

Intro To Parallel Computing

**John Urbanic
Pittsburgh Supercomputing Center
Parallel Computing Specialist**

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

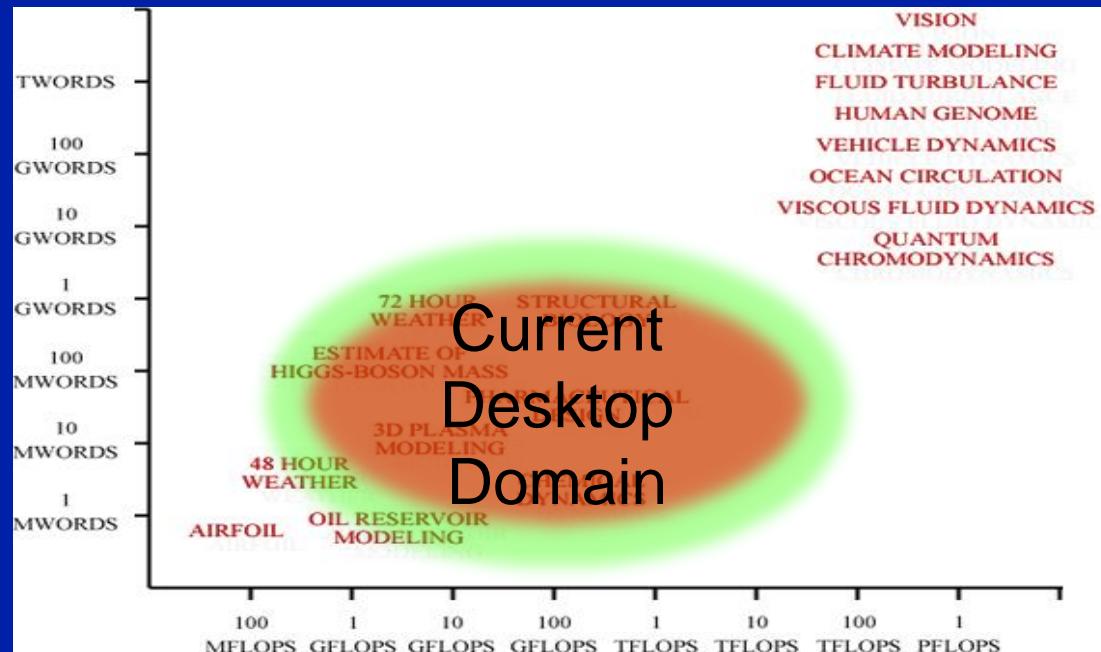
We need Exascale computing

We aren't getting to Exascale without parallel

What does parallel computing look like

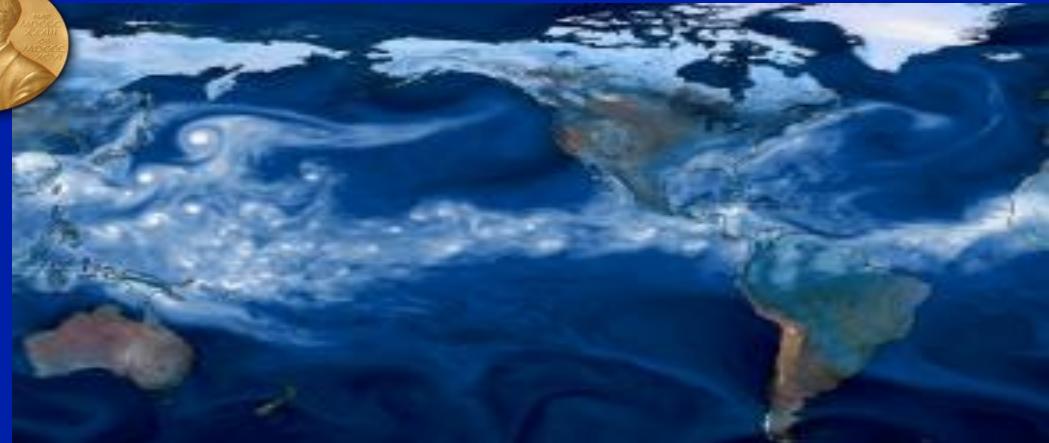
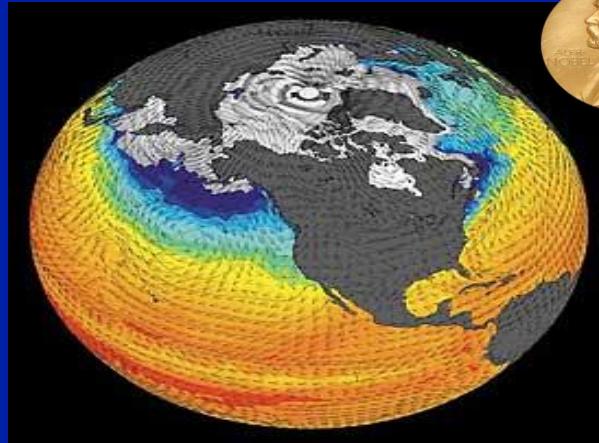
Where is this going

FLOPS we can use now



Which axis is most important?

FLOPS we need: Climate change analysis



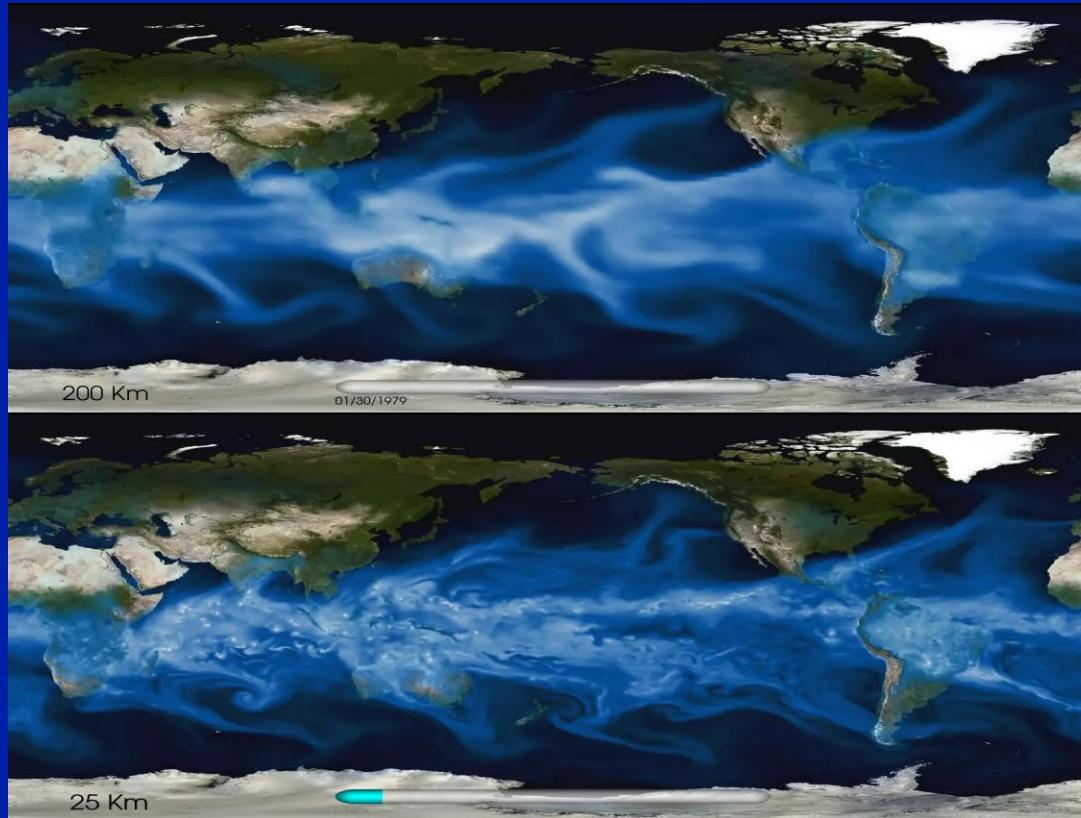
Simulations

- 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

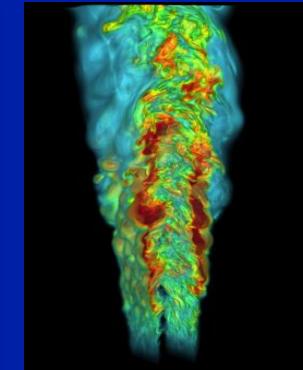
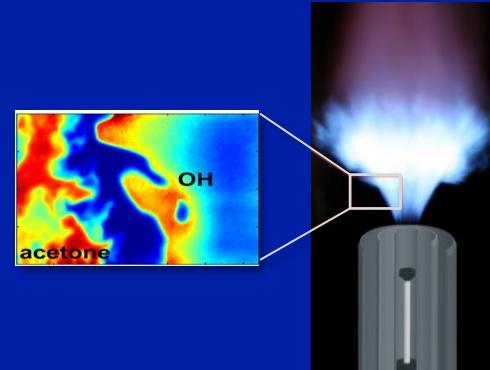
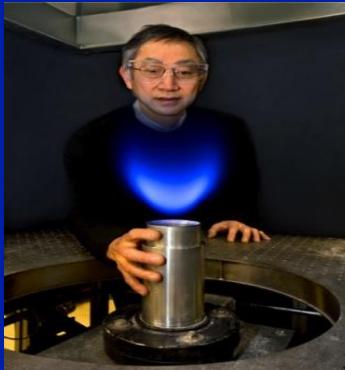
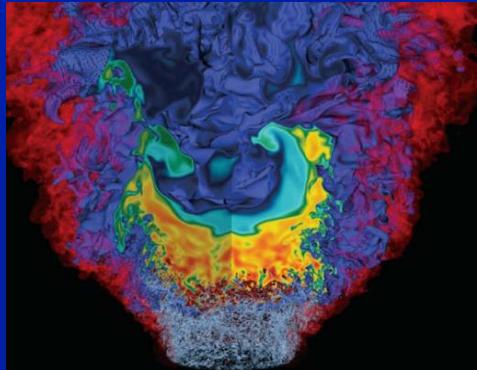
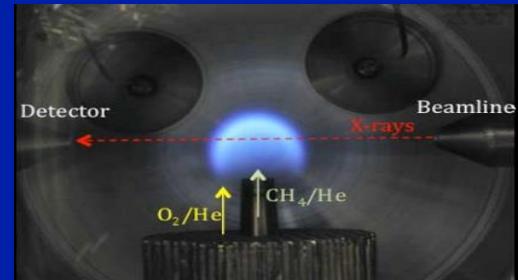
Qualitative Improvement of Simulation with Higher Resolution (2011)



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

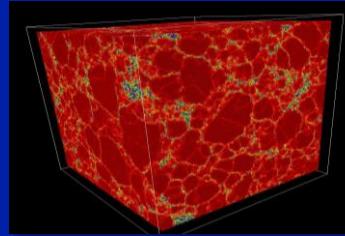
- Goal: 50% improvement in engine efficiency
- Center for Exascale Simulation of Combustion in Turbulence (ExaCT)
 - Combines M&S and experimentation
 - Uses new algorithms, programming models, and computer science



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



Modha Group at IBM Almaden

Mouse	Rat	Cat	Monkey	Human
N: 16×10^6	56×10^6	763×10^6	2×10^9	22×10^9
S: 128×10^9	448×10^9	6.1×10^{12}	20×10^{12}	220×10^{12}



Almaden

BG/L

December, 2006



Watson

BG/L

April, 2007



WatsonShaheen

BG/P

March, 2009



LLNL Dawn

BG/P

May, 2009

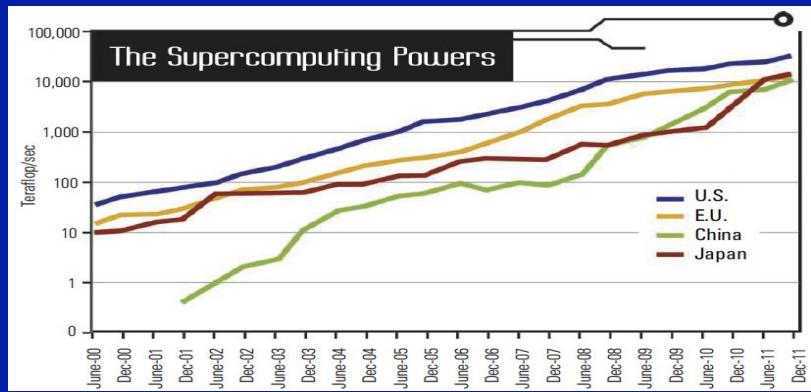
Recent simulations achieve
unprecedented scale of
 65×10^9 neurons and 16×10^{12} synapses

LLNL Sequoia

BG/Q

June, 2012

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



Data from TOP500.org; Figure from Science; Vol 335; 27 January 2012

2nd Theme

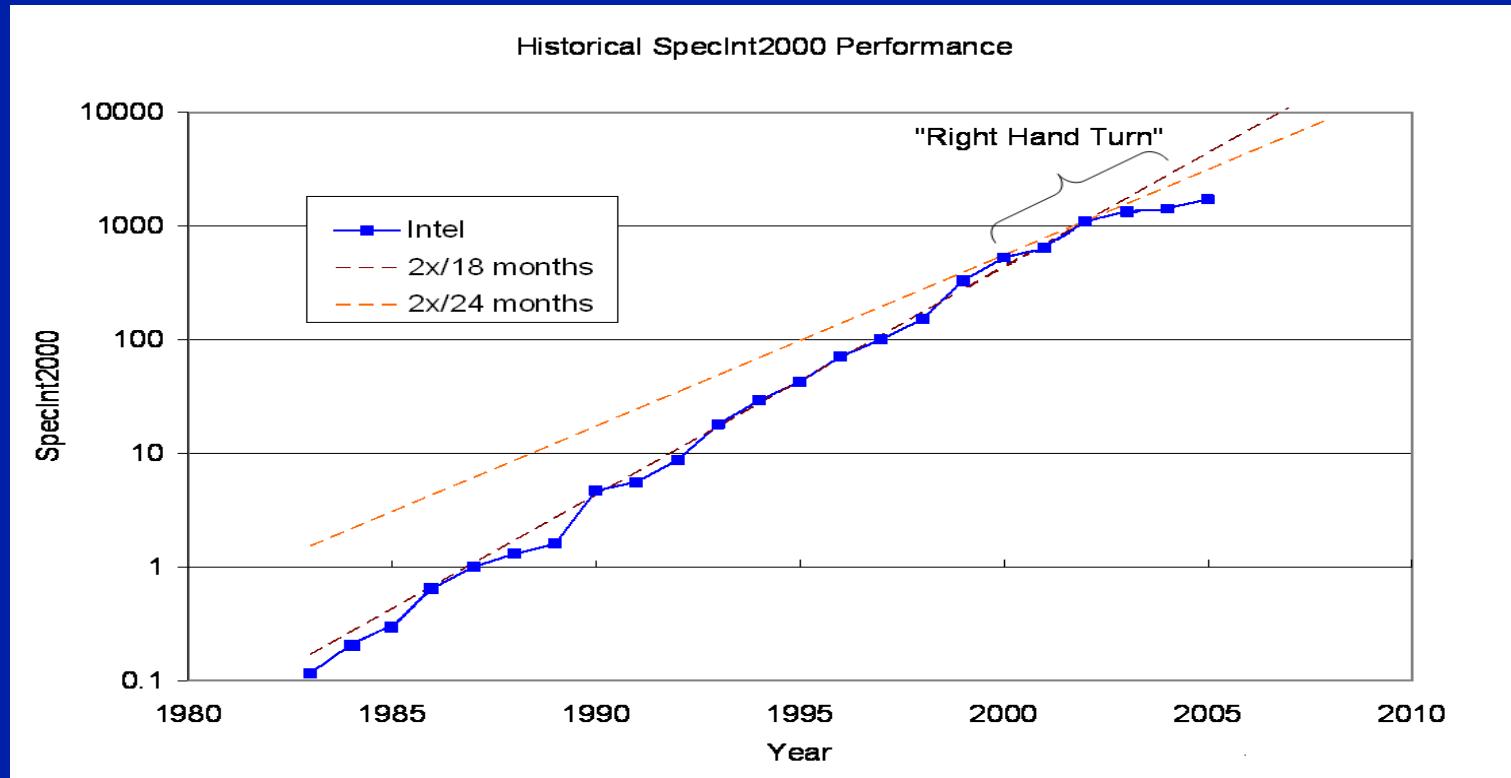
We will have Exascale computing

You aren't getting to Exascale without going very parallel

What does parallel computing look like

Where is this going

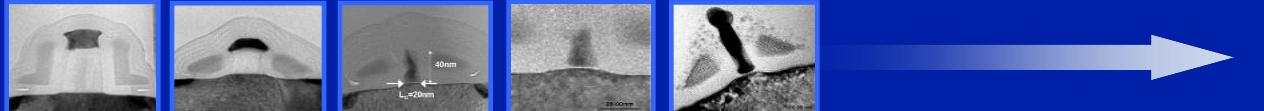
Waiting for Moore's Law to save your serial code start getting bleak in 2004



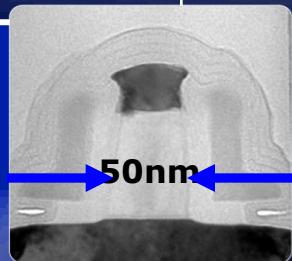
Source: published SPECInt data

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

Intel process technology capabilities

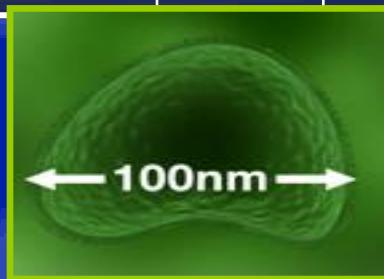


High Volume Manufacturing	2004	2006	2008	2010	2012	2014	2016	2018
Feature Size	90nm	65nm	45nm	32nm	22nm	16nm	11nm	8nm
Integration Capacity (Billions of Transistors)	2	4	8	16	32	64	128	256



Transistor for
90nm Process

Source: Intel

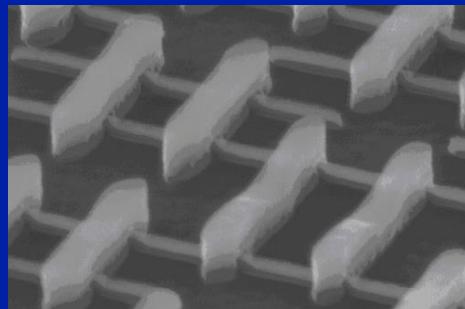


Influenza Virus

Source: CDC

...but our metrics are less clear.

