How to Run Applications Fasters?

- There are 3 ways to improve performance
  1) Work harder
  2) Work smarter
  3) Get help

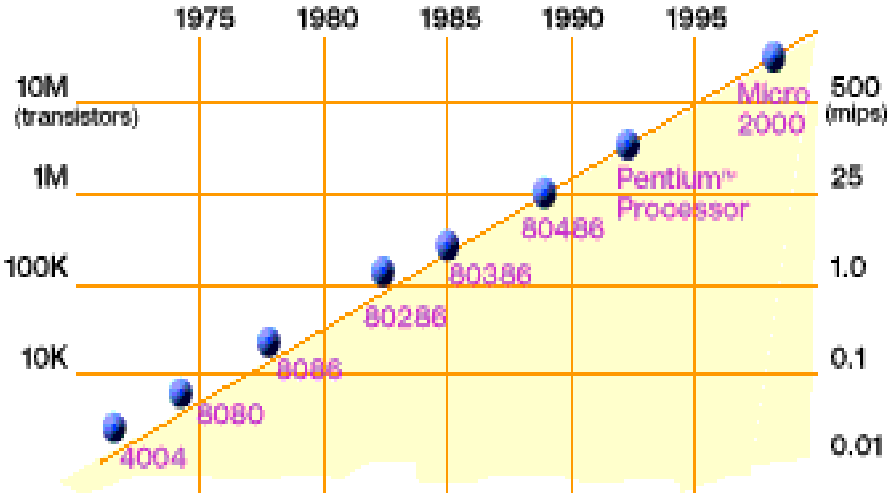
- Computer analogy
  1) Use faster hardware : e.g., reduce the time per instruction (clock cycle)
  2) Optimize algorithms and techniques, customized hardware
  3) Multiple computers to solve the problem = to increase no. of instructions executed per clock cycle

Sequential Architecture Limitations

- Sequential architectures reaching physical limitation
- Hardware improvements like pipelining, superscalar, …., are not scalable and require sophisticated compiler technology
- Vector processing works well for certain problems

=> Technology push!

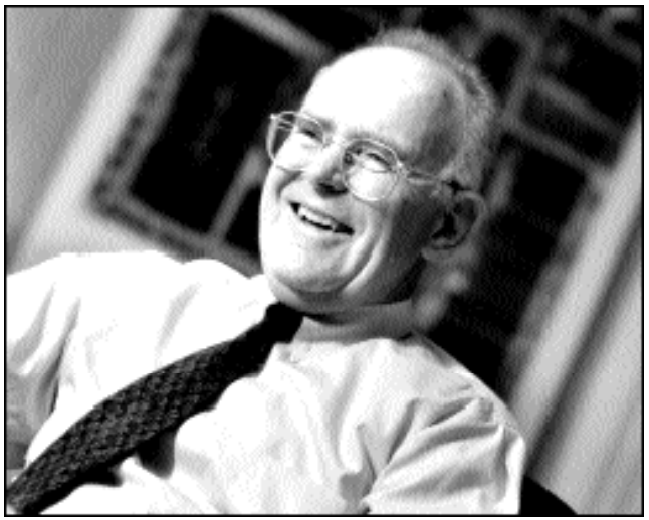
Technology Trends: Microprocessor Capacity



2X transistors/Chip Every 1.5 years

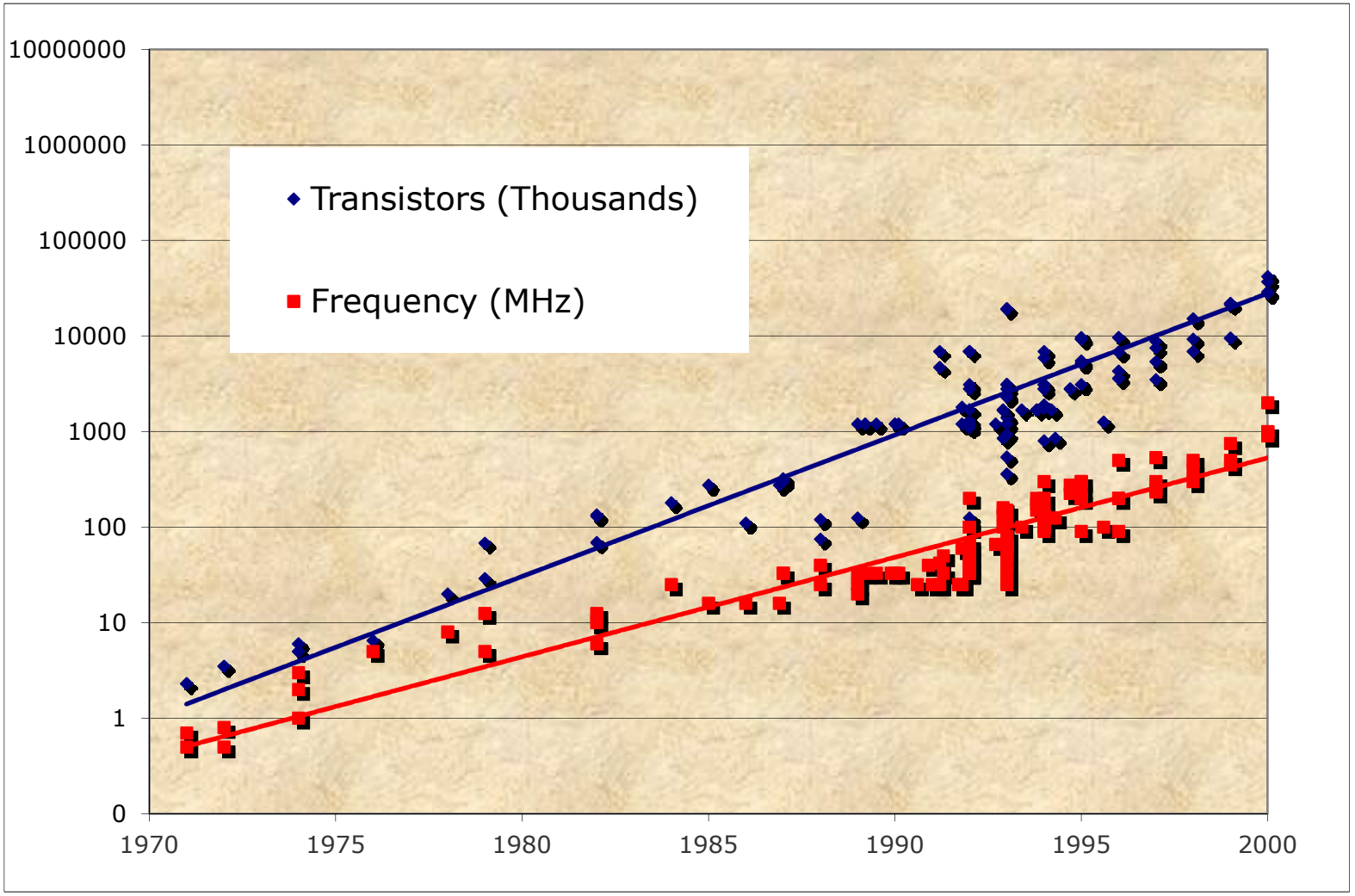
Called "Moore's Law"

Microprocessors have become smaller, denser, and more powerful.



Gordon Moore (co-founder of Intel) predicted in 1965 that the transistors density of semiconductor chips would double roughly every 18 months.

Microprocessor (1970-2000)



Hidden Parallelism Tapped Out

- In the past: increasing circuit density to performance
- Microprocessors - Superscalar designs were the state of the art :
  - What is superscalar?
  - Dynamic scheduling
  - Speculative execution
  - Non-blocking caches
- Apparent path to higher performance?
  - Wider instruction issue
  - Support for more speculation
- Unfortunately, these sources have been used up!

Hidden Parallelism Tapped Out

- Unfortunately, these sources have been used up!
- Two factors:
  - Fundamental circuit limitations
    - delays ⇑ as issue queues ⇑ & multi-port register files ⇑
    - Increasing delays limit performance returns from wider issue
  - Limited amount of ILP
    - ?

"The case for a single-chip multiprocessor", K. Olukotun, B. Nayfeh, L. Hammond, K. Wilson, and K. Chang, ASPLOS-VII, 1996.

Performance Comparison

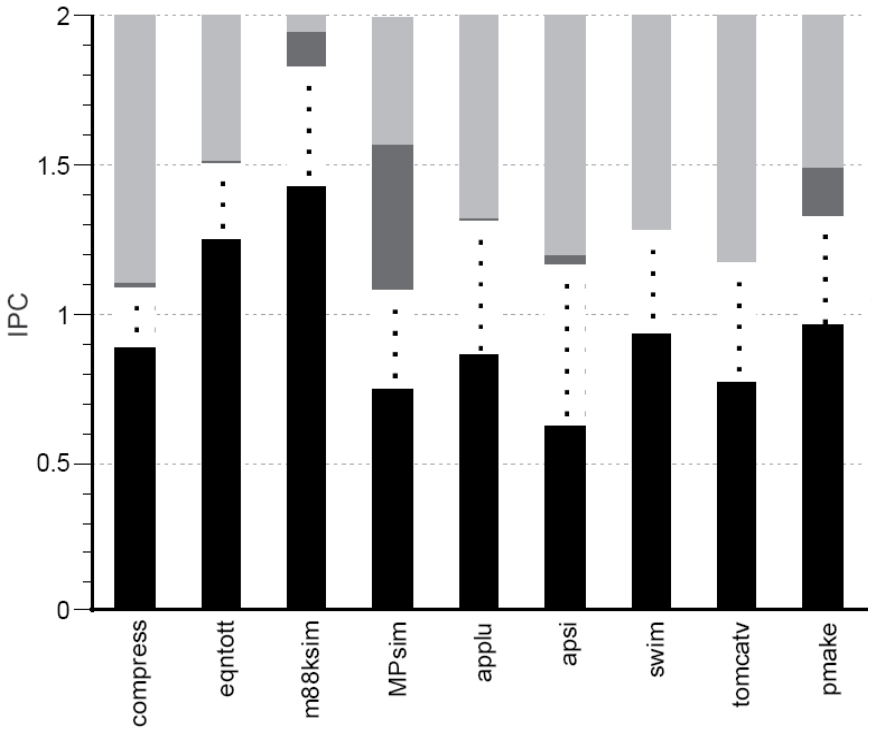


Figure 4. IPC Breakdown for a single 2-issue

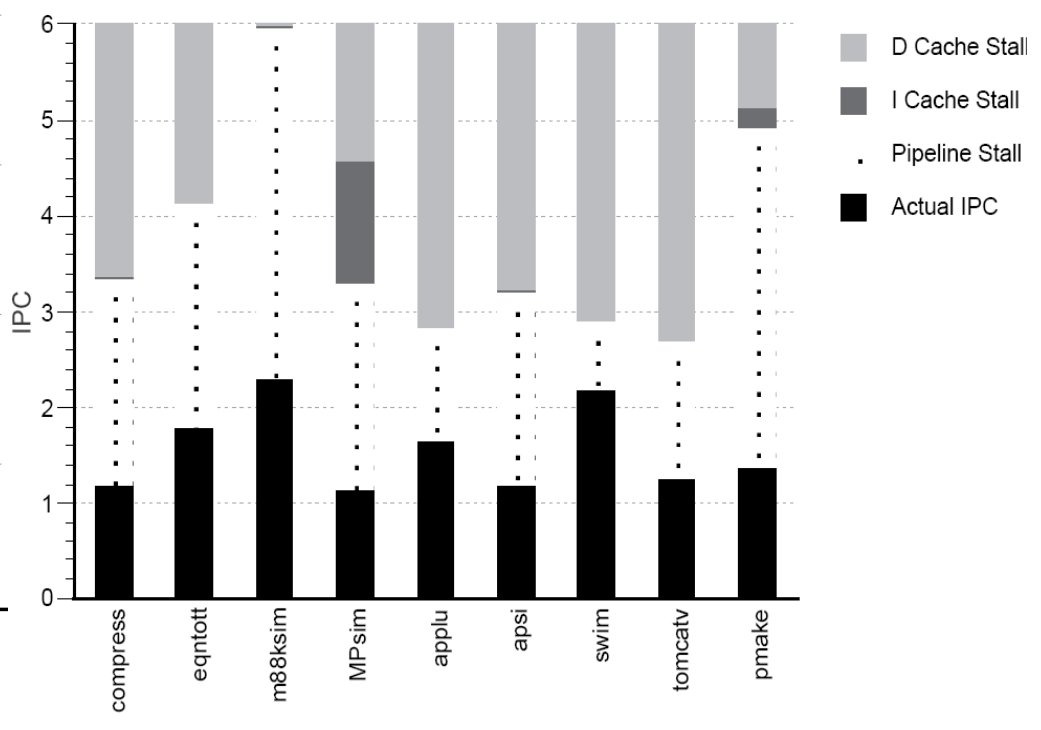


Figure 5. IPC Breakdown for the 6-issue processor.

