Lecture 19

Architectural Directions

Today's lecture

- Advanced Architectures
  - NUMA
  - Blue Gene

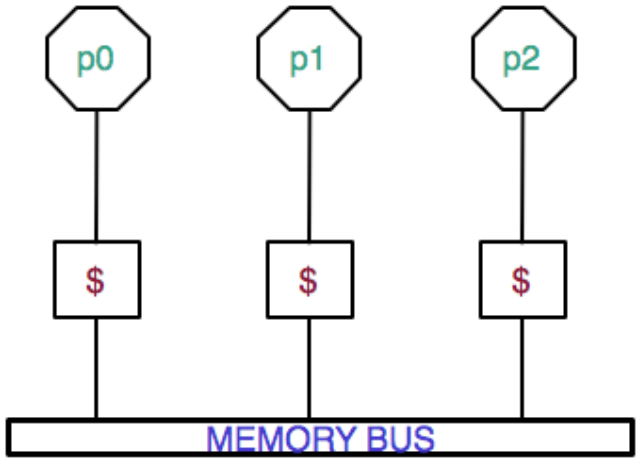
Announcements

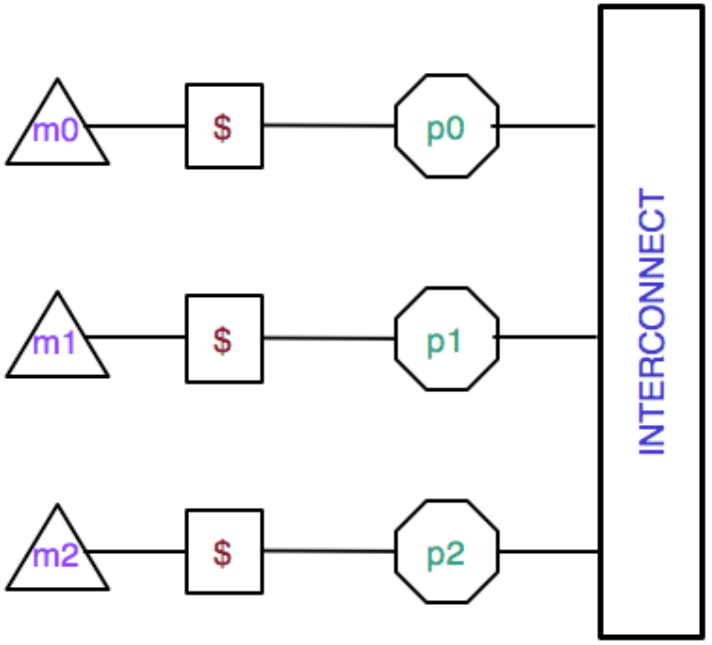
- Final examination
  - Thursday, March 17, in this room: 3pm to 6pm
  - You may bring your textbook and one piece of notebook sized paper
- Office hours during examination week
  - Wednesday
    - 11AM to 12 noon
    - 4pm to 5pm
  - Or by appointment

NUMA Architectures

NUMA Architectures

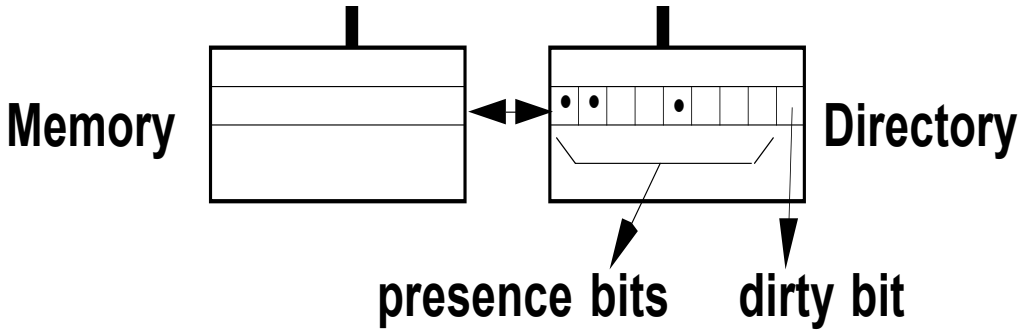
- Address space is global to all processors
- Distributed shared memory
- A **directory** keeps track of sharers
- Point-to-point messages manage coherence
- Stanford Dash, SGI UV, Altix, Origin 2000





Inside a directory

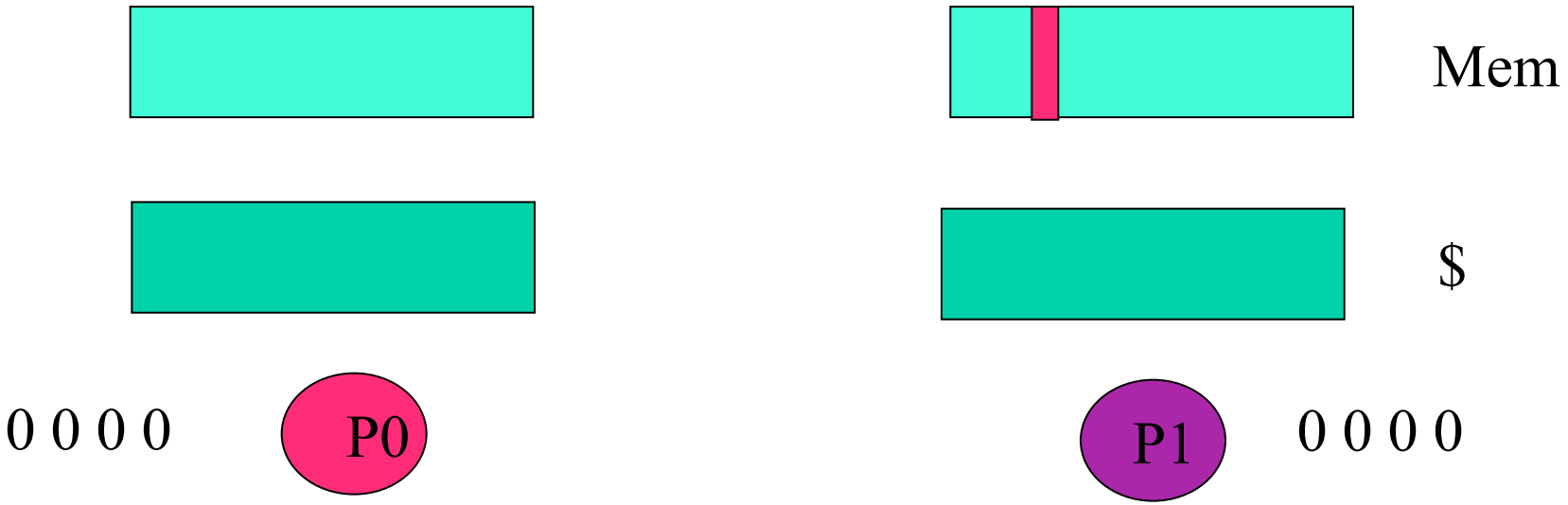
- Each processor has a 1-bit "sharer" entry in the directory
- There is also a dirty bit and a PID identifying the owner in the case of a dirt block
- Every block of memory has a **home** and an **owner**
- Initially home = owner, but this can change