After a while, “there was no one design rule that people could point to and say, ‘That defines the node name’ ... The minimum dimensions are getting smaller, but I’m the first to admit that I can’t point to the one dimension that’s 32 nm or 22 nm or 14 nm. Some dimensions are smaller than the stated node name, and others are larger.”



Mark Bohr, Senior fellow at Intel.

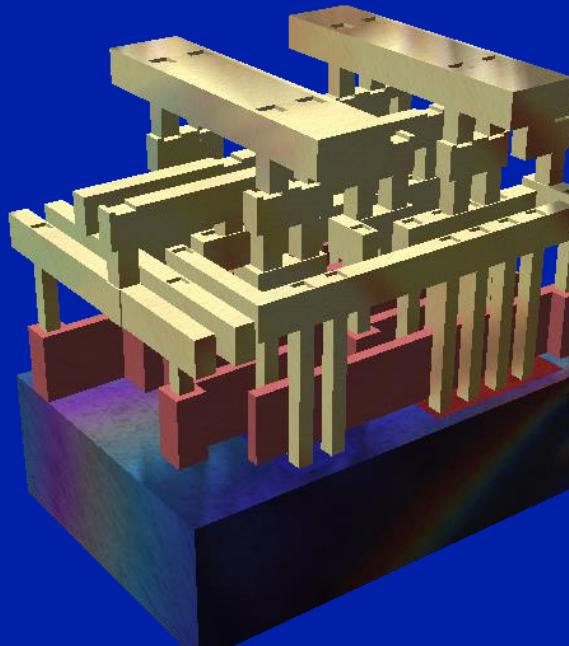
From *The Status of Moore's Law: It's Complicated* (IEEE Spectrum)

For a while things were better than they appeared.

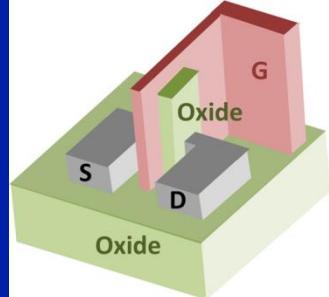
Intel's 0.13- μm chips, which debuted in 2001, had transistor gates that were actually just 70 nm long. Nevertheless, Intel called them 0.13- μm chips because they were the next in line.

Manufacturers continued to pack the devices closer and closer together, assigning each successive chip generation a number about 70 percent that of the previous one.

A 30 percent reduction in both the x and y dimensions corresponds to a 50 percent reduction in the area occupied by a transistor, and therefore the potential to double transistor density on the chip.



Then new technologies carried the load.



After years of aggressive gate trimming, simple transistor scaling reached a limit in the early 2000s: Making a transistor smaller no longer meant it would be faster or less power hungry. So Intel, followed by others, introduced new technologies to help boost transistor performance.

- Strain engineering, adding impurities to silicon to alter the crystal, which had the effect of boosting speed without changing the physical dimensions of the transistor.
- New insulating and gate materials.
- And most recently, they rejigged the transistor structure to create the more efficient FinFET, with a current-carrying channel that juts out of the plane of the chip.

The switch to FinFETs has made the situation even more complex. Intel's 22-nm chips, the current state of the art, have FinFET transistors with gates that are 35 nm long but fins that are just 8 nm wide.

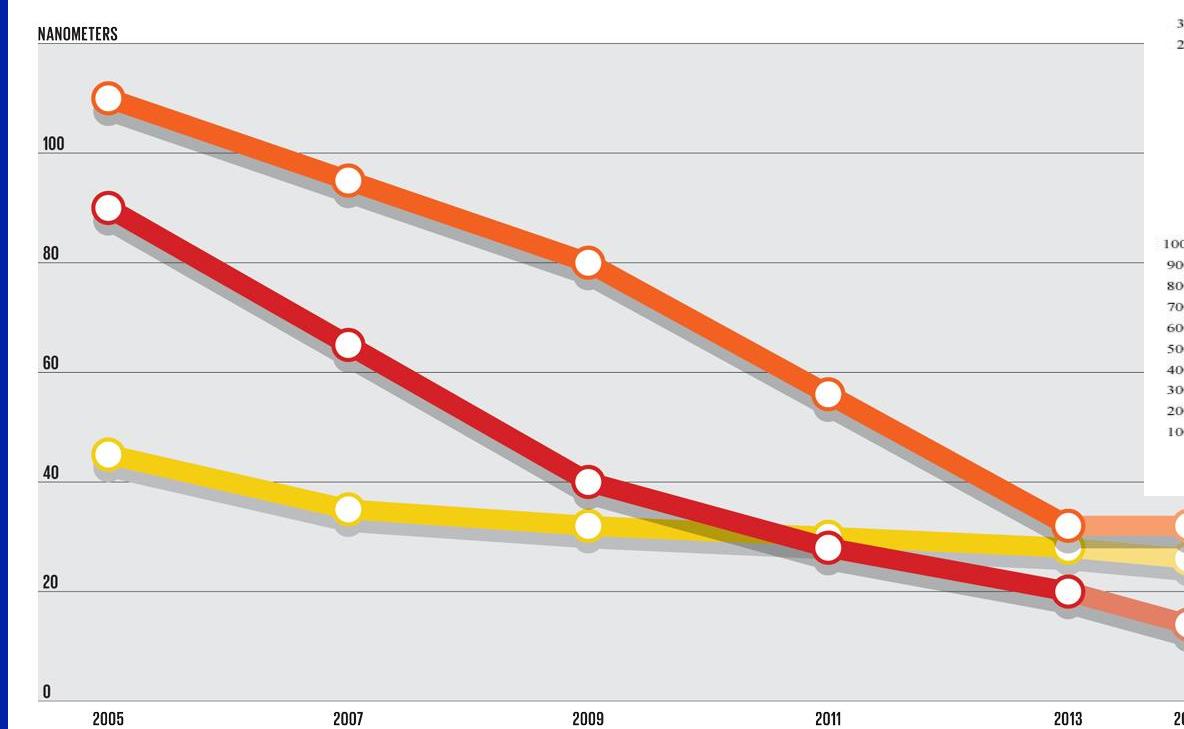
Now tradeoffs are stealing these gains.

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

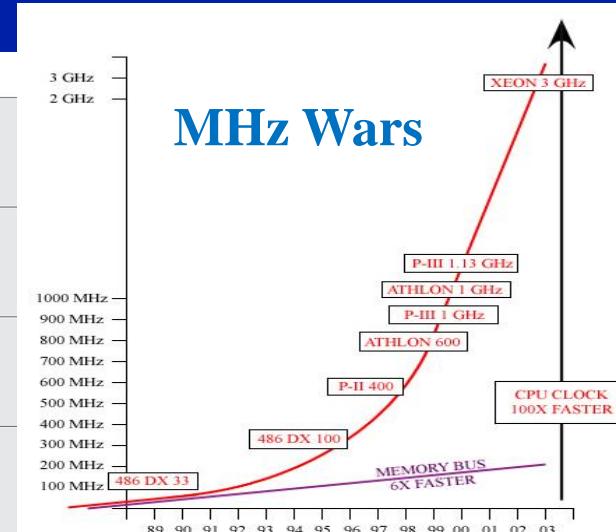
- **error-correction circuitry**
- **redundancy**
- **read- and write-boosting circuitry for failing static RAM cells**
- **circuits to track and adapt to performance variations**
- **complicated memory hierarchies to handle multicore architectures.**

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.

Maybe they are even becoming “marketing”.



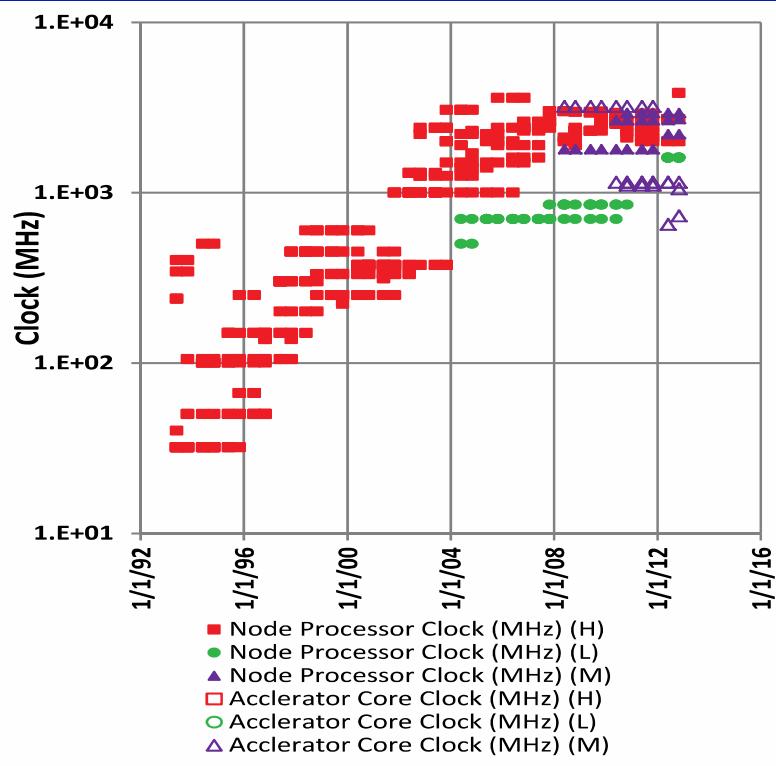
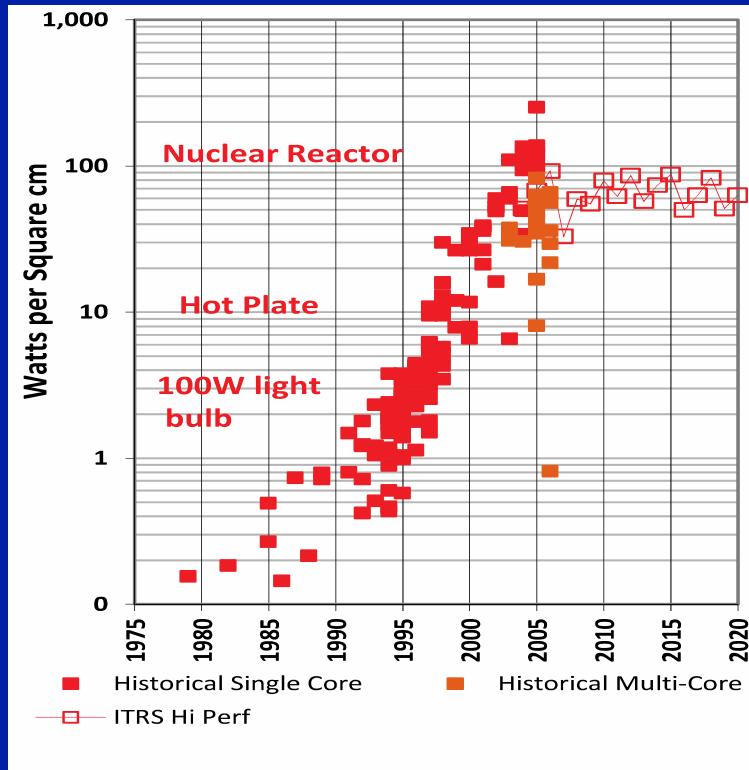
MHz Wars



Global Foundries planned 2013 14nm chip introduction.

As reported by IEEE Spectrum

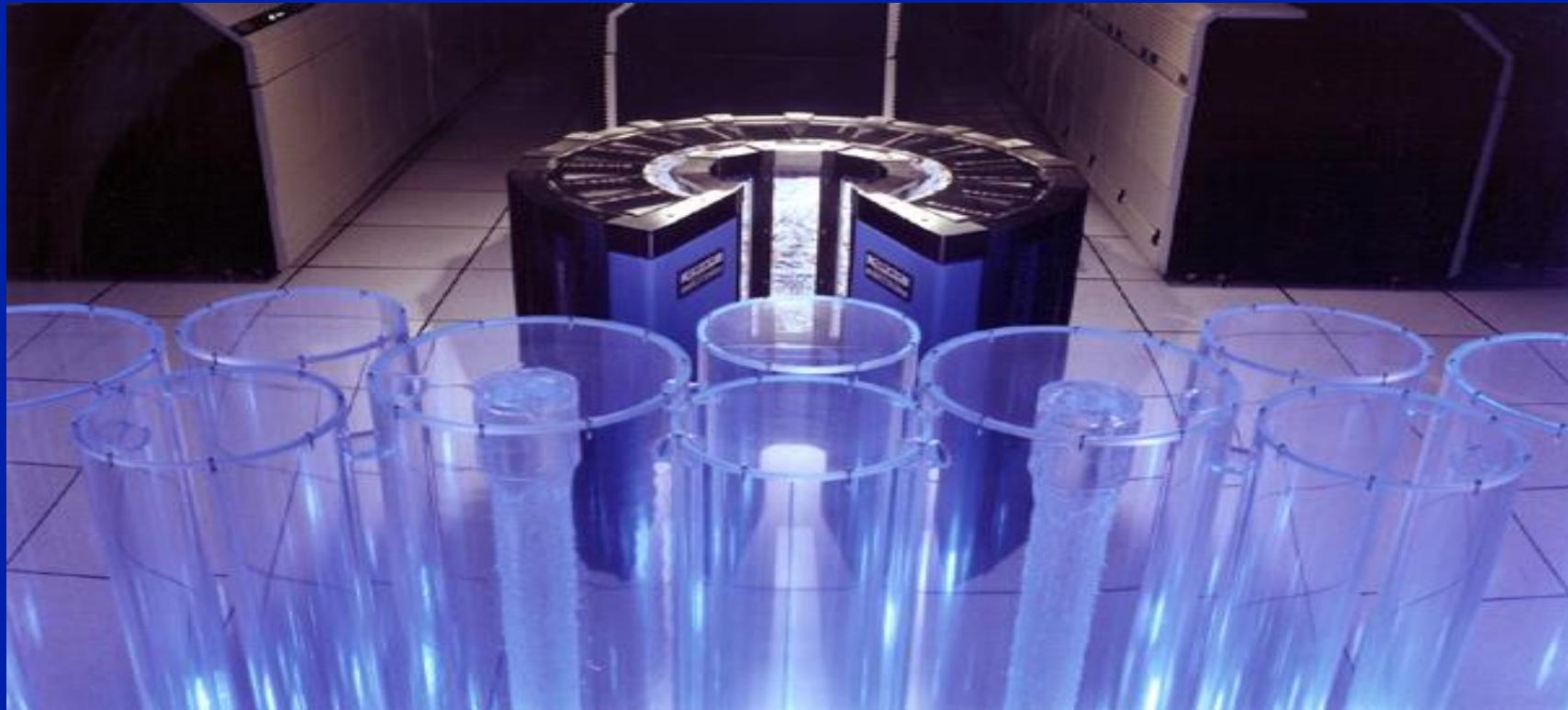
That Power and Clock Inflection Point in 2004... didn't get better.