- The 6-issue has higher IPC than 2-issue, but far less than 3x
- Possible reasons??

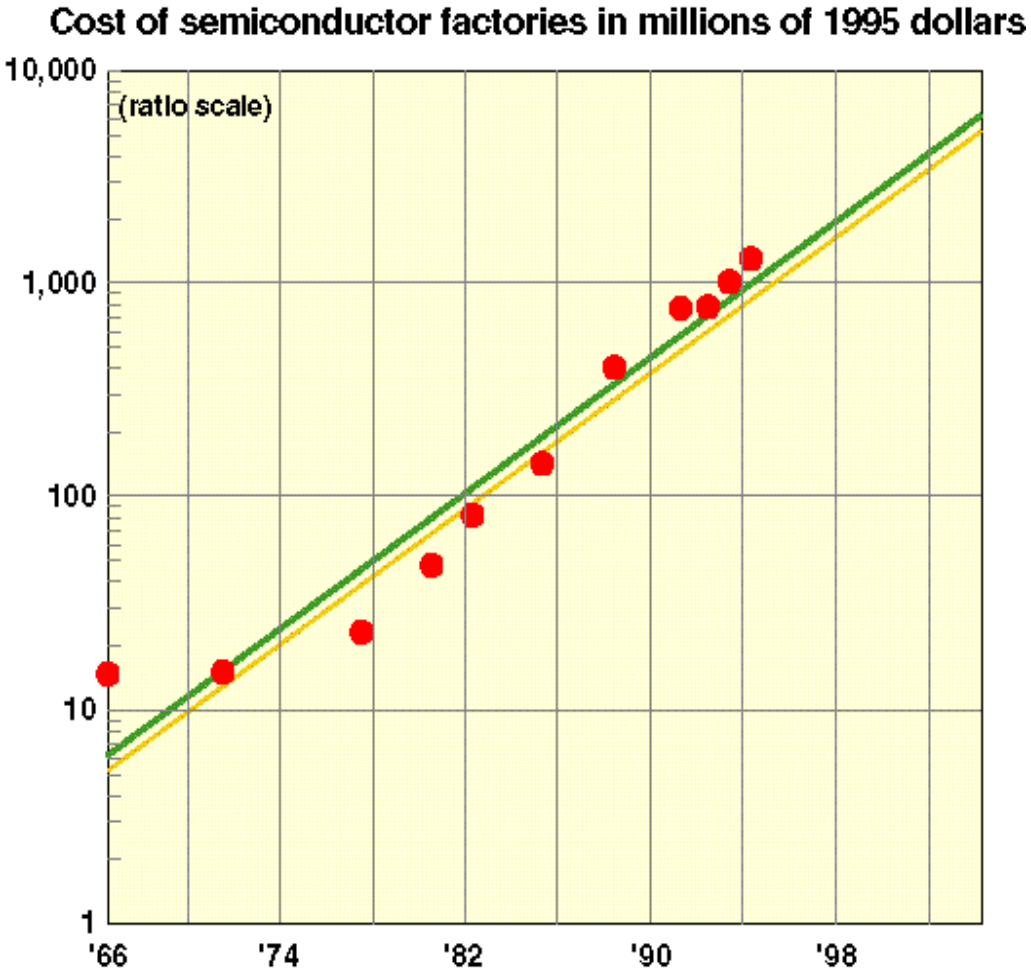
Impact of Device Shrinkage

- What happens when the feature size (transistor size) shrinks by a factor of x ?
- Clock rate goes up by x because wires are shorter
  - actually less than x, because of power consumption
- Transistors per unit area goes up by x^2
- Die size also tends to increase
  - typically another factor of ~x
- Raw computing power of the chip goes up by ~ x^4 !
  - typically x^3 is devoted to on-chip

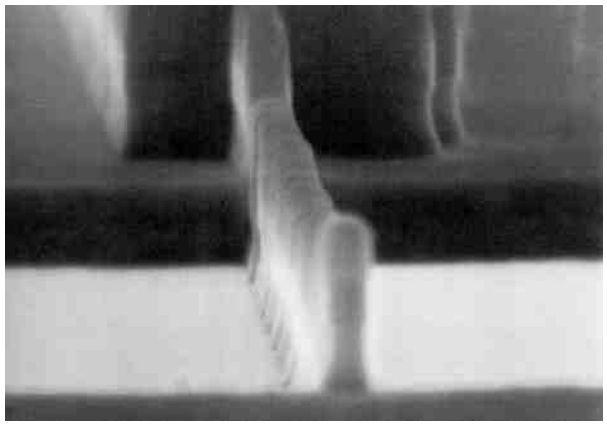
- So most programs x^3 times faster, without changing them

Manufacturing Issues Limit Performance

Manufacturing costs and yield problems limit use of density



- **Moore's 2nd law (Rock's law): ??**



Demo of 0.06 micron CMOS

Source: Forbes Magazine

- **Yield**

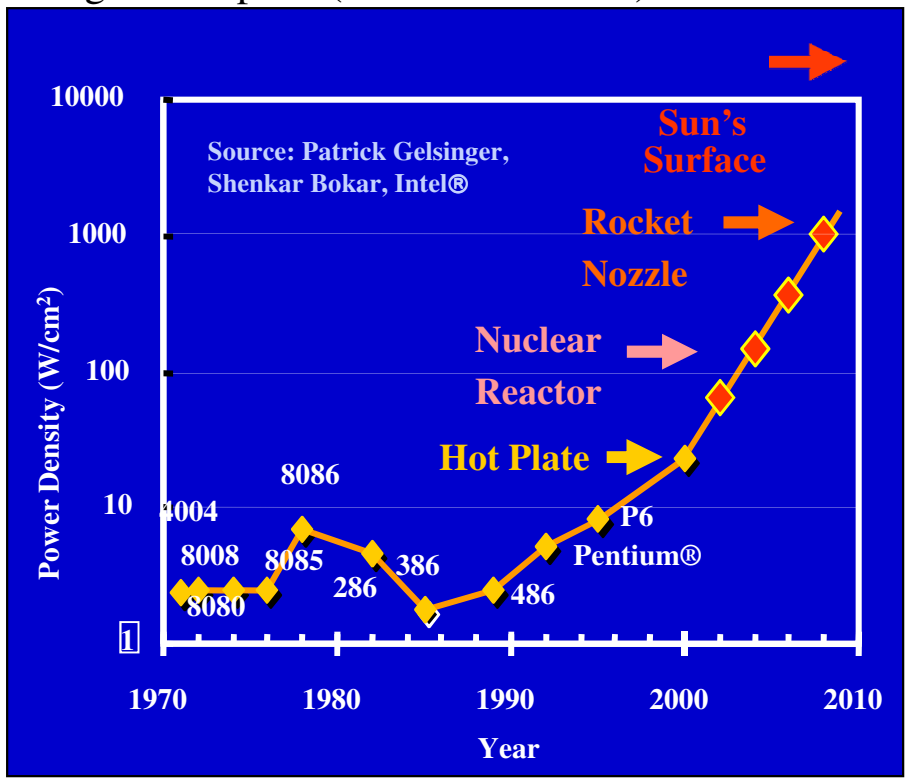
-What percentage of the chips are usable?

-E.g., Cell processor (PS3) is sold with 7 out of 8 "on" to improve yield

Power Density Limits

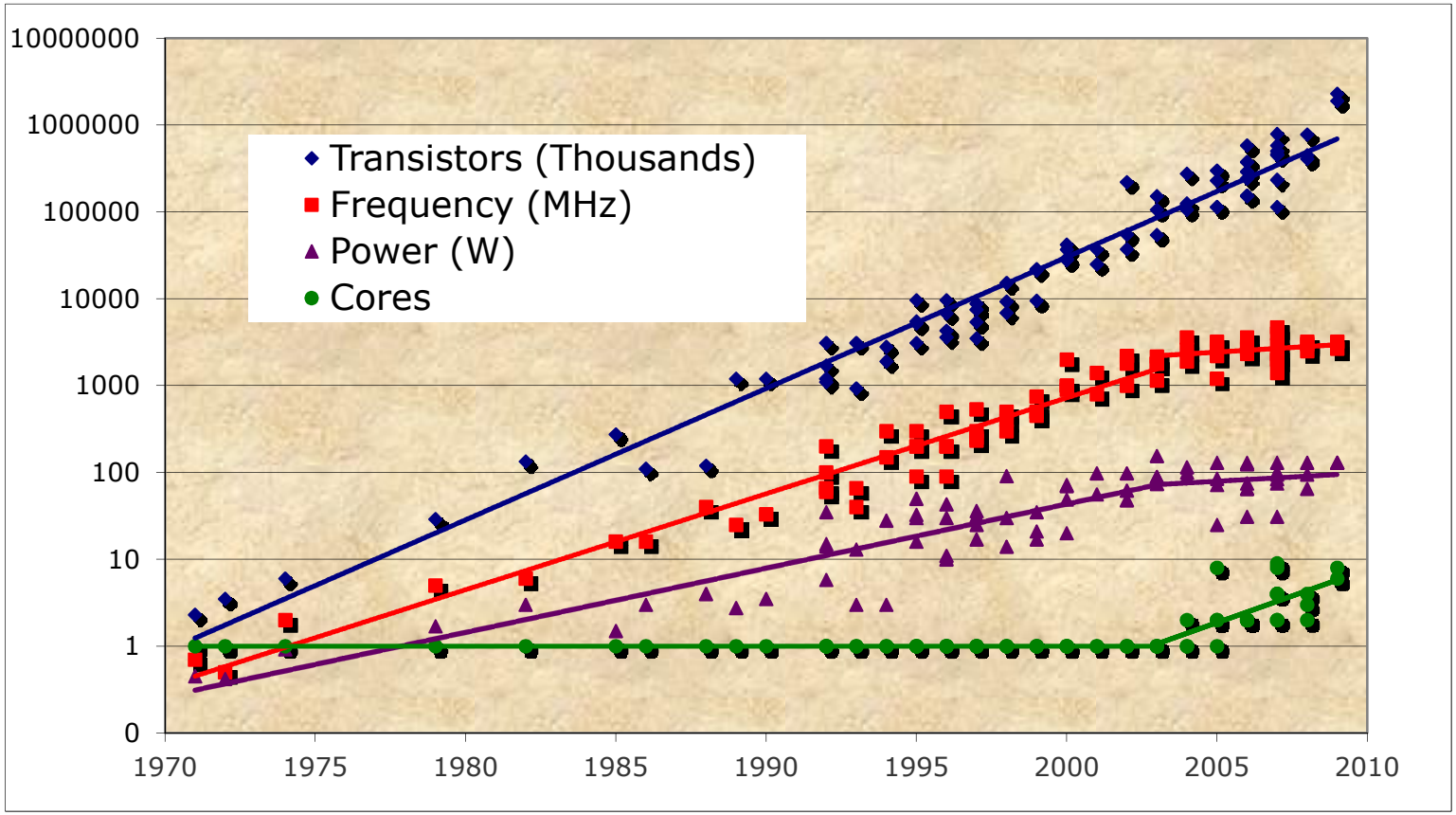
- Concurrent systems are more power efficient
  - Why??

