**CSC410**

*Assignment G3*

Diana Lewis

Alex Houser

Marty Moradian

Yanlin Li

Stephen Chilcote

**Time Testing**

Sequential

|  |  |  |  |
| --- | --- | --- | --- |
|  | 300 | 1000 | 2000 |
| Real | 0m0.274s | 0m9.636s | 1m29.187s |
| User | 0m0.272s | 0m9.609s | 1m29.126s |
| Sys | 0m0.000s | 0m0.020s | 0m0.024s |

Pthread

|  |  |  |  |
| --- | --- | --- | --- |
|  | 300 | 1000 | 2000 |
| Real | 0m0.659s | 0m7.803s | 1m10.423s |
| User | 0m0.536s | 0m6.556s | 1m0.913s |
| Sys | 0m0.000s | 0m0.036s | 0m0.080s |

Omp

|  |  |  |  |
| --- | --- | --- | --- |
|  | 300 | 1000 | 2000 |
| Real | 0m0.117s | 0m1.941s | 0m9.780s |
| User | 0m1.052s | 0m23.461s | 3m7.972s |
| Sys | 0m0.008s | 0m0.396s | 0m0.060s |

**Conclusions/comparisons**

Ease of writing: It was very easy to write the OpenMP code. Everything could be written as if for a sequential program, and adding one pragma (and compiling with -fopenmp) is enough to add parallelism. In addition to OpenMP, sequential was also very easy to write. Mostly due to the fact that we have more experience coding this way verses OpenMP or Pthread.

Performance: Using OpenMP led to a jump in the system time required versus the sequential code, probably due to scheduling. The real time was significantly less in the OpenMP version, however. For human purposes, real time is usually the most important. The sequential code version uses a different method to populate two grids. It uses random numbers to populate the grids while the other versions (omp & pthread) use an input file. We think using either method makes little difference in the amount of time and in user experience. Calculating the random numbers and reading in a file should take up approximately the same amount of time.

Timing & speed-up tests: