Applications of Mutual Exclusion in Concurrency

Dr. Sean Peisert
Guest Lecture for UCD ECS 150 (Operating Systems)
April 11, 2008

peisert@cs.ucdavis.edu
http://www.sdsc.edu/~peisert

Race Conditions = Nondeterminism: Threats to Reliability & Security

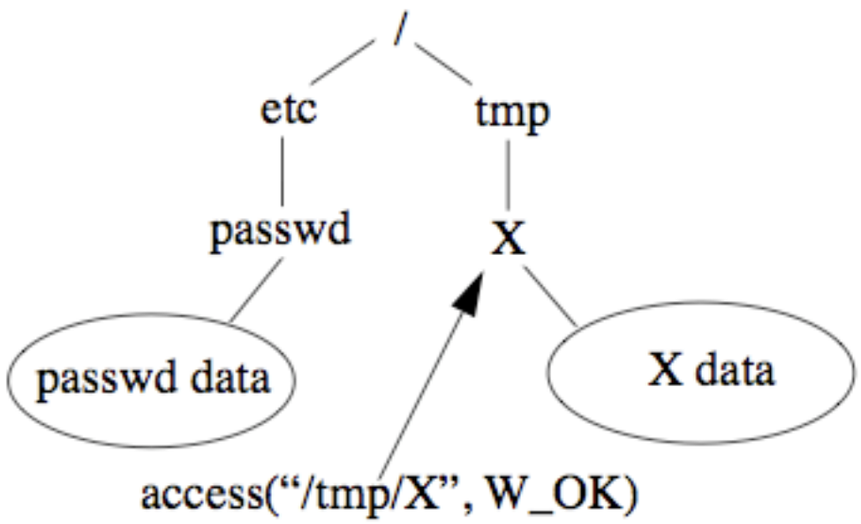
Example

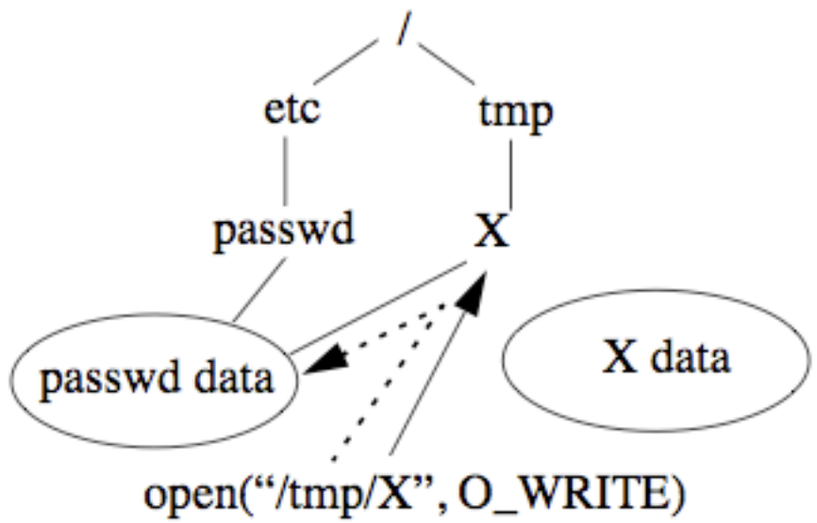
```
if ((fd = open(filename, O_WRONLY)) == NULL){
   perror(filename);
   return(0);
}
/* write to the file */
```

Improved Example?

```
if (access(filename, W_OK) == 0){
   if ((fd = open(filename, O_WRONLY)) == NULL){
      perror(filename);
      return(0);
   }
   /* now write to the file */

}
```


/
etc
tmp
passwd
X
passwd data
X data
access("/tmp/X", W_OK)


/
etc
tmp
passwd
X
passwd data
X data
open("/tmp/X", O_WRITE)

