# Hybrid Parallel Programming
## Part 1

*Bhupender Thakur*
*HPC @ LSU*

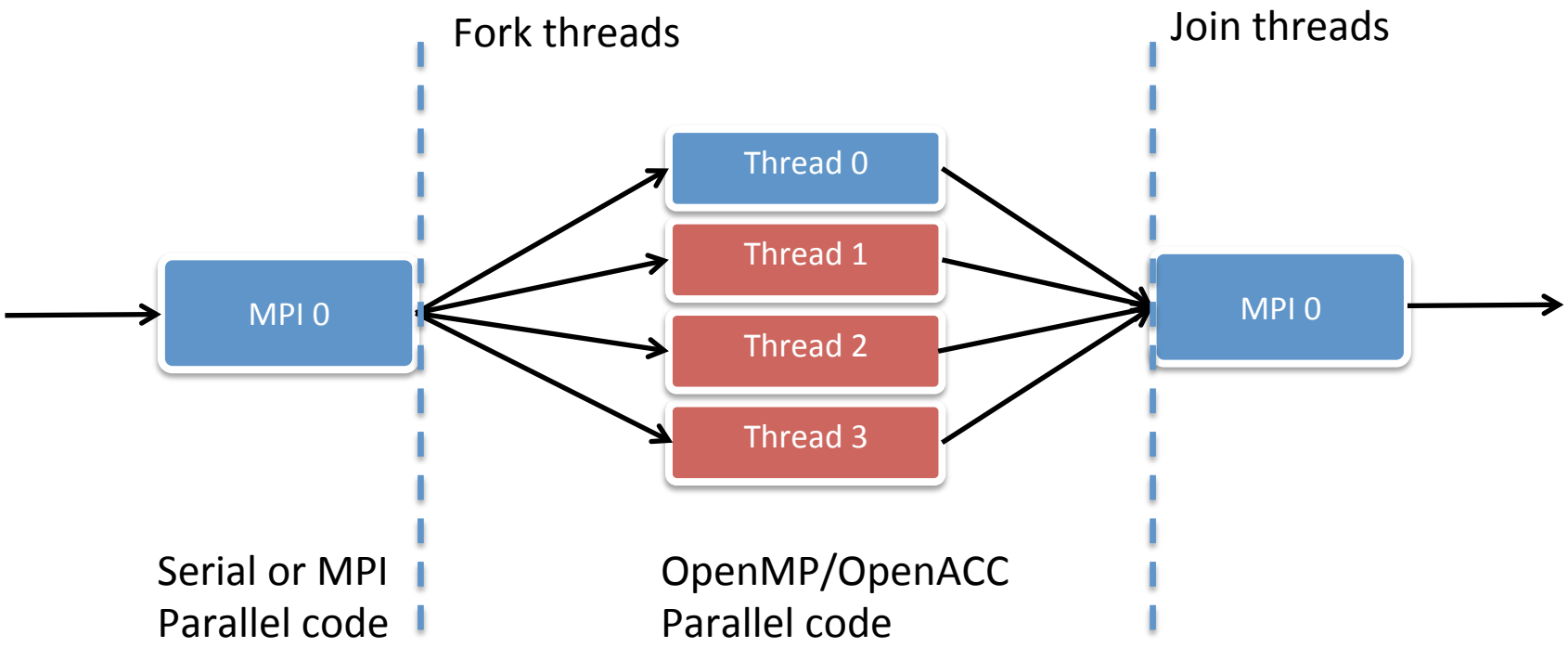
# Overview

- Basic Hybrid model
- NUMA Challenges
  - Understanding Hardware
- Communication support for threads in MPI-2
- Hello hybrid

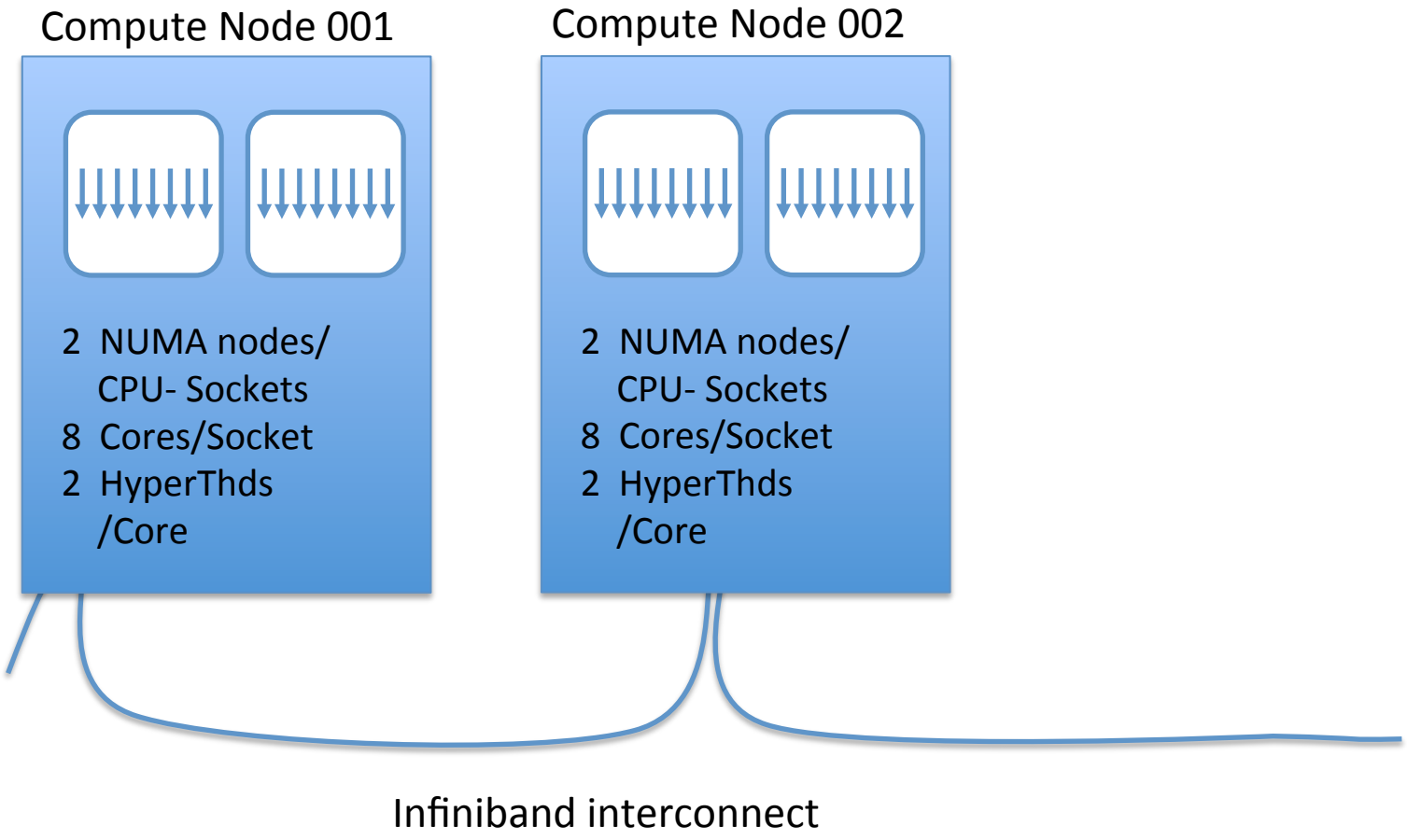
# Why do we need a hybrid approach

- Threading reduces memory footprint of MPI processes. MPI processes on a node replicate memory.
- Threading reduces MPI communication between processes
- Reduces programming effort for certain OpenMP parallel sections of the code

# Hybrid Programming: Basic Model



# Understanding NUMA architecture



# NUMA: SuperMike-II

```
=====  Processor  =====

Processor name    : Intel(R) Xeon(R)  E5-2670 0
Packages(sockets) :    2
Cores :                16
Processors(CPUs)  :    16
Cores per package :    8
Threads per core  :    1
```

```
$ numactl --hardware |grep cpus
----------------
----------------
node 0 cpus: 0 1 2 3 4 5 6 7
node 1 cpus: 8 9 10 11 12 13 14 15
```

| Proc | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 2 | 0 |
| 3 | 0 | 3 | 0 |
| 4 | 0 | 4 | 0 |
| 5 | 0 | 5 | 0 |
| 6 | 0 | 6 | 0 |
| 7 | 0 | 7 | 0 |
| 8 | 0 | 0 | 1 |
| 9 | 0 | 1 | 1 |
| 10 | 0 | 2 | 1 |
| 11 | 0 | 3 | 1 |
| 12 | 0 | 4 | 1 |
| 13 | 0 | 5 | 1 |
| 14 | 0 | 6 | 1 |
| 15 | 0 | 7 | 1 |

# NUMA: SuperMike-II

===== Processor identification =====

| Proc | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 2 | 0 |
| 3 | 0 | 3 | 0 |
| 4 | 0 | 4 | 0 |
| 5 | 0 | 5 | 0 |
| 6 | 0 | 6 | 0 |
| 7 | 0 | 7 | 0 |
| 8 | 0 | 0 | 1 |
| 9 | 0 | 1 | 1 |
| 10 | 0 | 2 | 1 |
| 11 | 0 | 3 | 1 |
| 12 | 0 | 4 | 1 |
| 13 | 0 | 5 | 1 |
| 14 | 0 | 6 | 1 |
| 15 | 0 | 7 | 1 |

