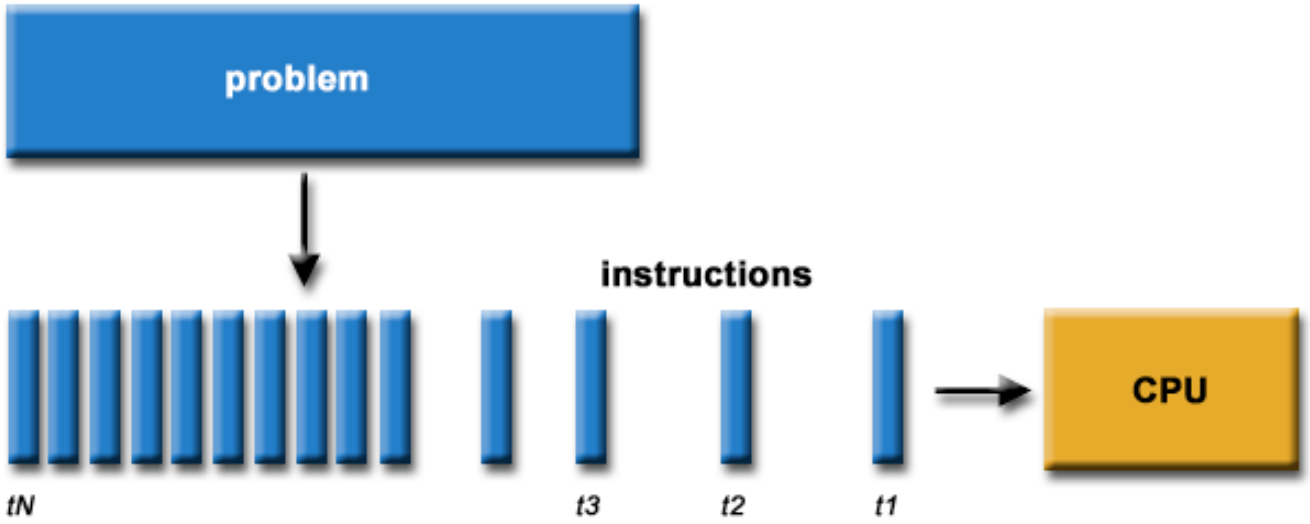
High Performance Parallel Computing: A Beginner's Guide

Chris H. Rycroft

SPAMS, October 2006

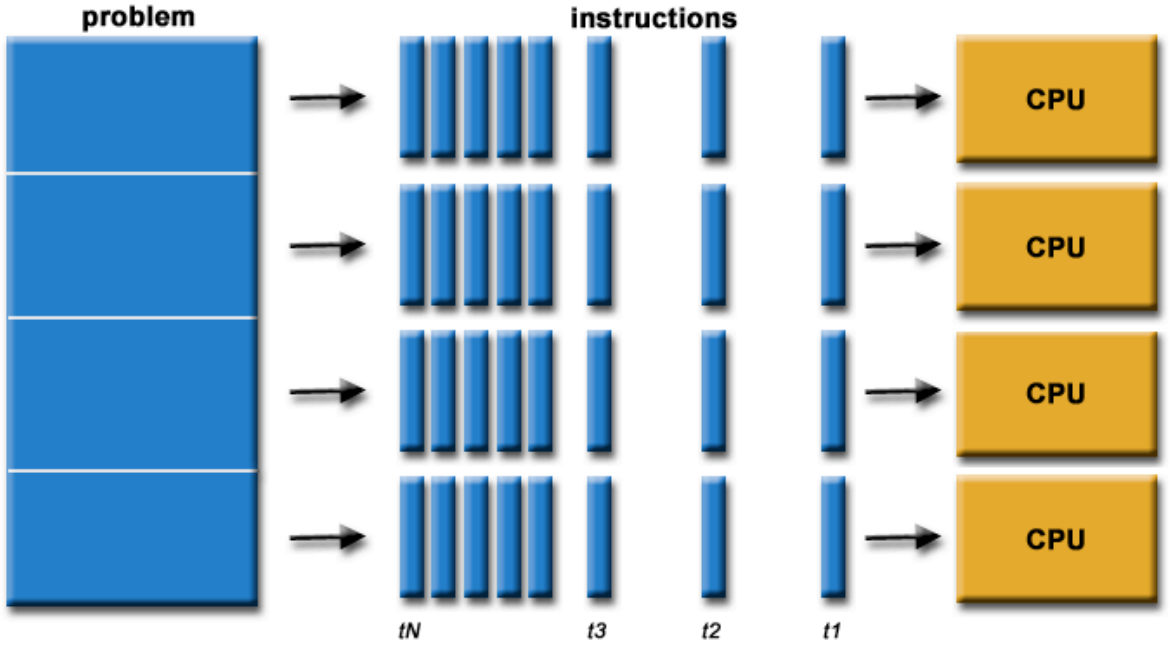
Parallel versus serial

- Traditional model of computing: a single processor executes instructions sequentially

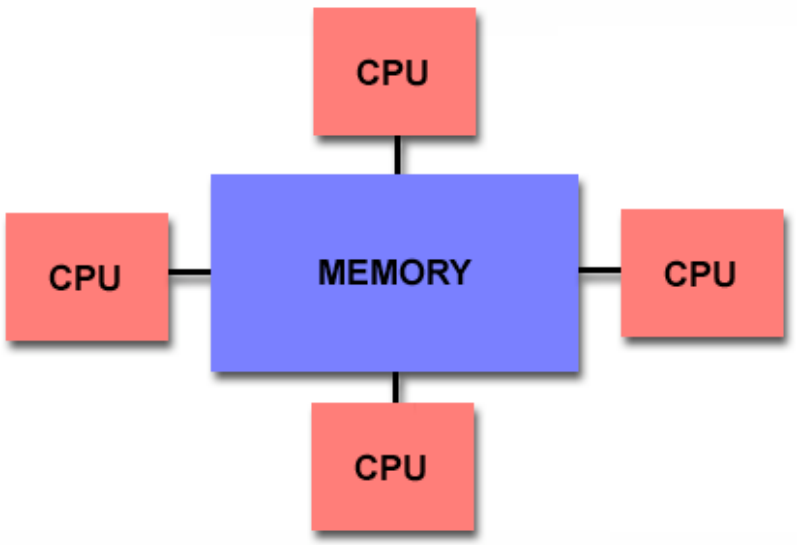


Parallel versus serial

- Parallel computer: execute multiple programs simultaneously

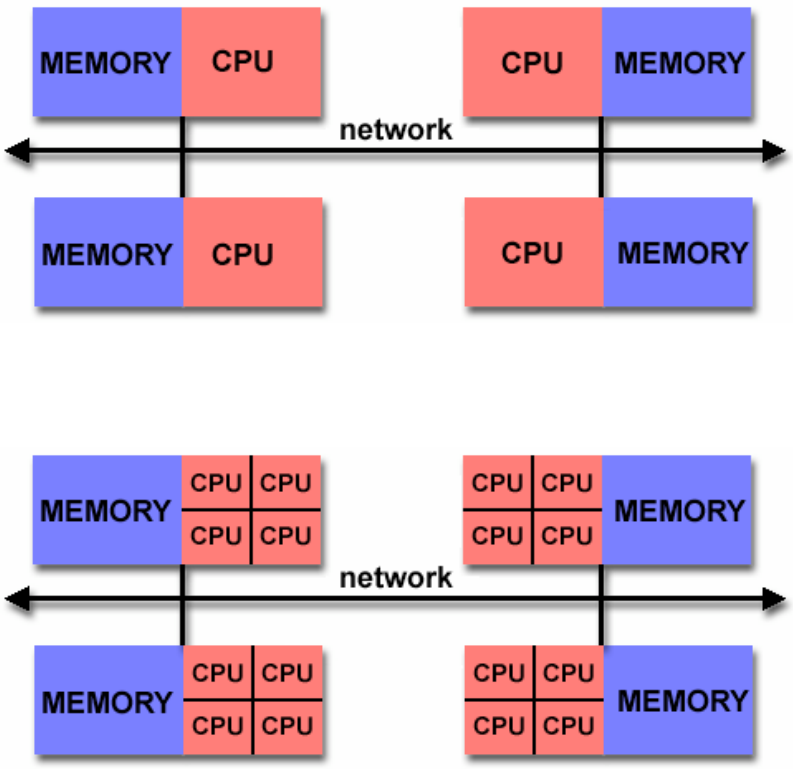


Shared Memory Machine



- Multiple CPUs access the same memory concurrently
- Typical of a dual/quad processor computer

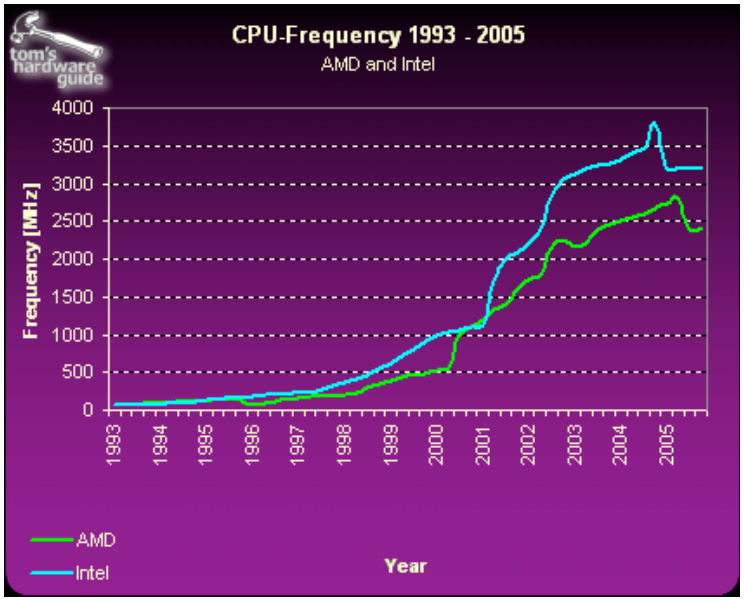
Distributed Memory Machine



- Each CPU has its own memory
- Each connected to a network (ethernet, internet, etc.)
- Can also hybrid model: most common for the fastest parallel machines

Trends towards parallel

- Moore's law breaking down
- Absolute physical limits on the speed of one processor (9cm/ns for signals in copper wire)
- Manufacturers switching to multiple core processors, and advertising it!

