Introduction to Programming Laboratory

Lab2 - MPI

2017/7/4

Outline

- Compile and execute program on the platform
- Calculate the value of π using MPI

Outline

- Compile and execute program on the platform
- Calculate the value of π using MPI

Compile and Execute Parallel Program

MPI_COMMAND For **Intel MPI :**

	C	C++
gcc	mpicc	mpicxx
icc	mpiicc	mpiicpc

- Compile : MPI_COMMAND MPI_CODE.c [-o MPI_EXE]
- Execute (job queue): mpirun ./MPI_EXE
- Execute (directly): mpirun [-n N_PROCS] [-hostfile HOST_FILE] ./MPI_EXE
- **NOTE: DO NOT execute MPI on headnode directly!**
- **=> Submit your job through resource manager.**

Lab2-1 Compile and run MPI program

Login to server and copy lab2 directory to your home directory

- cp -r /home/ipl2017/shared/lab2 . && cd lab2

You should be able to see these files in your lab2 directory:

- HelloWorld.c
- job.sh
- README.md
- pi.c

Lab2-1 Compile and run MPI program

The program HelloWorld.c will print " Hello world", the hostname of the running node, and rank number of your processes.

[Compile]

```
mpicc HelloWorld.c -o HelloWorld
or
mpiicc HelloWorld.c -o HelloWorld
```

Lab2-1 Compile and run MPI program

[Run]

```
mpirun –np process_num ./HelloWorld
```

hostname of the node this program is running on

rank number of the process

```
[tiffanykuo@apollo31 lab2]$ mpirun -np 4 ./HelloWorld
Hello world from node apollo31, rank 1 out of 4 processes
Hello world from node apollo31, rank 2 out of 4 processes
Hello world from node apollo31, rank 3 out of 4 processes
Hello world from node apollo31, rank 0 out of 4 processes
```

Because you are running a parallel program, the execution order of the processes will not be the same !!!

Job Queues

Resource Manager: TORQUE-6.1.1.1 (Terascale Open-source Resource and QUEue Manager)

Scheduler: Maui-3.3.1

There are 2 queues in the system:

- **debug** for quick debugging purpose
- **batch** for benchmarking purpose

Job Queues: constraints

debug --- for quick debugging purpose

- Max nodes = 2
- Max total processes = 8 两台最多跑8个线程
- Max walltime = 5 minute
- Max jobs queuable at any time = 2
- Max jobs runnable at any time = 1 同时只能有1个程序运行

batch --- for benchmarking purpose

- Max nodes = 4
- Max total processes = 48
- Max walltime = 30 minutes
- Max jobs queuable at any time = 8
- Max jobs runnable at any time = 2

测时间选batch

Job Queues: priority

The scheduler will

- favor **short running** jobs (based on walltime)
- favor **less resource demanding** jobs (based on nodes, ppn)
- favor jobs which are **queued for a long time**

