High Performance Computing: History and Trend

What is HPC?

Supercomputing is the biggest, fastest computing right this minute.

Likewise, a supercomputer is one of the biggest, fastest computers right this minute.

So, the definition of supercomputing is constantly changing.

Rule of Thumb : a supercomputer is typically at least 100 times as powerful as a PC.

The definition might be outdated.

What is HPC About?

- Size: many problems that are interesting to scientists and engineers can't fit on a PC – usually because they need more than a few GB of RAM, or more than a few 100 GB of disk.
- Speed: many problems that are interesting to scientists and engineers would take a very very long time to run on a PC: months or even years. But a problem that would take a month on a PC might take only a few hours on a supercomputer.

What is HPC Used For?

- Simulation of physical phenomena, such as
  - Weather forecasting
  - Galaxy formation
  - Oil reservoir management
- Data mining: finding useful information in a sea of data,
  such as
  - Gene sequencing
  - Signal processing
  - Detecting storms that could produce tornados
- Visualization: turning a vast sea of data into pictures that a scientist can understand
- Consumer applications

HPC Issues

- The tyranny of the storage hierarchy
- Parallelism: doing many things at the same time
  - Instruction-level parallelism: doing multiple operations at the same time within a single processor (e.g., add, multiply, load and store simultaneously)
  - Multiprocessing: multiple CPUs working on different parts of a problem at the same time
    - Shared Memory Multithreading
    - Distributed Multiprocessing
- High performance compilers
- Scientific Libraries
- Visualization

Why Bother with HPC at All?

- It's clear that making effective use of HPC takes quite a bit of effort, both learning how and developing software.
- That seems like a lot of trouble to go to just to get your code to run faster.
- It's nice to have a code that used to take a day run in an hour. But if you can afford to wait a day, what's the point of HPC?
- Why go to all that trouble just to get your code to run faster?

Why HPC is Worth the Bother

- What HPC gives you that you won't get elsewhere is the ability to do bigger, better, more exciting science. If your code can run faster, that means that you can tackle much bigger problems in the same amount of time that you used to need for smaller problems.
- HPC is important not only for its own sake, but also because what happens in HPC today will be on your desktop in about 15 years: it puts you ahead of the curve.

Modern Definitions (I)

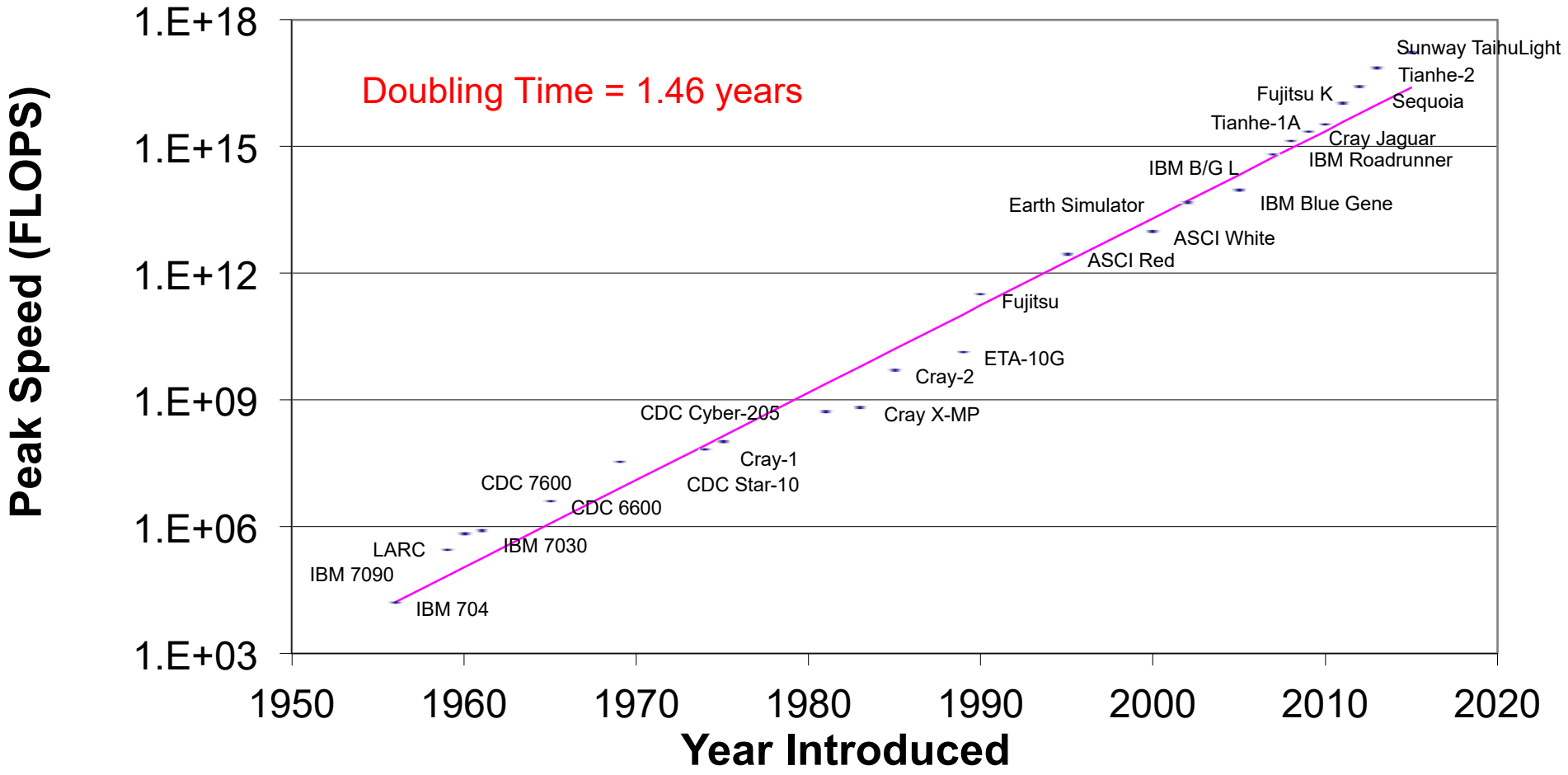
Mainframe (a.k.a. “big iron”)

- not so much fast speed but massive memory, I/O, high-quality internal engineering and technical support
- typically runs proprietary OS (e.g. UNIX variant) or may host multiple OS to replace 100s of PCs with “virtual” PCs (e.g. Linux)
- runs for years without interruption, with “hot” repairs
- company-wide resource
- cost ~ £100-500k

