Compiling and Job Submission

Turning your source code into an executable code, then running it in batch mode.

April 23, 2002

C compiler

- -g option for debugging
- -X option to hardcode # of Pes
- -l to link with a library
- -O[0-3] for optimization

Fortran compiler

- -g option for debugging
- -X option to hardcode # of Pes
- -l to link with a library
- -O[0-3] for optimization

MPI library

- Link with –lmpi
- This is automatically done for your on jaromir, but you must remember to link if you are using mpi on most other systems

Running your program

- To run your program in parallel you need to issue the mpprun command
- Indicate the number of processors with –nX
- Example mpprun –n4 a.out

Interactive

- Interactive Mode
  - Used for compiling and debugging
  - Should not be used for production runs
  - Do not run multiple interactive jobs at the same time
  - Limit of 10 CPU minutes in interactive mode

Batch

- Batch Mode
  - Create a script
  - Submit to the queueing system
  - Available 24 hours
  - Should be used for production runs

Sample batch file

```
#QSUB –l mpp_t=3600
#QSUB –l mpp_p=2
#QSUB –o t3e.output –eo
set echo
ja
cd $TMP
cp ~/prog prog
cp ~/data data
mpprun –n2 prog > results
far store results results
ja -csthlMe
```

Submit the job

- While logged into jaromir, use the qsub command

qsub jobfile

Monitor the job

- The qstat command displays the status of the job

qstat –a

Will show information about your job

Qstat output

88059.jaromir.psc.edu test.job username qm_12h_128@jaromir 34455 24 690 7113 R05

Delete a job

- The qdel command will delete a job, use –k if the job is running
  - qdel jobid
  - qdel –k jobid

Output and Error files

- Upon completion of your batch job, you should receive an output and an error file(unless you combined them with the –eo option)

Typical Errors

- The current csh(23395) has received signal 26 (cpu limit exceeded)
  - Ask for more time in your batch job
- Warning: no access to tty; thus no job control in this shell
  - Simply indicating that it is a batch request, ignore this message

Exercises

- Login to jaromir and cd to your staging directory(you may need to create this)
  - mkdir /tmp/username
  - cd /tmp/username

Exercises Cont.

- Copy exer.f from /tmp/training to your staging directory
  - cp /tmp/training/exer.f .
- Compile
  - f90 exer.f –o exer
- Run interactively, enter in 3 integers
  - mpprun –n4 exer

Fortran Sample Code

- exer.f
  - Compile, link with the mpi library.
  - Run on 2 – 8 processors.
  - Enter 3 integers, the first being the size of the problem, the second being the number of iterations and the third being the number of processors used.
  - Outputs the time and flops.

Exercises Cont.

- Copy shuf.c from /tmp/training to your staging directory
  - cp /tmp/training/shuf.c .
- Compile
  - cc shuf.c –o shuf
- Run interactively on 4 processors
  - mpprun –n4 shuf

C Sample Code

- shuf.c
  - Compile, link with the mpi library.
  - Run on 2-8 processors.
  - Passes numbers via mpi.

Exercises – Job Submission

- Create a job that will
  - Request 50 seconds of execution time and 2 Pes
  - Change directory to $TMP
  - Copy the shuf executable from your /tmp/username directory to $TMP
  - Run shuf
  - Redirect the output to a file called output.shuf
  - Copy output.shuf to /tmp/username

Exercises – Job Submission 2

- Submit the job
- Check the status
- Check the error and output files
- Store output.shuf to far