Parallel Computer Architecture, Culler, Singh, & Gupta

Operation of a directory

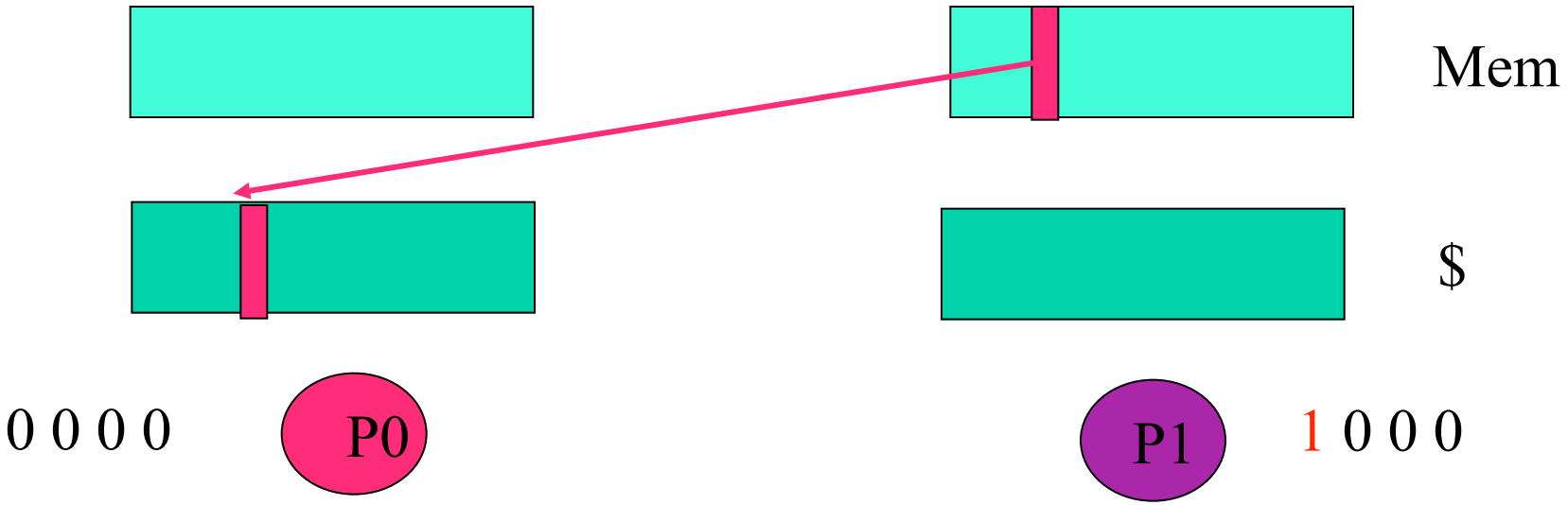
- Assume a 4 processor system (only P0 & P1 shown)
- A is a location with **home** P1
- Initial directory entry for block containing A is empty



©2010 Scott B. Baden / CSE 160 / Winter 2010

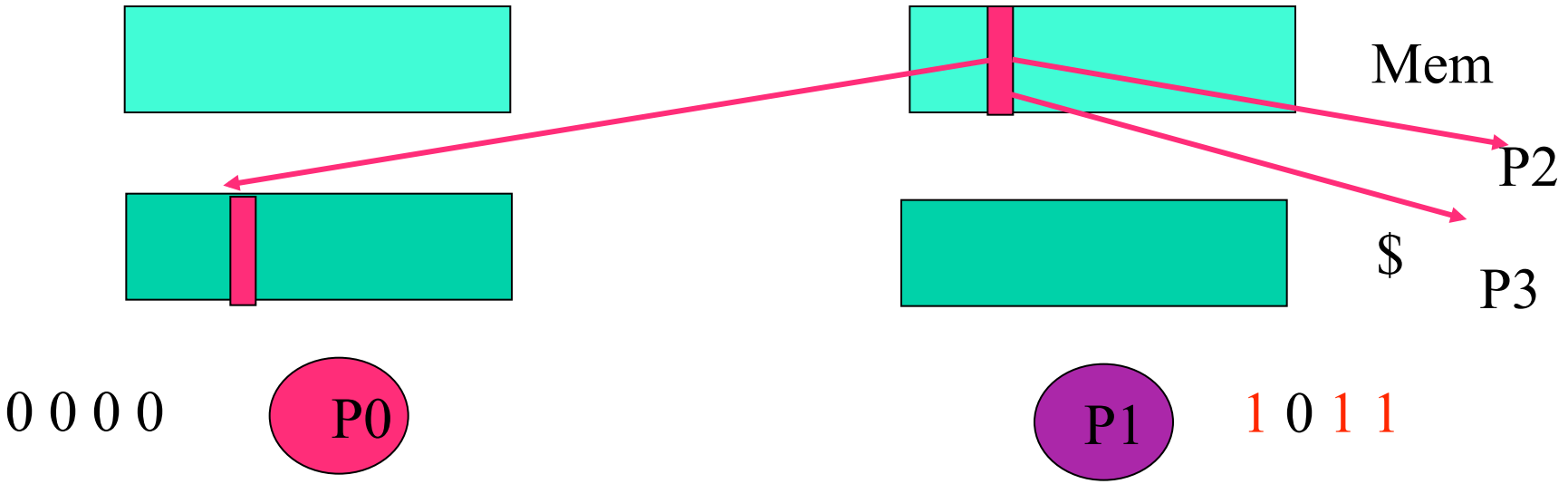
Operation of a directory

- P0 loads A
- Set directory entry for A (on P1) to indicate that P0 is a sharer



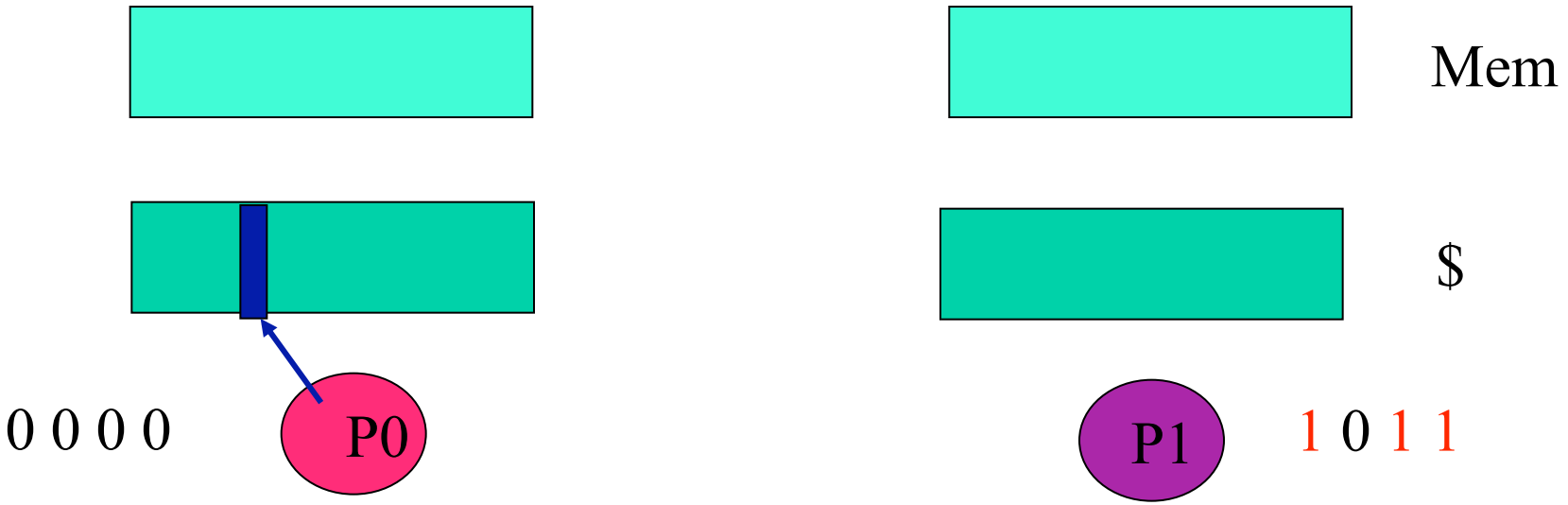
Operation of a directory

- P2, P3 load A (not shown)
- Set directory entry for A (on P1) to indicate that P0 is a sharer



