



UBDA Platform

Example for Python

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	4
2. Perform the test.....	5
3. Job submission	8
4. How to check the result	9
5. Useful Reference.....	11

1. Introduction

This document is shown a Python example 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/python
$ cd $HOME/python
```

2.3 Prepare the C program (Filename: pi_mp.py)

```
import random
import multiprocessing
from multiprocessing import Pool

#calculate the number of points in the unit circle
#out of n points
def monte_carlo_pi_part(n):

    count = 0
    for i in range(n):
        x=random.random()
        y=random.random()

        # if it is within the unit circle
        if x*x + y*y <= 1:
            count=count+1

    #return
    return count

if __name__=='__main__':

    np = multiprocessing.cpu_count()
    print 'You have {0:1d} CPUs'.format(np)

    # Number of points to use for the Pi estimation
    n = 10000000

    # iterable with a list of points to generate in each worker
    # each worker process gets n/np number of points to
    # calculate Pi from

    part_count=[n/np for i in range(np)]

    #Create the worker pool
    # http://docs.python.org/library/
    pool = Pool(processes=np)
    # parallel map
```

```
count=pool.map(monte_carlo_pi_part, part_count)
print "Estimated value of Pi:: ", sum(count)/(n*1.0)*4
```

2.4 Prepare the job script file.

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

```
#!/bin/sh
#PBS -N pytest
#PBS -l nodes=3:ppn=10
#PBS -l walltime=12:00:00
#PBS -q q2s01
#PBS -V
#PBS -S /bin/bash
module load python-2.7.15
module load openmpi-3.0.1-gcc-5.5.0
#####
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 python2 pi_mp.py > result.out
echo "+++++"
echo "process will sleep 5 minutes"
#sleep 300
echo "process end at : "
date
rm -f /tmp/nodefile.$$
rm -f /tmp/nodes.$$
module unload openmpi-3.0.1-gcc-5.5.0
module unload python-2.7.15
```

Please remind to change the values for your application.

```
#PBS -N pytest {your job name}
#PBS -l nodes=3:ppn=10{your requested resource;nodes and
processors per node}
#PBS -q q2s01 {the job queue}
py_mp.py {your python file name}
```

Example files can be found at:

/ubda/apps/examples/python/

pi_mp.py

python.pbs

3 Job submission

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

```
$ qsub python.pbs
2182.ubda-mgt01
```

- 3.2 Enquiry the submitted job status.

```
$ qstat -na

ubda-mgt01:
Job ID              Username      Queue      Jobname      SessID  NDS   TSK      Req'd      Req'd      Elap
-----            -
2182.ubda-mgt01    ubdademo9    q2s01     pytest       0       3     30       --      12:00:00 R   00:00:00
ubda-d050/0-9+ubda-d049/0-9+ubda-d048/0-9
```

Job status field name	Explanation	Example
JOB ID	Unique Job id.	2182.ubda-mgt01
Job name	Name for the job allocation	pytest
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	Running
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	3
TSK	Total number of cores assigned	30

Estimated value of Pi::	3.1407612
Estimated value of Pi::	3.1416532
Estimated value of Pi::	3.1424864
Estimated value of Pi::	3.1417624
Estimated value of Pi::	3.1416748
Estimated value of Pi::	3.1417908
Estimated value of Pi::	3.1413128
Estimated value of Pi::	3.1415788
Estimated value of Pi::	3.1418268
Estimated value of Pi::	3.1418464
Estimated value of Pi::	3.1420204
Estimated value of Pi::	3.1407996
Estimated value of Pi::	3.1417516
Estimated value of Pi::	3.1420804
Estimated value of Pi::	3.141536
Estimated value of Pi::	3.1416024
Estimated value of Pi::	3.1426356
Estimated value of Pi::	3.1411584
Estimated value of Pi::	3.1419776
Estimated value of Pi::	3.1419948
Estimated value of Pi::	3.1421544
Estimated value of Pi::	3.1413532
Estimated value of Pi::	3.1417556
Estimated value of Pi::	3.141186
Estimated value of Pi::	3.1404652
Estimated value of Pi::	3.1418144
Estimated value of Pi::	3.14111
Estimated value of Pi::	3.1425032
Estimated value of Pi::	3.1418232

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>