MPI – Message Passing Interface

MPI is a way for a programme to communicate across multiple nodes. It can also be used on a single node although many packages will already be able to make use of multiple cores without using MPI (e.g. see the help pages on using R and Matlab). The HPC nodes have openmpi installed as a module, at the time of writing the openmpi version is 4.1.1, you can also install your own version.

Using pbsdsh

The Torque package comes with a programme that allows you to launch multiple copies of a command or script. This will only run as part of Torque/PBS job and it transparently uses the environment variable $PBS_NODEFILE to know what nodes to run on. Here is a copy of the pbsdsh help:

```
Usage: pbsdsh [-c copies][-o][-s][-u][-v] program [args]...]
       pbsdsh [-n nodenumber][-o][-s][-u][-v] program [args]...
       pbsdsh [-h hostname][-o][-v] program [args]...
Where -c copies =  run  copy of "args" on the first "copies" nodes,
      -n nodenumber = run a copy of "args" on the "nodenumber"-th node,
      -o = capture stdout of processes,
      -s = forces synchronous execution,
      -u = run on unique hostnames,
      -h = run on this specific hostname,

      -v = forces verbose output.
```

Note the option -u which allows you to run a copy of a particular command once for each unique node. In my tests standard output was captured and relayed to the primary host and either displayed on the screen when running an interactive job or in the .o output file.

Here is a copy of a simple PBS script that will allow you to run multiple copies of your own programme (here called myscript.sh). By default, it will run myscript.sh on all the cores you have been allocated.

```
#!/bin/sh

#PBS -l cput=55:30:00, walltime=100:00:00, nodes=2:ppn=2:centos7
#PBS –m abe
#PBS –M myemail@glasgow.ac.uk

/usr/local/bin/pbsdsh myscript.sh
```

To submit it to the cluster, assuming you have called the PBS script “parrallel-simple.sh”, just run

```
qsub parallel-simple.sh
```

This will submit your job to the cluster and will launch 4 copies of myscript.sh – 2 copies on each of the two nodes you have been assigned.

It is then up to you to write the code *if* you require the processes to communicate between themselves.

Using Open MPI

The package openmpi is available on the compute nodes as a module. To run a job using openmpi you use the commands mpirun or mpiexec. Openmpi also includes parallel compilers which you can use to compile your code such that it is able to run on multiple nodes. You can also use this to run your job on multiple cores on the same node.

Let us create a simple c programme that runs a number of processes on different nodes and prints out information on where it is running. Save this to a file called hello-mpi.c.

```
#include "stdio.h"
#include <stdlib.h>
#include </usr/include/openmpi-x86_64/mpi.h>
int main(int argc, char *argv[])
{
 int threadid,totalthreads;
 char *host_name;
  /* Start MPI processes on each node */
  MPI_Init(&argc,&argv);
  /* request a thread id from the MPI master process, which has threadid ==
0 */
  MPI_Comm_rank(MPI_COMM_WORLD, &threadid);
  /* Get the number of processes launched by MPI  */
  MPI_Comm_size(MPI_COMM_WORLD, &totalthreads);
  /* For each process, allocate space for the node name */
  host_name    = (char *)calloc(80,sizeof(char));
  /* populate “host_name” with the name of the node  */
  gethostname(host_name,80);
  /* Print a statement to standard output */
  printf("hello: from process = %i on machine=%s, of NCPU=%i processes\n",
         threadid, host_name, totalthreads);
  MPI_Finalize();

return(0); }
```

Start an interactive session on one of the centos7 nodes with `qsub -I -l nodes=1:centos7` then compile the file you just created using `mpicc -c hello-mpi.c -o hello-mpi.out` (very basic command, check manuals for further optimization options). See the captured session in the next box.

```
-bash-3.2$ qsub -I -l nodes=1:centos7
qsub: waiting for job 274547.headnode03.cent.gla.ac.uk to start
qsub: job 274547.headnode03.cent.gla.ac.uk ready
Prologue Args:
Job ID: 274547.headnode03.cent.gla.ac.uk
User ID: mjmtest2
Group ID: mjmtest2
MAchine: node051.hpc.gla.ac.uk
```

```
-bash-4.1$ cd mpi-testing/
-bash-4.1$ module load openmpi
-bash-4.1$ mpicc -c hello-mpi.c -o hello-mpi.out
-bash-4.1$ ls
hello-mpi.c  hello-mpi.out
-bash-4.1$ exit
logout

qsub: job 274547.headnode03.cent.gla.ac.uk completed
```

