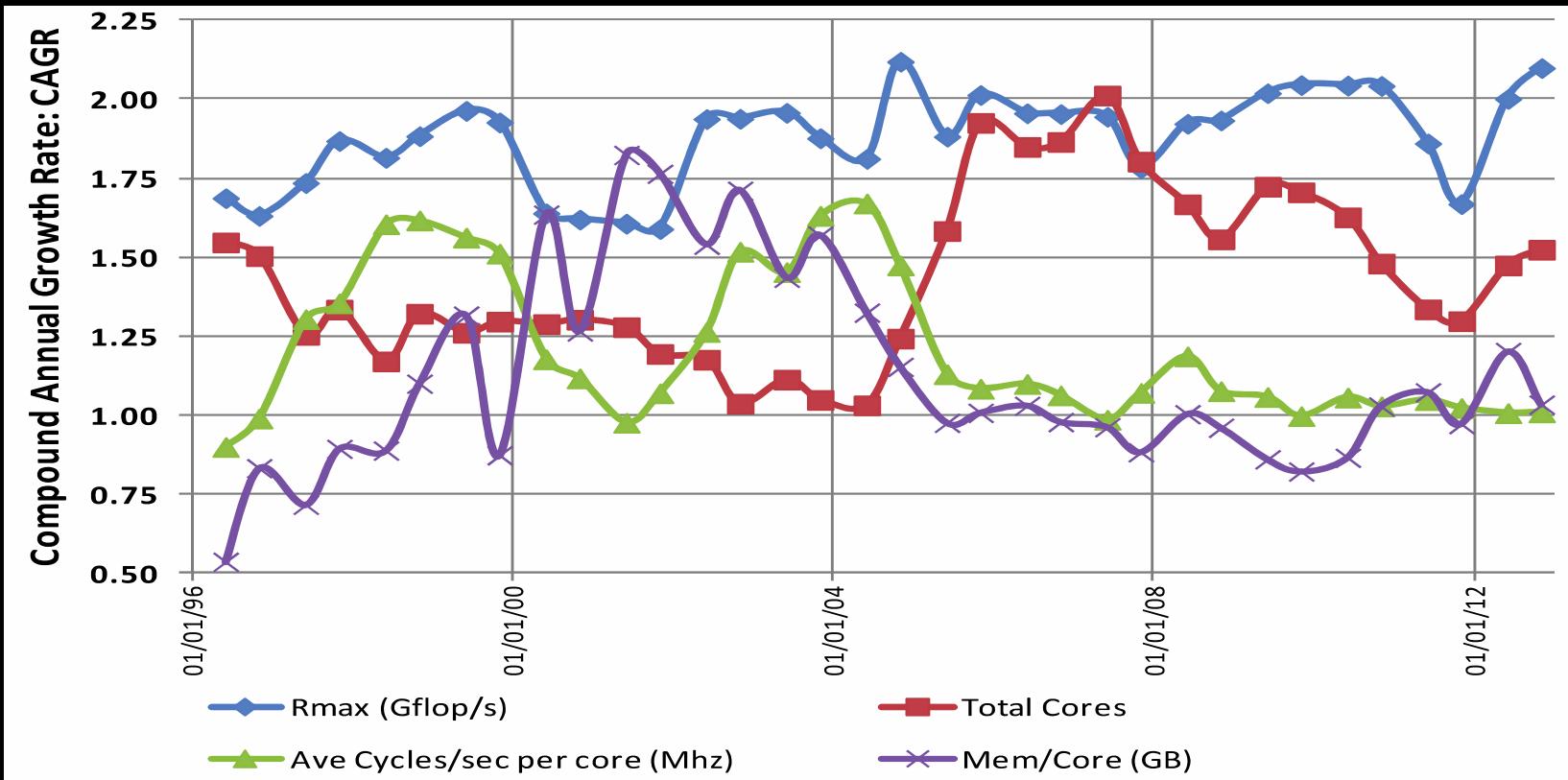
Cray Red Storm  
2004  
42 Tflops

NVIDIA K40  
1.2 Tflops

# Projected Performance Development



# Trends with ends.



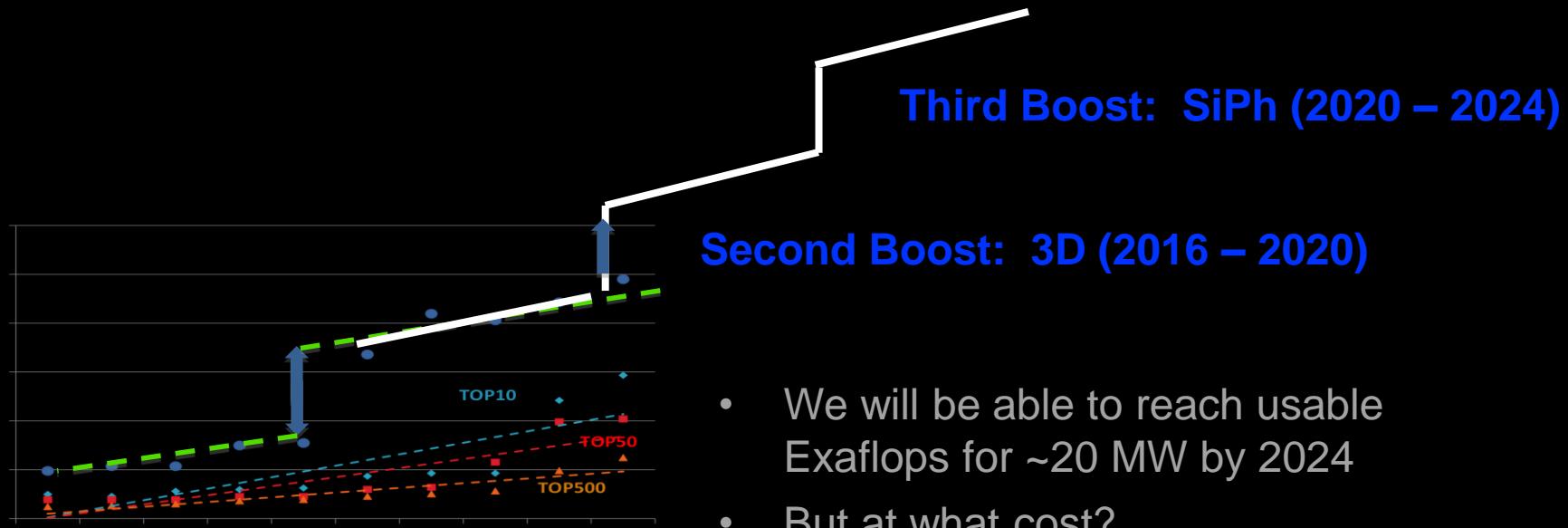
Courtesy Horst Simon, LBNL

