

CSE 160

Lecture 26

Supercomputing

Announcements

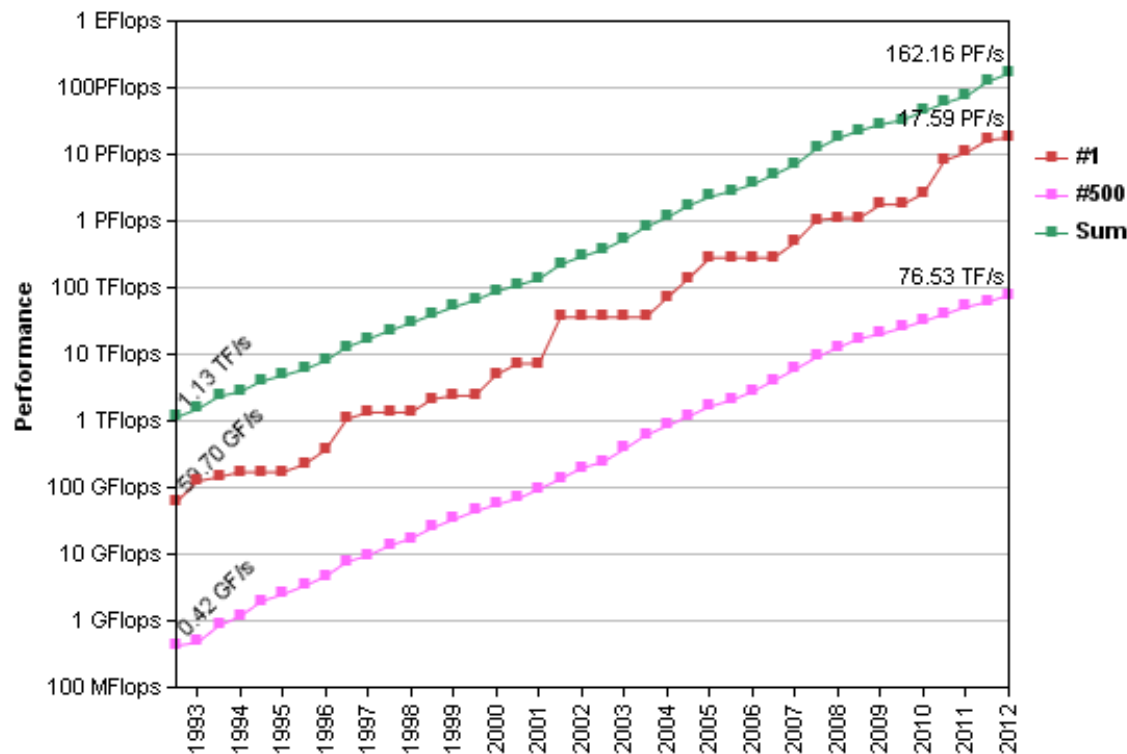
- On Weds 3/13 we have a 2 hour session
 - Usual class time at the Rady school – final exam review
 - SDSC tour @ 2.15pm (replaces class on 3/15)
 - See the web page for details
 - No class on Friday
- Section: Thursday @ 5pm csb 004
 - Bring question for me to answer, problems to work out

Today's lecture

- Technological Trends
- Application Requirements
- Architectural Directions

What is the worlds fastest supercomputer?

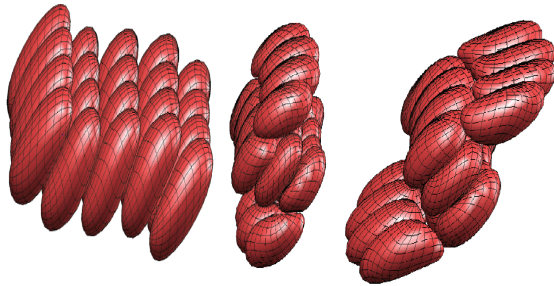
- Top500 #1, Titan – Cray XK7 @ Oak Ridge Nat'l Lab: 17.6 Tflops/sec, 8.2 MW power



top500.org

State-of-the-art capabilities

Blood Simulation on Jaguar Gatech team



p	48	384	3072	24576
Time (sec)	899.8	116.7	16.7	4.9
Efficiency	1.00	0.96	0.84	0.35

Strong scaling

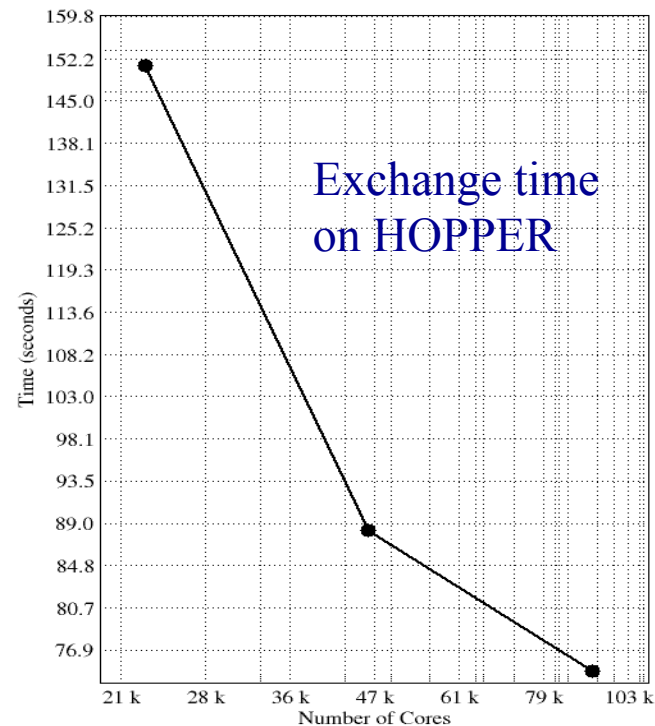
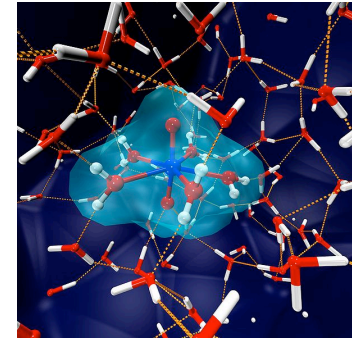
p	24576	98304	196608
Time (sec)	228.3	258	304.9
Efficiency	1.00	0.88	0.75

Weak scaling

Tan Nguyen, UCSD

Ab Initio Molecular Dynamics (AIMD) using Plane Waves Density Functional Theory

Eric Bylaska (PNNL)