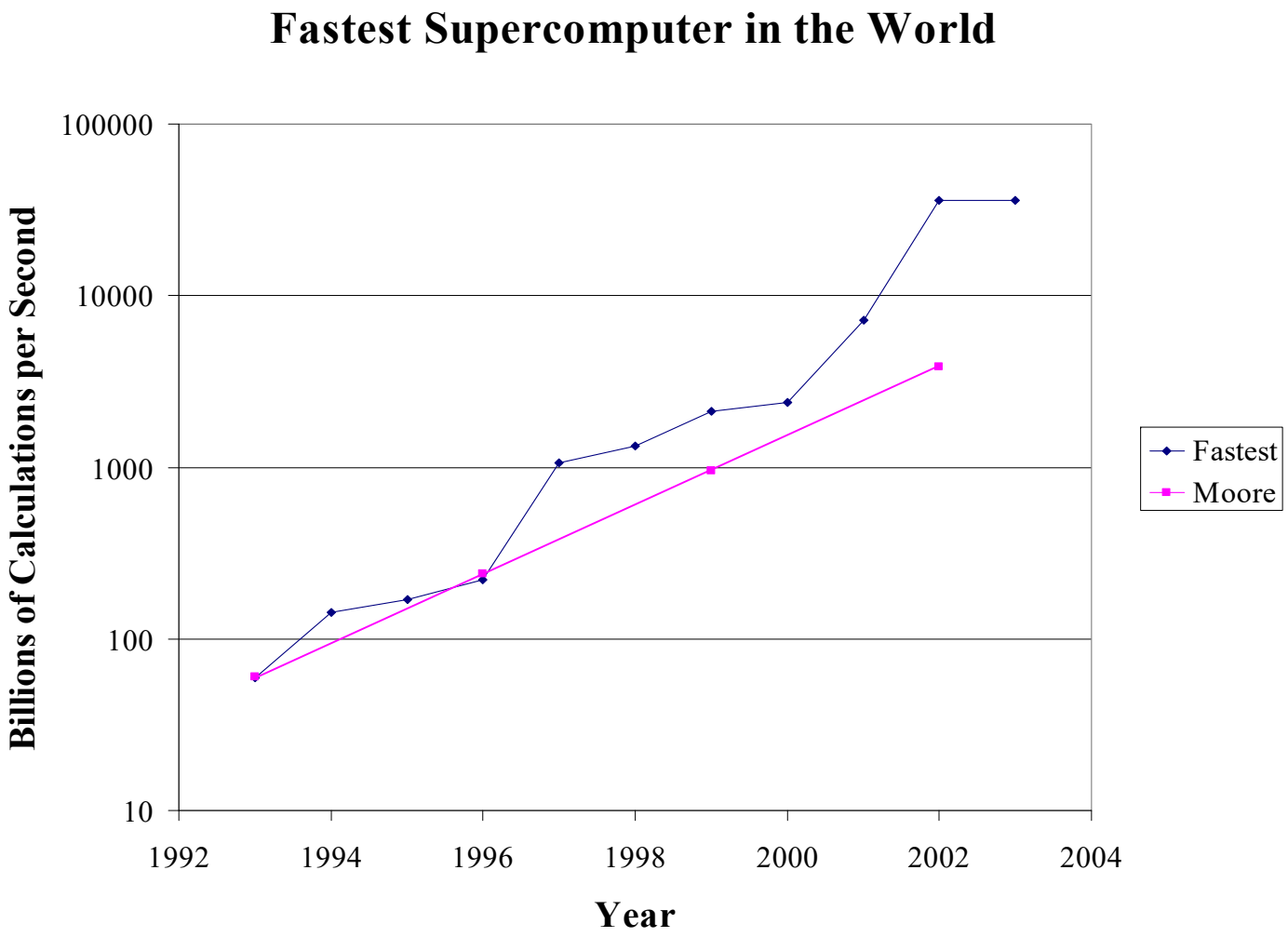
Supercomputer

- class of fastest computers currently available
- national research resource or multi-national company
- cost ~ £1m+

Performance of Supercomputing



Fastest Supercomputer



History (-1969)

Year	Name	Peak speed	Location
1943	Colossus	5000 char/sec	Bletchley Park, England
1945	Manchester Mark I	500 inst/sec	Manchester, England
1950	MIT Whirlwind	20 kIPS	MIT, USA
1956	IBM 704	20 kIPS 12 kFLOPS	
1959	IBM 7090	210 kFLOPS	USAF, USA
1960	LARC	500 kFLOPS (2 CPUs)	Lawrence Livermore Lab. USA
1961	IBM 7030	1.2 MIPS 600 kFLOPS	Los Alamos Lab. USA
1965	CDC 6600	10 MIPS 3 MFLOPS	Lawrence Livermore Lab. USA
1969	CDC 7600	36 MFLOPS	Lawrence Livermore Lab. USA

The Early Days - 1943

- Colossus was the first *programmable, digital, electronic* computer
  - Used *vacuum valves*
  - Designed for a single task – code breaking – for which it was only overtaken by general purpose PC chips in mid-1990s!
  - Top secret => little impact on subsequent computer designs

The Early Days - 1945

- Manchester Mark I is significant as it was the first machine to use *index registers* for modifying base addresses
  - Had a 40-bit *accumulator* and 20-bit *address registers* and could perform 40-bit serial arithmetic, with hardware add, subtract and multiply and logical instructions
  - Used two *Williams tubes* (CRTs) for memory – based upon charge measurement at each pixel (lit or unlit). Each tube could store 64 rows of 40 points = 32 “words” = 2 “pages”
  - Also used two *magnetic drums* for permanent storage, each of which could store 64 pages

The Early Days - 1950

- MIT Whirlwind was the first computer that operated in *real time*, used video displays for output and was the first digital flight simulator!
  - All previous computers ran in *batch mode*, i.e. a series of paper tapes/cards were set up as input in advance, fed into the computer to calculate and print results.
  - For simulating an aircraft control panel, Whirlwind need to operate continually on an ever-changing series of inputs => need high-speed stored-program computer.
  - Original design was too slow due to the Williams tubes and so *core memory* (ferrite rings that store data in polarity of magnetic field) was created => doubled speed => design successfully mass-produced by IBM

The Early Days - 1956

- IBM (originated with mechanical analogue calculators) used the core memory developed for the Whirlwind to make the IBM 704, which was also the first mass-produced computer to have *floating-point* hardware.
- The 709 improved the 704 by adding *overlapped I/O*, *indirect addressing* and *decimal instructions*.
- The 7090 improved the 709 by the use of *transistors* and not valves

IBM 704

The Early Days - 1960

- The LARC was the first true supercomputer – it was designed for *multiprocessing*, with 2 CPUS and a separate I/O processor.
  - Used 48 bits per word with a special form of decimal arithmetic => 11 digit signed numbers. Had 26 general purpose registers with an access time of 1 microsecond.
  - The I/O processor controlled 12 magnetic drums, 4 tape drives, a printer and a punched card reader.
  - Had 8 banks of core memory (20000 words) with an access time of 8 microseconds and a cycle time of 4 microseconds.

The Early Days - 1961

- IBM feared the success of the LARC so designed the 7030 to beat it.
  - Initially designed to be 100*x* faster than the 7090, it was only 30*x* faster once built. An embarrassment for IBM => big price drops and in the end only 9 were sold (but only 2 LARCs were built so not all bad!)
  - Many of the ideas developed for the 7030 were used elsewhere, e.g. *multiprogramming*, *memory protection*, *generalized interrupts*, the 8-bit *byte*, *instruction pipelining*, *prefetch* and *decoding*, and *memory interleaving* were used in many later supercomputer designs.
  - Ideas also used in modern commodity CPUs!

The Early Days - 1965

- CDC (Control Data Corporation) employed Seymour Cray to design the CDC 6600 which was 10*x* faster than any other computer when built.
  - First ever *RISC* system – simple instruction set – which simplified timing within the CPU and allowed for *instruction pipelining* leading to higher throughput and a higher clock speed, 10 MHz.
  - Used *logical address translation* to map addresses in user programs and restrict to using only a portion of contiguous core memory. Hence user program can be moved around in core memory by the operating system.
  - System contained 10 other “Peripheral Processors” to handle I/O and run the operating system.