Scaling clock speed (business as usual) will not work



- High performance serial processors waste power
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- More transistors, but not faster serial processors

Revolution in Processors



- Chip density is continuing increase ~2x every 2 years
- Clock speed is not
- Number of processor cores may double instead
- Power is under control, no longer growing

Multicore Chips

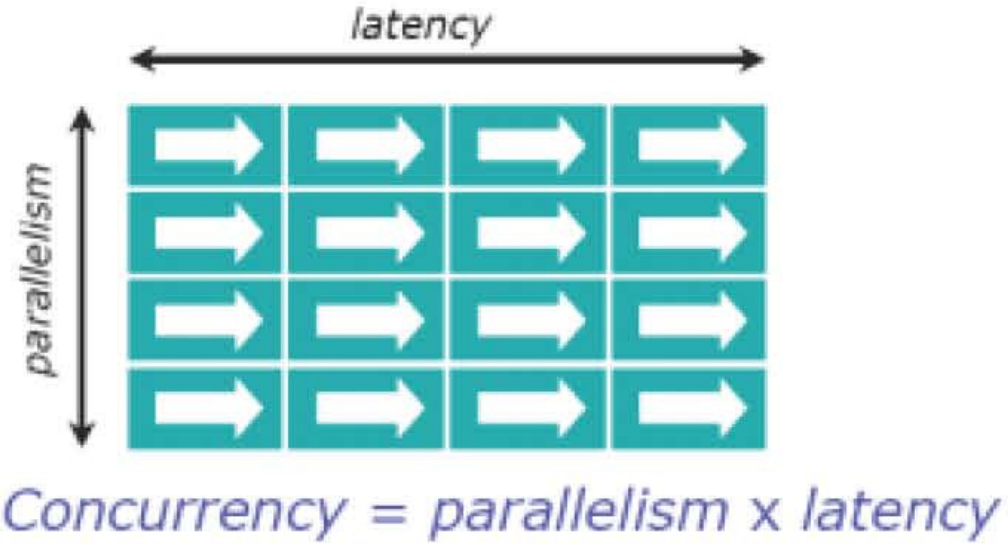
- Single core trends are gloomy
  - Performance is difficult to exploit
  - Perf/power deteriorate
- Fill the die w/ cache has limited value
  - Between 2MB and 4MB cache – gain is diminishing
- Solution: more cores on a die
  - Benefits: higher performance, constant perf/power, less complexity, lower wire delays
  - Challenges: scalability of interconnect, scalability of application, limited memory BW, power
- Other solutions: more integration - memory controller, special functions/accelerators, graphics, …

Why ~~powerful~~ *all* computers are parallel

All major processor vendors are producing parallel computers, because ???

Memory Bottleneck

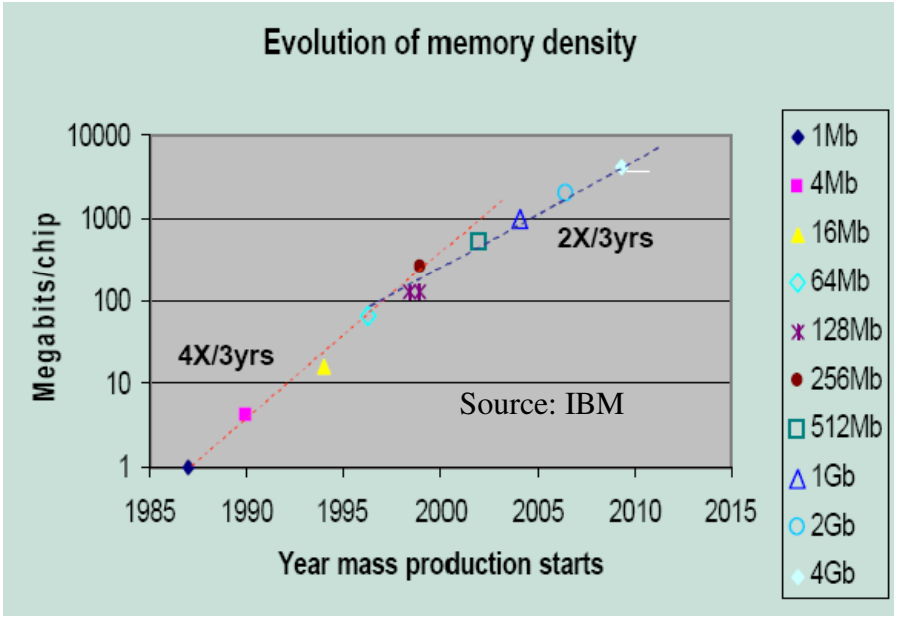
- Cores share finite off-chip bandwidth
- Latency cannot be improved significantly
  - It is already 100s of cycles
- To hide latency, need **more parallelism** than number of cores, schedule data

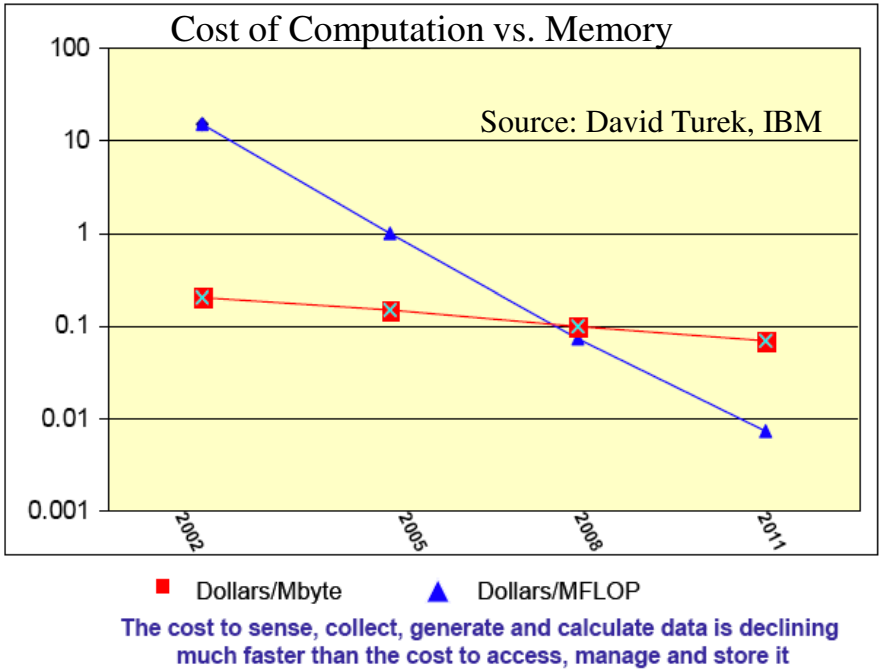


Memory is Not Keeping Pace

Technology trends against a constant or increasing memory per core

- Memory density is doubling every three years; processor logic is every two
- Storage costs ($/MB) are dropping gradually compared to logic costs



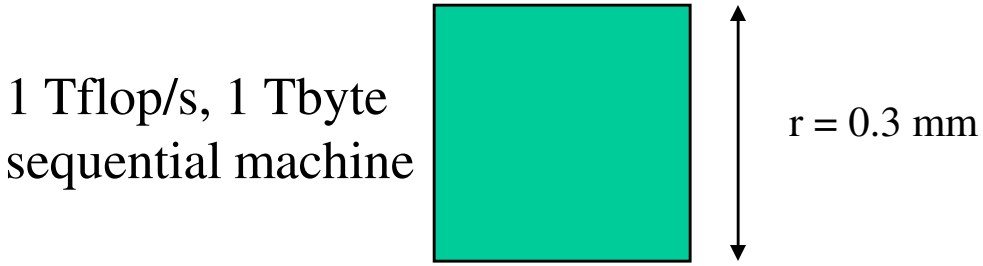


Question: Can we double concurrency without doubling memory?

- Strong scaling: ?
- Weak scaling: ?

More Limits:

How fast can a serial computer be?



- Consider the 1 Tflop/s sequential machine:
  - Data must travel some distance, r, to get from memory to processor.
  - To get 1 data element per cycle, this means 10^{12} times per second at the speed of light, c = $3x10^8$ m/s. Thus r < $c/10^{12}$ = 0.3 mm.
- Now put 1 Tbyte of storage in a 0.3 mm x 0.3 mm area:
  - Each bit occupies about 1 square Angstrom
- No choice but parallelism

The TOP500 Project

- Listing the 500 most powerful computers in the world
- Yardstick: Rmax of Linpack
  - Solve Ax=b, dense problem, matrix is random
  - Dominated by dense matrix-matrix multiply
- Update twice a year:
  - ISC'xy in June in Germany
  - SCxy in November in the U.S.
- All information available from the TOP500 web site at: www.top500.org