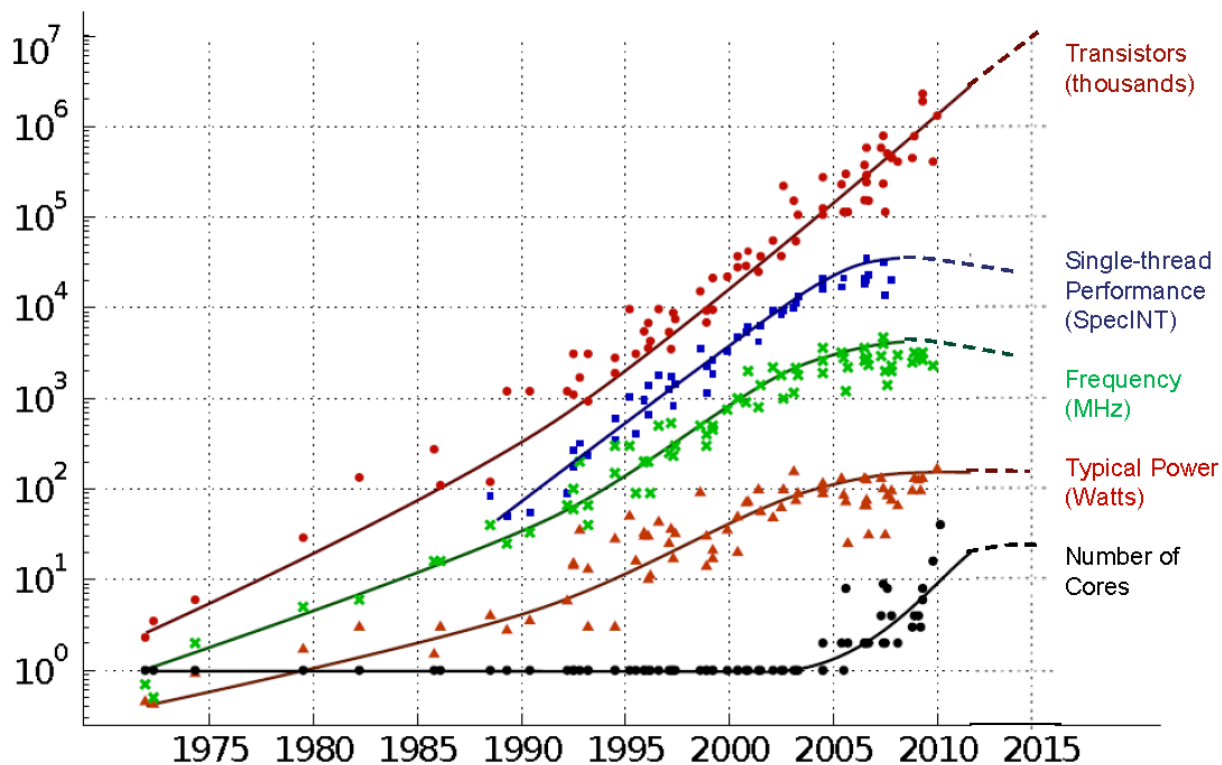
Up and beyond to Exascale

- In 1961, President Kennedy mandated a landing on the Moon by the end of the decade
- July 20, 1969 at tranquility base
“The Eagle has landed”
- The US Govt set an ambitious schedule to reach 10^{18} flops by 2018, x1000 performance increase
- DOE will take the lead in the US, EU also engaged
- Massive technical challenges

The Challenges to landing “Eagle”

- High levels of parallelism within and across nodes
10⁶ devices. 10⁹⁺ threads
- Power consumption: >20 MW @ Exascale, but nearly at 10MW today, with “just” 0.02 ExaFlops
Data storage & access consumes most of the energy
- Processor technology hasn’t settled
 - Mixture of accelerators and conventional cores
 - NUMA processors, not fully cache coherent on-chip
- Data motion costs continue to grow
 - Complicated memory hierarchies
 - Hide latency, conserve locality
- Software, software, software
 - Code become more complicated, in order to effectively utilize the hardware’s capabilities