Source: Kogge and Shalf, IEEE CISE

Courtesy Horst Simon, LBNL

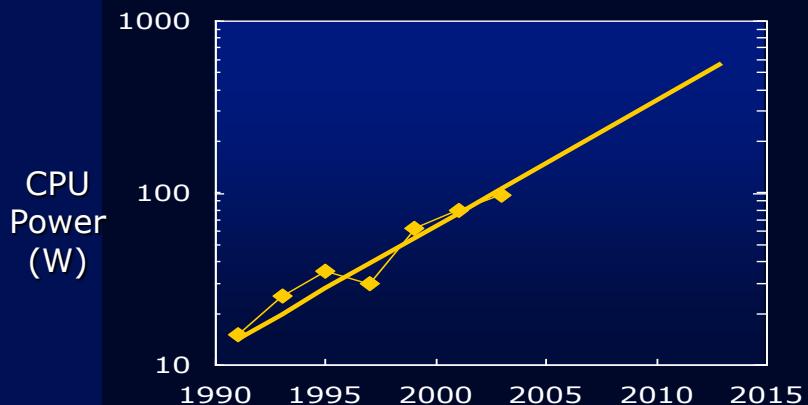
Not a new problem, just a new scale...



Cray-2 with cooling tower in foreground, circa 1985

Basic Energy Efficiency Performance Parameters

Power Limitations



$$\text{Power} = C \times V^2 \times \text{Frequency}$$

$$\text{Frequency} \sim \text{Voltage}$$

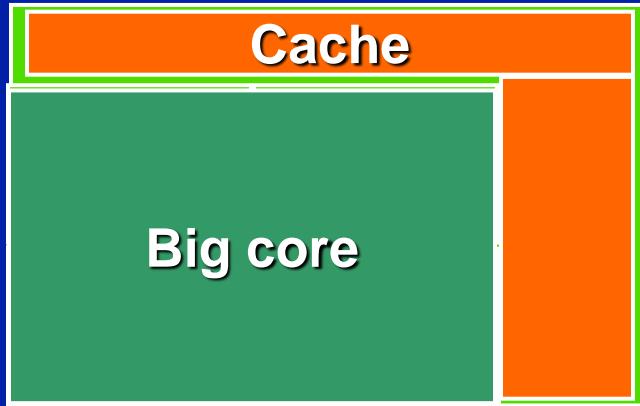
$$\text{Power} \sim \text{Voltage}^3$$

$C \sim$ transistors switching

Reduce frequency and voltage to get a cubic reduction in power

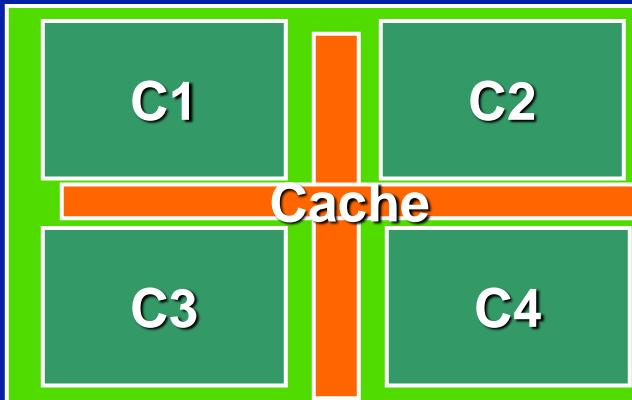
Use more transistors for performance

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



Key is that

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



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

$\text{Performance} = 1/2$

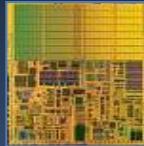
And how to get more performance from more transistors with the same power.

A 15%
Reduction
In Voltage
Yields

RULE OF THUMB

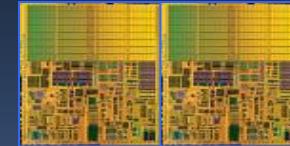
Frequency Reduction	Power Reduction	Performance Reduction
15%	45%	10%

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

3rd Theme

We will have Exascale computing

You will get there by going very parallel

What does parallel computing look like?

Where is this going

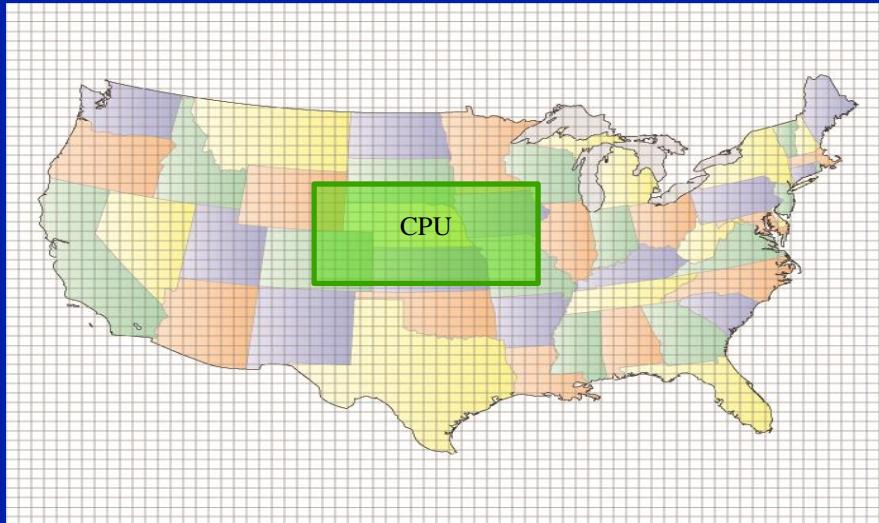
Parallel Computing

One woman can make a baby in 9 months.

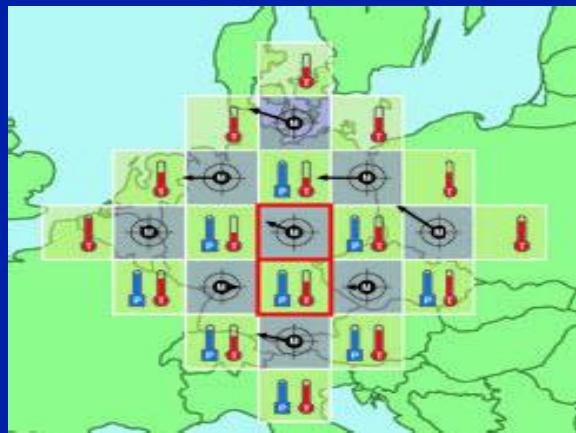
Can 9 women make a baby in 1 month?

But 9 women can make 9 babies in 9 months.

Prototypical Application: Serial Weather Model

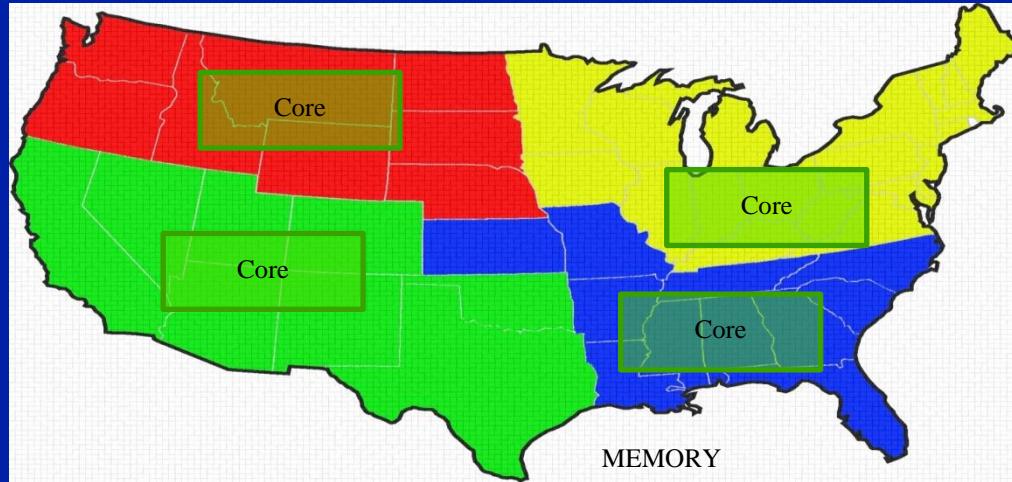


First parallel Weather Modeling algorithm: Richardson in 1917



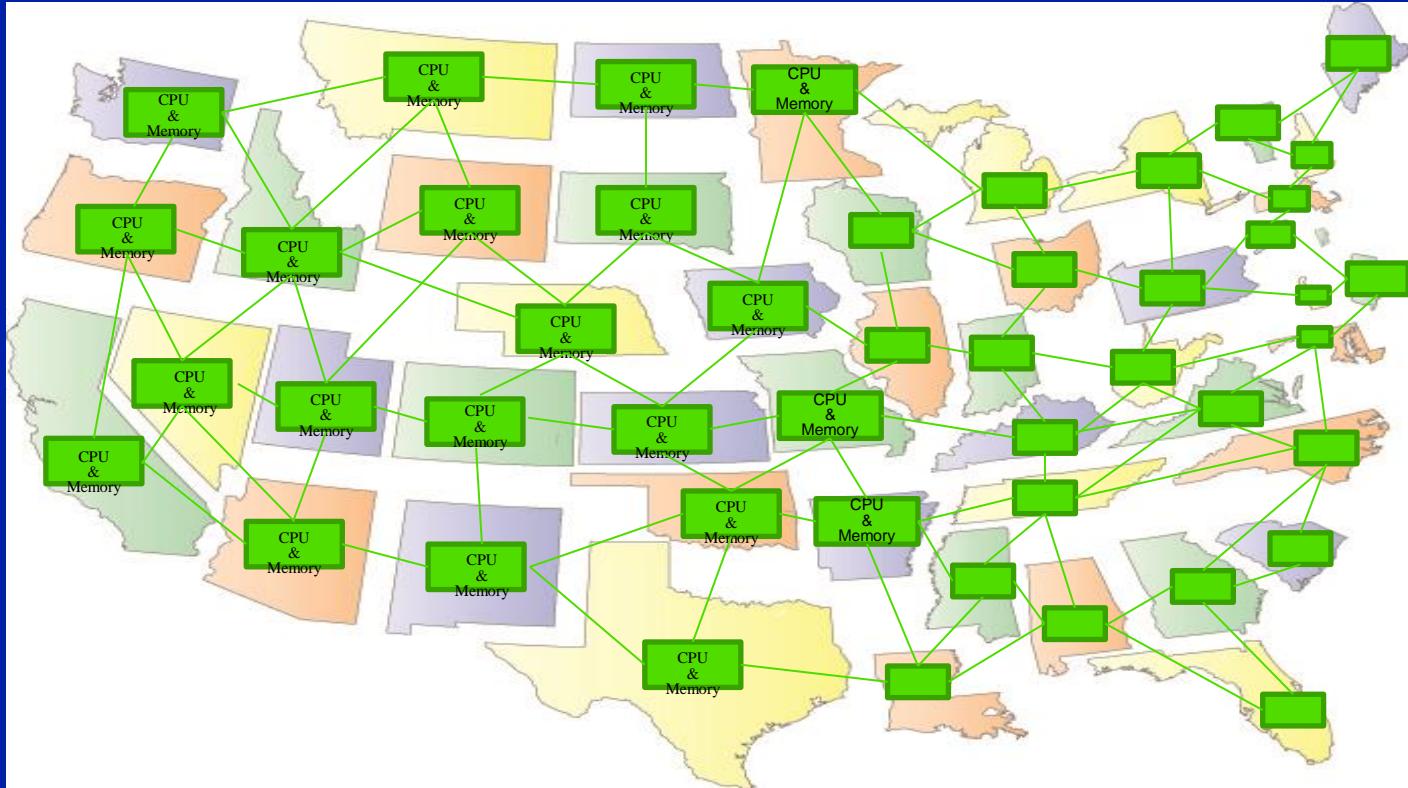
Courtesy John Burkhardt, Virginia Tech

Weather Model: Shared Memory



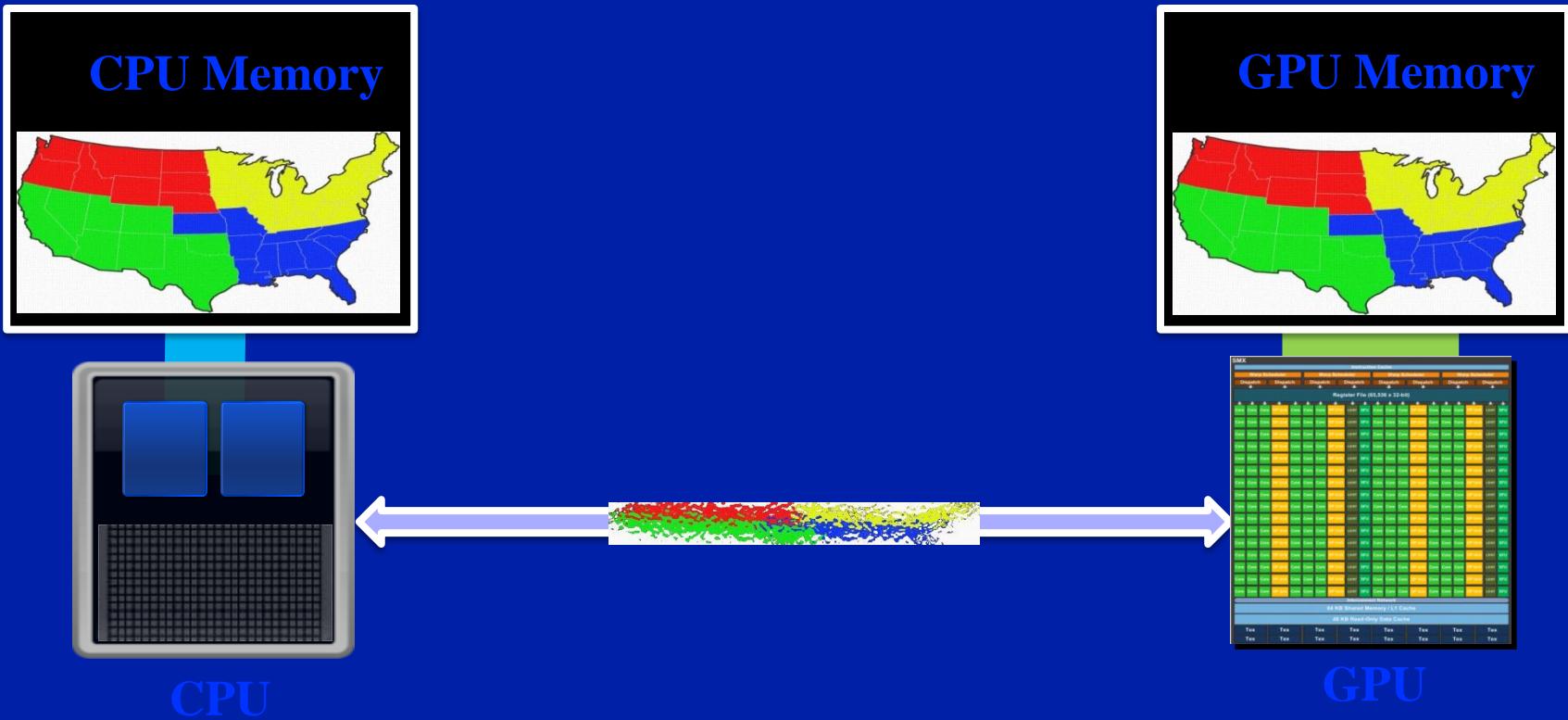
Four guys in the same room sharing the map.

