SciDAC
Scientific Discovery through Advanced Computing

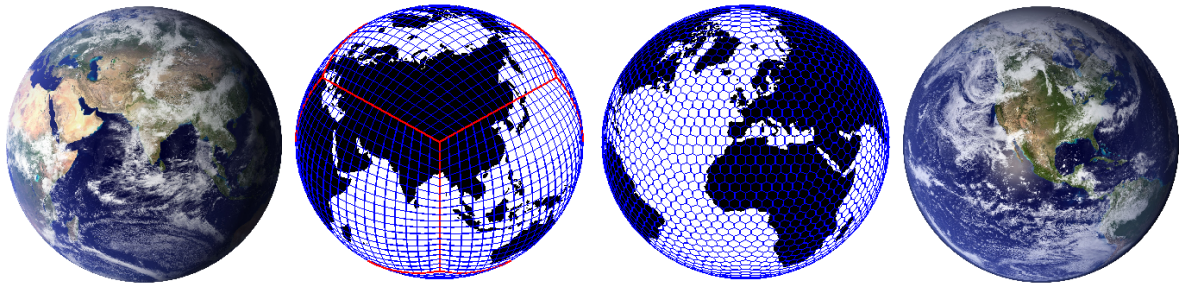
NSF

CGD Climate & Global Dynamics

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Supercomputing and climate modeling

Peter Hjort Lauritzen
National Center for Atmospheric Research (NCAR)



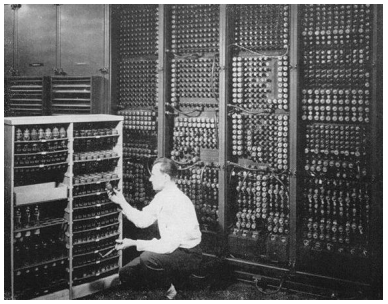
Lecture at Summer school: Introduction to Climate Modeling
University of Stockholm, May 28, 2012

NCAR Earth System Laboratory NESL

Disclaimer

I am NOT a computer scientist, however, we might be at a way-point where model developers need to pay extra attention to computer architecture development … we might all have to become more "computer scientists" at some level

Thanks to Rory Kelly (CISL/NCAR) and Rich Loft (CISL/NCAR) for input to this talk



ENIAC, 1947-1955

Outline

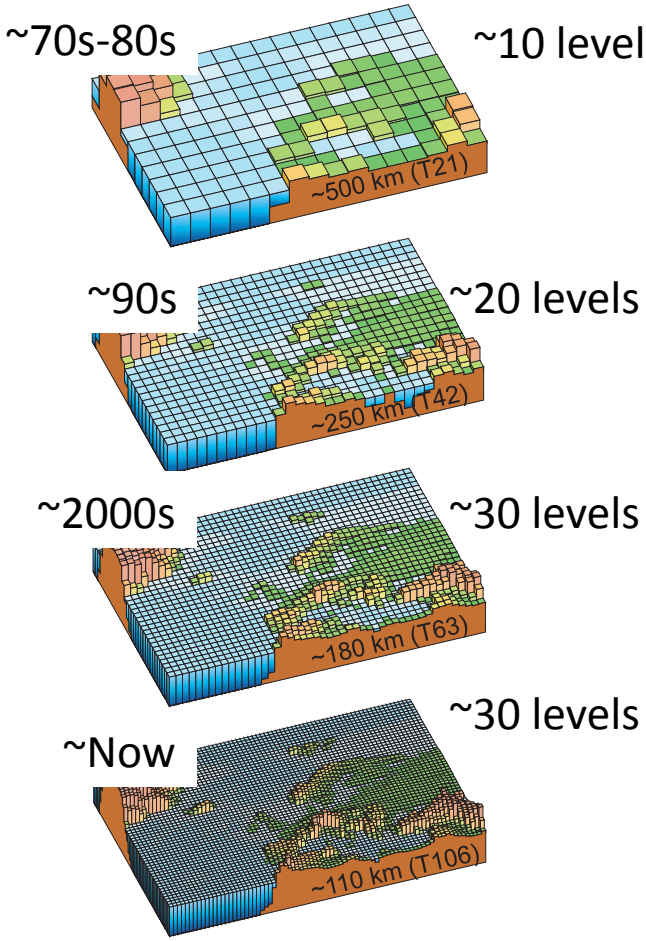
Bluefire, 2008-2012

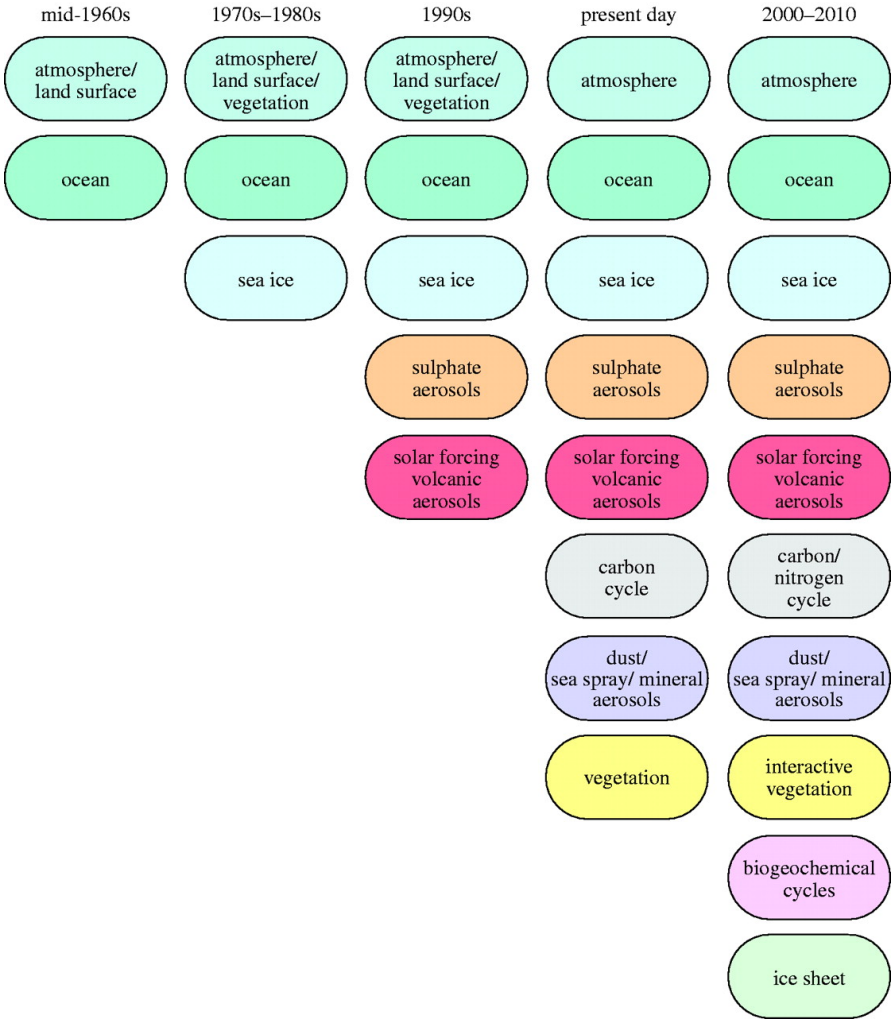
- History of climate model resolution and complexity made possible by increases in computing power and science

- History of super computer performance
- How did/do we increase peak performance?

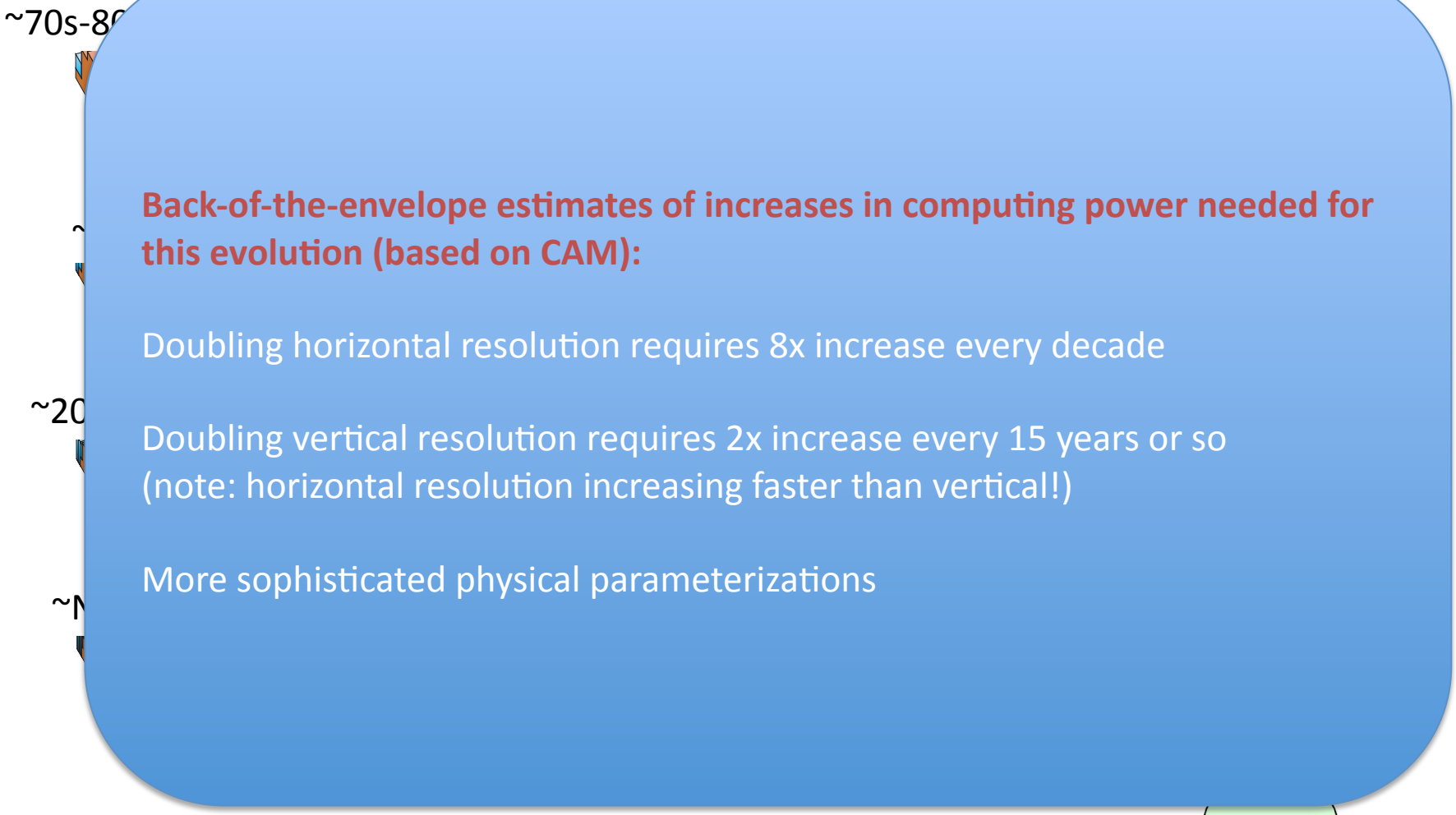
  (increasing CPU clock-speed -> multiple CPUs -> multi everything?
- Climate modeling: `parallelization' and moving to isotropic grids
- Future of supercomputing?