CDC 6600

Vector Years (before 1989)

Year	Name	Peak speed	Location
1974	CDC Star-100	100 MFLOPS (vector) ~2 MFLOPS (scalar)	Lawrence Livermore Lab. USA
1975	Cray-1	80 MFLOPS (vector) 72 MFLOPS (scalar)	Los Alamos Lab. USA
1981	CDC Cyber-205	400 MFLOPS (vector) peak, avg much lower	
1983	Cray X-MP	500 MFLOPS (4 CPUs)	Los Alamos Lab. USA
1985	Cray-2	1.95 GFLOPS (4 CPUs) 3.9 GFLOPS (8 CPUs)	Lawrence Livermore Lab. USA
1989	ETA-10G	10.3 GFLOPS (vector) peak, avg much lower (8 CPUs)	
1990	Fujitsu Numerical Wind Tunnel	236 GFLOPS	National Aerospace Lab, Japan

The Vector Years 1974

- The CDC Star-100 was one of the first machines to use a *vector processor*
  - Used “deep” pipelines (25 vs. 8 on 7600) which need to be "filled" with data constantly and had high setup cost. The vector pipeline only broke even with >50 data points in each set.
  - But the number of algorithms that can be effectively vectorised is very low and need careful coding otherwise the high vector setup cost dominates.
  - And the basic scalar performance had been sacrificed in order to improve vector performance => machine was generally considered a failure.
  - Today almost all high-performance commodity CPU designs include vector processing instructions, e.g. SIMD.

The Vector Years 1975

- Seymour Cray left CDC to form Cray Research to make the Cray-1.
  - A vector processor without compromising the scalar performance using *vector registers* not pipelined memory operations
  - Uses ECL transistors
  - No wires more than 4´ long
  - 8 MB RAM and 80 MHz clock speed
  - Cost $5-$8m, with ~80 sold worldwide
  - Ships with Cray OS, Cray Assembler and Cray FORTRAN – the world's first auto-vectorising FORTRAN compiler

Cray 1

The Years Vector 1081

- CDC Cyber-205
  - CDC put right the mistakes made with the Star-100
  - 1-4 separate vector units
  - Rarely got anywhere near peak speed except with hand-crafted assembly code
  - Used semiconductor memory and *virtual memory* concept

The Vector Years 1983

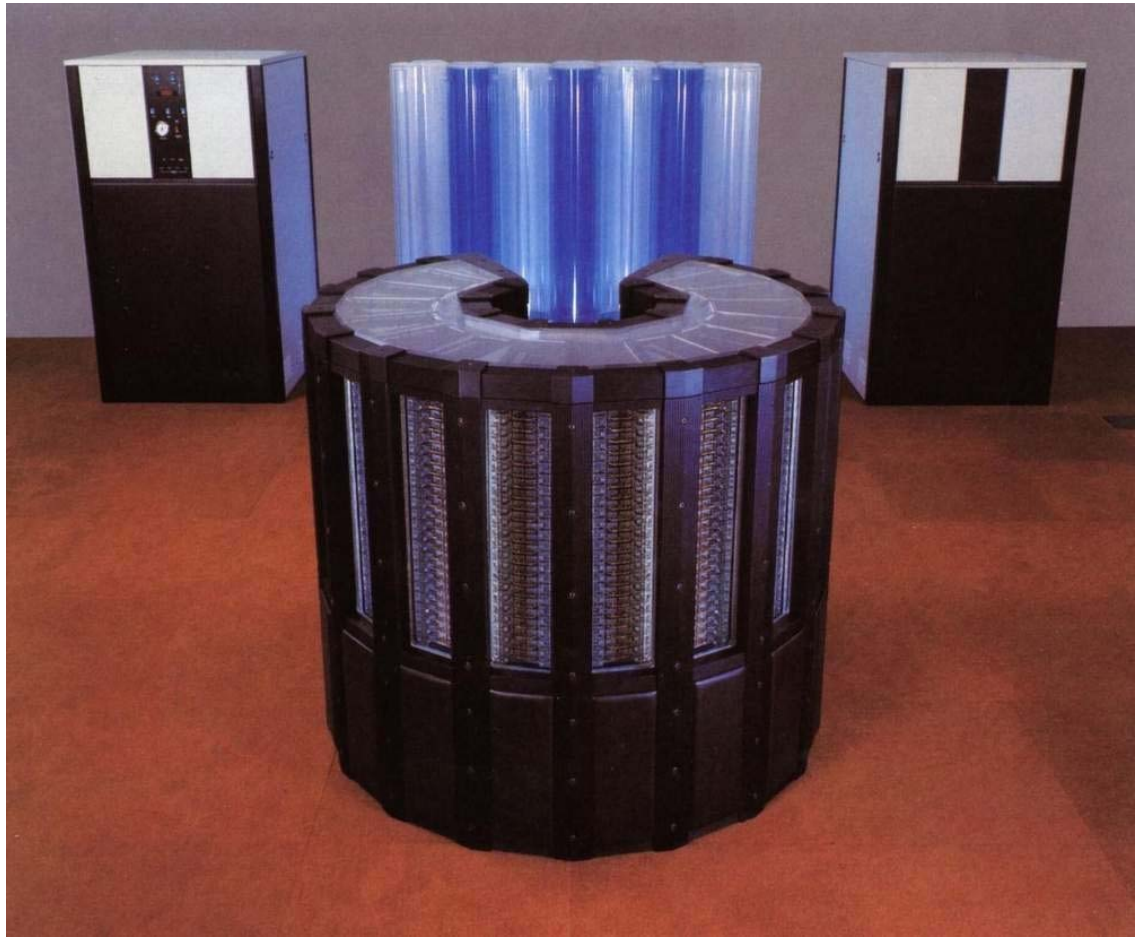
- Cray X-MP
  - A parallel (1-4) vector processor machine with 120 MHz clock speed for ~125 MFLOPS/ CPU with 8-128 MB of RAM main memory
  - Better chaining support, parallel arithmetic pipelines and *shared memory access* with multiple pipelines per processor.
  - Switched from Cray OS to UniCOS (a UNIX variant) in 1984
  - Typical cost ~$15m plus disks!

Cray XMP

The Vector Years: 1985

- Cray-2
  - A completely new, compact 4-8 processor design
  - Had 512 MB to 4 GB of main memory but with higher memory latency than X-MP
  - Hence X-MP faster than Cray-2 on certain problems – impact of memory architecture on compute speed
- Cray Y-MP introduced in 1988 – an evolution of the X-MP with up to 16 processors (new type).

Cray 2



Cray YMP

The Years Vector 1980

- ETA-10G
  - Spin-off company from CDC due to competition from Cray, with only one product – the ETA-10.
  - Compatible with CDC Cyber-205, including pipelined memory not vector registers.
  - Shared memory multiprocessor (up to 8) with up to 32MB of private memory/CPU plus common access to up to 2GB of shared memory.
  - 2 variants – one with liquid nitrogen cooling and the other with air cooling for CMOS components
  - 7 liq-N2 and 27 air-cooled units sold
  - A failure - remaining units given to high schools!

The Years Vector 1990

- Fujitsu Numerical Wind Tunnel
  - Another vector parallel architecture with advanced Ga-As CPUs for lowest gate delay
  - Each CPU had four independent pipelines with a peak speed of 1.7 GFLOPS/CPU and 256 MB main memory.
  - Had sustained performance ~100 GFLOPS for CFD codes c.f. peak=236 GFLOPS!

