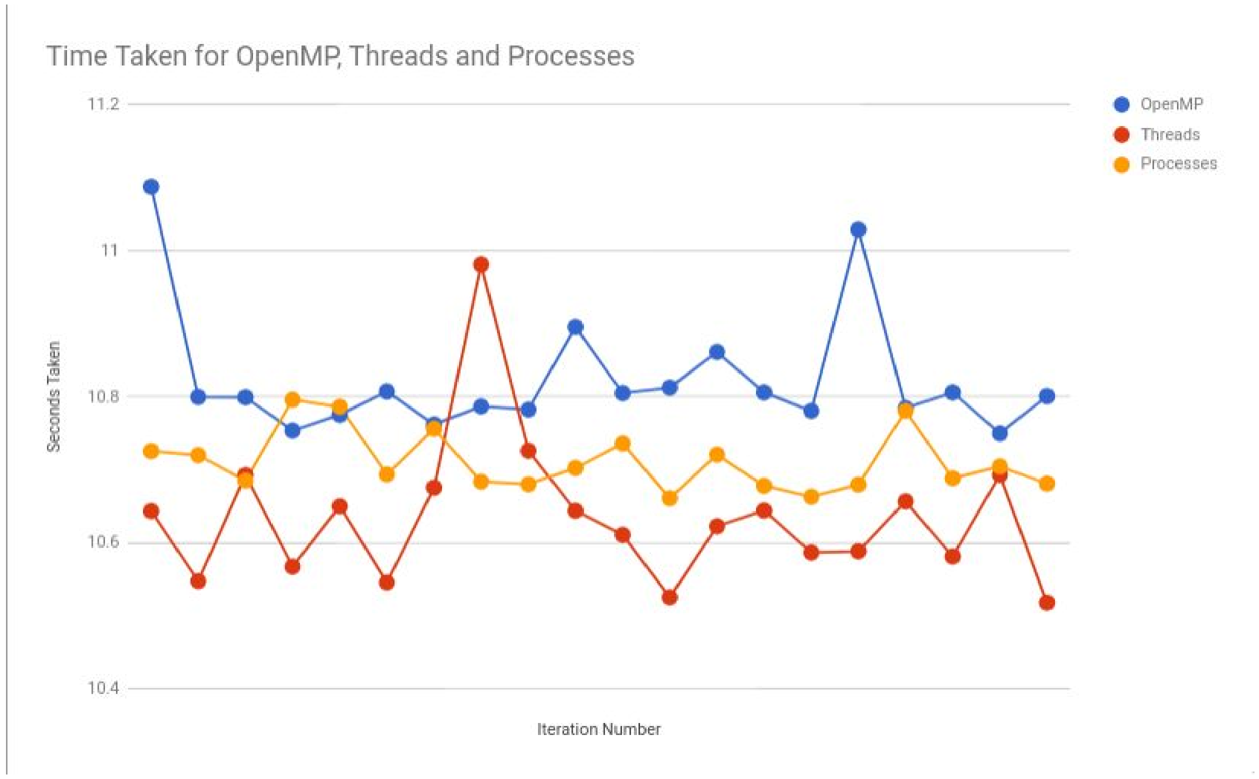
8005 Asn1 - Report

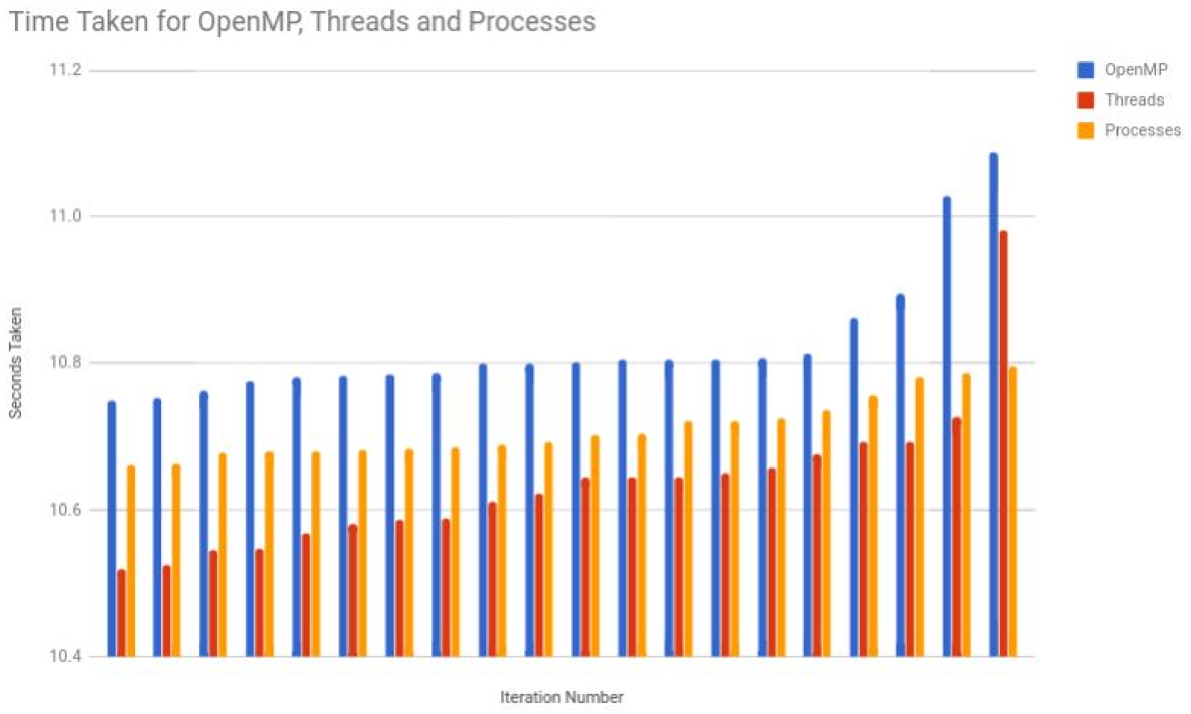
When knowing is half the fun

Isaac Morneau; A009584050

Analysis of Dataset



This data, as shown raw in Appendix A, shows the iteration timing in order compared to each parallelism technique. The overall fastest for this is threads followed by processes and finally OpenMP.



A sorted comparison of the same time graph shows that with the exception of a couple runs the time taken was largely consistent. It's clear with the sorted data that threads still clearly win overall despite the large jump.

	OpenMP	Threads	Processes
Average Time	10.82429	10.635195	10.711265

When the average is calculated it is found that what the graphs indicate is correct and that threads are the fastest followed by processes and then OpenMP.

This results in threads being 1.778% faster than OpenMP and 0.715% faster than processes.

Conclusion

This analysis was directly the opposite of the starting assumptions. I had assumed that OpenMP would be optimized to the point it was useless to write your own task breakup. Often times it is nontrivial and thus OpenMP still holds an advantage there. This does not change the fact that the data shows OpenMP to be the slowest.
Next slowest was processes which again surprised me. This was due to the fact I thought that the scheduler would give higher priority to processes than threads. Instead it was found that processes were in fact still slower than threads though faster than OpenMP.

Threads after having been assumed to be the slowest for the above reasons. Though the fastest this is still only a difference of ~0.1 seconds over a nearly 11 second run time and less than 2% speed overall.

Appendix A

Raw Output data

The following was used to run the programs in all three modes for 500 branches and 20 iterations each and was used for the dataset:

```
18:17:41(master)isaac@isaacbox:8005-asn1$ ./bin/8005-asn1 -o -b 500 -i
20 && ./bin/8005-asn1 -t -b 500 -i 20 && ./bin/8005-asn1 -p -b 500 -i 20
OpenMP seconds:
11.0873
10.7998
10.7994
10.7537
10.7753
10.8073
10.7621
10.7868
10.7825
10.8955
10.805
10.8125
10.8615
10.806
10.7807
11.0289
10.7847
10.806
10.7499
10.8009
Threads seconds:
10.6436
10.5478
10.6929
10.5676
10.6501
10.5459
10.6757
10.9811
10.7262
10.6441
10.6111
10.5255
10.6228
10.6444
```

```
10.5869
10.5886
10.6569
10.5815
10.693
10.5182
Processes seconds:
10.7257
10.7202
10.6853
10.7961
10.7864
10.6938
10.7563
10.6839
10.6802
10.7027
10.7361
10.6611
10.721
10.6781
10.6632
10.6798
10.7808
10.6886
10.7048
10.6812
18:28:35(master)isaac@isaacbox:8005-asn1$
```

Appendix B

The following is the grid of the data from appendix A used to produce the graphs.

Unsorted Data

OpenMP	Threads	Processes
11.0873	10.6436	10.7257
10.7998	10.5478	10.7202
10.7994	10.6929	10.6853
10.7537	10.5676	10.7961
10.7753	10.6501	10.7864
10.8073	10.5459	10.6938
10.7621	10.6757	10.7563
10.7868	10.9811	10.6839

10.7825	10.7262	10.6802
10.8955	10.6441	10.7027
10.805	10.6111	10.7361
10.8125	10.5255	10.6611
10.8615	10.6228	10.721
10.806	10.6444	10.6781
10.7807	10.5869	10.6632
11.0289	10.5886	10.6798
10.7847	10.6569	10.7808
10.806	10.5815	10.6886
10.7499	10.693	10.7048
10.8009	10.5182	10.6812

Sorted Data

OpenMP	Threads	Processes
10.7499	10.5182	10.6611
10.7537	10.5255	10.6632
10.7621	10.5459	10.6781
10.7753	10.5478	10.6798
10.7807	10.5676	10.6802
10.7825	10.5815	10.6812
10.7847	10.5869	10.6839
10.7868	10.5886	10.6853
10.7994	10.6111	10.6886
10.7998	10.6228	10.6938
10.8009	10.6436	10.7027
10.805	10.6441	10.7048
10.806	10.6444	10.7202
10.806	10.6501	10.721
10.8073	10.6569	10.7257
10.8125	10.6757	10.7361
10.8615	10.6929	10.7563
10.8955	10.693	10.7808

11.0289	10.7262	10.7864
11.0873	10.9811	10.7961

Averages

10.82429	10.635195	10.711265