35 years of processor trends



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

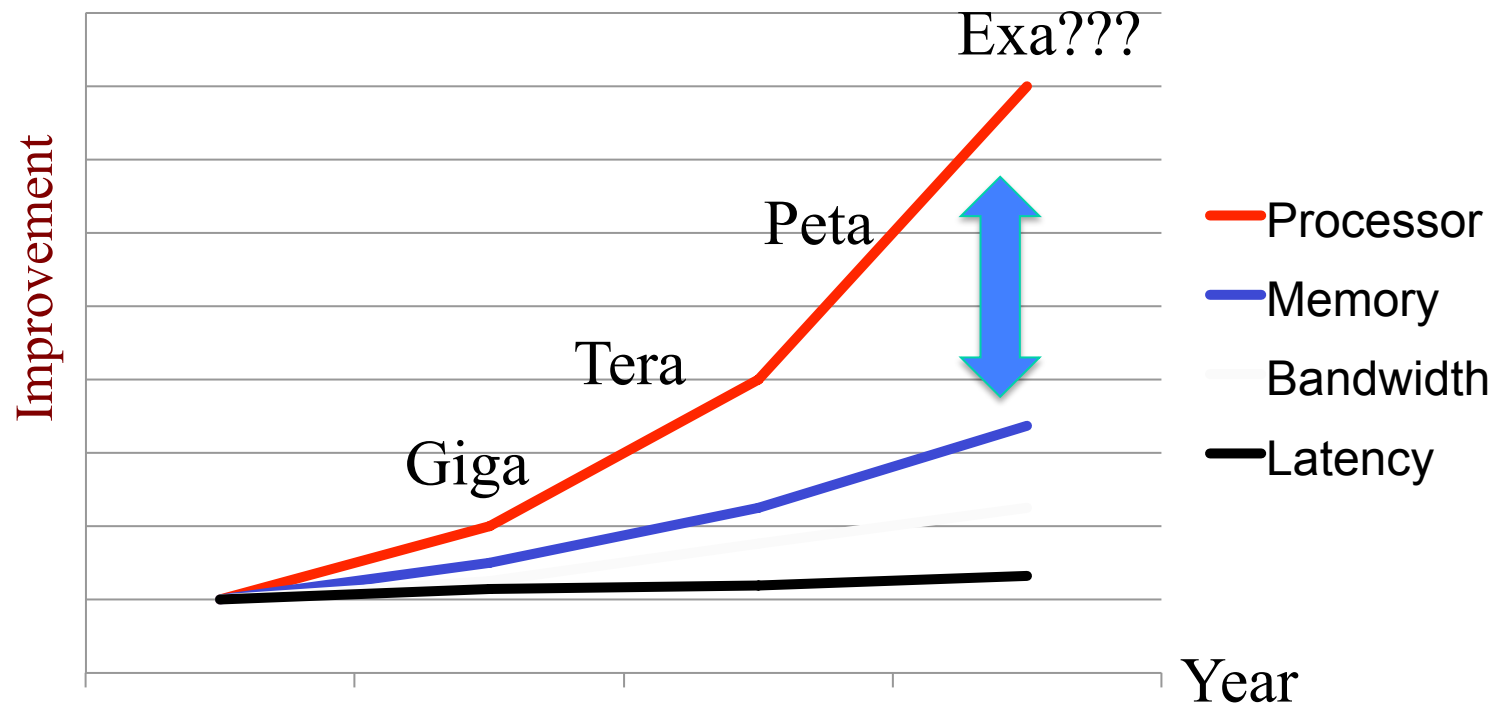
3

Data Processing in Exascale-class Computing Systems | April 27, 2011 | CRM



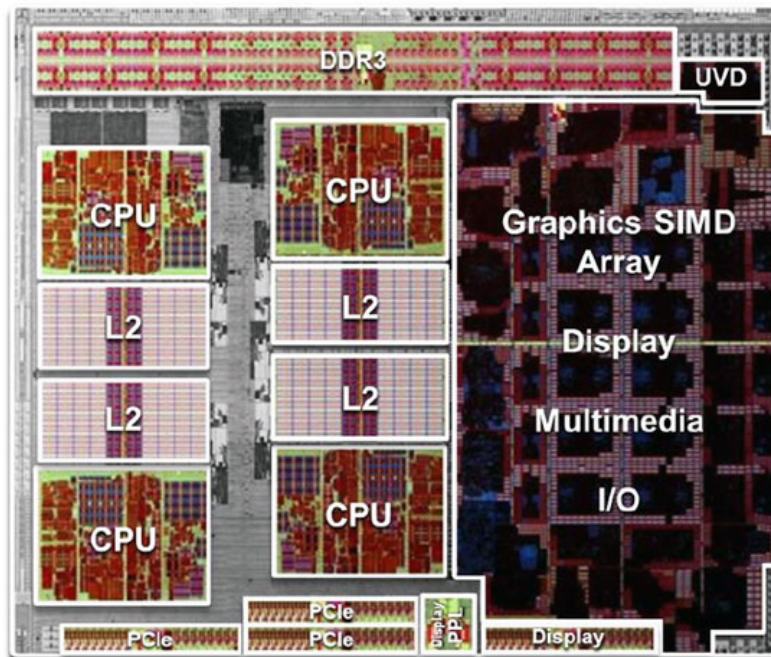
Trend: data motion costs are rising

- Increase amount of computation performed per unit of communication
 - Conserve locality, tolerate or avoid communication
- Many threads



Hybrid processing

- Two types of processors: general purpose + accelerator
 - AMD fusion: 4 x86 cores + hundreds of Radeon GPU cores
- Accelerator can perform certain tasks more quickly than the conventional cores
- Accelerator amplifies relative cost of communication



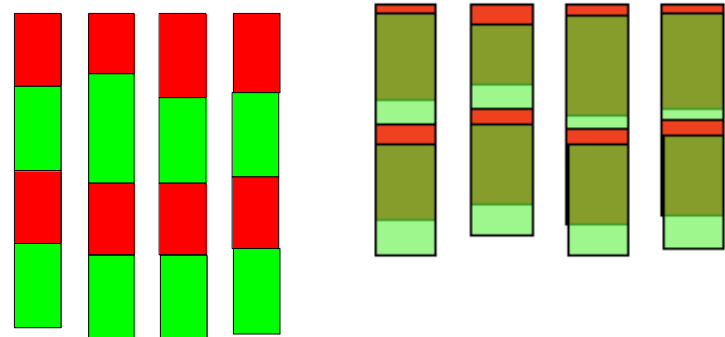
Hothardware.com

Today's lecture

- Technological Trends
- **Application Requirements**
- Architectural Directions

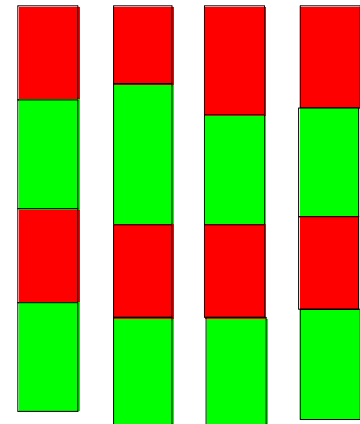
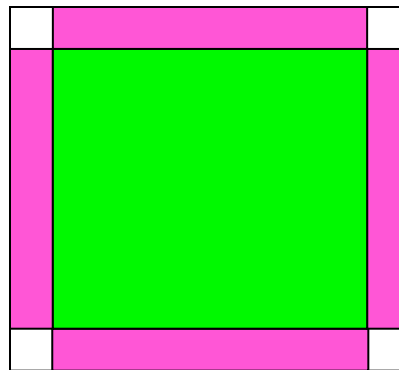
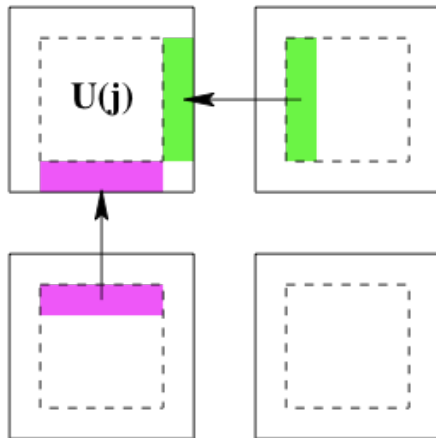
A Crosscutting issue: Hiding latency

- Little's law [1961]
 - The number of threads must equal the parallelism times the latency
 $T = p \times \lambda$
 - p and λ are increasing with time
- Difficult to implement
 - Split phase algorithms
 - Partitioning and scheduling
- The state-of-the-art enables but doesn't support the activity
- Distracts from the focus on the domain science
- Implementation policies entangled with correctness issues
 - Non-robust performance
 - High development costs



A Classic message passing implementation of a stencil method

- Decompose domain into sub-regions, one per process
 - Transmit **halo regions** between processes
 - Compute **inner region** after communication completes
- Loop carried dependences impose a strict ordering on communication and computation

