

Семинар 4 (26)

Программирование сопроцессора Intel Xeon Phi (MPI)

Михаил Курносов

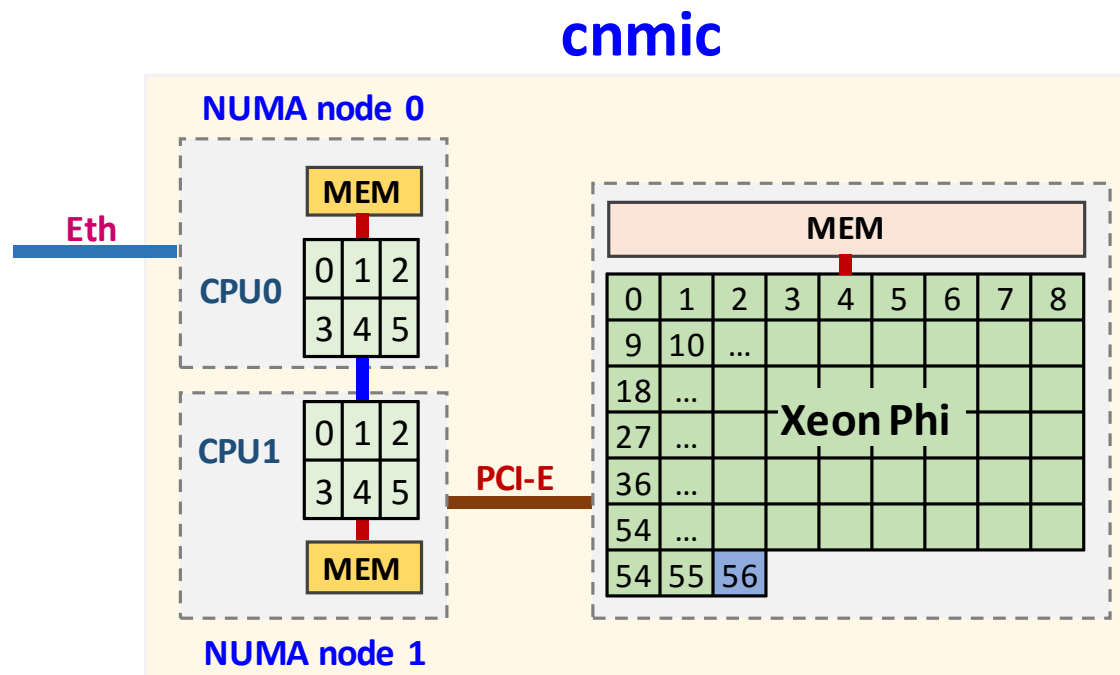
E-mail: mkurnosov@gmail.com

WWW: www.mkurnosov.net

Цикл семинаров «Основы параллельного программирования»
Институт физики полупроводников им. А. В. Ржанова СО РАН
Новосибирск, 2016

