CS 4593/5463 Cloud and Big Data

Assignment 1: Benchmarking MPI Cluster in the Cloud
<u>Due Midnight Monday, Feb 13, 2017</u>

Pre-requisite Steps:

- *Login to OpenStack dashboard using the default password, and change the password.*
- *Setup Networking in the cloud*
- *Create Key-pair, and edit security rules to allow SSH, and ICMP.*
- *Create a VM on the openstack cloud*
- *Allocate floating IP, and associate it with your VM*
- *Install open-mpi on your VM*

Q. 1. This exercise is to measure the performance of a ray tracing engine, executing parallel tasks in a MPI cluster. Report the "Ray Tracing Time" for various cluster sizes (1, 2, 3, 4 VMs). For each cluster size, use two default slots per node, and run processes to fill up all the default slots. Compare the performance for two different scheduling policies (-byslot and –bynode) for each cluster size. Plot a graph to compare the performance, and conclude which is faster (and give an explanation why).

Steps to install Ray Tracing Program (Tachyon)

```
 wget http://jedi.ks.uiuc.edu/~johns/raytracer/files/0.99b2/tachyon-0.99b2.tar.gz
tar -zxf tachyon-0.99b2.tar.gz
cd tachyon/unix
sudo apt-get install make
make linux-mpi
```

Steps to run Ray Tracing Program (Tachyon)

```
cd ~/tachyon/compile/linux-mpi
mpirun –np 2 --hostfile ~/mpi_hosts ./tachyon ../../scenes/teapot.dat
```

Reference: Tachyon
http://jedi.ks.uiuc.edu/~johns/raytracer/

Q. 2. Evaluate the performance of two HPCC (High Performance Computing Challenge) benchmarks: HPL(the Linpack benchmark which measures the floating point rate of execution for solving a system of linear equations) and PTRANS (Parallel matrix transpose). Perform an analysis similar to exercise 1.

To install HPCC benchmark, please download "installer.sh" file, change execution permission and run the installer.sh script using the following commands..

wget http://www.cs.utsa.edu/~plama/CS5463/installer.sh

chmod +x installer.sh
./installer.sh

After installation, go inside the hpcc-1.4.3 directory, which contains the executable "**hpcc**", and input file "**hpccinf.txt**". The input file contains the input parameters for the benchmarks. Make sure that the input parameters P and Q for the HPL benchmark is set such that P * Q <= -np option in the mpirun command, and Q is slightly larger than P. Furthermore, use the following tool to find a suitable value for other input parameters such as N (problem size), and NB (block size).