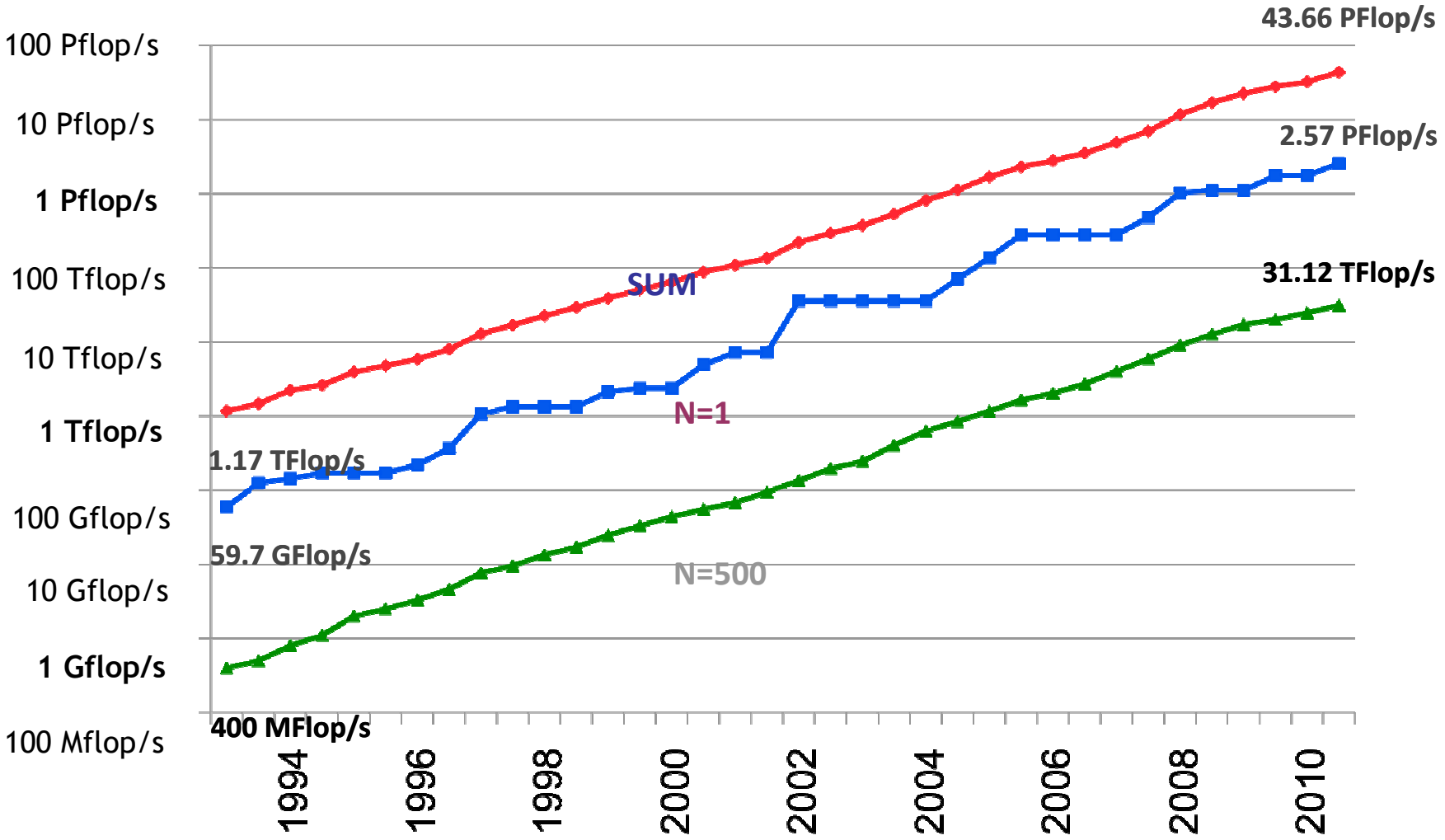
Units of Measure in HPC

- High Performance Computing (HPC) units are:
  - Flop: floating point operation
  - Flops/s: floating point operations per second
  - Bytes: size of data (a double precision floating point number is 8)

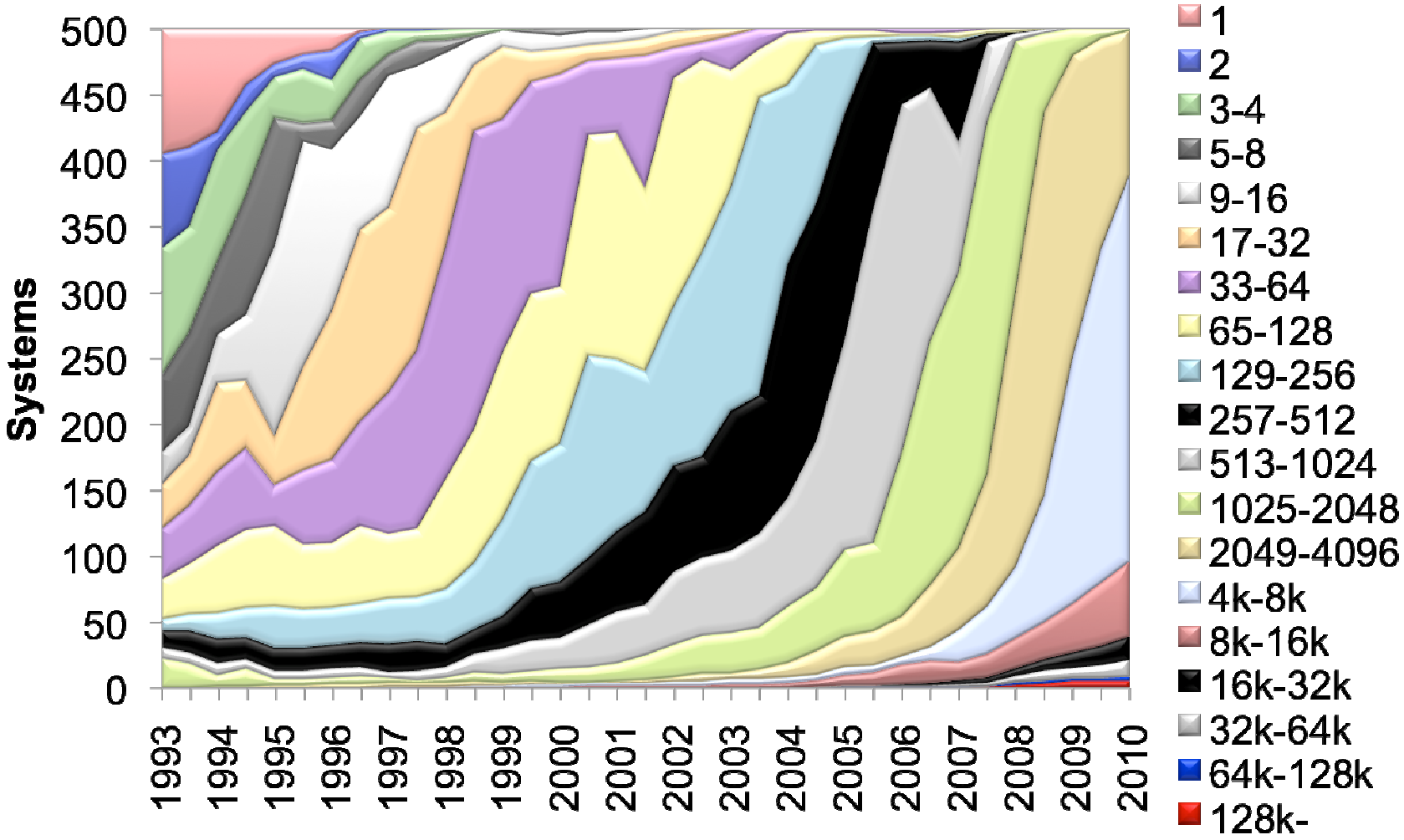
- Typical sizes are millions, billions, trillions…

Mega	Mflop/s = 10^6 flop/sec	Mbyte = 2^{20} = 1048576 ~ 10^6 bytes
Giga	Gflop/s = 10^9 flop/sec	Gbyte = 2^{30} ~ 10^9 bytes
Tera	Tflop/s = 10^{12} flop/sec	Tbyte = 2^{40} ~ 10^{12} bytes
Peta	Pflop/s = 10^{15} flop/sec	Pbyte = 2^{50} ~ 10^{15} bytes
Exa	Eflop/s = 10^{18} flop/sec	Ebyte = 2^{60} ~ 10^{18} bytes
Zetta	Zflop/s = 10^{21} flop/sec	Zbyte = 2^{70} ~ 10^{21} bytes
Yotta	Yflop/s = 10^{24} flop/sec	Ybyte = 2^{80} ~ 10^{24} bytes

Performance Development



Core Count

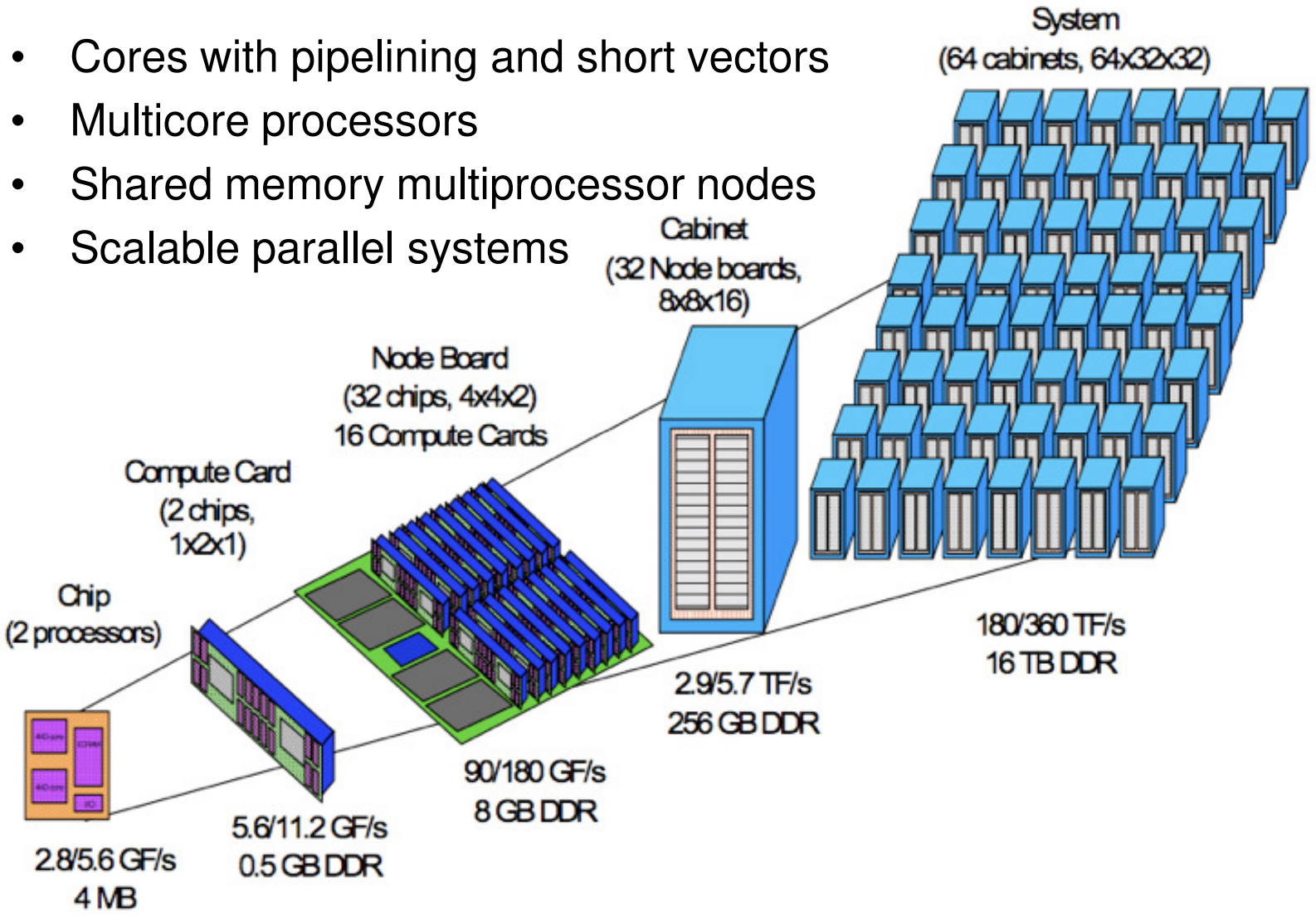

Systems
500
450
400
350
300
250
200
150
100
50
0
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
1
2
3-4
5-8
9-16
17-32
33-64
65-128
129-256
257-512
513-1024
1025-2048
2049-4096
4k-8k
8k-16k
16k-32k
32k-64k
64k-128k
128k-