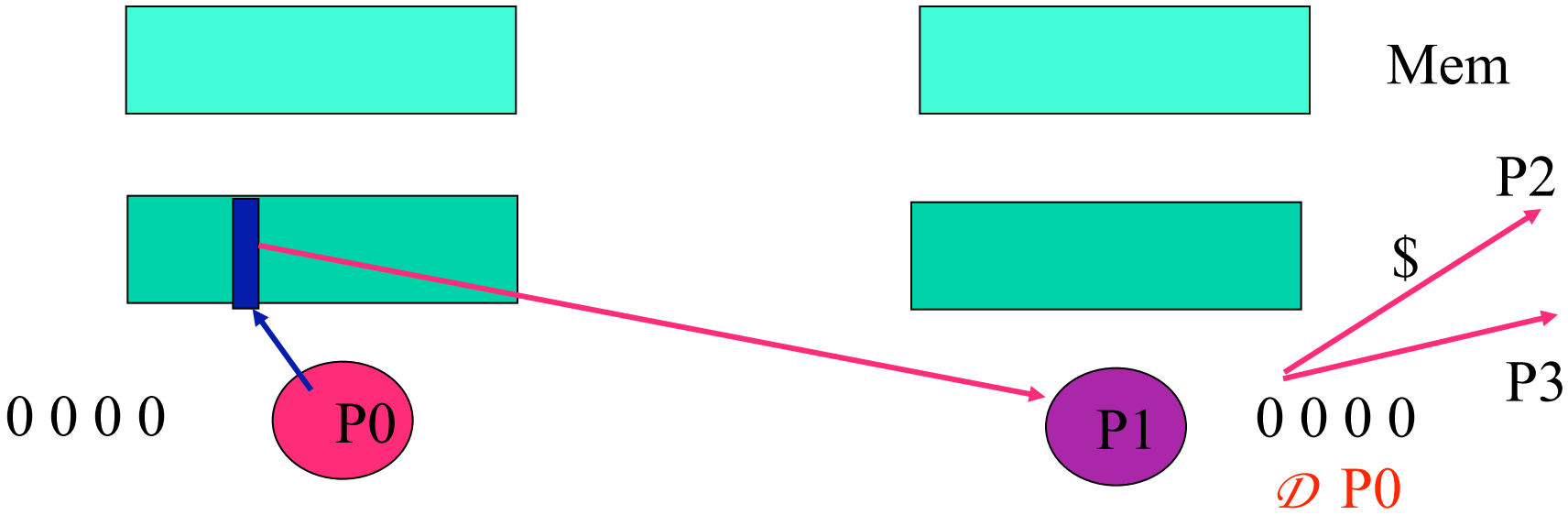
Acquiring ownership of a block

- P0 writes A
- P0 becomes the owner of A



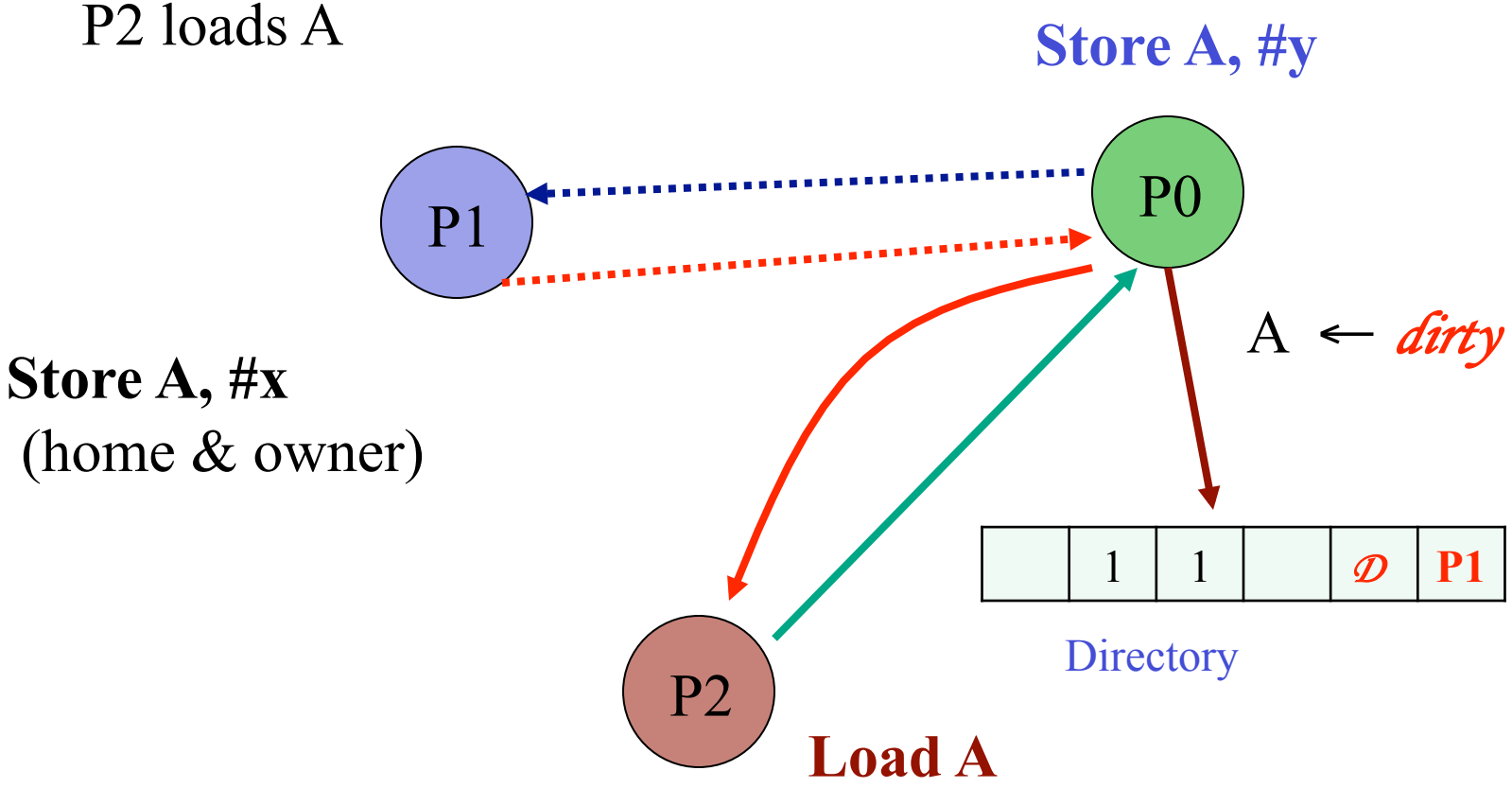
Acquiring ownership of a block

- P0 becomes the owner of A
- P1's directory entry for A is set to *Dirty*
- Outstanding sharers are invalidated
- Access to line is blocked until all invalidations are acknowledged



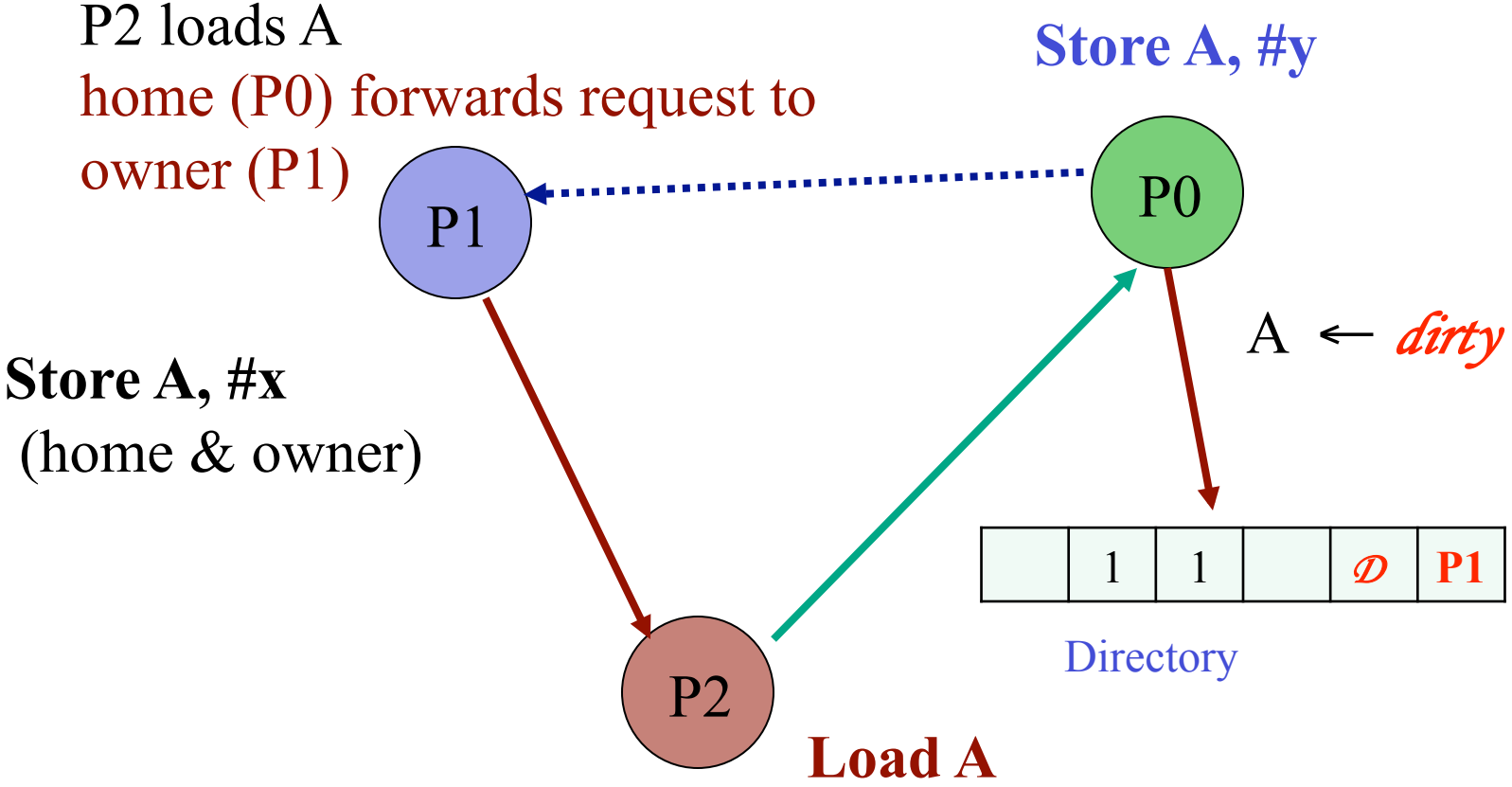
Change of ownership

P0 stores into A (home & owner)
P1 stores into A (becomes owner)
P2 loads A



Forwarding

P0 stores into A (home & owner)
P1 stores into A (becomes owner)
P2 loads A
home (P0) forwards request to owner (P1)