History of horizontal resolution and complexity in climate models used for century scale simulations (climate change)





History of horizontal resolution and complexity in climate models used for century scale simulations (climate change)



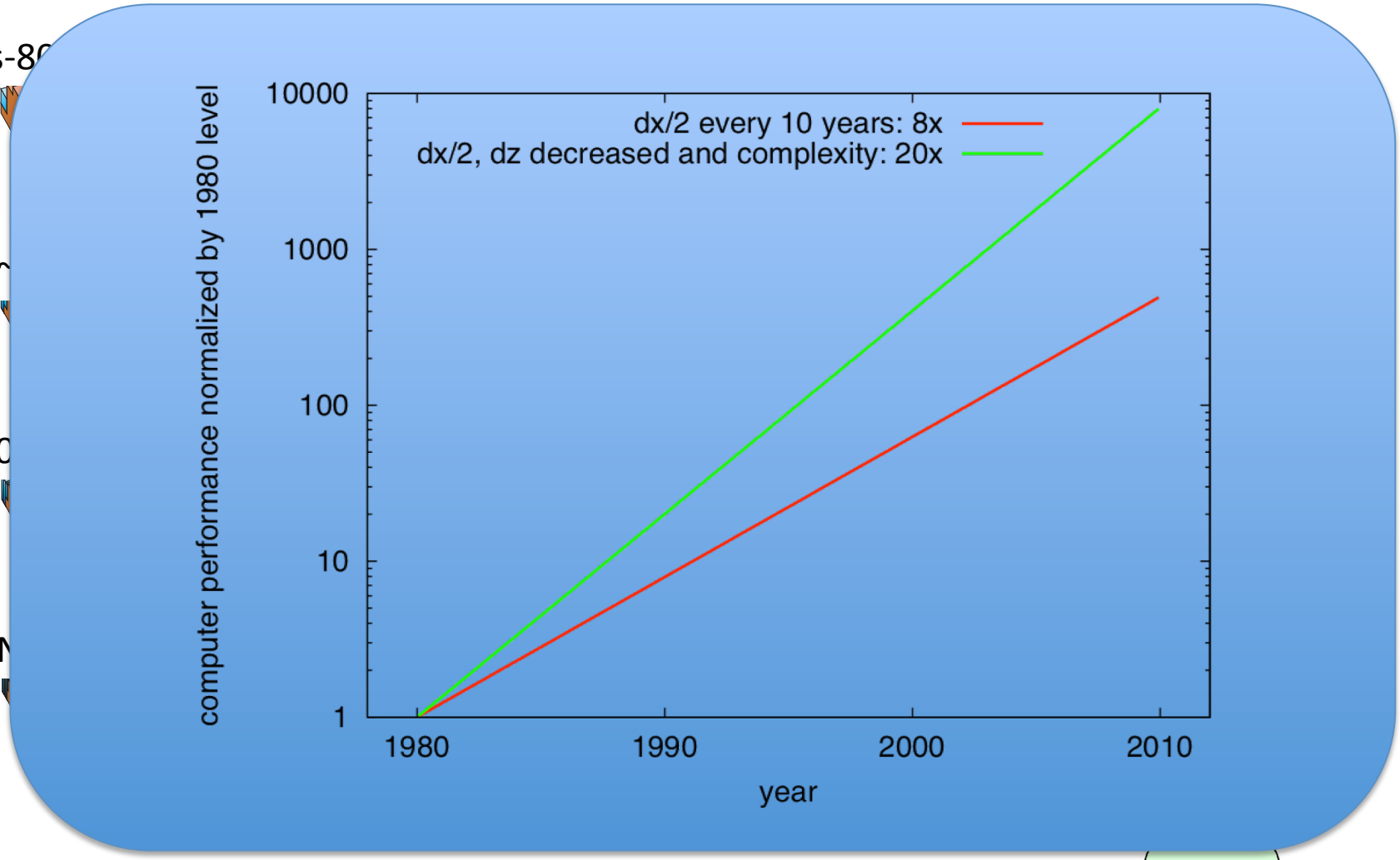
Back-of-the-envelope estimates of increases in computing power needed for this evolution (based on CAM):

Doubling horizontal resolution requires 8x increase every decade

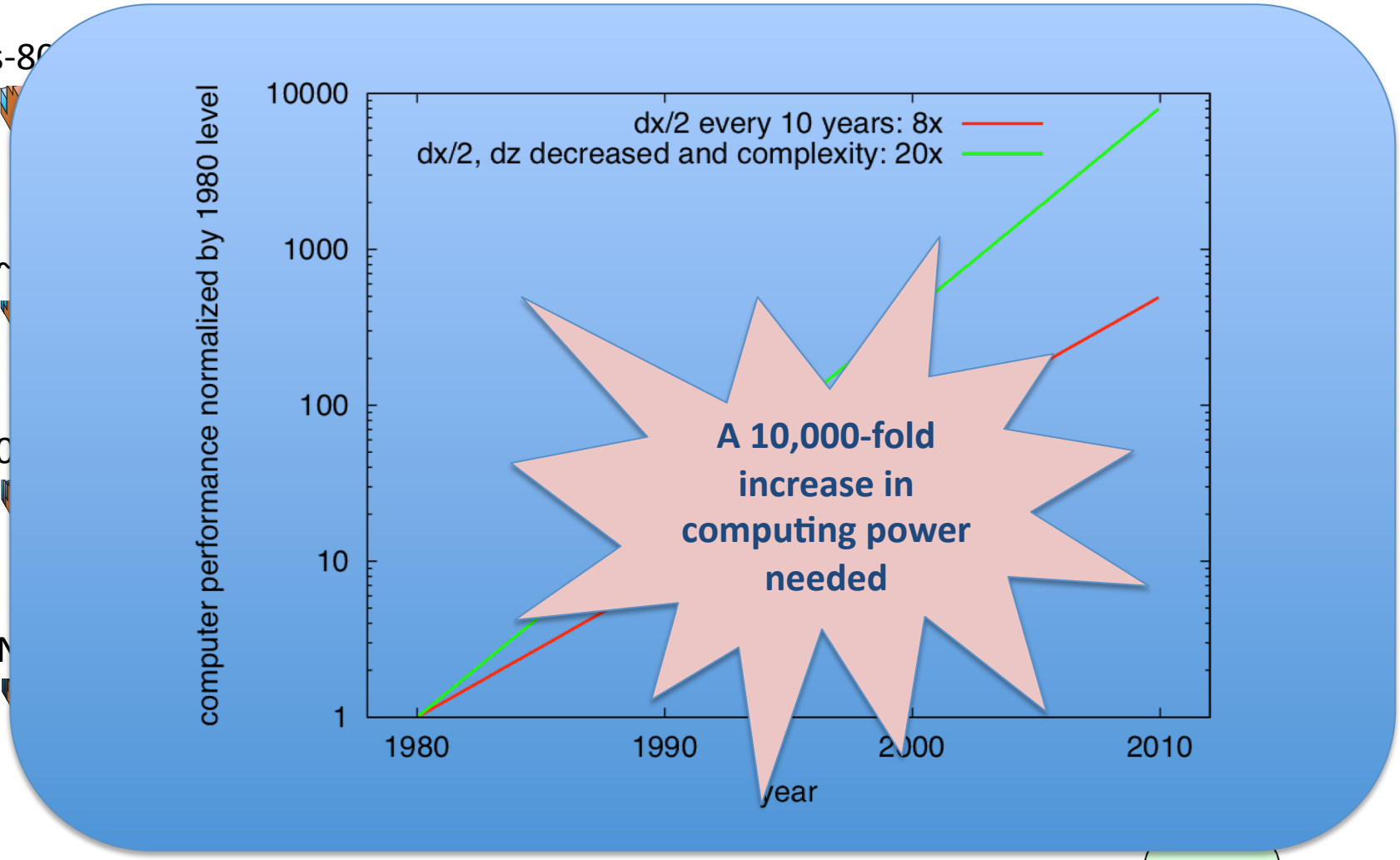
Doubling vertical resolution requires 2x increase every 15 years or so
(note: horizontal resolution increasing faster than vertical!)

More sophisticated physical parameterizations

History of horizontal resolution and complexity in climate models used for century scale simulations (climate change)



History of horizontal resolution and complexity in climate models used for century scale simulations (climate change)



Home ▸ Project

TOP500 Description

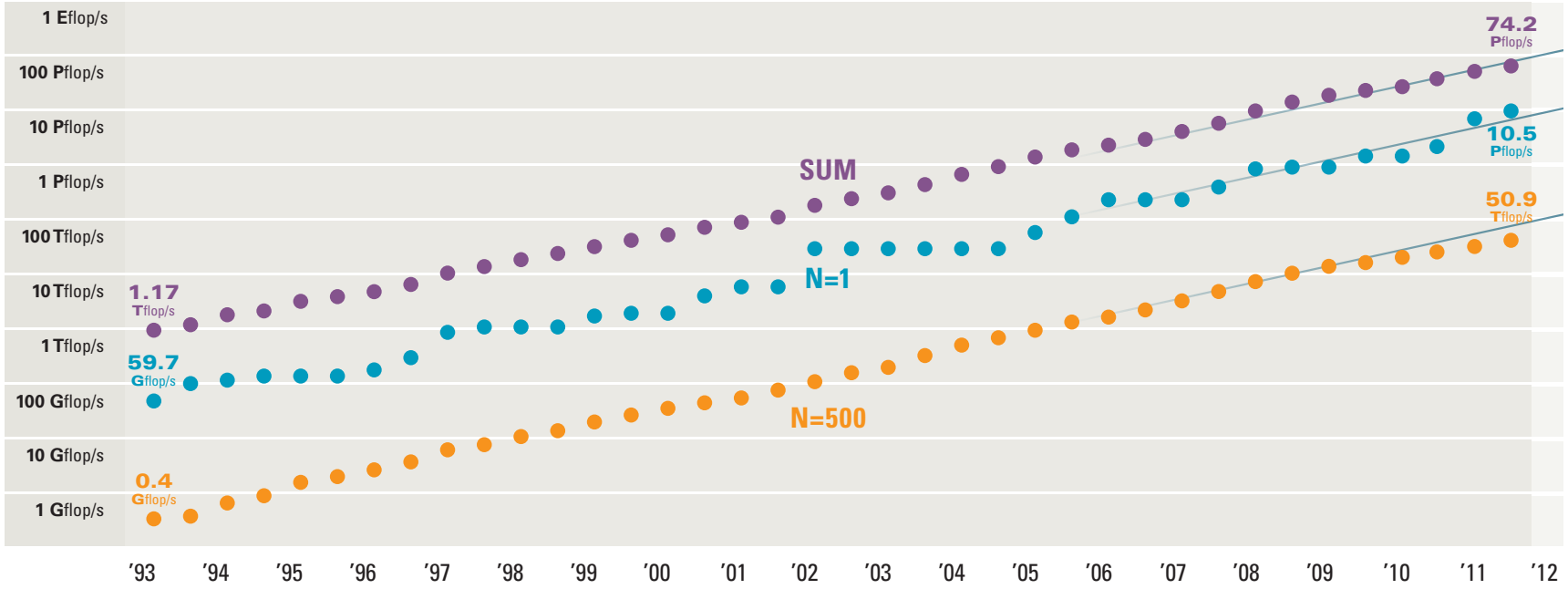
The TOP500 table shows the 500 most powerful commercially available computer systems known to us. To keep the list as compact as possible, we show only a part of our information here:

- Nworld - Position within the TOP500 ranking
- Manufacturer - Manufacturer or vendor
- Computer - Type indicated by manufacturer or vendor
- Installation Site - Customer
- Location - Location and country
- Year - Year of installation/last major update
- Field of Application
- #Proc. - Number of processors (Cores)
- Rmax - Maximal LINPACK performance achieved
- Rpeak - Theoretical peak performance
- Nmax - Problem size for achieving Rmax
- N1/2 - Problem size for achieving half of Rmax

www.top500.org

Performance development (top500)

- How is performance measured? Sparse matrix inversion (Ax=b) using LINPACK
- TOP500-list updated twice a year
- Note: the y-axis is logarithmic!!!
- **Computing power doubles roughly every 14 months (in 20 years computing power has increased 10,000-fold)**



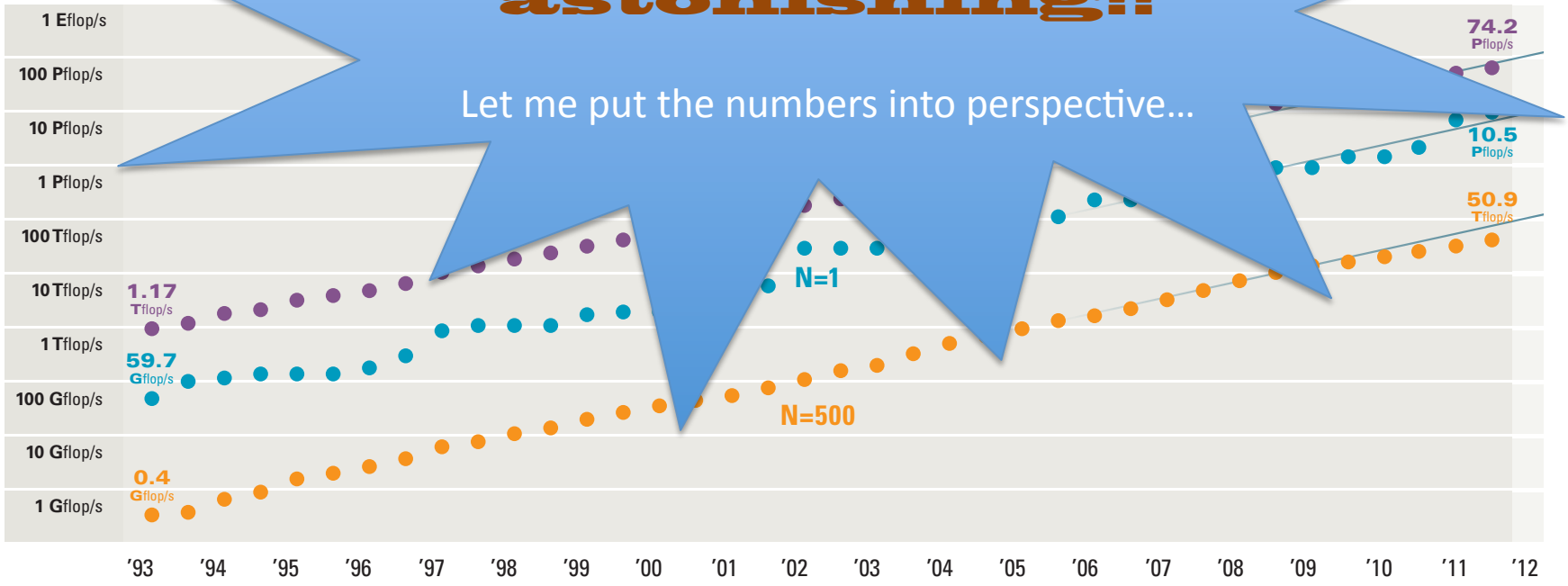
Source: www.top500.org

Performance development (top500)

- How is pe… mance measured? Sp… matrix inve… ng LINPACK
- TOP500-list up… ice a year
- Note: the y-axis is …
- **Computing power d… power has in…**

This is absolutely astonishing!!