===== Placement on packages =====

| Package Id. | Core Id. | Processors |
|---|---|---|
| 0 | 0,1,2,3,4,5,6,7 | 0,1,2,3,4,5,6,7 |
| 1 | 0,1,2,3,4,5,6,7 | 8,9,10,11,12,13,14,15 |

===== Cache sharing =====

| Cache | Size | Processors |
|---|---|---|
| L1 | 32 KB | no sharing |
| L2 | 256 KB | no sharing |
| L3 | 20 MB | (0,1,2,3,4,5,6,7) (8,9,10,11,12,13,14,15) |

===== Processor composition =====

Processor name : Intel(R) Xeon(R) E5-2670 0
Packages(sockets) : 2
Cores : 16
Processors(CPUs) : 16
Cores per package : 8
Threads per core : 1

# NUMA : Quiz 1

**Quiz: Can you identify the node structure here ?**

```
$ numactl --hardware|grep cpus
----------------
----------------
node 0 cpus: 0 1 2 3 4 5 12 13 14 15 16 17
node 1 cpus: 6 7 8 9 10 11 18 19 20 21 22 23
```

| Proc | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 2 | 0 |
| 3 | 0 | 8 | 0 |
| 4 | 0 | 9 | 0 |
| 5 | 0 | 10 | 0 |
| 6 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 |
| 8 | 0 | 2 | 1 |
| 9 | 0 | 8 | 1 |
| 10 | 0 | 9 | 1 |
| 11 | 0 | 10 | 1 |
| 12 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 |
| 14 | 1 | 2 | 0 |
| 15 | 1 | 8 | 0 |
| 16 | 1 | 9 | 0 |
| 17 | 1 | 10 | 0 |
| 18 | 1 | 0 | 1 |
| 19 | 1 | 1 | 1 |
| 20 | 1 | 2 | 1 |
| 21 | 1 | 8 | 1 |
| 22 | 1 | 9 | 1 |
| 23 | 1 | 10 | 1 |

# NUMA : Quiz 1

```
=====  Processor composition  =====

Processor name    : Intel(R) Xeon(R)  X5670
Packages(sockets) :    2
Cores             :    12
Processors(CPUs)  :    24
Cores per package :    6
Threads per core  :    2
```

```
$ numactl --hardware|grep cpus
----------------
----------------
node 0 cpus: 0 1 2 3 4 5 12 13 14 15 16 17
node 1 cpus: 6 7 8 9 10 11 18 19 20 21 22 23
```

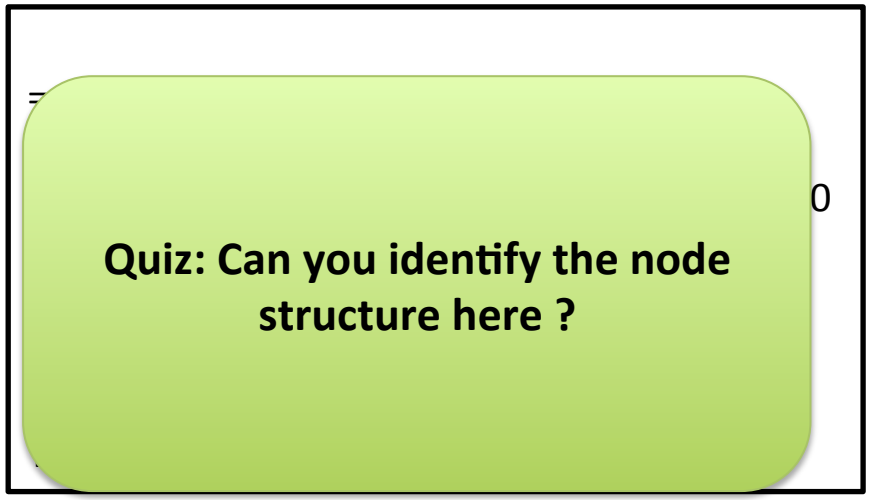
| Proc | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 2 | 0 |
| 3 | 0 | 8 | 0 |
| 4 | 0 | 9 | 0 |
| 5 | 0 | 10 | 0 |
| 6 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 |
| 8 | 0 | 2 | 1 |
| 9 | 0 | 8 | 1 |
| 10 | 0 | 9 | 1 |
| 11 | 0 | 10 | 1 |
| 12 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 |
| 14 | 1 | 2 | 0 |
| 15 | 1 | 8 | 0 |
| 16 | 1 | 9 | 0 |
| 17 | 1 | 10 | 0 |
| 18 | 1 | 0 | 1 |
| 19 | 1 | 1 | 1 |
| 20 | 1 | 2 | 1 |
| 21 | 1 | 8 | 1 |
| 22 | 1 | 9 | 1 |
| 23 | 1 | 10 | 1 |

# NUMA : Quiz 1

```
=====  Placement on packages  =====

      Package Id.          Processors
      0                    (0,12)(1,13)(2,14)(3,15)(4,16)(5,17)
      1                    (6,18)(7,19)(8,20)(9,21)(10,22)(11,23)
```

```
=====  Cache sharing  =====

Cache      Size        Processors
L1     32  KB          (0,12)(1,13)(2,14)(3,15)(4,16)(5,17)(6,18)(7,19)(8,20)(9,21)(10,22)(11,23)
L2     256 KB          (0,12)(1,13)(2,14)(3,15)(4,16)(5,17)(6,18)(7,19)(8,20)(9,21)(10,22)(11,23)
L3     12  MB          (0,1,2,3,4,5,12,13,14,15,16,17)(6,7,8,9,10,11,18,19,20,21,22,23)
```

# NUMA : Quiz 2




```
$ numactl --hardware |grep cpus

node 0 cpus: 0 2 4 6 8 10 12 14 16 18 20 22
node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23
```

| Processor | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 0 | 2 | 0 |
| 5 | 0 | 2 | 1 |
| 6 | 0 | 3 | 0 |
| 7 | 0 | 3 | 1 |
| 8 | 0 | 4 | 0 |
| 9 | 0 | 4 | 1 |
| 10 | 0 | 5 | 0 |
| 11 | 0 | 5 | 1 |
| 12 | 1 | 0 | 0 |
| 13 | 1 | 0 | 1 |
| 14 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 |
| 16 | 1 | 2 | 0 |
| 17 | 1 | 2 | 1 |
| 18 | 1 | 3 | 0 |
| 19 | 1 | 3 | 1 |
| 20 | 1 | 4 | 0 |
| 21 | 1 | 4 | 1 |
| 22 | 1 | 5 | 0 |
| 23 | 1 | 5 | 1 |

# NUMA : Quiz 2

```
=====  Processor  =====

Processor name    : Intel(R) Xeon(R)  E5-2640 0
Packages(sockets) :    2
Cores             :    12
Processors(CPUs)  :    24
Cores per package :    6
Threads per core  :    2
```

```
$ numactl --hardware |grep cpus

node 0 cpus: 0 2 4 6 8 10 12 14 16 18 20 22
node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23
```

| Processor | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 0 | 2 | 0 |
| 5 | 0 | 2 | 1 |
| 6 | 0 | 3 | 0 |
| 7 | 0 | 3 | 1 |
| 8 | 0 | 4 | 0 |
| 9 | 0 | 4 | 1 |
| 10 | 0 | 5 | 0 |
| 11 | 0 | 5 | 1 |
| 12 | 1 | 0 | 0 |
| 13 | 1 | 0 | 1 |
| 14 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 |
| 16 | 1 | 2 | 0 |
| 17 | 1 | 2 | 1 |
| 18 | 1 | 3 | 0 |
| 19 | 1 | 3 | 1 |
| 20 | 1 | 4 | 0 |
| 21 | 1 | 4 | 1 |
| 22 | 1 | 5 | 0 |
| 23 | 1 | 5 | 1 |

# NUMA : Quiz 3

**Quiz: Can you identify the node structure here ?**

```
$numactl --hardware |grep cpus
----------------
mike441
----------------
node 0 cpus: 0 4 8 12 16 20 24 28 32 36
node 1 cpus: 1 5 9 13 17 21 25 29 33 37
node 2 cpus: 2 6 10 14 18 22 26 30 34 38
node 3 cpus: 3 7 11 15 19 23 27 31 35 39
```

| Proc | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 2 |
| 3 | 0 | 0 | 3 |
| 4 | 0 | 1 | 0 |
| 5 | 0 | 1 | 1 |
| 6 | 0 | 1 | 2 |
| 7 | 0 | 1 | 3 |
| 8 | 0 | 2 | 0 |
| 9 | 0 | 2 | 1 |
| 10 | 0 | 2 | 2 |
| 11 | 0 | 2 | 3 |
| 12 | 0 | 8 | 0 |
| 13 | 0 | 8 | 1 |
| 14 | 0 | 8 | 2 |
| 15 | 0 | 8 | 3 |
| 16 | 0 | 9 | 0 |
| 17 | 0 | 9 | 1 |
| 18 | 0 | 9 | 2 |
| 19 | 0 | 9 | 3 |
| 20 | 0 | 16 | 0 |
| 21 | 0 | 16 | 1 |
| 22 | 0 | 16 | 2 |
| 23 | 0 | 16 | 3 |
| 24 | 0 | 17 | 0 |
| 25 | 0 | 17 | 1 |
| 26 | 0 | 17 | 2 |
| 27 | 0 | 17 | 3 |
| 28 | 0 | 18 | 0 |
| 29 | 0 | 18 | 1 |
| 30 | 0 | 18 | 2 |
| 31 | 0 | 18 | 3 |
| 32 | 0 | 24 | 0 |
| 33 | 0 | 24 | 1 |
| 34 | 0 | 24 | 2 |
| 35 | 0 | 24 | 3 |
| 36 | 0 | 25 | 0 |
| 37 | 0 | 25 | 1 |
| 38 | 0 | 25 | 2 |
| 39 | 0 | 25 | 3 |

