UBDA Platform

Example for C with OpenMPI

User Guide

Version 1.0

16 July 2018

Revision History

Version	Date	Prepared By	Summary of Changes
1.0	Jul 16, 2018		Initial release

Table of Contents

1. Introduction

This document is shown a C example by using OpenMPI method running on the UBDA platform.

Note: User should first register a user account through UBDA website at: https://www.polyu.edu.hk/pfs/index.php/177729 to access the UBDA Platform

2. Perform the test

2.1 Login to ubdaplatform.polyu.edu.hk via SSH

2.2 Create the testing directory

```
$ mkdir -p $HOME/mpi_pi_c
$ cd $HOME/mpi_pi_c
```

2.3 Prepare the C program (Filename: mpi_pi.c)

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
int main( int argc, char *argv[] )
{
    int n, myid, numprocs, i;
    double PI25DT = 3.14159265358979323846264338327950288419716939937510;
    double mypi, pi, h, sum, x;
    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    while (1) {
        if (myid == 0) {
            printf("Enter the number of intervals: (0 quits) ");
            scanf("%d",&n);
        }
        MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
        if (n == 0)
            break;
        else {
            h   = 1.0 / (double) n;
            sum = 0.0;
            for (i = myid + 1; i <= n; i += numprocs) {
                x = h * ((double)i - 0.5);
                sum += (4.0 / (1.0 + x*x));
            }
            mypi = h * sum;
            MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
                       MPI_COMM_WORLD);
            if (myid == 0)
                printf("pi is approximately %.16f, Error
is %.16f\n",
                       pi, fabs(pi - PI25DT));
        }
    }
    MPI_Finalize();
    return 0;
}
```

2.4 Load the MPI user environment. You can select OpenMPI with gcc.

For select OpenMPI, use the following:

```
$ module available
----------------- /ubda/apps/modules/modules-
4.1.2/modulefiles -----------------
anaconda3-5.2.0    cuda-9.1                  perl-5.26.1
ansys-17.2-fluent  fsl-5.0.11                python-2.7.15
atlas-3.10.3       gcc-5.5.0                 python-3.6.6
autodock-4.2.6     ips_xe_2018_u3            quantum-
espresso-6.2.1
cmake-3.12.1       mpich-3.2.1-gcc-5.5.0     R-3.5.0
cuda-9.0           openmpi-3.0.1-gcc-5.5.0   singularity-
2.6.0

$ module load openmpi-3.0.1-gcc-5.5.0
$ module list
Currently Loaded Modulefiles:
     1) openmpi-3.0.1-gcc-5.5.0
$ which mpicc
/ubda/apps/openmpi/openmpi-3.0.1-gcc-5.5.0/bin/mpicc
```

2.5 Build the executable binary by using the selected MPI

```
$ mpicc -o mpi_pi mpi_pi.c
$ file mpi_pi
mpi_pi: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.18, not
stripped
```

2.6 Prepare the input data file (Filename: input.dat)

```
19
299
3999
49999
599999
6999999
79999999
899999999
0
```

2.7 Prepare the job script file.

For select OpenMPI, use the following (Filename: mpi_pi_c_openmpi.pbs)

```
#!/bin/sh
#PBS -N mpi_pi_c_test
#PBS -l nodes=1:ppn=8
#PBS -l walltime=12:00:00
#PBS -q q2s01
#PBS -V
#PBS -S /bin/bash

module load openmpi-3.0.1-gcc-5.5.0

EXEC=mpi_pi
############################################################################
NP=`cat $PBS_NODEFILE | wc -l`
NN=`cat $PBS_NODEFILE | sort | uniq | tee /tmp/nodes.$$ | wc -l`
cat $PBS_NODEFILE > /tmp/nodefile.$$

echo "process will start at : "
date
echo "++++++++++++++++++++++++++++++++++++++++"
cd $PBS_O_WORKDIR

mpirun -n $NP --mca blt self,openib $EXEC < input.dat >
result.out


echo "++++++++++++++++++++++++++++++++++++++++"
echo "process end at : "
date
rm -f /tmp/nodefile.$$
rm -f /tmp/nodes.$$
module unload openmpi-3.0.1-gcc-5.5.0
```

Please remind to change the values for your application.

```
#PBS -N mpi_pi_c_test {your job name}
#PBS -l nodes=1:ppn=8 {your requested resource;nodes and
processors per node}
#PBS -q q2s01 {the job queue}
EXEC=mpi_pi {your execute file name}
```

Example files can be found at:

/ubda/apps/examples/C/mpi-pi/

input.dat
mpi_pi.c
mpi_pi_c_openmpi.pbs

3 Job submission

3.1 Submit the script (***mpi_pi_c_openmpi.pbs***) to job queue.
A job ID number will be returned.

```
$ qsub mpi_pi_c_openmpi.pbs
742.ubda-mgt01
```

3.2 Enquiry the submitted job status.

```
$ qstat -na

ubda-mgt01:
                                                                  Req'd       Req'd       Elap
Job ID                  Username    Queue    Jobname          SessID  NDS   TSK   Memory      Time    S   Time
----------------------- ----------- -------- ---------------- ------ ----- ------ --------- --------- - ---------
742.ubda-mgt01          ubdademo9   q2s01    mpi_pi_c_test       --      1      8       --   12:00:00 Q       --
```

Job status field name	Explanation	Example
JOB ID	Unique Job id.	742.ubda-mgt01
Job name	Name for the job allocation	mpi_pi_c_test
Queue name	Name of the job queue that the job has assigned.	q2s01
Username	Your NetID	ubdademo9
S	Job current status. Q = queued R = Job is executing C = Job was completed	Queued
ELAP TIME	Time for the job executed	
Req'd Time	The maximum execution time	12:00:00 (1 day)
NDS	Total number of nodes assigned	1
TSK	Total number of cores assigned	8

4 How to check the result

4.1 Check for any error messages for your job with the JobID

```
$ more mpi_pi_c_test.e742
Host key verification failed.(know error message can be ignored)

$ more mpi_pi_c_test.o742
process will start at :
Tue Jun 12 17:33:56 HKT 2018
++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++
process end at :
Tue Jun 12 17:33:57 HKT 2018
```

4.2 Check for result

```
$ wc -l result.out
8 result.out

$ cat result.out
Enter the number of intervals: (0 quits) pi is approximately
3.1418234938074781, Error is 0.0002308402176849
Enter the number of intervals: (0 quits) pi is approximately
3.1415935857195603, Error is 0.0000009321297672
Enter the number of intervals: (0 quits) pi is approximately
3.1415926588007332, Error is 0.0000000052109401
Enter the number of intervals: (0 quits) pi is approximately
3.1415926536231282, Error is 0.0000000000333351
Enter the number of intervals: (0 quits) pi is approximately
3.1415926535900298, Error is 0.0000000000002367
Enter the number of intervals: (0 quits) pi is approximately
3.1415926535897869, Error is 0.0000000000000062
Enter the number of intervals: (0 quits) pi is approximately
3.1415926535897092, Error is 0.0000000000000839
Enter the number of intervals: (0 quits) pi is approximately
3.1415926535897700, Error is 0.0000000000000231
```

5 Useful Reference

- Manpage for OpenMPI - mpicc
  URL: https://www.open-mpi.org/doc/v3.0/man1/mpicc.1.php

- Manpage for OpenMPI - mpirun
  URL: https://www.open-mpi.org/doc/v3.0/man1/mpirun.1.php

- Command reference for qstat
  URL: http://docs.adaptivecomputing.com/torque/6-0-0/help.htm#topics/torque/commands/qstat.htm?Highlight=qstat