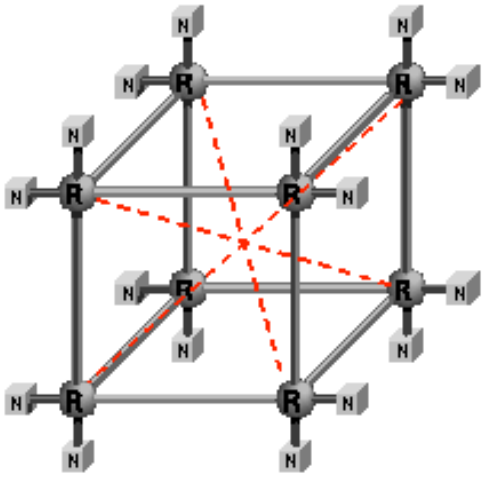
Performance issues

- Locality, locality, locality
- False sharing

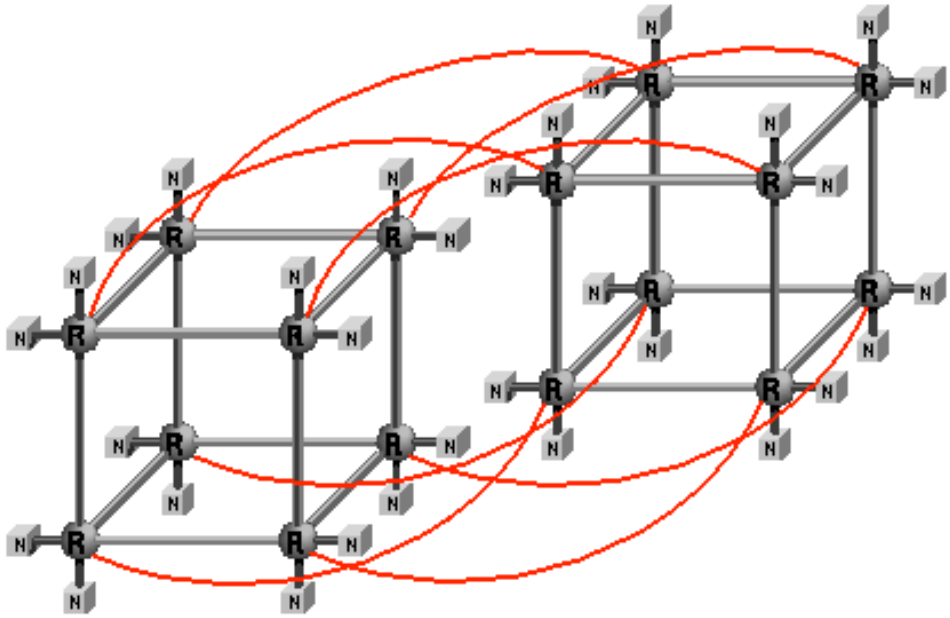
Case Study

- SGI Origin 2000

Origin 2000 Interconnect

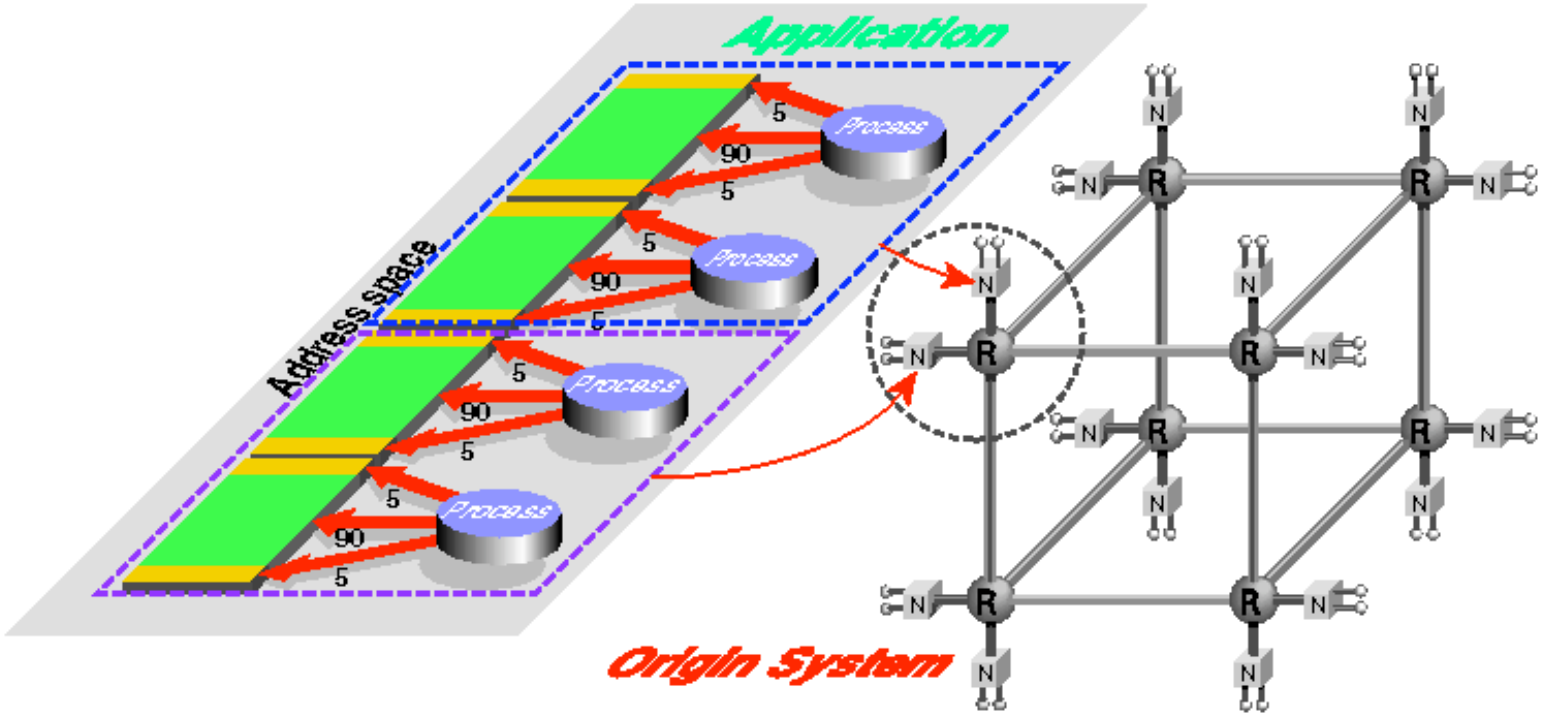


32 processor system

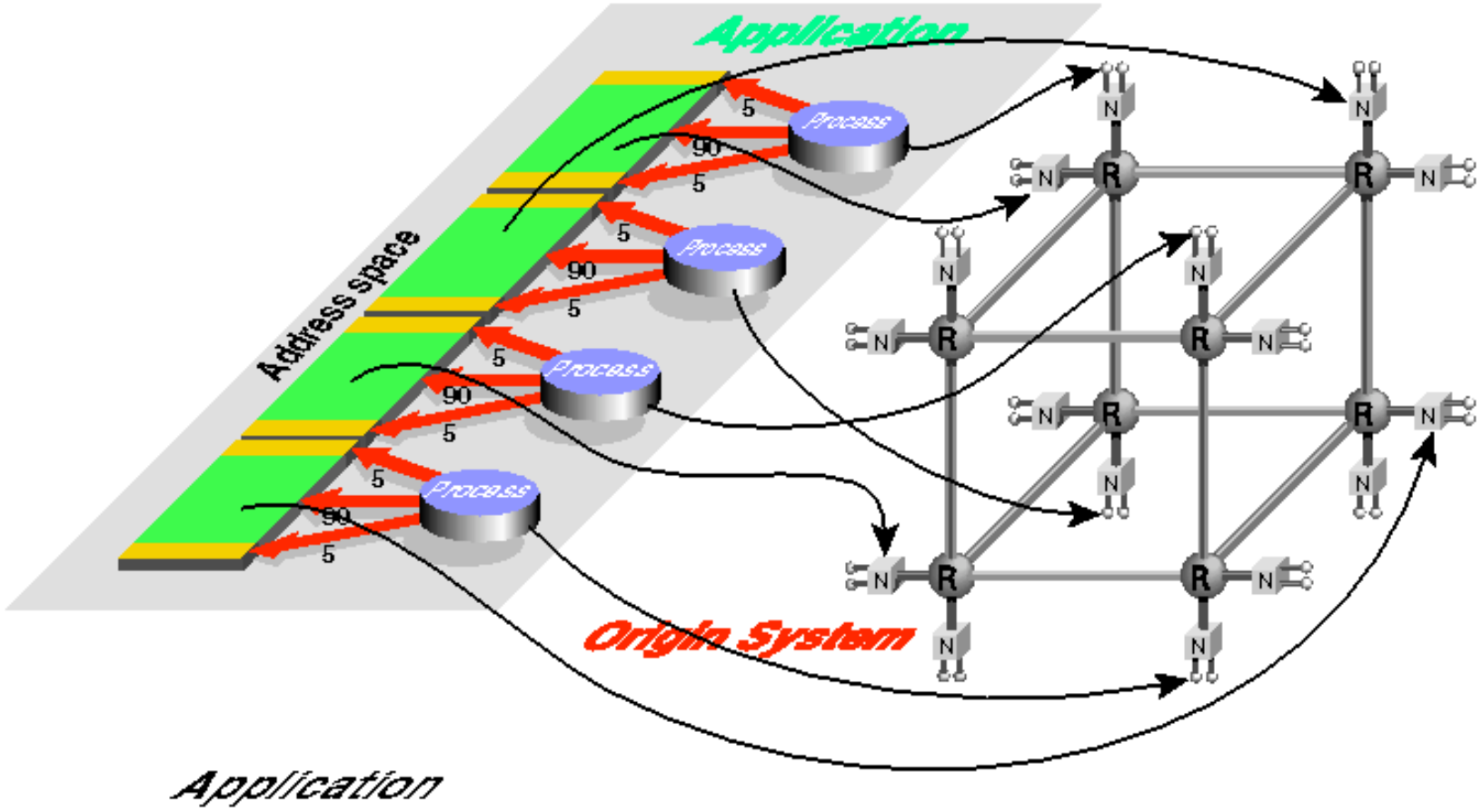


64 processor system

Locality


Application
Address space
Process
Process
Process
Process
5
90
5
Origin System
N
R

Poor Locality

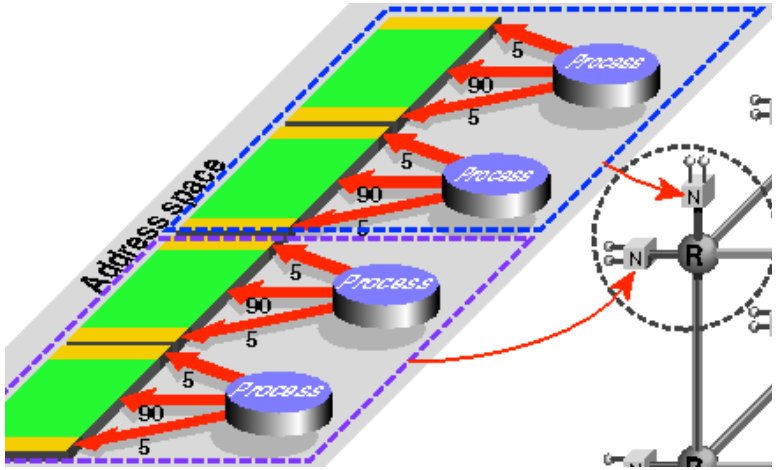


Quick primer on paging

- We group the physical and virtual address spaces into units called *pages*
- Pages are backed up on disk
