CS484: Parallel Programming - Spring 2017
MP0
Due Date : Monday, January 30, 5:00PM
Submission Method : SVN

January 23, 2017

1 Introduction

The objective of this assignment is to introduce you to Taub, the campus cluster that you will be using for most of this course. We have provided you with login accounts. More information regarding Taub can found here : `https://campuscluster.illinois.edu/user_info/doc/beginner.html`.

1.1 Access

Access to the cluster is using your **NetID**. You can login to the cluster using the ssh command :

ssh -Y NetID@taub.campuscluster.illinois.edu

The cluster has several packages for software development. You can add or delete necessary packages with the *module* command. Some useful commands on Taub are provided below :

1. module avail : provides a list of all available packages on Taub.
2. module list : provides a list of packages that are currently available in your environment.
3. module load *package* : add the software *package* to your environment.
4. module unload *package* : removes the software *package* from your environment.

There are typically two types of nodes on Taub - login nodes and compute nodes. You can find more information about Taub nodes on the campus cluster website. You should submit jobs to the compute nodes via the *Torque* job scheduler. We have provided you with a script to do that. Please don't execute any of your programs on the login nodes.

Some useful commands for accessing the job scheduler are :

1. qsub *mp0_script* : submit the job with name *mp0_script* to the scheduler.

2. qdel *job_id* : delete the job with sequence number *job_id* from the scheduler.

3. qstat -u *Netid* : use this command to check the status of the jobs you have submitted to the scheduler.

Login to campus cluster and make sure you don't have any issues accessing the nodes. In case you are denied permission, please send an e-mail to the instructor or the TAs. Copy the assignment to your Taub account using the following command :

scp mp0.tar NetID@taub.campuscluster.illinois.edu://home/NetID

Untar mp0.tar :

tar -xvf mp0.tar

This should create a folder *mp0* in your home directory. We want you to use the intel compiler for this assignment. Load the intel compiler using :
module load intel

2 Assignment

There are two parts to this assignment - part A and part B.

2.1 Part A

This assignment introduces you to spatial and temporal locality in memory accesses. We have provided you with a code that performs a series of irregular accesses to a contiguous array in memory. You don't have to make any changes to the program. Compile the code using the *makefile* provided and execute it on the compute nodes (qsub batch_script). When the job completes, you should find the output file(.txt) in the same folder.
Answer the following questions :

1. Read the code, plot a graph of the execution time vs number of irregular memory accesses.

2. Explain the trend in the number of irregular memory accesses and the total running time of the code.

3. Compare your results with the cache sizes of the compute nodes.

2.2 Part B

This section is meant to familiarize you with some compiler optimizations. We have provided you with five small functions - each with a different loop structure. You have to do the following :

1. Compile the test programs using -O0, -O1, -O2 and -O3 optimization flags. The output file (.txt) should contain the execution times for the test programs. Examine the optimization reports generated for each function. Explain the optimizations performed by the compiler and correlate it with the execution times observed.

3 Submission Guidelines:

You have to submit the following files :

1. Report containing the graph and analysis for part A.
2. Execution times for each test program in Part B, along with the compiler optimization level. Explain the optimization performed by the compiler and justify the execution time you observed.

Submission Method : You have to submit your assignment via svn.

1. Create a directory **MP0** in your svn folder.
2. Add the report containing the graph and analysis for Part A and the measurements and justification for Part B.
3. Add the directory to svn using **svn add**
4. Submit your changes to svn using **svn commit -m**

Here is a link to some useful svn commands that may be of use to you :
`http://www.linuxfromscratch.org/blfs/edguide/chapter03.html`