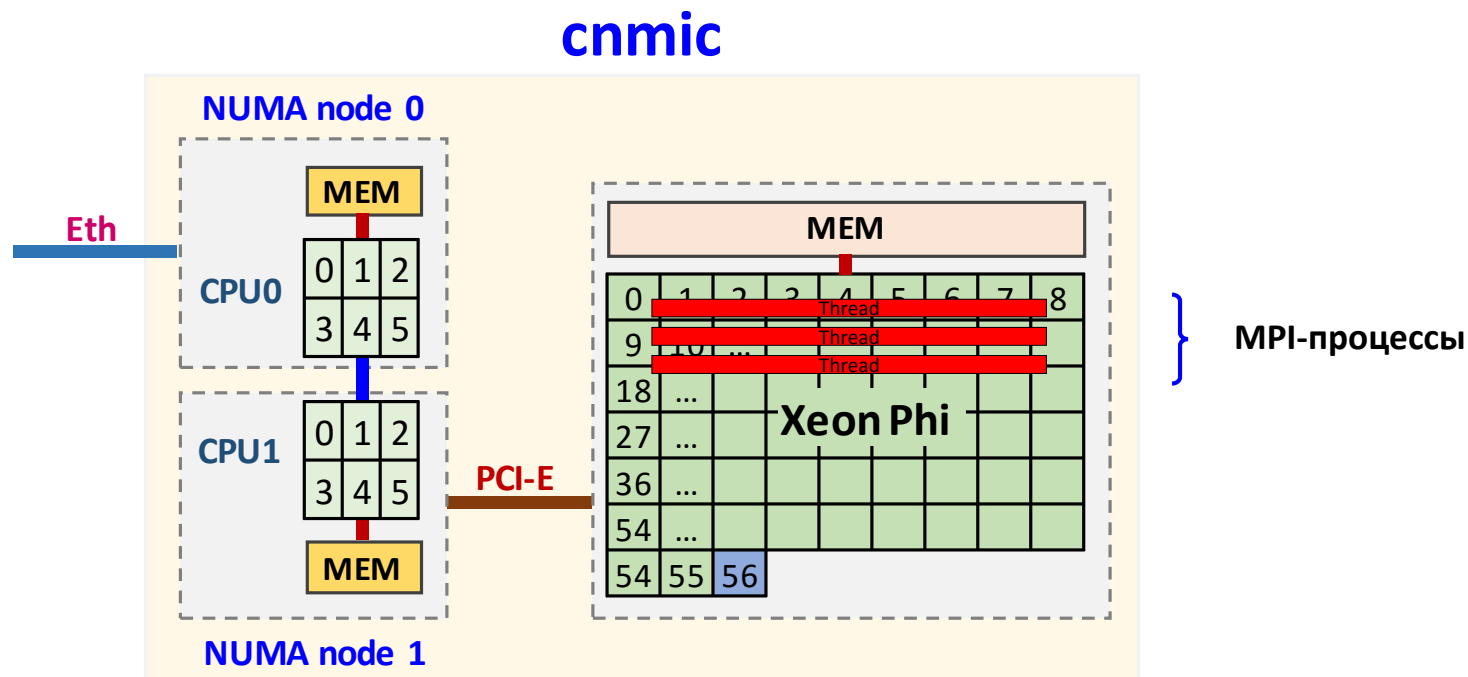
Использование MPI на Intel Xeon Phi

- **Native model** – все процессы MPI-программы выполняются на сопроцессоре
- Программа копируется на сопроцессор и запускается