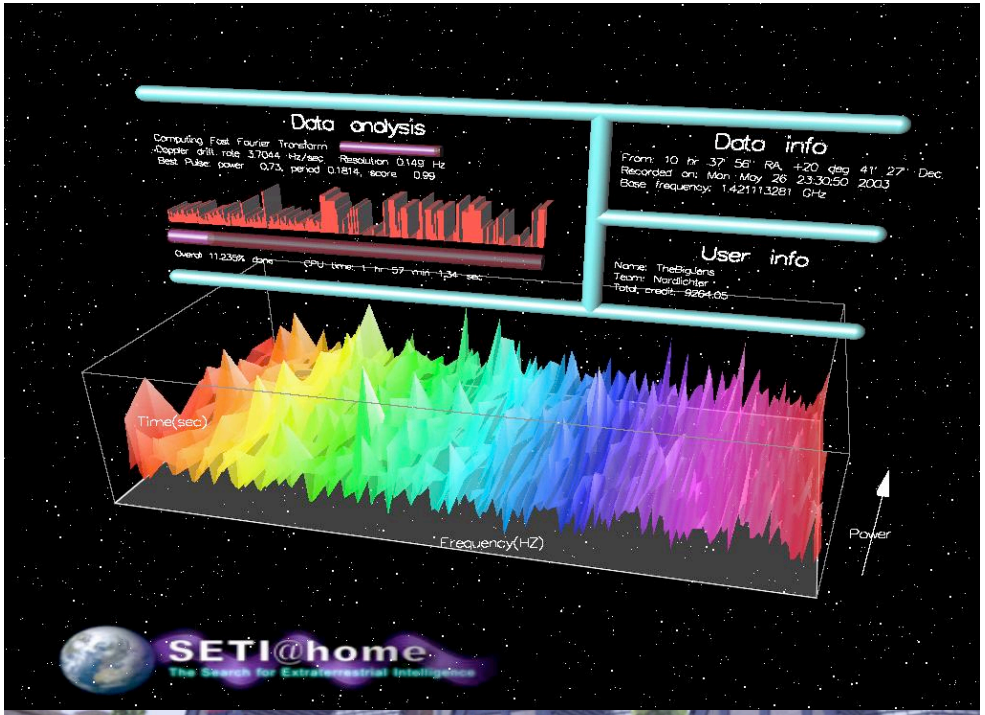


Disadvantages of parallel

- Always a communication overhead
- One worker can dig a hole in six hours, but six workers can’t dig the hole in one hour
- Try and split up the computation as efficiently as possible

“Embarrassingly parallel”

- Some tasks can very easily be split into independent chunks
- Each can be done in series
- Examples:
  - SETI@Home
  - Folding@Home
  - Pixar/Dreamworks



TOP500 - www.top500.org

- "The TOP500 project was started in 1993 to provide a reliable basis for tracking and detecting trends in high-performance computing"
- Releases lists of the 500 most powerful computers in the world every six months
- Measures rate of execution of FLoating point OPerations, or "Flops".

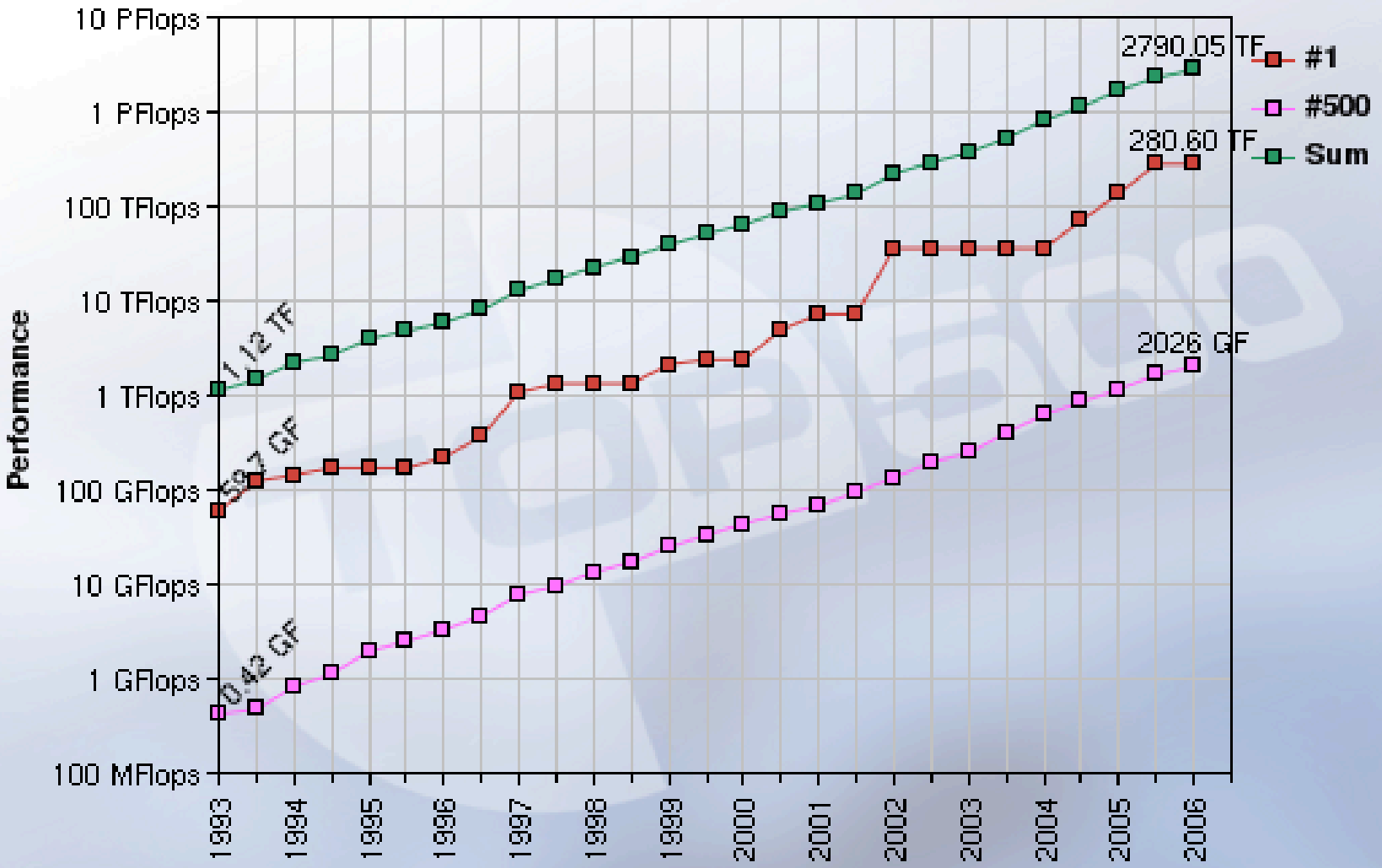
LINPACK Benchmark

- LINPACK: a standard numerical linear algebra library
- Solve a dense set of n equations using LU factorization with partial pivoting
- Takes $2/3\ n^3 + O(n^2)$ operations
- Test different values of n, and find when the Flop execution rate is maximized: RMax
- Also report RPeak: the theoretical peak performance of the system