Let me put the numbers into perspective…

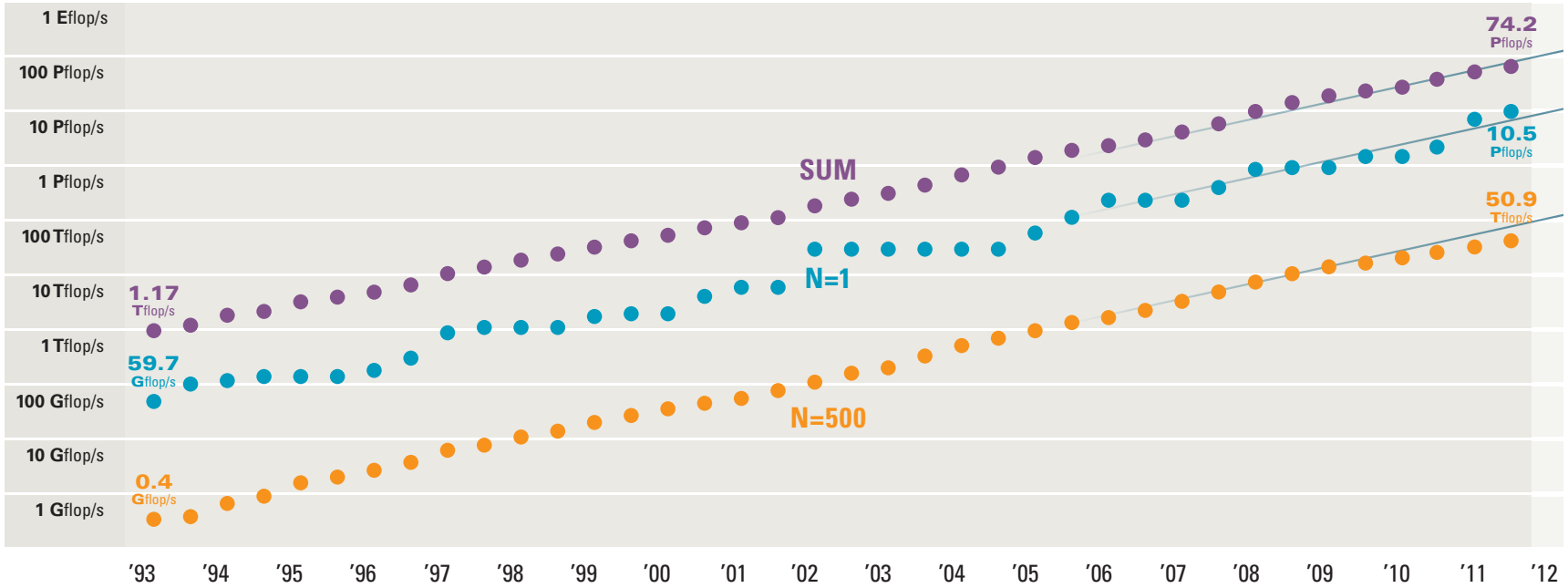


Source: www.top500.org

Performance development (top500)

How long ago would an iPhone 4S have made the top500 list?

When would a modern laptop have made the top500 list?

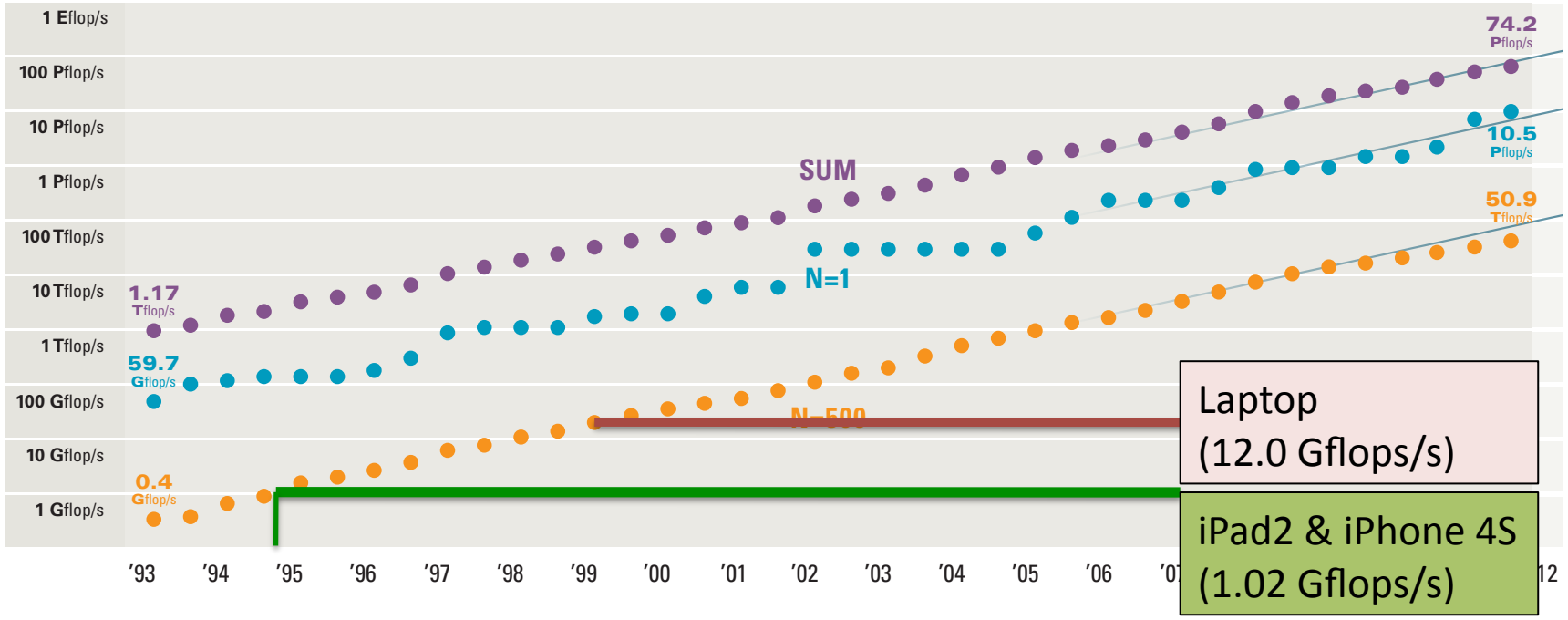


Source: www.top500.org

Performance development (top500)

An iPhone 4S would have made the top500 list less than 20 years ago!

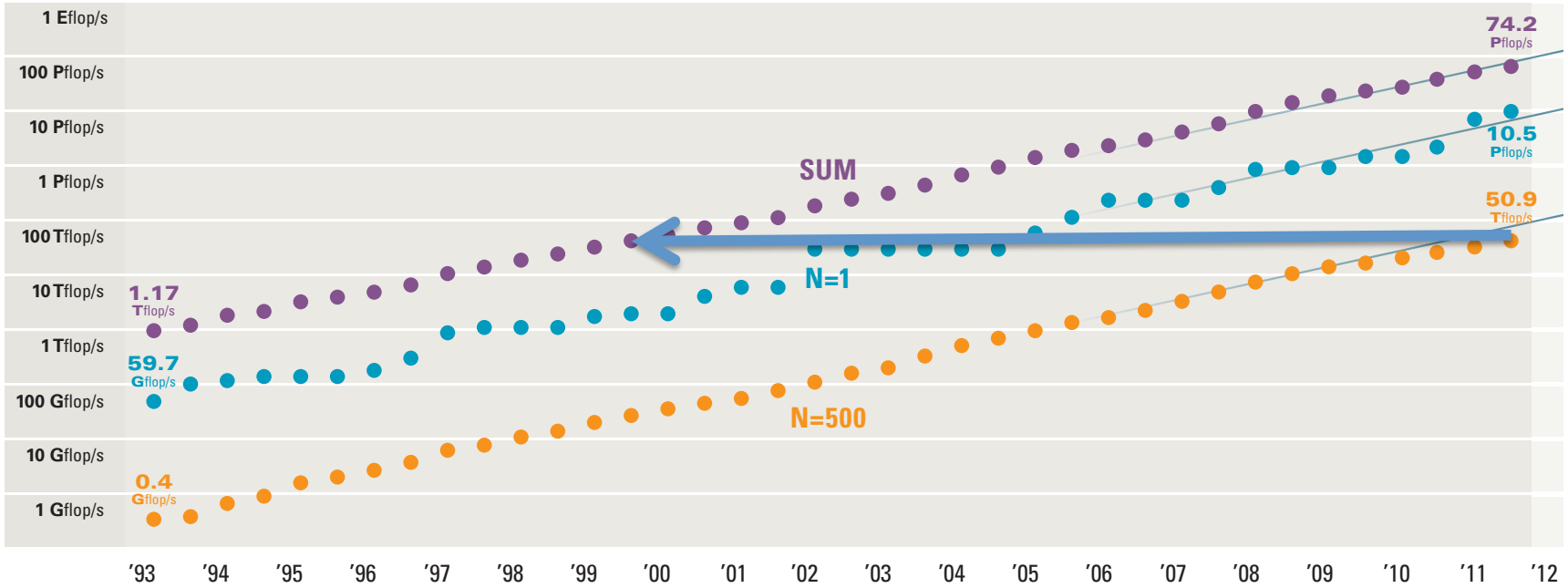
A modern laptop would have made the top500 list about 15 years ago!



Source: www.top500.org

Performance development (top500)

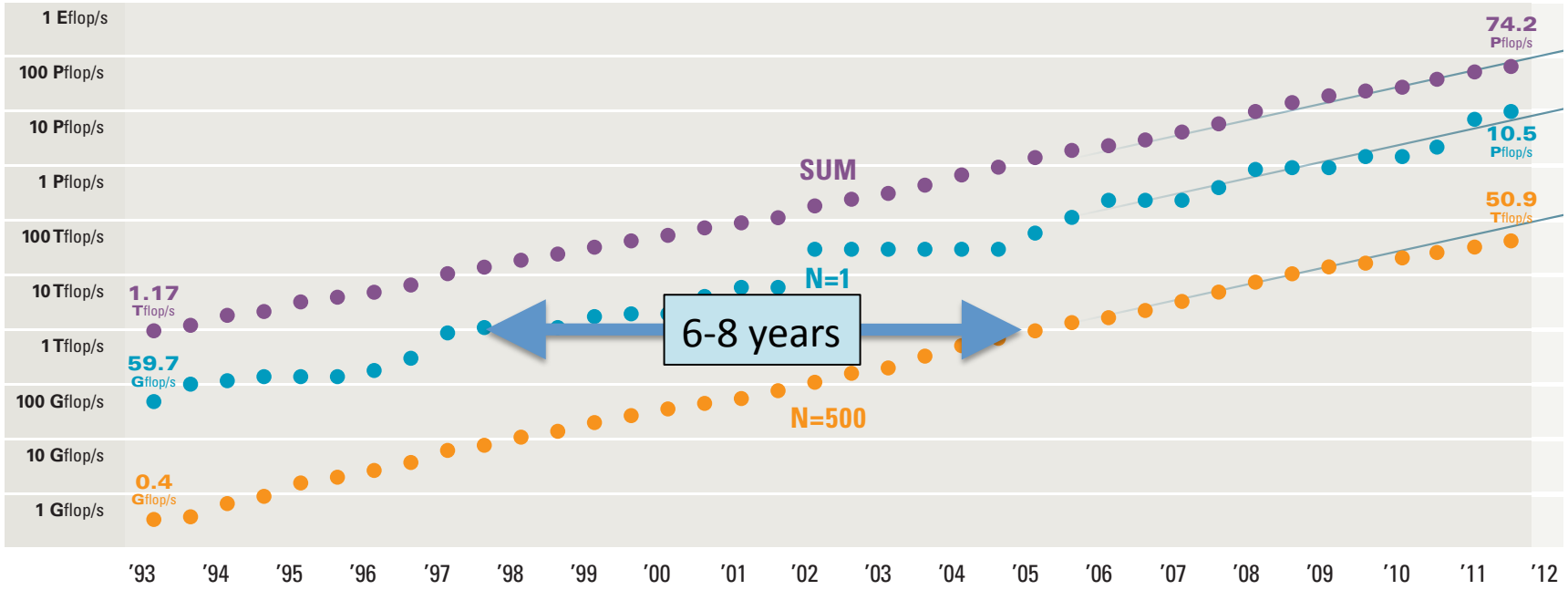
The slowest computer on the top500 list today is as `fast' as all the top500 computing power about 13 years ago!



