Matrix Multiplication using MPI:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MPI | | | | | | |
| Results are time in nanoseconds. | | | | | | |
|  | Slave Count | 1 | 2 | 3 | 4 | 5 |
| Size |  |
| 5x5 | | 33303900 | 31880700 | 32827200 | 32640100 | 31946600 |
| 10x10 | | 25886900 | 32103200 | 31703600 | 32398400 | 32797200 |
| 25x25 | | 32126700 | 32298800 | 31903800 | 34389900 | 32572600 |
| 50x50 | | 33242900 | 31215800 | 32392800 | 32236100 | 21738800 |
| 100x100 | | 33801900 | 31552200 | 31153900 | 23885300 | 32027800 |
| 250x250 | | 49838900 | 27814000 | 37091800 | 28127000 | 34925700 |
| 500x500 | | Crashed | Crashed | 147604000 | 123780700 | 97636500 |

For some reason, when using MPI in the 2 initial tests, it would crash when using 1 or 2 threads, and I was unable to figure out why. Double checking the output matrix showed that the program was correctly outputting the matrix, but I would get a MPI error in the command prompt. At 3 or higher threads, no matter how many times I ran the program, it would work perfectly. I can make a guess that sending the larger chunks of 2d arrays caused an error for the MPI functions, but I couldn’t figure out a way to rectify it.

Matrix Multiplication with MPI and OpenMP:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MPI & OpenMP | | | | | | |
| Results are time in nanoseconds. | | | | | | |
|  | Slave Count | 1 | 2 | 3 | 4 | 5 |
| Size |  |
| 5x5 | | 33598200 | 32837600 | 70354200 | 1322300 | 37533200 |
| 10x10 | | 31958300 | 34563300 | 21345300 | 7200900 | 74083500 |
| 25x25 | | 34247400 | 19040800 | 3592900 | 1219300 | 74299300 |
| 50x50 | | 32699300 | 44777700 | 58130300 | 1498300 | 112484600 |
| 100x100 | | 32253300 | 32506500 | 36426700 | 1698800 | 33089200 |
| 250x250 | | 28055000 | 15422700 | 32949800 | 33533300 | 65501300 |
| 500x500 | | Crash | Crash | 94307700 | 120510400 | 98478000 |

Matrix Multiplication with OpenCL and MPI:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MPI & OpenCL | | | | | | |
| Results are time in nanoseconds. | | | | | | |
|  | Slave Count | 1 | 2 | 3 | 4 | 5 |
| Size |  |
| 5x5 | | 1174900 | 1426400 | 1895800 | 2294300 | 3048600 |
| 10x10 | | 1234000 | 1451700 | 2859100 | 2402500 | 2767000 |
| 25x25 | | 1248800 | 1527300 | 1851800 | 2330800 | 2767800 |
| 50x50 | | 1258100 | 1480600 | 2054400 | 2323600 | 3521100 |
| 100x100 | | 1395700 | 1650500 | 1933100 | 2442100 | 2669500 |
| 250x250 | | 3556000 | 2654000 | 2918500 | 3482000 | 3805000 |
| 500x500 | | 9937000 | 9682100 | 11607200 | 5719800 | 6381100 |

The results from all 3 of these tests are inconsistent and strange. There seems to be some improvement, but not as much as I would expect. I have tested that the system is running as I would hope, and MPI and OpenCL are both running like expected, but the results are a tiny, if not insignificant improvement. This may be because threads are imitating multiple PCs on a network for MPI. Either way, the MPI and OpenCL proved to be the best method of the 3.