# NUMA : Quiz 3

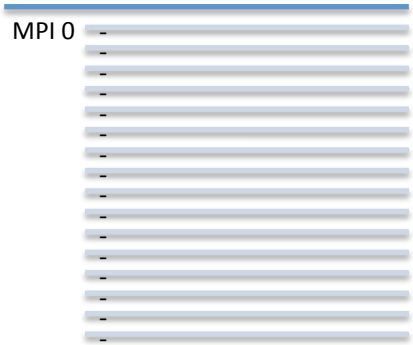
```
=====  Processor =====

Processor name    : Intel(R) Xeon(R)  E7- 4870
Packages(sockets) :    4
Cores             :    40
Processors(CPUs)  :    40
Cores per package :    10
Threads per core  :    1
```

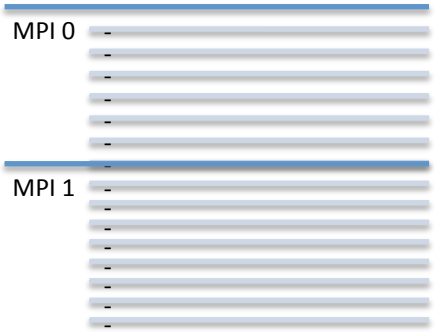
```
$numactl --hardware |grep cpus
----------------
mike441
----------------
node 0 cpus: 0 4 8 12 16 20 24 28 32 36
node 1 cpus: 1 5 9 13 17 21 25 29 33 37
node 2 cpus: 2 6 10 14 18 22 26 30 34 38
node 3 cpus: 3 7 11 15 19 23 27 31 35 39
```

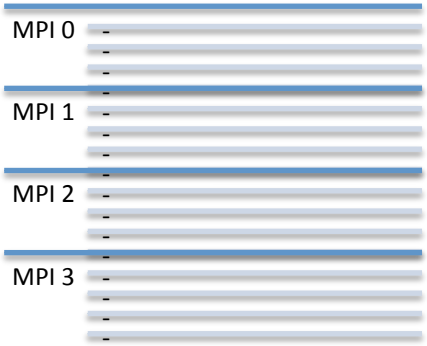
| Proc | Thread Id. | Core Id. | Package Id. |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 2 |
| 3 | 0 | 0 | 3 |
| 4 | 0 | 1 | 0 |
| 5 | 0 | 1 | 1 |
| 6 | 0 | 1 | 2 |
| 7 | 0 | 1 | 3 |
| 8 | 0 | 2 | 0 |
| 9 | 0 | 2 | 1 |
| 10 | 0 | 2 | 2 |
| 11 | 0 | 2 | 3 |
| 12 | 0 | 8 | 0 |
| 13 | 0 | 8 | 1 |
| 14 | 0 | 8 | 2 |
| 15 | 0 | 8 | 3 |
| 16 | 0 | 9 | 0 |
| 17 | 0 | 9 | 1 |
| 18 | 0 | 9 | 2 |
| 19 | 0 | 9 | 3 |
| 20 | 0 | 16 | 0 |
| 21 | 0 | 16 | 1 |
| 22 | 0 | 16 | 2 |
| 23 | 0 | 16 | 3 |
| 24 | 0 | 17 | 0 |
| 25 | 0 | 17 | 1 |
| 26 | 0 | 17 | 2 |
| 27 | 0 | 17 | 3 |
| 28 | 0 | 18 | 0 |
| 29 | 0 | 18 | 1 |
| 30 | 0 | 18 | 2 |
| 31 | 0 | 18 | 3 |
| 32 | 0 | 24 | 0 |
| 33 | 0 | 24 | 1 |
| 34 | 0 | 24 | 2 |
| 35 | 0 | 24 | 3 |
| 36 | 0 | 25 | 0 |
| 37 | 0 | 25 | 1 |
| 38 | 0 | 25 | 2 |
| 39 | 0 | 25 | 3 |

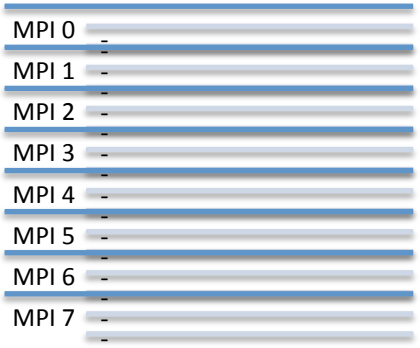
# Design decisions

- How many MPI processes per node, per socket?
  How many OpenMP threads per MPI process