Supercomputing since 90's

Year	Name	Peak speed	Location
1995	Intel ASCI Red	2.15 TFLOPS	Sandia National Lab. USA
2000	IBM ASCI White	7.226 TFLOPS	Lawrence Livermore Lab. USA
2002	Earth Simulator	35.86 TFLOPS	Yokohama Institute for Earth Sciences, Japan
2005	IBM ASCI Blue Gene	70 - 478 TFLOPS	Lawrence Livermore Lab. USA
2008	IBM Roadrunner	1.105 PFLOPS	Los Alamos Lab. USA
2009	Cray Jaguar	1.75 PFLOPS	Oak Ridge Lab. USA
2010	Tianhe-1A	2.57 PFLOPS	National Supercomputer Centre, Tianjin, China
2011	Fujitsu K computer	8.2 – 10.5 PFLOPS	RIKEN Advanced Institute for Computational Science, Japan
2012	IBM Sequoia	20.1 PFLOPS	Lawrence Livermore Lab. USA
2013	Tianhe-2	54.9 PFLOPS	National Super Computer Center in Guangzhou, China
2015	Sunway TaihuLight	125.4 PFLOPS	National Supercomputing Center in Wuxi, China

The Modern Era-1995

- Intel ASCI-Red
  - Developed under the Accelerated Strategic Computing Initiative (ASCI) of the DoE and NNSA (National Nuclear Security Administration) to build nuclear weapon simulators following moratorium on nuclear weapon testing.
  - Used commodity components for low-cost
  - Designed to be very scalable
  - A *massively-parallel processing* machine consisting of 38*x*32*x*2 CPUs (Pentium II Xeons) with 4510 compute nodes, 1212 GB of distributed RAM and 12.5 TB of disk storage.
  - Used *MIMD* (multiple instruction, multiple data) paradigm
  - See http://www.sandia.gov/ASCI/Red/RedFacts.htm

ASCI Red

The Modern Era - 2000

- IBM ASCI-White
  - A cluster computer based upon the commercial IBM RS/6000 SP computer
  - 512 machines, each containing 16 CPUs, in the cluster for a total of 6 TB of RAM and 160 TB of disk
  - Also had 28 node “Ice” and the 68 node “Frost”
  - Consumed 3 MW electricity to run and additional 3 MW to cool
  - Ran AIX (UNIX variant)
  - Cost $110 million
  - See https://www.llnl.gov/str/Seager.html

ASCI White

The Modern Era - 2002

- Earth Simulator
  - A cluster computer based upon the NEC SX-6, which comprised 8 vector processors and 16 GB RAM.
  - Contained 640 nodes for a total of 5120 CPU and 10 TB RAM
  - Ran NEC SUPER-UX (UNIX variant)
  - Designed for simulations of global climate in both the atmosphere and ocean with a max. resolution of 10 km
  - Codes written using HPF with optimised NEC compiler
  - Cost ¥7.2b (~£36m)
  - See http://www.jamstec.go.jp/es/en/index.html

Earth Simulator

The Modern Era – 2005

- IBM ASCI-Blue Gene
  - Had 65,536 power4 CPUs (dual core) with 3 integrated networks (different underlying topologies)
  - A *constellation* computer – made of an integrated collection of smaller parallel nodes
  - Designed to be 10*x* faster than the Earth Simulator
  - Hierarchical design – viable system in wide range of sizes from single board upwards
  - Cost $100m
  - See https://asc.llnl.gov/computing_resources/bluegenel

Blue Gene/L

The Modern Era - 2008

- IBM Roadrunner
  - Had 6,480 dual-core Opteron CPUs to handle O/S, interconnect, scheduling etc. and 12,960 PowerXCell 8i CPUS – one per Opteron core – to handle computation
  - *An Opteron cluster with Cell accelerators*
  - TriBlade design – 2 dual Opterons with 16 GB and 4 PowerXCell 8i also with 16 GB
  - 3 TriBlades per chassis; 180 TriBlades per Connected Unit; 18 Connected Units in total
  - A unique hybrid architecture that required all software to be specially written
  - A BIG challenge to program
  - Cost $133m and 2.35 MW to operate
  - See http://www.lanl.gov/roadrunner

IBM Roadrunner

The Modern Era - 2009

- Cray Jaguar
  - Conventional architecture
  - $20m upgrade in 2009 from quad-core AMD-based XT4 to Cray XT5 and AMD hex-core CPUs – now got 224,256 cores!
  - Requires 6.9MW to operate
  - Another US Govt lab supercomputer
  - See http://www.nccs.gov/computing-resources/jaguar for more details

Cray Jaguar

The Modern Era - 2010

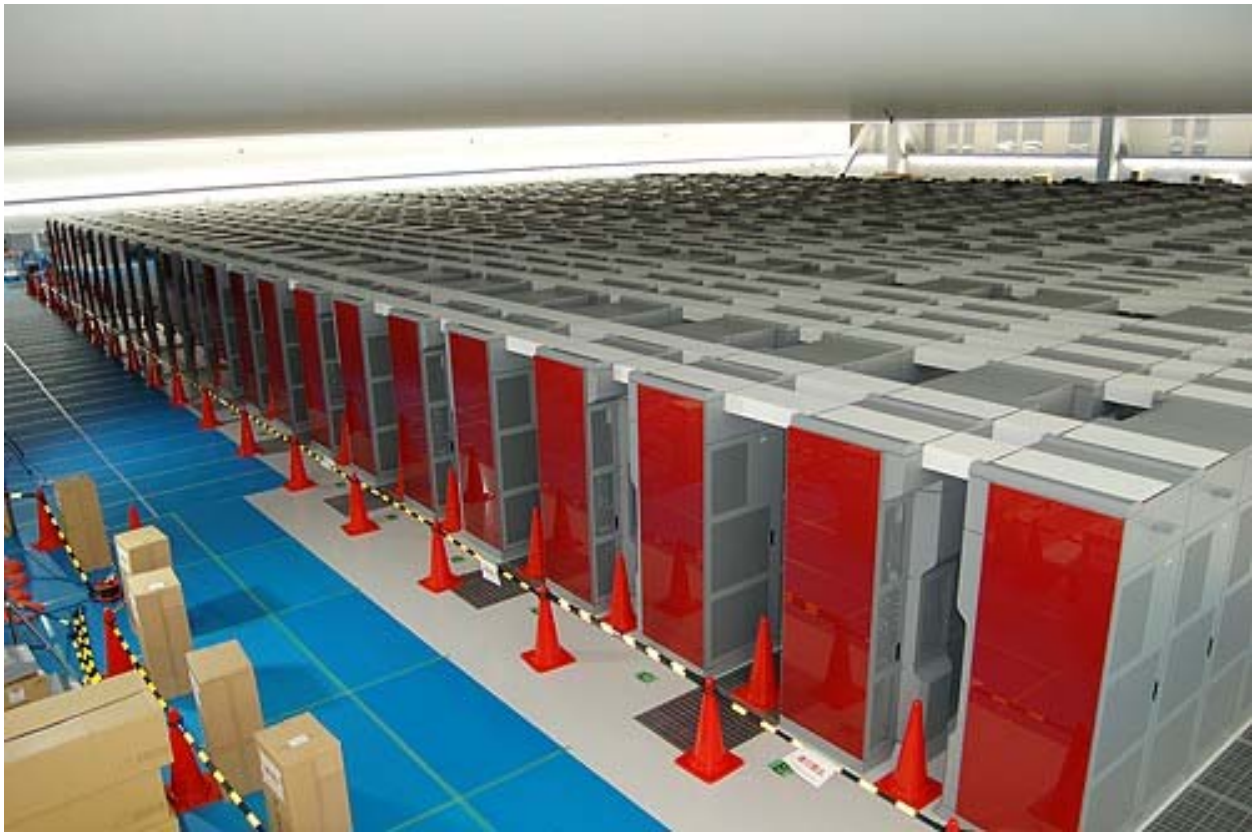
- Tianhe-1A
  - First Chinese supercomputer to top 500!
  - Hybrid design – mix of CPU and GPU
    - Implications for usage beyond LINPACK?
    - See later lectures for GPGPU programming
  - Blade system with 14,336 Intel Xeon X5670 CPUs and 7,168 Nvidia Tesla M2050 GPGPUs connected by Infiniband
  - Cost $88m and requires 4MW to operate
  - See http://www.nscc-tj.gov.cn/en for more details

Tianhe-1A

The Modern Era – 2011

- Fujitsu K
  - Conventional design – no accelerators
  - SPARC64 VIIIfx 2.0 GHz 8-core CPUs
    - June 2011= 68,544 CPUs in 672 racks
    - Nov 2011 = 88,128 CPUs for 10.5 PFLOPs
  - 3D torus topology with no switches
  - Requires 12.6 MW to operate
    - 2nd most efficient system on Top500 list!
    - 1 kW for 800 GFLOPs
    - Cost $1.0bn plus $10m/year to run ...
  - See http://www.nsc.riken.jp/project-eng.html

Fujitsu K



Official press release ~~photo~~ image!

The Modern Era - 2012

- IBM BlueGene/Q
- 1,572,864 cores in 98,304 IBM Power CPUs
- Very energy efficient = only 7.9 MW
  - 2066 GFLOP/ kW
- Achieved 16.32 PFLOPs vs peak=20 PFLOP
- Design can scale to 100 PFLOP …
- Cost $97m
- See https://asc.llnl.gov/computing_resources/sequoia/

IBM Sequoia

The Modern Era–2013

- Tianhe-2
- Intel Xeon Ivy Bridge + Intel Xeon-Phi
  - 3,120,000 cores in 16,000 nodes (2 Ivy Bridge + 3 Xeon-Phi per node – conventional coding)
- Needs lot of power = 17.8 MW
  - 3084 GFLOP/ kW
- LAPACK only 62% of peak
- Cost $390m
- See http://www.nudt.edu.cn/

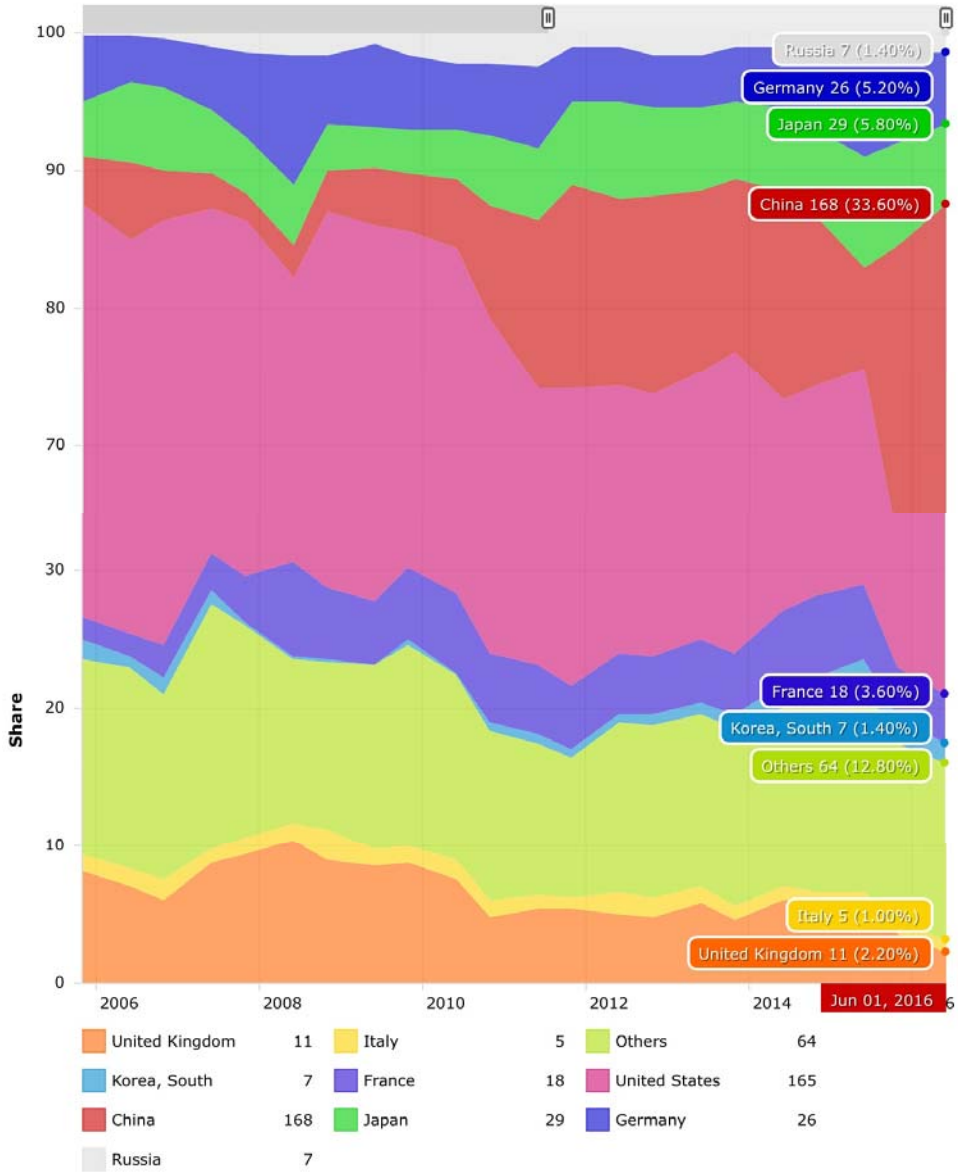
Tianhe-2

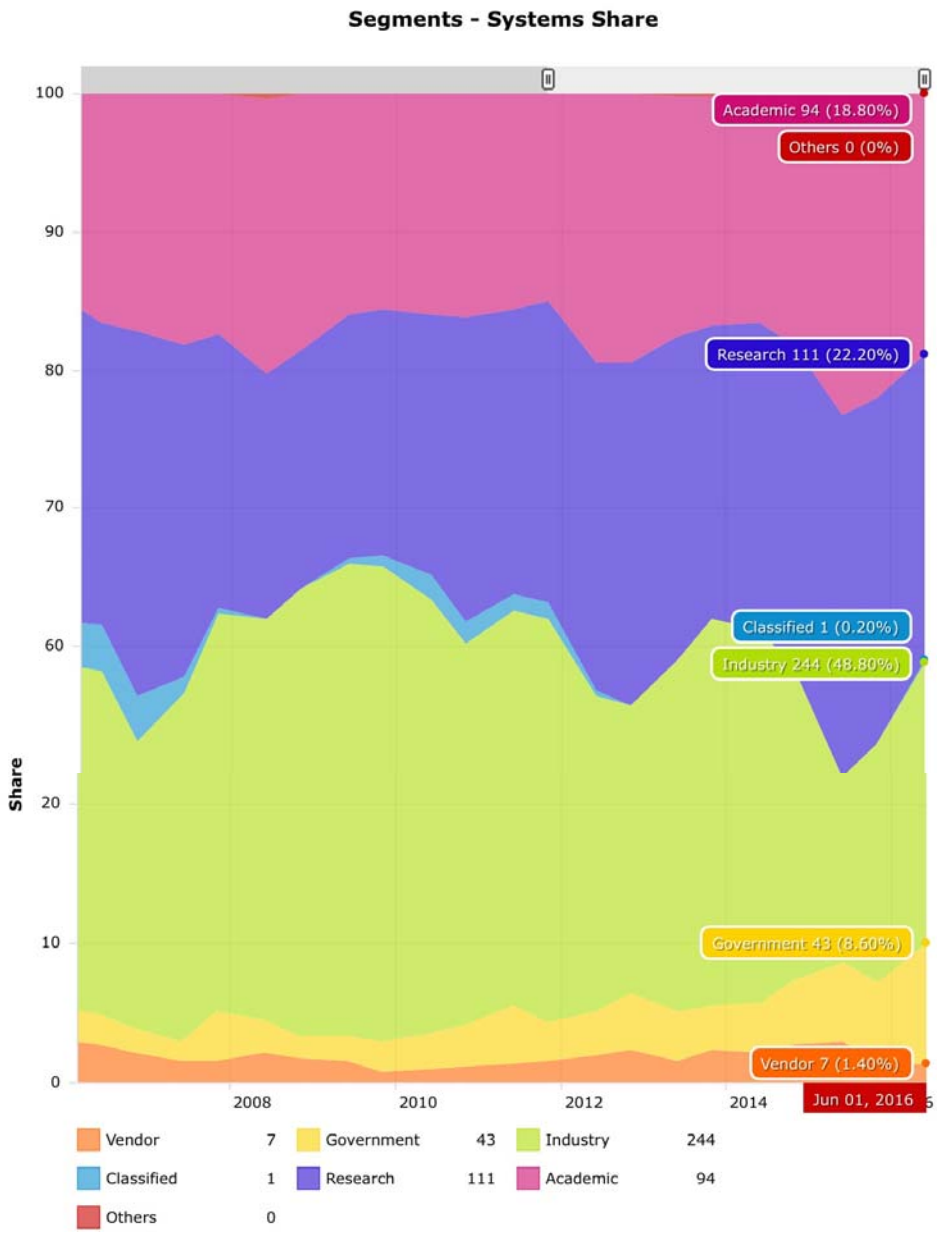
Current Best –Sunway TaihuLight

- Made entirely out of Chinese chips!
- 40,960 nodes with 10,649,600 cores
- Twice as fast and three times as efficient as Tianhe-2
- LINPACK 93 TFLOPs vs Peak=125 TFLOPs ie 75% peak
- Peak power consumption = 15.37 MW
  - 6060 Gflops/kW
  - Also very high in Green500 table