Moore's Law reinterpreted

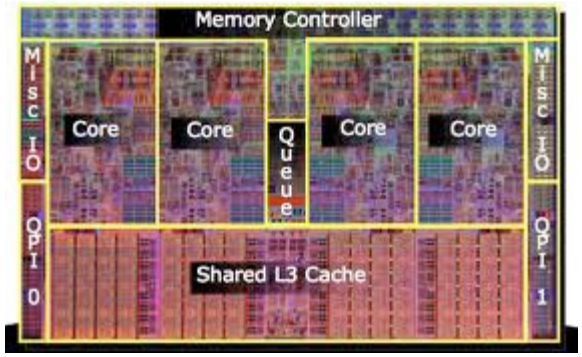
- Number of cores per chip will double every two years
- Clock speed will not increase (possibly decrease)
- Need to deal with systems with millions of concurrent threads
- Need to deal with inter-chip parallelism as well as intra-chip parallelism

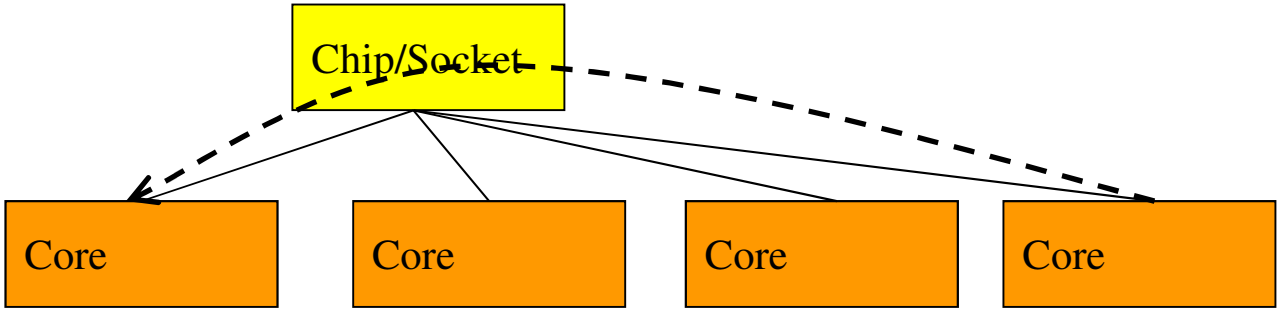
Parallel Computers

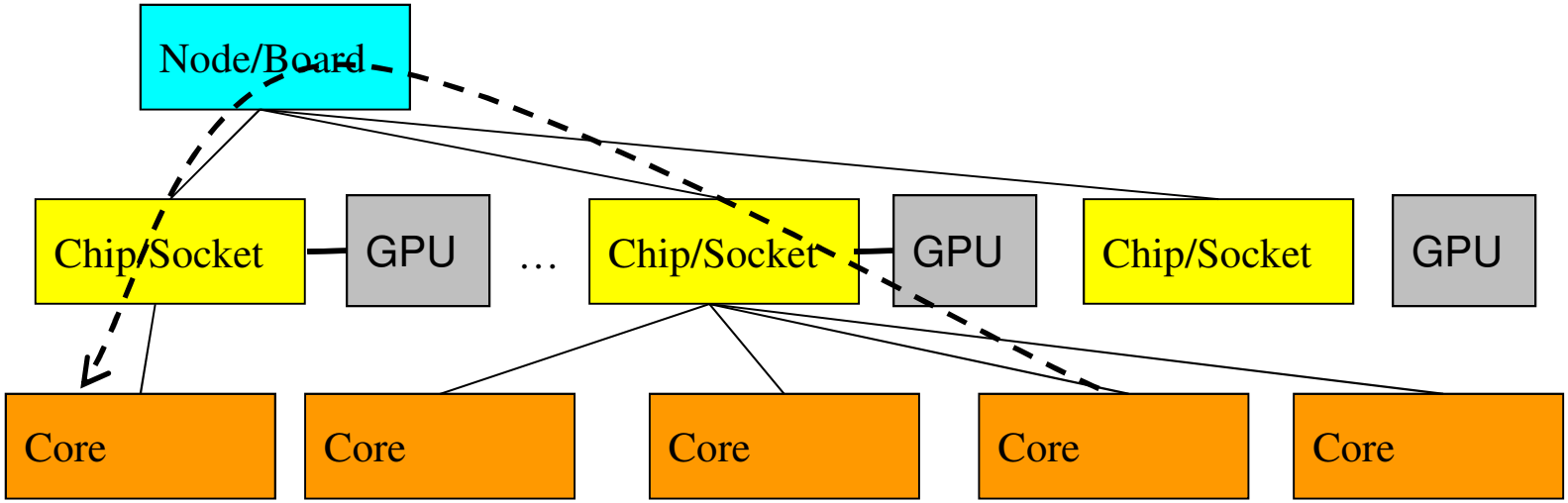
- Cores with pipelining and short vectors
- Multicore processors
- Shared memory multiprocessor nodes
- Scalable parallel systems



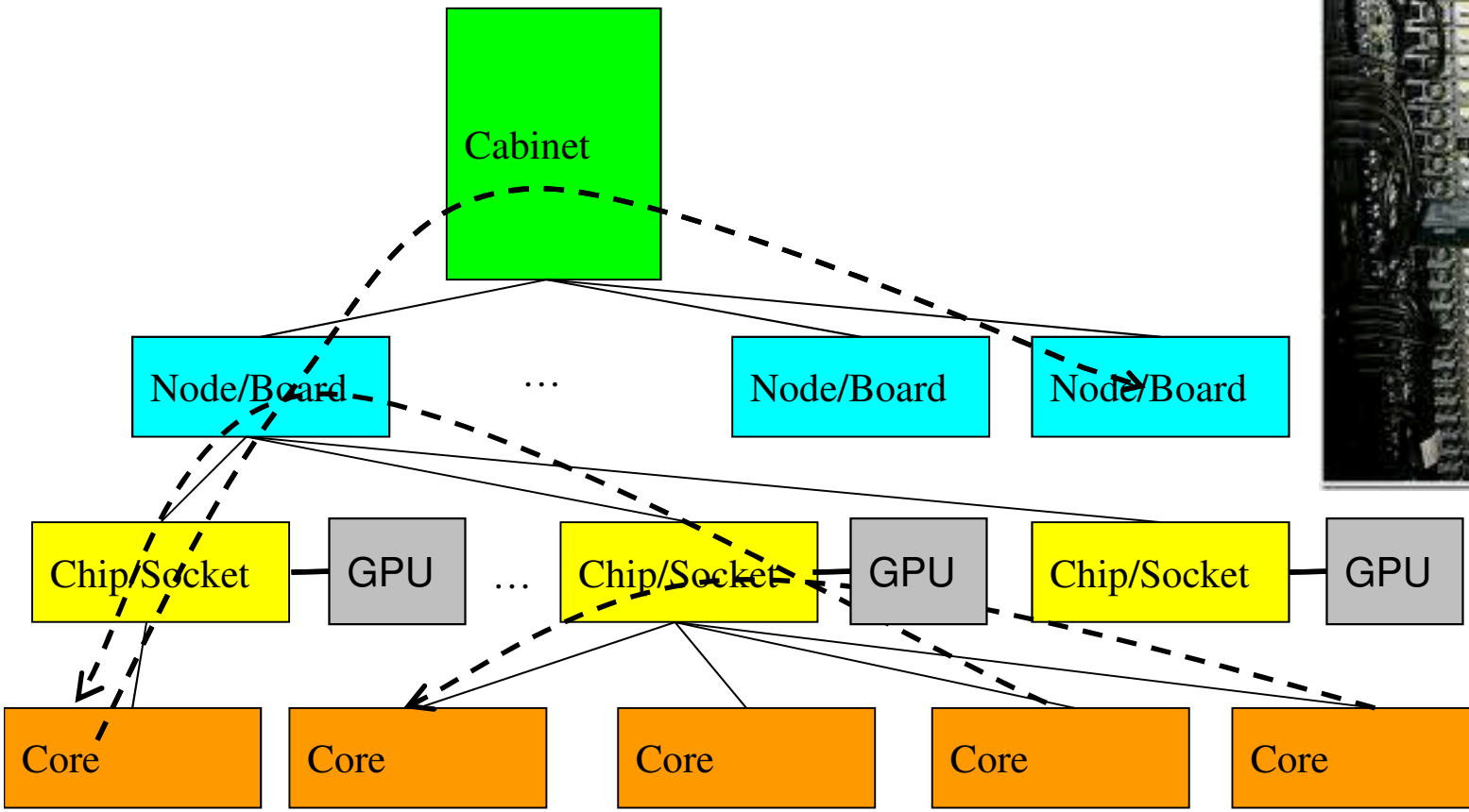
Typical Parallel Machine

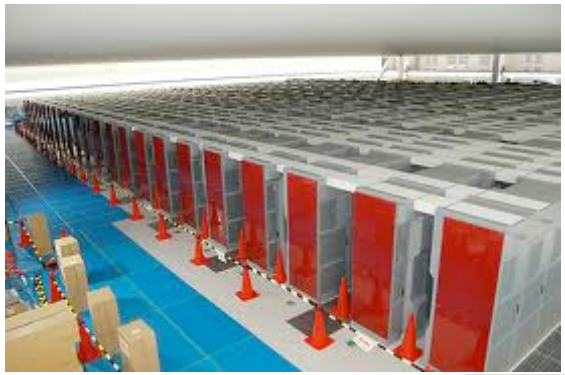





Node/Board
Chip/Socket
GPU
…
Chip/Socket
GPU
Chip/Socket
GPU
Core
Core
Core
Core
Core

Shared memory programming between processes on a board and a combination of shared memory and distributed memory programming between nodes and cabinets





Combination of shared memory and distributed memory programming

Exascale Systems w/ a cap of $200M and 20MW

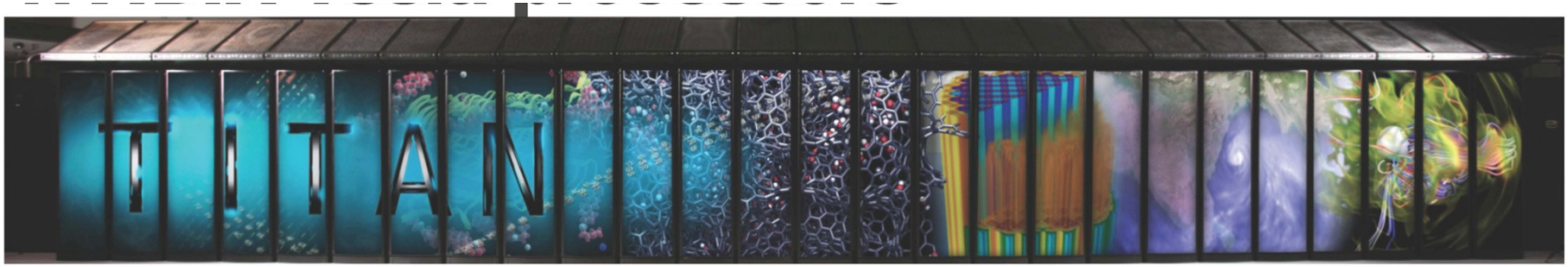
Systems	2013 Tianhe-2	2020-2022	Difference Today & Exa
System peak	55 Pflop/s	1 Eflop/s	~20x
Power	18 MW (3 Gflops/W)	~20 MW (50 Gflops/W)	O(1) ~15x
System memory	1.4 PB (1.024 PB CPU + .384 PB CoP)	32 - 64 PB	~50x
Node performance	3.43 TF/s (.4 CPU +3 CoP)	1.2 or 15TF/s	O(1)
Node concurrency	24 cores CPU + 171 cores CoP	O(1k) or 10k	~5x - ~50x
Node Interconnect BW	6.36 GB/s	200-400GB/s	~40x
System size (nodes)	16,000	O(100,000) or O(1M)	~6x - ~60x
Total concurrency	3.12 M 12.48M threads (4/core)	O(billion)	~100x
MTTF	Few / day	Many / day	O(?)