Weather Model: Distributed Memory



50 guys using a telegraph.

Weather Model: Accelerator

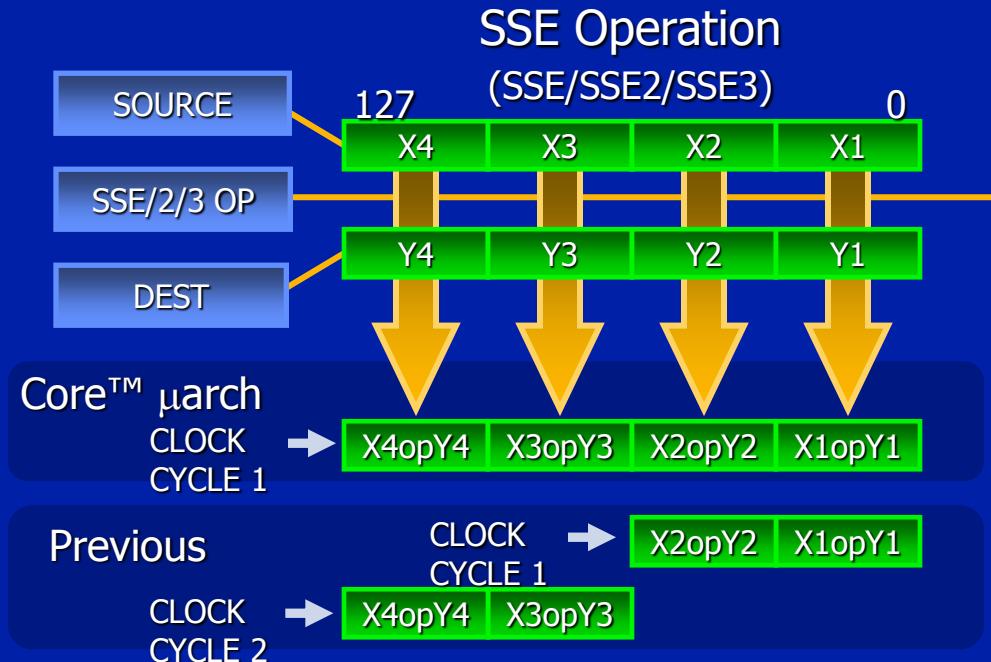
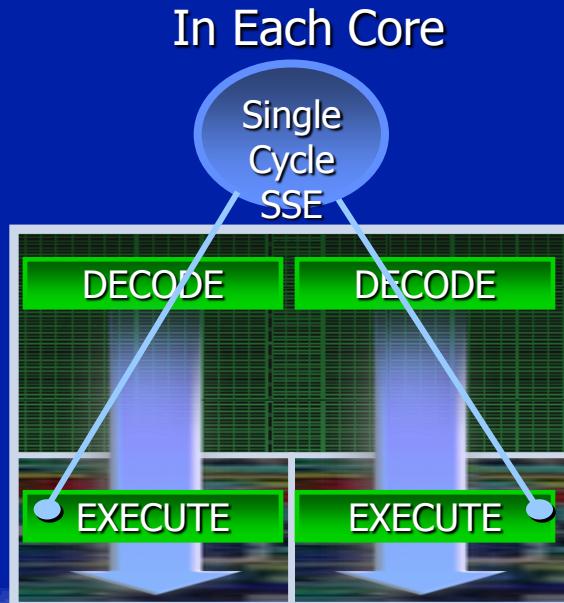


1 guy coordinating 1000 using tin cans and a string.

Many levels and types of parallelism:

- Instruction
- Multi-Core
- SMP/Multi-socket
- Clusters
- MPPs
- *Accelerators: GPU & MIC, soon APU*
- ASIC/FPGA/DSP
- RAID/I/O

Instruction Level: Intel® SSE



Perf ↑
Energy ↓

SIMD instructions compute multiple operations per instruction

*Graphics not representative of actual die photo or relative size

Multi-Core: Just a few of the current offerings

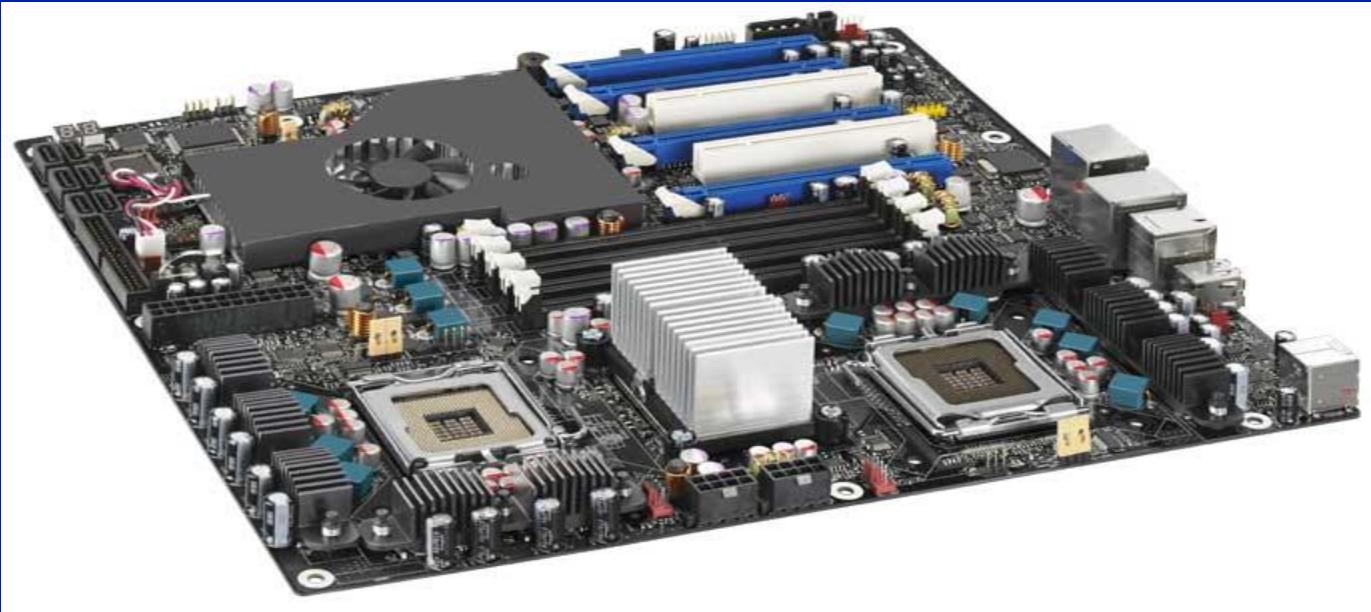
- AMD
 - Opteron (16 core, Bulldozer based Interlagos)
 - Athlon
 - Phenom II
 - Radeon
- ARM
 - MPCore (ARM9 and ARM11)
- Broadcom
 - SiByte
- Cavium Networks
 - Octeon (16 MIPS cores)
- IBM
 - Cell (Sony, and Toshiba)
 - POWER4,5,6,7 (8 core)
- Intel
 - Core i7 (6 core)
 - Xeon (10 core)
 - Itanium 2
 - MIC (60+ core)
- Microsoft
 - Xbox 360 (IBM 3 cores)
- Motorola
 - Freescale dual-core PowerPC
- NetLogic
 - XLP MPIS64 (32 core)
- Picochip
 - DSP devices (300 16-bit processor MIMD cores on one die)
- Stream Processors
 - Storm-1 (2 MIPS CPUs and DSP)
- Sun Microsystems
 - UltraSPARC IV, UltraSPARC IV+,
 - UltraSPARC T1, T2, T3, T5 (16 cores)
- Tilera
 - TILE-Gx (100cores)

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

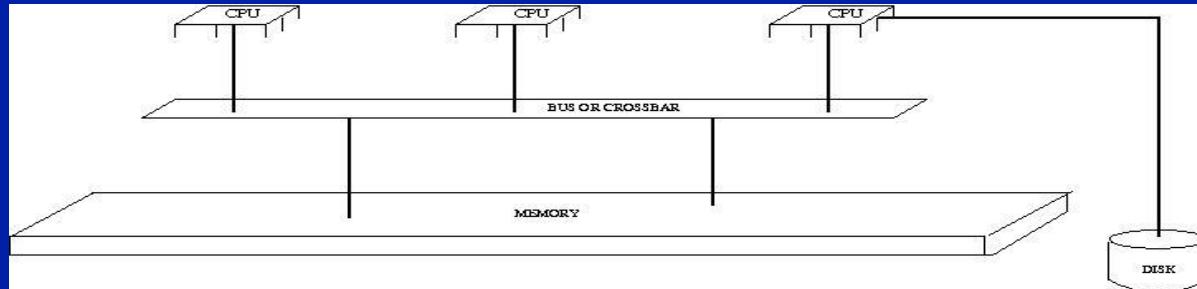
Each processor can access the entire data space

- Pro's

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

- Con's

- 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

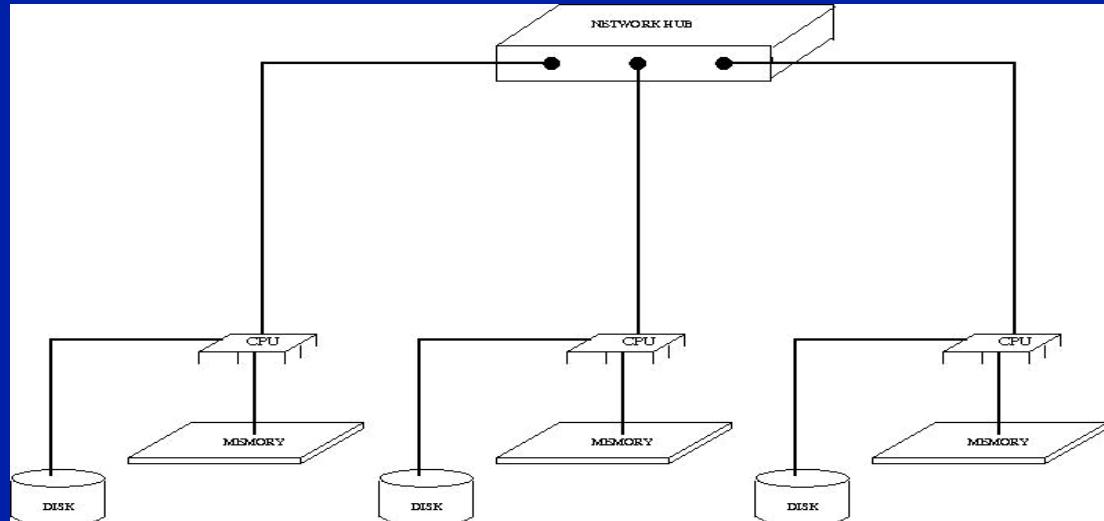
- Each node in the computer has a locally addressable memory space
- The computers are connected together via some high-speed network
 - Infiniband, Myrinet, Giganet, etc..

- 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



Networks

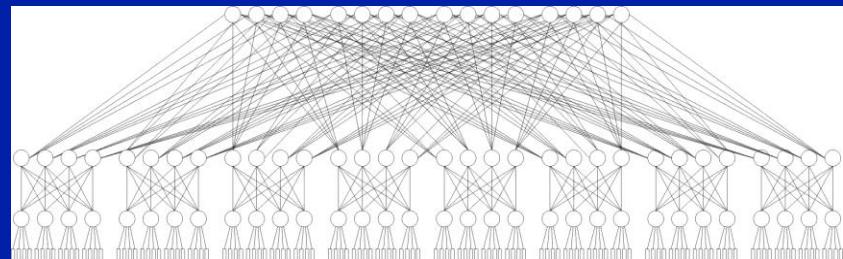
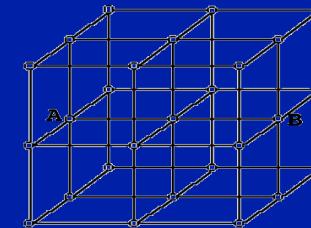
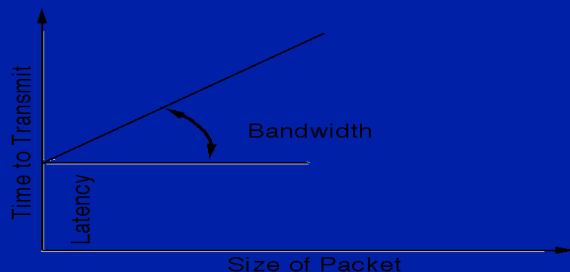
3 characteristics sum up the network:

- **Latency**

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

- **Bandwidth**

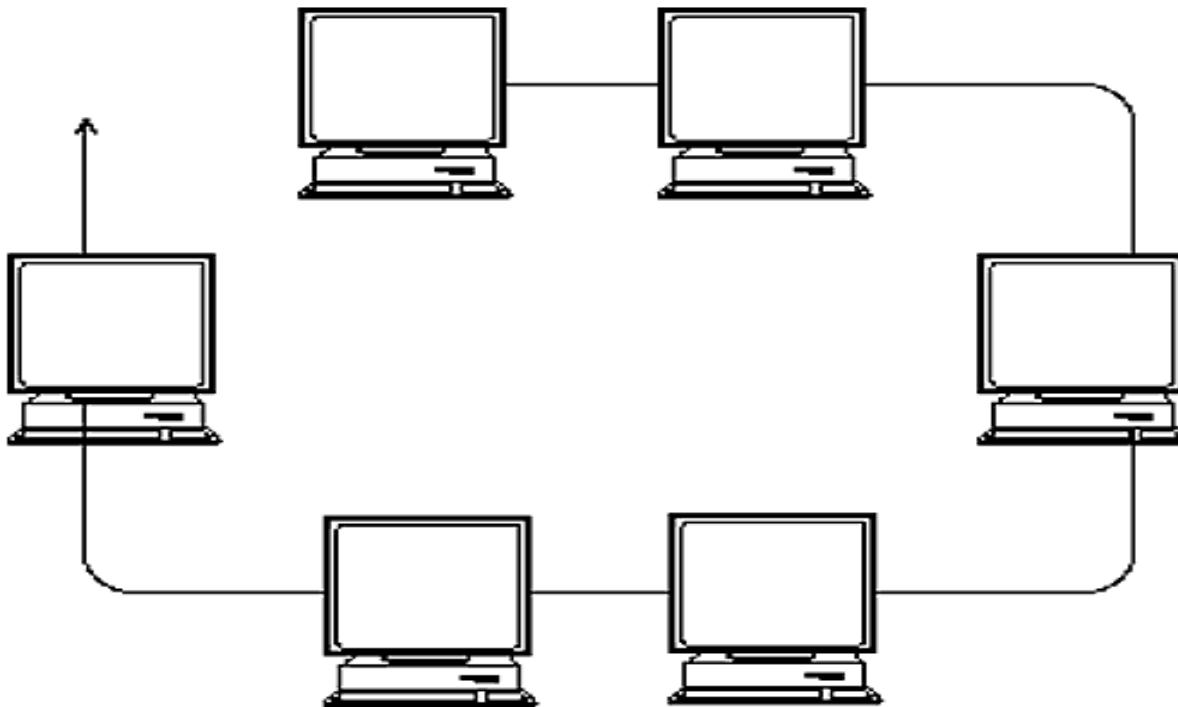
The rate at which a very large packet of information can be sent