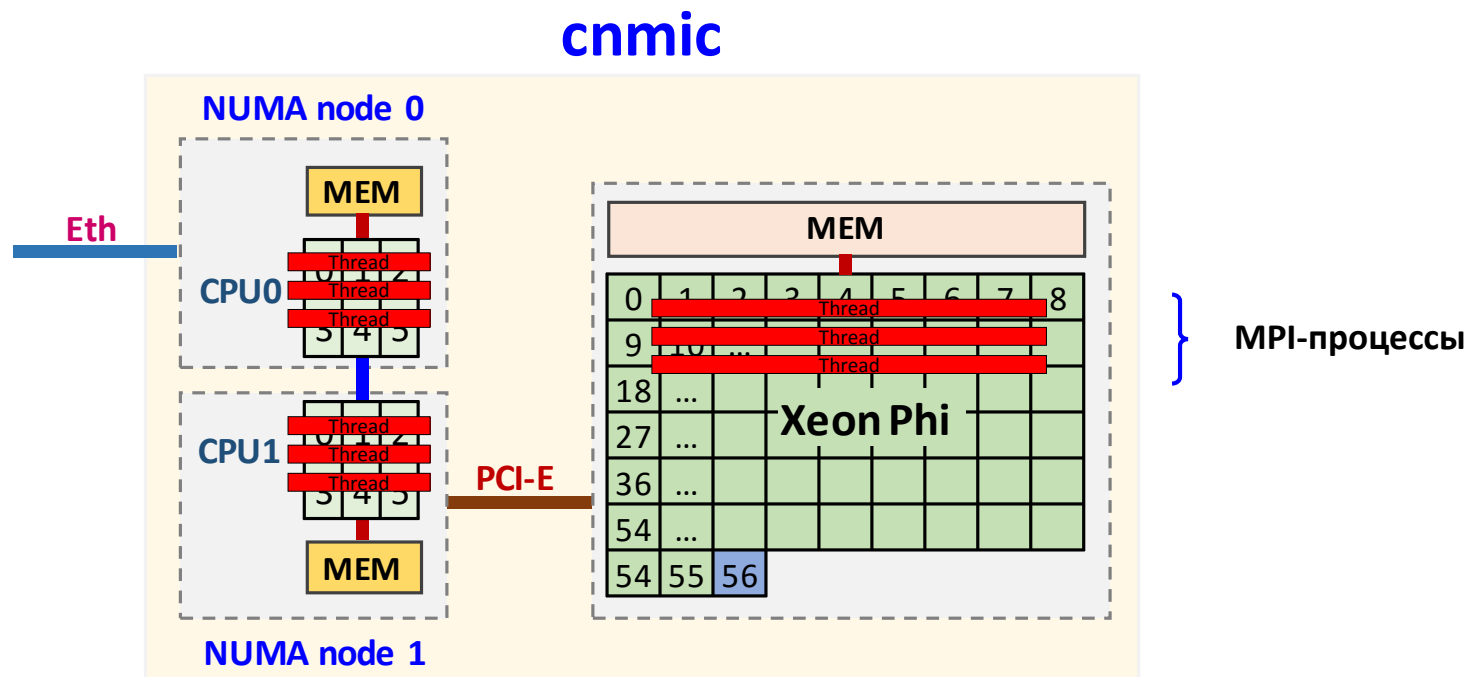
Использование MPI на Intel Xeon Phi

- **Native model** – все процессы MPI-программы выполняются на сопроцессоре
- Программа копируется на сопроцессор и запускается



