Midwestern State University

Parallel Computing | Homework # 4

Pthreads Programming

**READ THESE INSTRUCTIONS**

1. You are allowed to work this assignment in groups of two
2. Only one submission per Team
3. If you decide to work your assignment alone, make sure that your name is not included in ANY OTHER file being sent by any other team. If you have two or more submissions your grade will be the LOWEST grade.
4. At the top of any code, you are required to include
   1. Name and Last Name for Student # 1
   2. Name and Last Name for Student # 2
   3. Compilation instructions
   4. Execution Instructions
   5. **MUST USE STAMPEDE2**

**Deadlines:**

**Via D2L:** Tuesday November 12 at 8:00 am.

* **E-copies of both \*.c files**
* **E-copies of your report (Word Document), with tasks 3 and 4 fully documented**

**In Class:** Tuesday November 12 (at the Beginning of class): **A hard copy of all your code**.

Goals:

* To reinforce the concept of shared memory programing by using pthreads
* Prepare and develop parallel programming skills and proficiency.
* To understand how to compile, run, and write pthreaded programs.
* To reinforce the concept of speedup and performance analysis

You are going to create a pthreaded array calculator. The calculator must be able to perform the typical traditional arithmetic operations, such as addition, substraction, multiplication, division.

* Arrays a, and b are integer arrays, with 1’000.000 million integers each.
* All the elements in array a are 1, while all those in b are 2.
* The addition function will create array “addition”, in which addition[0] = a[0] + b[0], etc (You get the idea). Follow a similar logic for substraction, multiplication and division.

**Task # 1**

You need to write serial code that uses functions calls, where the functions will be called in the following order.

**Clock tic here**

addition()

susbtraction()

multiplication()

division()

**clock tic here**

report the time on screen

This File must be named as follows: **LastNamesA4NoThreads.c** [15 points]

**Task # 2**

Take the code that you wrote for Task # 1, and convert it into a threaded program that uses four (4) threads. In this program

**Thread 0 will take care of the addition**

**Thread 1 susbtraction**

**Thread 2 multipliaction**

**Thread 3 division**

**Time it in the correct place, report the time on screen**

This File must be named as follows: **TeamNameA4YesThreads.c**  [70 Points]

**Task # 3:** [10 points]

**Run each program 5 times, take notes of the times [same scale], complete the performance analysis and complete the table below: handwriting is not valid. Type the values in word**

|  |  |  |
| --- | --- | --- |
| **Task 1 [secs]** | **Task 2 [secs]** | **Speedup (Task1/Task2)** |
| 0.092827 | 0.021843 | 4.249736 |
| 0.090528 | 0.021576 | 4.195773 |
| 0.093221 | 0.021826 | 4.271098 |
| 0.091507 | 0.021997 | 4.159976 |
| 0.092617 | 0.021915 | 4.226192 |
| **0.09214‬0** | **0.0218314** | **4.220555** |

**Task # 4:** [5 points]

**Write your conclusions and let me know which one is faster? Handwriting is not valid.**

In conclusion, the pthread version of this program is clearly faster. With over four times faster speeds the use of pthreads outweighs the sequential programming in this instance. Likely due to the pthreads ability to complete each function at the same time, and creating threads has little overhead on time.