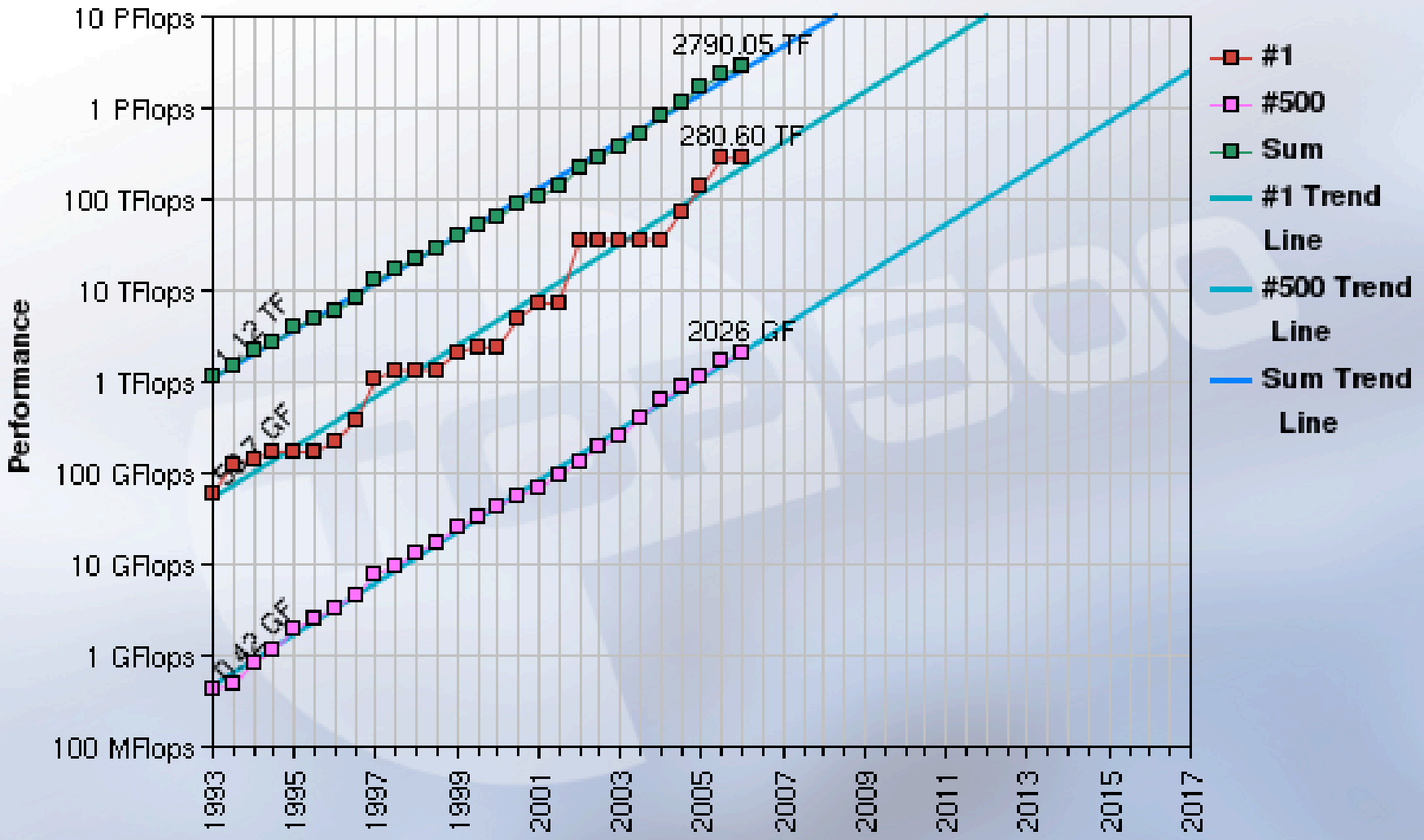
Top 10

Rank	Site	Man.	Computer	Country	Year	Procs	RMax	RPeak	Nmax
1	DOE/NNSA/LLNL	IBM	eServer Blue Gene Solution	USA	2005	131072	280600	367000	1769471
2	IBM Thomas J. Watson Research Center	IBM	eServer Blue Gene Solution	USA	2005	40960	91290	114688	983039
3	DOE/NNSA/LLNL	IBM	eServer pSeries p5 575 1.9 GHz	USA	2006	12208	75760	92781	1383600
4	NASA/Ames Research Center/NAS	SGI	SGI Altix 1.5 GHz, Voltaire Infiniband	USA	2004	10160	51870	60960	1290240
5	Commissariat a l'Energie Atomique (CEA)	Bull SA	NovaScale 5160, Itanium2 1.6 GHz, Quadrics	France	2006	8704	42900	55705.6	
6	Sandia National Laboratories	Dell	PowerEdge 1850, 3.6 GHz, Infiniband	USA	2006	9024	38270	64972.8	
7	GSIC Center, Tokyo Institute of Technology	NEC/Sun	Sun Fire X4600 Cluster, Opteron 2.4/2.6 GHz, Infiniband	Japan	2006	10368	38180	49868.8	1334160
8	Forschungszentrum Juelich (FZJ)	IBM	eServer Blue Gene Solution	Germany	2006	16384	37330	45875	663551
9	Sandia National Laboratories	Cray Inc.	Red Storm Cray XT3, 2.0 GHz	USA	2005	10880	36190	43520	1100000
10	The Earth Simulator Center	NEC	Earth-Simulator	Japan	2002	5120	35860	40960	1075200

