Universidad Carlos III de Madrid
www.uc3m.es

ARCOS

COMPUTER

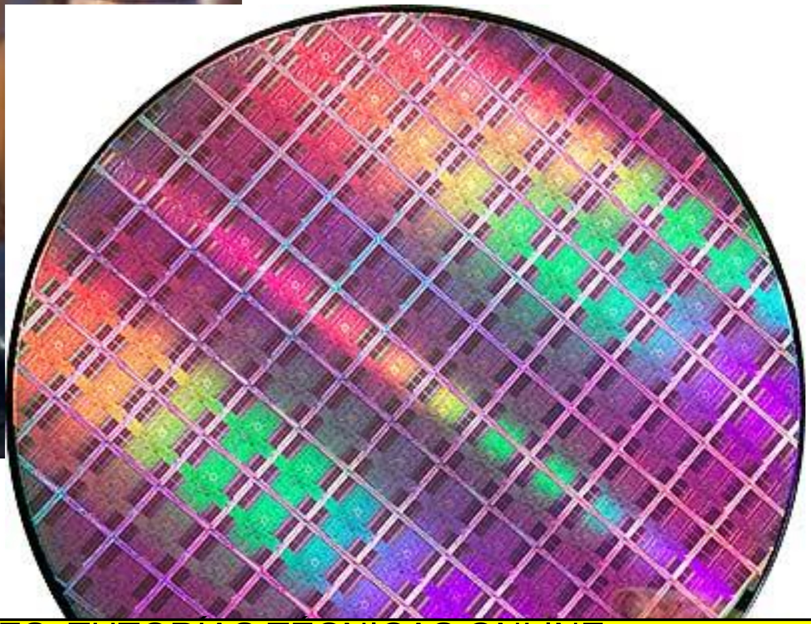
Parallel Architectures: Models and Tools

- Performance improvements:
  - Improvements in semiconductor technology
    - Feature size, clock speed
  - Improvements in computer architectures
    - Enabled by HLL compilers, UNIX
    - Lead to RISC architectures
  - Together have enabled:
    - Lightweight computers
    - Productivity-based managed/interpreted programming

Universidad Carlos III de Madrid
www.uc3m.es

Technology

ARCOS



Integrated Circuit Cost

- Integrated circuit

$$\text{Cost of integrated circuit} = \frac{\text{Cost of die} + \text{Cost of testing die} + \text{Cost of packaging and final test}}{\text{Final test yield}}$$

$$\text{Cost of die} = \frac{\text{Cost of wafer}}{\text{Dies per wafer} \times \text{Die yield}}$$

$$\text{Dies per wafer} = \frac{\pi \times (\text{Wafer diameter}/2)^2}{\text{Die area}} - \frac{\pi \times \text{Wafer diameter}}{\sqrt{2 \times \text{Die area}}}$$

- Bose-Einstein formula:

$$\text{Die yield} = \text{Wafer yield} \times 1/(1 + \text{Defects per unit area} \times \text{Die area})^N$$

- Defects per unit area = 0.016-0.057 defects per square cm (2010)

Example

- Wafer with a diameter of 30 cm.
  - Dies of 1.5 cm side.
    - Dies per wafer: 269.
  - Dies of 1 cm side
    - Dies per wafer: 640.

Trends in Technology

- Integrated circuit technology
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year

- DRAM capacity: 25-40%/year (slowing)

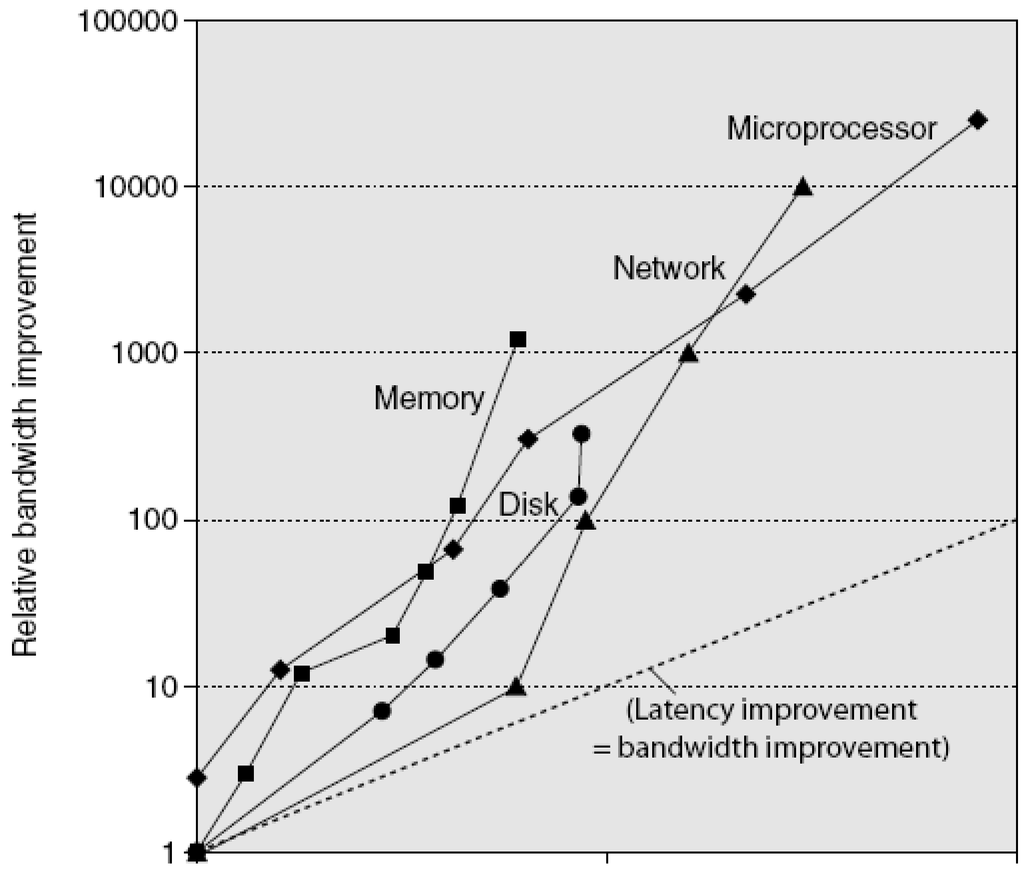
- Flash capacity: 50-60%/year
  - 15-20X cheaper/bit than DRAM