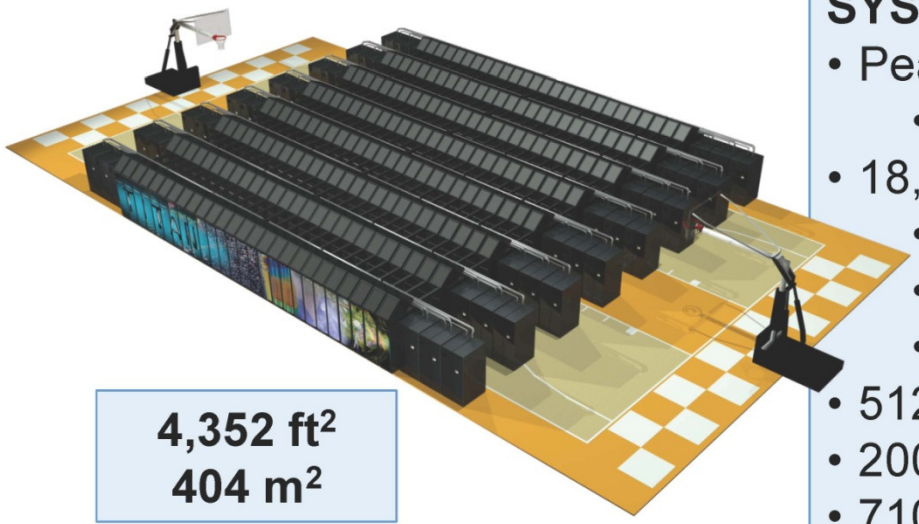
Jack Dongarra SC'13

Future Computer Systems

- Most likely to be a **hybrid** design
  - Think standard multicore chips and accelerators (GPUs)
- Today accelerators are attached
- Next generation will be more **integrated**
- DOE CORAL (Collaboration of Oak Ridge, Argonne, and Livermore)
  - Procurement of pre-exascale supercomputers
  - Summit at ORNL and Sierra at LLNL: Nvidia GPU-accelerated systems based on the IBM OpenPOWER platforms
    - *NVLink* is a key building block in these machines, which enables IBM POWER CPUs and Nvidia GPUs to access each other's memory fast
  - Aurora at ANL: Intel Xeon Phi Gen 3 with Cray "Shasta" platform

Titan at ORNL: Cray XK7 with AMD Opteron and Nvidia Tesla





4,352 ft²
404 m²

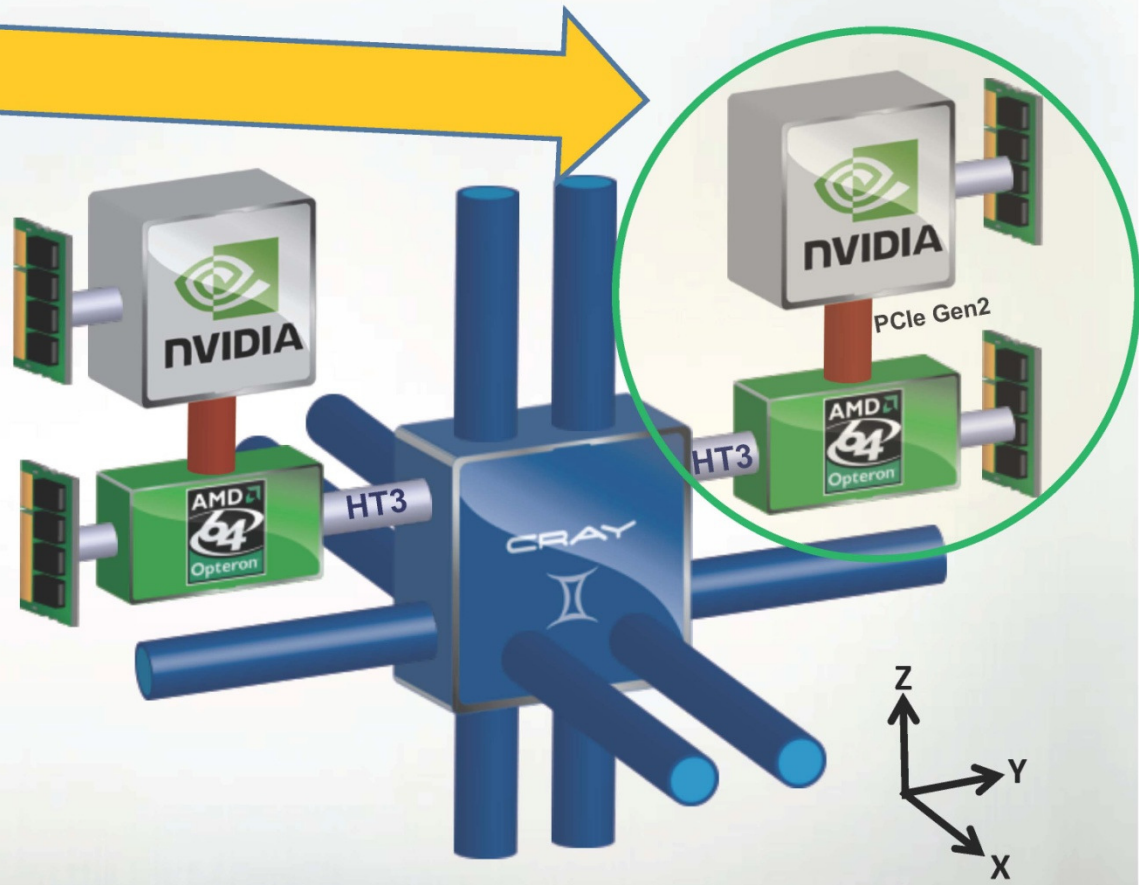
SYSTEM SPECIFICATIONS:

- Peak performance of 27 PF
  - 24.5 Pflop/s GPU + 2.6 Pflop/s AMD
- 18,688 Compute Nodes each with:
  - 16-Core AMD Opteron CPU
  - NVIDIA Tesla "K20x" GPU
  - 32 + 6 GB memory
- 512 Service and I/O nodes
- 200 Cabinets
- 710 TB total system memory
- Cray Gemini 3D Torus Interconnect
- 9 MW peak power

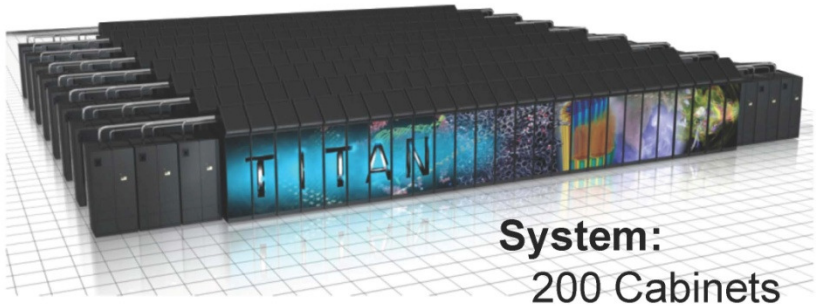
OAK RIDGE

Cray XK7 Compute Node

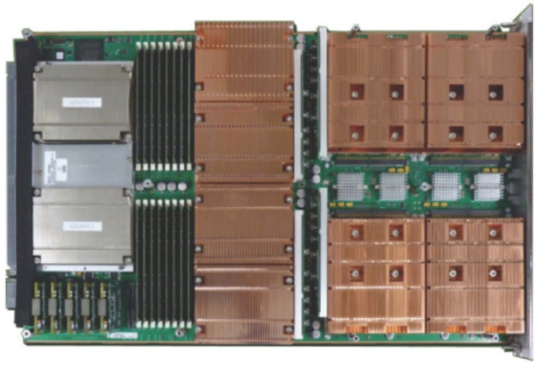
XK7 Compute Node Characteristics
AMD Opteron 6274 Interlagos 16 core processor
Tesla K20x @ 1311 GF
Host Memory 32GB 1600 MHz DDR3
Tesla K20x Memory 6GB GDDR5
Gemini High Speed Interconnect



Titan:
Cray XK7 System

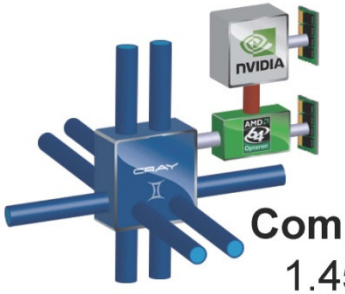


System:
200 Cabinets
18,688 Nodes
27 PF
710 TB



Cabinet:
24 Boards
96 Nodes
139 TF
3.6 TB

Board:
4 Compute Nodes
5.8 TF
152 GB



Compute Node:
1.45 TF
38 GB

Principles of Parallel Computing

- Parallelism and Amdahl's Law
- Finding and exploiting granularity
- Preserving data locality
- Load balancing
- Coordination and synchronization
- Performance modeling

All of these things make parallel programming more difficult than sequential programming!

Challenges of Parallelism

- "Automatic" parallelism in modern machines
  - Bit level parallelism – within floating point operations
  - Instruction level parallelism (ILP) – multiple instructions per clock cycle
  - Memory system parallelism – overlap of memory accesses with computation
  - OS parallelism – multiple jobs run in parallel on SMPs

- There are limitations to all of these!
- To achieve high performance, the programmer needs to identify, schedule and coordinate parallel tasks and data!

Challenges of Parallelism

- Applications are often very sophisticated
  - E.g., adaptive algorithms may require dynamic load balancing
- Algorithm development is harder
  - Complexity of specifying and coordinating concurrent activities
  - Algorithmic scalability losses
  - Serialization and load imbalance
- Software development is much harder
  - Lack of development tools and programming models
  - Subtle program errors: race conditions
  - Multilevel parallelism is difficult to manage
  - Communication and/or IO bottlenecks
- Rapid pace of change in computer system architecture
  - A parallel algorithm for one machine may not be a good match for another
  - Homogeneous multicore processors vs GPGPU
- Extreme scale exacerbates inefficiencies