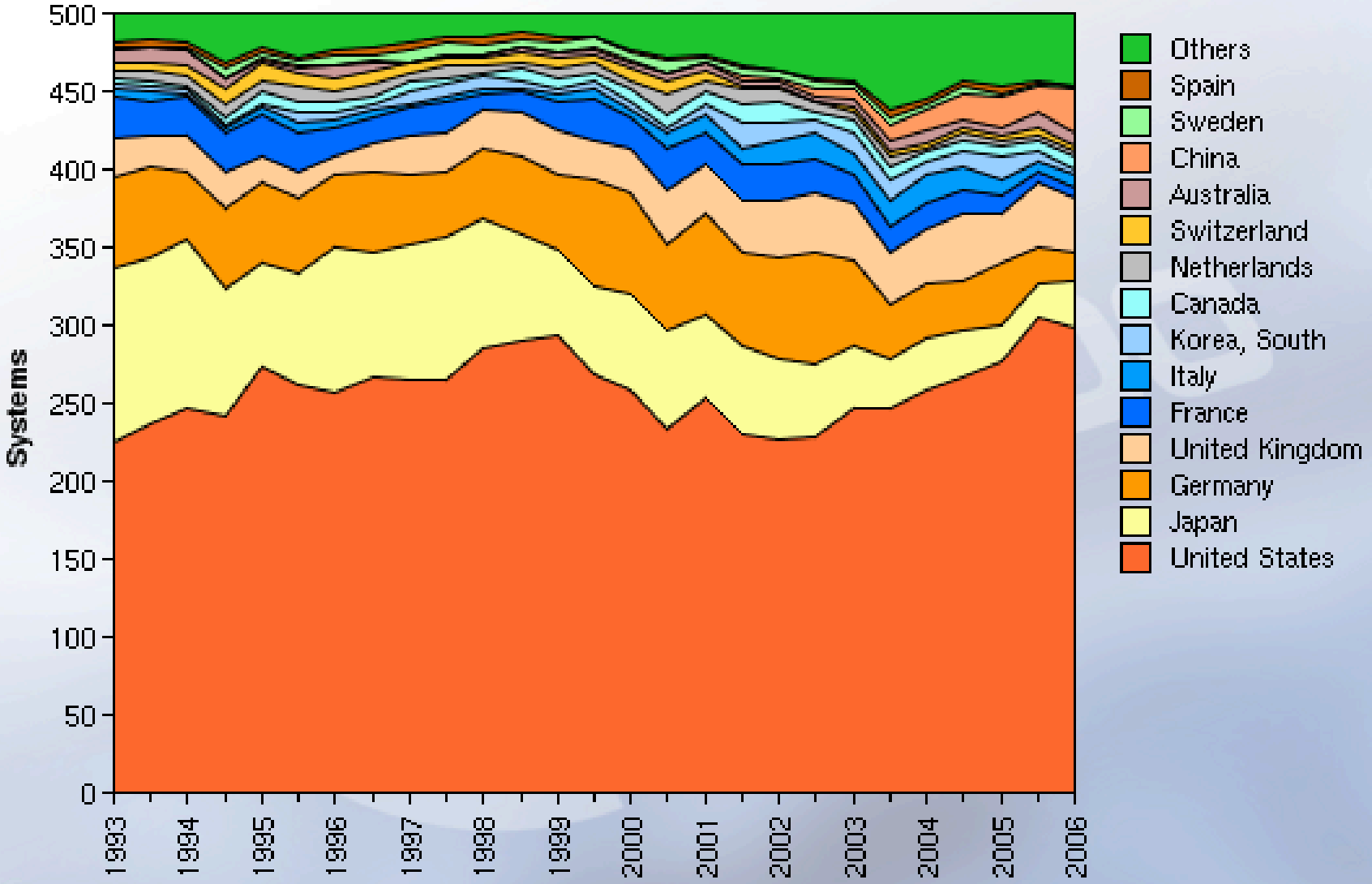
Performance Development



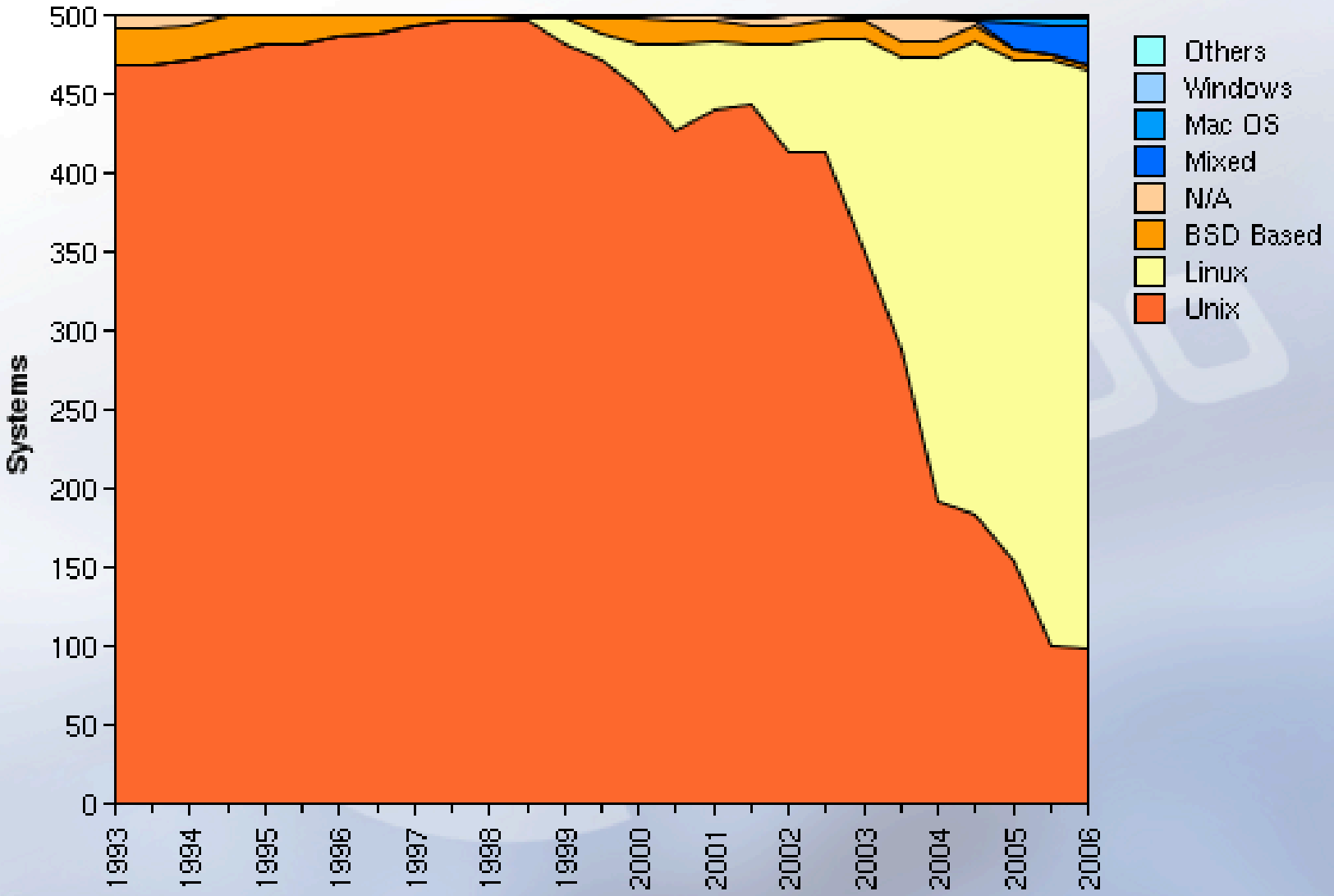
Projected Performance Development



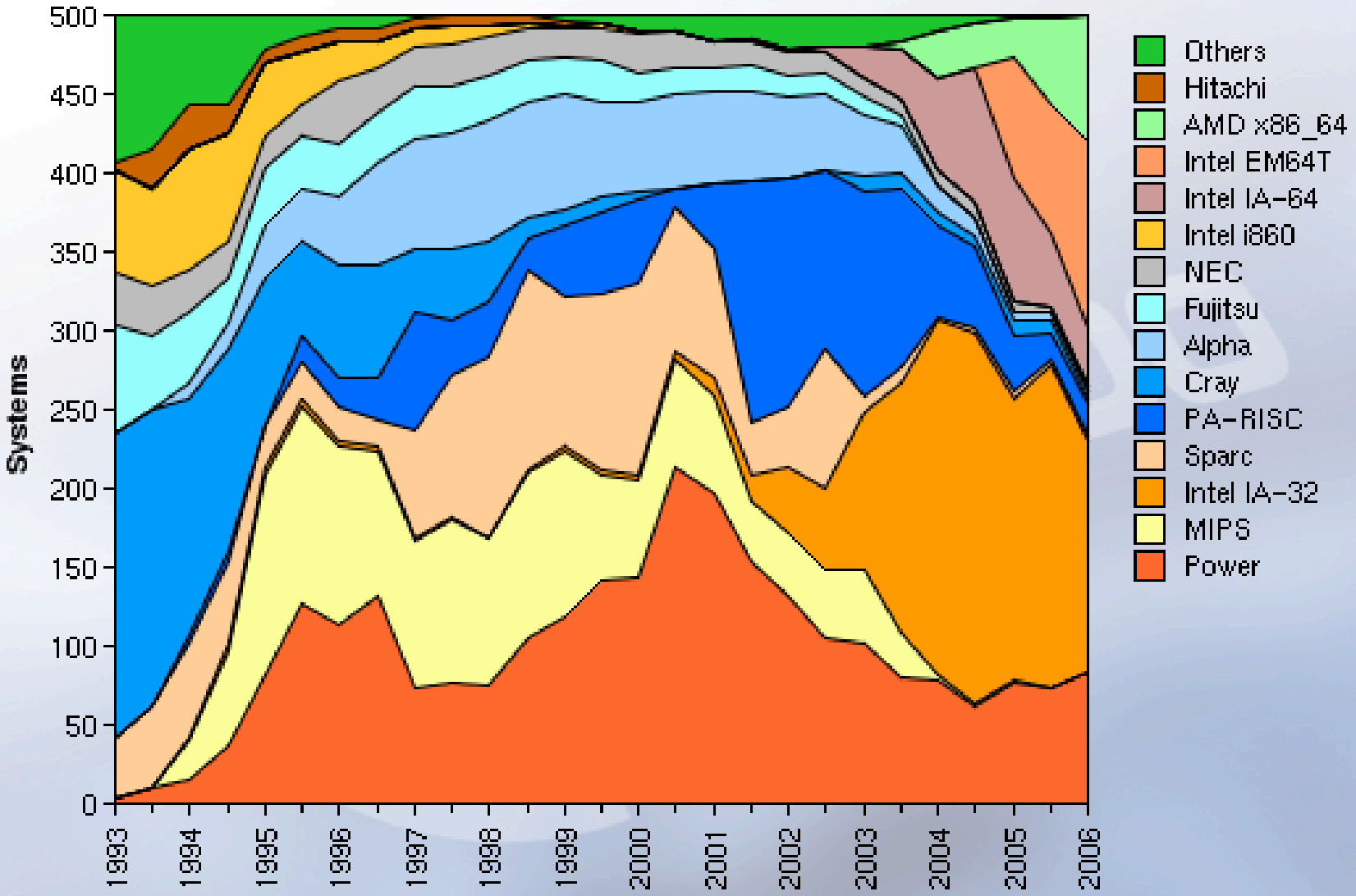
Countries / Systems



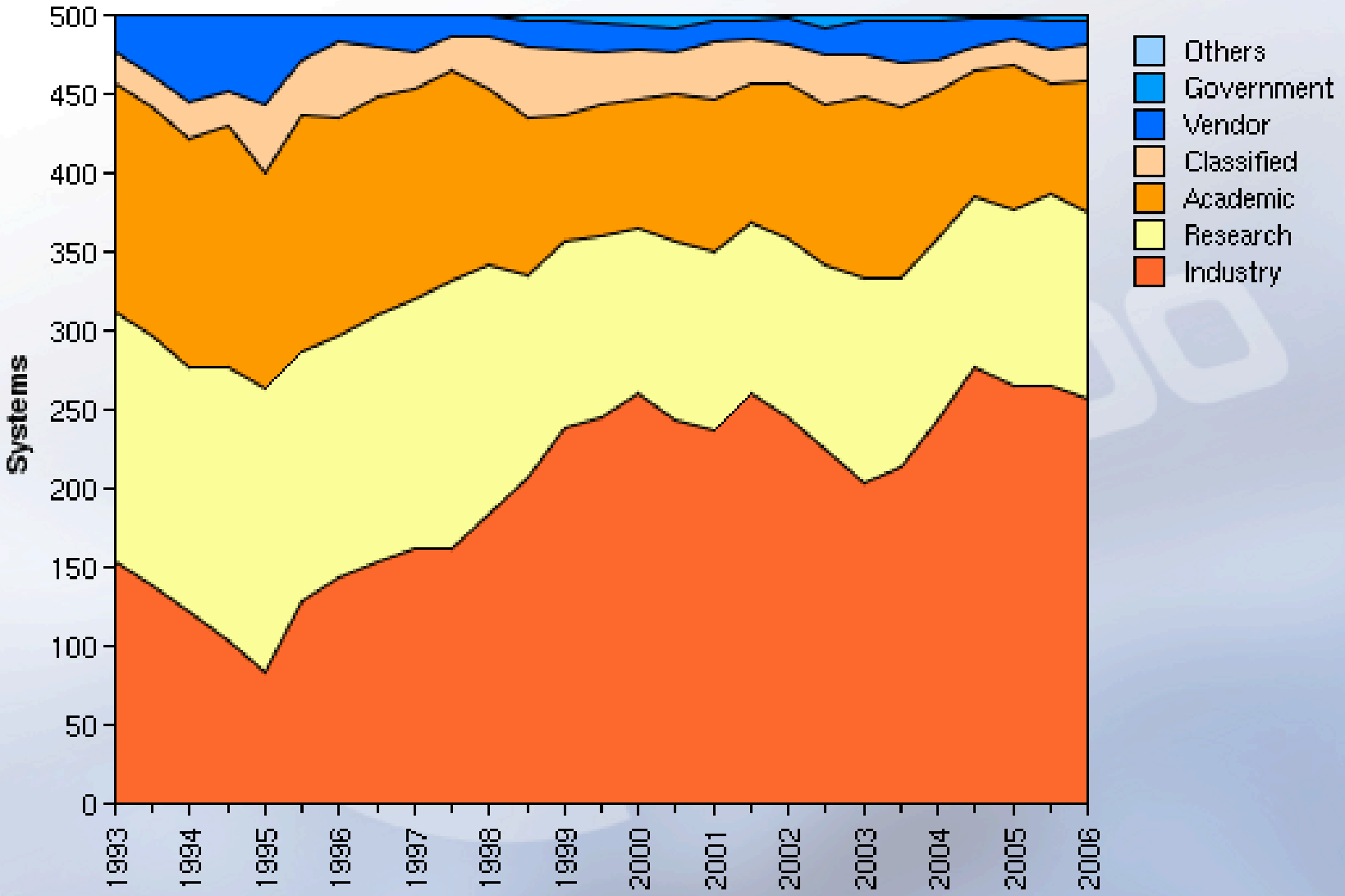
Operating System / Systems



Processor Family / Systems



