

SINGAPORE-MIT ALLIANCE
COMPUTATIONAL ENGINEERING PROGRAMME
CME5232: Cluster and Grid Computing Technologies for Scientific Computing
COMPUTATIONAL LAB No.1
4th July, 2009

INTRODUCTION TO HIGH PERFORMANCE COMPUTING RESOURCES

Objectives

To be familiar with the SMA Hydra Linux Cluster as the main High Performance Computing (HPC) platform for parallel computing tasks.

Background Knowledge

C/C++ programming languages, elementary knowledge of Unix/Linux commands.

- Question1.
- Question2.
- Question3.

Question1: Getting to know the SMA Hydra Linux Cluster

SMA Hydra Linux cluster (hydra.sma.nus.edu.sg) consists of 65 nodes. (Note: Some nodes are spoiled).

- How to access SMA Hydra Linux Cluster?

- You are provided your own userID and passwords.
- You need to install the **SSH client** software to personal computer to transfer your program from your own computer to SMA Hydra Linux Cluster.

The software available at http://www.comp.nus.edu.sg/cs1101x/2_resources/online.html

- Using your own userID and passwords to access the SMA Hydra Linux Cluster through SSH Client software.
- Manage your upload files through folders: For example Clab1, Clab2, Clab3
- Transfer your code to SMA Hydra Linux Cluster correspond to each Clab.
- Compile your code.
- Run your program.
- Download your results to your PC.

Some softwares/libraries installed in the SMA Hydra Linux Cluster:

<i>Application</i>	<i>Setting</i>
<i>Operatingsystem</i>	<i>CentOS4</i>
<i>GCCCompiler</i>	<i>GCCv3.4.3</i>
<i>IntelCompiler</i>	<i>C++Compiler : /usr/local/intel_cc_81</i>
<i>IntelCompiler</i>	<i>FortranComplier : /usr/local/intel_fc_81</i>
<i>IntelCompiler</i>	<i>Inteldebuger : /usr/local/intel_idb_81</i>
<i>mpichv1.2.7</i>	<i>/usr/local/mpich/gnu(default)</i>
<i>mpichv1.2.7</i>	<i>/usr/local/mpich/intel</i>
<i>mpichv1.2.7</i>	<i>mpicc : TouseCMPI</i>
<i>mpichv1.2.7</i>	<i>mpicxx : TouseC++MPI</i>
<i>SunGridEngine</i>	
<i>Condorv6.6.9</i>	<i>/home/condor</i>
<i>BLASTv2.2.11</i>	<i>/usr/local/BLAST</i>
<i>LAPACK</i>	<i>/usr/local/LAPACK</i>
<i>LAPACK</i>	<i>Libraries : /usr/local/LAPACK/lib</i>
<i>SCALAPACKv1.1</i>	<i>/usr/local/SCALAPACK</i>
<i>BLAS</i>	<i>CBLAS : /usr/local/CBLAS/lib/LINUX</i>
<i>BLAS</i>	<i>FortranBLAS : /usr/local/BLAS/lib/LINUX</i>
<i>BLACSv2.2.11</i>	<i>/usr/local/BLAST</i>
<i>PETScv2.3.1</i>	<i>/usr/local/PETSc/petsc - 2.3.1</i>
<i>Javajdk - 1.5 - 03</i>	<i>/usr/local/jdk1.5_03</i>

Go to <http://hydra.sma.nus.edu.sg> for more information about SMA Hydra Linux Cluster. This page is maintained by NovaGlobal Pte Ltd, <http://www.novaglobal.com.sg>

Question 2: The continuous integration problem

You are required to write two programs to evaluate the integration of a continuous function approximated by using rectangular and trapezoidal method. The approximated methods are given as in the formulation (2) and (3) in the problem. The two skeleton programs (*rectangular_integral.c* and *trapezoidal_integral.c*) are provided.

- How to locate and copy the programs *rectangular_integral.c* and *trapezoidal_integral.c*? The CME5232 course directory in SMA Hydra Linux Cluster is **/home/cme5232**. All the sample program code will store in **/home/cme5232/clab/clabx** (where x is either 1, 2, 3, 4 or 5).

Note: Your account are given to read and execute permission for the CME5232 course directory so that you can copy and read the files.

