18-645: How to write fast code, Spring 2015

HW #1 – Setting up the infrastructure

Due: January 26, 2015, 11:00pm EST, 8:00pm PST

For this homework go through the tasks mentioned below:

- Task 1: Logging into a cluster machine
- Task 2: Creating a *git* repository
- Task 3: Installing CppUnit and configuring library path
- Task 4: Test running the code
- Task 5: Get information about your processor

Submission:

To submit your homework, you must carefully follow these instructions:

1. Use the text-based answer template for filling in your answers. You can find it under:

   Module 1 Homework 1: **hw1_answer_sheet.txt**

2. To get credit for the homework, please submit your completed answer sheet, save it as: **18645_HW1_Team{0#}_{Last_First}.txt** on Acatar website under Module 1 Homework 1.

Note: Completing HW1 and submitting it as instructed is required to stay in the course.

This is an individual homework. As you start the homework project, you will realize that the homework is designed to help you get started on the project.

Task 1: Logging into a cluster machine

In this task you will test logging into the cluster machines.

Step 1: Use *ssh* login to any cluster. Use your Andrew id & password to login.

```
ssh YOUR_OWN_ANDREWID@ghc52.ghc.andrew.cmu.edu
```

Cluster machines available for your use:

ghcXX.ghc.andrew.cmu.edu, XX = {52,…,80}

Note: The cluster machines share a common home directory. Any change you make in your home directory on one cluster machine (say ghc52) will be immediately reflected in all other machines.

Note: Some cluster machine may be in maintenance at any given time. If one machine doesn't work, please try another machine.

Task 2: Creating a *git* repository

In this task, you are expected to learn how to create the course code repository. Please refer to the *git* tutorial at the end of Lecture 1 for more details.

After creating git repository, please provide YOUR_OWN_ANDREWID and YOUR_PARTNER_ANDREWID to **18645@sv.cmu.edu** and wait for a confirmation mail containing your team id.

Steps:[1]

```
# Create a directory within AFS to house the GIT repository
# ONE partner should execute this command within her/his AFS space for the
# team. The person doing this should have enough quota to hold the repository.
$ mkdir -p ~/645/REPOSITORY

# Give the "Other partner(s)" and course staff member AFS access to this space, while exlcuding others
$ find ~/645 -type d -exec fs sa -dir '{}' -acl YOUR_OWN_ANDREWID all -clear \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl OTHER_PARTNER_ANDREWID all \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl jchong all \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl ianlane all \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl akshayc all \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl jungsuk1 all \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl bingl1 all \;
$ find ~/645 -type d -exec fs sa -dir '{}' -acl dmcfarli all \;


# Execute these commands only once on AFS to actually create the repository
$ cd ~/645/REPOSITORY
$ git init --bare fastcode

# You and all partners do this on AFS as well, to give each of you a
# place to actually do your individual work
$ cd ~/645
$ git clone --no-hardlinks ~ANDREWID_OF_HOSTING_PARTNER/645/REPOSITORY/fastcode

# One of you adds the project handout via your personal repository
$ cd ~/645/fastcode
$ wget https://cmu.box.com/shared/static/j23oi4x7pavszgi5yx6sx39zsoygwqax.gz -O 18645_spring_15.tar.gz
$ tar xzf 18645_spring_15.tar.gz
$ git add .
$ git commit -a -m "Initial commit"
```

[1] Steps taken from 15-440 (https://www.andrew.cmu.edu/course/15-440-s12/applications/labs/lab2/git.html)

Task 3: Installing CppUnit and configuring library path

Step 1: Run the following commands to download CppUnit and install it in your home directory.

```
wget http://downloads.sourceforge.net/cppunit/cppunit-1.12.1.tar.gz -O cppunit.tar.gz
mkdir ~/cppunit
tar xzvf cppunit.tar.gz
cd cppunit-1.12.1/
./configure –prefix=${HOME}/cppunit
make
make install
rm –rf ~/cppunit-1.12.1/
```

Step 2: Specify LD_LIBRARY_PATH to include the CppUnit library.
If you're using bash shell,

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/cppunit/lib
```

If you're using csh shell,
Environment variable LD_LIBRARY_PATH needs to be set before running the code. You can add it to your shell login file to avoid typing it every time.

```
setenv LD_LIBRARY_PATH "${LD_LIBRARY_PATH}:${HOME}/cppunit/lib"
```

Task 4: Test running the code

Step 1: Compile the sequential version of matrix multiplication:

```
cd ~/645/fastcode/matrix_mul/sequential
make
./matrix_mul –i  ../matrix_mul_01.dat –o
```

Note: If you can't run ./matrix_mul, probably you need to set LD_LIBRARY_PATH (see Task 4).

Question 1: Report the speed of execution of all three test cases using the sequential version on one of the cluster machines.

Task 5: Get information about processor

To answer the following questions, please indicate the source of the data or the command line used.

Question 1: For *ghc52.ghc.andrew.cmu.edu*, log in and determine:

a) The exact processor model
b) The number of physical cores it has
c) The number of virtual cores it has
d) L1 cache size specification
e) L2 cache size specification
f) The level of SSE the processor supports

Question 2: For your local machine, determine:

a) The exact processor model
b) The number of physical cores it has
c) The number of virtual cores it has
d) L1 cache size specification
e) L2 cache size specification
f) The level of SSE the processor supports