Source: www.top500.org

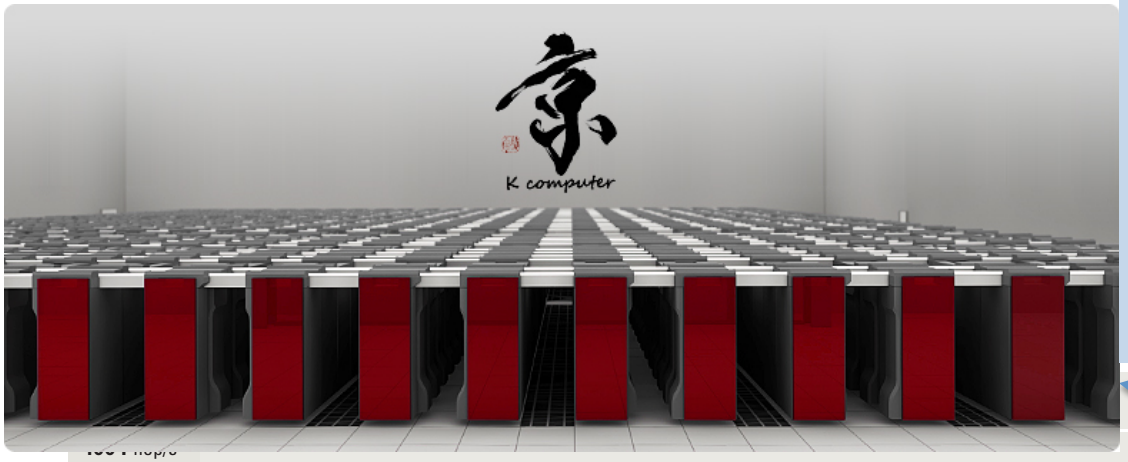
Performance development (top500)

- It takes 6-8 years from being the fastest computer to moving off the top500 list (lifetime of hardware <<<< lifetime of software)



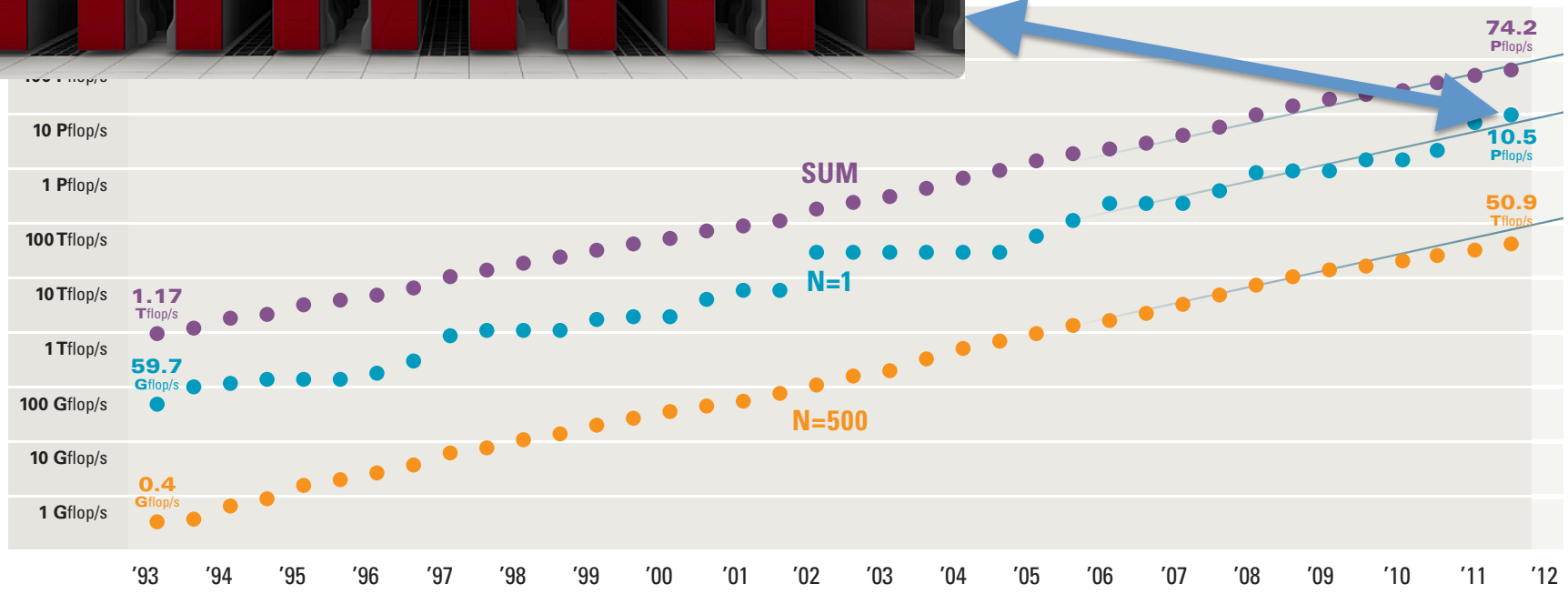
Source: www.top500.org

Performance development (top500)



Nr. 1: Japanese K computer:

- 705,024 processing cores
- Linpack benchmark performance: 10 quadrillion operations/second (a.k.a. 10 petaflops)

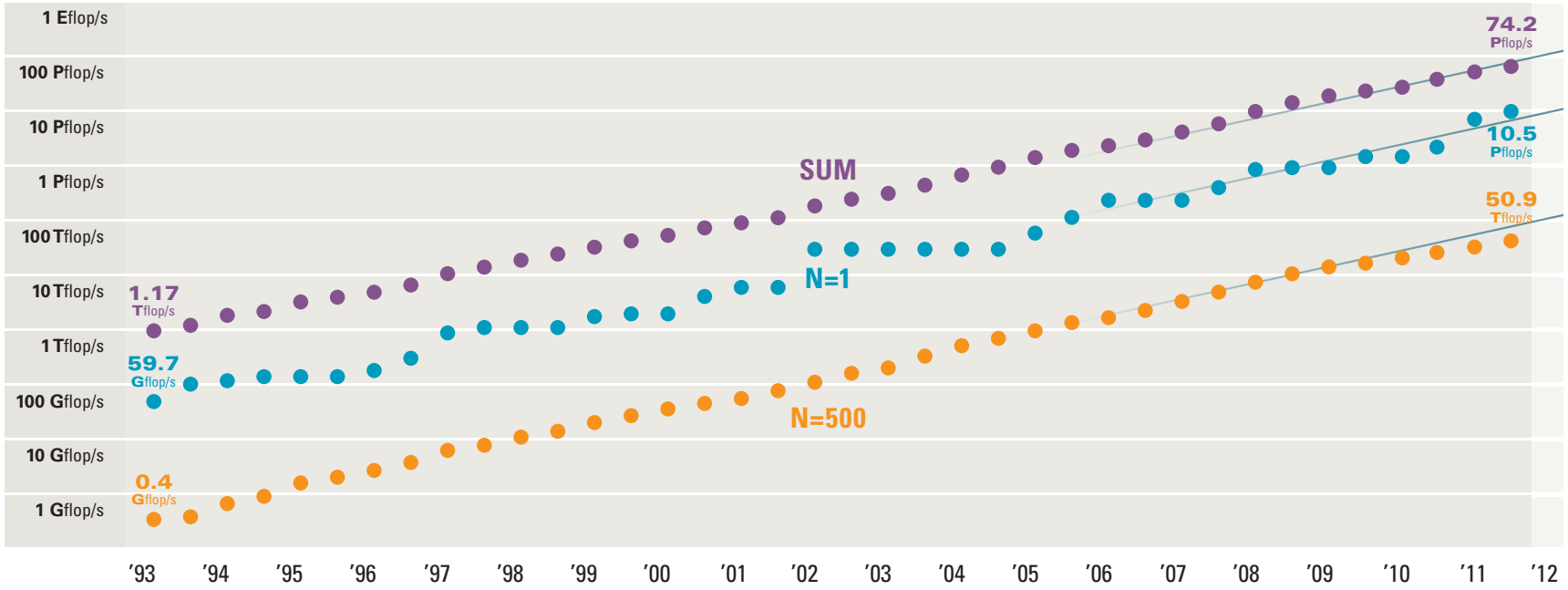


Source: www.top500.org

Performance development (top500)

How did we get here?

The hardware and software challenges haven been faced to maintain the astonishing constant slope of the performance curve were/are far from trivial!



Source: www.top500.org

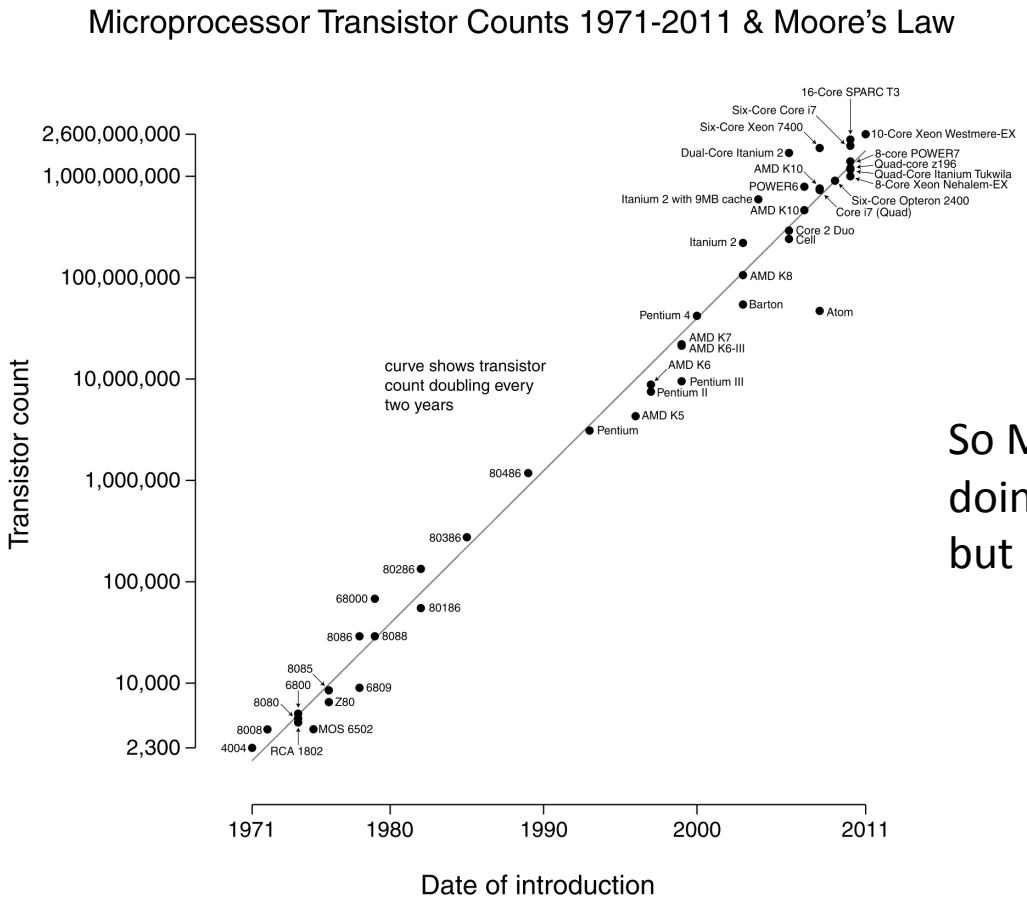
CPU and transistors

- CPU = Central Processing Unit (loosely speaking the brain of your computer)
- Transistor = basic building block of a CPU

CPU's (processors) are composed of thin layers of millions/billions of transistors. Transistors are tiny, nearly microscopic bits of material that will block electricity when the electricity is only a weak charge, but will allow the electricity to pass through when the electricity is strong enough.