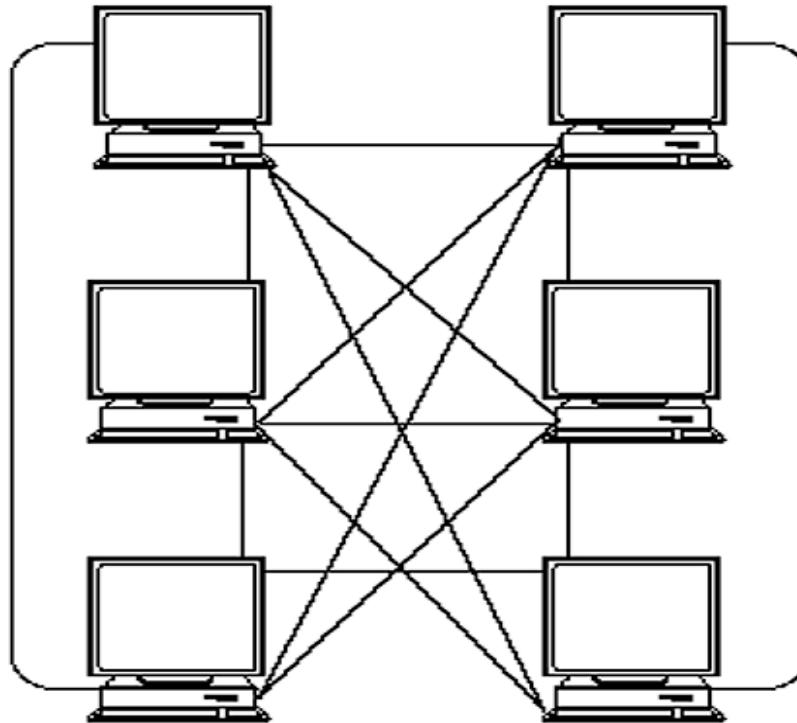
- **Topology**

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

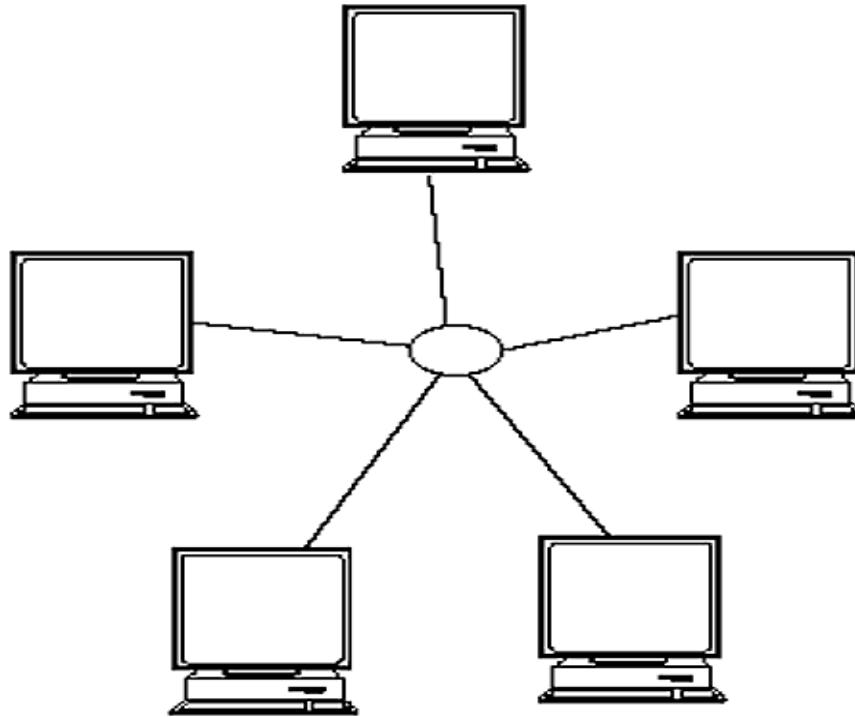
Ethernet with Workstations



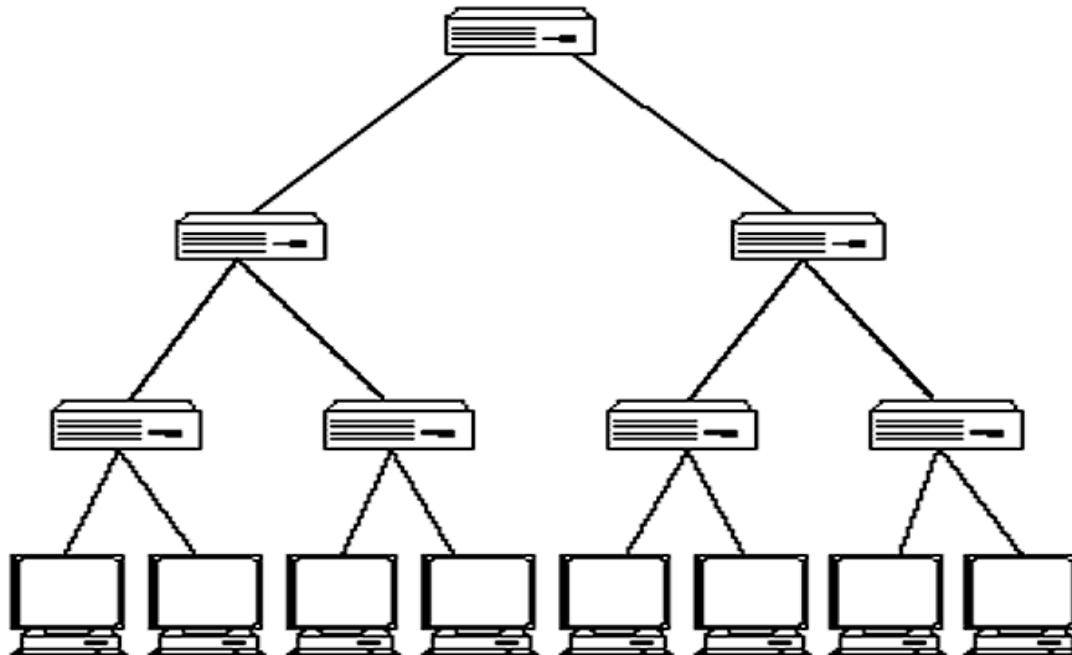
Complete Connectivity



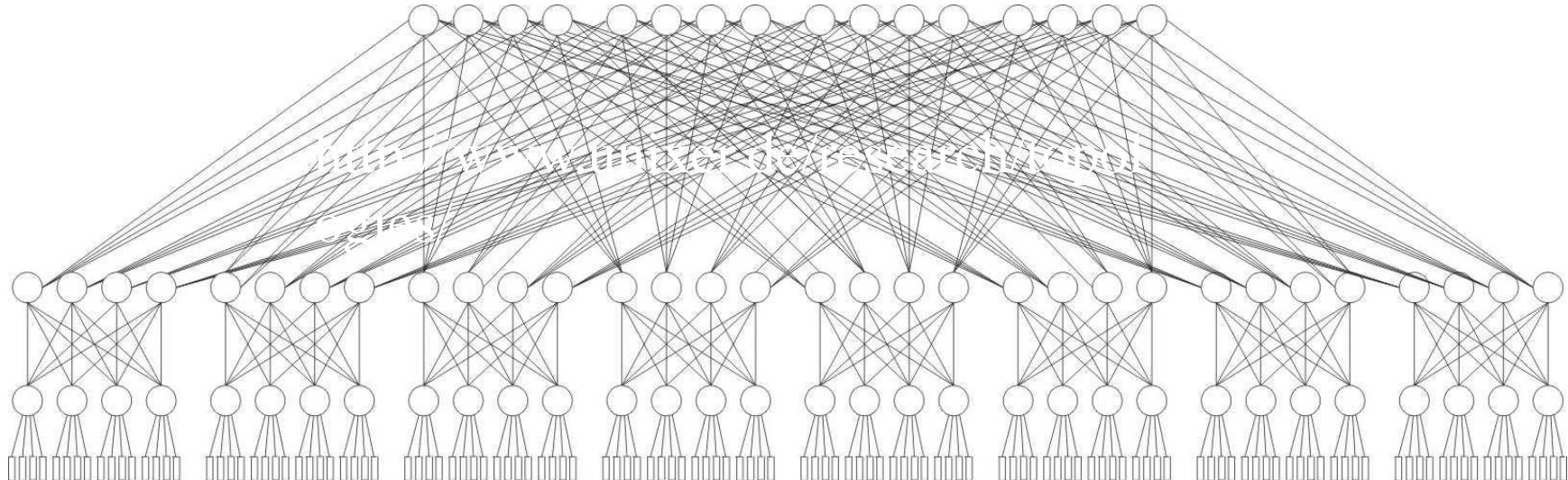
Crossbar



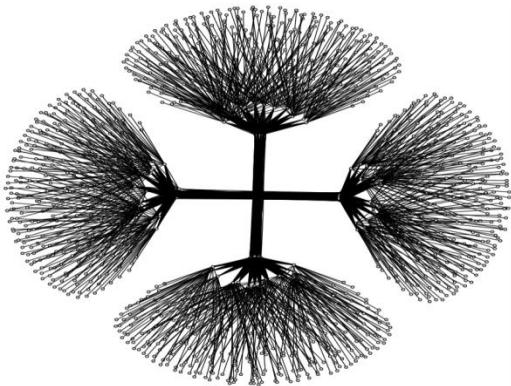
Binary Tree



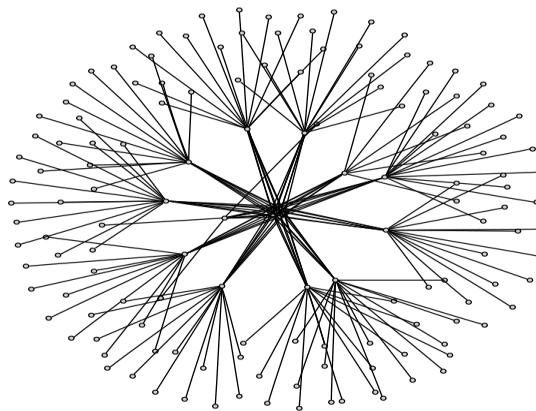
TCS Fat Tree



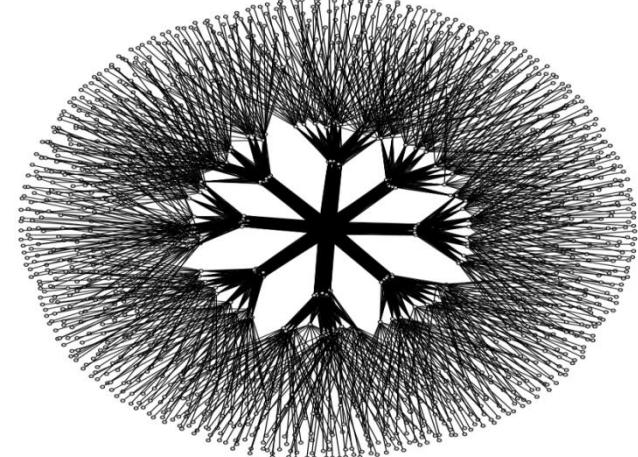
Other Fat Trees



Big Red @ IU

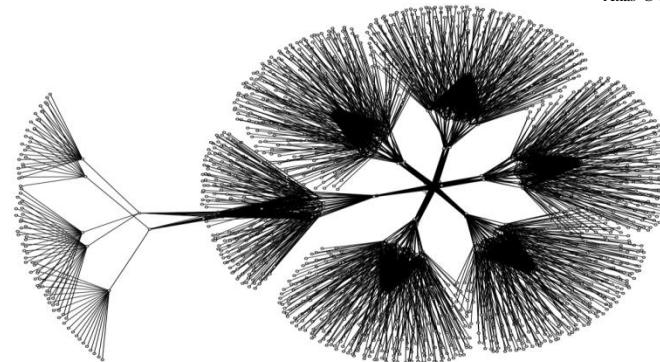
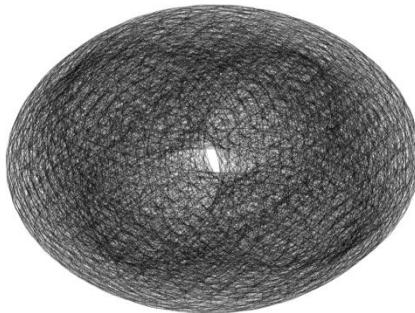


Odin @ IU



Atlas @ LLNL

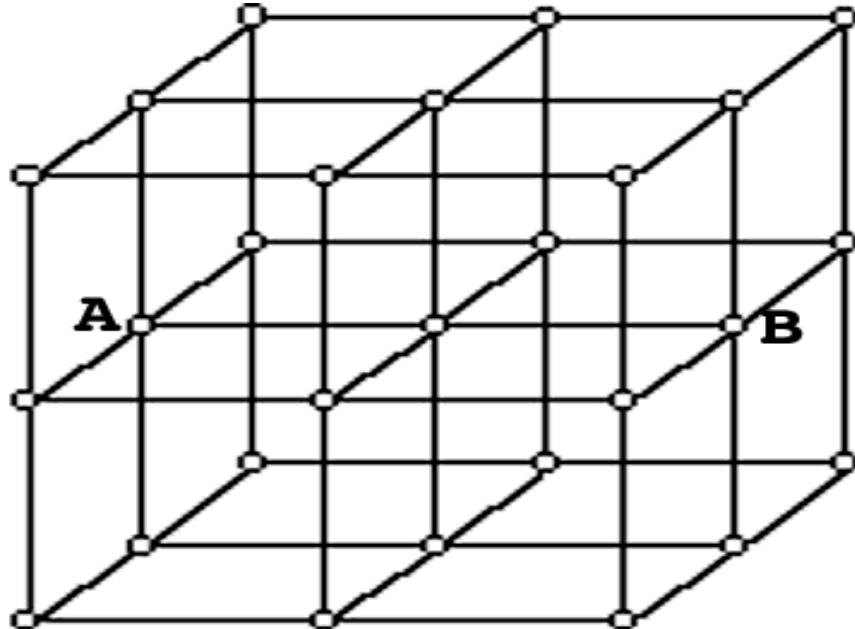
Jaguar @ ORNL



Tsubame @ Tokyo Inst. of Tech

From Torsten Hoefer'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.

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

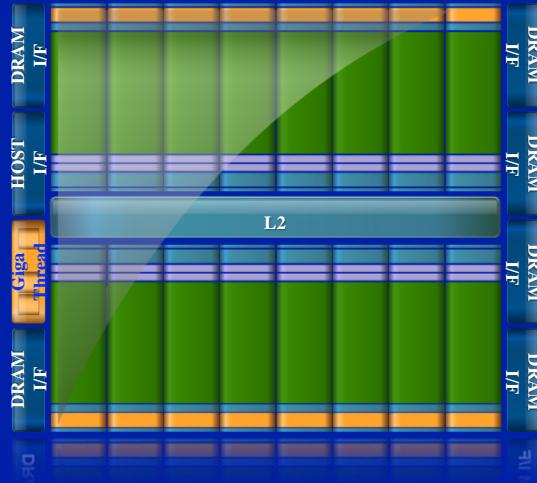
GPU Architecture: Two Main Components

- **Global memory**

- Analogous to RAM in a CPU server
- Accessible by both GPU and CPU
- Currently up to 24 GB
- Bandwidth currently up to almost 288 GB/s for Tesla products
- ECC on/off option for Quadro and Tesla products

- **Streaming Multiprocessors (SMs)**

- Perform the actual computations
- Each SM has its own:
 - Control units, registers, execution pipelines, caches



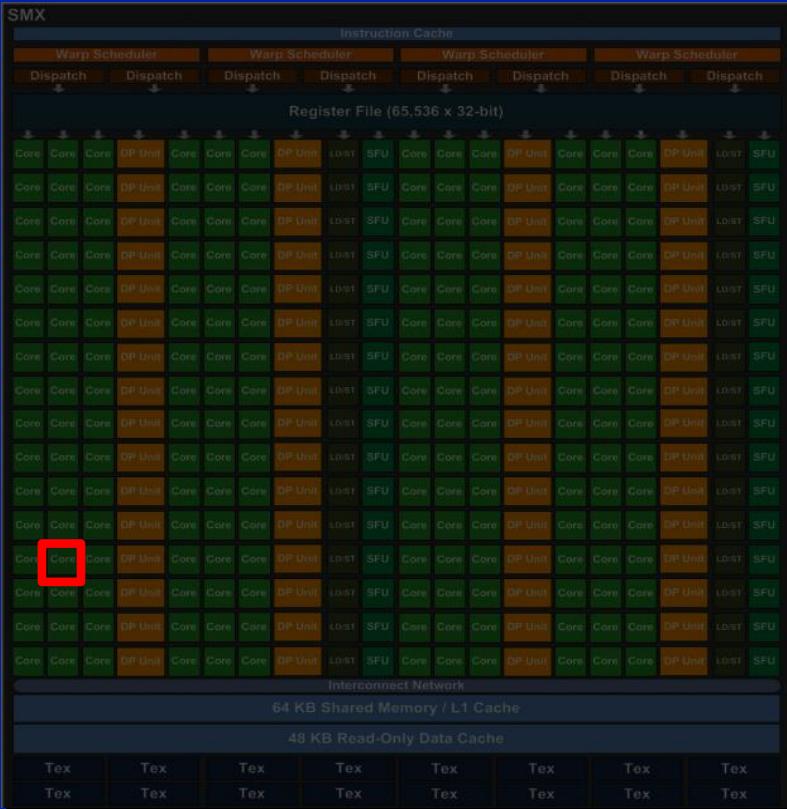
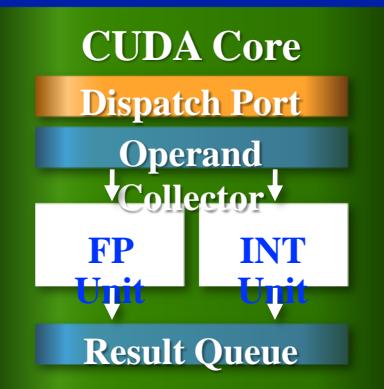
GPU Architecture – Kepler: Streaming Multiprocessor (SMX)