# Obstacles?

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-storage
- 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.

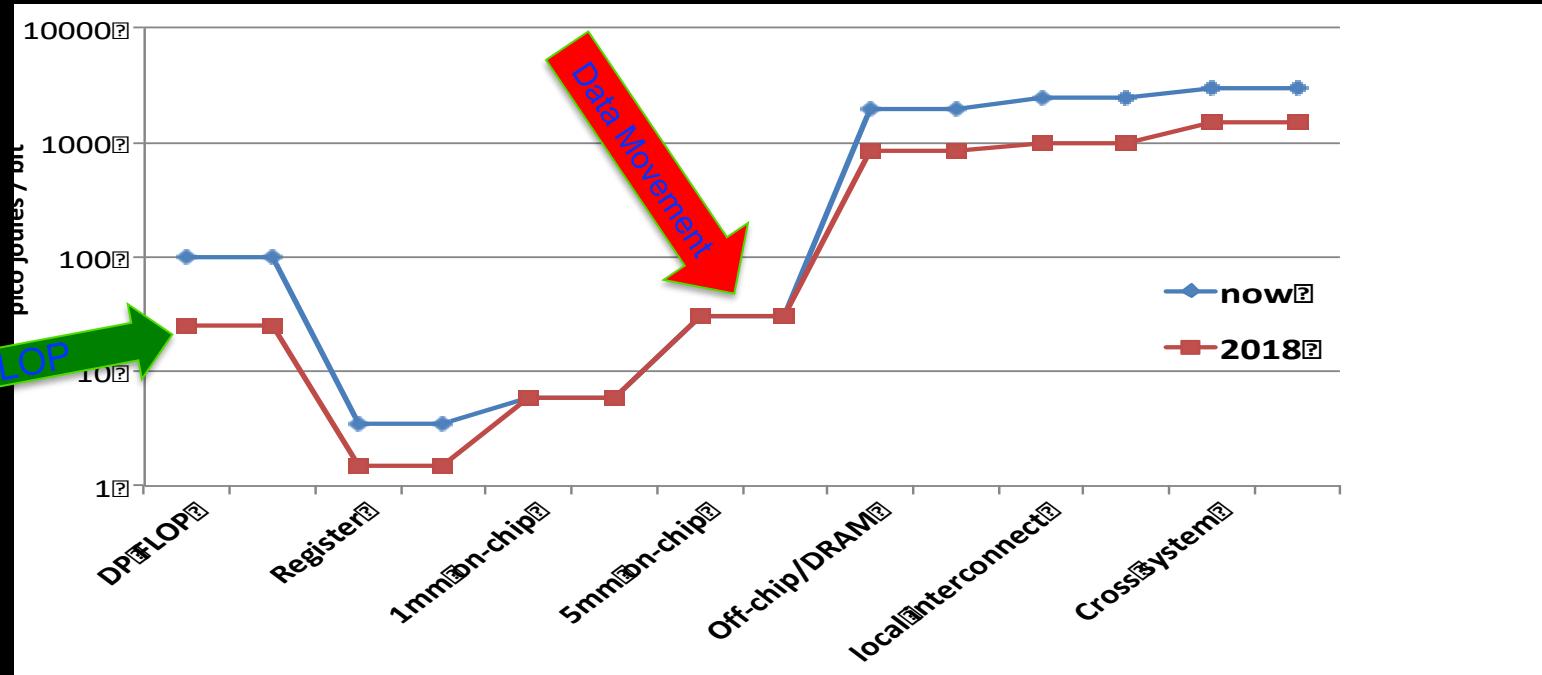
# Two Additional Boosts to Improve Flops/Watt and Reach Exascale Target