Moore’s law

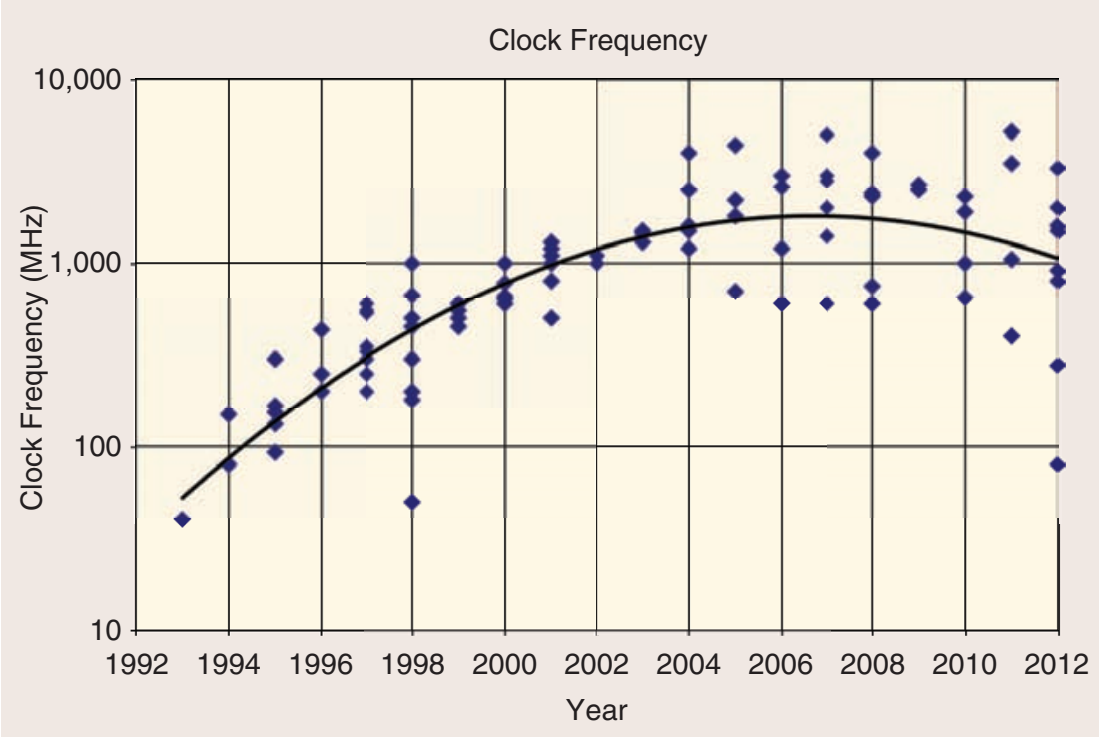
- Moore's law is a rule of thumb in the history of computing hardware whereby the number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.



So Moore’s law is still doing remarkably well but ….

Source: http://en.wikipedia.org/wiki/Moore's_law

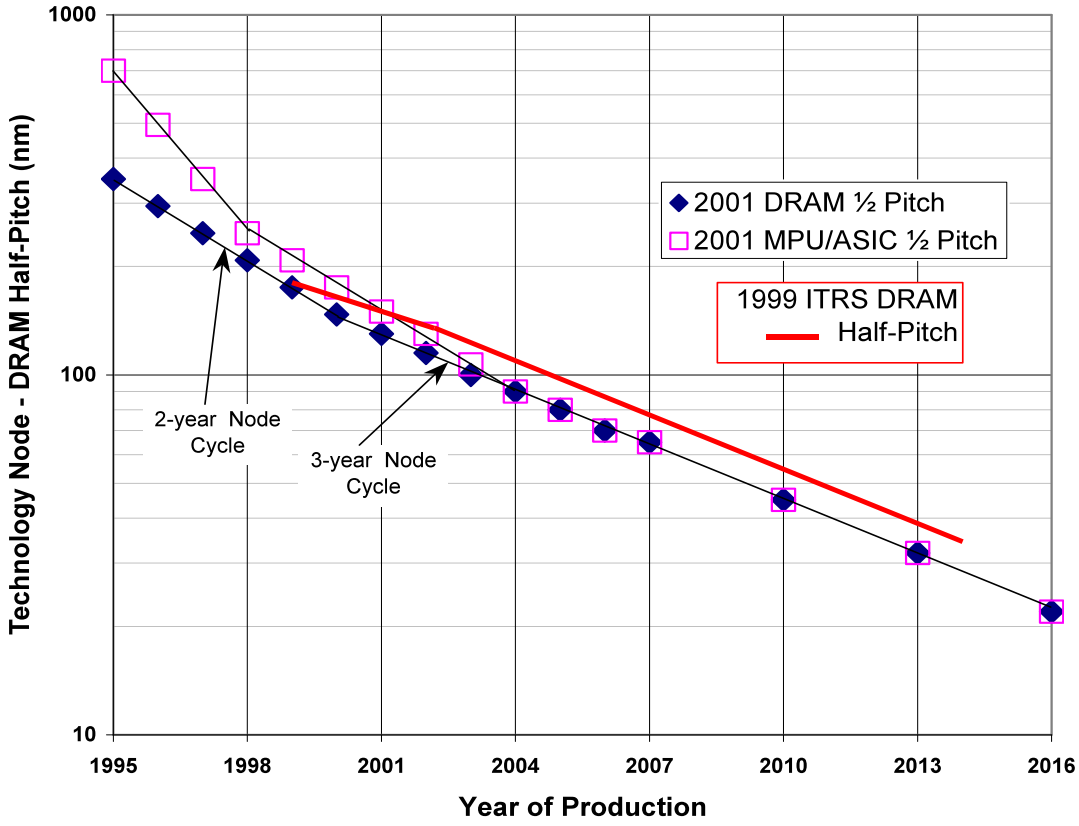
Clock speed for single CPU



- Clock speed stopped doubling at the same rate as transistors (Moore's law) around 2005.
- Your laptop is not using 10+GHz processors today as it should have if clock-speed scaled as Moore's law!
- Clock speed has stagnated around 3.4 GHz! "THE ERA OF FREE LUNCH IS OVER"!
- Why?

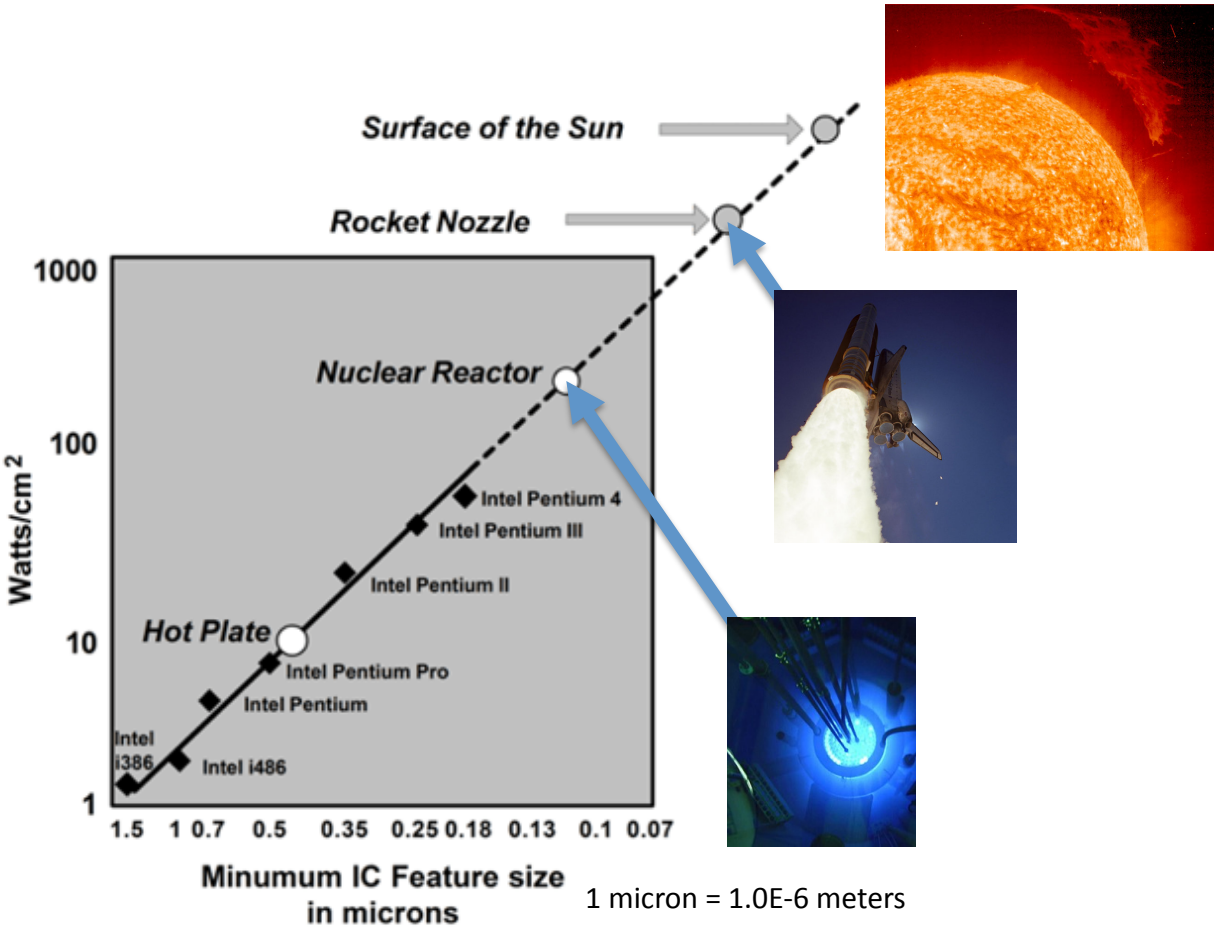
Source: ISSCC 2012 trend report Shttp://isscc.org/doc/2012/2012_Trends.pdf

Feature size keeps decreasing



- "Naive extrapolation": By roughly 2050 feature size will reach the size of an atom!
- Decreasing feature size affects power density if you want to increase clock speed (see next slide)

Rice in power density



1 micron = 1.0E-6 meters

- ~ 80% increase in power density /generation; ~225% increase in current consumption/unit area
- Power density is too high to keep microprocessors cool enough

Source: F Pollack, "New microarchitecture challenges in the coming generations of CMOS process technologies," MICRO-32, Haifa, Israel, 1999.

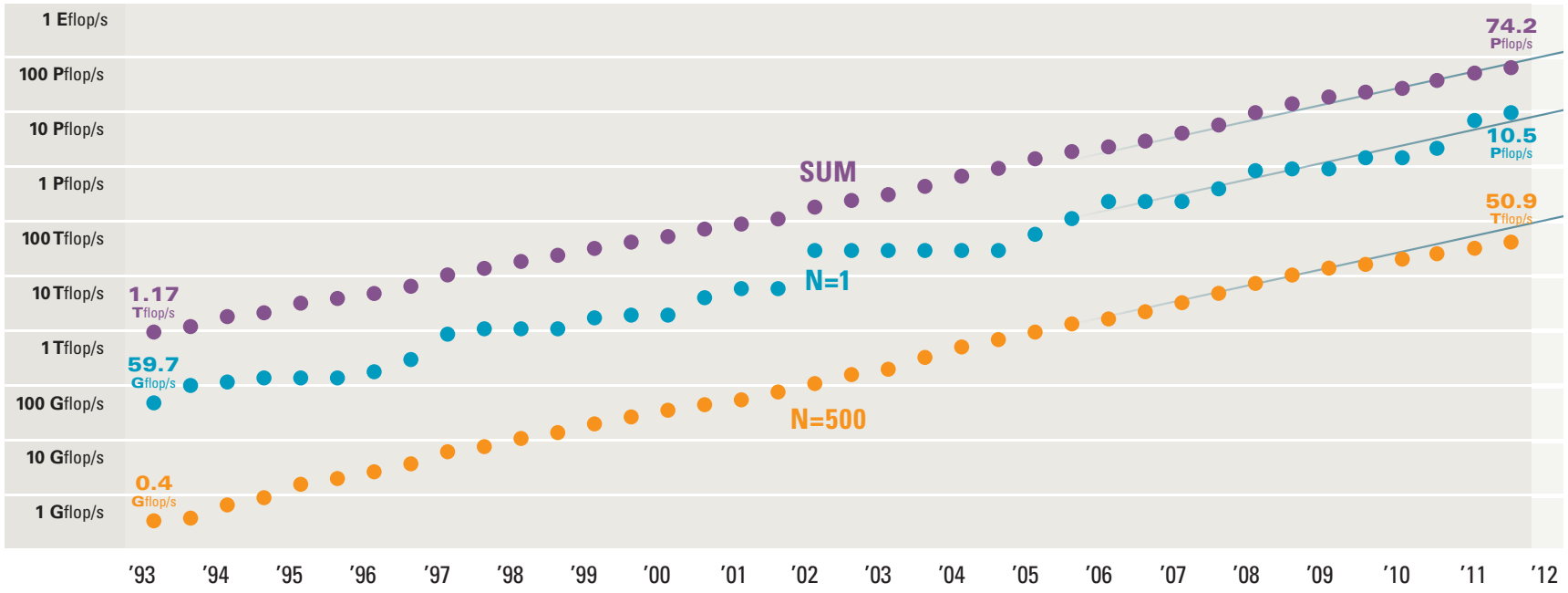
Another challenge when feature size decrease

(not considering challenges in manufacturing)

Leakage currents: Long before we hit atomic scales, quantum mechanics will start working against current designs

Exponential increase of gate direct tunneling currents (loosely speaking, it gets harder to distinguish between 0 and 1; and there is more waste energy since “0” is not zero current)

- With the stagnation of clock frequency increased peak performance could no longer be achieved by “waiting” for faster CPUs
- Why is performance still doubling every 14 months or so?
  (which is even faster than 18 months predicted by Moore’s lave)