- 192 SP CUDA Cores per SMX
 - 192 fp32 ops/clock
 - 192 int32 ops/clock
- 64 DP CUDA Cores per SMX
 - 64 fp64 ops/clock
- 4 warp schedulers
 - Up to 2048 threads concurrently
- 32 special-function units
- 64KB shared mem + L1 cache
- 48KB Read-Only Data cache
- 64K 32-bit registers

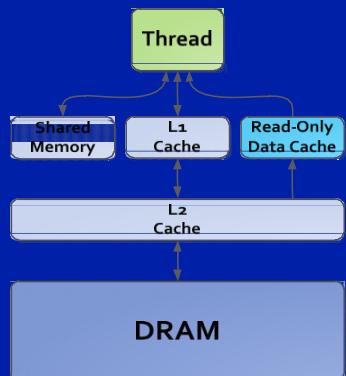
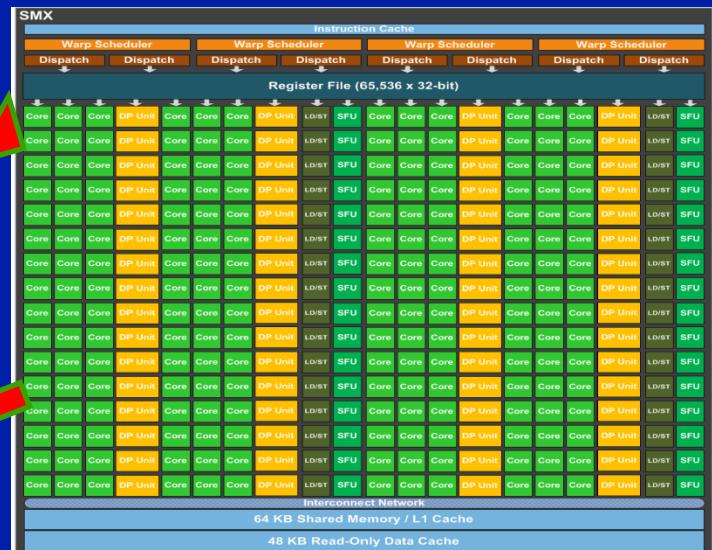


GPU Architecture – Kepler: CUDA Core

- Floating point & Integer unit
 - IEEE 754-2008 floating-point standard
 - Fused multiply-add (FMA) instruction for both single and double precision
- Logic unit
- Move, compare unit
- Branch unit



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?”

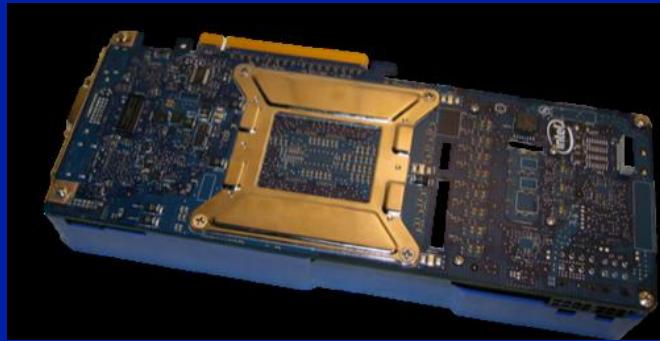
The Intel Many Integrated Core (MIC) architecture is about large die, simpler circuit, much more parallelism, in the x86 line.



What is MIC?

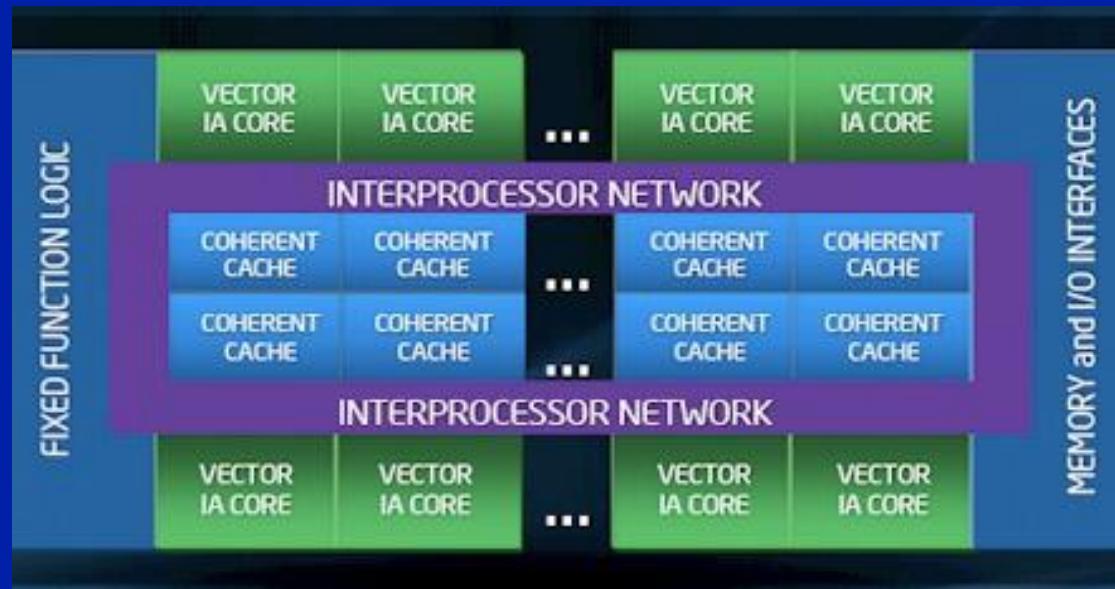
Basic Design Ideas:

- Leverage x86 architecture (a CPU with many cores)
- Use x86 cores that are simpler, but allow for more compute throughput
- Leverage existing x86 programming models
- Dedicate much of the silicon to floating point ops., keep some cache(s)
- Keep cache-coherency protocol
- Increase floating-point throughput per core
- Implement as a separate device
- Strip expensive features (out-of-order execution, branch prediction, etc.)
- Widened SIMD registers for more throughput (512 bit)
- Fast (GDDR5) memory on card

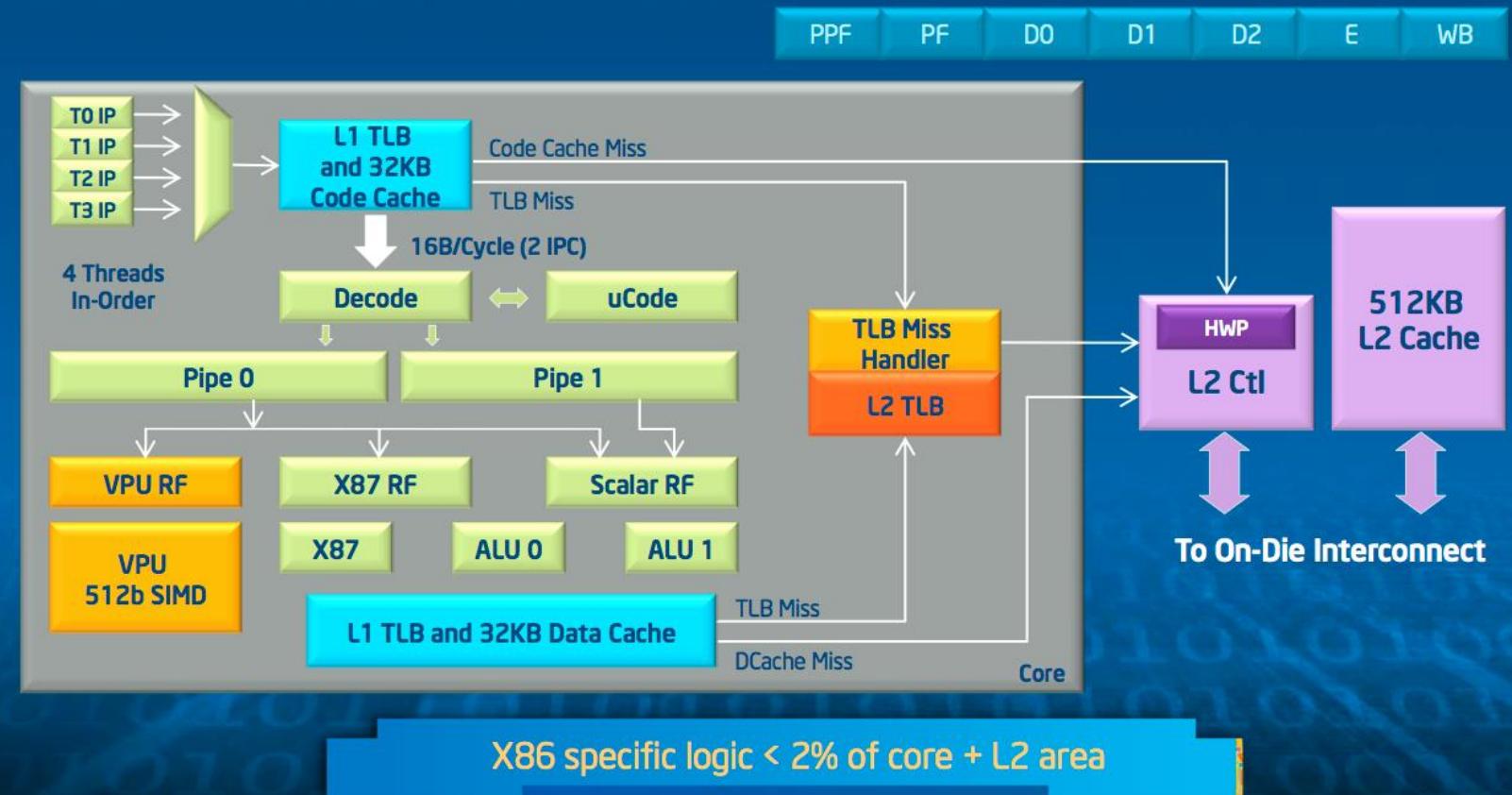


MIC Architecture

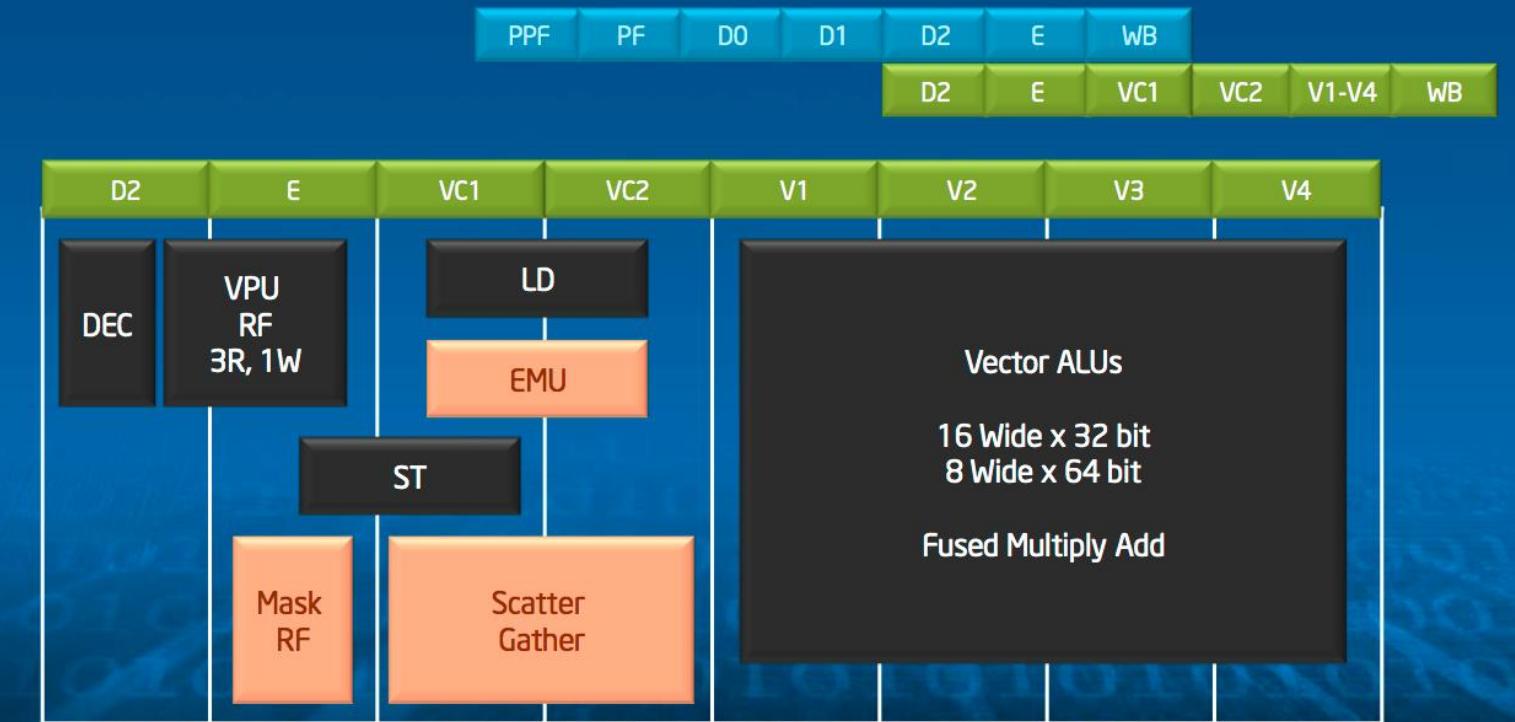
- Many cores on the die
- L1 and L2 cache
- Bidirectional ring network for **L2** Memory and PCIe connection



Knights Corner Core



Vector Processing Unit



Current Implementation: Xeon Phi on Stampede

- SE10P version
- Processor ~1.1 GHz
- 61 cores
- 512-bit wide vector unit
- 1.074 TF peak DP
- Data Cache: L1 32KB/core, L2 512KB/core
- Memory: 8GB GDDR5 DRAM, 5.5 GT/s, 512-bit*
- PCIe 5.0 GT/s, 16-bit

Top 10 Systems as of November 2013

#	Site	Manufacturer	Computer	CPU Interconnect [Accelerator]	Cores	Rmax (Tflops)	Rpeak (Tflops)	Power (MW)
1	National Super Computer Center in Guangzhou China	NUDT	Tianhe-2 (MilkyWay)	Intel Xeon E5-2692 2.2 GHz TH Express-2 Intel Xeon Phi 31S1P	3,120,000	33,862	54,902	17.8
2	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
3	DOE/NNSA/LLNL United States	IBM	Sequoia BlueGene/Q	Power BQC 1.6 GHz Custom	1,572,864	17,173	20,132	7.8
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	Fujitsu	K Computer	SPARC64 VIIIfx 2.0 GHz Tofu	705,024	10,510	11,280	12.6
5	DOE/SC/Argonne National Laboratory United States	IBM	Mira BlueGene/Q	Power BQC 1.6 GHz Custom	786,432	8,586	10,066	3.9
6	Swiss National Supercomputing Centre (CSCS) Switzerland	Cray	Piz Daint Cray XC30	Xeon E5-2670 2.6 GHz Aries NVIDIA K20x	115,984	6,271	7,788	2.3
7	Texas Advanced Computing Center/Univ. of Texas United States	Dell	Stampede PowerEdge C8220	Xeon E5-2680 2.7 GHz Infiniband FDR Intel Xeon Phi SE10P	462,462	5,168	8,520	4.5
8	Forschungszentrum Juelich (FZJ) Germany	IBM	JUQUEEN BlueGene/Q	Power BQC 1.6 GHz Custom	458,752	5,008	5,872	2.3
9	DOE/NNSA/LLNL United States	IBM	Vulcan BlueGene/Q	Power BQC 1.6 GHz Custom	393,216	4,293	5,033	1.9
10	Leibniz Rechenzentrum Germany	IBM	SuperMUC iDataPlex DX360M4	Xeon E5-2680 2.7 GHz Infiniband FDR	147,456	2,897	3,185	3.4

