**Reagan Leonard and Caroline Case**

Project 1: Benchmarking computers

Assigned: 2/3

Due: 11:59 PM Wednesday, 2/26 on Canvas

Each student must turn in his/her project report. You may work together with another student from the class in a study group. Include the names of your study group members on the solution set you submit.

In this project, you are going to select two different computers with different processors and memory configurations, use a program to measure the computers’ performance, and analyze the performance data and report.

Linux machines are required for this project. You have the freedom to select your own machines or the lab machines. Don’t use virtual machines for this project.

The benchmark y-cruncher can be downloaded from http://www.numberworld.org/y-cruncher/y-cruncher%20v0.7.8.9503-static.tar.xz, which is available from the website http://www.numberworld.org/y-cruncher/#Features. Note that the downloadable is Linux (Static), which I have tested on a lab machine.

**Experiments**

1. Download and install y-cruncher on the machines you select.

**ada15 vs joey15**

1. Run y-cruncher and collect the execution time for calculating pi with one thread. For example, the following command executes pi with 50 million decimal digits with 1 thread and outputs the execution time. Repeat five times and record the average execution time.

v0.7.8.9503-static [261] y-cruncher custom pi -TD:1

#execute pi with 1 thread. One line of the outputs gives the time to calculate pi. The line is similar as follows:

###############

Pi: xx.xxx seconds ( xx.xxx minutes )

###############

1. Repeat step 2 for different numbers of threads from 1 to the total number of logical processors. For example, the following command executes pi with 50 million decimal digits with 24 threads, and outputs the execution time

v0.7.8.9503-static [262] y-cruncher custom pi -TD:24

#execute pi with 24 threads

1. Repeat steps 2 & 3 for two other different constants: log2 and gamma. Simply replace pi with log2 or gamma in the commands. Record “Log(2): xxx.xxx seconds ( xx.xxx minutes )” for log2, and “Euler's Constant: xxx.xxx seconds ( xx.xxx minutes )” for gamma.

**Experiment report**

Include the following information in your report.

1. The manufacturer, model, CPU clock of the processor on the computer. For example, machine newton is Intel(R) Xeon(R) CPU E5645 with 2.40GHz. There are two processors, each with 6 physical cores, and each physical core supports 2-way hyperthreading. Totally, there are 24 logical processors. You can find this info with Linux command “cat /proc/cpuinfo” and search online what processor, CPU cores, physical id, siblings, and core id mean.

**Joey15**: Intel(R) Core(TM) i7-4790 CPU @ 2.60GHz, 2 processors, each with 4 physical cores, and each physical core supports 2-way hyperthreading. So in total, there are 8 logical processors.

*(Processors: 0-7, physical id: all 0, siblings: 8, core id: 0,1,2,3,0,1,2,3, cpu cores: all 4)*

**Ada15**: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz, 2 processors, each with 4 physical cores, each running one thread. So in total, there are 4 logical processors.

*(Processors: 0-3, physical id: all 0, siblings: 4, core id: 0-3, cpu cores: all 4)*

1. Available physical memory on the computer. You can find this info with Linux command “cat /proc/meminfo”.

**Ada15**: 14372048 kB, **Joey15**: 29485592 kB

1. Report the average execution times in seconds in a table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Pi | | Log(2) | | Gamma | |
| Machine 1 | Machine 2 | Machine 1 | Machine 2 | Machine 1 | Machine 2 |
| TD:1 | 11.359 | 14.381 | 31.7436 | 40.0064 | 292.5061 | 367.3363 |
| TD:2 | 4.2618 | 5.2432 | 11.109 | 14.0122 | 105.9424 | 127.3187 |
| TD:3 | 4.0468 | 5.182 | 10.8432 | 13.9709 | 103.6612 | 127.9865 |
| TD:4 | 3.513 | 4.06 | 9.2877 | 11.0927 | 88.397 | 101.3972 |
| TD:5 |  | 4.1588 |  | 11.4126 |  | 103.412 |
| TD:6 |  | 4.0456 |  | 11.1563 |  | 100.8211 |
| TD:7 |  | 4.0328 |  | 11.292 |  | 100.8993 |
| TD:8 |  | 3.9438 |  | 10.742 |  | 98.335 |

1. Analyze the parallel speedup for each constant execution on each machine, and present the results with speedup-#threads plots. You should have two figures, one for computer I and one for computer II. Each figure has three speedup curves for the three constant executions.