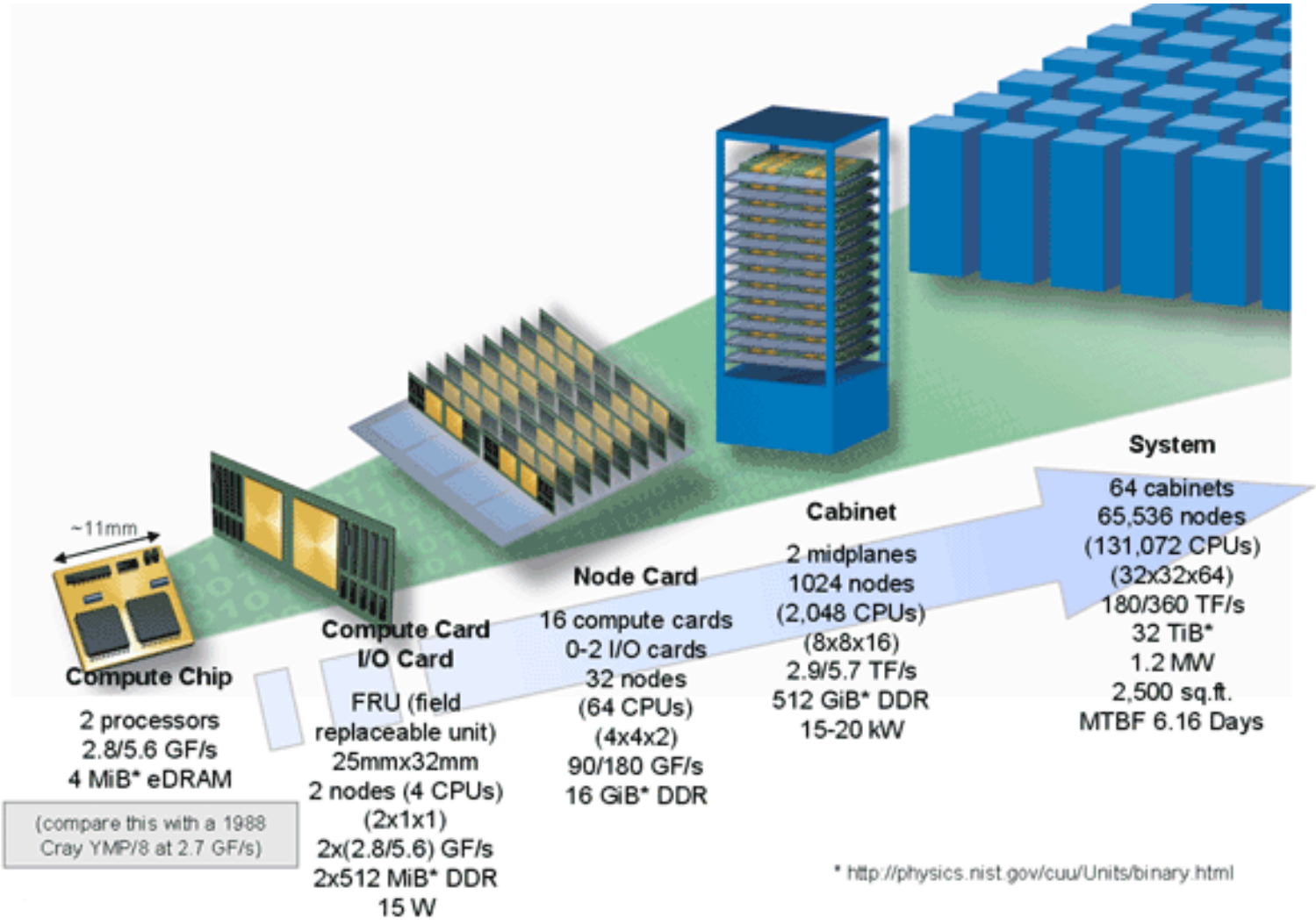
Customer Segment / Systems

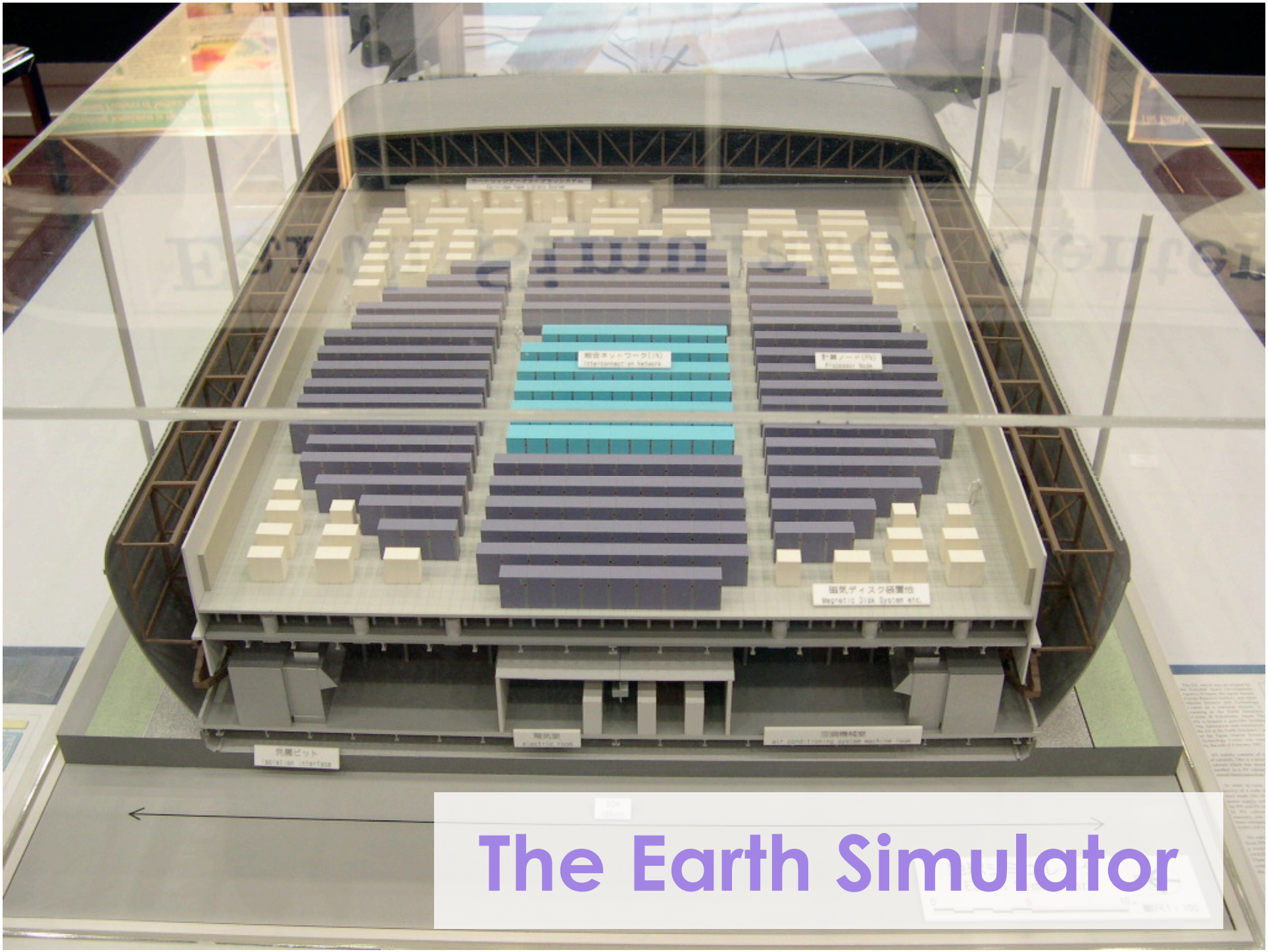


Blue Gene: the fastest computer in the world

- 2^{16} dual processors
- Interconnected in a three-dimensional torus network
- 33TB of main memory
- Several implementations of this architecture