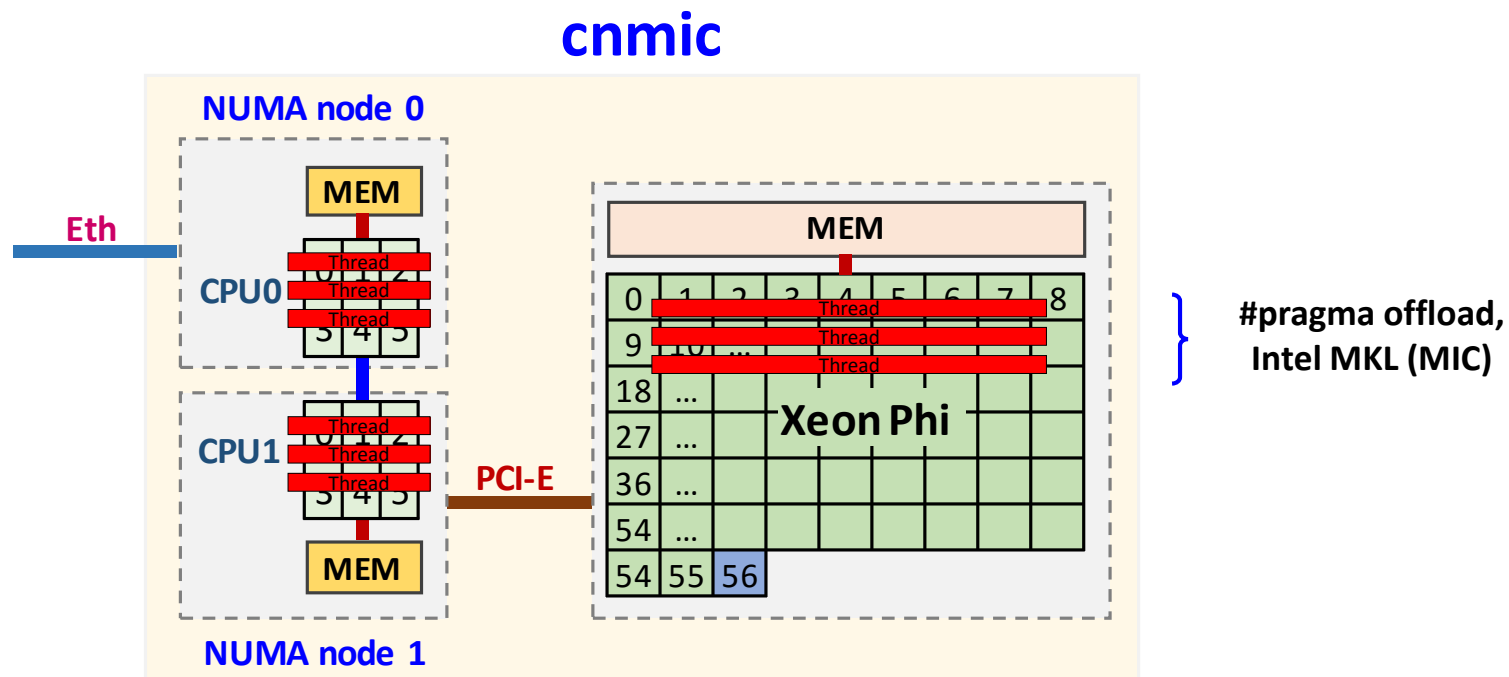
Использование MPI на Intel Xeon Phi

- **Symmetric model** – процессы программы выполняются на хост-машине и сопроцессоре



Использование MPI на Intel Xeon Phi

- **Offload model** – процессы MPI-программы выполняются на хост-машине и выгружают (offload) часть кода на сопроцессор
- `#pragma offload`, Intel MKL (MIC)



MPI Native mode

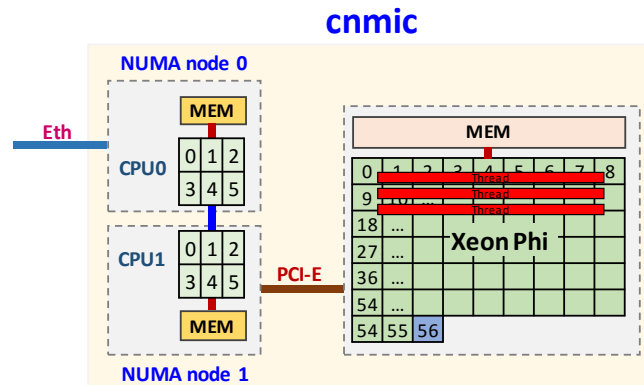
```
#include <unistd.h>
#include <stdio.h>
#include <mpi.h>

int main(int argc, char **argv)
{
    int rank, len;
    char procname[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc, &argv);
    MPI_Get_processor_name(procname, &len);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    printf("Rank %d on %s\n", rank, procname);

    MPI_Finalize();
    return 0;
}
```



Сборка программы

```
$ cat ./build.sh
```

```
#!/bin/sh
```

```
mpiicc -mmic -Wall -std=c99 -O2 -o \
    hello ./hello.c
```

Запуск программы

```
$ cat ./launch.sh
```

```
#!/bin/sh
```

```
TMP=`mktemp -p /home/micshare/tmp -u`
cp -f ./hello $TMP
export I_MPI_MIC=enable
```

mic0

```
mpirun -n 3 -f hosts $TMP
```

```
rm -f $TMP
```

```
$ ./launch.sh
```

```
Rank 2 on cnmic-mic0
```

```
Rank 0 on cnmic-mic0
```

```
Rank 1 on cnmic-mic0
```

MPI Symmetric mode

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

```
int main(int argc, char **argv)
```

```
$ ./launch.sh
```

```
Rank 0 on cnmic
```