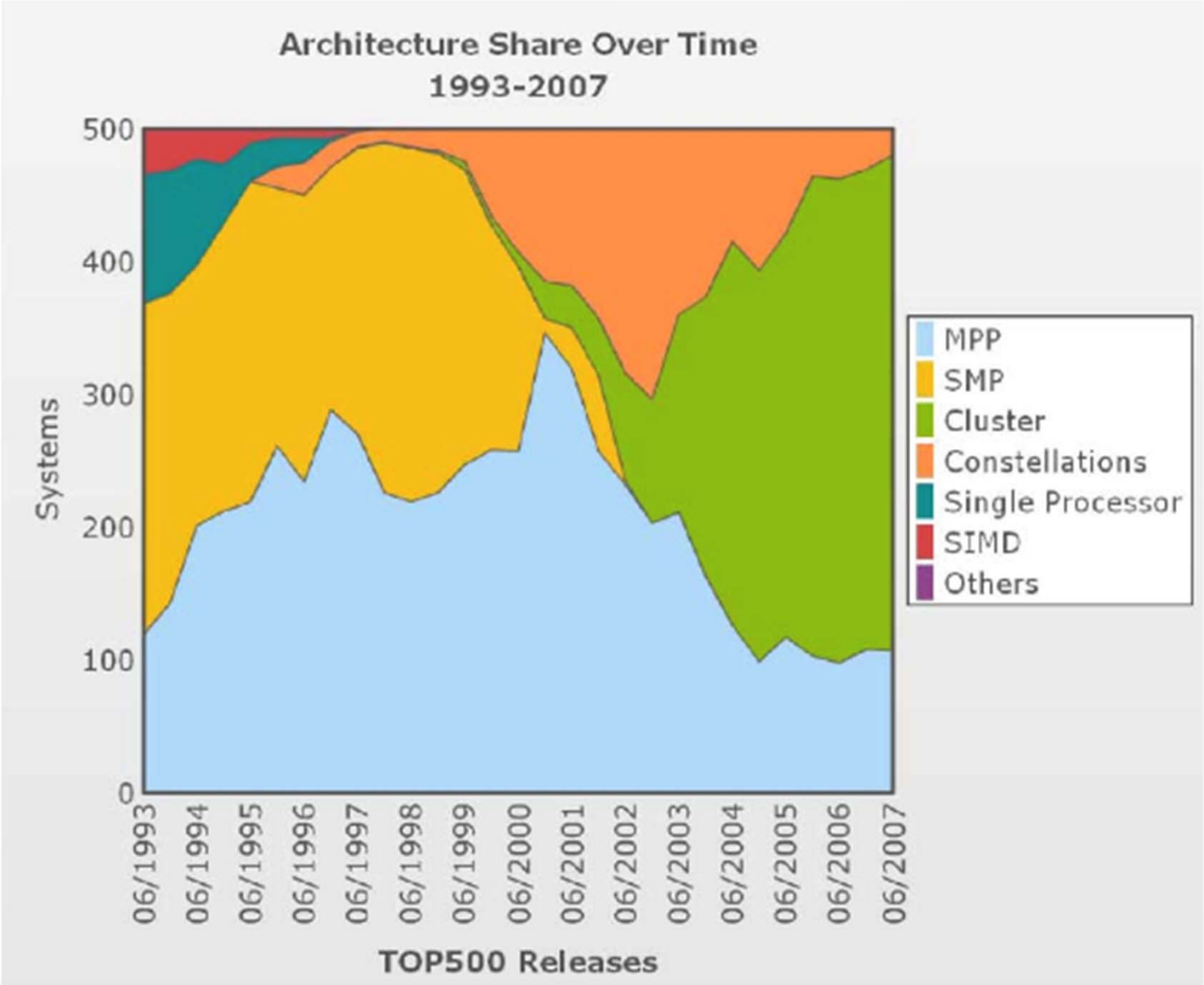
Sunway TaihuLight

Countries - Systems Share

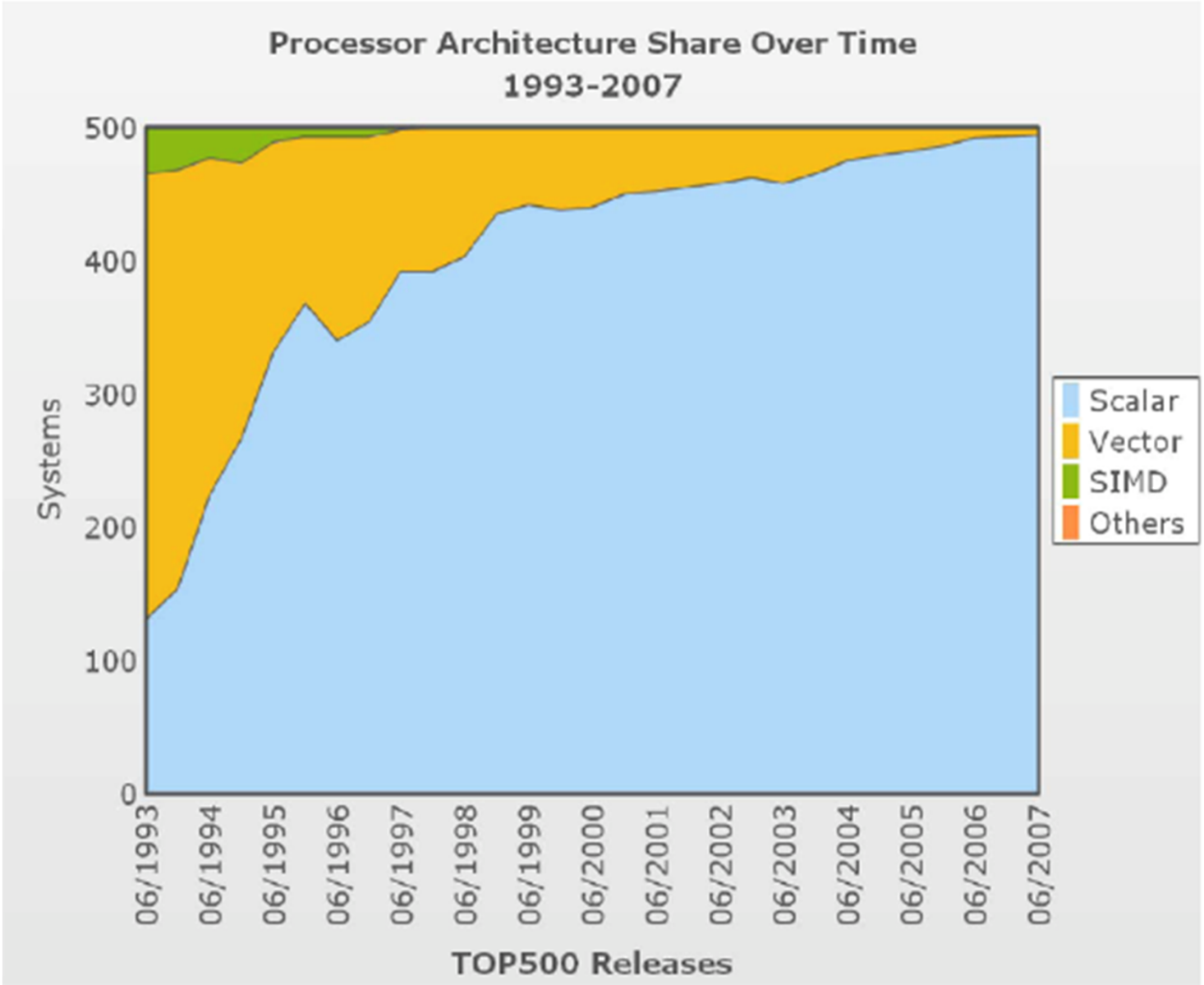



Segments - Systems Share
Share
100
90
80
70
60
20
10
0
2008
2010
2012
2014
Jun 01, 2016
Academic 94 (18.80%)
Others 0 (0%)
Research 111 (22.20%)
Classified 1 (0.20%)
Industry 244 (48.80%)
Government 43 (8.60%)
Vendor 7 (1.40%)
Vendor 7
Government 43
Industry 244
Classified 1
Research 111
Academic 94
Others 0