Type exit to get out of your Interactive qsub job and this will return you to the headnode.
Now you can use this executable, hello-mpi.out, in a normal qsub script running it with mpirun

create a qsub script – openmpi-test.sh – with the following contents:

```
#!/bin/bash
#PBS -l nodes=4:ppn=4:centos7
module load openmpi

mpirun -np 16 -machinefile $PBS_NODEFILE /export/home/mjmtest2/mpi-
testing/hello-mpi.out
```

This example requests 4 cores on each of 4 nodes for a total of 16 cores. Then we tell the mpirun command to use 16 cores with '-np 16'. The environment variable $PBS_NODEFILE will pass the mpirun command a reference to a file containing the nodes that your job has been allocated.

You can then submit this script with the qsub command.

```
         -bash-3.2$ qsub openmpi-test.sh
         274550.headnode03.cent.gla.ac.uk
```

Note that you get back a job reference – 274550.headnode03.cent.gla.ac.uk .
The job will be submitted and once it completes (the example job should only take seconds assuming resource is available) you can see the results of your command in the output file - openmpi-test.sh.o274550 – note the number at the end refers to the job reference.

If you look at the contents of the output file you will see something like this

```
Prologue Args:
Job ID: 274550.headnode03.cent.gla.ac.uk
User ID: mjmtest2
Group ID: mjmtest2
MAchine: node072.hpc.gla.ac.uk
hello: from process = 2 on machine=node072.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 3 on machine=node072.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 0 on machine=node072.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 1 on machine=node072.hpc.gla.ac.uk, of NCPU=16
processes
```

```
hello: from process = 8 on machine=node070.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 9 on machine=node070.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 10 on machine=node070.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 11 on machine=node070.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 7 on machine=node071.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 4 on machine=node071.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 5 on machine=node071.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 6 on machine=node071.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 13 on machine=node069.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 14 on machine=node069.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 12 on machine=node069.hpc.gla.ac.uk, of NCPU=16
processes
hello: from process = 15 on machine=node069.hpc.gla.ac.uk, of NCPU=16
processes
Epilogue Args:
Job ID: 274550.headnode03.cent.gla.ac.uk
User ID: mjmtest2
Group ID: mjmtest2
Job Name: openmpi-test.sh
Session ID: 3161
Resource List:
cput=01:00:00,neednodes=4:ppn=4:centos6,nodes=4:ppn=4:centos6,walltime=01:0
0:00
Resources Used: cput=00:00:01,mem=2420kb,vmem=58896kb,walltime=00:00:03
Queue Name: bioinf-stud2
Account String:
No user epilogue file found at /export/home/mjmtest2/epilogue
Process's killed on node072.hpc.gla.ac.uk
tmp directory removed on node072.hpc.gla.ac.uk
Process's killed on node071.hpc.gla.ac.uk
tmp directory removed on node071.hpc.gla.ac.uk
Process's killed on node070.hpc.gla.ac.uk
tmp directory removed on node070.hpc.gla.ac.uk
Process's killed on node069.hpc.gla.ac.uk
tmp directory removed on node069.hpc.gla.ac.uk
```

The lines that start hello: are the output from openmpi-test.sh (the rest is standard output from the qsub command itself) you can see that your job ran 16 processes – 4 on each of 4 different nodes.

The links section at the end of the document will point you at some websites where you can do some further reading.

Warning

Please ensure that you do not try to use more cores with openmpi than you request with qsub. This can unbalance the allocation of processors to nodes and take resource away from other users’ jobs, if this happens, we may delete your job to prevent the other jobs being adversely affected.

Links

The Open MPI website.

The Documentation for the version on the system, 4.1.