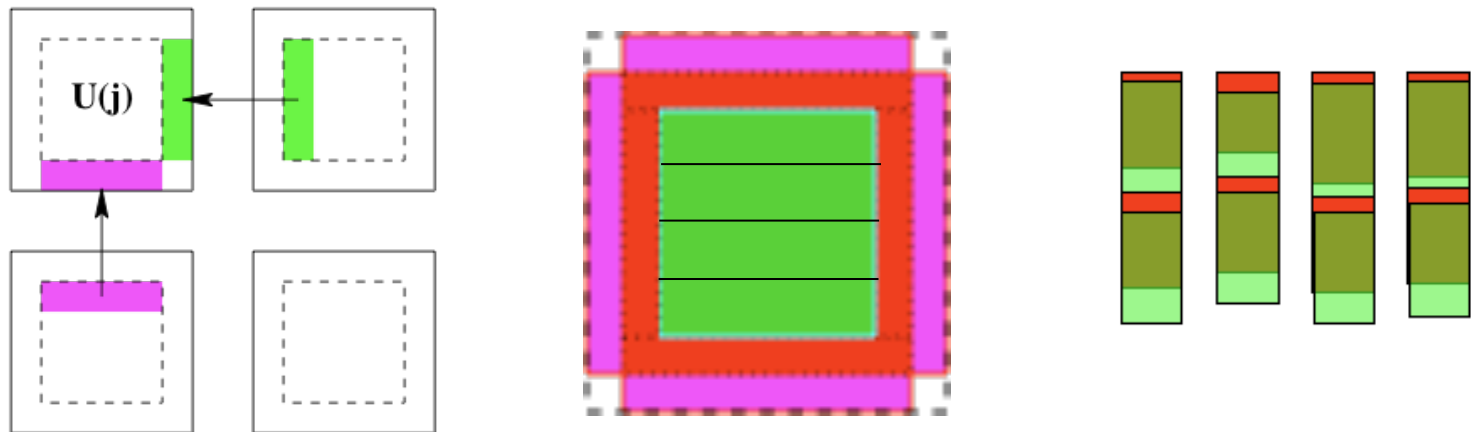


Communication tolerant variant

- Only a subset of the domain exhibits **loop carried dependences** with respect to the **halo region**
- Subdivide the domain to remove some of the dependences
 - We can sweep the **inner region** in parallel with **communication**
 - Sweep the **annulus** after communication finishes
- MPI may not be able to realize overlap with MPI_IRecv and MPI_Isend
- We can use multithreading to handle the overlap
- 1+ cores (proxy threads) handle communication