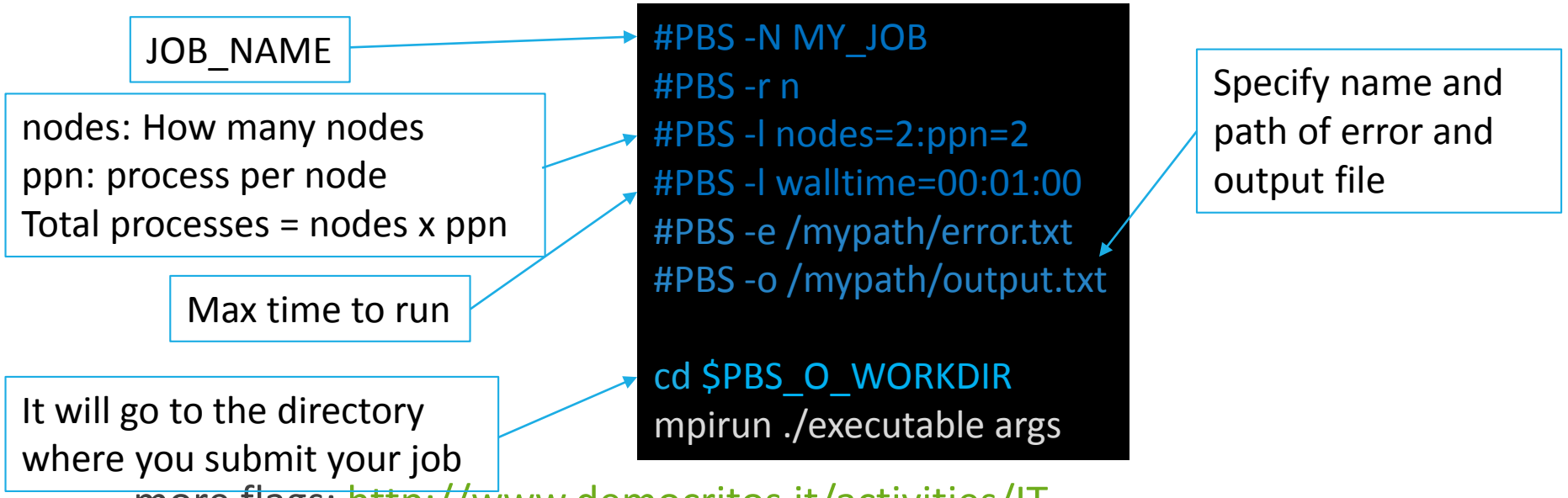
If you submit job to debug server, you will run with others' program. But if you submit job to batch server, you will not run with others' program.

Be sure to request *reasonable* amount of resources according to your own requirements.

Before Submit a Job: Job script

See job.sh in your lab2 directory:

- vim job.sh

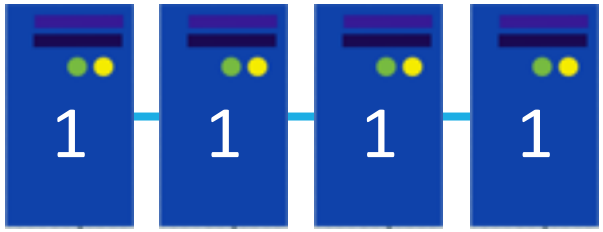


more flags: http://www.democritos.it/activities/IT-MC/documentation/newinterface/pages/runningcodes.html

Processes Layout (nodes & ppn)

For example, how to request **4** processes?
There are 3 possible ways:

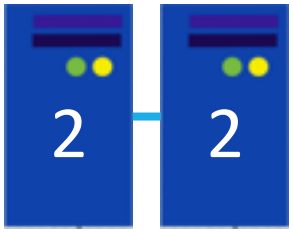
nodes=4:ppn=1



In this case, the performance my suffer.

Use this if you want to observe network overhead.

nodes=2:ppn=2



Hybrid parallelism

Use this when you have MPI + OpenMP

nodes=1:ppn=4

For Pthread & OpenMP only this configuration works.

NOTE: ppn must <= 12, because we only have 12 cores per node

Submit a Job: Job control

Submit:

- qsub JOB_SCRIPT.sh

Kill jobs:

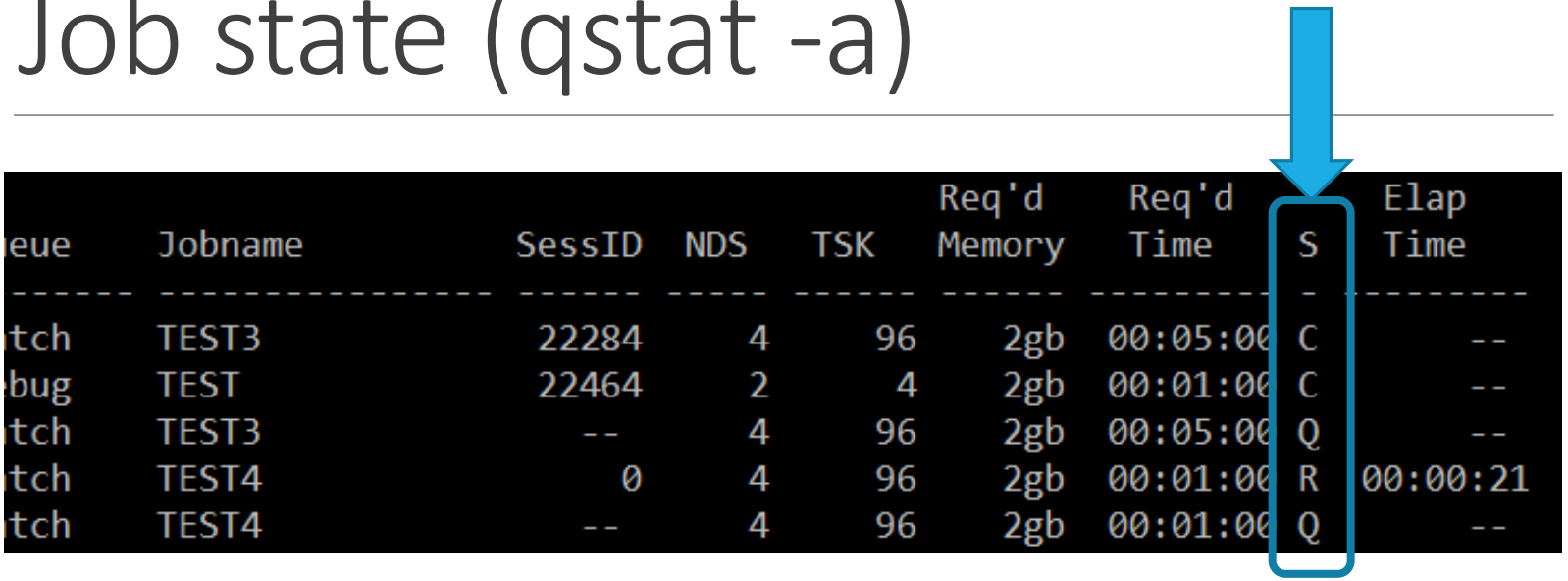
- qdel JOB_ID [JOB_ID2 [JOB_ID3…]]
- qdel all

Monitor:

- qstat -a

Again, **DO NOT TRY TO ssh DIRECTLY TO COMPUTING NODES!**

Job state (qstat -a)



eue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	S	Elap Time
tch	TEST3	22284	4	96	2gb	00:05:00	C	--
bug	TEST	22464	2	4	2gb	00:01:00	C	--
tch	TEST3	--	4	96	2gb	00:05:00	Q	--
tch	TEST4	0	4	96	2gb	00:01:00	R	00:00:21
tch	TEST4	--	4	96	2gb	00:01:00	Q	--

C: Completed

R: Running

Q: Queuing

Lab2-2 Run MPI program with job scheduler

1. **Use 1 node and 4 ppn, submit the job to debug server.**

[Edit job script] vim job.sh:

- `PBS -l nodes=1:ppn=4`
- `mpirun ./HelloWorld`

[Run] `qsub job.sh`

[Check status] `qstat -a`

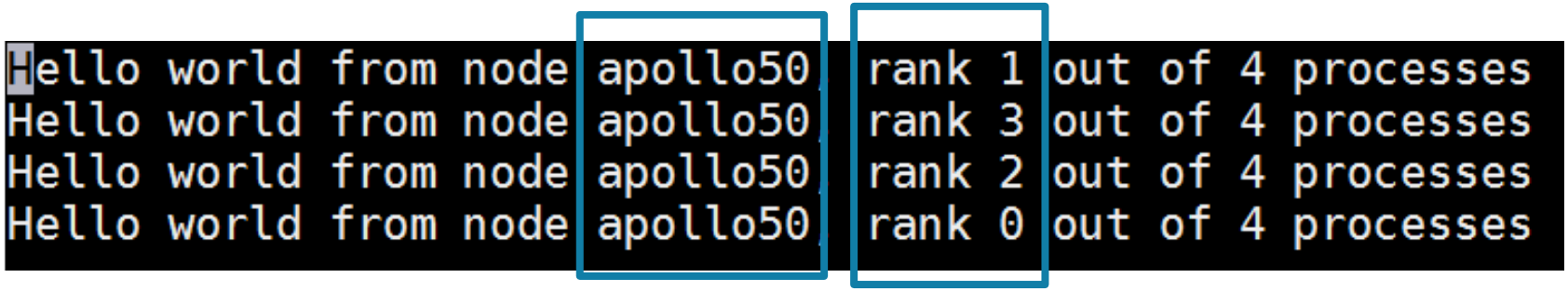
If it has finished running, you will see MY_JOB.e{jobID} and MY_JOB.o{jobID} in your directory.