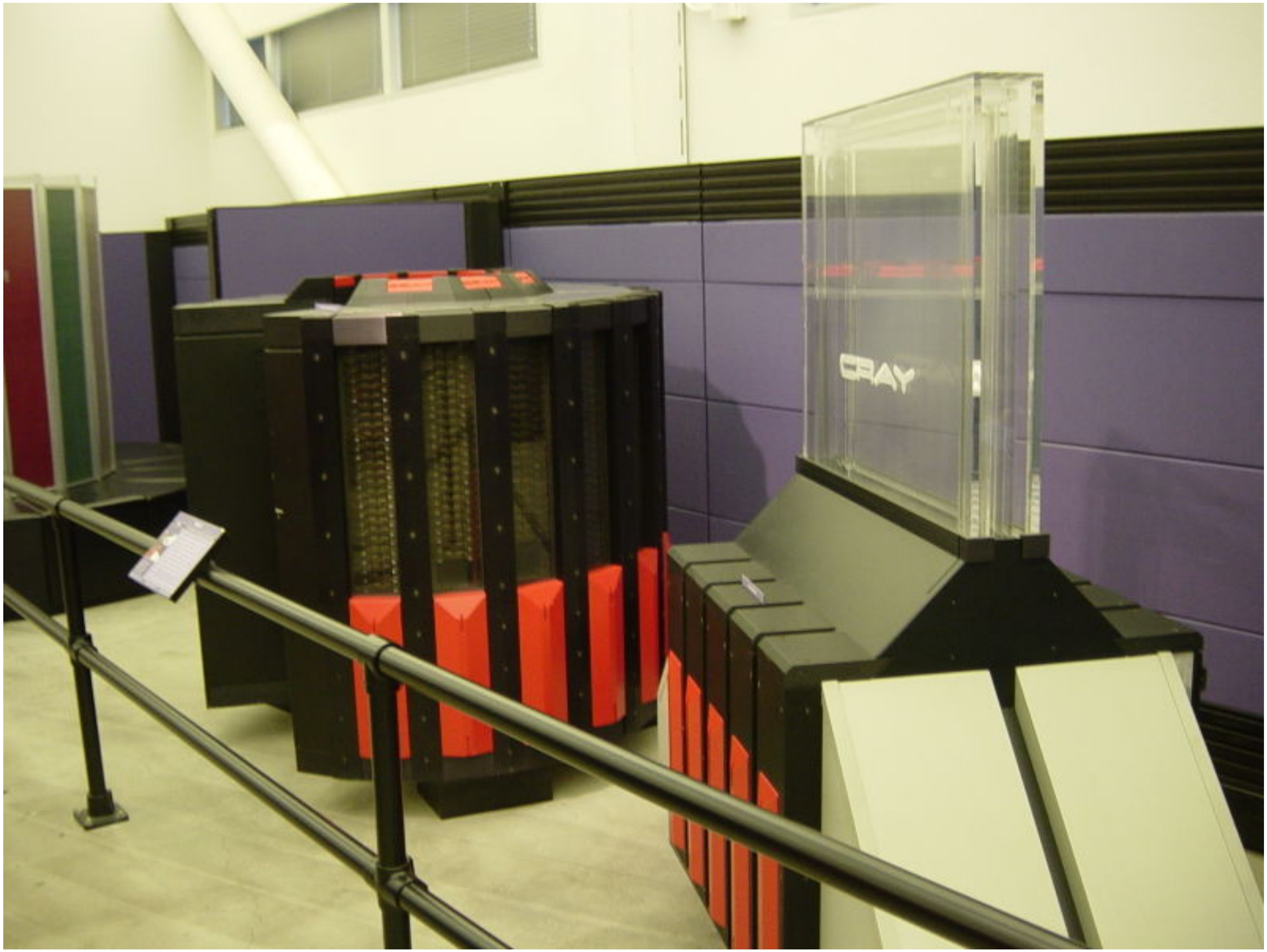
The Problem

Possible Corrections

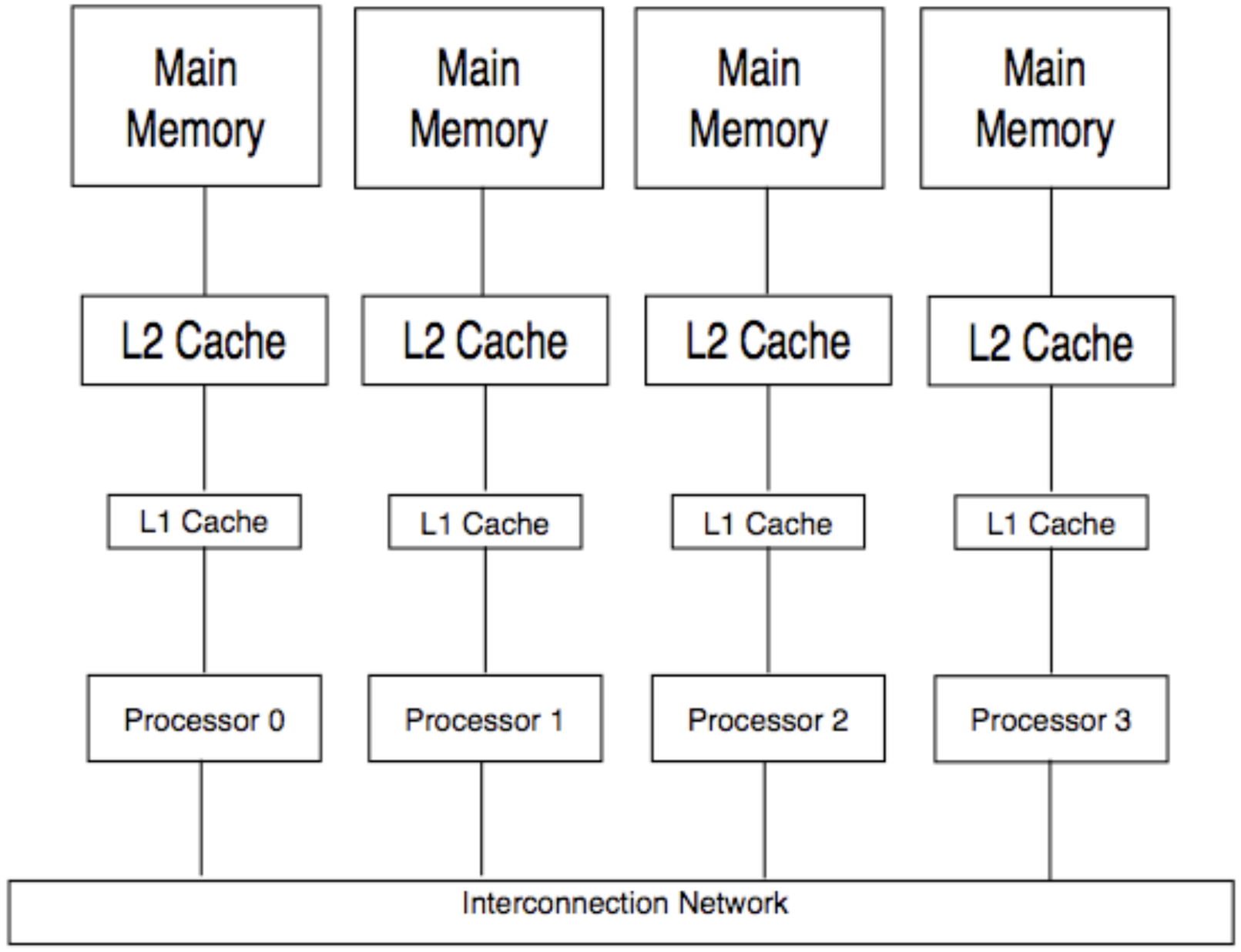
1. Put file in a directory that the *real* UID can't access
2. Drop *effective* UID permissions before opening
3. Use *mandatory* "locks"

```
//
// Correction #2 Example: Principle of Least Privilege
//
int savedEUID = geteuid();
seteuid(getuid());
if ((fd = open(filename, O_WRONLY)) == NULL){
      perror(filename);
      return(0);
   }
}
/* now write to the file */
seteuid(savedEUID);
```