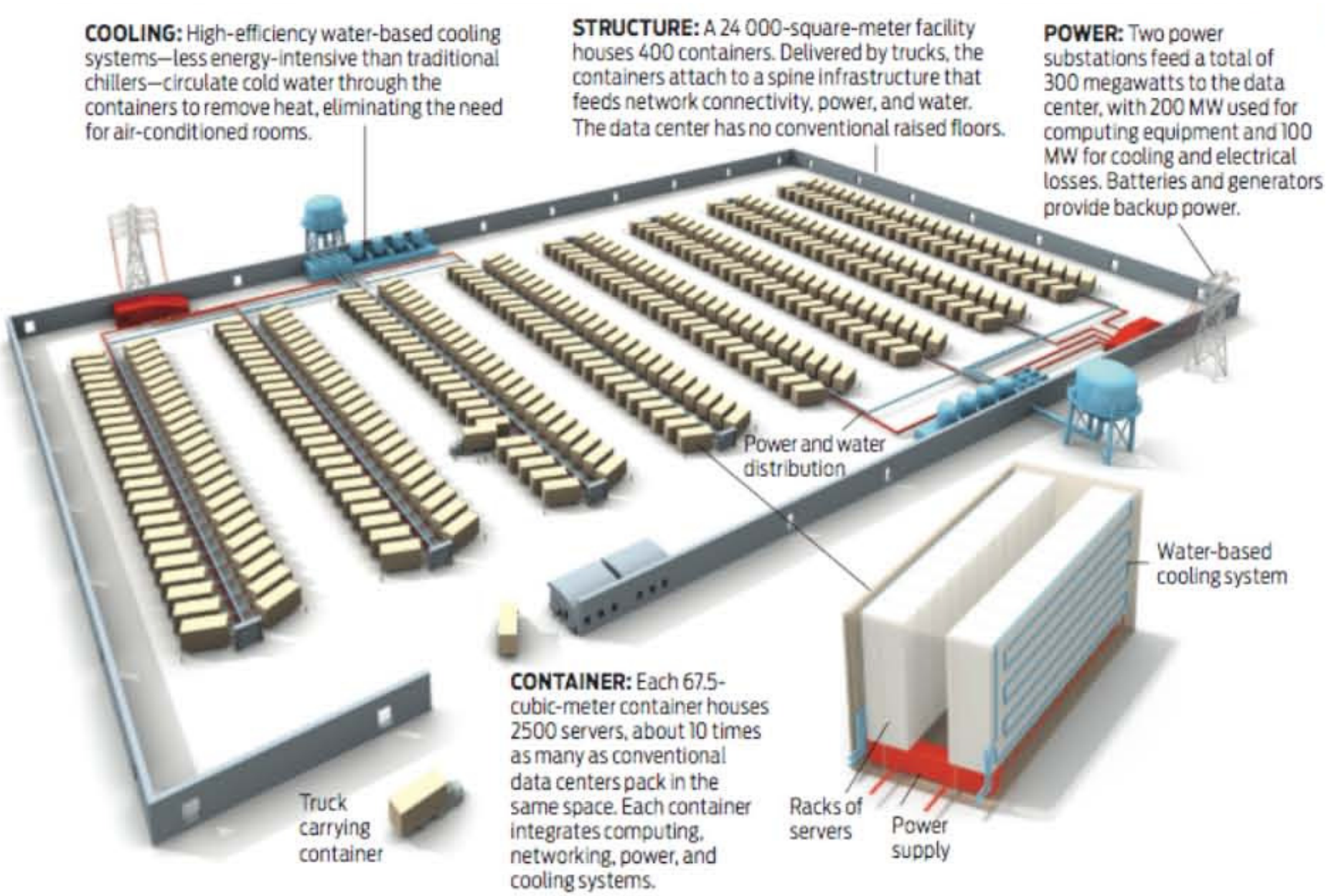
Parallel Systems

- All processors will be multi-core
- All computers will be massively parallel
- All programmers will be parallel programmers
- All programs will be parallel programs

- The key challenge: how to enable mainstream developers to scale their applications on machines!
  - Algorithms – scalable, asynchronous, fault tolerant
  - Programming models and systems
  - Data management
  - Runtime system support

Enterprise Applications

- Program = web search, email, map/GIS, …
- Computer = 1000's computers, storage, network
- Warehouse-sized facilities and workloads
- New datacenter ideas (2007-2008): truck container (Sun), floating (Google), datacenter-in-a-tent (Microsoft)
- How to enable innovation in new services w/o first building and capitalizing a large company?


COOLING: High-efficiency water-based cooling systems—less energy-intensive than traditional chillers—circulate cold water through the containers to remove heat, eliminating the need for air-conditioned rooms.
STRUCTURE: A 24 000-square-meter facility houses 400 containers. Delivered by trucks, the containers attach to a spine infrastructure that feeds network connectivity, power, and water. The data center has no conventional raised floors.
POWER: Two power substations feed a total of 300 megawatts to the data center, with 200 MW used for computing equipment and 100 MW for cooling and electrical losses. Batteries and generators provide backup power.
Power and water distribution
Water-based cooling system
CONTAINER: Each 67.5-cubic-meter container houses 2500 servers, about 10 times as many as conventional data centers pack in the same space. Each container integrates computing, networking, power, and cooling systems.
Truck carrying container
Racks of servers
Power supply

Enterprise Datacenter

- A 50-rack data center
  - X 64 blades per rack
  - X 2 sockets per blade
  - X 32 cores per socket
  - X 10 VMs per core
  - X 10 Java objects per VM

20,480,000 managed objects!

Enterprise Scalability - Problem

- How do you scale up?
  - Run jobs processing 100's of terabytes of data
  - Takes 11 days to read on 1 computer
- Need lots of cheap computers
  - Fixes speed problem (15 min on 1000 computers), but
  - Reliability issue: large computers fail every day
- Need common infrastructure
  - Must be efficient and reliable

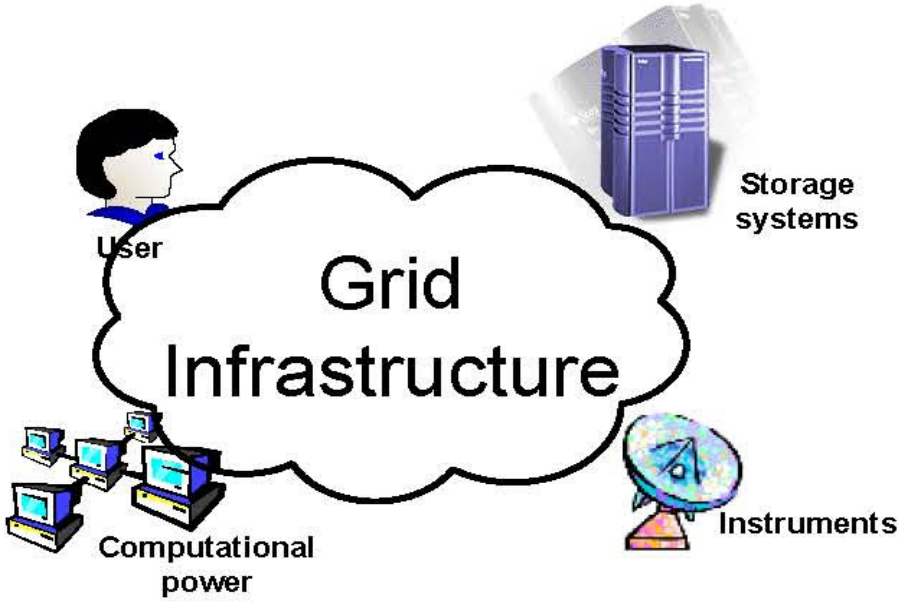
Enterprise Scalability - Solution

- Open source Apache Project
- Hadoop Core includes
  - Distributed file systems – distribute data
  - MapReduce – distribute application
- Written in Java
- Runs on commodity HW, Linux, Mac OS/X, Windows, …

Hadoop Clusters:
~20,000 machines running Hadoop @Yahoo!
Hundreds of thousands of jobs per month

Distributed and Cloud Computing

- Peer-to-peer systems
  - Flexible network of client machines logically connected by an overlay network
- Data/computational Grids
  - Heterogeneous clusters interconnected by high-speed network links over selected resource sites

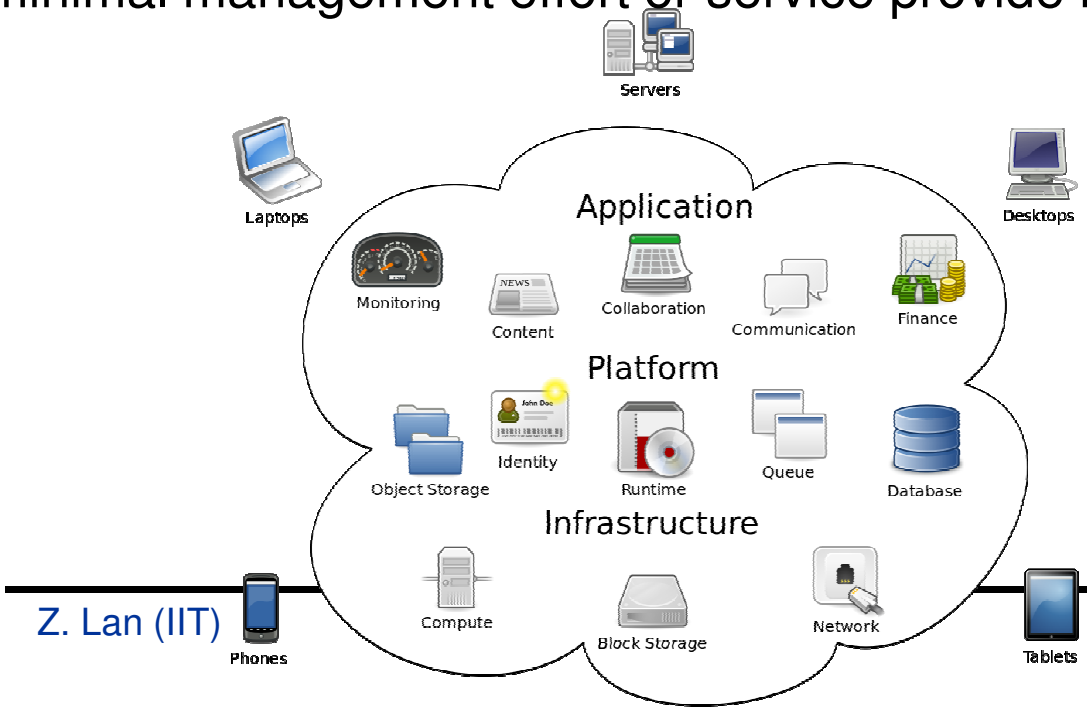


Key enabler:
- Network vs. computer performance
- Computer speed doubles every 18 mo. vs. network speed doubles every 9 mo.

Key characteristics:
- Sharing data
- Distributing computation
- Coordinating problem solving
- Accessing remove instrumentation