- Virtual to physical mapping done by the Translation Lookaside Buffer (TLB), backs up page tables set up by the OS
- When we allocate a block of memory, we don't need to allocate physical storage to pages; we do it on demand

Remote access latency

- When we allocate a block of memory, which processor(s) is (are) the owner(s)?
- Page allocation policies
  - First touch
  - Round robin
- Page placement and Page migration



- Copying v. redistribution
- Layout

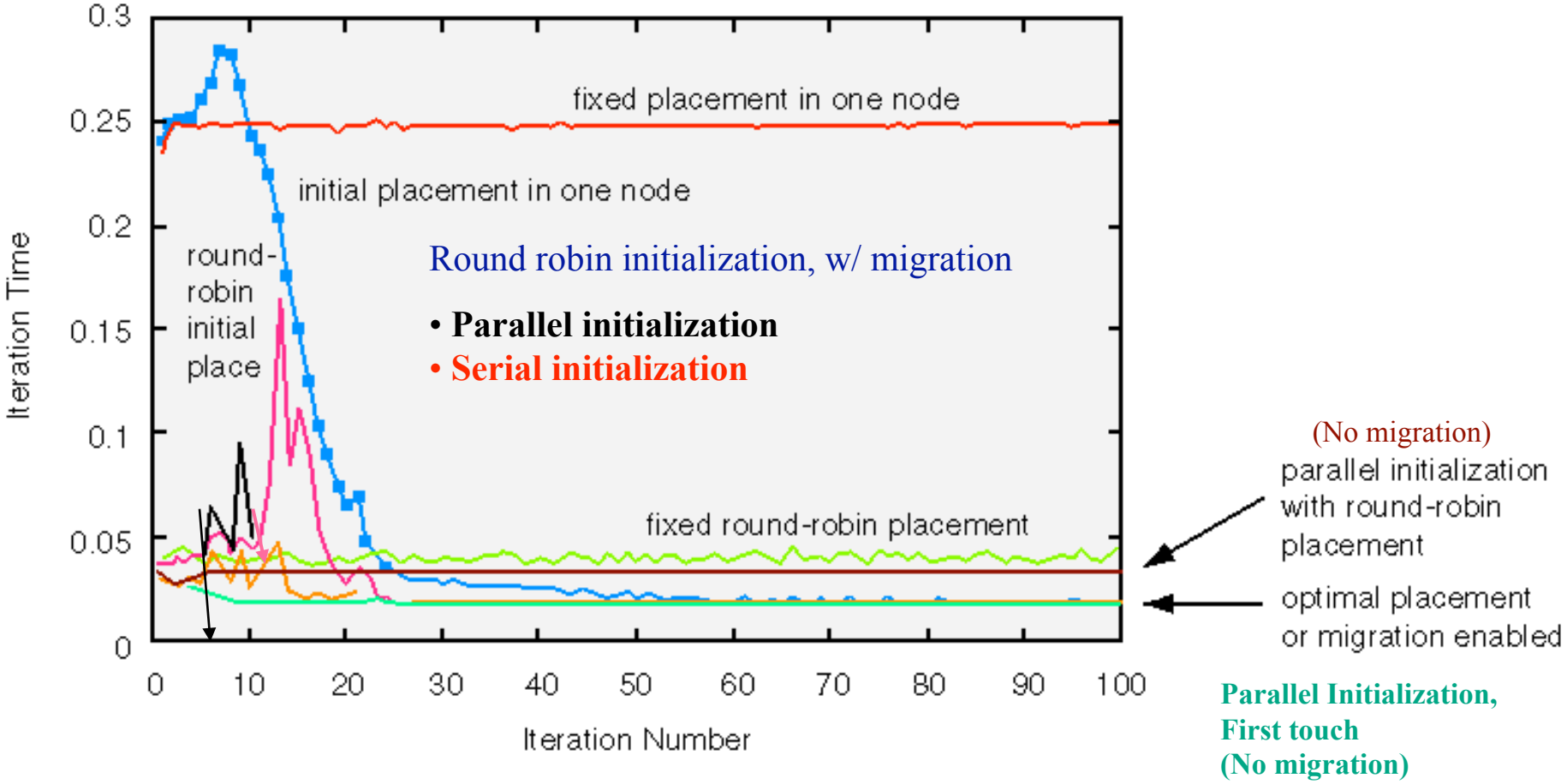
Example

- Consider the following loop

```
for r = 0 to nReps
    for i = 0 to n-1
        a[i] = b[i] + q*c[i]
```