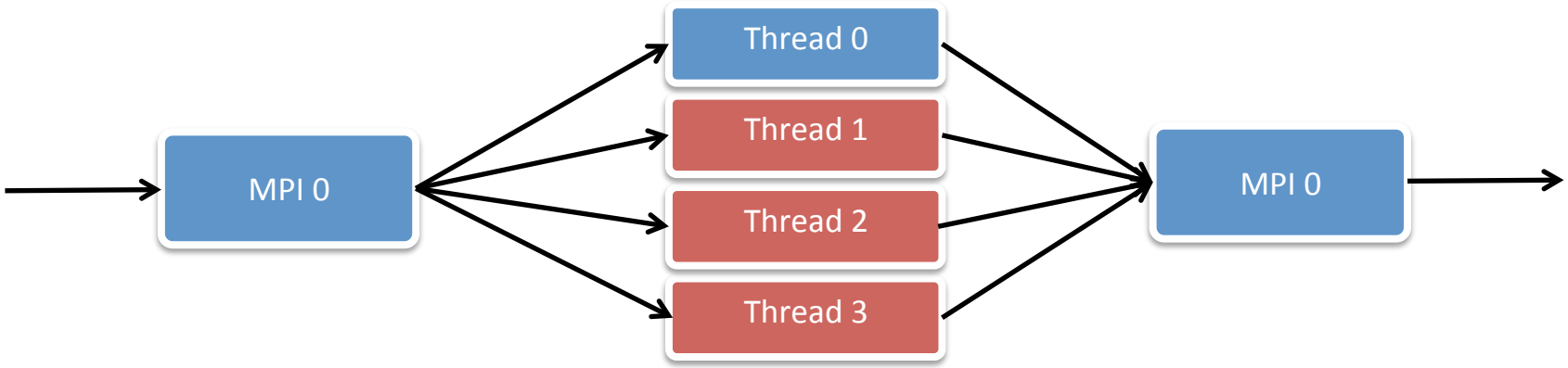


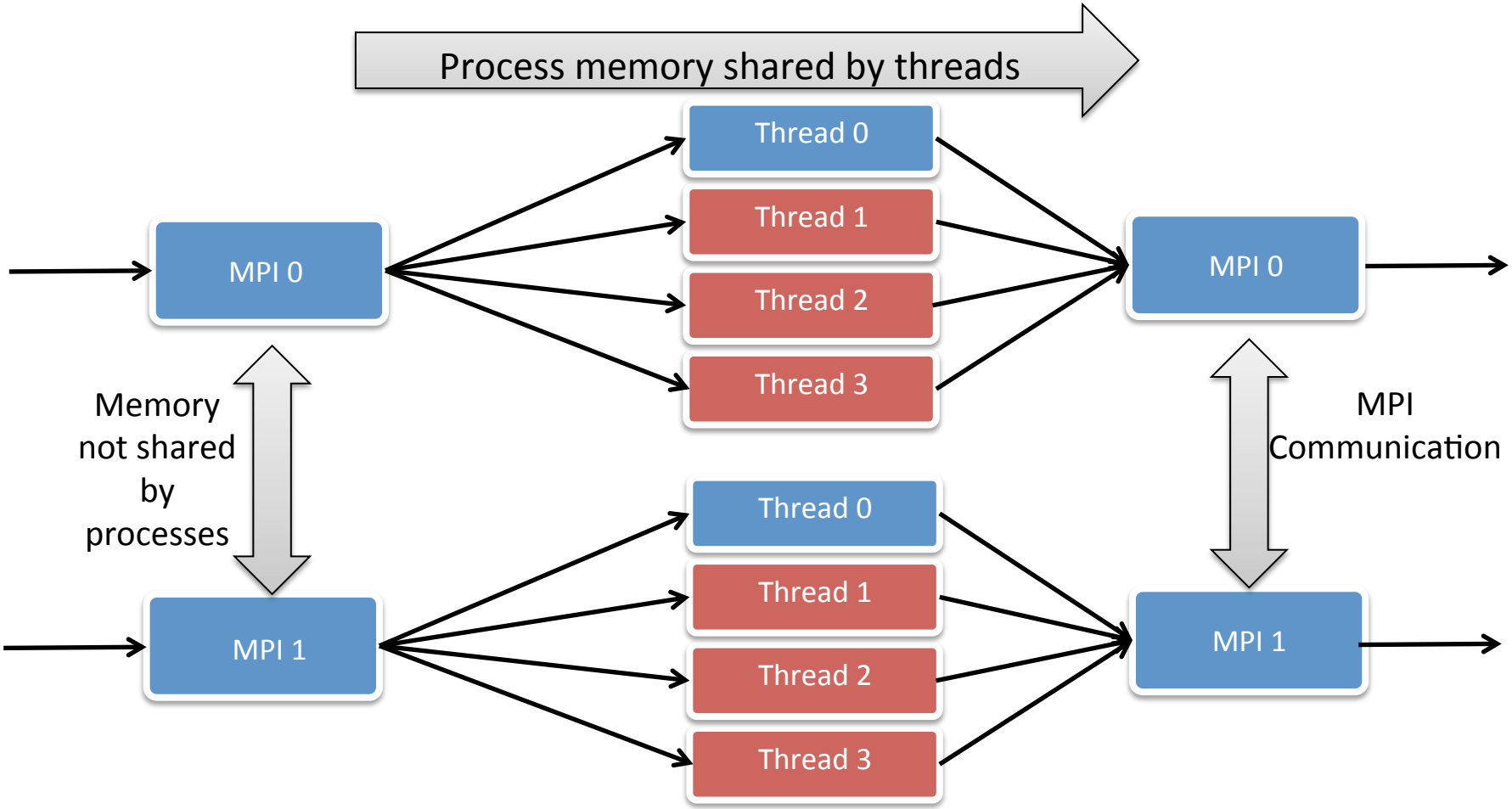



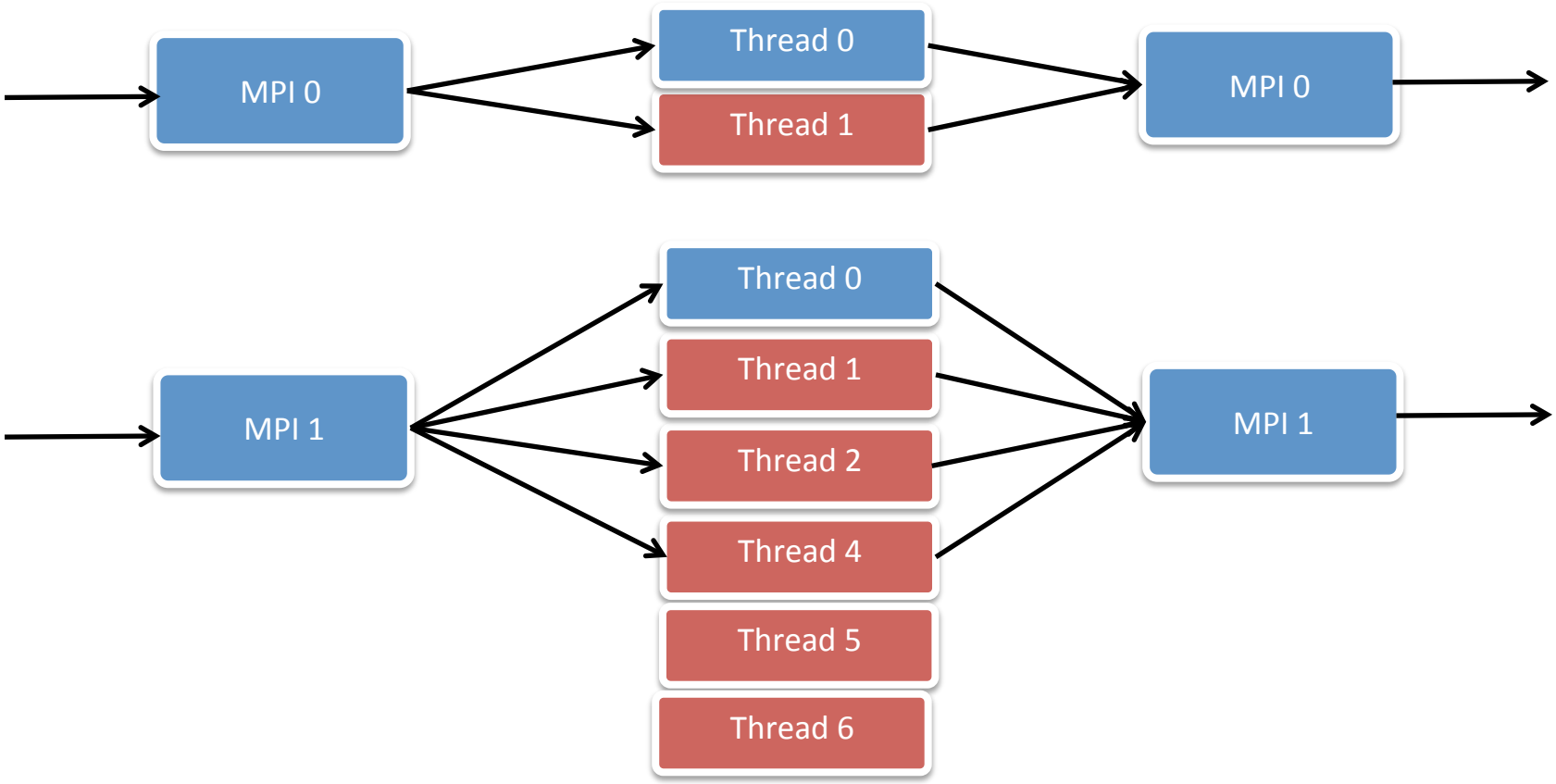
# Hybrid 0: Basic Model



# Hybrid 1: Basic Model



# Hybrid-2: Task Parallelism



# Communication support in MPI-2

**MPI-2 Init offers thread support query**

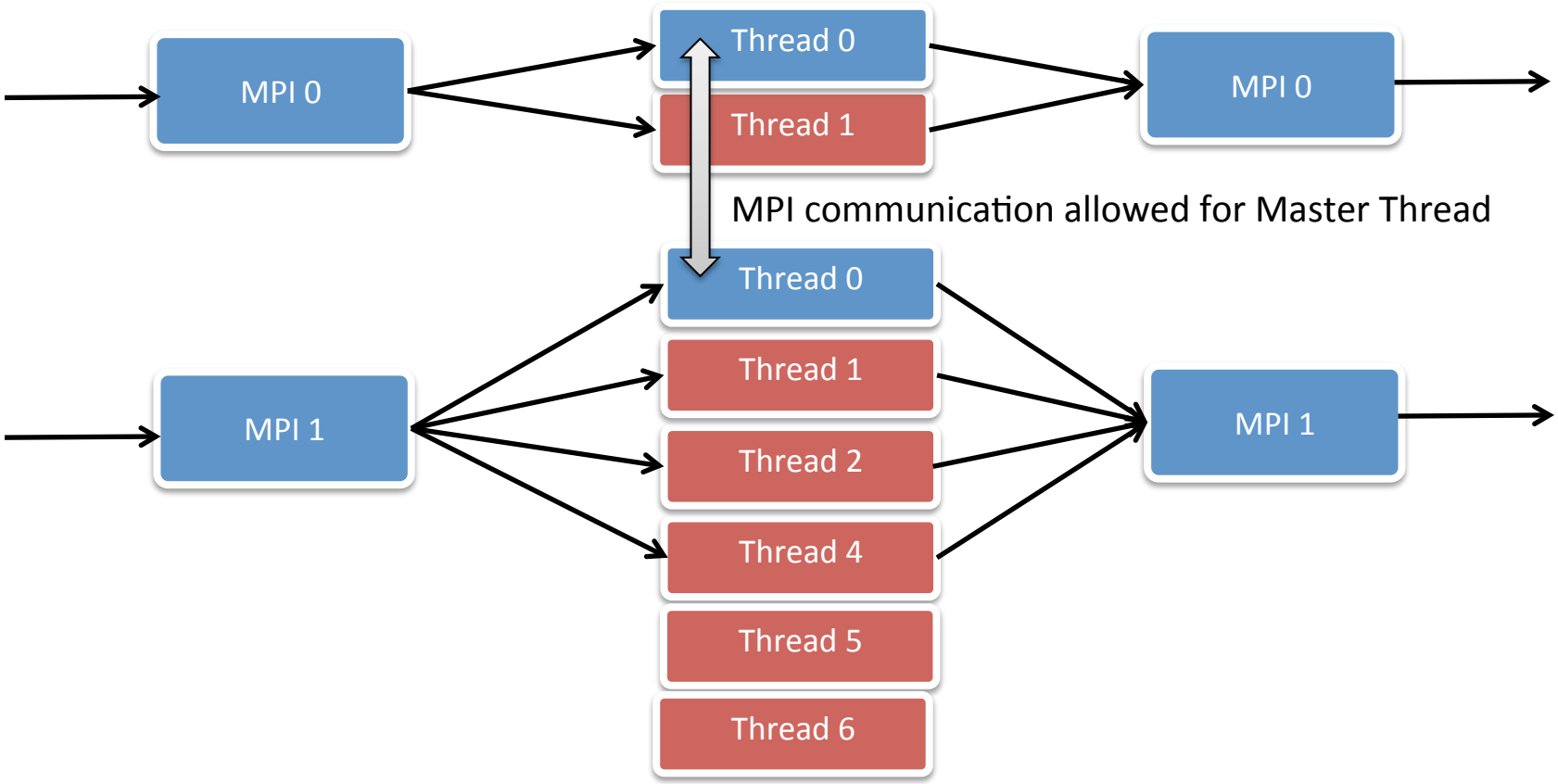
MPI_INIT_THREAD( request_thd_sup, provide_thd_sup )