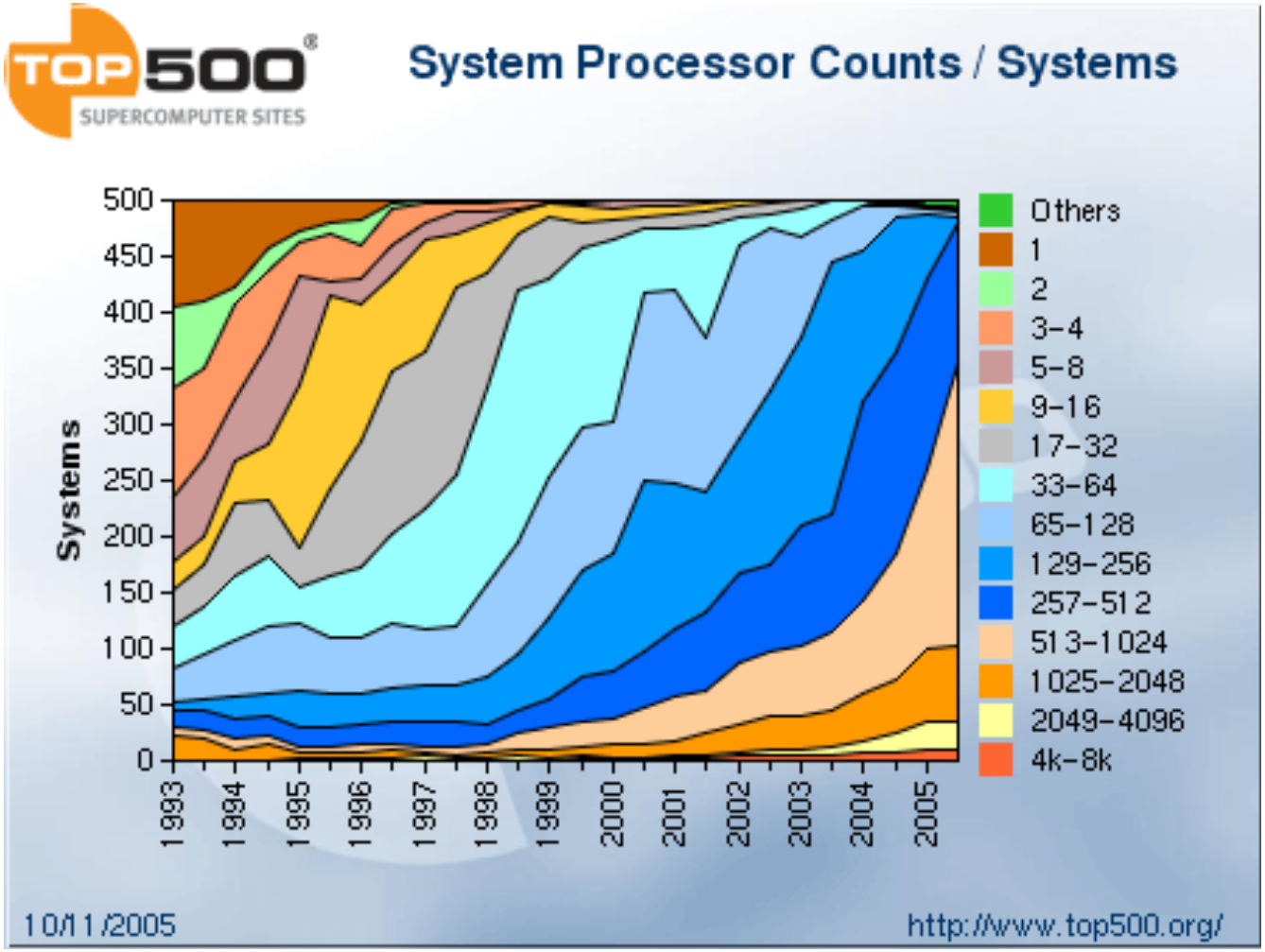
Parallel computing

Problems are divided into smaller problems that are solved concurrently – programming model: MPI (Message Passing Interface)

(note: this trend was already initiated before the CPU clock frequency stagnated)

IBM/LLNL's Blue Gene/L Supercomputer

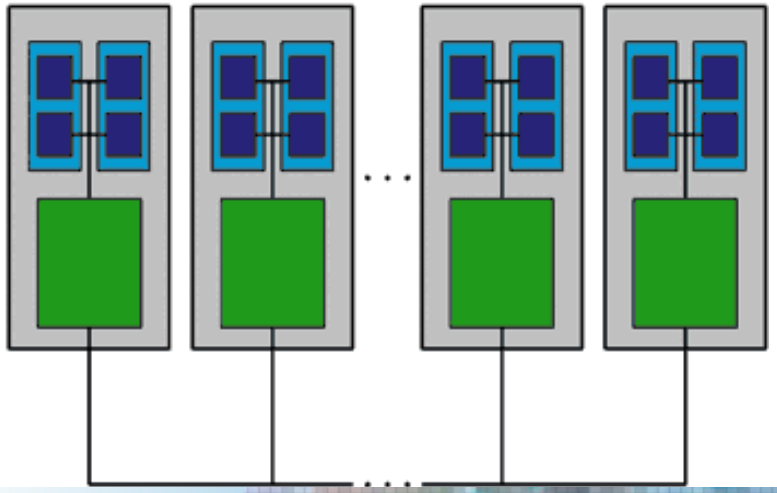
System Processor Counts Share Over Time



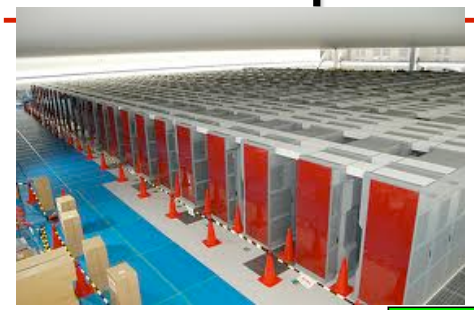
Parallel machines - basic idea

Each grey box is a node
Each node has typically 4-32 CPUs (blue boxes) sharing the same memory (green boxes)
The nodes are connected via an interconnect

Communication between nodes is much more expensive than local computation
(it can “pay off” to compute the same thing on two different nodes to avoid communication!)
Communication between all nodes (global gather) is very expensive if done often!



Example of typical parallel machine

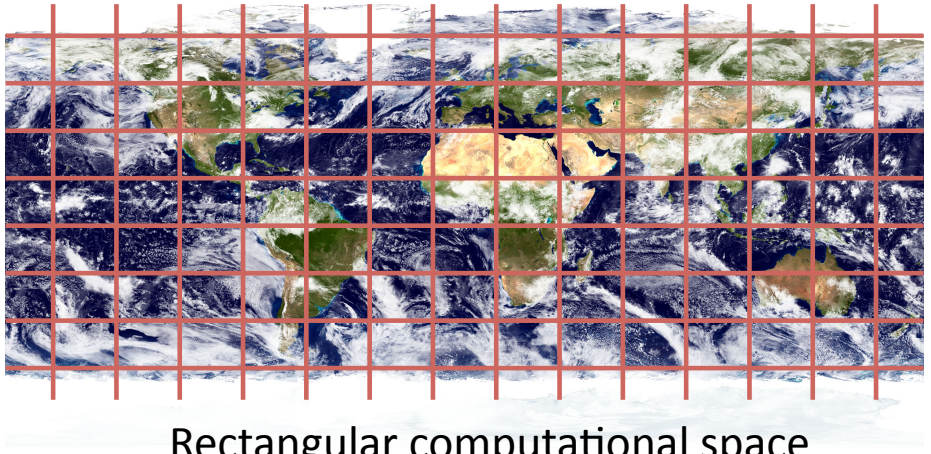


Combination of shared memory and distributed memory programming

Slide from J. Dongarra talk

Parallel computing architectures have been a major motivator in the re-design of dynamical cores. Why?

Regular latitude-longitude grids need non-local (global) filters in the polar regions (e.g., NCAR CAM-FV) or use non-local spectral transform methods (e.g., ECMWF IFS).

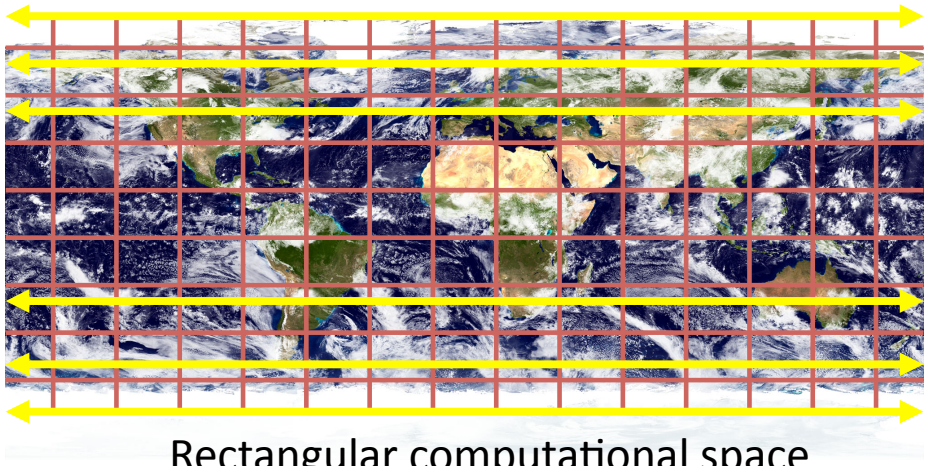


Rectangular computational space

Polar filtering example

Parallel computing architectures have been a major motivator in the re-design of dynamical cores. Why?

Regular latitude-longitude grids need non-local (global) filters in the polar regions (e.g., NCAR CAM-FV) or use non-local spectral transform methods (e.g., ECMWF IFS).

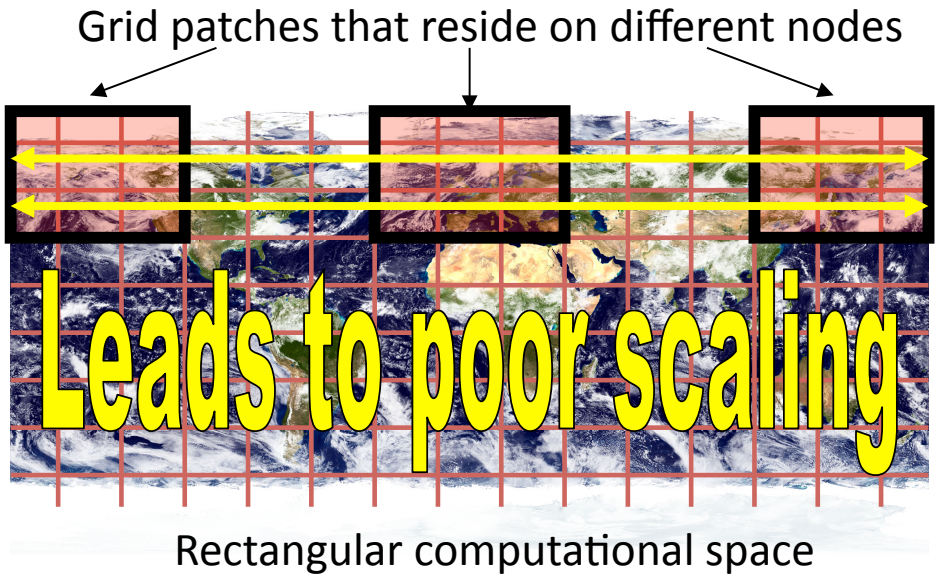


Rectangular computational space

Polar filtering example

Parallel computing architectures have been a major motivator in the re-design of dynamical cores. Why?

Regular latitude-longitude grids need non-local (global) filters in the polar regions (e.g., NCAR CAM-FV) or use non-local spectral transform methods (e.g., ECMWF IFS).