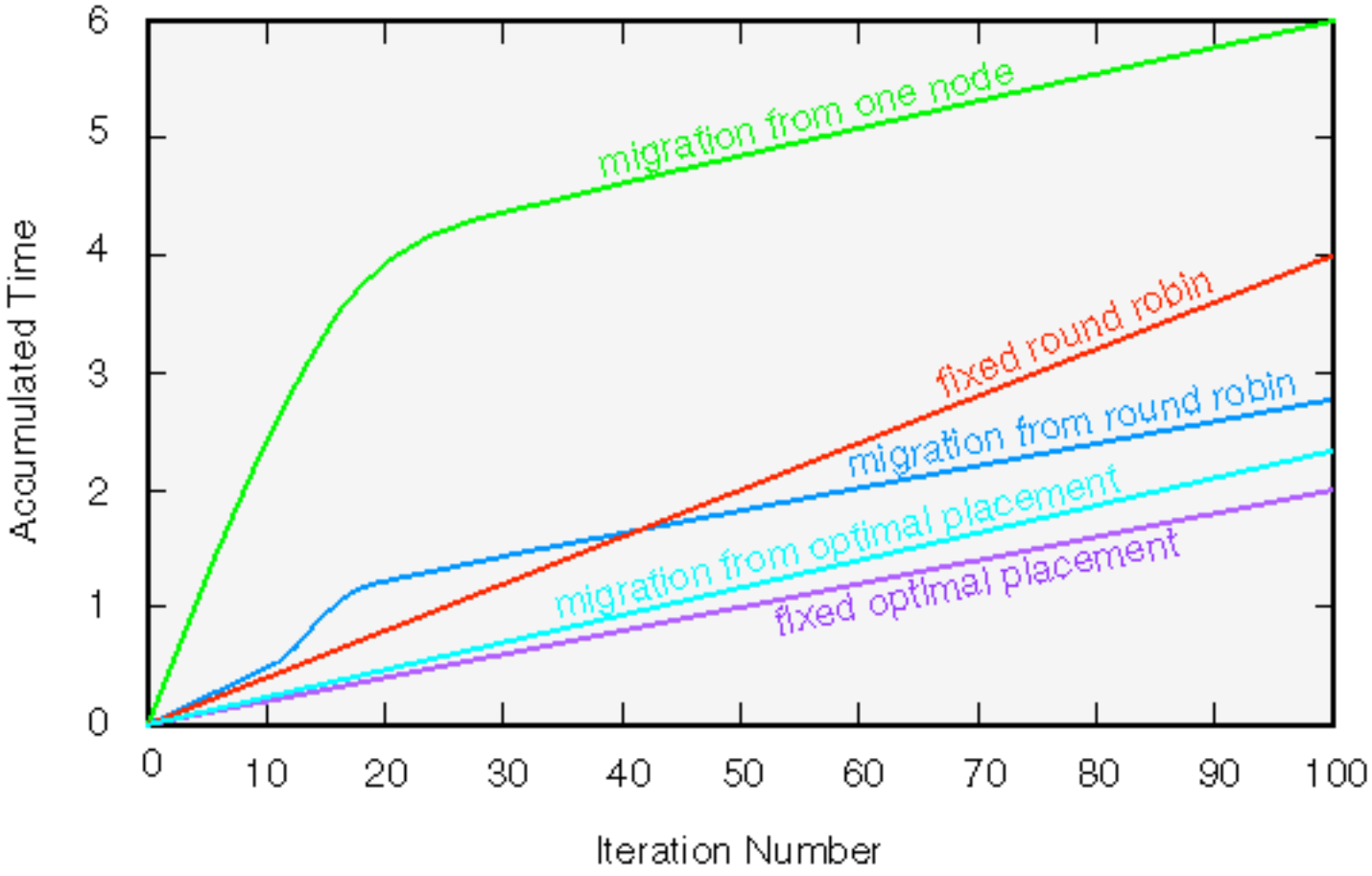
Page Migration

a[i] = b[i] + q*c[i]



techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/0650/bks/SGI_Developer/books/OrOn2_PfTune/sgi_html/ch08.html#id5224855

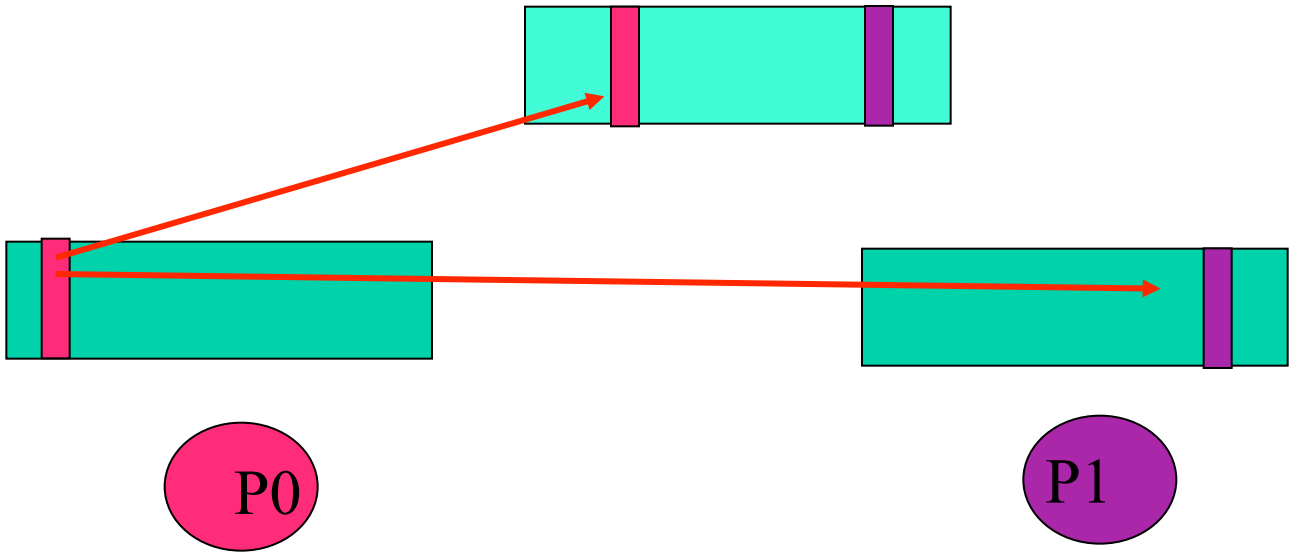
Cumulative effect of Page Migration



Eliminating false sharing

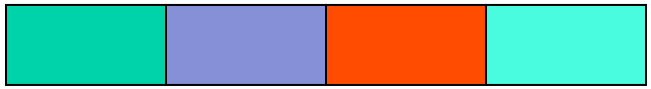
False sharing

Successive writes by P0 and P1 cause the processors to uselessly invalidate one another's cache



An example of false sharing

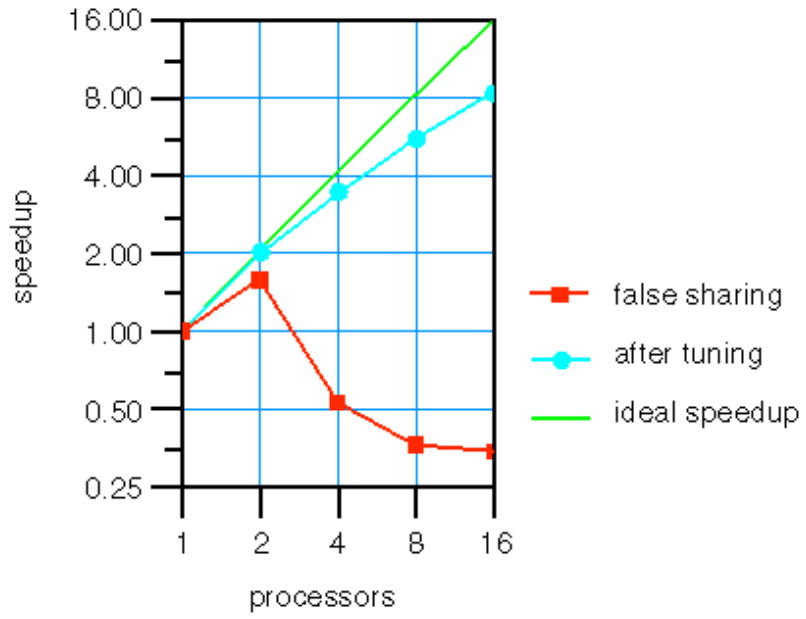
```
float  a[m,n], s[m]
// Outer loop is in parallel
// Consider m=4, 128 byte cache line size
// Thread i updates  element s[i]
#pragma omp parallel for private(i,j), shared(s,a)
     for i = 0, m-1
        s[i] = 0.0
        for  j = 0, n-1
           s[i] += a[i,j]
        end for
     end for
```