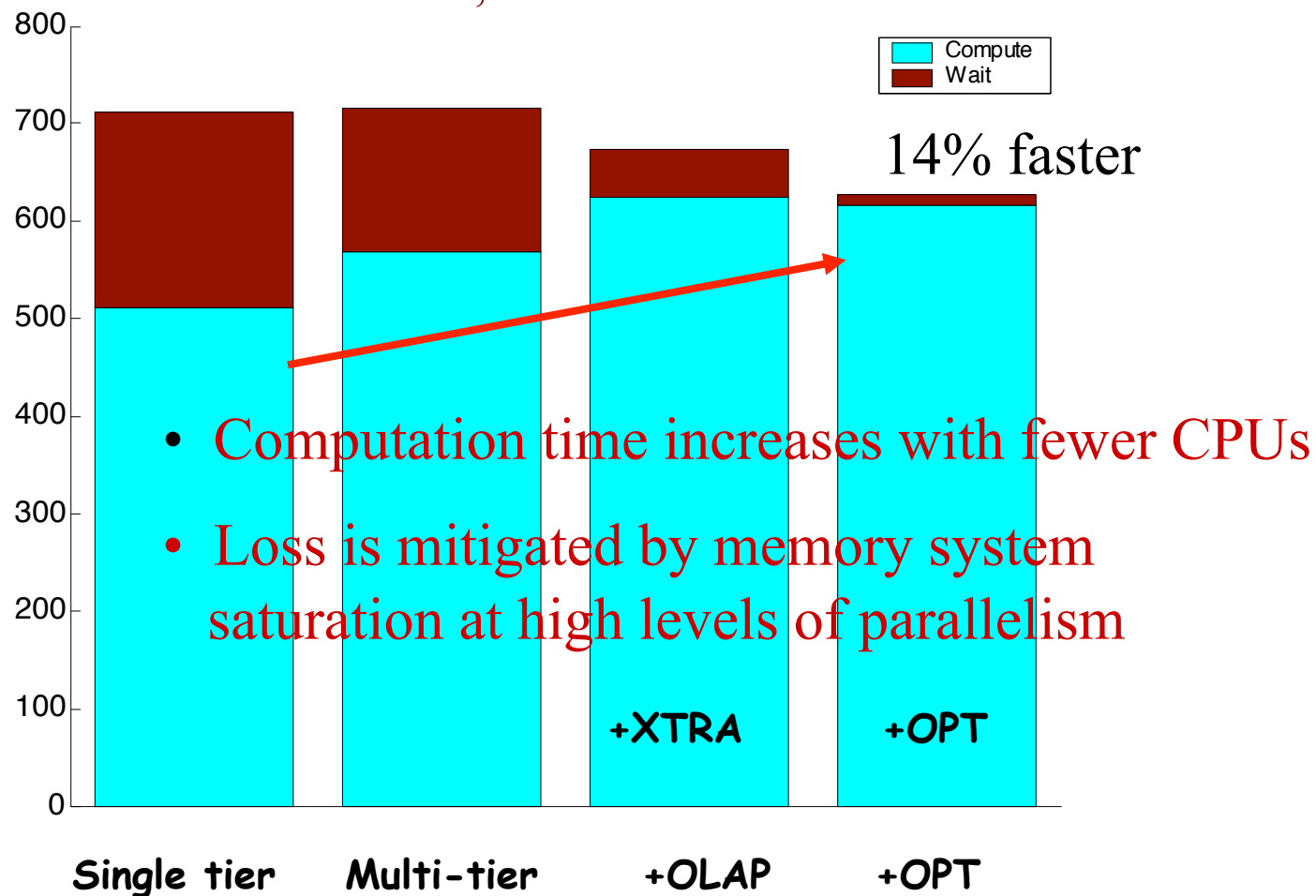
S. Fink, PhD thesis, UCSD, 1998

Baden & Fink, “Communication overlap in multi-tier parallel algorithms,” SC98



Performance improves with overlap

Shalit and Baden, ICCS 2001



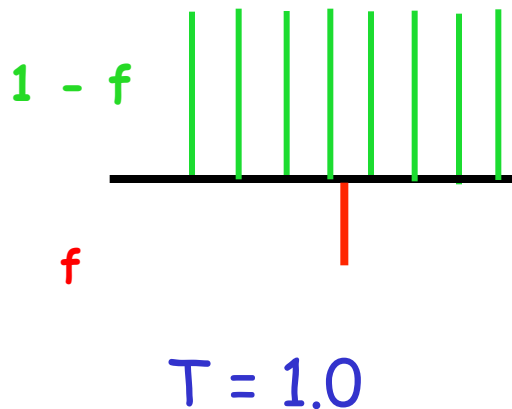
A performance model of overlap

- Assumptions

p = number of processors per node

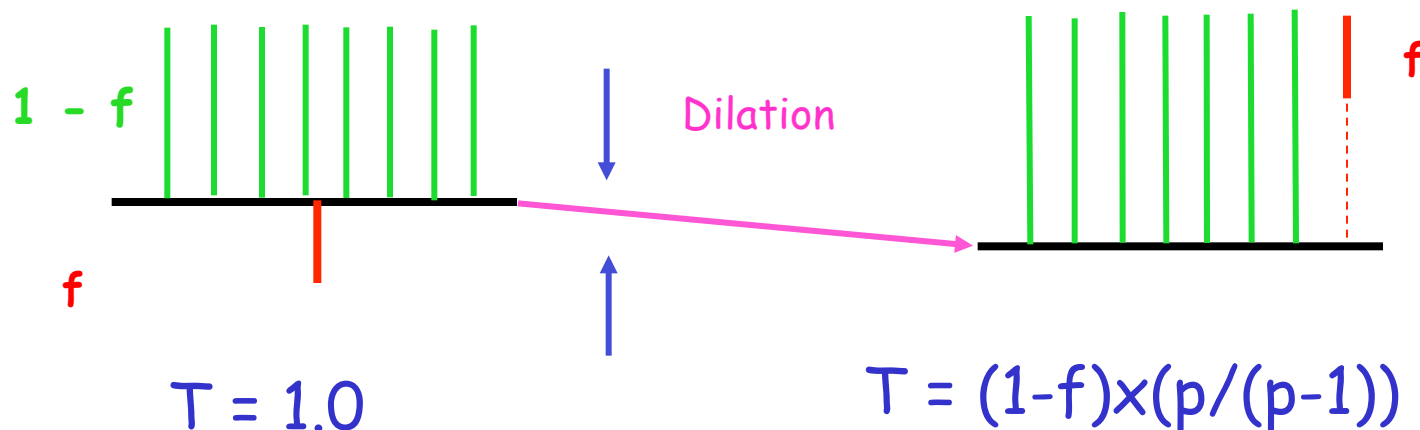
running time = 1.0

$f < 1$ = communication time
(i.e. not overlapped)



Performance

- When we displace computation to make way for the proxy, computation time *increases*
- Wait on communication drops to zero, ideally
- When $f < p/(2p-1)$: improvement is $(1-f) \times (p/(p-1))^{-1}$
- Communication bound: improvement is $1/(1-f)$

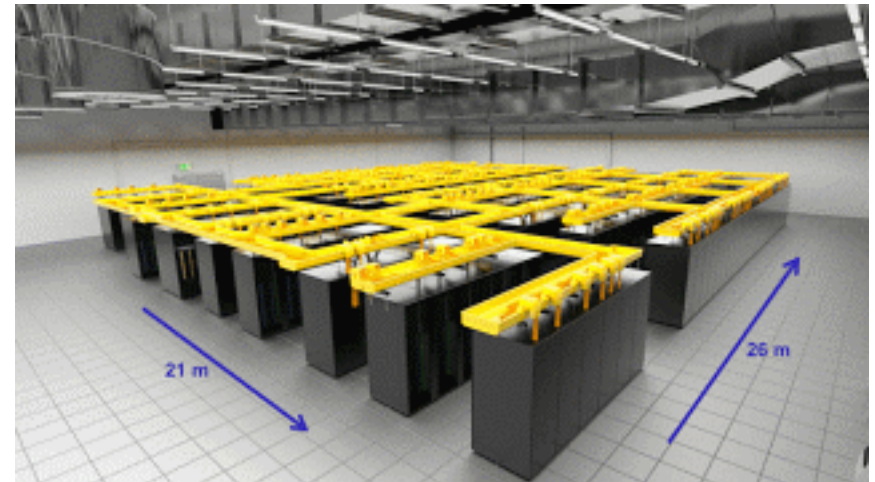


Today's lecture

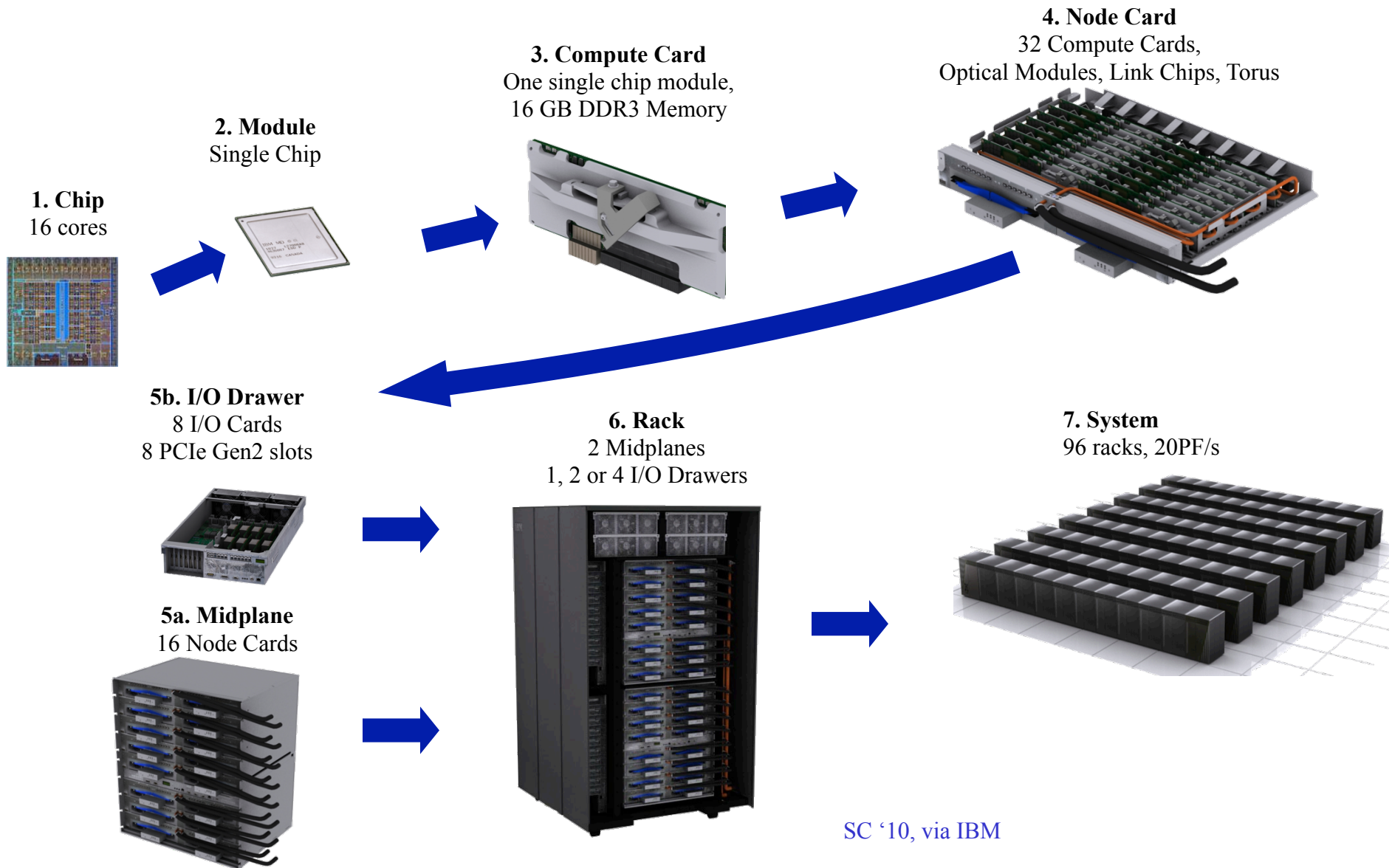
- Technological Trends
- Application Requirements
- **Architectural Directions**