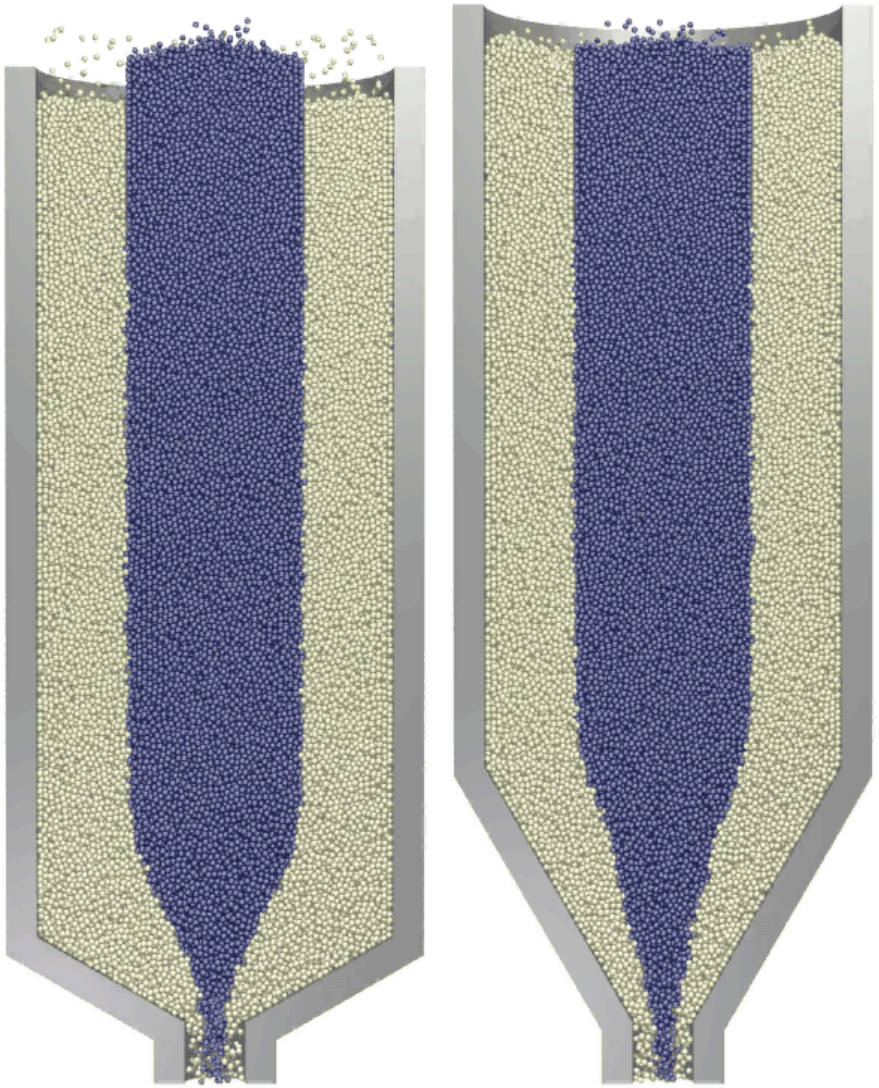
Blue Gene hierarchy




The Earth Simulator

Example #1: Large-scale atomistic modeling

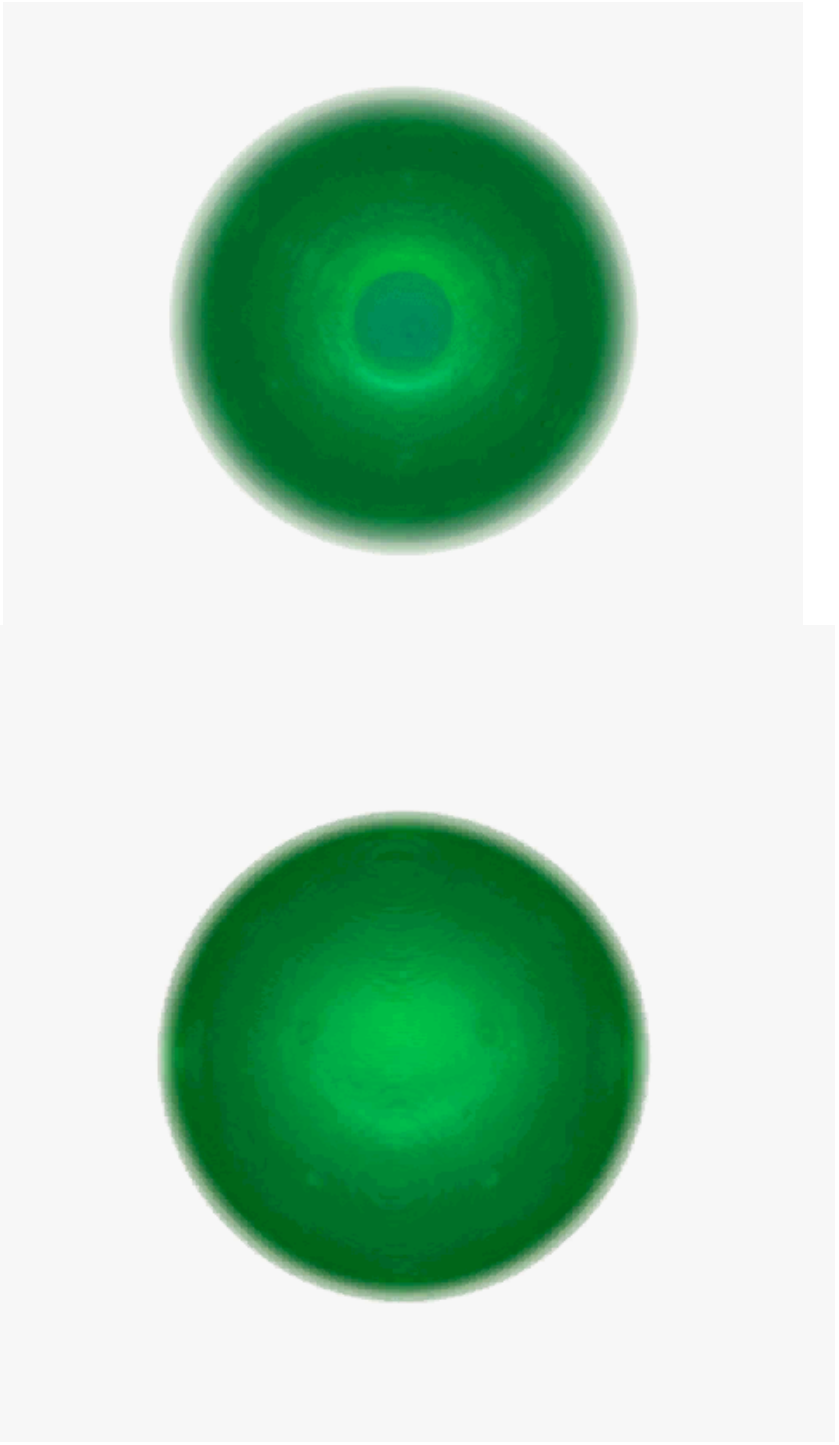
- Large-scale Atomic/ Molecular Massively Parallel Simulator
- http://lammps.sandia.gov
- Models a huge variety of molecular simulations
- Domain decomposition; very efficient
- 450,000 granular particles in a pebble-bed reactor, run on Sandia's Xeon cluster



C. H. Rycroft et al., Phys. Rev. E **74**, 021306 (2006).

Example #2: Supernovae collapse

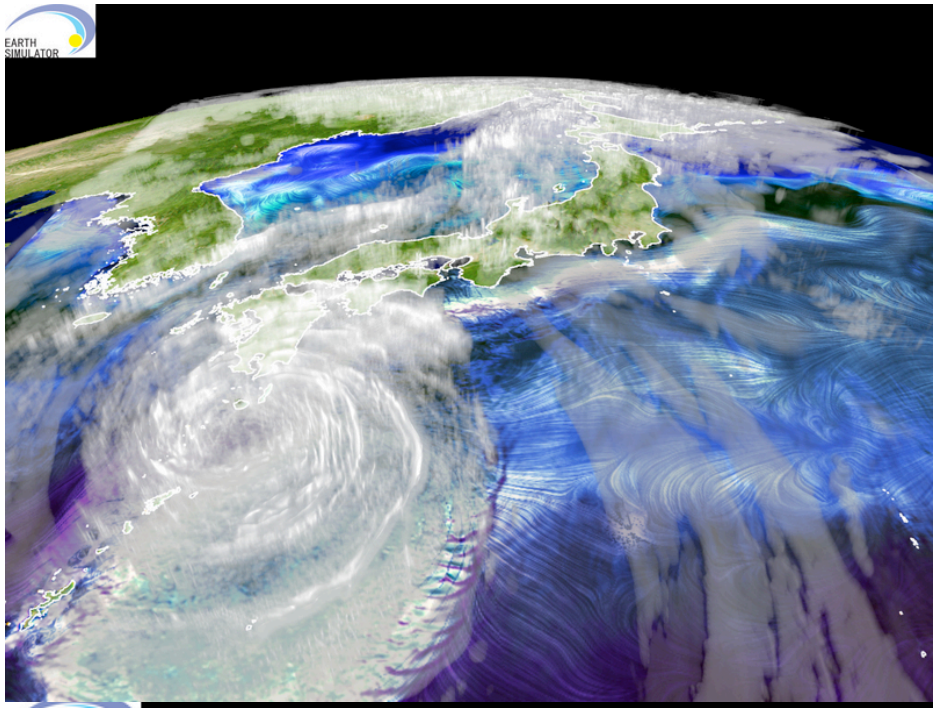
- Terascale Supernova Initiative
- http://www.phy.ornl.gov/tsi
- 3D simulations show behavior not seen in 2D simulations

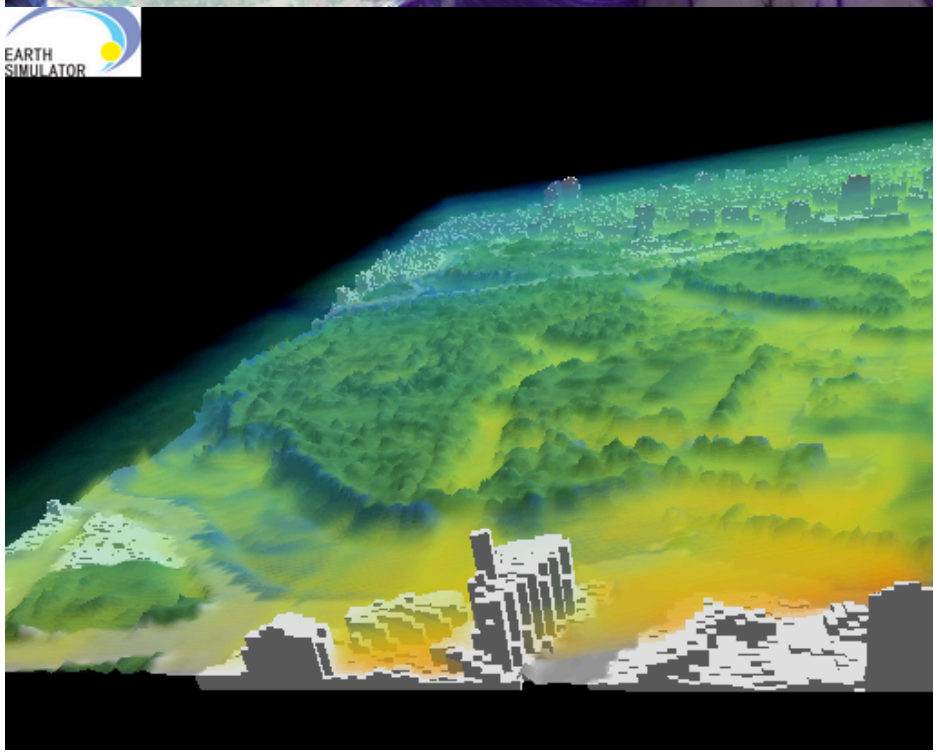


J. M. Blondin et al., Ap. J. **584**, 971-980 (2003).

Example #3: Weather simulations

- Carried out on the Earth Simulator
- http://www.es.jamstec.go.jp/esc
- An enormously complicated system; many coupled processes





Parallel computing for the masses

- Build a parallel computer using off-the-shelf components
- First done by Thomas Sterling and Donald Becker, NASA
- 16 100MHz processors, 16MB ram each, 10Mb ethernet
- Named it "Beowulf"

Beowulf Clusters

- Became a general term for cluster
- We have a 32 processor Beowulf called the AMCL in the basement
- Buying factors
  - Processors
  - Memory
  - Software
  - Interconnect

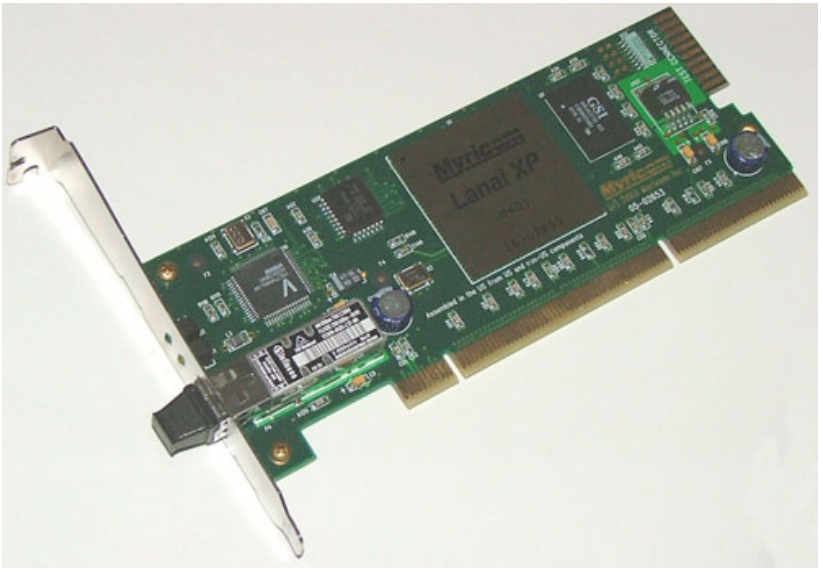
"In off the moors, down through the mist bands, god-cursed Grendel came greedily loping. The bane of the race of men roamed forth, hunting for prey in the high hall."