***request_thd_sup***: Requested level of communication from a threaded region
**provide_thd_sup**: Actual provided level of communication from a threaded region

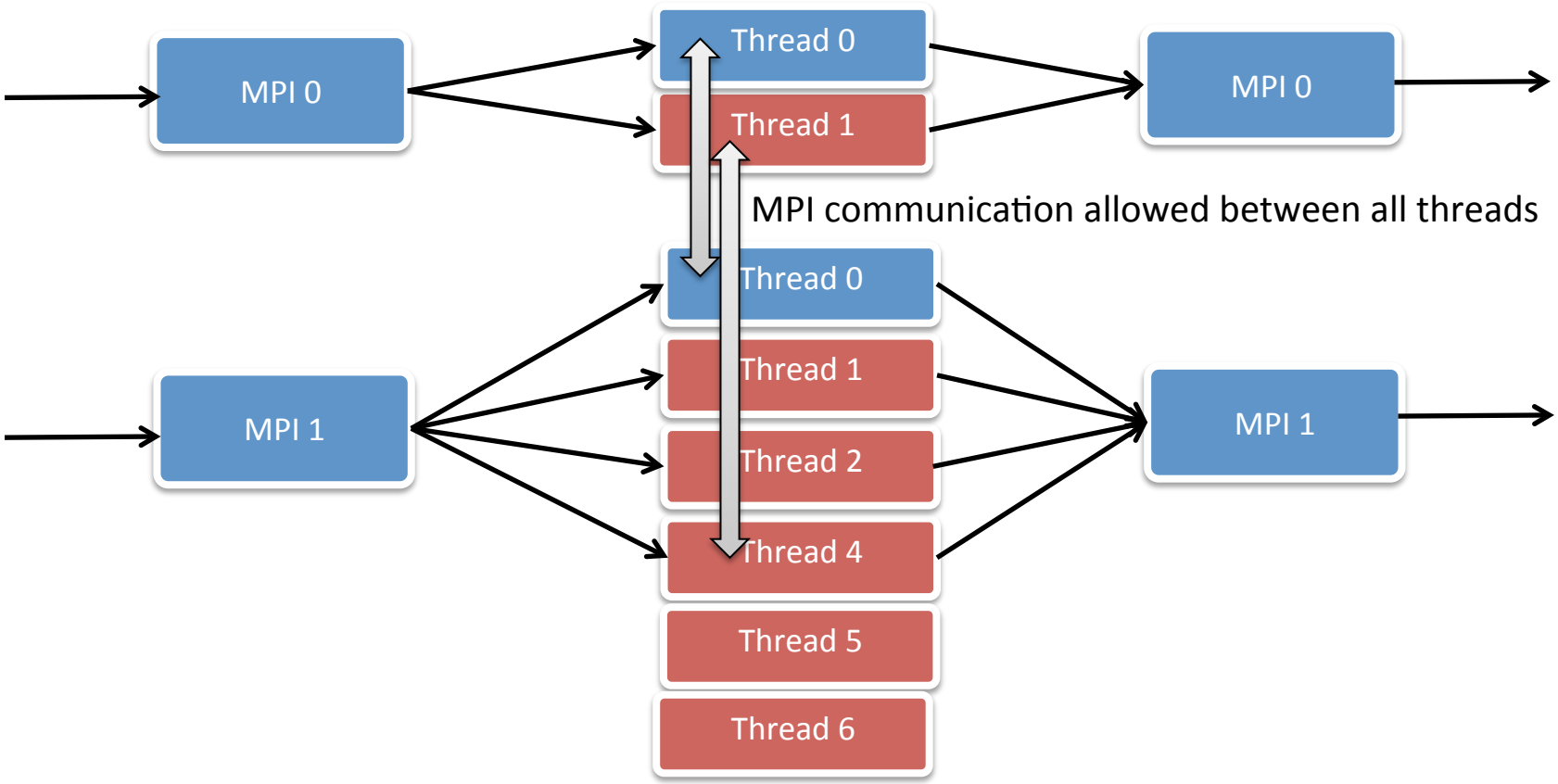
**Four levels of thread support on MPI-2**

| | |
|---|---|
| 0 : MPI_THREAD_SINGLE: | No threading for MPI processes |
| 1 : MPI_THREAD_FUNNELED: | Threading with Master Thread allowed MPI calls |
| 2 : MPI_THREAD_SERIALIZED: | Multiple threads make MPI calls, but serially |
| 3 : MPI_THREAD_MULTIPLE: | Any thread may make MPI calls at any time |

# MPI_THREAD_FUNNELED



# MPI_THREAD_{SERIALIZED,MULTIPLE}



# Thread support in MPI-2

**OpenMPI**

ompi_info will also show you the level of support

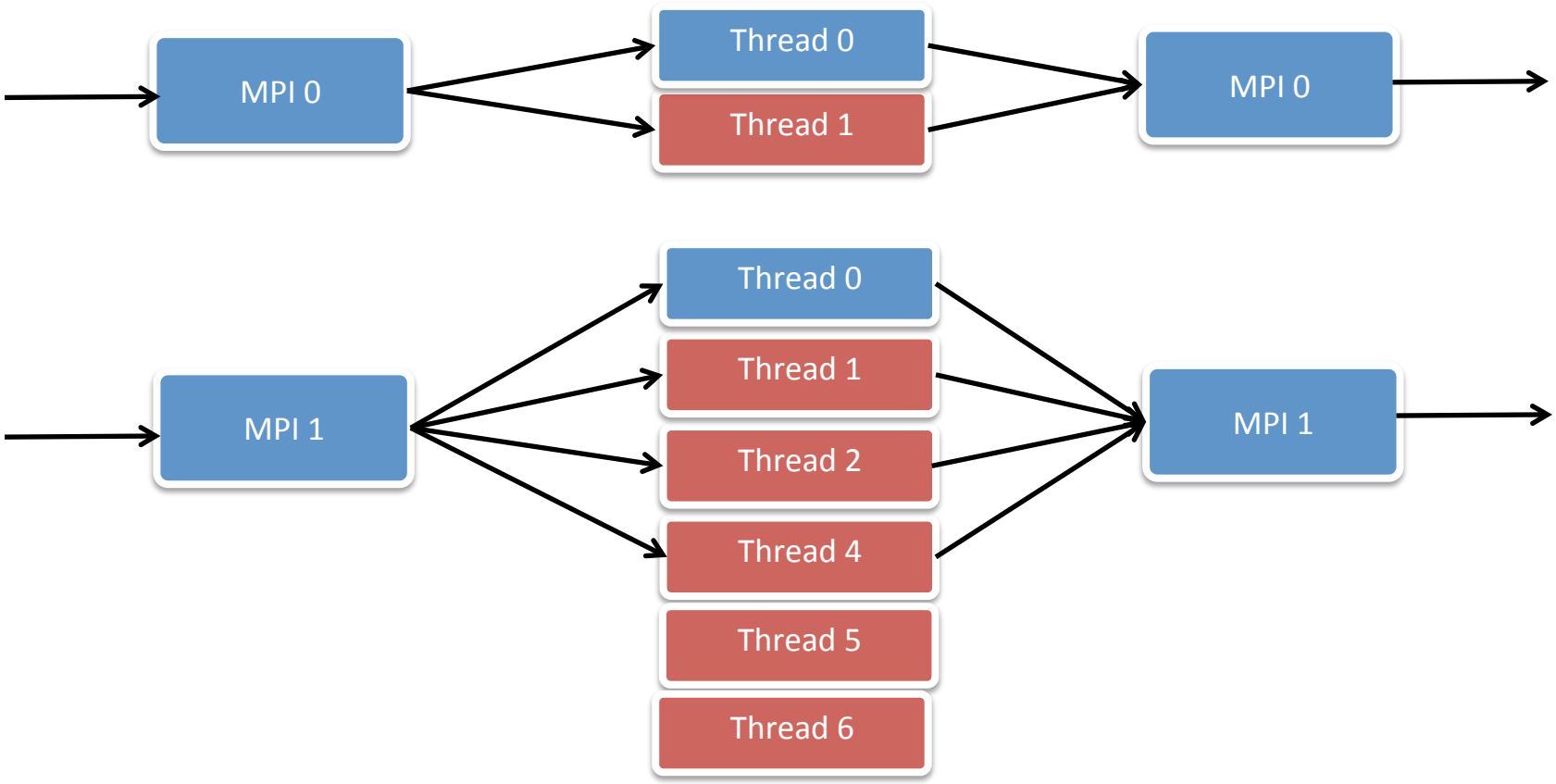
```
program thread_support

  use mpi
  implicit none

  integer :: mpierr
  integer :: thd_sup_req, thd_sup_pvd

  thd_sup_req=MPI_THREAD_MULTIPLE

  call MPI_INIT_THREAD( thd_sup_req, thd_sup_pvd, mpierr)
  call MPI_FINALIZE(mpierr)

  print *, "Requested", thd_sup_req
  print *, "Provided ", thd_sup_pvd

end program
~
~
```