Blue Gene

- IBM-US Dept of Energy collaboration
- Low power, high performance interconnect
- 3rd Generation: Blue Gene/Q
- Sequoia at LLNL, #2 on top 500
 - ▶ 16.3 Tflops, 7.89 MW
 - ▶ 1.6M cores/93,304 compute nodes, 16 GB mem/node
 - ▶ 20 Petaflops = 20,000 TeraFlops = 20M GFlops
 - ▶ 96 racks, 3,000 square feet
 - ▶ ×7 more power efficient than previous generation
 - ▶ 64-bit PowerPC (A2 core)
 - ▶ Weighs about the same as 30 elephants

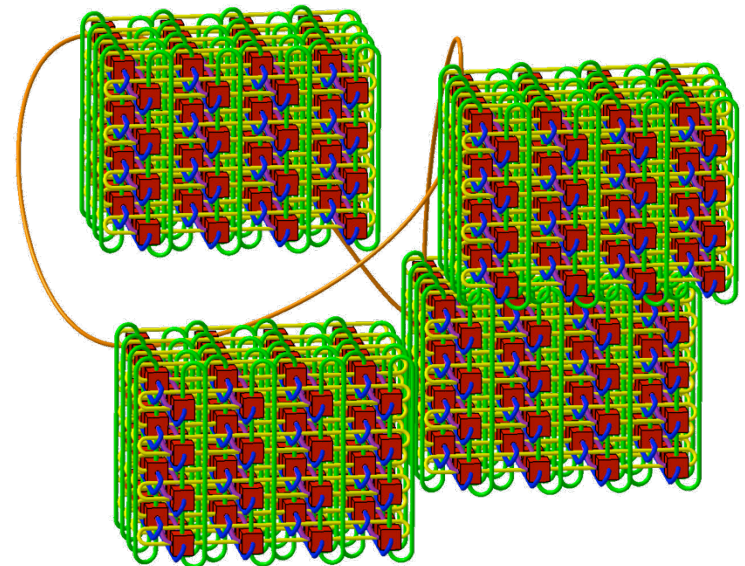


Hierarchical Packaging

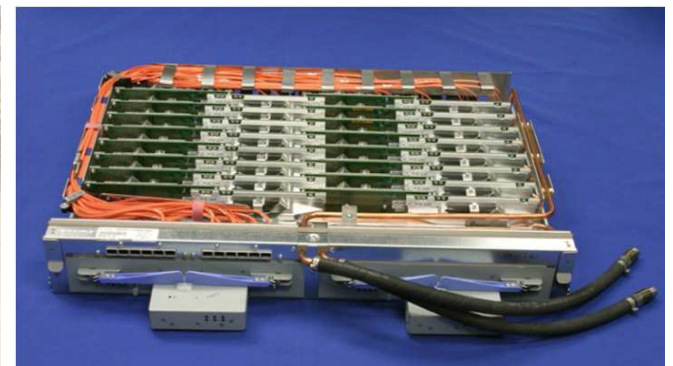
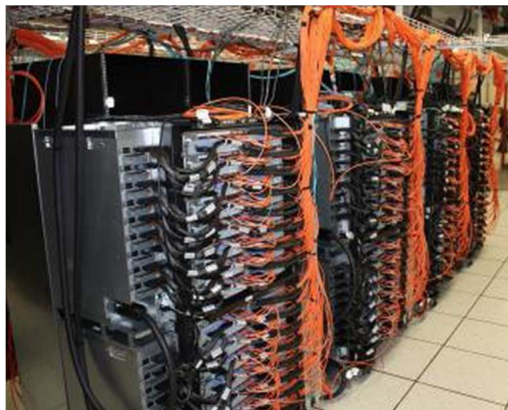
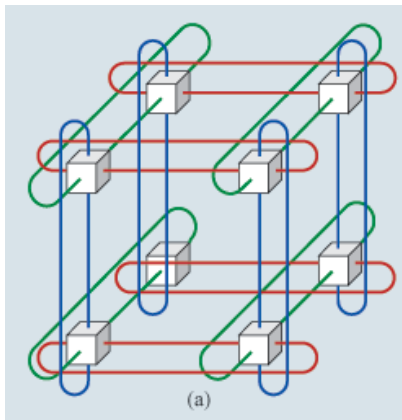