Design Choices of the Leaders

Multicore

Maintain complex cores, and replicate
(x86, SPARC, Power7) [#4 and 10]

Manycore
/Embedded

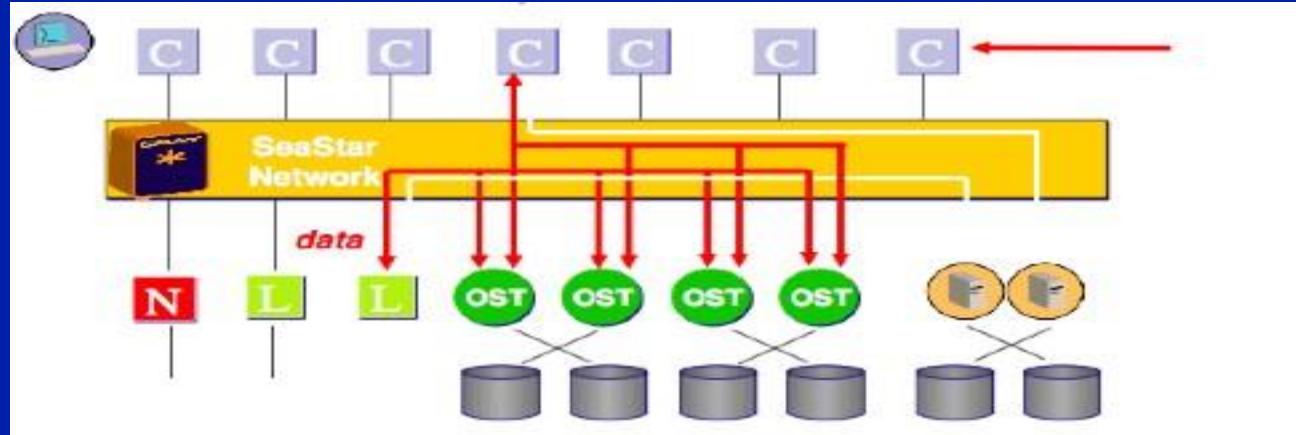
Use many simpler, low power cores from
embedded (BlueGene) [#3, 5, 8 and 9]

GPU /
Accelerator

Use highly specialized processors from
graphics market space (NVidia Fermi,
Cell, Intel Phi (MIC)) [# 1, 2, 6 and 7]

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

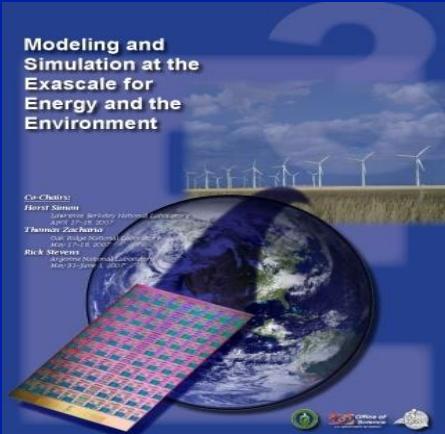
You will get there by going very parallel

What does parallel computing look like?

Where is this going?

Towards Exascale!

- Pflops computing fully established with more than 20 machines
- Exascale has been discussed in numerous workshops, conferences, planning meetings for about six years
- Exascale projects have been started in the U.S. and many other countries and regions
- Key challenges to exascale remain



The image displays three posters for "Scientific Grand Challenges" workshops, each featuring a circular inset showing a scientific visualization.

Top right poster:
Scientific Grand Challenges
FOREFRONT QUESTIONS IN NUCLEAR SCIENCE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE
January 26-28, 2009 • Washington, D.C.
U.S. DEPARTMENT OF ENERGY

Middle left poster:
Scientific Grand Challenges
CHALLENGES FOR UNDERSTANDING THE QUANTUM UNIVERSE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE
December 9-11, 2008 • Menlo Park, CA
U.S. DEPARTMENT OF ENERGY

Middle right poster:
Scientific Grand Challenges
CHALLENGES IN CLIMATE CHANGE SCIENCE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE
November 6-7, 2008 • Washington D.C.
U.S. DEPARTMENT OF ENERGY

Bottom right logo:
U.S. DEPARTMENT OF ENERGY

Courtesy Horst Simon, LBNL

Exascale?

exa = 10^{18} = 1,000,000,000,000,000,000 = quintillion

23,800 X



Cray Red Storm
2004
42 Tflops

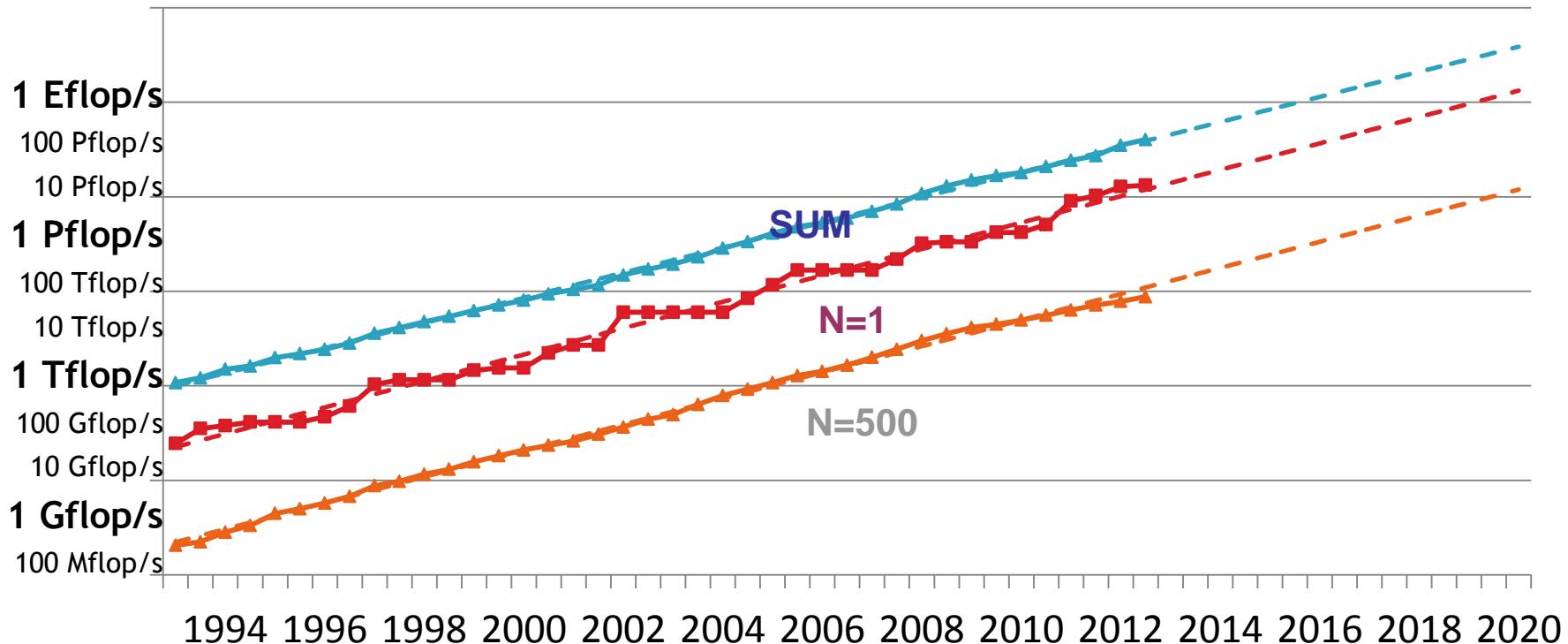
833,000 X



or

NVIDIA K40
1.2 Tflops

Projected Performance Development



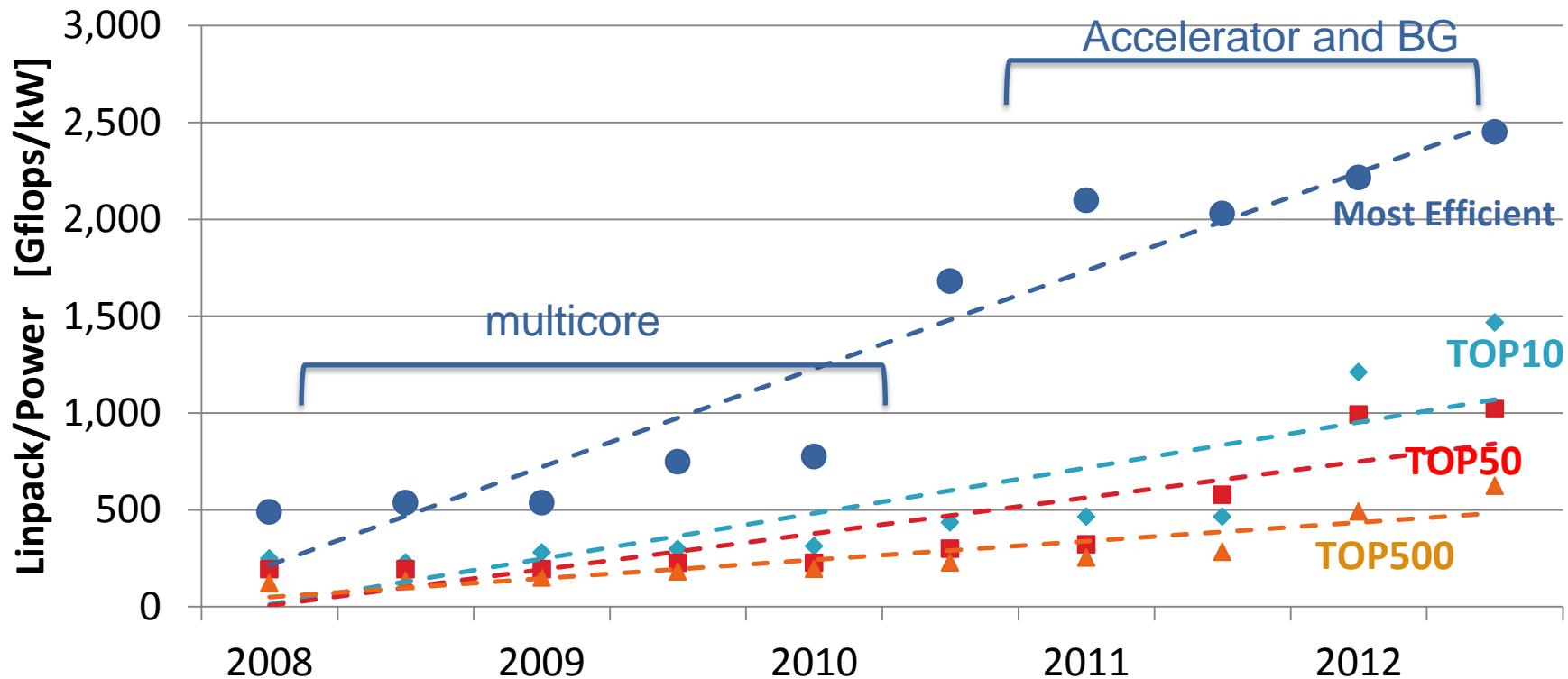
Courtesy Horst Simon, LBNL
Data from: TOP500 November 2012

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.

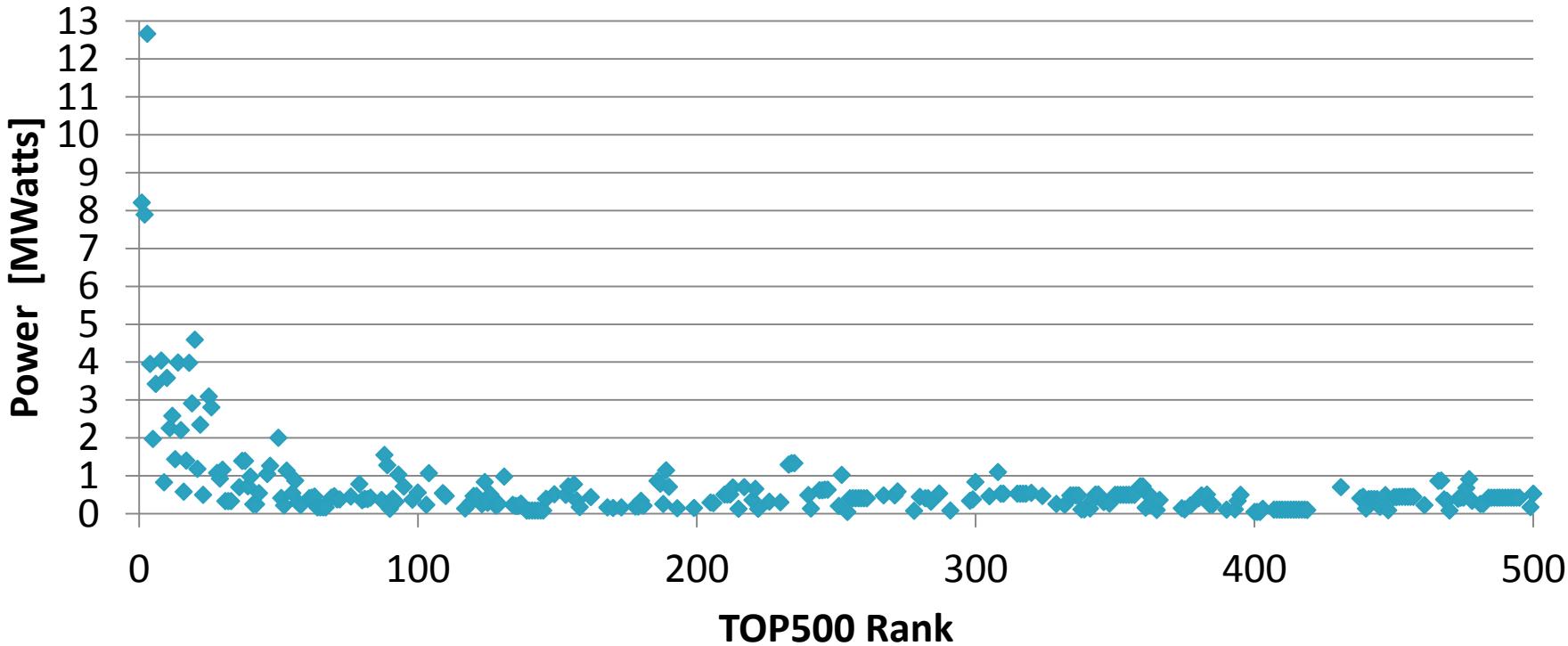
Power Efficiency over Time



Data from: TOP500 November 2012

Courtesy Horst Simon, LBNL

Total Power Levels (kW) for TOP500 Systems



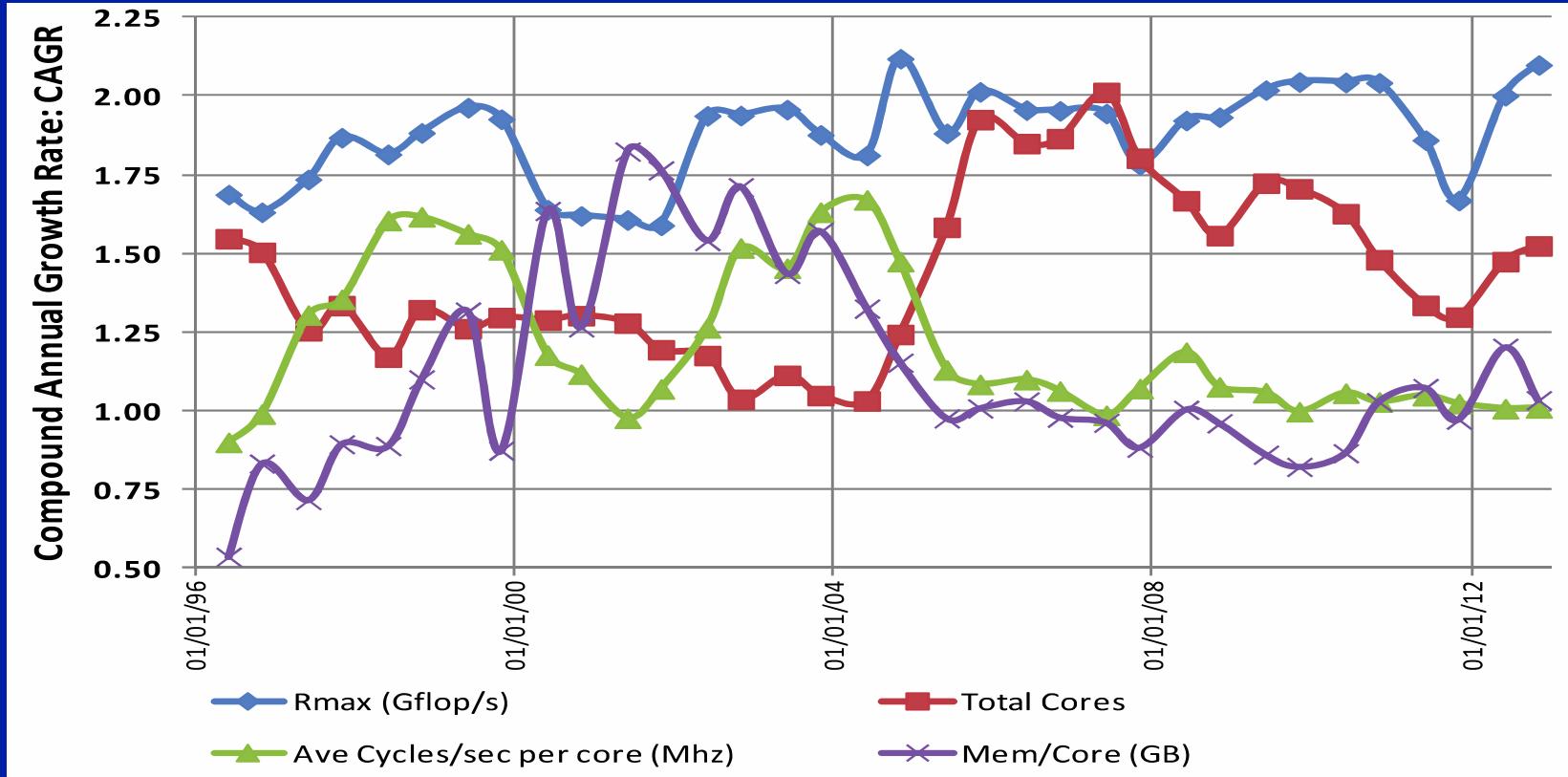
Courtesy Horst Simon, LBNL
Source: TOP500 November 2012