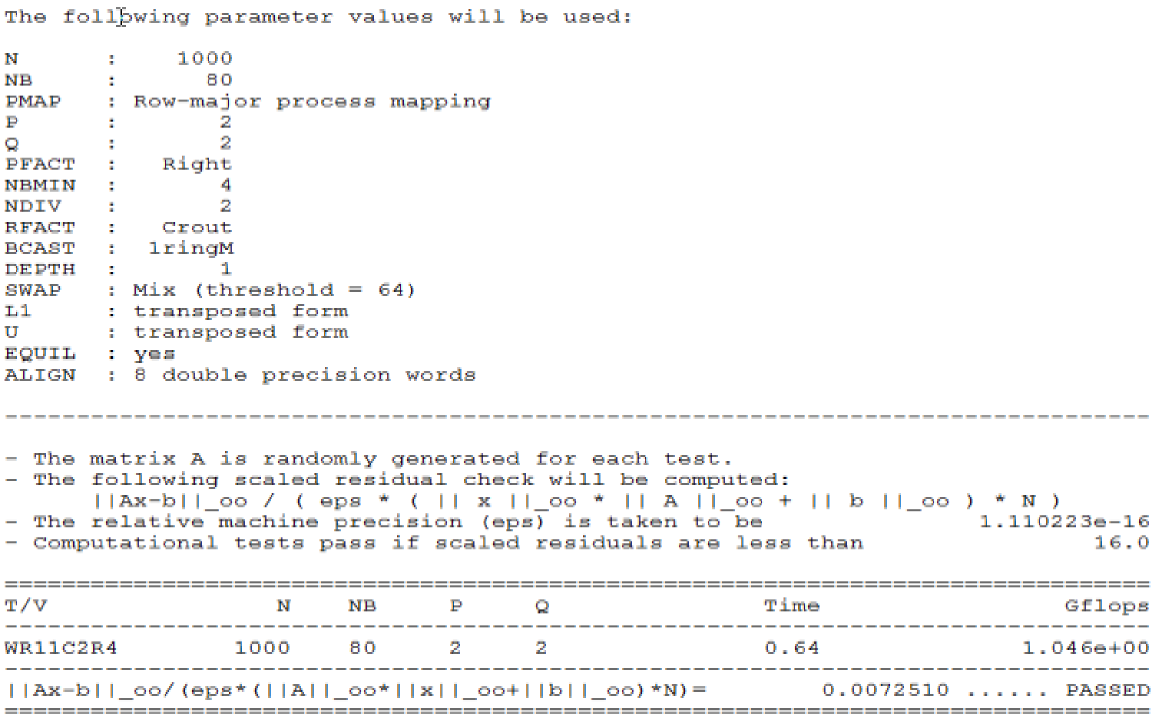
http://hpl-calculator.sourceforge.net/

When you execute the benchmark "hpcc" using mpirun command, it will generate an output file named "**hpccoutf.txt**", which contains the benchmark results. The performance of HPL and PTRANS benchmarks are reported in terms of Gflops, and GB/s respectively. In case of PTRANS, you may see multiple values of performance results. You can take an average of the performance values related to the WALL clock time.

Steps to run hpcc benchmark:
cd hpcc-1.4.3

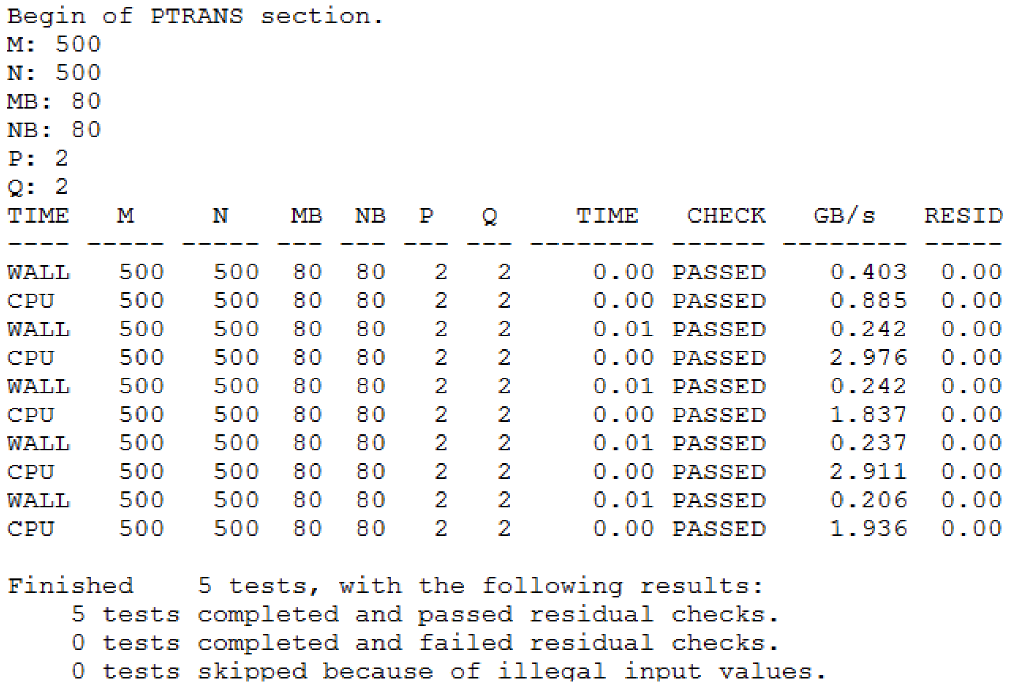
```
mpirun -np 2 --hostfile ~/mpi_hosts ./hpcc
```