- Magnetic disk technology: 40%/year
  - 15-25X cheaper/bit than Flash

Bandwidth and Latency

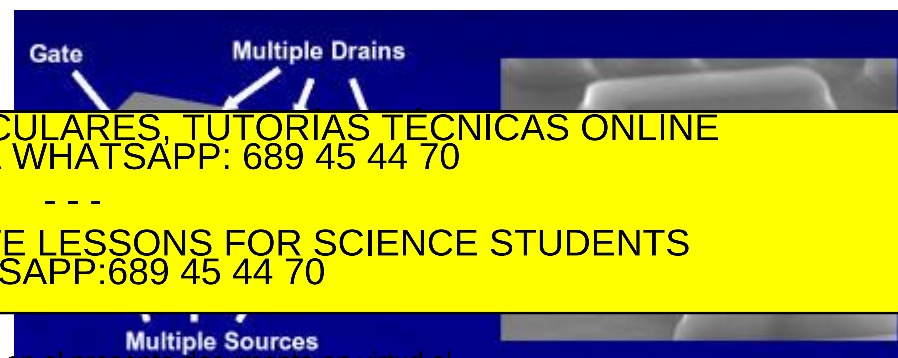
- Bandwidth or throughput
  - Total work done in a given time
  - 10,000-25,000X improvement for processors
  - 300-1200X improvement for memory and disks
- Latency or response time
  - Time between start and completion of an event
  - 30-80X improvement for processors
  - 6-8X improvement for memory and disks

Bandwidth and Latency



Transistors and Wires

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .014 microns in 2014
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically



Power and Energy concerns

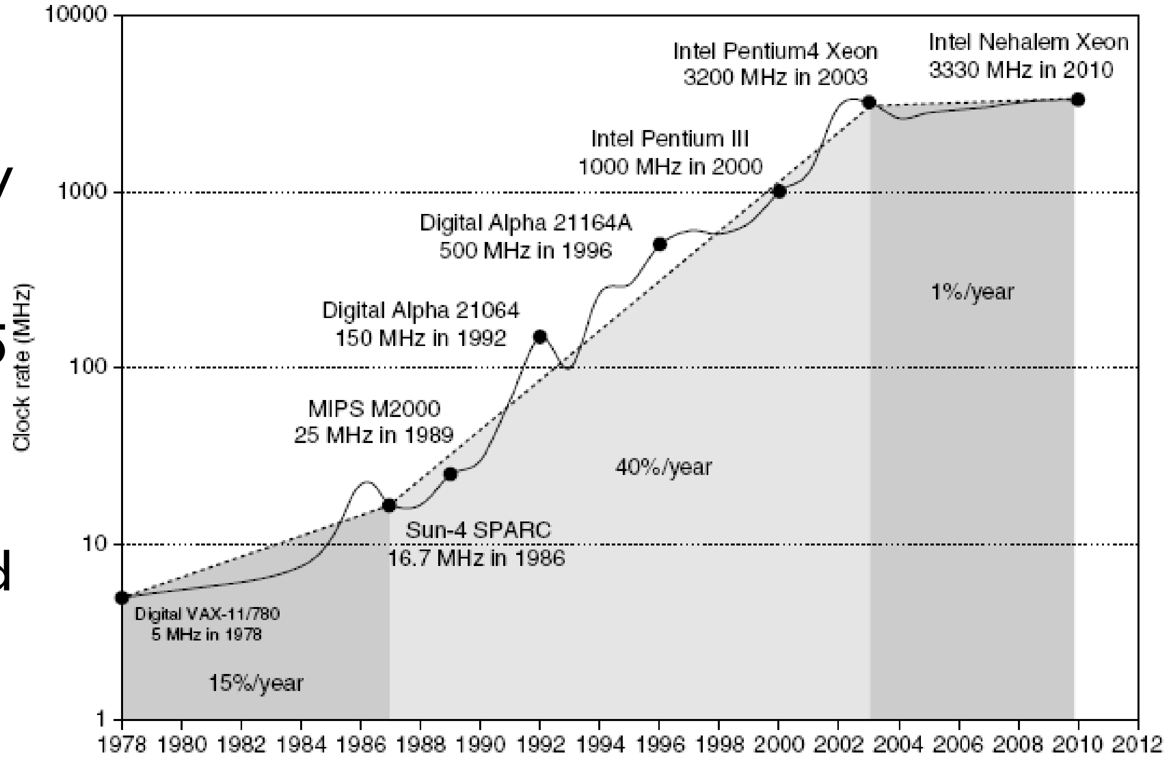
- Problem: Get power in, get power out
  - Distribute power to increasingly complex circuitry
- Thermal Design Power (TDP)
  - Characterizes sustained power consumption
  - Used as target for power supply and cooling system
  - Lower than peak power, higher than average power consumption
  - Dark silicon
- Clock rate can be reduced dynamically to limit power consumption

Dynamic Energy and Power

- Dynamic energy
  - Transistor switch from 0 -> 1 or 1 -> 0
  - ½ x Capacitive load x Voltage^2
- Dynamic power
  - ½ x Capacitive load x Voltage^2 x Frequency switched
- For a fixed task reducing clock rate reduces power, not energy
- Voltage reduces both: has dropped from 5V to 1V in

Power

- Intel 80386 consumed ~ 2 W
- 3.3 GHz Intel Core i7 consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip
- This is the limit of what can be cooled by air



Static Power

- Static power consumption
  - Due to leakage current flow

    $Power_{static} = Current_{static} \times Voltage$

  - Scales with number of transistors
  - To reduce: power gating even to inactive modules
  - Goal 2006 for leakage: 25% o total power

Reducing Power

- Techniques for reducing power:
  - Do nothing well
  - Dynamic Voltage-Frequency Scaling
  - Low power state for DRAM, disks
  - Overclocking, turning off cores

Trends in Cost

- Cost driven down by learning curve
  - Yield
- DRAM: price closely tracks cost
- Microprocessors: price depends on volume
  - Volume decrease the time needed to get down the learning curve.
  - Volume decreases cost, since it increases purchasing and manufacturing efficiency.

