Parallel Matrix Multiplication -

Our task after creating a serial matrix multiplication program was to make a parallel version of the same program using pymp (OpenMP library for python). The documentation for pymp is pretty scarce with a a few basic examples, so most of it was trial and error. Since this is the first time I've ever written a parallel program I reviewed the notes from our class and watched a couple of Youtube videos on OpenMP. One challenged I faced was that my results array was generating incorrect output, this was fixed by making it a shared array with an int datatype. This program took me about 2 hours total, this includes researching and watching videos to familiarize myself with OpemMP.

Performance Measurements - I'm multiplying a (16 x 26) X (26 x 10) matrix and getting a (16 x 10) matrix as the result. I'm using timeit library to time the execution of the program and using the average of 5 tests for these results.

1 thread -

```
linux-4oae:/home/student/Desktop/parallel/parallel-
elll # python3 matrixparallel.py
Time:  0.04962505200091982
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
```

2 threads -

```
linux-4oae:/home/student/Desktop/parallel/parallel-c
elll # python3 matrixparallel.py
Time:  0.0409761340051773
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
```

4 threads -

```
linux-4oae:/home/student/Desktop/parallel/parallel-c
elll # python3 matrixparallel.py
Time:  0.04073269400396384
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
```

8 threads -

```
linux-4oae:/home/student/Desktop/parallel/parallel-c
elll # python3 matrixparallel.py
Time:  0.046723619001568295
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
[2600 2600 2600 2600 2600 2600 2600 2600 2600 2600]
```

Analysis - All the times for my results were pretty much about the same even when changing the amount of threads, this is something that I'm going to research more.

CPU Info Dump -

```
student@linux-4oae:~> dumpCPUInfo.sh cpuinfo.txt
model name      : Intel(R) Core(TM) i5-4278U CPU @ 2.60GHz
      4      36      216
student@linux-4oae:~>
```