INSTRUCTION FOR RUNNING A MPI JOB AT BLUEGIRT

Ashwini Lahane and Shujia Zhou

This document is applied to blade1 to blade32, cell1 to cell10 as of March 10, 2009, and combination of blade1-32 and cell1-10. For exercises and homeworks, use blade1 to blade32 only.

Setting up MPD ring

This document describes the steps involved in creating a MPD ring. MPD ring essentially is a group of hosts running the MP daemon (MPD) with knowledge about the other hosts present in the ring.

Suppose we have a group of hosts (say) cell4, cell5 and cell7. Let us consider cell7 to be the master host and the other 2 to be slave hosts. Each host in the ring must have a mpd.hosts file configured containing names of all hosts participating in the ring including itself. In our case the mpd.hosts file will be as follows:

```
cell7
cell5
cell4
```

Now we need to start mpd on each of the hosts by executing just one command from the master host. It will automatically pick up each of the hosts from the mpd.hosts file and start a MP daemon on each of them, thus forming an MPD ring. (PVFS file system makes mpd.hosts available on all the hosts.) The command to be executed on cell7 is:

```
$ mpdboot -n 3 -f mpd.hosts
```

where,

n - Specifies total number of MP daemons to be started,
f - Specifies the file containing the list of host names.

Check whether MPD has started on each host with ps –ef|grep mpd:
On cell7,

```
alahane1  9969     1  0 12:14 ?        00:00:00 python2.5
/bluegrit/data/mpi/bin/mpd.py --ncpus=1 -e -d
```

It takes 1 cpu by default.

On cell5,

```
alahane1 30666     1  0 12:17 ?        00:00:00 python2.5
/bluegrit/data/mpi/bin/mpd.py -h cell007 -p 40792 --ncpus=1 -e -d
```

Here,

-h – specifies the master host that invoked the MPD on current host
-p – specifies the port number on the master host.

On cell4,

```
alahane1 16116     1  0 12:14 ?        00:00:00 python2.5
/bluegrit/data/mpi/bin/mpd.py -h cell007 -p 40792 --ncpus=1 -e -d
```

Entries on cell4 are similar to that on cell5.

Thus we have established an MPD ring on hosts cell7, cell5 and cell4 with 1 CPU from each host.

Execution of a parallel code using the above described MPD ring of hosts

We will be using the parallel code written for the particle-system simulation (particle.c) to demonstrate the distribution of computation over the CPUs available in the MPD ring. For convenience the value of end_Of_Time variable has been hard-coded to 0.002 in the program. Also we add a line to print the current hostname.

```
$ mpicc -o particle particle.c
$ mpirun -np 12 ./particle > out.file
```

The contents of the out.file as as follows:

```
cell007
cell004
cell005
My id = 2
My id = 0
 numproc=12
My id = 1
cell005
cell007
My id = 3
cell004
My id = 5
cell004
My id = 4
My id = 7
cell004
My id = 10
cell007
cell007
cell005
My id = 6
My id = 8
My id = 9
cell005
My id = 11
Total Seconds: 0.167180
Temperature is: 1.000065
```

cell0007 etc are obtained with system("$hostname") in particle.c

While executing the code we had requested for 12 CPUs. There are 4 entries of each of the 3 hosts in the ring. This shows that the computation was equally distributed between the 3 hosts.

To explicitly request the number of mpi processes in each host, add the number of mpi processes into the mpd.hosts file. For example,

blade01:2
blade03:2
blade05:2

Where each blade have two mpi processes. To setting up the MPD ring, use
Mpdboot –n 3 –f mpd.hosts –-ncups=2

Note that –ncpups=2 is needed for ensuring 2 mpi processes in the host launching the MPD ring.

We can try the same program with different number of CPUs each time. After completing mpi jobs, use mpdallexit to tear down the mpd ring.