- Module reliability
  - Mean time to failure (MTTF)
  - Mean time to repair (MTTR)
  - Mean time between failures (MTBF) = MTTF + MTTR
  - Availability = MTTF / MTBF

Measuring Performance

- Typical performance metrics:
  - Response time
  - Throughput
- Speedup of X relative to Y
  - Execution time$_Y$ / Execution time$_X$
- Execution time
  - Wall clock time: includes all system overheads
  - CPU time: only computation time in the CPU
- Benchmarks
  - Kernels (e.g. matrix multiply)
  - Toy programs (e.g. sorting)

Benchmarks

- Embedded
  - Dhrystone .
  - EEMBC (kernels).
- Desktop:
  - SPEC2006 (interger and floating point programs).
- Servers:
  - SPECWeb, SPECSFS, SPECjbb, SPECvirt_Sc2010.
  - TPC

Metrics

- The only valid performance metric is the execution of real programs.
  - Any other metric is prone to errors.
  - Any other alternative to real programs is prone to errors.

Universidad Carlos III de Madrid
www.uc3m.es

Benchmarks: SPEC 2006

ARCOS

SPEC2006 benchmark description	SPEC2006	SPEC2000	SPEC95	SPEC92	SPEC89
	Benchmark name by SPEC generation				
GNU C compiler					gcc
Interpreted string processing			perl		espresso
Combinatorial optimization		mcf			li
Block-sorting compression		bzip2		compress	eqntott
Go game (AI)	go	vortex	go	sc	
Video compression	h264avc	gzip	ijpeg		
Games/path finding	astar	eon	m88ksim		
Search gene sequence	hmmer	twolf			
Quantum computer simulation	libquantum	vortex			
Discrete event simulation library	omnetpp	vpr			
Chess game (AI)	sjeng	crafty			
XML parsing	xalancbmk	parser			
CFD/blast waves	bwaves				fpppp
Numerical relativity	cactusADM				tomcatv
Finite element code	calculix				doduc
Differential equation solver framework	dealII				nasa7
Quantum chemistry	gamess				spice
EM solver (freq/time domain)	GemsFDTD			swim	matrix300
Scalable molecular dynamics (~NAMD)	gromacs		apsi	hydro2d	
Lattice Boltzman method (fluid/air flow)	lbm		mgrid	su2cor	
Large eddie simulation/turbulent CFD	LESlie3d	wupwise	applu	wave5	
Lattice quantum chromodynamics	milc	apply	turb3d		
Molecular dynamics	namd	galgel			
Image ray tracing	povray	mesa			
		fma3d			
		sixtrack			

Speedup

- Speedup (plus low prog. effort and resource needs)

$$Speedup\ (p) = \frac{Performance(p)}{Performance(1)}$$

- For a fixed problem:

$$Speedup\ (p) = \frac{Time(1)}{Time(p)}$$

- Take Advantage of Parallelism
  - e.g. multiple processors, disks, memory banks, pipelining, multiple functional units
- Principle of Locality
  - Reuse of data and instructions
- Focus on the Common Case
  - Amdahl's Law

$$\text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)$$

Amdahl´s law

- Suppose a fraction f of your application is not parallelizable

- 1-f : parallelizable on p processors

$$Speedup(P) = T_1 / T_p$$

$$<= T_1/(f\ T_1 + (1\text{-}f)\ T_1 /p) = 1/(f + (1\text{-}f)/p)$$

$$<= 1/f$$

Example 1

- A web server has the following ratio of the execution time:
  - Computation: 40%
  - I/O: 60%

- If we replace this computer with another that is10 times faster in computation, what is the overall speedup?