Blue Gene/Q Interconnect

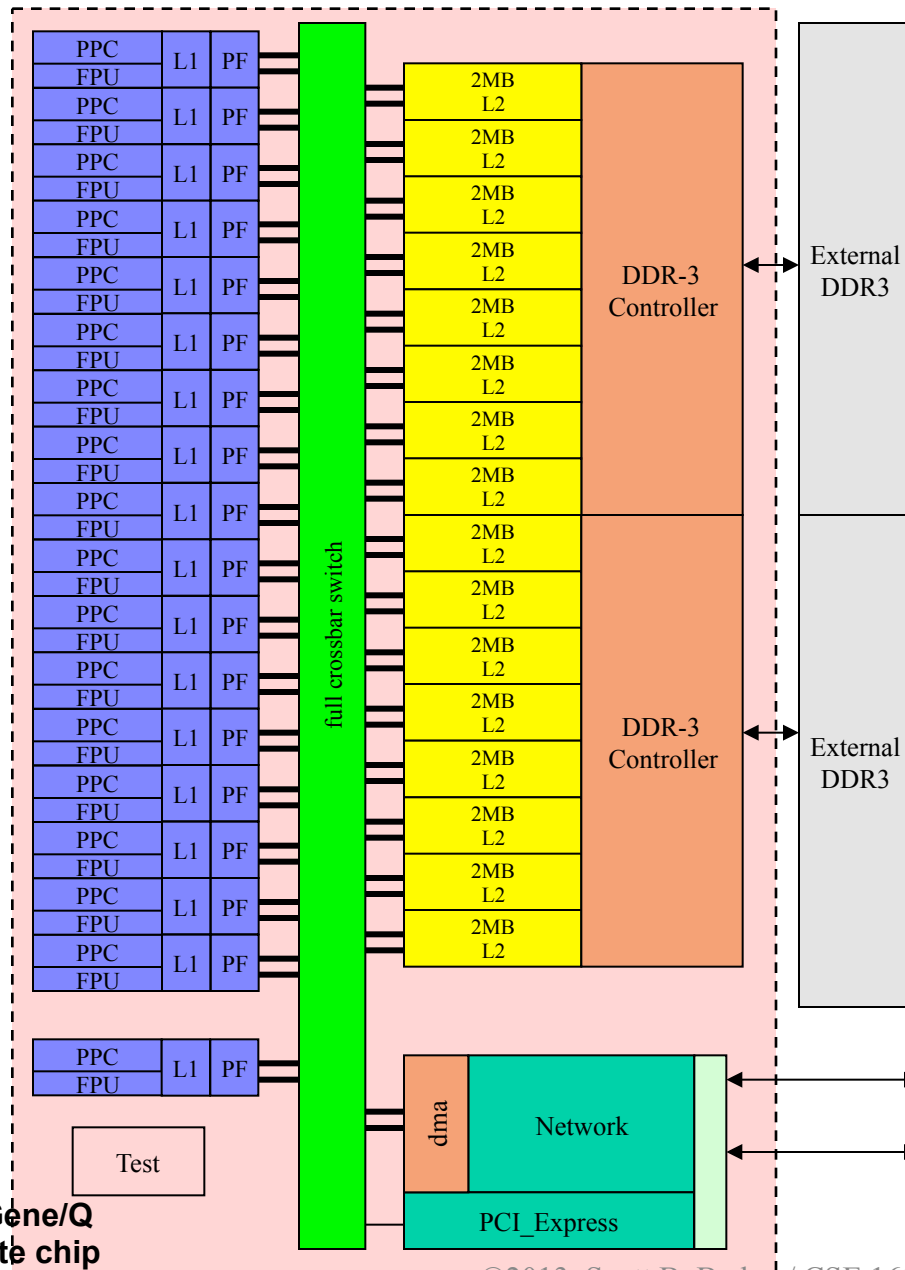
- 5D toroidal mesh (end around)
 - Can scale to > 2M cores
 - 2 GB/sec bidirectional bandwidth (raw)
on all 10 links, 1.8 GB available to the user
 - 5D nearest neighbor exchange
~1.8GB/s/link, 98% efficiency
 - Hardware latency ranges from 80 ns to 3μs
- Collective network
 - Global Barrier, Allreduce, Prefix Sum
 - Floating point reductions at ~95% of peak



IBM



Processor Block diagram

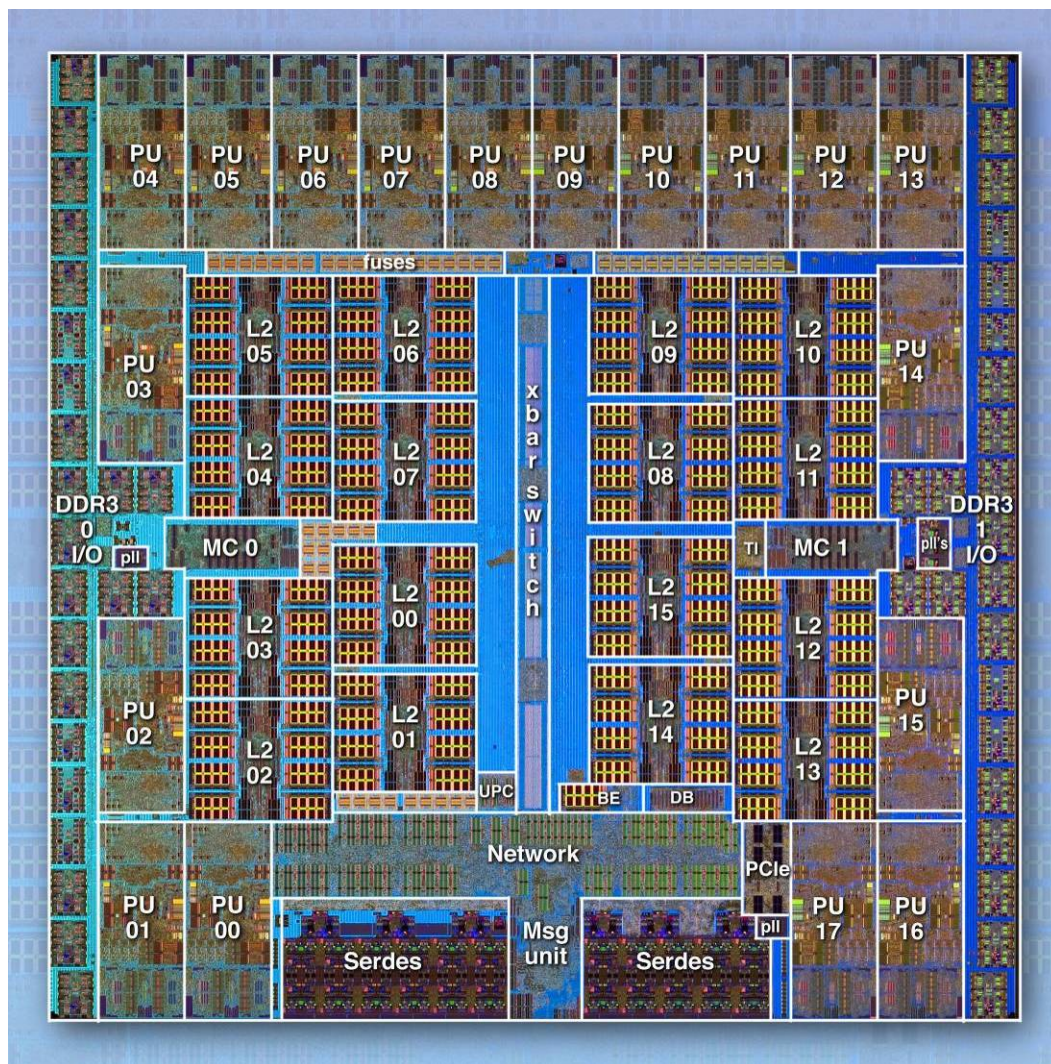


- 16 (user) + 1 (OS) cores
- 18th redundant core
- Each core 4-way multithreaded
- 1.6 GHz
- 16 GB memory per node
- Shared L2\$: 32MB
- Network routing integrated on-chip

