Roberto Garcia Torres

A00822089

**Tarea PI OpenMP**

Create a single program that does the following 3 things:

1.- Using the original NON-Multithreaded code, run it 5 times, record and average them to get a baseline. Add the results to a table.

2.- Use Windows/Posix multithreaded code. Use the amount of threads equal to the number of virtual processors you have in your computer. Run it 5 times and also get the average. Record the results to the table.

3.- Use OpenMP to parallelize your code. Again, use the same amount of threads you used in #2. Run it 5 times and get the average. Record results in the table.

| Corrida/Metotdo | Plain | PosixThreads | OpenMP |
|---|---|---|---|
| 1 | 64 | 29 | 33 |
| 2 | 62 | 45 | 55 |
| 3 | 58 | 44 | 39 |
| 4 | 57 | 53 | 44 |
| 5 | 55 | 48 | 27 |
| Promedio | **59,2** | **43,8** | **39,6** |

In a document, answer these:

**a.- Which code ran faster?**

The OpenMP code.

**b.- Was it easier of harder to use OMP over Windows Threads?**

No, OMP was way easier as I didn't had to manually do the fork and join operations ans well as using the mutex lock.

**c.- Include your table with all your runs and averages.**

The table is above.

**Conclusions**

Honestly I thought that the posix threads alternative was going to be the fastest one since I was doing all the thread related instructions manually in a way that's optimal for the code I'm doing. However, to my surprise, the openmp code ran faster. My hypothesis as to why this happened is that the people that coded openmp know how to use the instructions I used in a very efficient way and therefore achieve better performance; if they were to make the same code I did with these more efficient methods probably they'll do better than the openmp library. It'll be good to research the best practices for optimal performance when doing code with posix threads. Also it's important to remark that doing the openmp code was way easier and involved les LOCs.

Include your code. Visual Studio, a ZIP file with the full Solution. If Linux, a ZIP file with all the code files; in each code file you must include the compile line used as a comment at the beginning of the file.

Careful! Watch out for RACE CONDITIONS!