**CS420/CSE402/ECE492 Parallel Programming for Scientists and Engineers**

**Fall 2012**

**Machine Problem 1: Effects of Locality**

**Due: Monday, September 17, 2012 at 11:59:59 p.m**

This MP is related to cache locality and is divided into three parts. You are asked to implement three different numerical operations in two ways. The first way is a naïve, straightforward implementation. The second way uses tiling as discussed in class to improve cache locality and hence, execution time. The purpose is to compare both these solutions for each of the three problems and calculate the speedup resulting from using tiling. The three parts are as follows:

**Part A**

Calculate the transpose of a square matrix A and store it back in it i.e. A.

*A = AT*

**Part B**

Calculate the Matrix-Matrix product of square matrices A and B and store it in C.

*C = AB*

*(This A should be the original matrix A and not the transpose)*

**Part C**

Calculate the 5-point stencil over a 2D array (matrix) ‘E’ and store it in the same matrix i.e. E. Given a **square** grid (matrix E) in two dimensions, the 5-point stencil of a point in the grid is made up of the point itself together with its four neighbors. For example, in the figure below, the value of ‘x’ would be the average of the values in cells ‘n’, ‘w’, ‘s’, ‘e’, and ‘x’ itself. The following loop would update all the points in the grid.

**for(i=1;i<n-1;i++)**

**{**

**for(j=1;j<n-1;j++)**

**E[i][j] = (E[i][j] + E[i][j-1] + E[i][j+1] + E[i-1][j] + E[i+1][j])/5;**

**}**

Notice that the first and last columns and the first and last rows remain constant. That is **E[0][:]**, **E[n-1][:]**, **E[:][0]**, and **E[:][n-1]** are not changed by the loop. If E is (x,x) matrix, you can assume that (x-2) will be divisible by tile size.

Your program should be able to read input matrices from data set files. You will be required to read each matrix from a txt file where the first line specifies the number of rows (It is a square matrix. Therefore, we don’t need to input the number of columns since it is the same as the number of rows). There will be n\*n lines following the first line. Each line will contain the value of one element in row major format i.e. first n lines will contain all elements in row M[0][:] where M is the input matrix. Your program should be able to accept as the first command line argument the name of the data set. Each data set will be stored in three files. Each file will correspond to one of the four input matrices: *A*, *B*, *C* and *E*. Furthermore, your program should also be able to accept the tile size as the second command line parameter.

For example, if my dataset name is **set1**, the following files would contain the matrices.

**set1\_A.txt**

**set1\_B.txt**

**set1\_E.txt**

The following command should be able to perform all three operations using both naïve and tiled versions:

**./mp1 set1 16**

(**set1** is the name for dataset, tile size is **16** and mp1 is your program executable)

You should write the output matrices using the following names in the same format as the input matrices:

**out\_A.txt** (naïve transpose of matrix A)

**out\_A\_t.txt** (tiled transpose of matrix A)

**out\_C.txt** (naïve product of B and C)

**out\_C\_t.txt** (tiled product of B and C)

**out\_E.txt** (naïve 5-point stencil over E)

**out\_E\_t.txt** (tiled 5-point stencil over E)

Other than writing the matrices, you are also required to output the execution time and speedup in a txt file (results.txt) using the following format.

**Operation Naïve Time Tiled Time Speedup**

Transpose XXXXXX XXXXX NaïveTime/TiledTime

Matmul XXXXXX XXXXX NaïveTime/TiledTime

Stencil XXXXXX XXXXX NaïveTime/TiledTime

**Note: Failure to follow the naming conventions would result in zero points.**

**How to use the MTL machines**

You can login to your machines using the account name and password provided to you. Please make sure your folder is not viewable to other people. You can use **gcc** compiler to compile your C code i.e. **gcc –O3 –o mp1 mp1.c**.

Since it is a shared machine, you need to submit a job in order to run your program. You can do it using the **qsub** command. You need to create a script file that the **qsub** command uses. You can just copy paste the following line into a file named **runmp1** and change its permissions to 777 (using **chmod 777 runmp1**).

**#!/bin/sh**

**#PBS -N mp1**

**#PBS -j oe**

**<path\_to\_Mp1>/mp1 set1 16**

Once you have created and changed permissions for the **runmp1** file, you can use the following command to run your program.

**qsub -l select=1:ncpus=1 runmp1**

After submitting the task using the above command you will see the following:

**[dapa@acano01 check]$ qsub -l select=1:ncpus=1 runmp1**

**98655.acaad01**

**[dapa@acano01 check]$**

It means that the task was submitted successfully with the job Id given in bold.

You can check the status of the job by typing ‘**qstat –a**’ and looking at the job id you got when submitting the job. A value of ‘R’ in the ‘S’ column means that your task is running. After the task completes, you will see an output file generated in the same folder from where you submitted the task (with the name mp1.oXXXX).

**Testing your Program**

We are providing two datasets on the assignments page. You can run these sets and match your outputs with the output files that we have provided. You can match the outputs by using **diff** command of linux e.g. **diff my\_output TA\_output**. If the **diff** command does not return anything, this means that the files are identical. Your program will be correct if the six output files are identical to the files provided.

**Connecting to MTL**

You can get to the MTL machines using your own machines or the EWS workstations located in the basement of Siebel Center. Although you need to connect to MTL using VPN when using your own computers, you can straight away ssh to MTL when using EWS machines in Siebel Center’s basement. You can find the directions for creating and connecting to VPN and SSHing to the machines.

**How to use VPN**

Cisco® owns the rights for the VPN client – it can be obtained from this site:

http://www.supreme.state.az.us/downloads/VPN/

Download and install the Cisco VPN client either the 32-bit or 64-bit client as appropriate.

*Note: These only work with Windows.*

Within the VPN client please specify the following (under Group Authentication):

**Host: 192.55.51.80**

**Name:** VPN2

**Password: <Professor will tell in class>**

Connection Entry & Description (not significant)

Note: when the VPN tunnel is connected you will not be able to use your machine to connect to anything else, i.e. the web or other systems.

**How to SSH**

After you are on MTL network, you just need to ssh to the machine. You can do that by:

ssh <username>@192.55.51.81

It will ask for your password. Once you punch in the password, you will be ready to use the machine.