$$S = \frac{1}{0.6 + \frac{0.4}{\;}} = \frac{1}{0.64} = 1.5625 < 1.666 = 1/0.6$$

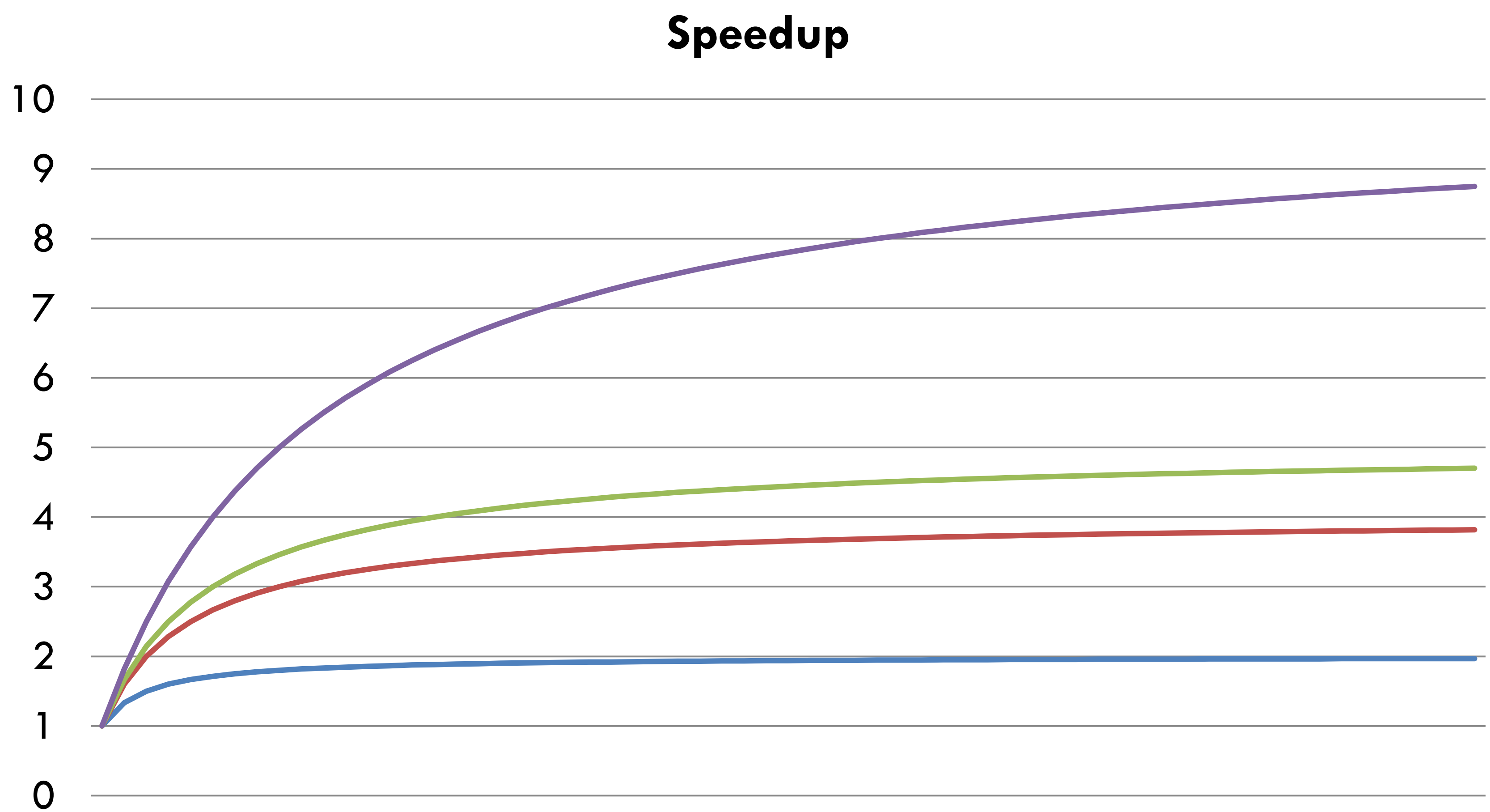
Example 2

- An application has a parallel portion that takes 50% of the execution time.

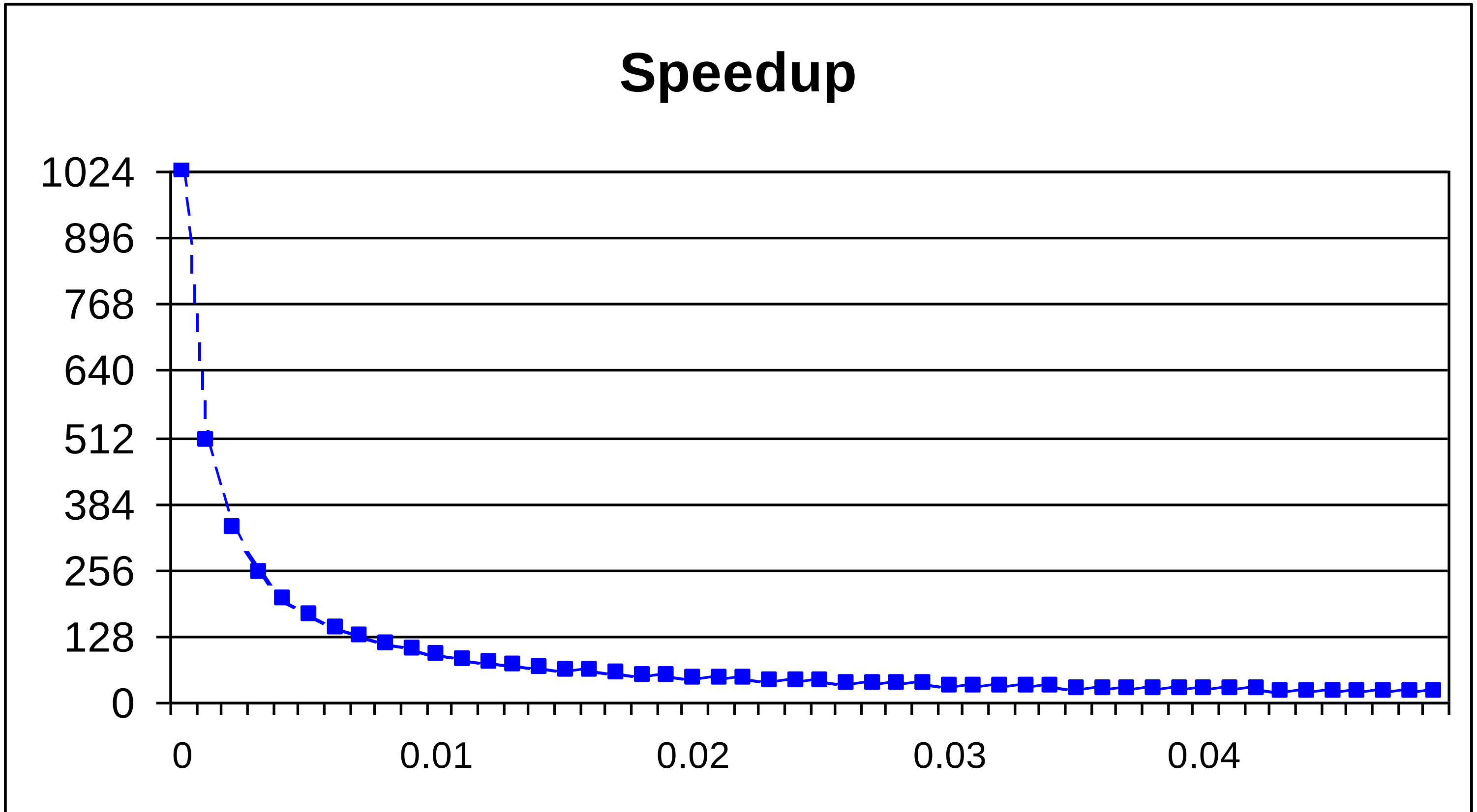
  - We execute the application in a 32-processor computer, what is the maximum speedup?