Avoiding false sharing

```
float  a[m,n], s[m,32]
#pragma omp parallel for private(i,j), shared(s,a)
for i = 0, m-1
     s[i,1] = 0.0
     for  j = 0, n-1
        s[i,1] += a[i,j]
     end for
 end for
```

0	1	2	3	4	5	6	7	•••	31
0	1	2	3	4	5	6	7	•••	31
0	1	2	3	4	5	6	7	•••	31
0	1	2	3	4	5	6	7	•••	31

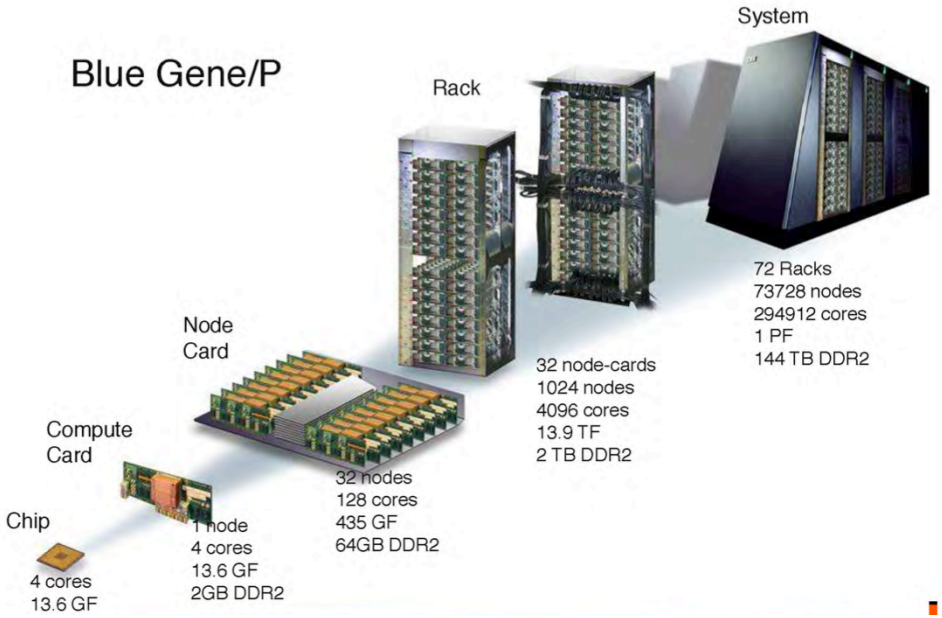


Blue Gene

- IBM-US Dept of Energy collaboration
- First generation: Blue Gene/L
  - 64K dual processor nodes: 180 (360) TeraFlop peak
  - 1 TeraFlop = 1,000 GigaFlops
- Low power
  - Relatively slow processors; power PC 440
  - Small memory (256 MB)
- High performance interconnect

Current Generation: Blue Gene/P

- Largest Installation at Argonne National Lab: 294,912 cores
  - 4-way SMP nodes
  - PowerPC 450 (850 MHz)
  - 2 GB memory per node
- Peak performance: 13.6 Gflops/node = 557 TeraFlops total = 0.56 Petaflops



http://www.redbooks.ibm.com/redbooks/SG247287

Blue Gene/P Interconnect

- 3D toroidal mesh (end around)
  - 5.1 GB/sec bidirectional bandwidth / node (6 birectional links @ 425MB/sec)
  - 5µs worst case latency, 0.5µs best case (nearest neighbor
  - MPI: 3 µs to 10 µs
- Collective network
  - Broadcast
  - Reduction for integers and doubles
  - Ne way tree latency 1.3 µs (5 µs in MPI)
- Low latency barrier and interrupt
  - One way: 0.65µs
  - 1.6 µs in MPI

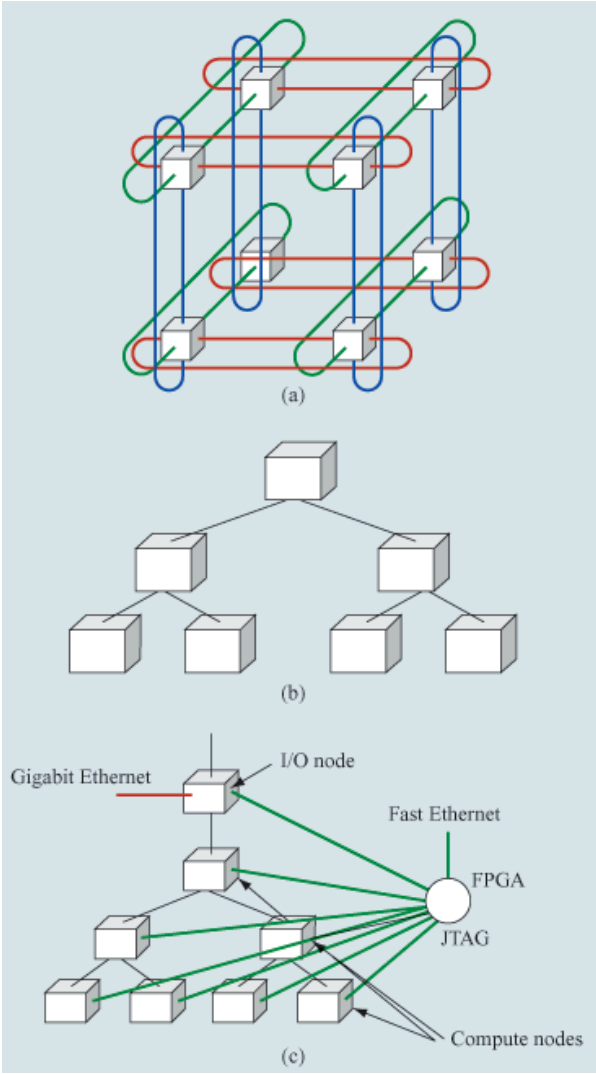
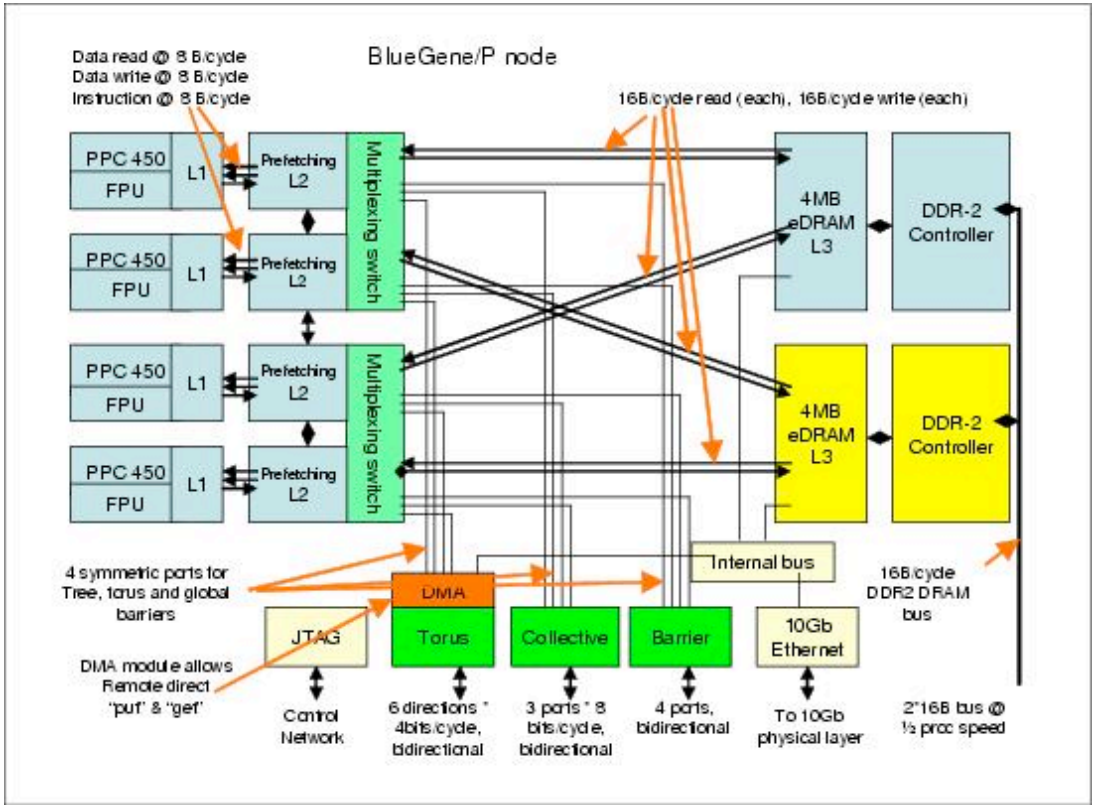