Supercomputers

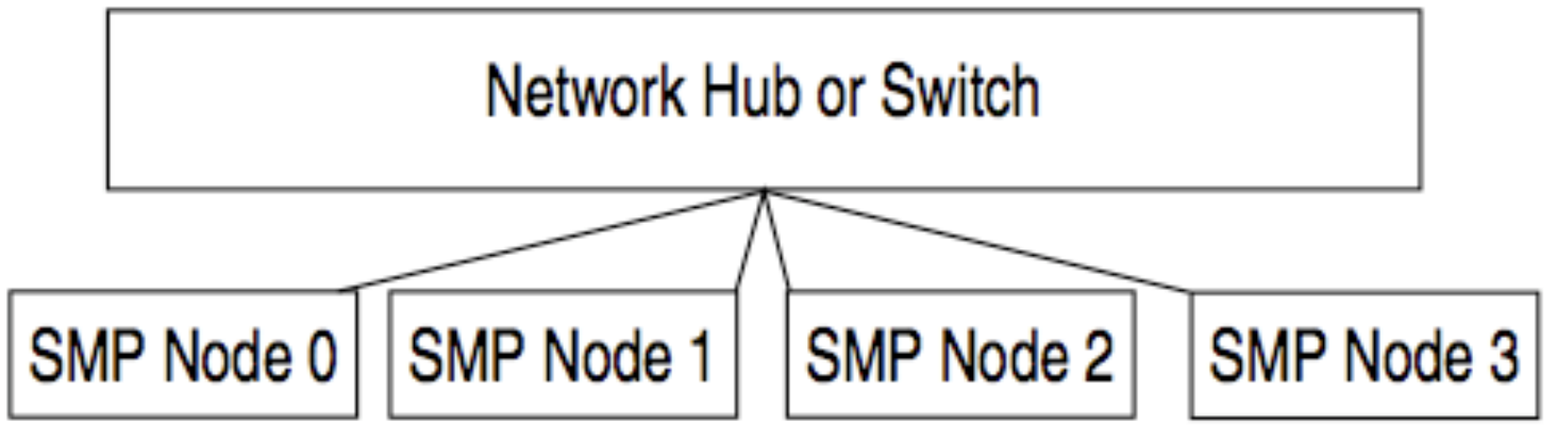


Cray 2 (Supercomputer)