$$S = \frac{1}{0.5 + \frac{0.5}{32}} = \frac{1}{0.515625} = 1.9393$$

Example 2



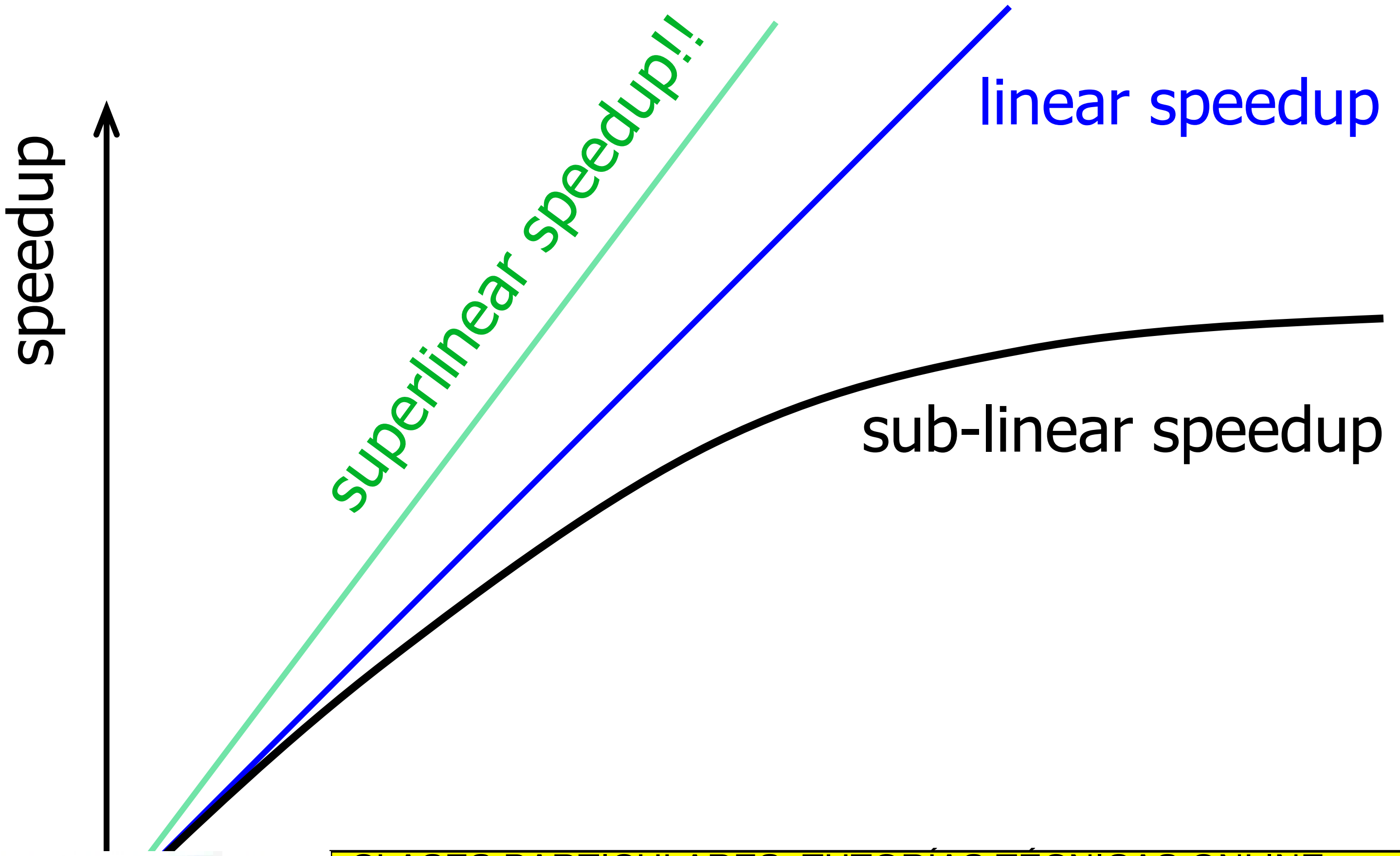
Amdahl's Law (for 1024 processors)



Amdahl´s law

- But:
  - There are many problems can be “embarrassingly” parallelized
    - Ex: image processing, differential equation solver
  - “In some cases the serial fraction does not increase with the problem size
  - Additional speedup can be achieved from additional resources (super-linear speedup due to more memory)

Speedup


speedup
superlinear speedup!!
linear speedup
sub-linear speedup

Superlinear Speedup?