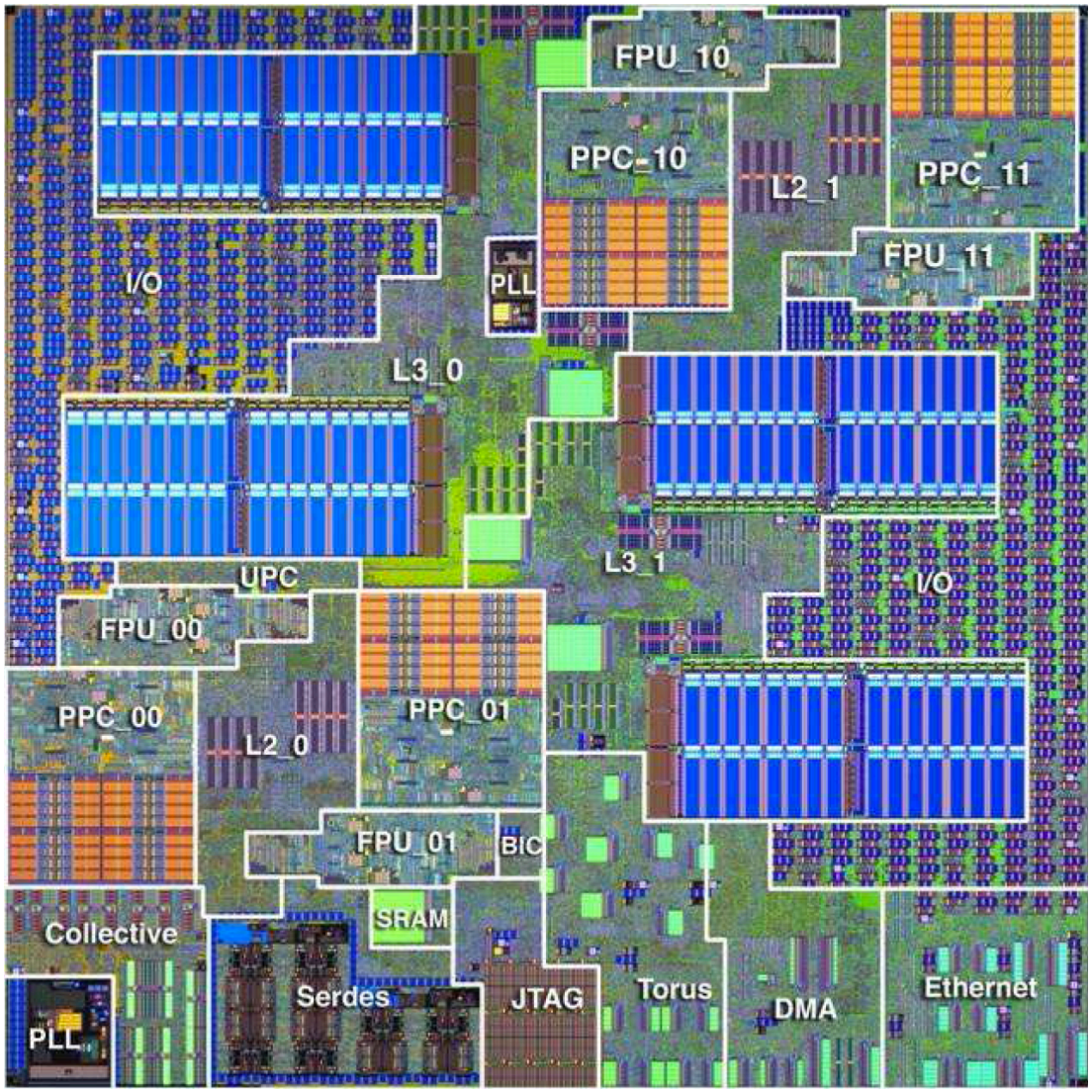
Image courtesy of IBM
http://www.research.ibm.com/journal/rd/492/gara4.gif

Compute nodes

- Six connections to torus network @ 425 MB/sec/link (duplex)
- Three connections to global collective network @ 850 MB/sec/link
- Network routers are embedded within the processor



Die photograph



Argonne National Lab

Programming modes

- Virtual node: each node runs 4 MPI processes, 1/core
  - Memory and torus network shared by all processes
  - Shared memory is available between processes.
- Dual node: each node runs 2 MPI processes, 1 or 2 threads/process
- Symmetrical Multiprocessing:
  each node runs 1 MPI process, up to 4 threads

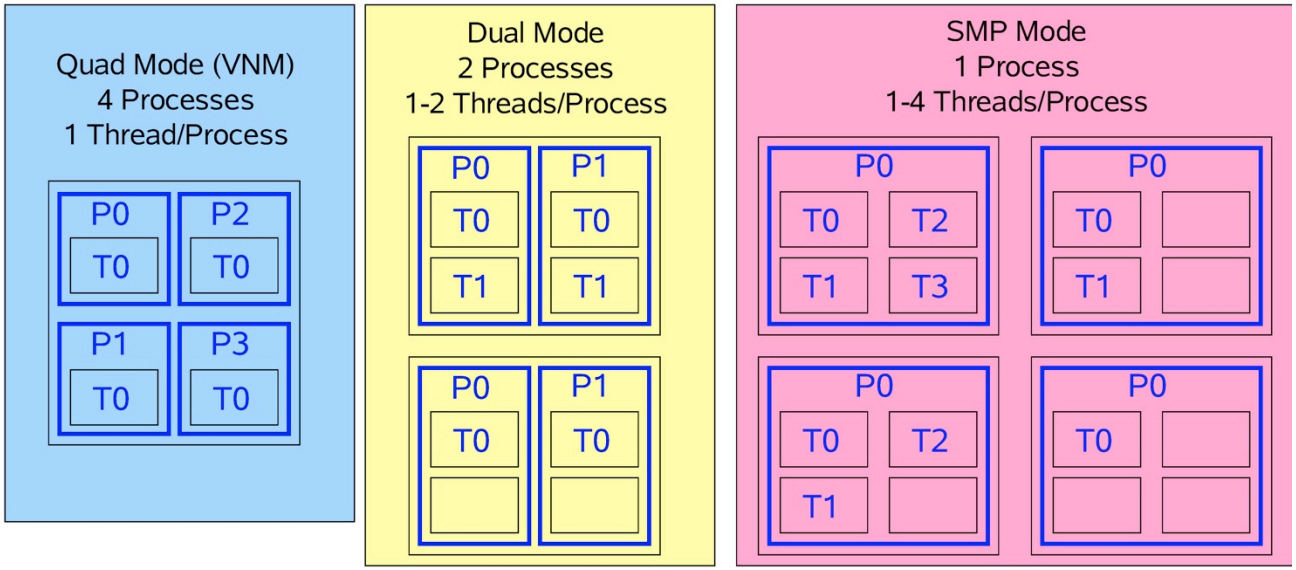


Image courtesy of IBM

Next Generation: Blue Gene Q

- Sequoia: Lawrence Livermore National Lab
- 1.6M cores in 93,304 compute nodes
- 20 Petaflops = 2×10^{16} flops
  - 20,000 TerFlops = 20M GFlops
- 96 racks, 3,000 square feet
- 6M Watts (×7 more power efficient than BGP)

What is the worlds fastest supercomputer?

- Go to top500.org
- #1: Tianhe-1A (China)
  - 2.57 Petaflops/sec = 1M GFlops
  - Nvidia processors
- #2: Jaguar (US)
  - 1.75 Petaflops/sec
  - Cray XT5-HE, 6-core Opteron
- #3: Nebulae (China)
  - 1.27 PF
- #4: Tsubame (Japan)
  - 1.19 PF

Fin