Lab2-2 Run MPI program with job scheduler

You should see the result like below. Because of the scheduler, you may run on other nodes(apollo32~apollo50)

hostname of the node this program is running on

rank number of the process



Because you are running a parallel program, the execution order of the processes will not be the same !!!

Lab2-2 Run MPI program with job scheduler

2. Use 2 node and 4 ppn, submit the job to batch server, you should see the result like below. Because of the scheduler, you may run on other nodes.(apollo32~apollo50)

hostname of the node this program is running on

rank number of the process

```
Hello world from node apollo50  rank 1 out of 8 processes
Hello world from node apollo50  rank 3 out of 8 processes
Hello world from node apollo49  rank 5 out of 8 processes
Hello world from node apollo49  rank 7 out of 8 processes
Hello world from node apollo50  rank 2 out of 8 processes
Hello world from node apollo50  rank 0 out of 8 processes
Hello world from node apollo49  rank 4 out of 8 processes
Hello world from node apollo49  rank 6 out of 8 processes
```

Because you are running a parallel program, the execution order of the processes will not be the same !!!

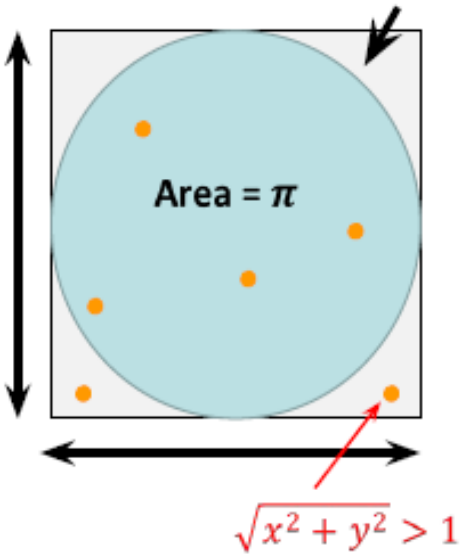
Outline

- Compile and execute program on the platform
- Calculate the value of π using MPI

Lab2-3 Calculate the value of π using MPI

Monte Carlo Methods : A class of computational algorithms that rely on repeated random sampling to compute their results.

- How to use it to compute **π** ?
  - We know: $\frac{Area\ of\ circle}{Area\ of\ square} = \frac{\pi}{4}$
  - Randomly choose points from the square
  - Giving sufficient number of samples, the fraction of the circle will be $\frac{\pi}{4}$
  - $\pi = 4 * \frac{\text{number of points in circle}}{\text{number of points in square}}$



Lab2-3 Calculate the value of π using MPI

We provide sample sequential code in lab2 directory: pi.c

[Compile]

```
gcc pi.c -o pi
```

[Edit job script] vim job.sh:

```
#PBS -q debug
#PBS -l nodes=1:ppn=1
./pi 500000
```

→ number of total points

[Run]

```
qsub job.sh
```

Lab2-3 Calculate the value of π using MPI

Modify the sample sequential code to MPI code!

Hint:

- Each process will be assigned part of the points to calculate.

- Each process send their partial result to one process to print final result.

Lab2-4 Measure and compare the time of sequential and parallel program

Use time command to measure sequential pi program and your mpi program.

- Use batch queue to run your job.

- Try different number of points to see the result