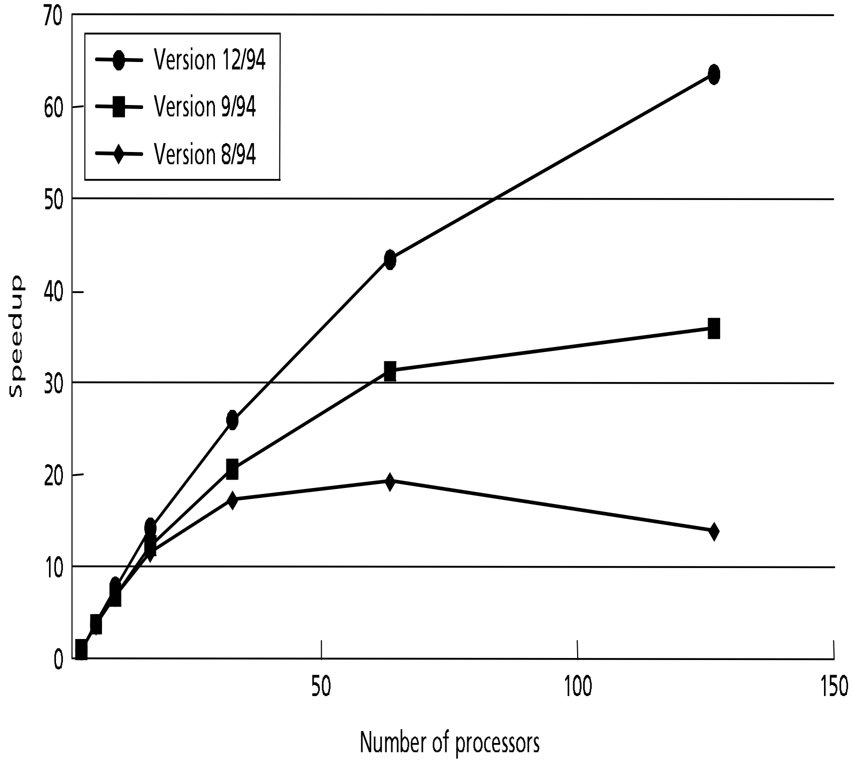
- Possible causes
- Algorithm
  - e.g., with optimization problems, throwing many processors at it increases the chances that one will “get lucky” and find the optimum fast
- Hardware
  - e.g., with many processors, it is possible that the entire application data resides in cache (vs. RAM) or in RAM (vs. Disk)

Parallel Efficiency

- $Eff_p = S_p / p$
- Typically 1, unless superlinear speedup
- Used to measure how well the processors are utilized
  - If increasing the number of process by a factor 10 increases the speedup by a factor 2, perhaps it's not worth it: efficiency drops by a factor 5

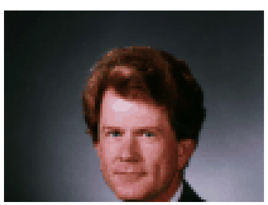
Performance Goal => Speedup

- Architect Goal
  - observe how program uses machine and improve the design to enhance performance
- Programmer Goal
  - observe how the program uses the machine and improve the implementation to enhance performance



Gustafson's law

- Amdahl's law focuses on the negative point of view of parallel processing
- However:
  - Parallel machines are used for solving large problems.
  - A sequential computer could never execute a large parallel program.
    - Memory limits.
    - Processing limits.



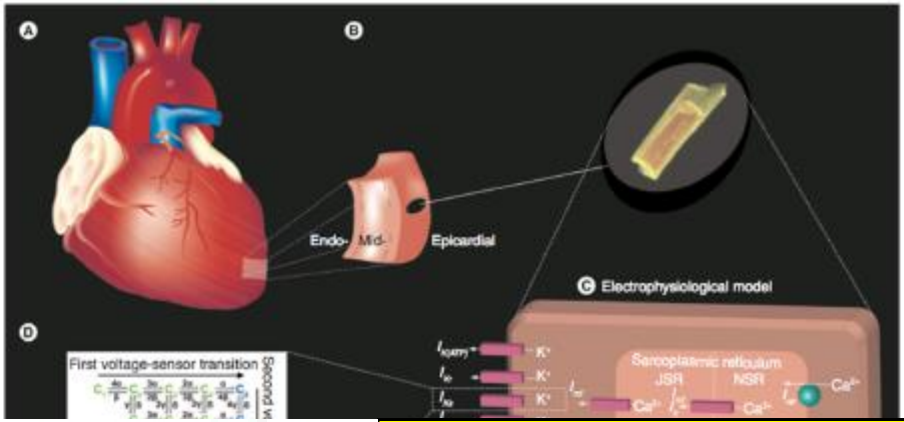
$$S = \frac{T_s}{T_p}$$

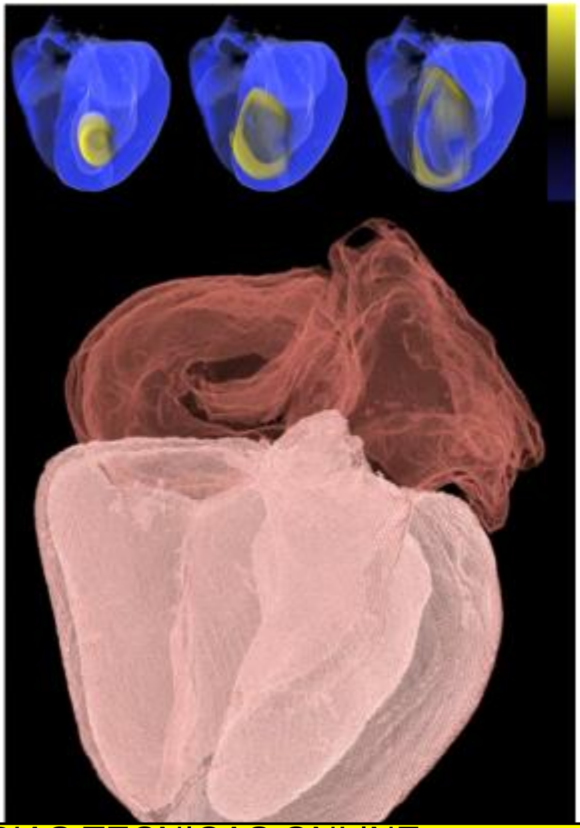
$$T_s = Time \text{ in a sequencial machine}$$

$$T_p = \text{Time in a paralle machine}$$

Computational Medicine: Whole Organ Simulation

- **Predictive Toxicology**
- **Multiscale Model of Organs**
  - from protein function through to cell function through to tissue function through to macroscale organ modeling.
- **Multiple model components and scales require Petascale to Exascale compute capability**
  - Usefulness requires "turnkey" modeling environment where many variations and scenarios can be attempted by the medical or pharmaceutical researcher quickly and accurately
  - Further increases the computational requirements