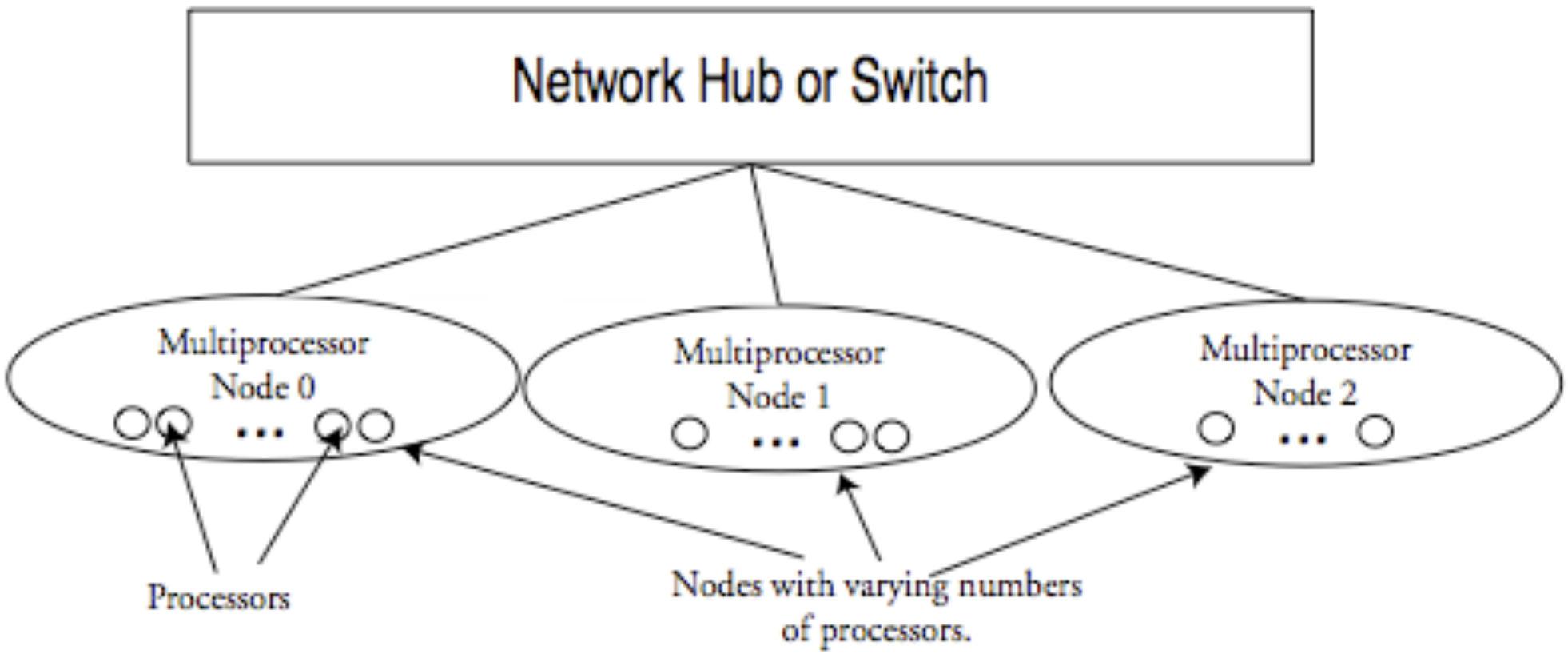


Distributed Memory

Clusters



Symmetric Cluster of Multiprocessors



Heterogeneous Cluster of Multiprocessors

Parallel Programming Modes

- Message Passing
    - Message Passing Interface (MPI)
    - Parallel Virtual Machine (PVM)
- Shared Memory
    - POSIX Threads (Pthreads)
    - Open Message-Passing (OpenMP)

MPI Library Calls

- MPI_Send() & MPI_Recv()
- MPIIsend() & MPI_Irecv()
- MPI_Wait() & MPI Test()
- MPI_Reduce()
- MPI_Gather() & MPI_Scatter()
- MPI_Alltoall()

MPI Example

```
#include <mpi.h>
int main(int argc, char *argv[]) {
   int numprocs, myid, i;
   MPI_Status stat;
   MPI_Init(&argc,&argv);
   MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
   MPI_Comm_rank(MPI_COMM_WORLD,&myid);

   if(myid == 0) {
       printf("%d: We have %d processors\n", myid, numprocs);
       for(i=1;i<numprocs;i++) {
          sprintf(buff, "Hello %d! ", i);
          MPI_Send(buff, BUFSIZE, MPI_CHAR, i, TAG, MPI_COMM_WORLD);
       }
       for(i=1;i<numprocs;i++) {
          MPI_Recv(buff, BUFSIZE, MPI_CHAR, i, TAG, MPI_COMM_WORLD, &stat);
          printf("%d: %s\n", myid, buff);
       }
  } else {
       MPI_Recv(buff, BUFSIZE, MPI_CHAR, 0, TAG, MPI_COMM_WORLD, &stat);
       sprintf(buffer, "Processor %d ", myid);
       strcat(buffer, "reporting for duty\n");
       MPI_Send(buffer, BUFSIZE, MPI_CHAR, 0, TAG, MPI_COMM_WORLD);
  }
  MPI_Finalize();
  return 0;
}
```

Pthreads Library Calls

- pthread_create()
- pthread_exit()
- pthread_join()
- pthread_mutex_lock()
- pthread_mutex_unlock()
- pthread_mutex_trylock()

Pthreads Example

```
#include <pthread.h>
static void wait(void) {
    time_t start_time = time(NULL);
    while (time(NULL) == start_time)
    { /* do nothing except chew CPU slices for up to one second */ }
}
static void *thread_func(void *vptr_args) {
    int i;
    for (i = 0; i < 20; i++) {
        fputs("  b\n", stderr);
        wait();
    }
    return NULL;
}
int main(void) {
    int i;
    pthread_t thread;

    if (pthread_create(&thread, NULL, thread_func, NULL) != 0)
        return EXIT_FAILURE;
    for (i = 0; i < 20; i++) {
        fputs("a\n", stdout);
        wait();
    }
    if (pthread_join(thread, NULL) != 0)
        return EXIT_FAILURE;
    return EXIT_SUCCESS;
}
```

OpenMP Pragmas

- Thread creation: omp parallel
- Work sharing: omp for, omp do, single, master
- Scoping: shared, private, default
- Synchronization: critical section, barrier, nowait

OpenMP Example

```
#include <omp.h>
int main (int argc, char *argv[]) {
  int th_id, nthreads;
  #pragma omp parallel private(th_id)
  {
       th_id = omp_get_thread_num();
       printf("Hello World from thread %d\n", th_id);
       #pragma omp barrier
       if ( th_id == 0 )
       {
            nthreads = omp_get_num_threads();
            printf("There are %d threads\n",nthreads);
       }
  }
  return 0;
}
```

Mixed-Mode Programming

Questions?

- Email: peisert@cs.ucdavis.edu
- Web: http://www.sdsc.edu/~peisert