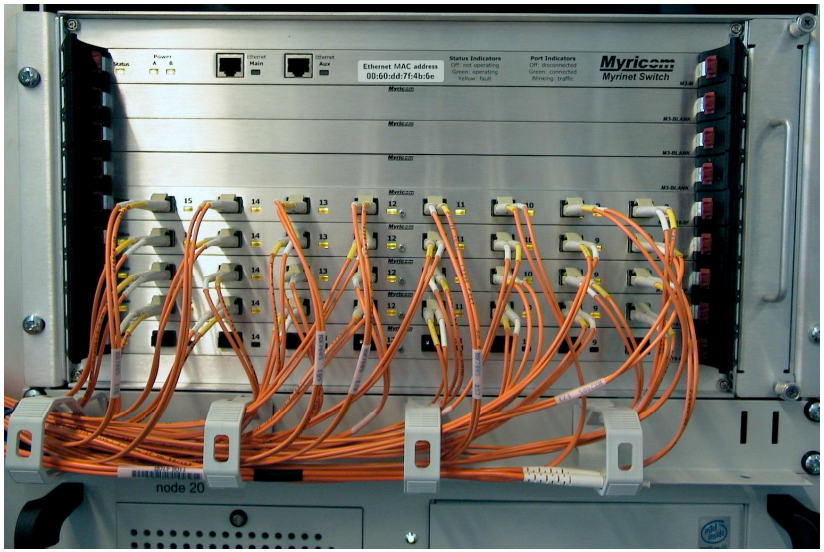
Interconnects: Ethernet

- Extremely popular, and fairly cheap
- Use gigabit internet for clusters
- Latency: 120 microseconds
- Bandwidth: 1 Gb/s

Interconnects : Myrinet

- Expensive, but very fast and popular
- Fiber-optic connections
- Network cards can bypass OS and talk directly to processes
- Latency: 2 microseconds
- Bandwidth: 10Gb/s





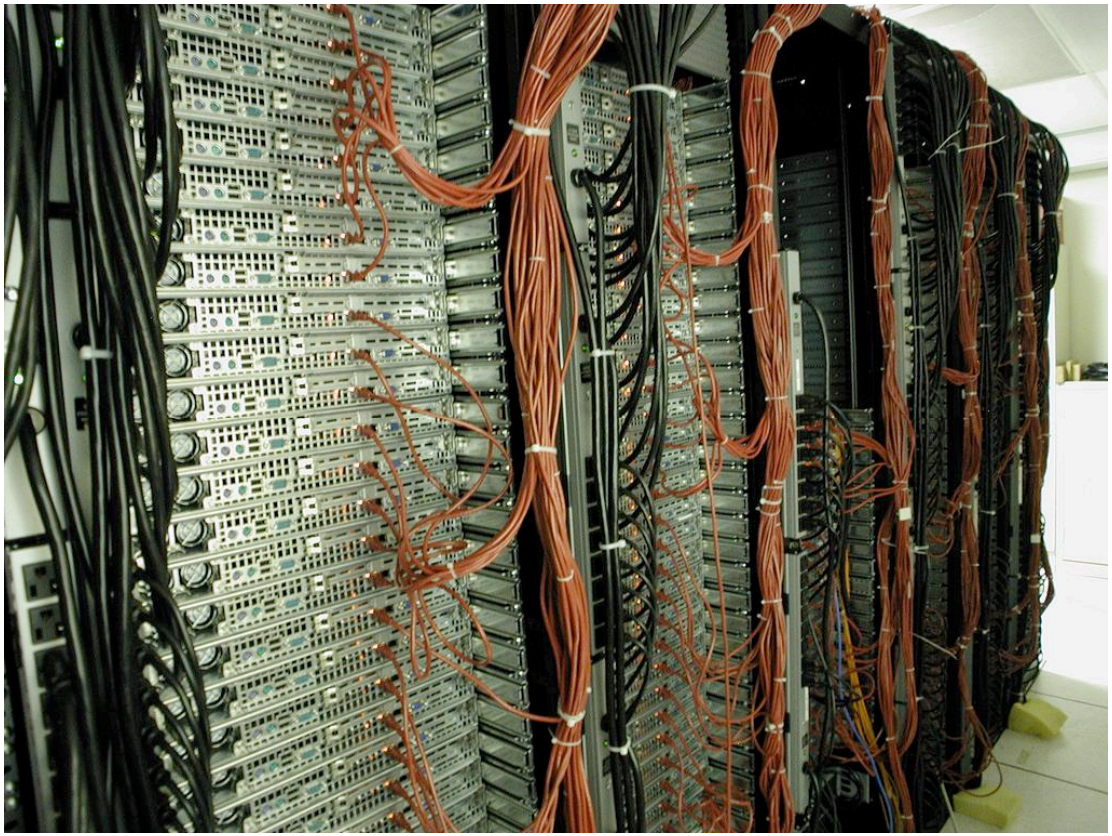
Software components

- Linux/FreeBSD; some versions specifically
- PBS queue management system
- Ganglia cluster toolkit

http://amcl.mit.edu/ganglia

Practical considerations

- Space
- Rackmount
- Power
- Cables
- Ventilation
- Noise



Parallel programming

- A wide variety of parallel programming languages exist
- Range from being very high level, to complete control
- Autoparallelization: let the compiler do it for you

A possible autoparallelization candidate:

```
void main () {
        int i;
        for(i=0;i<1000;i++) {
                hard_func(i);
        }
}
```

High-level scientific languages

- Star-P: a parallel version of Matlab
- Very similar, but matrices are distributed across parallel processors
- Most work is hidden from the user
- Other similar products: gridMathematica, HPC-Grid for Maple

Cilk

- A parallel extension to the C programming language
- Primarily shared-memory machines
- Developed at MIT: http://supertech.lcs.mit.edu/cilk/
- Simple commands to spawn functions to other processors

C and Cilk for Fibonacci

```
int fib (int n) {
    if (n<2) return n;
    else {
        int x,y;
        x=fib(n-1);
        y=fib(n-2);

        return (x+y);
    }
}
```

```
cilk int fib (int n) {
    if (n<2) return n;
    else {
        int x,y;
        x=spawn fib(n-1);
        y=spawn fib(n-2);
        sync;
        return (x+y);
    }
}
```

- Extra keywords cause function calls to be spawned to new processors
- Removing Cilk keywords gives back a serial program

Message Passing Interface (MPI)

- A widely used library for low-level access
- Can be used in many programming languages (C, C++, Fortran, etc.)
- Provides commands for message passing between processors
- http://www.lam-mpi.org/
- http://www-unix.mcs.anl.gov/mpi/mpich2/

Hello World!

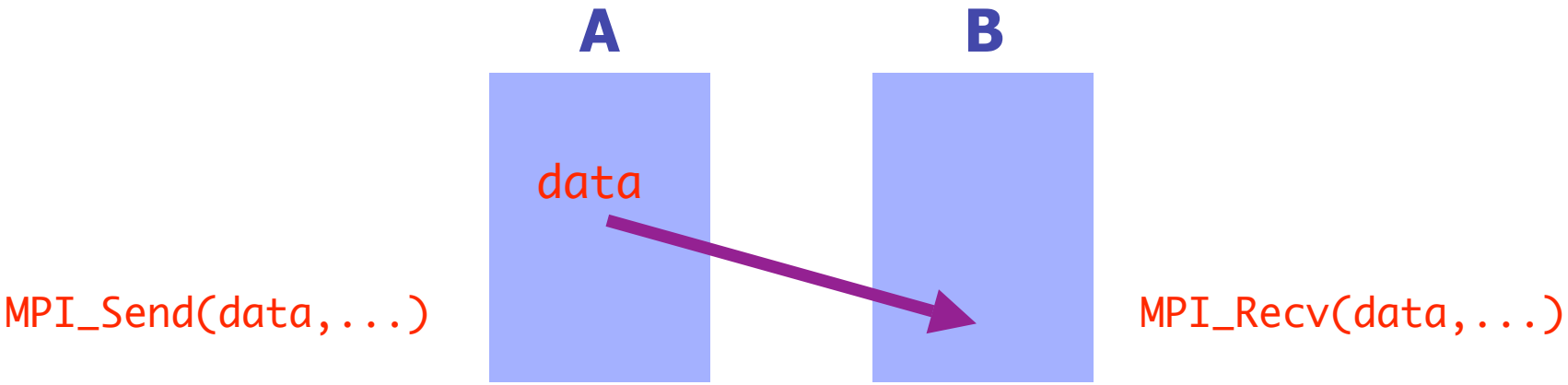
```
#include <mpi.h>

void main(int argc, char *argv[]) {
    int me, nprocs;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &me);
    printf("Hello from node %d of %d\n", me, nprocs);
    MPI_Finalize();
}
```

Program output:

```
Hello from node 2 of 4
Hello from node 3 of 4
Hello from node 0 of 4
Hello from node 1 of 4
```

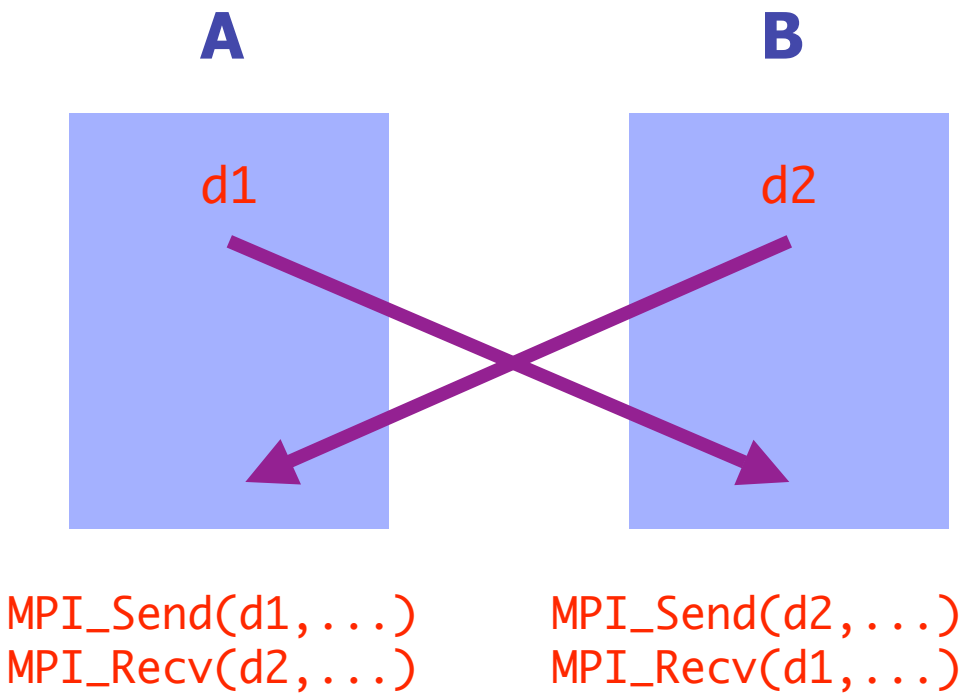
Passing a message

- Use the commands MPI_Send and MPI_Recv
- Can send an arbitrary chunk of data
- Usually faster to send one large message than lots of small ones