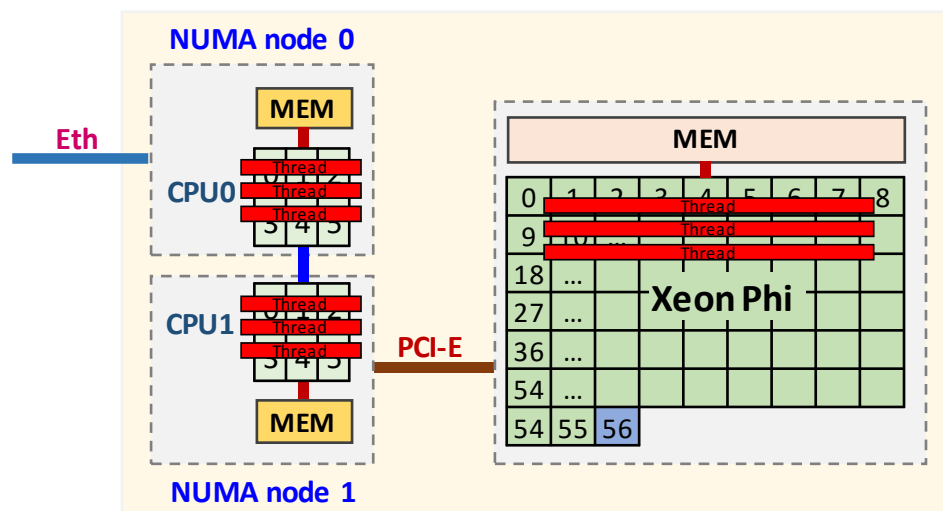
```
Rank 1 on cnmic
```

```
Rank 2 on cnmic-mic0
```

```
Rank 4 on cnmic-mic0
```

```
Rank 3 on cnmic-mic0
```

cnmic



```
# Сборка программы
```

```
$ cat ./build.sh
```

```
#!/bin/sh
```

```
# Binary for Xeon Phi
```

```
mpiicc -mmic -Wall -std=c99 -O2 -o hello.mic ./hello.c
```

```
# Binary for Host
```

```
mpiicc -Wall -std=c99 -O2 -o hello ./hello.c
```

```
# Запуск программы Host + Phi - Вариант 1
```

```
$ cat ./build.sh
```

```
#!/bin/sh
```

```
# Copy program to the NFS folder
```

```
TMP=`mktemp -p /home/micshare/tmp -u`
```

```
cp -f ./hello.mic $TMP
```

```
# Launch the program (see hosts file)
```

```
export I_MPI_MIC=enable
```

```
export I_MPI_FABRICS=shm:tcpmpi
```

```
run -n 2 -host cnmic ./hello : -n 3 -host mic0 $TMP
```

```
rm -f $TMP
```

MPI Symmetric mode

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

```
int main(int argc, char **argv)
```

```
$ ./launch.sh
```

```
Rank 0 on cnmic
```

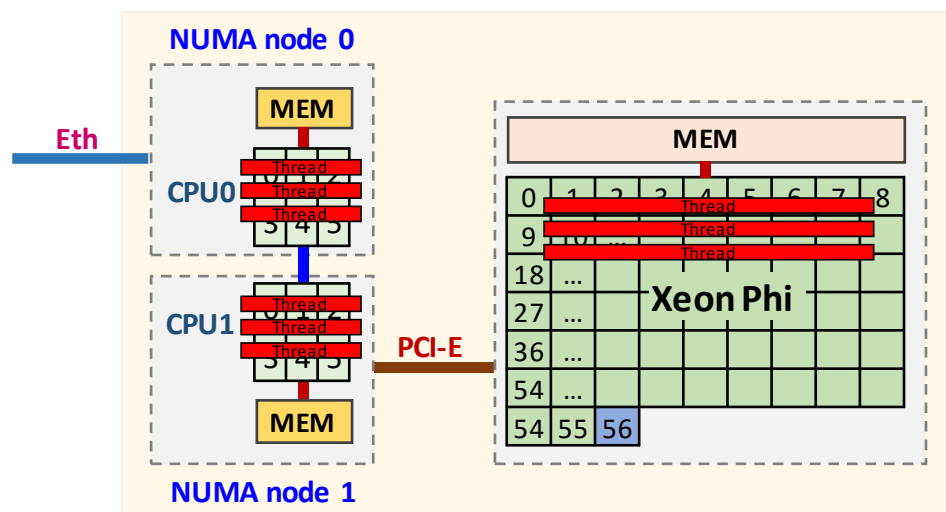
```
Rank 1 on cnmic
```

```
Rank 2 on cnmic-mic0
```

```
Rank 4 on cnmic-mic0
```

```
Rank 3 on cnmic-mic0
```