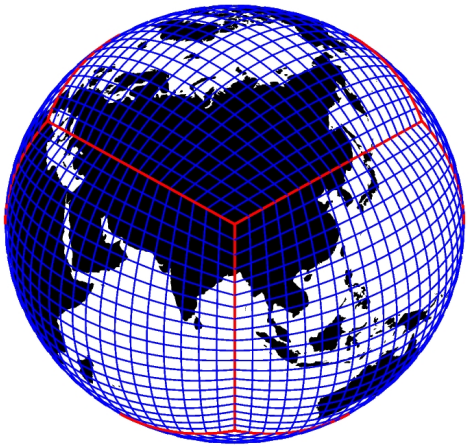


A solution

Use a more isotropic grid (avoid pole problem, can use full 2D domain decomposition in horizontal directions, if using local numerical method only nearest neighbor communication):

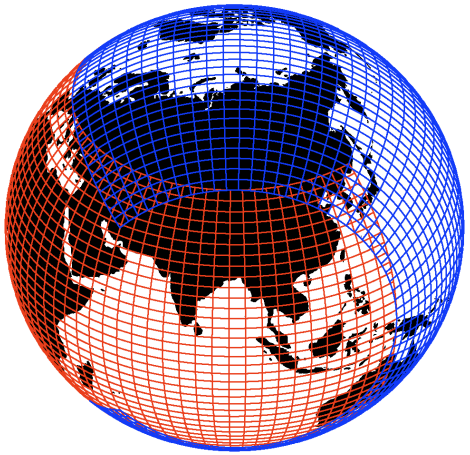
Regular Latitude-longitude

Cubed-sphere

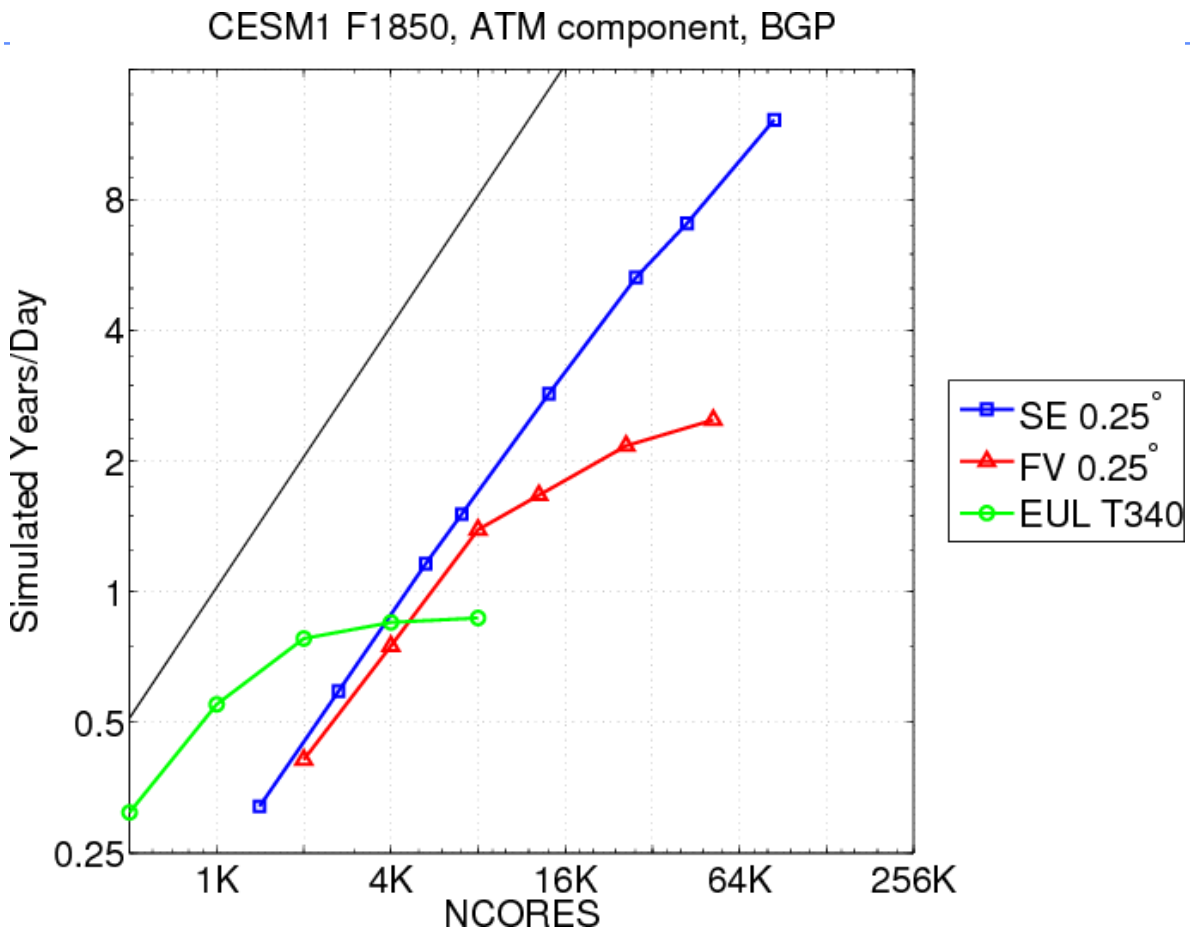


Icosahedral/Voronoi

Yin-Yang



A CESM example



To do science > 5 SYPD

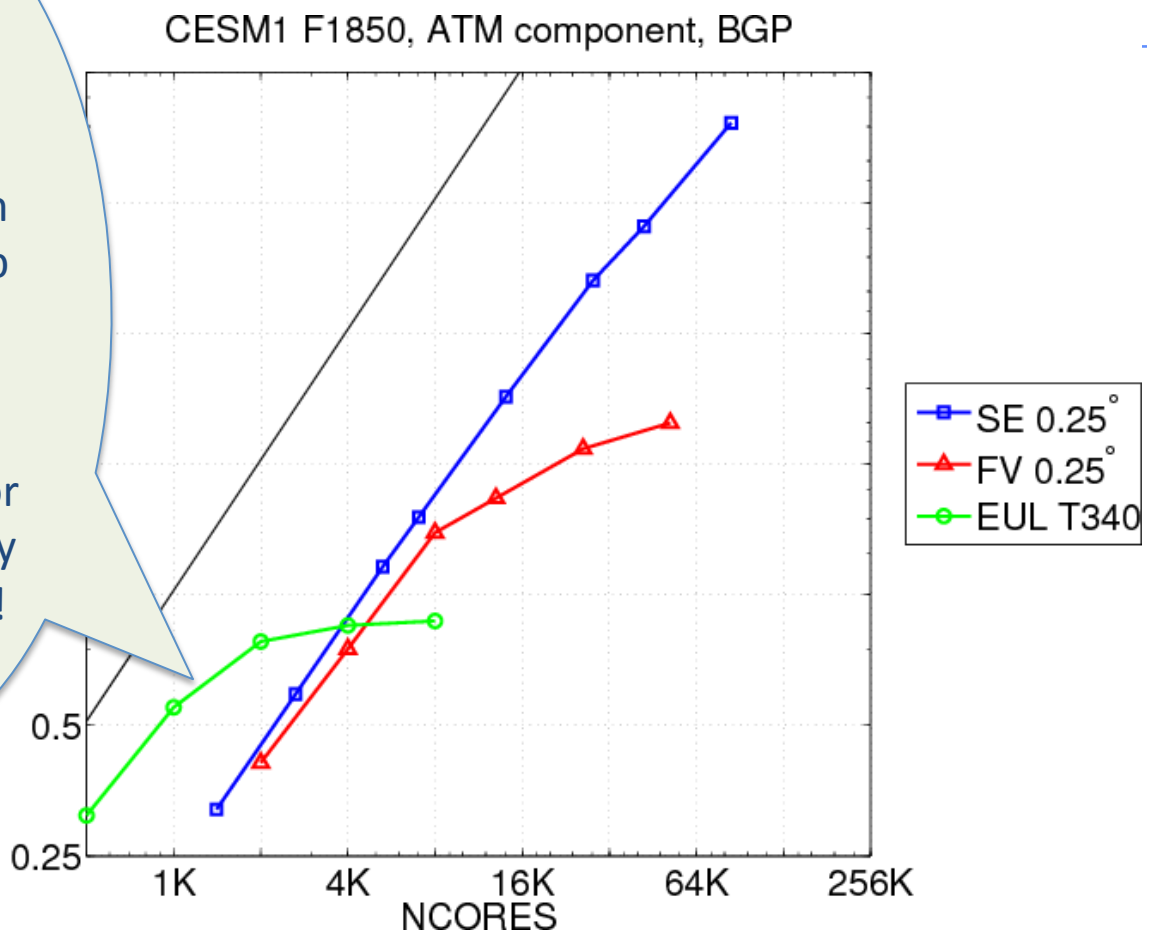
Plot: courtesy of Mark Taylor

A CESM example

Spherical harmonics: global communication every time-step

HOWEVER

At low processor counts very very very efficient!!!

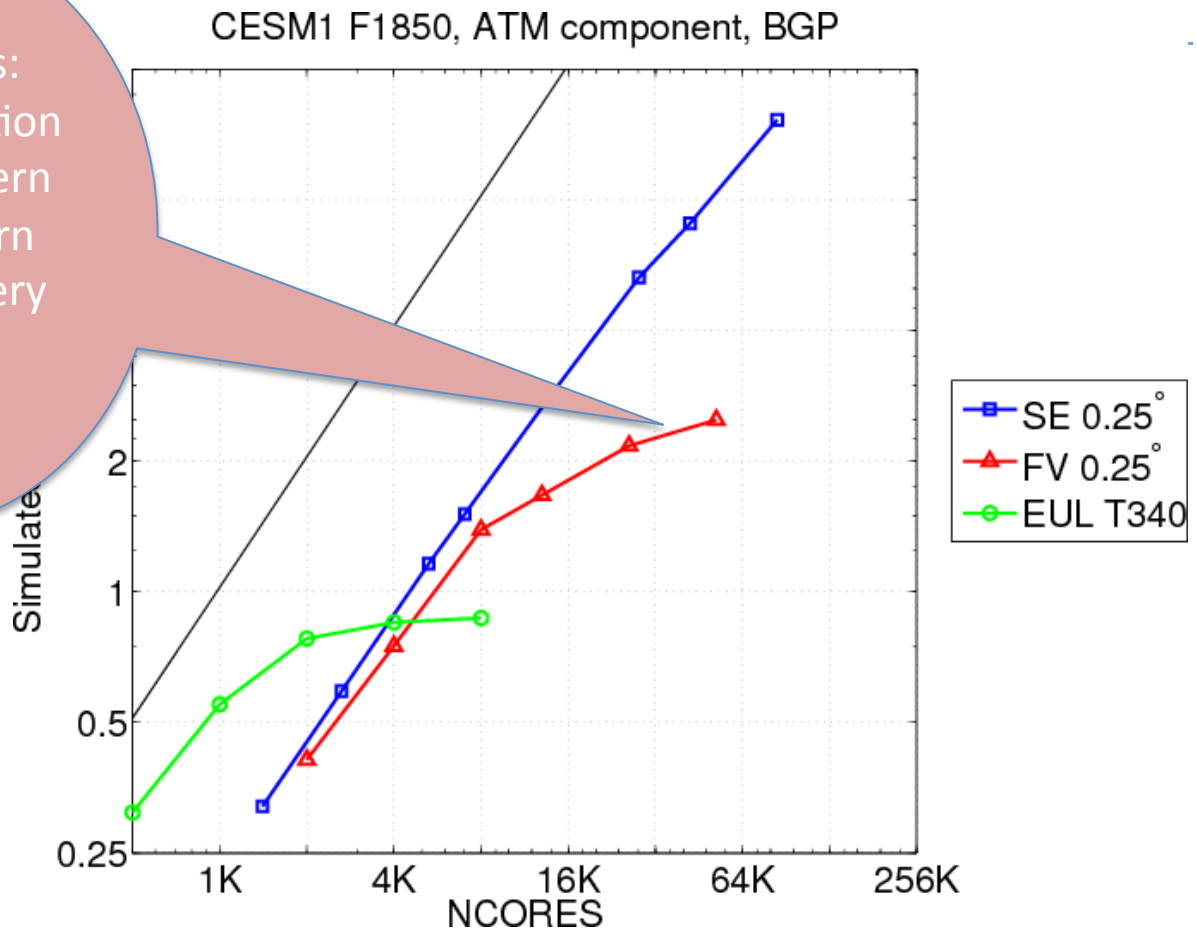


To do science > 5 SYPD

Plot: curtesy of Mark Taylor

A CESM example

Polar filters: global operation along Northern and Southern latitudes every time-step

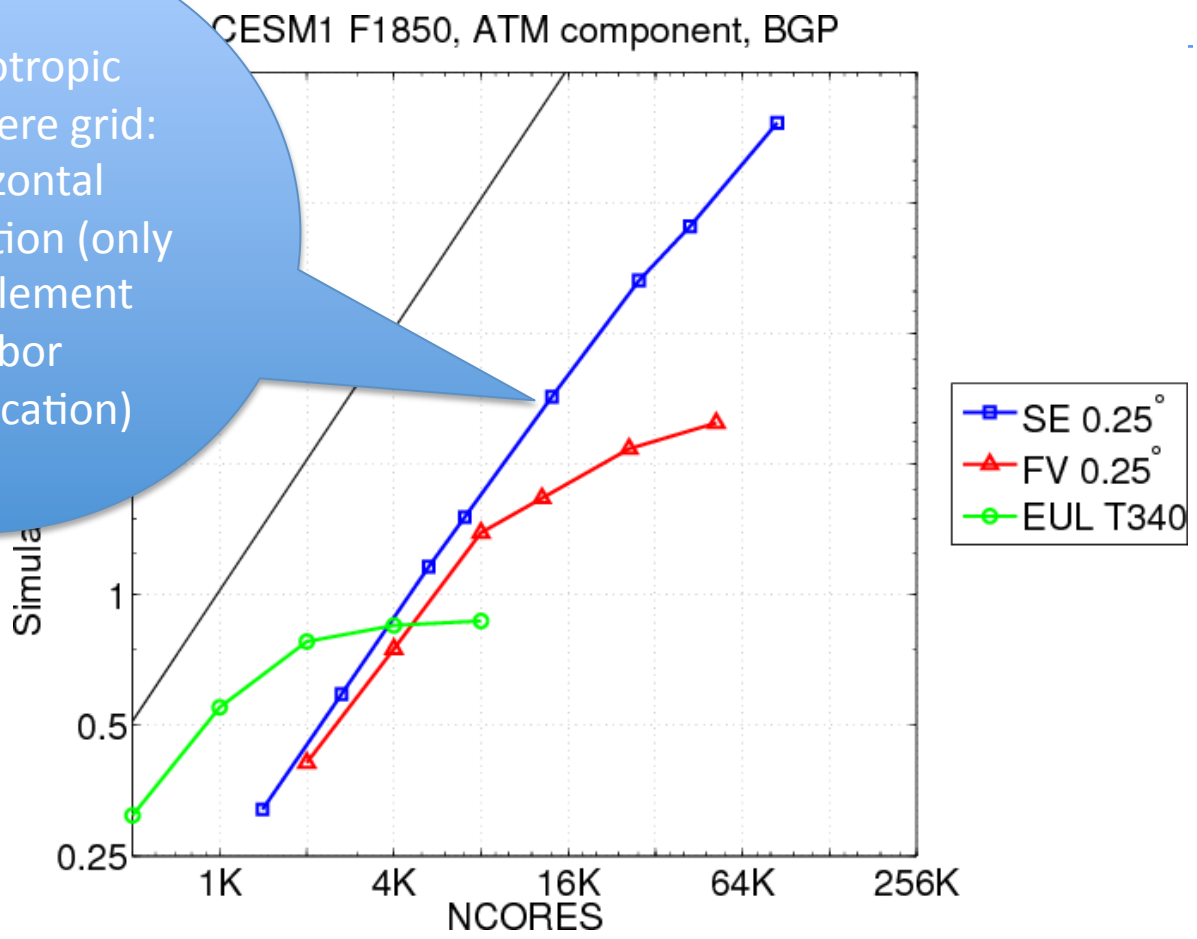


To do science > 5 SYPD

Plot: curtesy of Mark Taylor

A CESM example

Quasi-isotropic cubed-sphere grid: 2D horizontal decomposition (only nearest element neighbor communication)



