Username: user
Password: ihpc2016
IP Address: 172.16.241.116
Compute:

HANDS ON TASKS

Task 1:
Compile hw_serial.c using gcc
gcc hw_serial.c –o hw_serial

Run the executable
./hw_serial

Task 2:
Compile hw_parallel.c using mpicc
mpicc hw_parallel.c –o hw_parallel

Run the executable with mpirun and mpiexec, vary the number of cores
mpirun -np 2 hw_parallel
mpiexec -np 2 hw_parallel

Task 3:
Create a job script to run the serial job hw_serial (Task 1) and call it as hw_serial.sh using Notepad and transfer to cluster using WinSCP

Submit the job and check your job status
qsub hw_parallel.sh
qstat OR qstat job_ID

Task 4:
Create a job script to run hw_parallel (Task 2) with 6 processors and call it as hw_parallel.sh using Notepad and transfer to cluster using WinSCP

Submit the job and check your job status
qsub hw_parallel.sh
qstat OR qstat job_ID

Task 5:
Repeat Task 4 by creating new hw_parallel_new.c file and do not compile. Instead, insert the compile command in your job script and resubmit the job (rename the job but you can use the same job script file)

Task 6:

Create a job script to run the serial job triarea.m and call it as triarea.sh (you may edit hw_serial.sh and save as new file)

Hint: use `matlab  –nodisplay  –nosplash  –nojvm  < triarea.m > triarea.out`

Submit the job and check your job status

`qsub  triarea.sh`

`qstat OR qstat job_ID`

Task 7:

Create a job script to run the serial job new.R and call it as new.sh

Hint: use `R  CMD  BATCH  --vanilla  new.R`

Submit the job and check your job status

`qsub  hw_parallel.sh`

`qstat OR qstat job_ID`

Task 8:

Run new.R using Rscript without job script

Hint: use `Rscript new.r`

Task 9:

Run the serial job hws.py without job script

Hint: `python2.7  hws.py  OR  chmod 755 hws.py`
`                              ./hws.py`

Task 10:

Create a job script to run hwp.py with 8 processors and call it as hwp.sh

Hint: `chmod 755 hwp.py`
`    mpirun –np 8 hwp.py`

Submit the job and check your job status

`qsub  hwp.sh`

`qstat OR qstat job_ID`

Task 11:

Create a job script to run pi.c with 2/4/6/8 processors and call it as pi.sh

Submit the job and check your job status

`qsub  pi.sh`

`qstat OR qstat job_ID`

Task 12:

Repeat any of the above task using interactive job submission

Task 13:

Repeat any of the above task using interactive node request