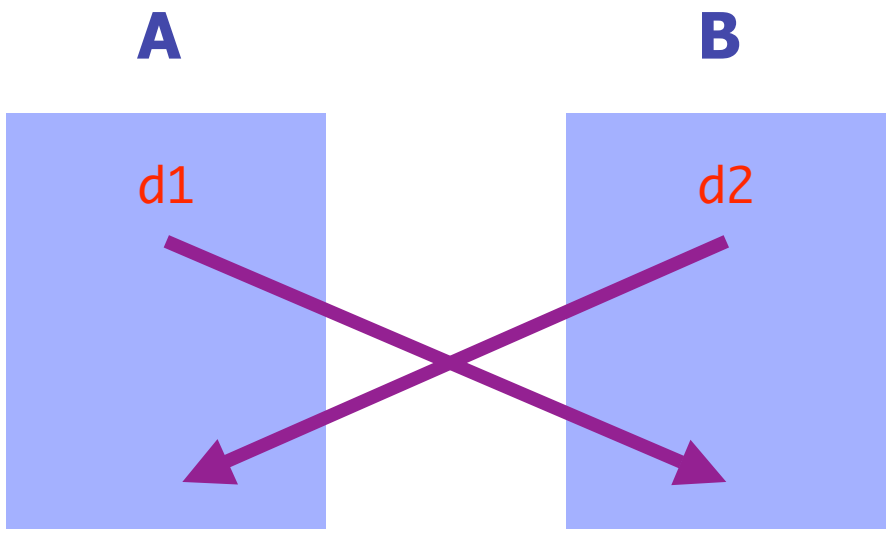
Data exchange

- Problems occur when data is exchanged between processes
- The send commands only complete when the data is received



Data exchange

- Use the command MPI_Isend to send data and not wait for a response
- Once the MPI_Recv is done, call MPI_Wait to wait for the Isend to complete



More advanced commands

- Use MPI_Barrier for synchronization: all processes must enter the barrier before leaving it
- Use MPI_Bcast to send messages to all nodes
- Use MPI_Allreduce to collect data from all nodes simultaneously

A parallel cumulative sum

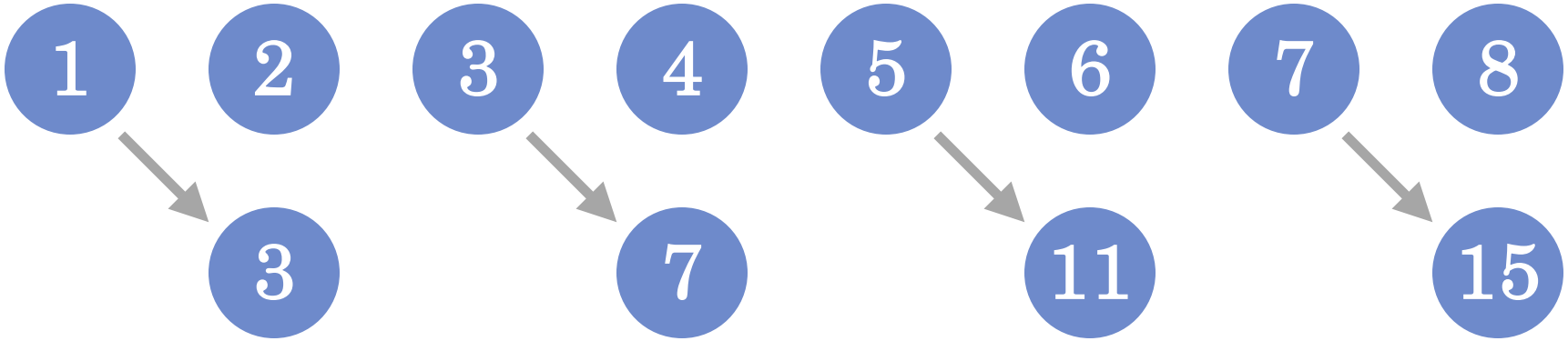
- Let $\{x_i\}$ be a set of numbers for $i = 0, 1, \ldots, M-1$ where $M = 2^N$
- We want to compute the cumulative sums

$$y_i = \sum_{j=0}^{i} x_j$$

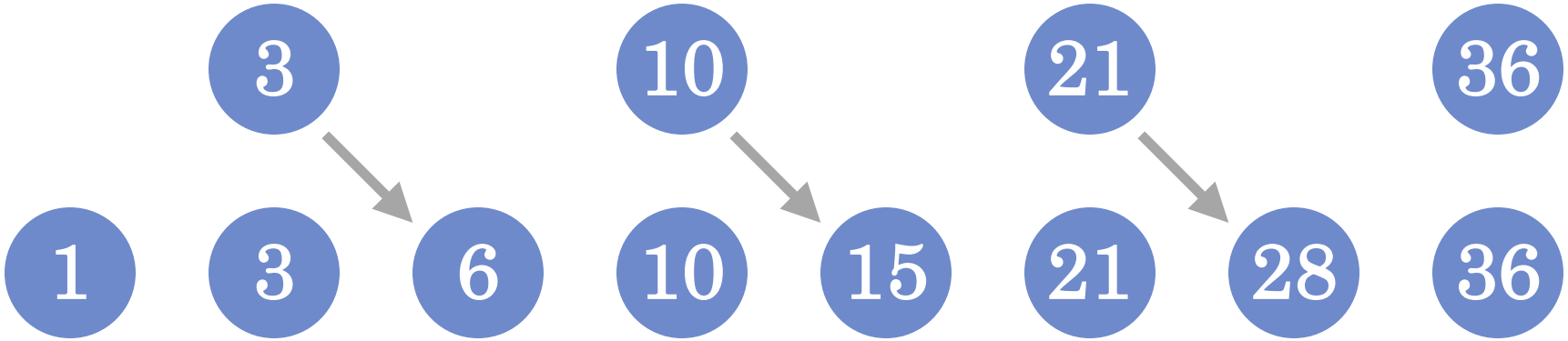
- Do in parallel?

```
y[0]=x[0];
for(i=1;i<M;i++) {
    y[i]=y[i-1]+x[i];
}
```

Parallel Prefix



(Recurse)

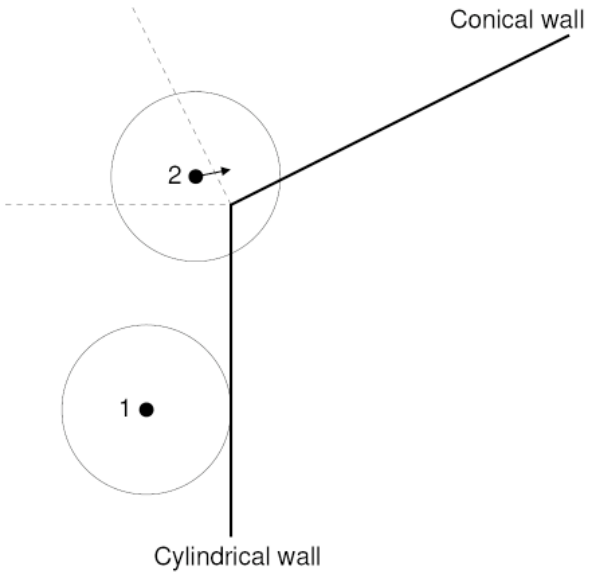


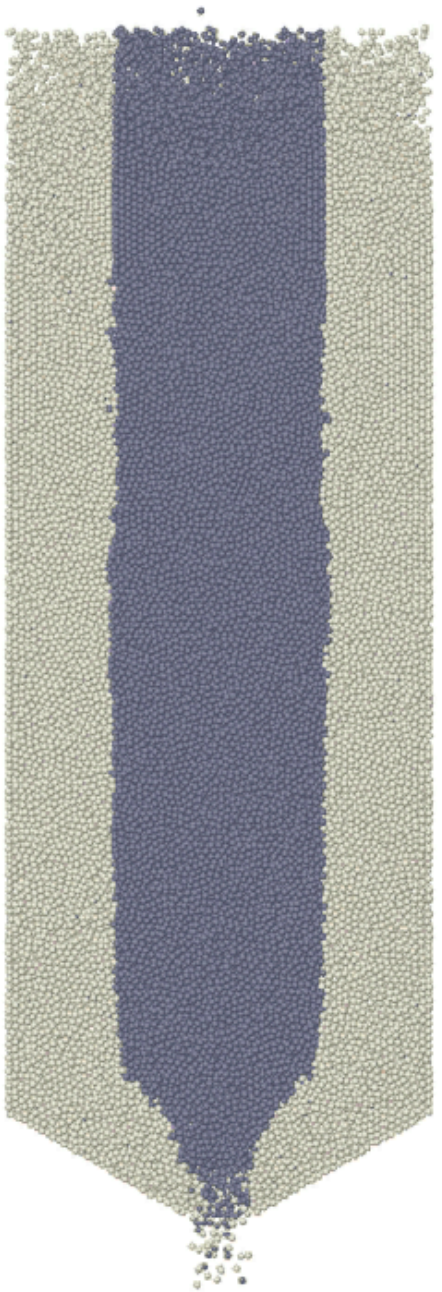
Parallel Prefix #2

- Total time is O(log N)
- Works for any associative operator
- Can be generalized to N which aren't powers of two
- Can be generalized when the number of processors is less than N

A lesson

- First set of data from Sandia
- Looks okay?





More issues