Most Power Efficient Architectures

Computer	Rmax/ Power
Appro GreenBlade, Xeon 8C 2.6GHz, Infiniband FDR, Intel Xeon Phi	2,450
Cray XK7 , Opteron 16C 2.1GHz, Gemini, NVIDIA Kepler	2,243
BlueGene/Q , Power BQC 16C 1.60 GHz, Custom	2,102
iDataPlex DX360M4, Xeon 8C 2.6GHz, Infiniband QDR, Intel Xeon Phi	1,935
RSC Tornado, Xeon 8C 2.9GHz, Infiniband FDR, Intel Xeon Phi	1,687
SGI Rackable, Xeon 8C 2.6GHz, Infiniband FDR, Intel Xeon Phi	1,613
Chundoong Cluster, Xeon 8C 2GHz, Infiniband QDR, AMD Radeon HD	1,467
Bullx B505, Xeon 6C 2.53GHz, Infiniband QDR, NVIDIA 2090	1,266
Intel Cluster, Xeon 8C 2.6GHz, Infiniband FDR, Intel Xeon Phi	1,265
Xtreme-X , Xeon 8C 2.6GHz, Infiniband QDR, NVIDIA 2090	1,050

$$[\text{Tflops/MW}] = [\text{Mflops/Watt}]$$

Trends with ends.



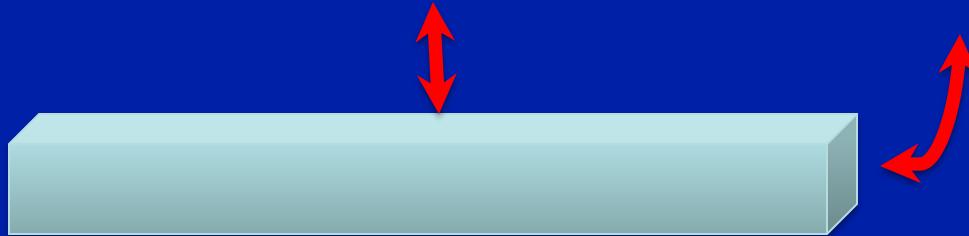
Source: Kogge and Shalf, IEEE CISE

Courtesy Horst Simon, LBNL

The Problem with Wires:

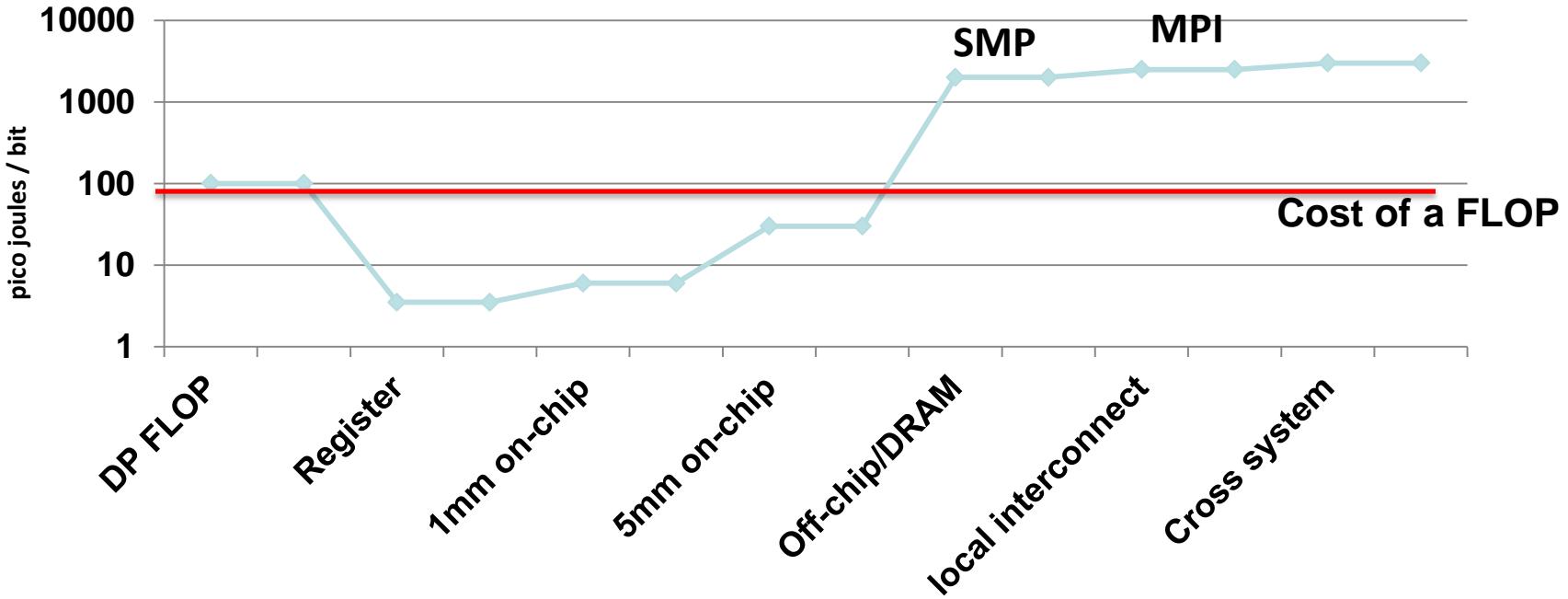
Energy to move data proportional to distance

- Cost to move a bit on copper wire:
 - $\text{power} = \text{bitrate} * \text{Length} / \text{cross-section area}$



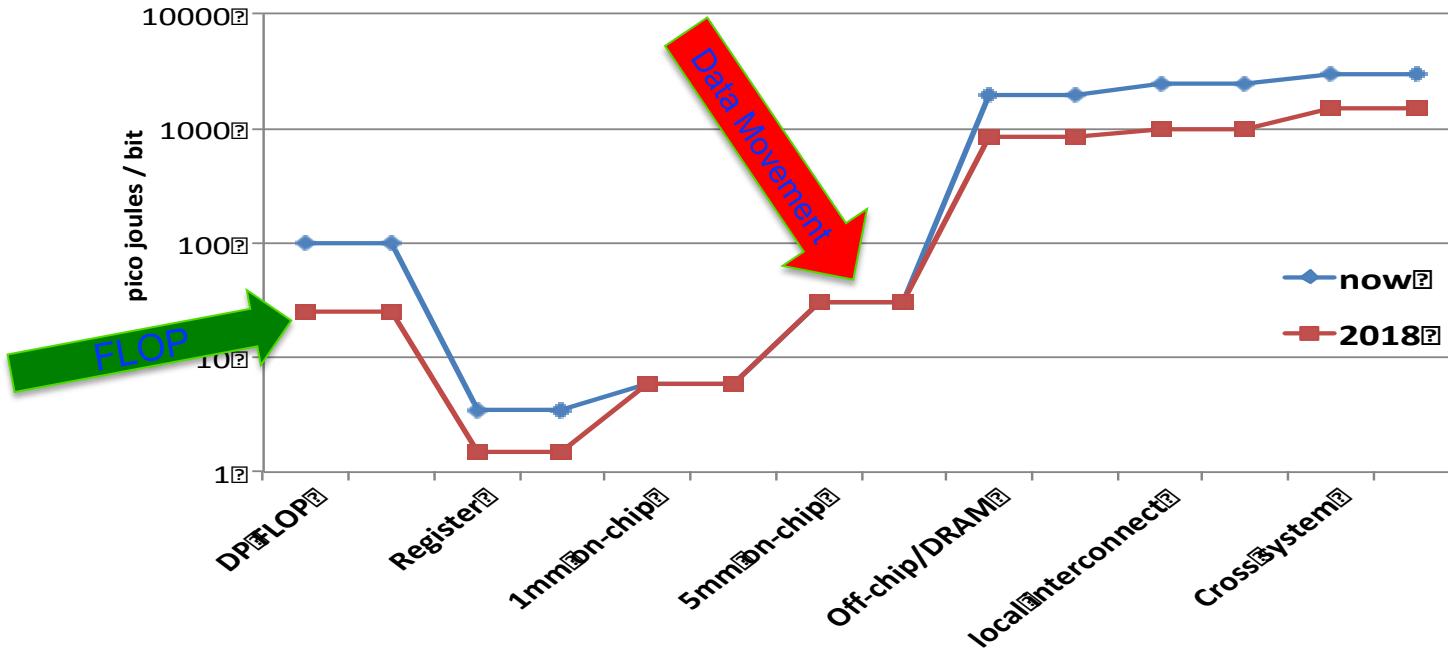
- Wire data capacity constant as feature size shrinks
- Cost to move bit proportional to distance
- ~1TByte/sec max feasible off-chip BW (10GHz/pin)
- Photonics reduces distance-dependence of bandwidth

The Cost of Data Movement



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

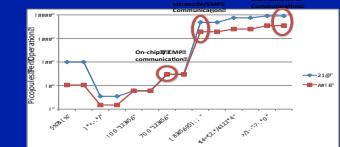
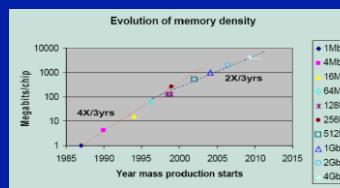
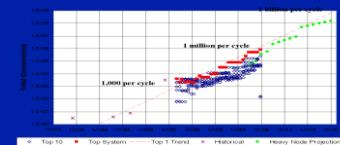
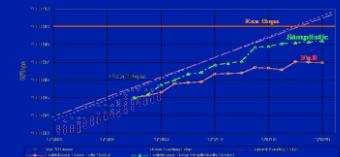
Current programming systems have WRONG optimization targets

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

Necessity is the mother... (of hardware revolutions)

- Quantum tunneling becomes the limitation at 5-7nm (~30 atoms!)
- The International Technology Roadmap for Semiconductors (ITRS) roadmap has that at 2024
- Options?

Clock increase with new materials, gates

Better fabrication control (margins and redundancy)

New configurations

3D packages

3D transistors

Non-volatile memory (PCM, not just flash)

Optical interconnects

New materials

III-IV, germanium, nanowires/tubes/graphene

Better lithography

Shorter
Interconnect

Less
Power

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 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 1 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.

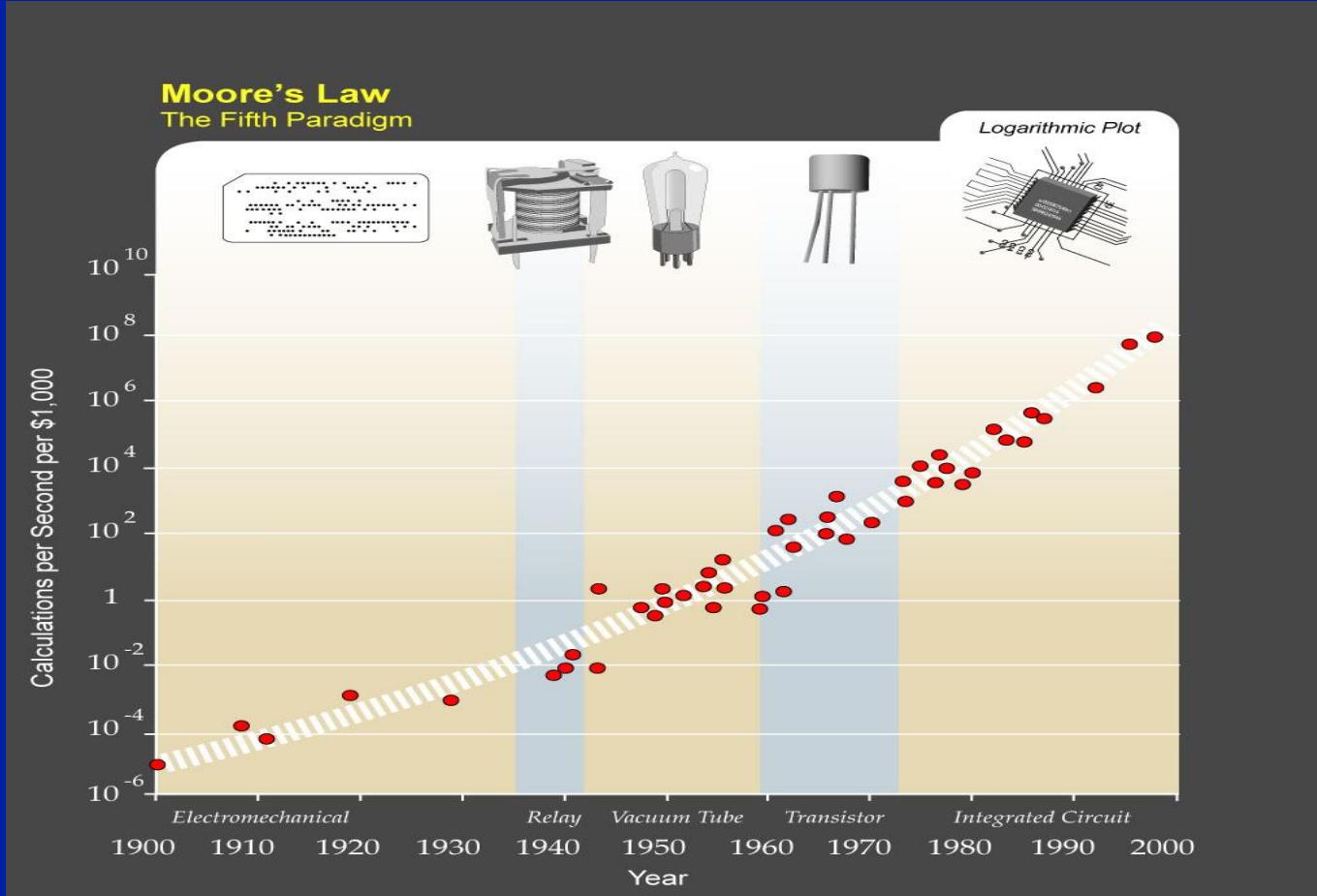
No more garbage collecting, script languages. I am looking at you, Java, Python, Matlab,

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 2020 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



It would only be the 6th paradigm.



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!