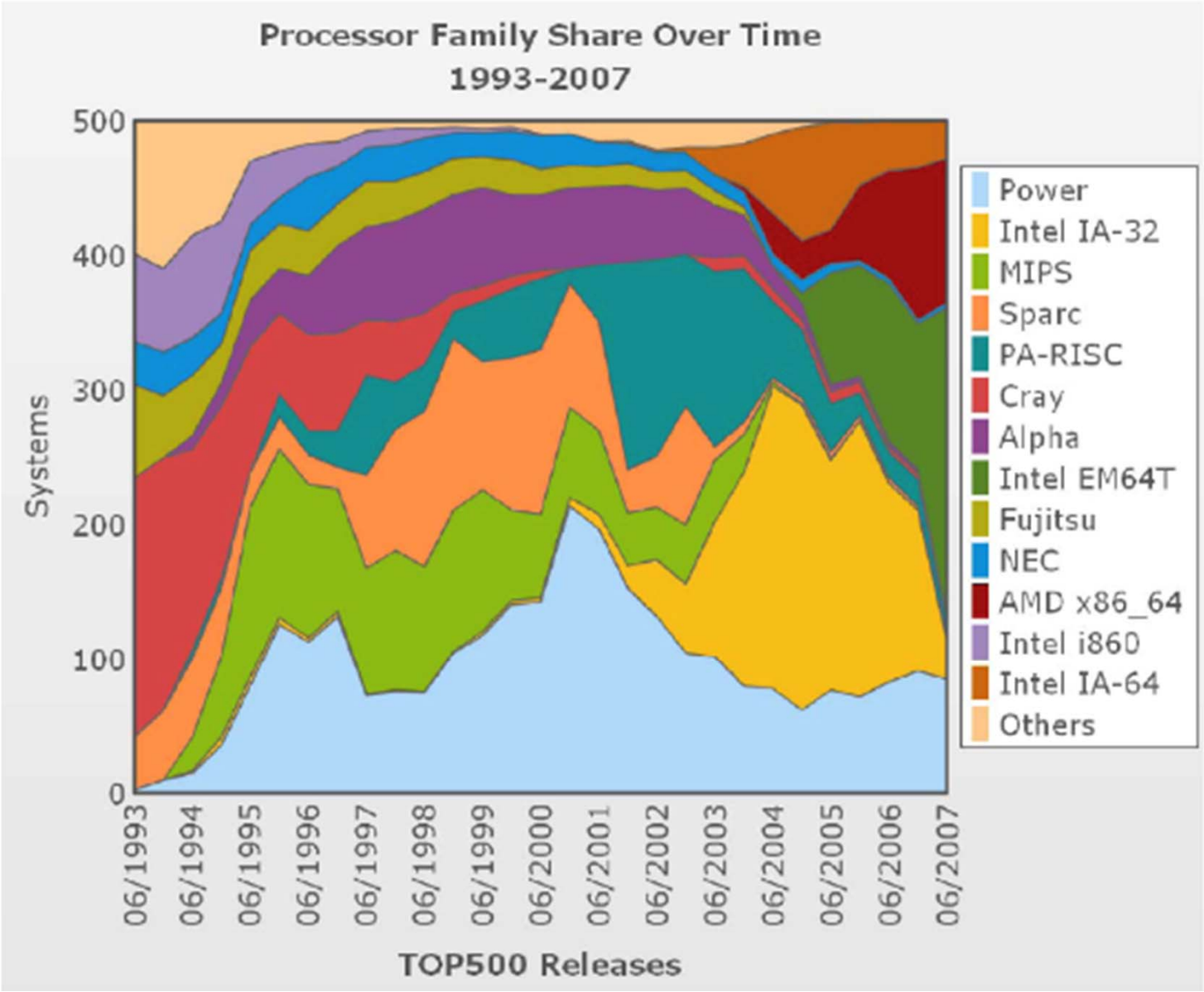
Top 500 Trends – System Architecture



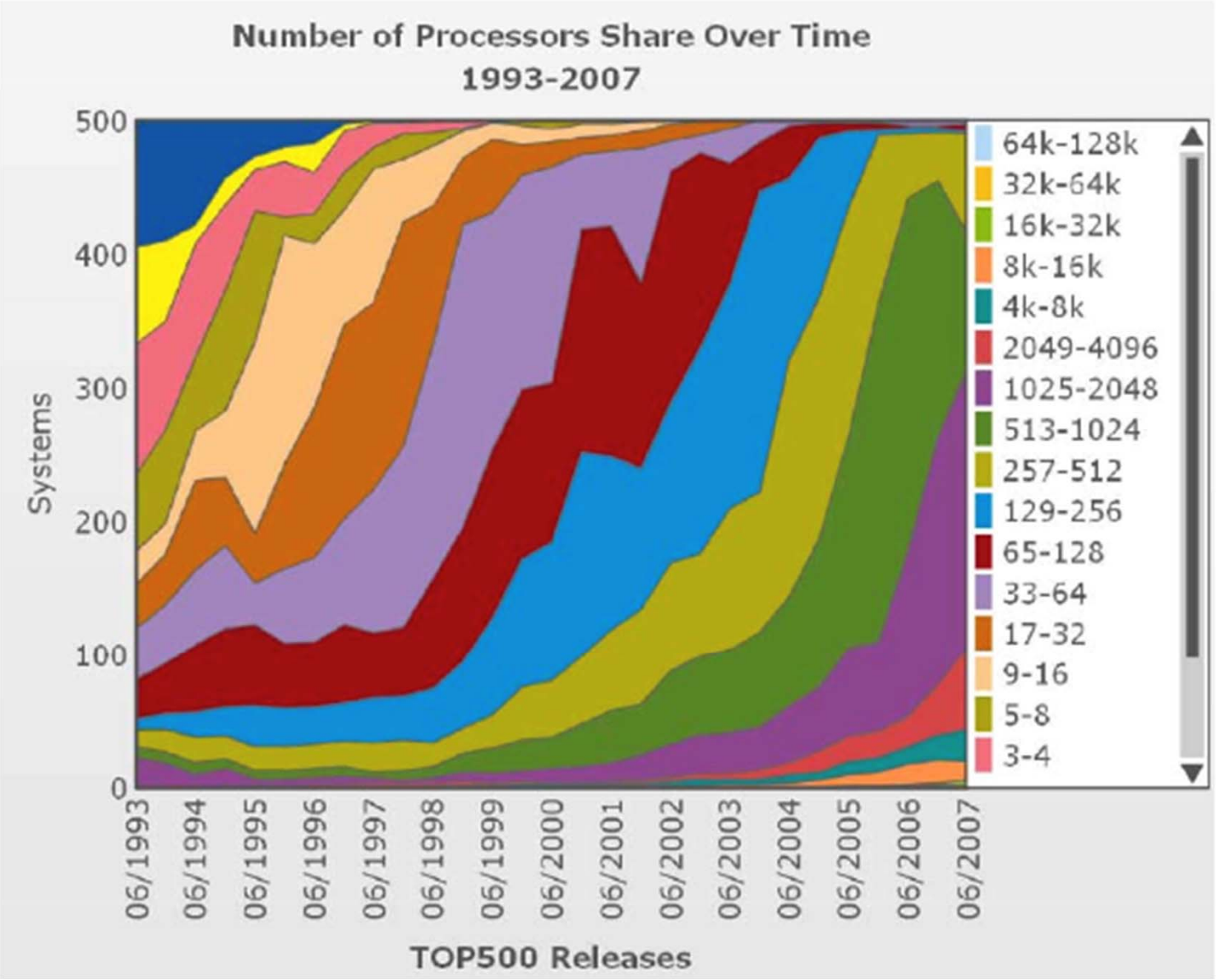
Top 500 Trends – Processor Architecture



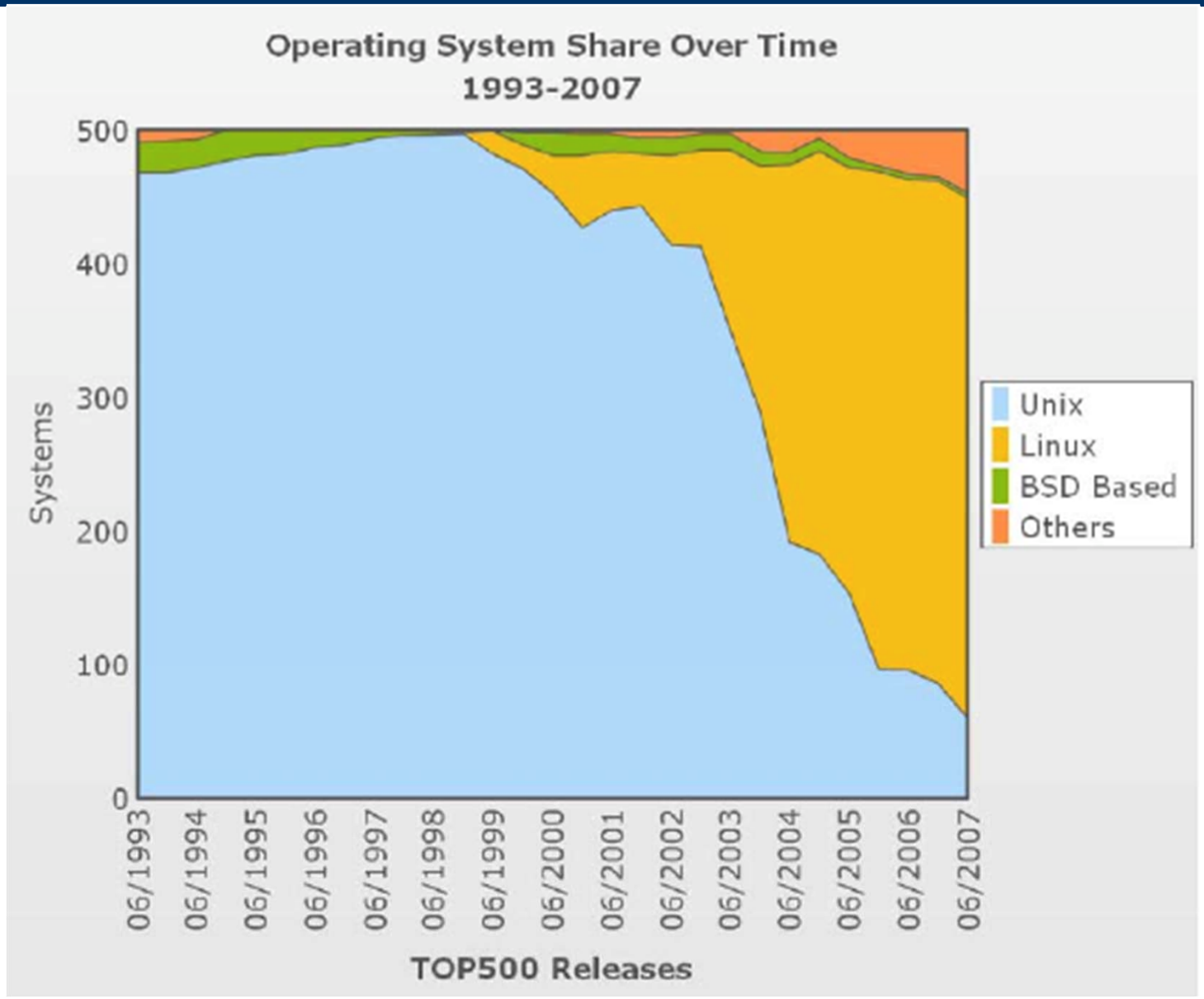
Top 500 Trends – Processor Type



Top 500 Trends – # of Processors



Top 500 Trends – OS



Trends

- Vector processing has come, gone and is now back
- Parallel processing is here to stay
- Hybrid architectures, with CPU + GPGPU or Xeon-Phi accelerator, becoming more common
- Cluster computing gives high performance with marginal cost using commodity components – big shift away from niche/proprietary components since 1990s
- Difficult to exploit many processors in a single task – need parallel programming concepts
- Many hardware & software concepts developed for supercomputers are now being used in latest commodity high-performance CPUs (Intel, AMD)
- Electrical power requirement now non-trivial – rise of "green computing"

Commodity Systems Satisfy Most HPC Needs

- Good parallel performance can be achieved by clusters of commodity processors connected by commodity switches and switch interfaces, e.g., ASC Q.

- For problems with good locality (e.g., bioinformatics) such systems provide better time-to-solution than customized systems at any cost level.

It will be harder in the future to "ride on the coattails" of Moore's Law.

- Memory latency increases relative to processor speed (the *memory wall*): by 2020 about 800 loads and 90,000 floating-point operations would be executed while waiting for one local memory access to complete.
- Global communication latency increases and bandwidth decreases relative to processor speed: by 2020 a global bandwidth of about 0.001 word/flops and global latency equivalent to about 0.7Mflops.
- Improvement in single processor performance is slowing down; future performance improvement in commodity processors will come from increasing on-chip parallelism.
- Mean Time to Failure is growing shorter as systems grow and devices shrink.

Software Productivity is Low

- Need high-level notations that capture parallelism and locality.
- Application development environment and execution environment in HPC are less advanced and less robust than for general computing.
- Will need increasing levels of parallelism in future supercomputing.
- Custom/hybrid systems can support a simpler programming model.
  - But that potential is largely unrealized