cnmic



Сборка программы

```
$ cat ./build.sh
```

```
#!/bin/sh
```

Binary for Xeon Phi

```
mpiicc -mmic -Wall -std=c99 -O2 -o hello.mic ./hello.c
```

Binary for Host

```
mpiicc -Wall -std=c99 -O2 -o hello ./hello.c
```

Запуск программы Host + Phi - Вариант 2

```
$ cat ./build.sh
```

Copy program to the NFS folder

```
TMP=`mktemp -p /home/micshare/tmp -u`
```

```
cp -f ./hello.mic "$TMP.mic"
```

```
cp -f ./hello $TMP
```

Launch the program (see hosts file)

```
export I_MPI_MIC=enable
```

```
export I_MPI_MIC_POSTFIX=.mic
```

```
export I_MPI_FABRICS=shm:tcp
```

```
mpirun -machinefile hosts $TMP
```

```
rm -f $TMP "$TMP.mic"
```

cnmic:2

mic0:3

MPI Offload mode

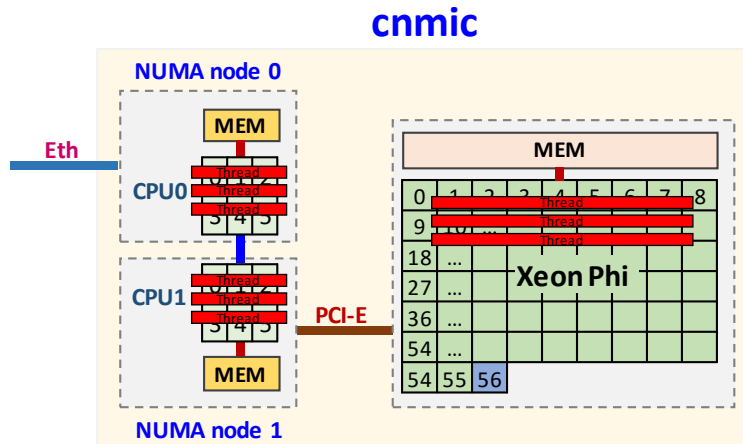
```
int main(int argc, char **argv)
{
    int rank, len;
    char procname[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc, &argv);
    MPI_Get_processor_name(procname, &len);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    printf("Rank %d on %s\n", rank, procname);

    #pragma offload target(mic) in(rank)
    {
        printf("Hello, from Xeon Phi (offloaded from proc %d)\n", rank);
    }

    MPI_Finalize();
    return 0;
}
```



Сборка программы

```
$ cat ./build.sh
```

```
#!/bin/sh
```

```
mpiicc -Wall -std=c99 -O2 -o hello ./hello.c
```

Запуск программы

```
$ cat ./launch.sh
```

```
#!/bin/sh
```

```
mpirun -n 4 -host cnmic ./hello ./hello
```

```
$ ./launch.sh
```

```
Rank 0 on cnmic
```

```
Rank 1 on cnmic
```

```
Rank 2 on cnmic
```

```
Rank 3 on cnmic
```

```
Hello, from Xeon Phi (offloaded from proc 3)
```

```
Hello, from Xeon Phi (offloaded from proc 0)
```

```
Hello, from Xeon Phi (offloaded from proc 2)
```

```
Hello, from Xeon Phi (offloaded from proc 1)
```