- 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?

# Power Issues by 2018

FLOPs will cost less than  
on-chip data movement!



# 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*

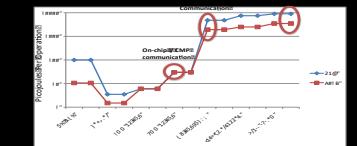
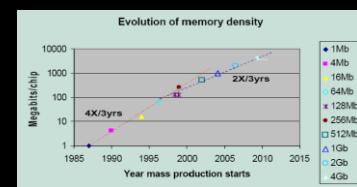
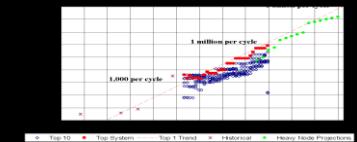
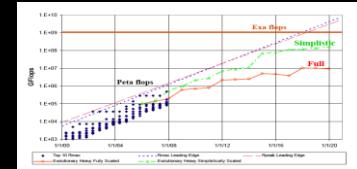
## Old Constraints

- 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*



# End of Moore's Law Will Lead to New Architectures

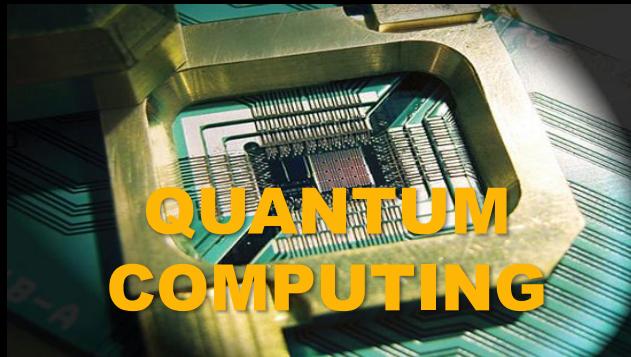
Non-von  
Neumann

ARCHITECTURE

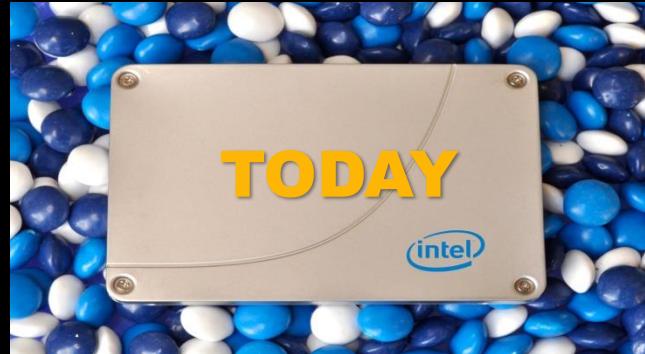
von Neumann



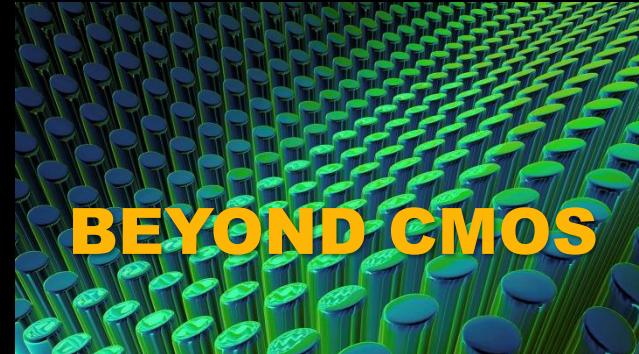
**NEUROMORPHIC**



**QUANTUM  
COMPUTING**