# Communication support in MPI-2

```
program thread_support

 use mpi
 implicit none

 integer :: mpierr
 integer :: thd_sup_req, thd_sup_pvd

 thd_sup_req=MPI_THREAD_MULTIPLE

 call MPI_INIT_THREAD( thd_sup_req, thd_sup_pvd, mpierr)
 call MPI_FINALIZE(mpierr)

 print *, "Requested", thd_sup_req
 print *, "Provided ", thd_sup_pvd

end program
```

```
#include <mpi.h>
#include <stdio.h>

 int main(int argc, const char* argv[])
 {
   int provided;
   int requested=MPI_THREAD_MULTIPLE;

   MPI_Init_thread(NULL, NULL, requested, &provided);
   printf("MPI_Init_thread Requested = %d\n", requested);
   printf("MPI_Init_thread Provided  = %d\n", provided);
   MPI_Finalize();

   return 0;
 }
```

# MPI Communication within Threads



# Hello:Hybrid parallel program

**Compiling Hybrid Hello world:**

mpif90 –openmp hello_hybrid.f90

```
! Init MPI
    call MPI_Init(mpierr)

! Get Rank Size
    call MPI_COMM_Rank(MPI_COMM_WORLD, nrank, mpierr)
    call MPI_COMM_Size(MPI_COMM_WORLD, nproc, mpierr)

! Print rank
    call MPI_GET_PROCESSOR_NAME(pname, nlen, mpierr)

! Get Date hostname etc
    if (nrank==0) then
        call system('hostname && date && echo rank-pid $$')
    end if
    call MPI_Barrier(MPI_COMM_WORLD, mpierr)

! OpenMP
    !$OMP PARALLEL PRIVATE(itd,gtd)
      itd= omp_get_thread_num()
      gtd= omp_get_num_threads()
      grank= nrank*gtd + itd
      write(*,'(4(a6,i6),a2,a8)')"Gid ", grank, " Im ", nrank, &
                                 " of ", nproc, &
                                 " thd", itd,    &
                                 " on ", pname
    !$OMP FLUSH
    !$OMP BARRIER
      if (nrank==0 .and. itd==0) then
         call system('pstree -ap -u bthakur ')
```

# Hello: Hybrid parallel program

**Compiling Hybrid Hello world:**

mpif90 –openmp hello_hybrid.f90

MPI Initialization stuff

OpenMP region

```
! Init MPI
    call MPI_Init(mpierr)

! Get Rank Size
    call MPI_COMM_Rank(MPI_COMM_WORLD, nrank, mpierr)
    call MPI_COMM_Size(MPI_COMM_WORLD, nproc, mpierr)

! Print rank
    call MPI_GET_PROCESSOR_NAME(pname, nlen, mpierr)

! Get Date hostname etc
    if (nrank==0) then
        call system('hostname && date && echo rank-pid $$')
    end if
    call MPI_Barrier(MPI_COMM_WORLD, mpierr)

! OpenMP
    !$OMP PARALLEL PRIVATE(itd,gtd)
      itd= omp_get_thread_num()
      gtd= omp_get_num_threads()
      grank= nrank*gtd + itd
      write(*,'(4(a6,i6),a2,a8)')"Gid ", grank, " Im ", nrank, &
                                 " of ", nproc, &
                                 " thd", itd,    &
                                 " on ", pname
    !$OMP FLUSH
    !$OMP BARRIER
      if (nrank==0 .and. itd==0) then
         call system('pstree -ap -u bthakur ')
```

# Analyzing a Hybrid parallel program

Running a hybrid (mpi +openmp) process

```
bthakur@bthakur-1:~ — ssh — 81×28
[bthakur@mike400 hello]$ export OMP_NUM_THREADS=2
[bthakur@mike400 hello]$ mpirun -npernode 2 -hostfile hosts.2 -x OMP_NUM_THREADS
./a.out
mike400
Tue Sep 17 20:52:05 CDT 2013
rank-pid 112576
  Gid      0   Im      0   of      4   thd     0 omike400
  Gid      1   Im      0   of      4   thd     1 omike400
  Gid      4   Im      2   of      4   thd     0 omike401
  Gid      5   Im      2   of      4   thd     1 omike401
  Gid      2   Im      1   of      4   thd     0 omike400
  Gid      3   Im      1   of      4   thd     1 omike400
  Gid      6   Im      3   of      4   thd     0 omike401
  Gid      7   Im      3   of      4   thd     1 omike401
sshd,111276
  └─bash,111277
      └─mpirun,112568 -npernode 2 -hostfile hosts.2 -x OMP_NUM_THREADS ./a.out
          ├─a.out,112570
          |   ├─pstree,112581 -ap -u bthakur
          |   ├─{a.out},112572
          |   ├─{a.out},112574
          |   ├─{a.out},112579
          |   └─{a.out},112580
          └─a.out,112571
              ├─{a.out},112573
              ├─{a.out},112575
              ├─{a.out},112582
              └─{a.out},112583
```

Global ID
MPI Rank
OpenMP ID

# Analyzing a Hybrid parallel program

```
$ mpirun -n 32 -hostfile $PBS_NODEFILE --bind-to-core -report-bindings  ./a.out
```

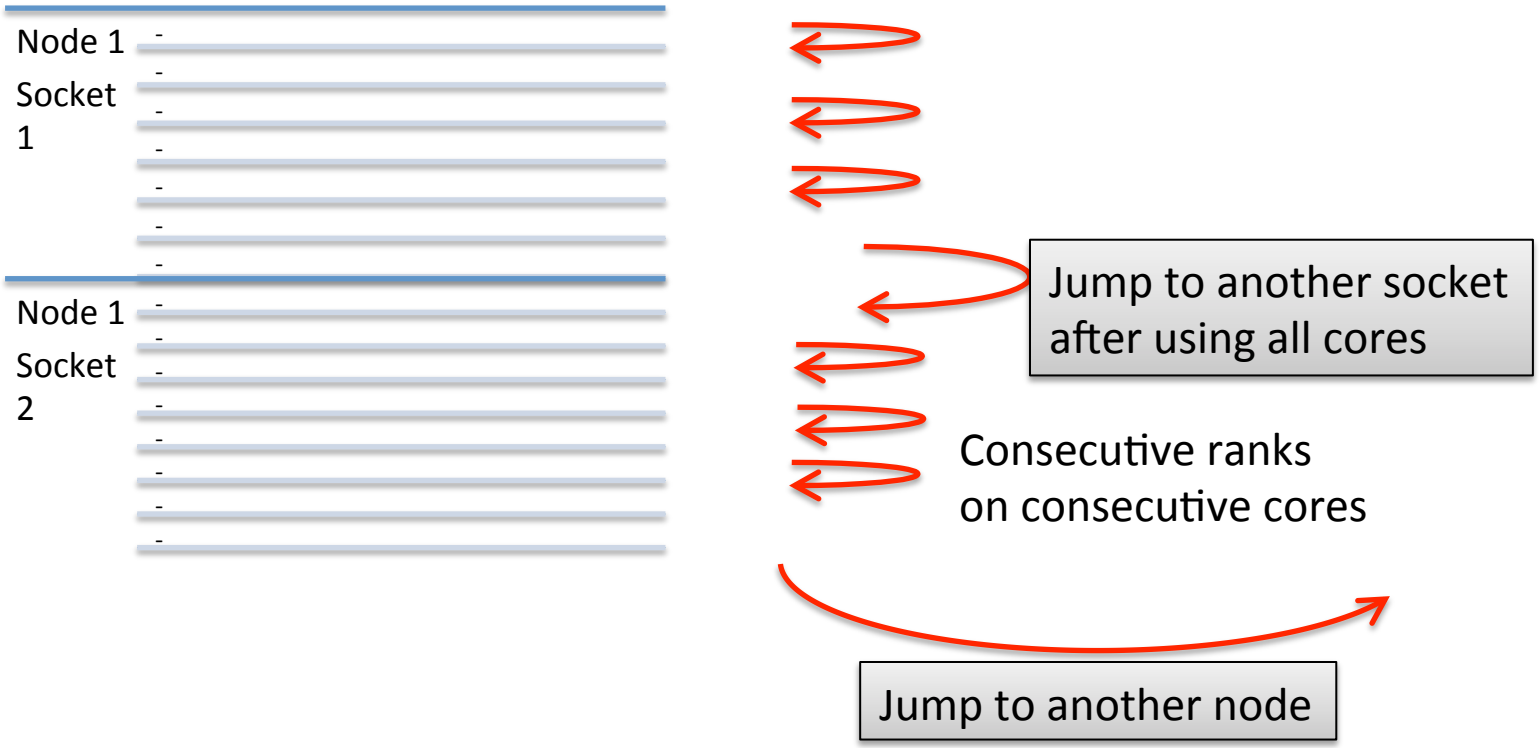
```
[mike253:45127] MCW rank 0 bound to socket 0[core 0]: [B . . . . . .][. . . . . .]
[mike253:45127] MCW rank 1 bound to socket 0[core 1]: [. B . . . . .][. . . . . .]
[mike253:45127] MCW rank 2 bound to socket 0[core 2]: [. . B . . . .][. . . . . .]
[mike253:45127] MCW rank 3 bound to socket 0[core 3]: [. . . B . . .][. . . . . .]
[mike253:45127] MCW rank 4 bound to socket 0[core 4]: [. . . . B . .][. . . . . .]
[mike253:45127] MCW rank 5 bound to socket 0[core 5]: [. . . . . B .][. . . . . .]
[mike253:45127] MCW rank 6 bound to socket 0[core 6]: [. . . . . B .][. . . . . .]
[mike253:45127] MCW rank 7 bound to socket 0[core 7]: [. . . . . B][. . . . . .]
[mike253:45127] MCW rank 8 bound to socket 1[core 0]: [. . . . . .][B . . . . . .]
[mike253:45127] MCW rank 9 bound to socket 1[core 1]: [. . . . . .][. B . . . . .]
[mike253:45127] MCW rank 10 bound to socket 1[core 2]: [. . . . . .][. . B . . . .]
[mike253:45127] MCW rank 11 bound to socket 1[core 3]: [. . . . . .][. . . B . . .]
[mike253:45127] MCW rank 12 bound to socket 1[core 4]: [. . . . . .][. . . . B . .]
[mike253:45127] MCW rank 13 bound to socket 1[core 5]: [. . . . . .][. . . . . B .]
[mike253:45127] MCW rank 14 bound to socket 1[core 6]: [. . . . . .][. . . . . B .]
[mike253:45127] MCW rank 15 bound to socket 1[core 7]: [. . . . . .][. . . . . B]
```

```
[mike254:98973] MCW rank 16 bound to socket 0[core 0]: [B . . . . . .][. . . . . .]
[mike254:98973] MCW rank 17 bound to socket 0[core 1]: [. B . . . . .][. . . . . .]
[mike254:98973] MCW rank 18 bound to socket 0[core 2]: [. . B . . . .][. . . . . .]
[mike254:98973] MCW rank 19 bound to socket 0[core 3]: [. . . B . . .][. . . . . .]
[mike254:98973] MCW rank 20 bound to socket 0[core 4]: [. . . . B . .][. . . . . .]
[mike254:98973] MCW rank 21 bound to socket 0[core 5]: [. . . . . B .][. . . . . .]
[mike254:98973] MCW rank 22 bound to socket 0[core 6]: [. . . . . B .][. . . . . .]
[mike254:98973] MCW rank 23 bound to socket 0[core 7]: [. . . . . B][. . . . . .]
[mike254:98973] MCW rank 24 bound to socket 1[core 0]: [. . . . . .][B . . . . . .]
[mike254:98973] MCW rank 25 bound to socket 1[core 1]: [. . . . . .][. B . . . . .]
[mike254:98973] MCW rank 26 bound to socket 1[core 2]: [. . . . . .][. . B . . . .]
[mike254:98973] MCW rank 27 bound to socket 1[core 3]: [. . . . . .][. . . B . . .]
[mike254:98973] MCW rank 28 bound to socket 1[core 4]: [. . . . . .][. . . . B . .]
[mike254:98973] MCW rank 29 bound to socket 1[core 5]: [. . . . . .][. . . . . B .]
[mike254:98973] MCW rank 30 bound to socket 1[core 6]: [. . . . . .][. . . . . B .]
[mike254:98973] MCW rank 31 bound to socket 1[core 7]: [. . . . . .][. . . . . B]
```

