The principle problem encountered was the difference in data-access overhead between python’s built-in data structures and pymp’s shared-memory data structures. Writing the serial matrix multiplication script using python’s lists, and writing the parallel e

quivalent using pymp’s shared arrays provided counter-intuitive results: that the serial version ran faster in all cases. This was only rectified when the serial version was rewritten to use pymp’s shared array structure to provide the same overhead to both variants of the matrix multiply script. This overhead difference was not apparent from pymp’s sparse documentation, but only from reviewing pymp’s source and identifying the mechanism by which it enables shared memory facilities in its data structures.

No problems or bugs remain in the program. Work is divided between multiple logical cores in the processor die.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Threads | 1 | 2 | 4 | 8 |
| Time | 20 | 10 | 10 | 10 |

The assignment took approx. 45 minutes to complete, but a day’s time to identify the discrepancy in overhead mentioned above. The performance measurements are located below.