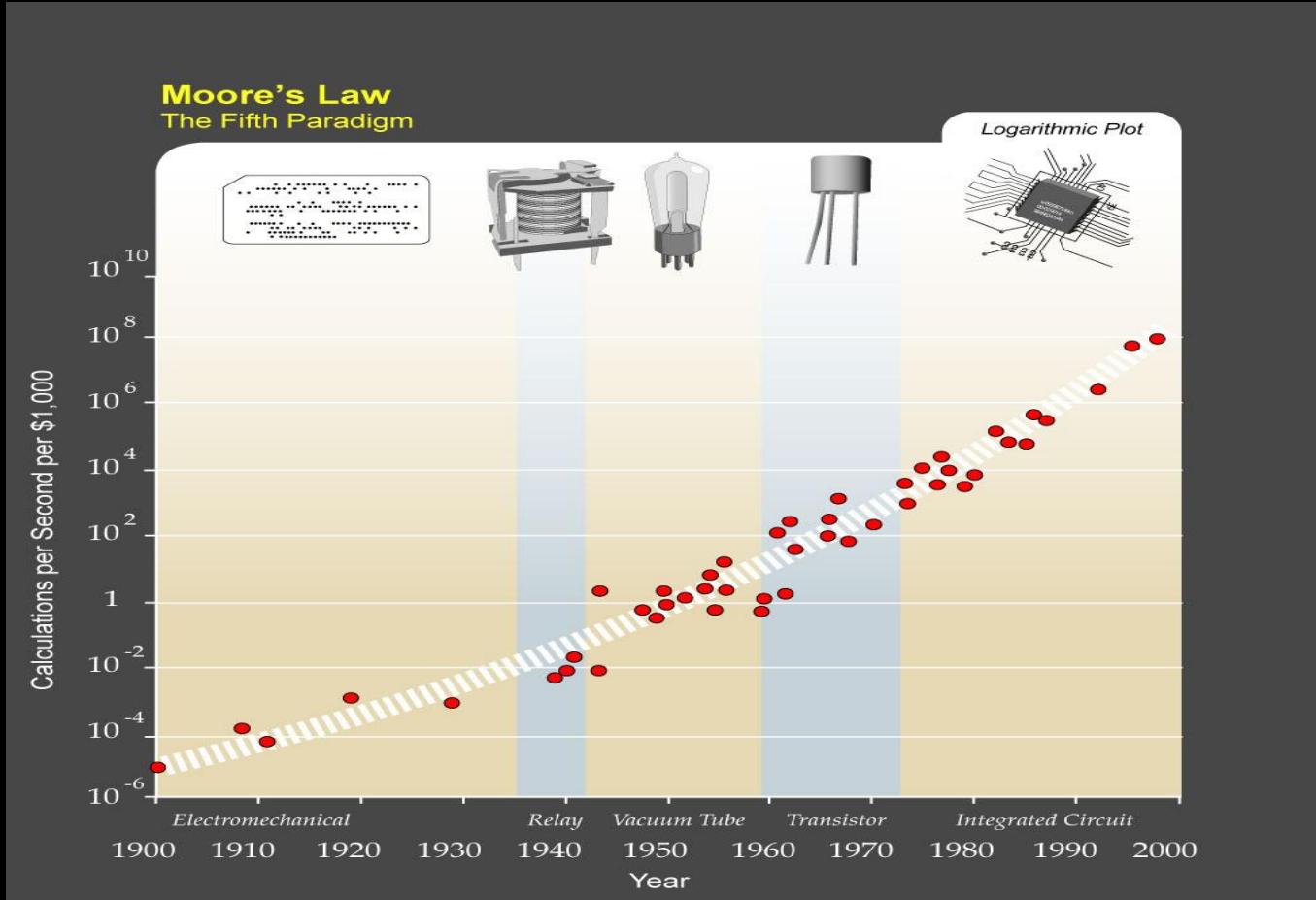
**CMOS**



**TECHNOLOGY**

**Beyond CMOS**

# It would only be the 6<sup>th</sup> 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



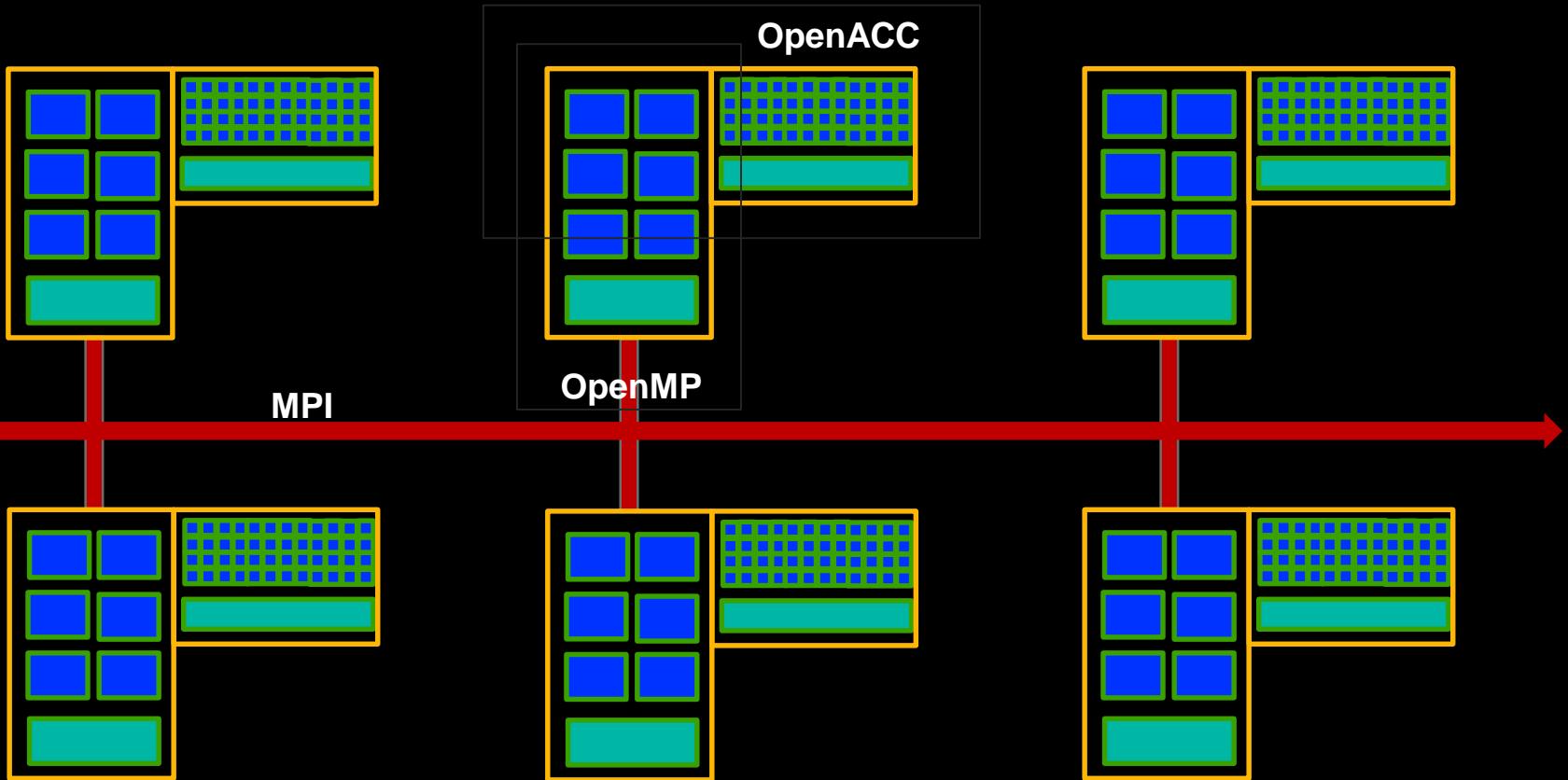
Courtesy Horst Simon, LBNL

# 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...



# 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)