IBM

Blue Gene/Q
compute chip

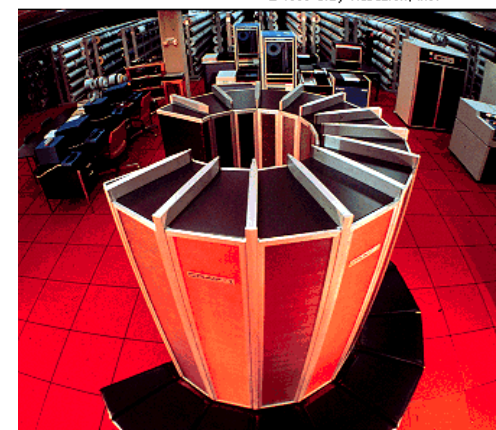
Die photograph



IBM

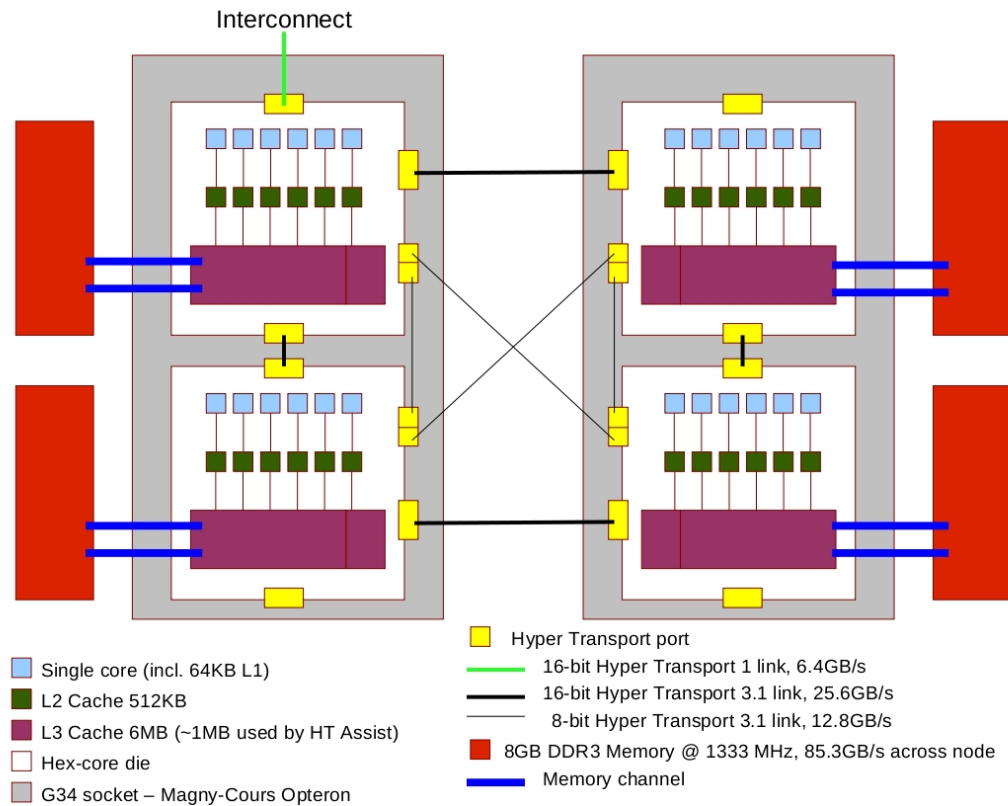
- 16 (user) + 1 (OS) cores
- 18th redundant core
- Each core 4-way multithreaded
- 1.6 GHz
- Shared L2\$: 32MB
- Network routing integrated on-chip
- Compare with layout of the Cray 1 (1976)

From Computer Desktop Encyclopedia
Reproduced with permission.
© 1996 Cray Research, Inc.



The future?

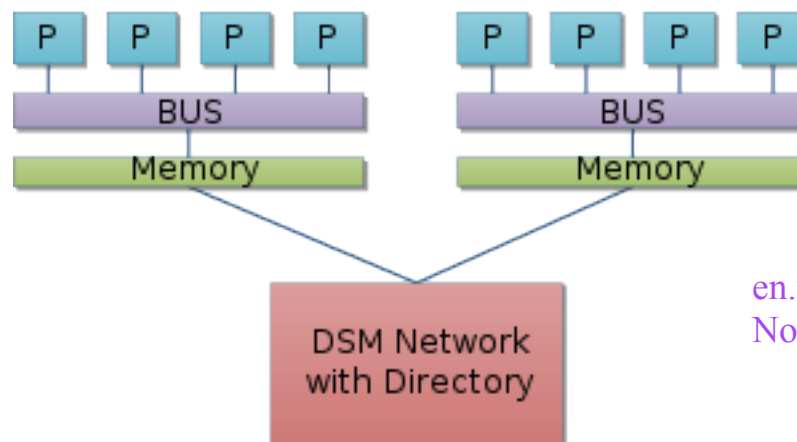
- Lower power is inevitable
- NUMA likely



<http://www.hector.ac.uk/cse/documentation/Phase2b/#arch>

NUMA Architectures

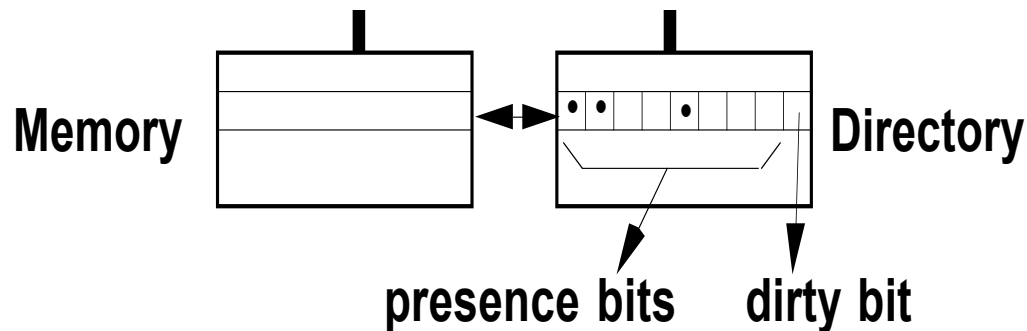
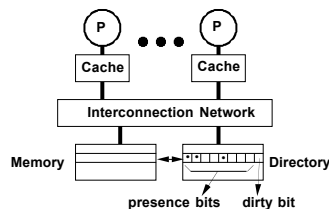
- The address space is global to all processors, but memory is physically distributed
- Point-to-point messages manage coherence
- A **directory** keeps track of sharers, one for each block of memory
- Stanford Dash; NUMA nodes of the Cray XE-6, SGI UV, Altix, Origin 2000



[en.wikipedia.org/wiki/
Non-Uniform_Memory_Access](http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access)

Inside a directory

- Each memory block has an associated **home**: the core physically holding the associated portion of the global address space
- Each block also has an **owner**: the processor whose memory contains the actual value of the data
- Initially home = owner, but this can change ...
- Each processor has a 1-bit “sharer” entry in the directory, also a dirty bit and a PID identifying the owner in the case of a dirty block



*Parallel
Computer
Architecture,
Culler, Singh,
& Gupta*

Software

- We need to think hard about hiding an explosion of details
- Parallel programming languages
- Domain specific languages
- Embedded domain specific languages
- Autotuning