Конфигурация гибридного NUMA-узла cnmic

```
$ hwloc-ls
```

```
Machine (64GB)
```

```
  NUMANode L#0 (P#0 32GB)
```

```
    Socket L#0 + L3 L#0 (15MB)
```

```
      L2 L#0 (256KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0
```

```
    HostBridge L#0
```

```
      PCIBridge
```

```
        PCI 10de:0f02
```

```
          GPU L#0 "card0"
```

```
          GPU L#1 "renderD128"
```

```
          GPU L#2 "controlD64"
```

```
      PCI 8086:8d62
```

```
      PCIBridge
```

```
        PCI 8086:1533
```

```
          Net L#3 "enp6s0"
```

```
      PCIBridge
```

```
        PCI 8086:1533
```

```
          Net L#4 "enp7s0"
```

```
      PCIBridge
```

```
        PCI 1b21:0612
```

```
      PCI 8086:8d02
```

```
        Block L#5 "sda"
```

```
  NUMANode L#1 (P#1 32GB)
```

```
    Socket L#1 + L3 L#1 (15MB)
```

```
      L2 L#6 (256KB) + L1d L#6 (32KB) + L1i L#6 (32KB) + Core L#6
```

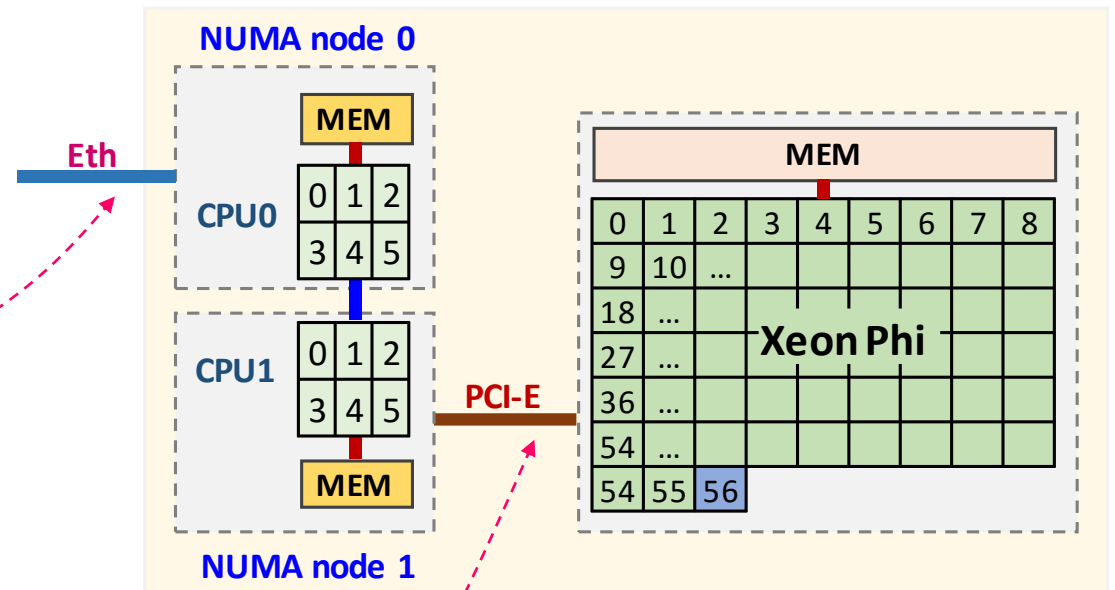
```
    HostBridge L#5
```

```
      PCIBridge
```

```
        PCI 8086:225d
```

```
          CoProc L#6 "mic0"
```

cnmic



Задание

- В каталоге GOL_2D_native размещена MPI-программа моделирования игры «жизнь» (Game of Life, GOL, 2D-декомпозиция), а также скрипты сборки и запуска программы в режиме native. Оцените ускорение программы в этом режиме.
- Запустите программу GOL_2D в симметричном режиме. Оцените ее ускорение.