NOTE: The host file $PBS_NODEFILE has 16 entries for each node

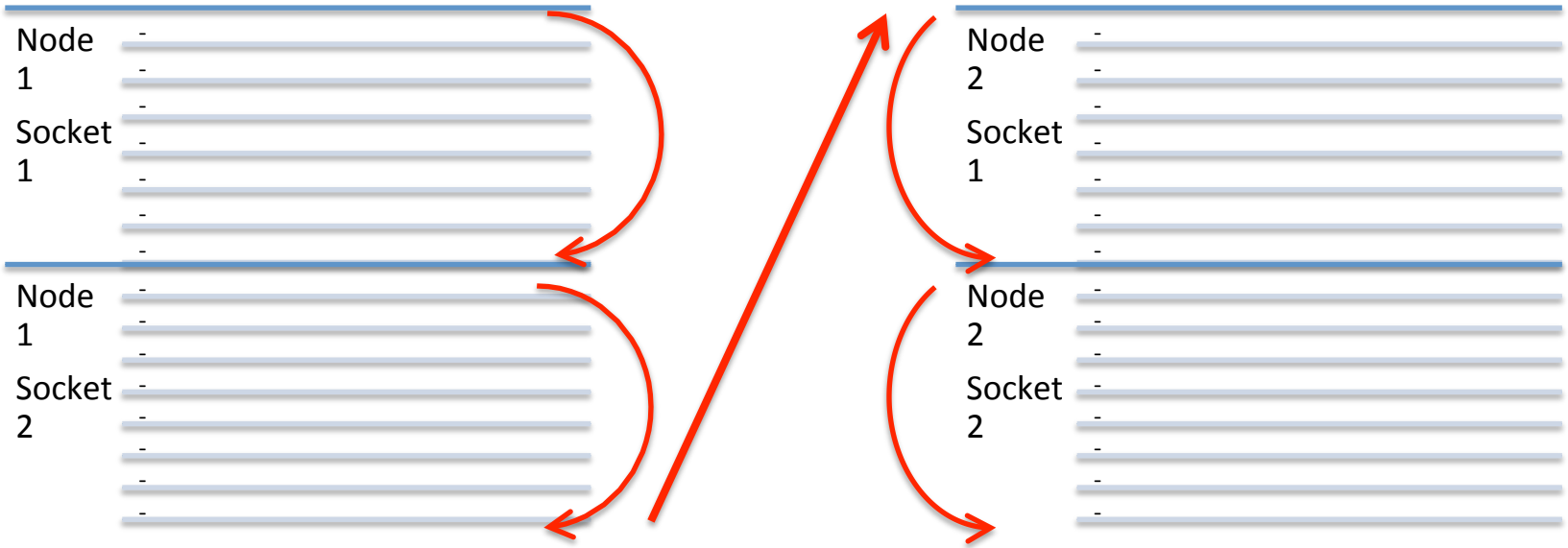
# Analyzing a Hybrid parallel program

```
$ mpirun -n 32 -hostfile $PBS_NODEFILE --bind-to-core -report-bindings  ./a.out
```



# Analyzing a Hybrid parallel program

To spawn mpi processes correctly you can either manipulate the hostfile or use other runtime options if available




Round robin socket and node

# Analyzing a Hybrid parallel program

```
$ mpirun -n 32 -hostfile hosts --bind-to-core -report-bindings  ./a.out
```

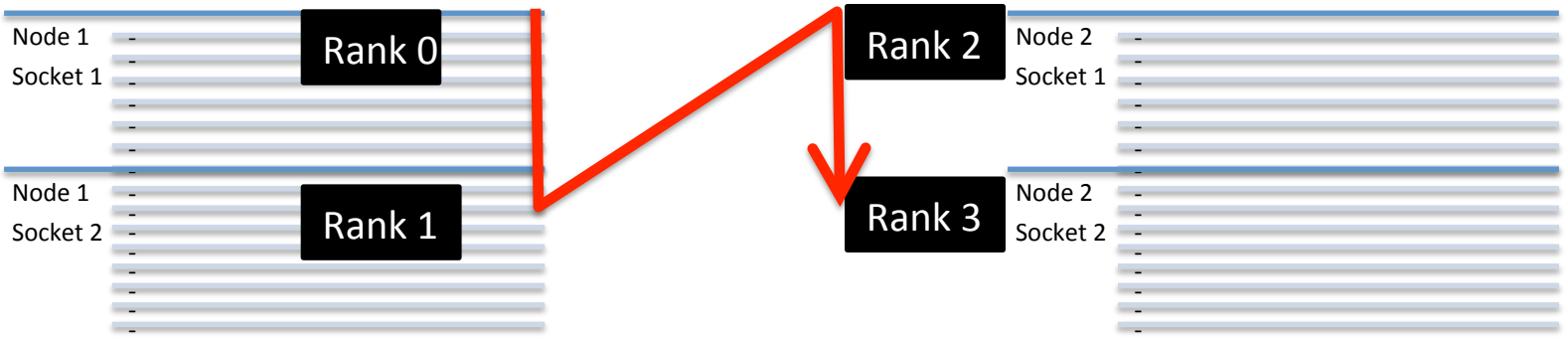
```
[mike253:45975] MCW rank 0 bound to socket 0[core 0]: [B . . . . . .][. . . . . .]
[mike253:45975] MCW rank 2 bound to socket 0[core 1]: [. B . . . . .][. . . . . .]
[mike253:45975] MCW rank 4 bound to socket 0[core 2]: [. . B . . . .][. . . . . .]
[mike253:45975] MCW rank 6 bound to socket 0[core 3]: [. . . B . . .][. . . . . .]
[mike253:45975] MCW rank 8 bound to socket 0[core 4]: [. . . . B . .][. . . . . .]
[mike253:45975] MCW rank 10 bound to socket 0[core 5]: [. . . . . B .][. . . . . .]
[mike253:45975] MCW rank 12 bound to socket 0[core 6]: [. . . . . B .][. . . . . .]
[mike253:45975] MCW rank 14 bound to socket 0[core 7]: [. . . . . B][. . . . . .]
[mike253:45975] MCW rank 16 bound to socket 1[core 0]: [. . . . . .][B . . . . . .]
[mike253:45975] MCW rank 18 bound to socket 1[core 1]: [. . . . . .][. B . . . . .]
[mike253:45975] MCW rank 20 bound to socket 1[core 2]: [. . . . . .][. . B . . . .]
[mike253:45975] MCW rank 22 bound to socket 1[core 3]: [. . . . . .][. . . B . . .]
[mike253:45975] MCW rank 24 bound to socket 1[core 4]: [. . . . . .][. . . . B . .]
[mike253:45975] MCW rank 26 bound to socket 1[core 5]: [. . . . . .][. . . . . B .]
[mike253:45975] MCW rank 28 bound to socket 1[core 6]: [. . . . . .][. . . . . B .]
[mike253:45975] MCW rank 30 bound to socket 1[core 7]: [. . . . . .][. . . . . B]
[mike254:99808] MCW rank 1 bound to socket 0[core 0]: [B . . . . . .][. . . . . .]
[mike254:99808] MCW rank 3 bound to socket 0[core 1]: [. B . . . . .][. . . . . .]
[mike254:99808] MCW rank 5 bound to socket 0[core 2]: [. . B . . . .][. . . . . .]
[mike254:99808] MCW rank 7 bound to socket 0[core 3]: [. . . B . . .][. . . . . .]
[mike254:99808] MCW rank 9 bound to socket 0[core 4]: [. . . . B . .][. . . . . .]
[mike254:99808] MCW rank 11 bound to socket 0[core 5]: [. . . . . B .][. . . . . .]
[mike254:99808] MCW rank 13 bound to socket 0[core 6]: [. . . . . B .][. . . . . .]
[mike254:99808] MCW rank 15 bound to socket 0[core 7]: [. . . . . B][. . . . . .]
[mike254:99808] MCW rank 17 bound to socket 1[core 0]: [. . . . . .][B . . . . . .]
[mike254:99808] MCW rank 19 bound to socket 1[core 1]: [. . . . . .][. B . . . . .]
[mike254:99808] MCW rank 21 bound to socket 1[core 2]: [. . . . . .][. . B . . . .]
[mike254:99808] MCW rank 23 bound to socket 1[core 3]: [. . . . . .][. . . B . . .]
[mike254:99808] MCW rank 25 bound to socket 1[core 4]: [. . . . . .][. . . . B . .]
[mike254:99808] MCW rank 27 bound to socket 1[core 5]: [. . . . . .][. . . . . B .]
[mike254:99808] MCW rank 29 bound to socket 1[core 6]: [. . . . . .][. . . . . B .]
[mike254:99808] MCW rank 31 bound to socket 1[core 7]: [. . . . . .][. . . . . B]
```