To do science > 5 SYPD

Plot: curtesy of Mark Taylor

Why can't we continue on this "track", that is (in simple terms), add more and more CPUs?

- *"Power consumption is becoming one of the most important aspects of computing. It will be the most important driving force for supercomputing in the future. Without focusing on that, building bigger machines will be prohibitive. We're trying to understand which machines are more efficient, why they're more efficient, and understand the trends in high-performance computing." – Dongarra (founder of the top500 effort)*

Why can't we continue on this "track", that is (in simple terms), add more and more CPUs?

- Power consumption: e.g., the K-computer's power consumption is ~10 MW (~6 million $)

  For reference: A typical coal power station produces around 600–700MW. A typical unit in a nuclear power plant has an electrical power output of 900 - 1300 MW which can power ~1.000.000 homes

  The K-computer consumes as much power as ~10.000 homes!

- With 1000s and tens of thousands CPUs hardware failures can no longer be ignored (software challenge …)

Trend: Multi everywhere

- Since doubling frequency requires 8 times the power and increasing number of CPUs has its own set of problems, manufacturers are putting more cores on the chip (4,8,16, ..) and thousands of chips are combined in "traditional" massively parallel setup

  Downside: using multi-cores is complicated from a programming/ algorithm perspective (fundamental change in how algorithms are expressed)

- Challenge: getting data into the processor fast

- Barriers to progress are increasingly on the software side: hardware has a half-time measured in years; while software has a half-time measured in decades
- => **the high performance ecosystem is out of balance**

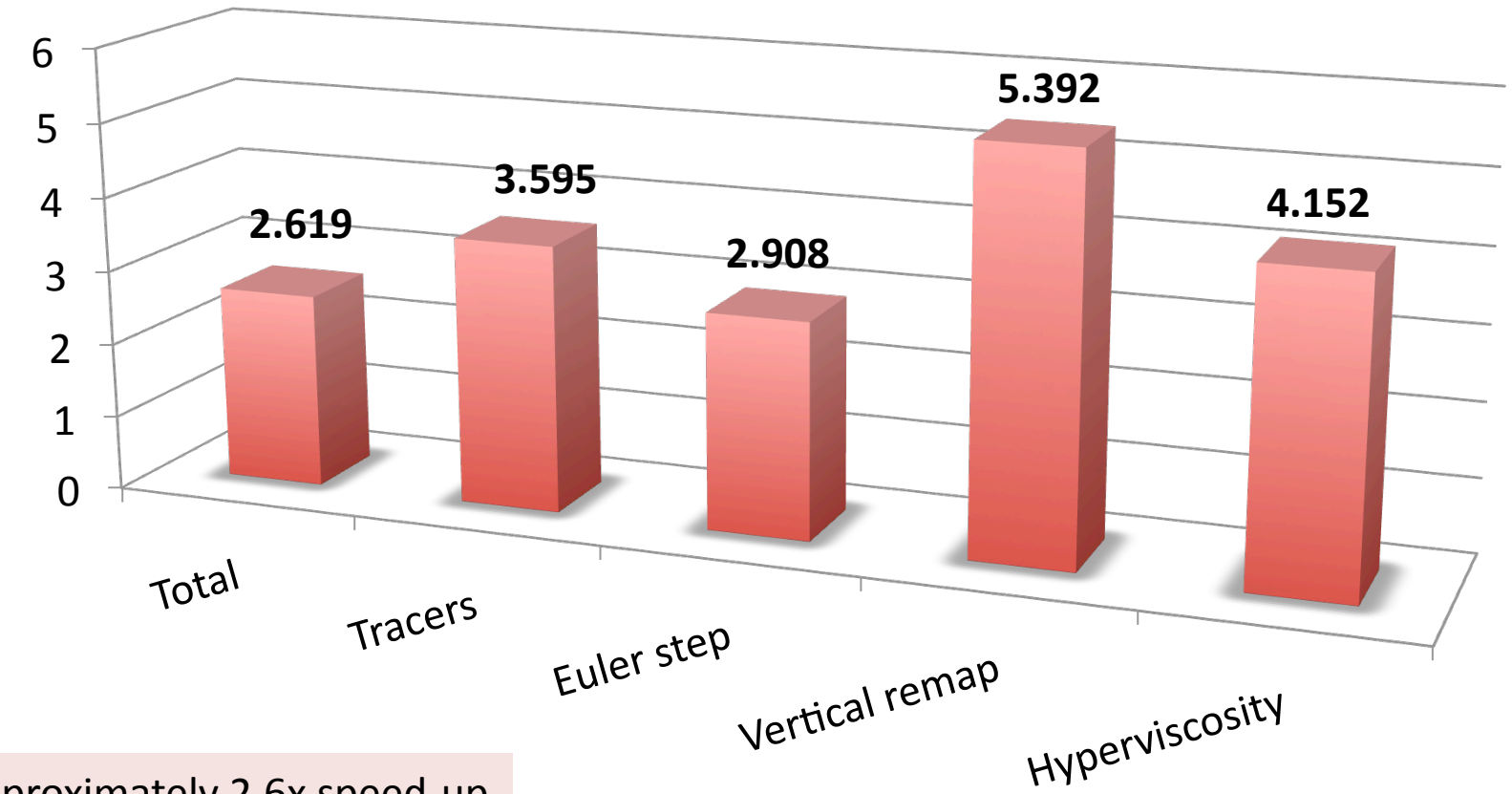
An example of multi-core: Graphics Processing Unit (GPU)

- GPU: collection of multi-core processors
  - Chunks of code are "launched" across these processors
  - Each core performs the same operation on different data
  - Can share data locally & sync only within a processor
  - No coordination between processors
  - Local cache can be user-managed, and it's very small
- GPUs have their own memory
  - Transfer to and from main memory with PCI-express bus
  - PCI-e is slow, so try to avoid it and keep data on-GPU
  - GPU memory is much slower than GPU computing
    - Access it in cohesive chunks & reuse as much as possible

Slide provided by Matthew Norman (ORNL)

CAM-SE dynamical core example of speed-Up: Fermi GPU vs 1 Interlagos / Node

- All PCI-e and MPI communication included



Approximately 2.6x speed-up

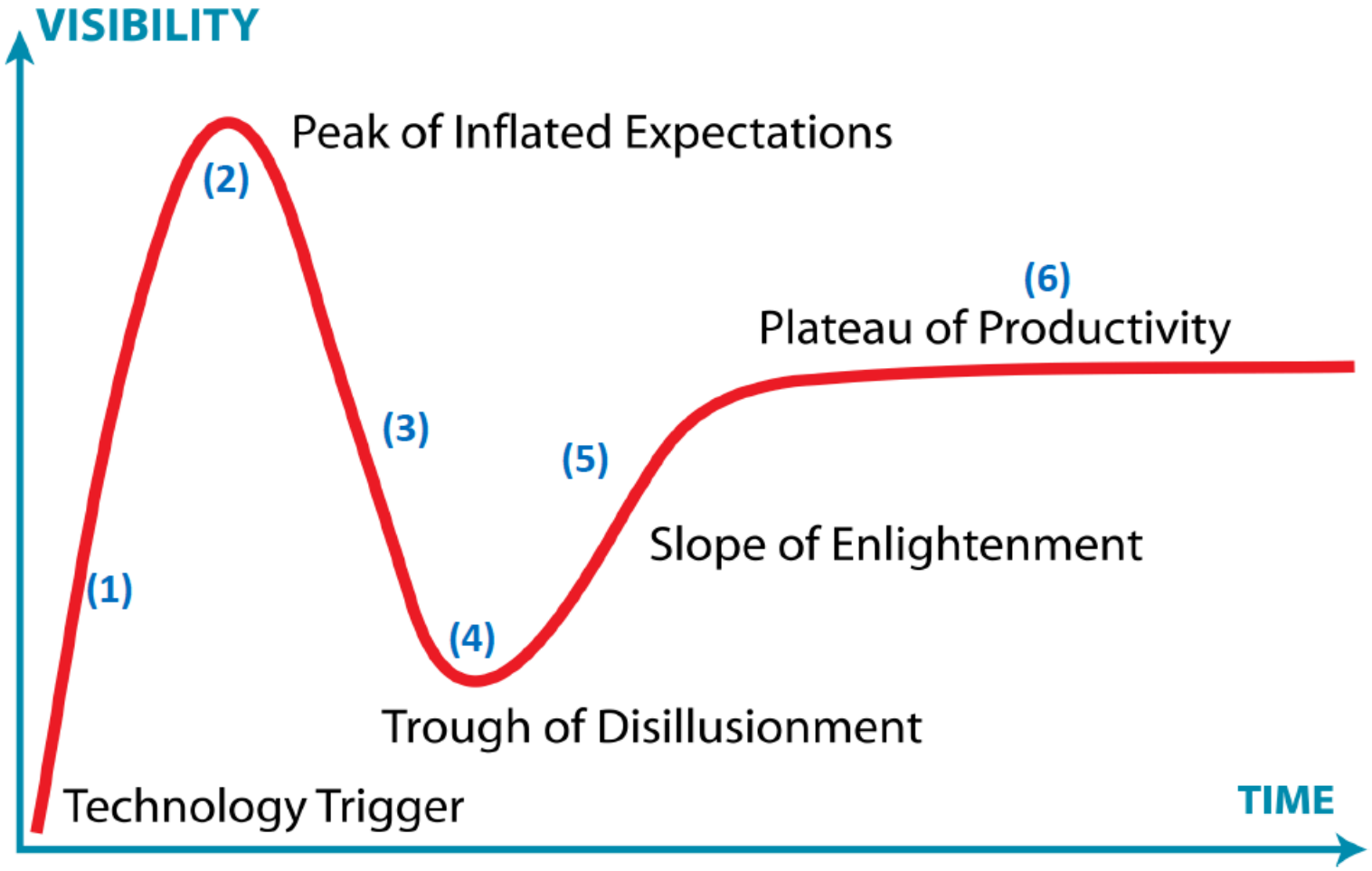
Slide provided by Matthew Norman (ORNL)

Only scratched the surface of high performance computing; some topics I did not talk about

- Concurrency when running coupled systems (keeping ocean, atmosphere, ice, land components in sync so that idle time is minimized)
- Storage: high resolution models produce enormous amounts of data
- Visualization and post-processing tools are increasingly challenged by the data amounts
- Software: CESM is run on small Linux clusters to the 3rd fastest computer in the world – challenge to optimize code for wide range of platforms
- Vendors are adding cores without increasing memory bandwidth – major challenge for climate modeling
- If special programming languages need to be used for multi-core architectures, scientists will need more expert software engineers to do coding!

  -in MPI programming the algorithm is still the same on each patch so scientists usually code themselves – this is not the case for multi-core applications with the present state of compilers and programming languages

Reality: where are we with many-core on the Gartner Hype cycle ?



NCAR's next supercomputing system: Yellowstone

- Specs:
  - 4,662 IBM dx360 M4 nodes – 16 cores, 32 GB memory per node
  - Intel Sandy Bridge EP processors with AVX – 2.6 GHz clock
  - 74,592 cores total – 1.552 PFLOPs peak
  - 149.2 TB total DDR3-1600 memory
  - 29.8 Bluefire (current NCAR machine) equivalents

Will go online later this summer …