

## Hands-on “Working with Intel Xeon and Intel Xeon Phi Architecture”

git clone <https://github.com/intel-unesp-mcp/workshop-HPC-ML.git>

### 1. Intel Xeon Phi Management

1.1. Execute the following command to verify if the service that controls the devices is up and running:

```
service mpss status
```

1.2. Execute the following command to obtain information about the devices:

```
micinfo or mpssinfo
```

How Many Intel Xeon Phi devices are deployed?

1.3. Execute the following commands in the main host and on one mic device.

- The following command returns the number of cores
  - `cat /proc/cpuinfo | grep 'cpu cores' | uniq`
- The following command returns the number of threads
  - `cat /proc/cpuinfo | grep processor | wc -l`

How many cores and threads is available on main host and on mic Device?

### 2. Intel Xeon and Intel Xeon Phi Compiling and Running

The code **helloWorld.c** shows the amount of logical threads available.

2.1. Compile and run in Intel Xeon using the following commands:

```
icc helloWorld.c -o helloWorld
./helloWorld
```

2.2. Compile to Intel Xeon Phi:

```
icc helloWorld.c -o helloWorld.mic -mmic
```

2.3. Run in Intel Xeon Phi using micnativeloadex:

```
micnativeloadex helloWorld.mic
```

2.4. Run in Intel Xeon Phi using SSH:

```
cp helloWorld.mic ~/
ssh mic0
~/helloWorld.mic
```

### 3. Offload

The code **helloWorldOffload.c** performs the offload of a region of code to Intel Xeon Phi.

3.1. Compile and run on Intel Xeon using the following commands:

```
icc helloWorldOffload.c -o helloWorldOffload
./helloWorldOffload
```

3.2. Debug the offload using variable OFFLOAD\_REPORT and run again:

```
export OFFLOAD_REPORT=2
./helloWorldOffload
```

3.3. Change the device to offload code using the following parameter (**mic:deviceId**)

```
#pragma offload target(mic:2)
```

### 4. Profiling with vtune

open vtune:

```
amplxe-gui
```

create new project:

```
project binary: python
```

```
parameter: git/workshop-HPC-ML/HPC-hands-on/vtune/run.py
```

```
source search: git/workshop-HPC-ML/HPC-hands-on/vtune/
```



start “basic hotspot” analysis