NOTE: The host file hosts has only 2 unique entries

# Analyzing a Hybrid parallel program

**2 Nodes, 2 MPI processes per node, 8 OMP Threads per process**

```
$ mpirun -n 4 -hostfile $PBS_NODEFILE --bysocket --bind-to-socket -report-bindings -display-map ./a.out
[mike253:42506] MCW rank 0 bound to socket 0[core 0-7]: [B B B B B B B B]   [. . . . . . . .]
[mike253:42506] MCW rank 1 bound to socket 1[core 0-7]: [. . . . . . . .]   [B B B B B B B B]
[mike253:42506] MCW rank 2 bound to socket 0[core 0-7]: [B B B B B B B B]   [. . . . . . . .]
[mike253:42506] MCW rank 3 bound to socket 1[core 0-7]: [. . . . . . . .]   [B B B B B B B B]
```




NOTE: The host file $PBS_NODEFILE has 16 entries for each node

# Analyzing a Hybrid parallel program

**2 Nodes, 2 MPI processes per node, 8 OMP Threads per process**

```
$ mpirun -n 4 -hostfile hosts --bysocket --bind-to-socket -report-bindings -display-map ./a.out
[mike015:62126] MCW rank 0 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike015:62126] MCW rank 2 bound to socket 1[core 0-7]: [. . . . . .][B B B B B B B B]
[mike017:34768] MCW rank 1 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:34768] MCW rank 3 bound to socket 1[core 0-7]: [. . . . . .][B B B B B B B B]
```

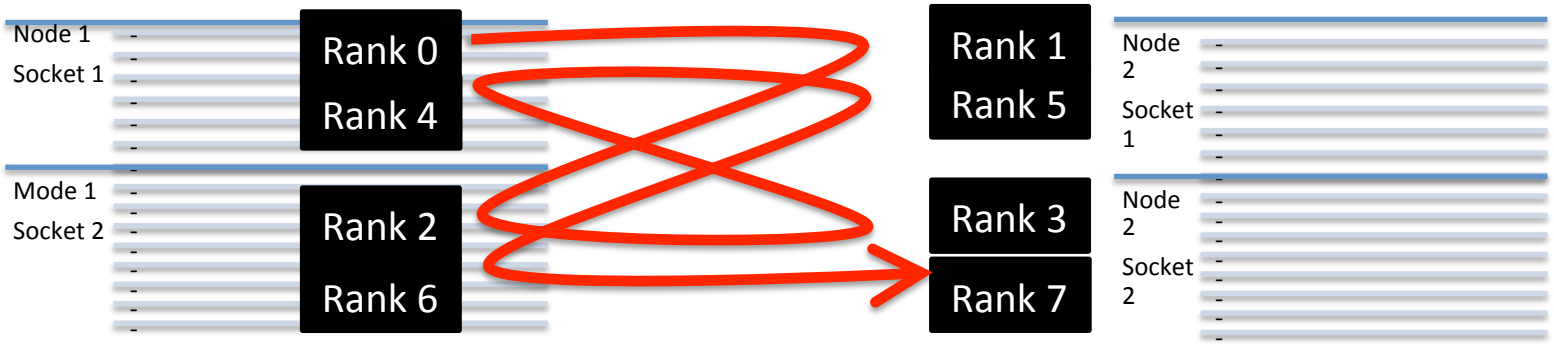

NOTE: the hostfile here has unique names

# Analyzing a Hybrid parallel program

**2 Nodes, 4 MPI processes per node, 4 OMP Threads per process**

$ **mpirun -n 8 -hostfile hosts --bysocket** --bind-to-socket -report-bindings -display-map ./a.out

[mike015:62526] MCW rank 6 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]
[mike015:62526] MCW rank 0 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike015:62526] MCW rank 2 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]
[mike015:62526] MCW rank 4 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:35161] MCW rank 1 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:35161] MCW rank 3 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]
[mike017:35161] MCW rank 5 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:35161] MCW rank 7 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]



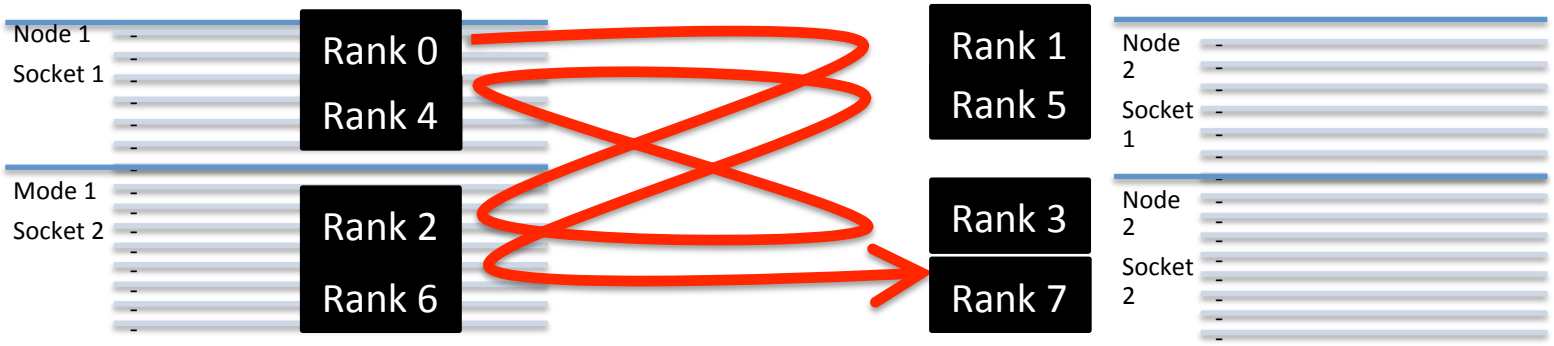

NOTE: The hostfile here has unique names

# Analyzing a Hybrid parallel program

**2 Nodes, 4 MPI processes per node, 4 OMP Threads per process**

$ **mpirun -n 8 -hostfile hosts --bysocket** --bind-to-socket -report-bindings -display-map ./a.out

[mike015:62526] MCW rank 6 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]
[mike015:62526] MCW rank 0 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike015:62526] MCW rank 2 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]
[mike015:62526] MCW rank 4 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:35161] MCW rank 1 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:35161] MCW rank 3 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]
[mike017:35161] MCW rank 5 bound to socket 0[core 0-7]: [B B B B B B B B][. . . . . .]
[mike017:35161] MCW rank 7 bound to socket 1[core 0-7]: [. . . . . .] [B B B B B B B B]




NOTE: The hostfile here has unique names

# Exercise

- Run the hello hybrid program on two nodes using:
  - 1 mpi process per node, 16 openmp threads per process
  - 2 mpi processes per node, 8 openmp threads per process
  - 4 mpi processes per node, 4 openmp threads per process
  - 8 mpi processes per node, 2 openmp threads per process

- Run the executable "shelob.exe.a" as in previous step:
  - What is difference between total time and cpu time?
    Hint: Run with "time mpirun …" to see real time
  - Calculate the total cpu utilization as a percentage of cpu time vs elapsed time from the output.
  - Find the most efficient combination of mpi and openmp threads