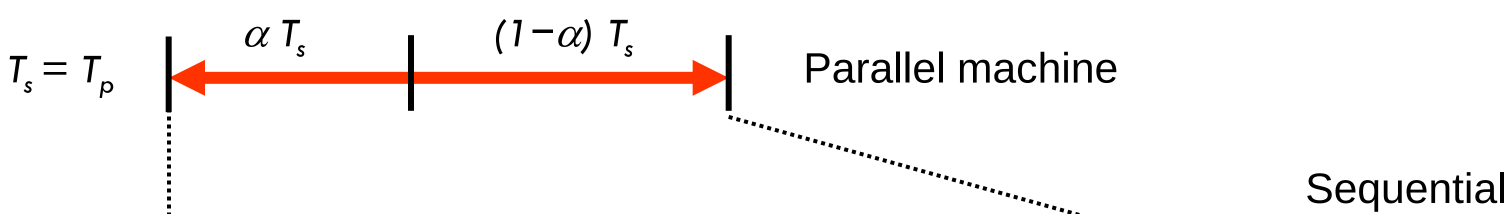




Gustafson's law

- The amount of work changes with the number of processors

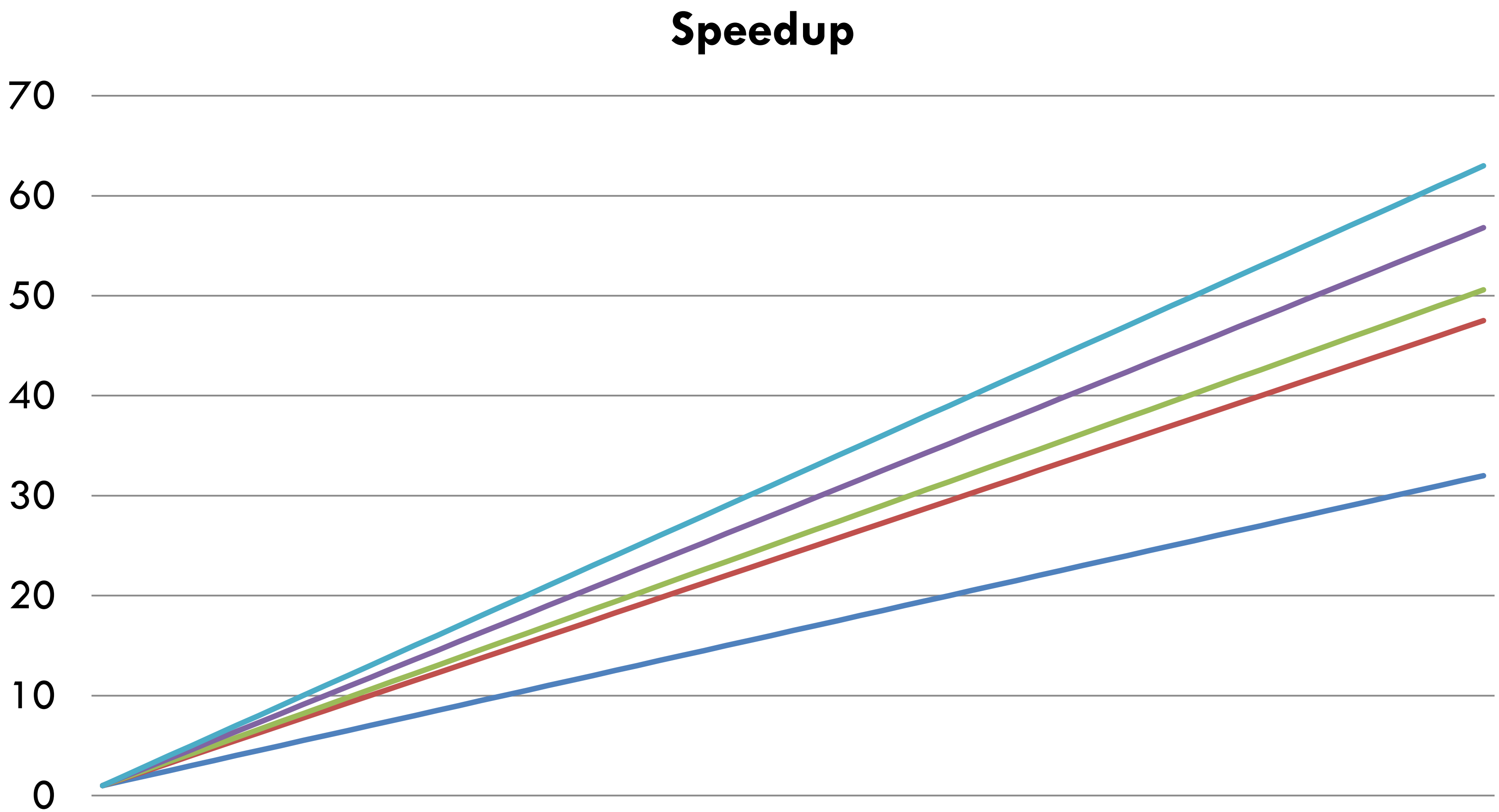
$$S_p = \frac{T'_s}{T_p} = \frac{\alpha T_s + (1-\alpha) p T_s}{T_s} = p + \alpha(1-p)$$



Gustafson's law

- The sequential portion of the program decreases with program size.
  - When the problem size grows we can assume a close-to-linear speedup (S≈p).
- Using parallelism, we can approach larger problems.