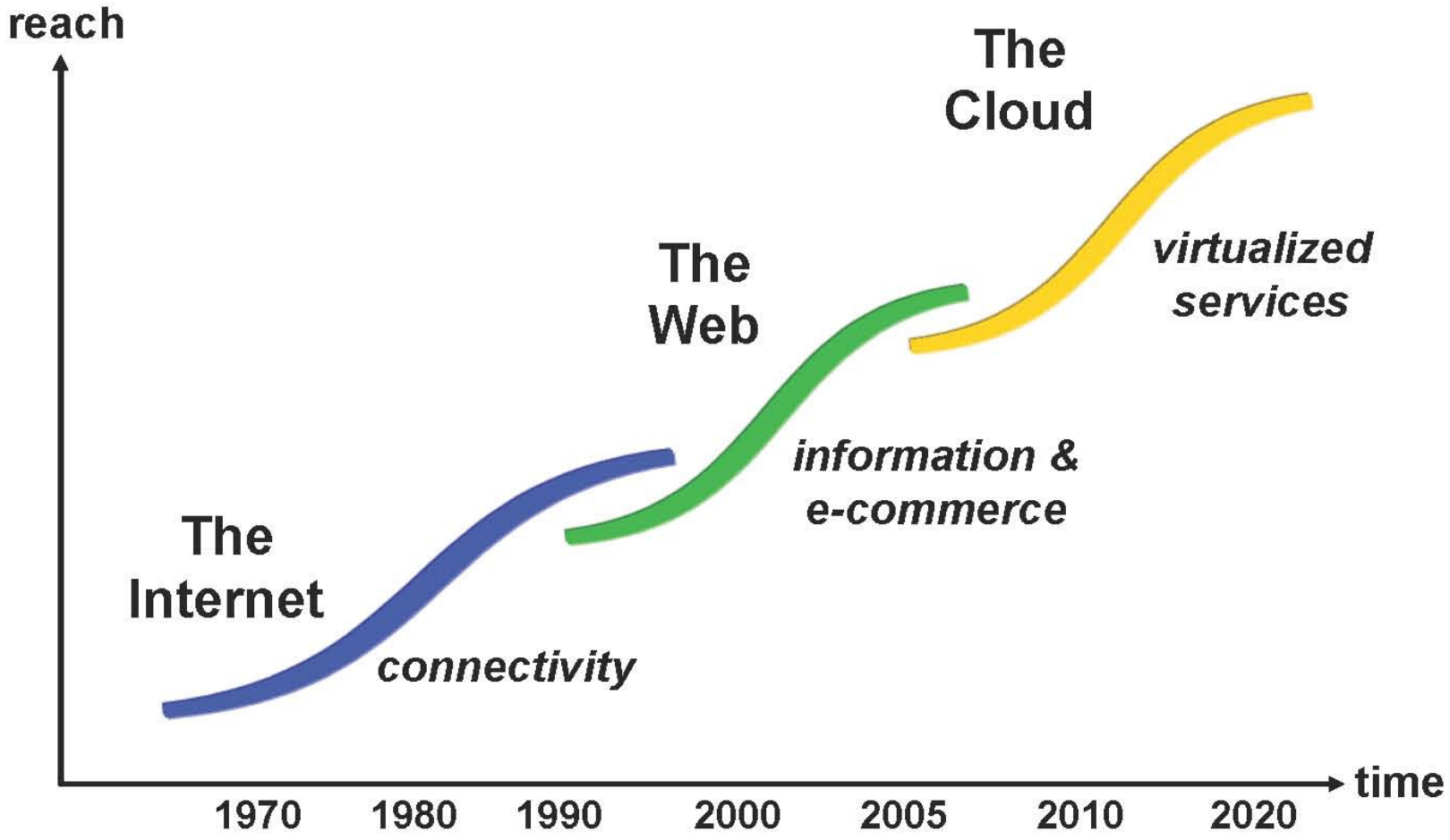
Cloud Computing

- **Cloud computing** is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand like a public utility (**Wikipedia**)
- **A cloud** is a computing capability that provides an abstraction between the computing resource and its underlying technical architecture, enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provide interaction (**NIST**)



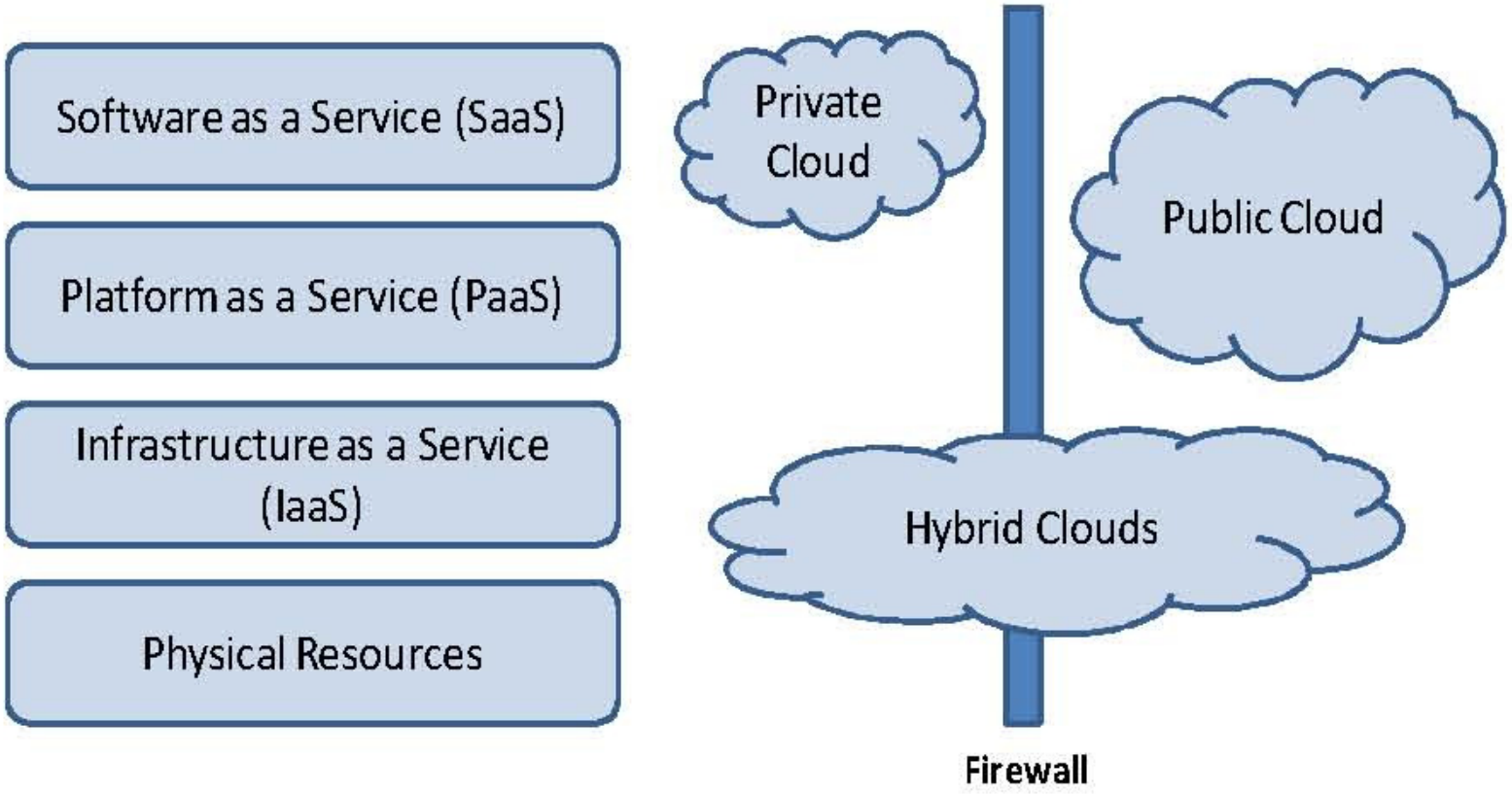
Deployment models:
- Public cloud
- Private cloud
- Hybrid cloud

The Cloud Evolution

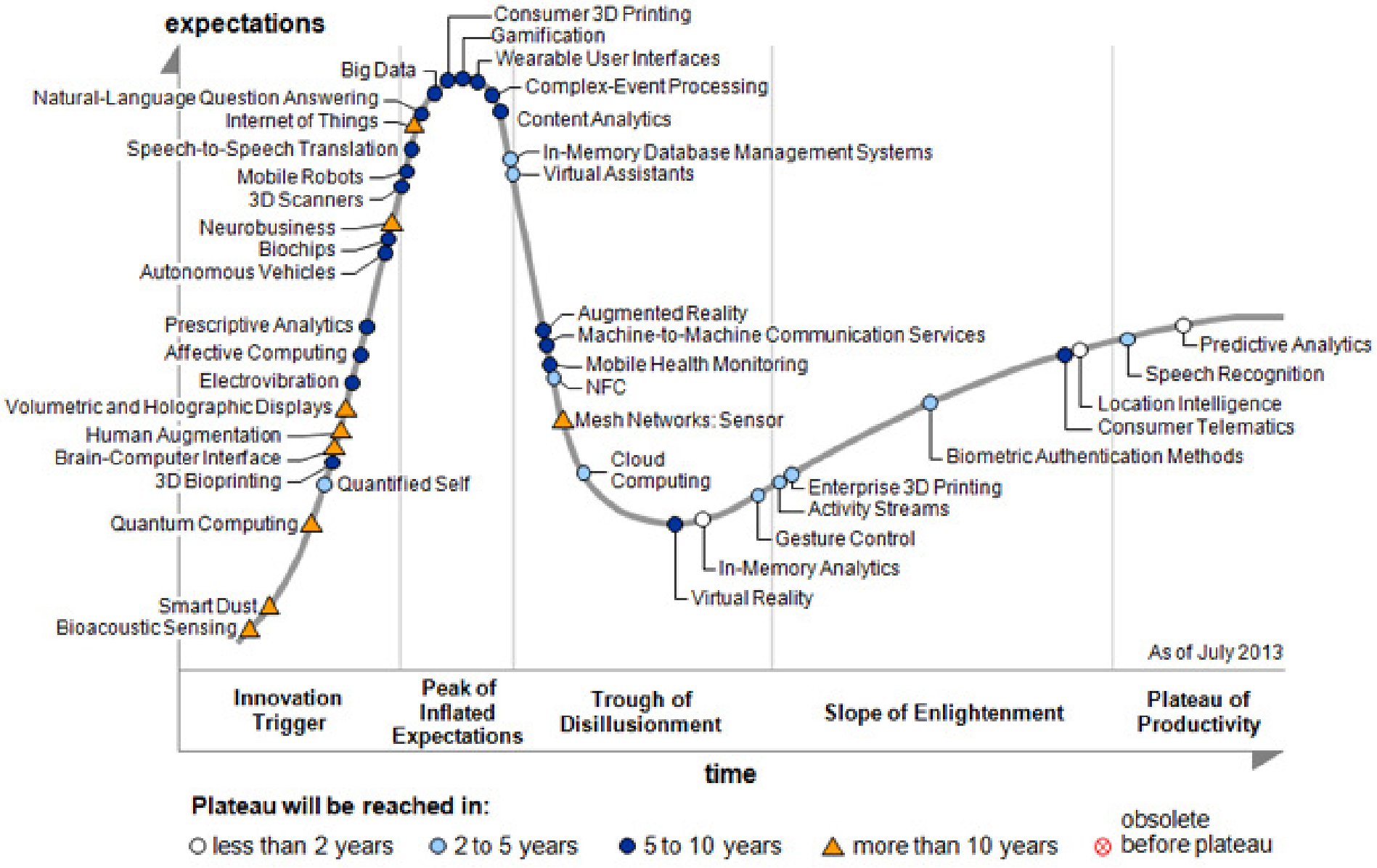


Ack. Manish Parashar (Rutgers)

Cloud Stack



- A seductive abstraction – unlimited resources, always on and accessible !
- Multiple entry points: SaaS, PaaS, IaaS, HaaS
- Pay as you go, for what you use!
- Economy - IT outsourcing, TCO, capital costs, operation costs, …


expectations
Consumer 3D Printing
Gamification
Wearable User Interfaces
Big Data
Complex-Event Processing
Natural-Language Question Answering
Content Analytics
Internet of Things
In-Memory Database Management Systems
Speech-to-Speech Translation
Virtual Assistants
Mobile Robots
3D Scanners
Neurobusiness
Biochips
Autonomous Vehicles
Augmented Reality
Prescriptive Analytics
Machine-to-Machine Communication Services
Affective Computing
Mobile Health Monitoring
Electrovibration
NFC
Volumetric and Holographic Displays
Mesh Networks: Sensor
Human Augmentation
Brain-Computer Interface
Cloud Computing
3D Bioprinting
Quantified Self
Enterprise 3D Printing
Activity Streams
Quantum Computing
Gesture Control
In-Memory Analytics
Virtual Reality
Smart Dust
Bioacoustic Sensing
Predictive Analytics
Speech Recognition
Location Intelligence
Consumer Telematics
Biometric Authentication Methods
As of July 2013
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached in:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau

Summary

- What is parallel computing
- Why parallel computing
  - Application pull
  - Technology push
- Distributed and cloud computing

- Reading:
  - Kumar – ch 1; Foster – ch 1