HPL Results Example (inside **hpccoutf.txt**)



```
The following parameter values will be used:

N      :    1000
NB     :      80
PMAP   : Row-major process mapping
P      :       2
Q      :       2
PFACT  :   Right
NBMIN  :       4
NDIV   :       2
RFACT  :   Crout
BCAST  :  1ringM
DEPTH  :       1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words

--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be               1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR11C2R4        1000    80     2     2               0.64              1.046e+00
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0072510 ...... PASSED
================================================================================
```

PTRANS Results Example (inside **hpccoutf.txt**)



```
Begin of PTRANS section.
M: 500
N: 500
MB: 80
NB: 80
P: 2
Q: 2
TIME   M     N    MB  NB  P   Q     TIME   CHECK    GB/s   RESID
---- ----- ----- --- --- --- --- -------- ------ -------- -----
WALL   500   500  80  80   2   2     0.00 PASSED    0.403  0.00
CPU    500   500  80  80   2   2     0.00 PASSED    0.885  0.00
WALL   500   500  80  80   2   2     0.01 PASSED    0.242  0.00
CPU    500   500  80  80   2   2     0.00 PASSED    2.976  0.00
WALL   500   500  80  80   2   2     0.01 PASSED    0.242  0.00
CPU    500   500  80  80   2   2     0.00 PASSED    1.837  0.00
WALL   500   500  80  80   2   2     0.01 PASSED    0.237  0.00
CPU    500   500  80  80   2   2     0.00 PASSED    2.911  0.00
WALL   500   500  80  80   2   2     0.01 PASSED    0.206  0.00
CPU    500   500  80  80   2   2     0.00 PASSED    1.936  0.00

Finished    5 tests, with the following results:
    5 tests completed and passed residual checks.
    0 tests completed and failed residual checks.
    0 tests skipped because of illegal input values.
```

More information about HPCC benchmark, and LINPACK can be found here:
http://icl.cs.utk.edu/hpcc/index.html
http://icl.cs.utk.edu/news_pub/submissions/sitka-ch-02.pdf
https://www.top500.org/project/linpack/

Q. 3. [Extra Credits for CS4593 / Mandatory for CS5463]

Write a python program to automate the experiments, and graph (results) plotting for Q. 1, and Q. 2. Note that for Q. 2. the input parameters P, Q, N, and NB needs to be tuned manually for each cluster size, and then the optimal values can be hard-coded in your program. Your program file should be named “assign1.py”. Here is how your program should be executed:

```
python assign1.py -benchmark=[tachyon|hpcc] -sched=[byslot|bynode]
```

Your program should run the given benchmark for various cluster sizes (1,2,3, and 4 VMs) using the given scheduling policy, and plot a graph showing the performance results. The X axis in the graph will be cluster size, and the Y axis will be the corresponding performance value. For graph plotting, you can use the matplotlib library. http://matplotlib.org/index.html

Grading:

This is a 100 pt homework.
For CS4593, Q. 1 (50 pts) and Q. 2 (50 pts).
For CS5463, Q. 1 (40 pts), Q. 2 (40 pts), and Q. 3 (20 pts).

Submission Guidelines:

You must submit your work using Blackboard Learn and respect the following rules:

- Only 1 submission per group
- All assignments must be submitted as either a zip or tar archive file.
- Assignment folder (assign1) must include the following :
    - source code (if any)
    - the private key that was used to login to your VMs. (Make sure that your VMs are still up and running even after you finish the assignment).
    - a PDF report containing the following:
        - List the group member names, and describe the contribution of each member.
        - Graphs showing the results obtained for Q. 1 and Q. 2, and a brief explanation of the results.