Results



The Processor Performance Equation

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$

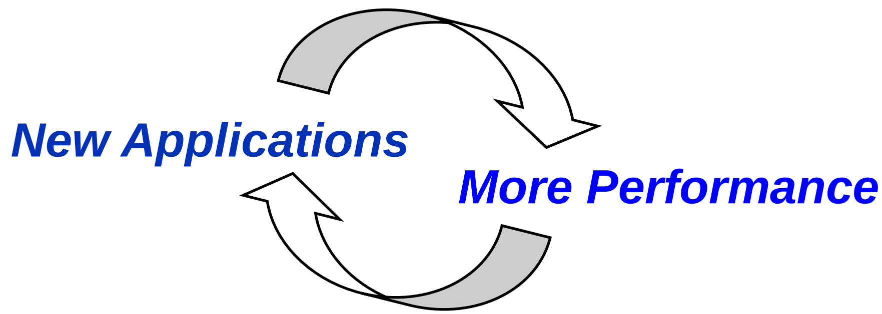
$$\text{CPU time} = \text{Instruction count} \times \text{Cycles per instruction} \times \text{Clock cycle time}$$

- Different instruction types having different CPIs

$$\text{CPU clock cycles} = \sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i$$

$$\text{CPU time} = \left(\sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock cycle time}$$

Application Trends



- Demand for cycles fuels advances in hardware, and vice-versa
  - Cycle drives exponential increase in microprocessor performance
  - Drives parallel architecture harder: most demanding applications
- Goal of applications in using parallel machines: Speedup

$$\text{Speedup (p processors)} = \frac{\textit{Performance (p processors)}}{\textit{Performance (1 processor)}}$$

Time (p processors)

Particularly Challenging Computations

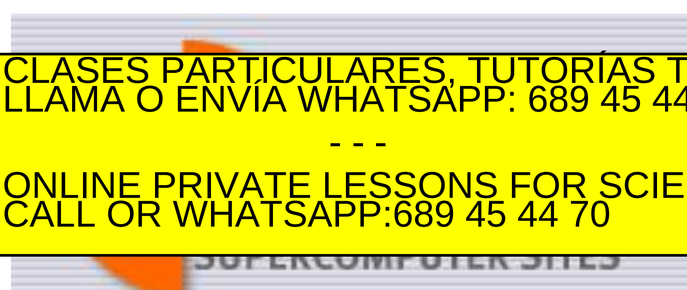
- Science
  - Global climate modeling
  - Astrophysical modeling
  - Biology: genomics; protein folding; drug design
  - Computational Chemistry
  - Computational Material Sciences and Nanosciences
- Engineering
  - Crash simulation
  - Semiconductor design
  - Earthquake and structural modeling
  - Computation fluid dynamics (airplane design)
  - Combustion (engine design)
- Business
  - Financial and economic modeling
  - Transaction processing, web services and search engines
- Defense

Supercomputing trends

- 1PFLOP has been surpassed in 2008
- Currently:
  - 33 PFLOPS
  - 3.1M cores system
- We head toward ExaScale age
  - 1,000,000,000 cores
- Increased probabilities of failures
  - Learn to live with failures
  - Fault tolerance
  - Learn to continue in the presence of failures
- Challenges in getting a global view of the system
- New challenges for applications and algorithms
- Scale invariance targeted
  - Local versus global
  - Learn from Internet

TOP 500

- Since 1993 twice a year: June and November
- Ranking of the most powerful computing systems in the world
- Ranking criteria: performance of the LINPACK benchmark
- Jack Dongarra alma máter
- Site web: www.top500.org
- Poster 2012:
  http://www.top500.org/static/lists/2012/06/TOP500_201206_Poster.pdf



Universidad Carlos III de Madrid
www.uc3m.es

Top 500 – June 2012

Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom / 2011 IBM	1572864	16324.75	20132.66	7890.0
2	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect / 2011 Fujitsu	705024	10510.00	11280.38	12659.9
3	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom / 2012 IBM	786432	8162.38	10066.33	3945.0
4	Leibniz Rechenzentrum Germany	**SuperMUC** - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR / 2012 IBM	147456	2897.00	3185.05	3422.7
5	National Supercomputing Center in Tianjin China	**Tianhe-1A** - NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050 / 2010 NUDT	186368	2566.00	4701.00	4040.0
6	DOE/SC/Oak Ridge National Laboratory United States	**Jaguar** - Cray XK6, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA 2090 / 2009 Cray Inc.	298592	1941.00	2627.61	5142.0
7	CINECA Italy	**Fermi** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom / 2012 IBM	163840	1725.49	2097.15	821.9
	Forschungszentrum Juelich (FZJ) [illegible]	**JuQUEEN** - BlueGene/Q, Power BQC [illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	National Supercomputing Centre in [illegible] China	**Nebulae** - Dawning TC3600 Blade System, Xeon X5650 6C 2.66GHz, [illegible] Dawning	120640	1271.00	2984.30	2580.0

Green 500

- For a long time performance has been the only metric
  - FLOPS
  - Total cost of ownership (TCO) neglected
- Conscience about increasing costs of power, maintenance, administration, failure recovery
- Ranking of the most energy-efficient supercomputers in the world
  - MFLOPS/Watt
- First edition: November 2007

Universidad Carlos III de Madrid
www.uc3m.es

Green500 – June 2012

ARCOS

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	2,100.88	DOE/NNSA/LLNL	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	41.10
2	2,100.88	IBM Thomas J. Watson Research Center	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	41.10
3	2,100.86	DOE/SC/Argonne National Laboratory	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	82.20
4	2,100.86	DOE/SC/Argonne National Laboratory	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	82.20
5	2,100.86	Rensselaer Polytechnic Institute	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	82.20
6	2,100.86	University of Rochester	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	82.20
7	2,100.86	IBM Thomas J. Watson Research Center	BlueGene/Q, Power BQC 16C 1.60 GHz, Custom	82.20
8	2,099.56	University of Edinburgh	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	493.10
[illegible]	[illegible]	Science and Technology Facilities Council - Daresbury	BlueGene/Q, Power BQC 16C 1.60GHz,	[illegible]