The following is the sample procedure to copy the given sample program from the CME5232 course directory.

- (1) **mkdir cme5232** in your home directory **cd cme5232**.

- (2) **mkdir clab** and **cd clab**.
- (3) **mkdir clab1** and **cd clab1**.
- (4) **mkdir qn2** and **cp/home/cme5232/clab/clab1/qn2/* qn2**.
- (5) **cd qn2**.

After copy the sample codes into your home directory. You are required to modify the code to get the results as the requirement in the questions. Then save the results into two files named *trapezoidal_result.txt* and *rectangular_result.txt*.

- How to compile the C/C++ program?

C: *gccrectangular_integral.c -orectangular_integral -lm*

C++: *g++rectangular_integral.cpp -orectangular-integral -lm*

- How to run/execute the compiled C/C++ program?

./rectangular_integral

- How to abort the program execution in the case it is require? (Not recommended unless necessary).

Press **Ctrl-C**.

- How to check if the process is successfully removed?

ps -ef: grep userID (e.g. my userID is g0701923).

- What if the job is not successfully removed?

kill -9 pid (where the pid is the process id).

After getting the results, you are required to download the both results files (*trapezoidal_result.txt* and *rectangular_result.txt*) to your own computer and use Matlab to plot the results out as the following procedure;

% Read the *trapezoidal_result.txt* file:

load 'trapezoidal_result.txt';

element1 = trapezoidal_result(:,2);

relerr1 = trapezoidal_result(:,6);

Comptime1 = trapezoidal_result(:,7);

% Read the *rectangular_result.txt* file:

load 'rectangular_result.txt';

element2 = rectangular_result(:,2);

relerr2 = rectangular_result(:,6);

Comptime2 = rectangular_result(:,7);

% Plot the convergence rate

```

loglog(element1,relerr1,'r*-')
hold on;
loglog(element2,relerr2,'bo-')
xlabel('Value of N') ylabel('Relative error') title('Comparison of convergence rate')
grid on;
% Plot the computational time
loglog(element1,Comptime1,'r*-')
hold on;
loglog(element2,Comptime2,'bo-')
xlabel('Value of N') ylabel('Computational time (s)') title('Comparison of Computational time')
grid on;

```

It is NOT recommended to execute the program on the submit node of a cluster, what should we do?

Submit the job using **Sun Grid Engine** (Latest version: N1 Grid Engine 6).

SGE is Sun's resource management solution. It provides the power and flexibility required for campus grids...

Read more.

A resource management software is extremely important in computation - intensive environment, especially when MPI program is required.

In SMA Hydra Linux Cluster, job submission via SGE is compulsory. The job submission will be introduce in the Computational Lab 2.

Question 3: Monte Carlo method to compute the value of pi

It is required to use the Monte Carlo simulation method to compute the value of **pi**. This method consists of inscribing a circle inside a square with length of 1 unit, then throwing the "darts" randomly at the square. The ratio of "darts" that land inside the circle (**N**) to the total number of "darts" thrown (**M**) will be the ratio of the area of the circle to the area of the square, which is **pi/4**. From this the value of **pi** can be calculated from **pi = 4N/M**.

You are expected to write out the sequential version of the code to compute the value of **pi**. Run the code with the varying the total number of the "darts" thrown, $N = 10^4, 10^5, 10^6, 10^7, 10^8, 10^9, 10^{10}$, to record the computational time and value of **pi**. You are also required to calculate the relative error of **pi** versus the true pi value up to 25 digits of accuracy ($PI25DT = 3.141592653589793238462643$) through the following formulation;

$$\epsilon_{rel} = \left| \frac{\pi - PI25DT}{PI25DT} \right|. \tag{1}$$

Save the results into a file named "pivalued.txt", then use Matlab software to plot the convergence rate. You are provided the Matlab program to open the "pivalued.txt" and plot the result via email

or as the following source code;

```
*****  
load 'pivalue.txt'  
stepsize=pivalue(:,2);  
relerr=pivalue(:,4);  
loglog(stepsize,relerr,'*-')  
xlabel('Value of N') ylabel('Relative error') title('Convergence speed of Monte Carlo  
method')  
grid on;  
*****
```