PARALLEL CONVEX HULL

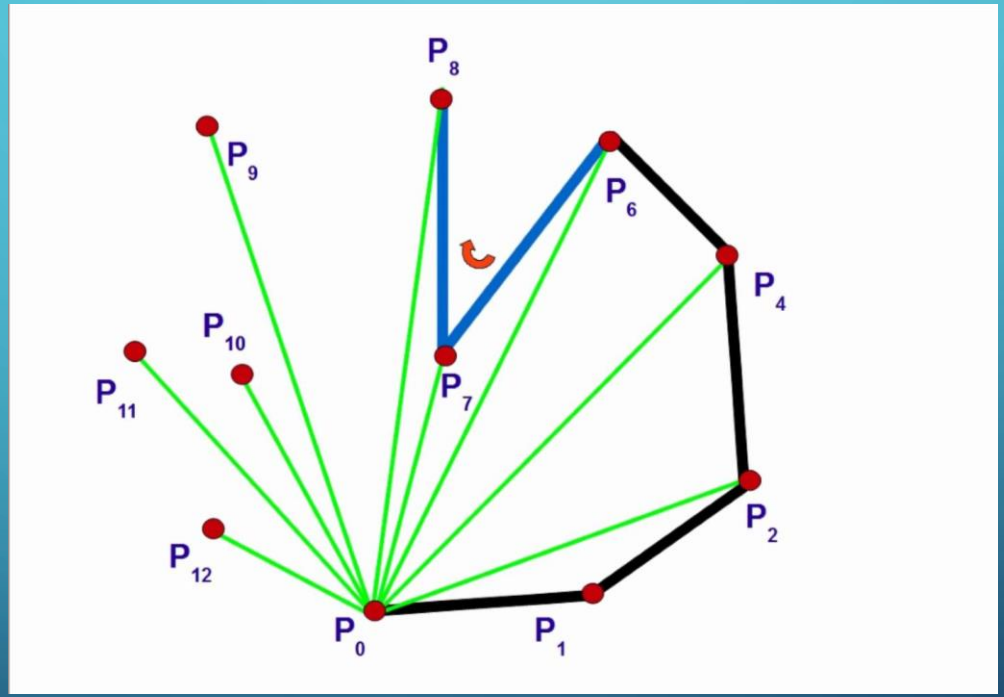
BY SANJAY MUTHU MANICKAM

UNDER THE SUPERVISION OF DR. RUSS MILLER

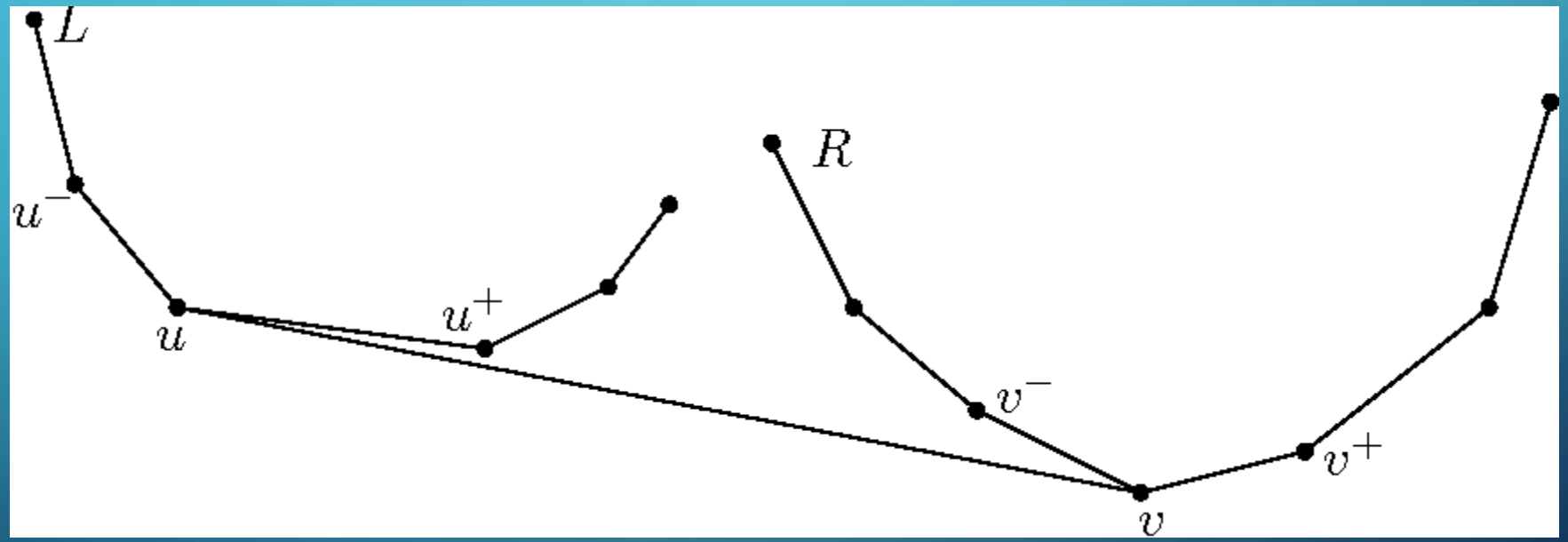
CONVEX HULL

- The Convex hull of a set of X points that is the smallest convex set that contains X.
- It can be solved using the Graham Scan method.

GRAHAM SCAN



CONVEX HULL PARALLEL IMPLEMENTATION



- The points are divided into two halves.
- Compute the convex hull of each subset.
- Now combine the two hulls, this can be done by finding the upper and lower common tangent and removing the points in between the overall outline.
- This gives us the new convex hull coordinates.

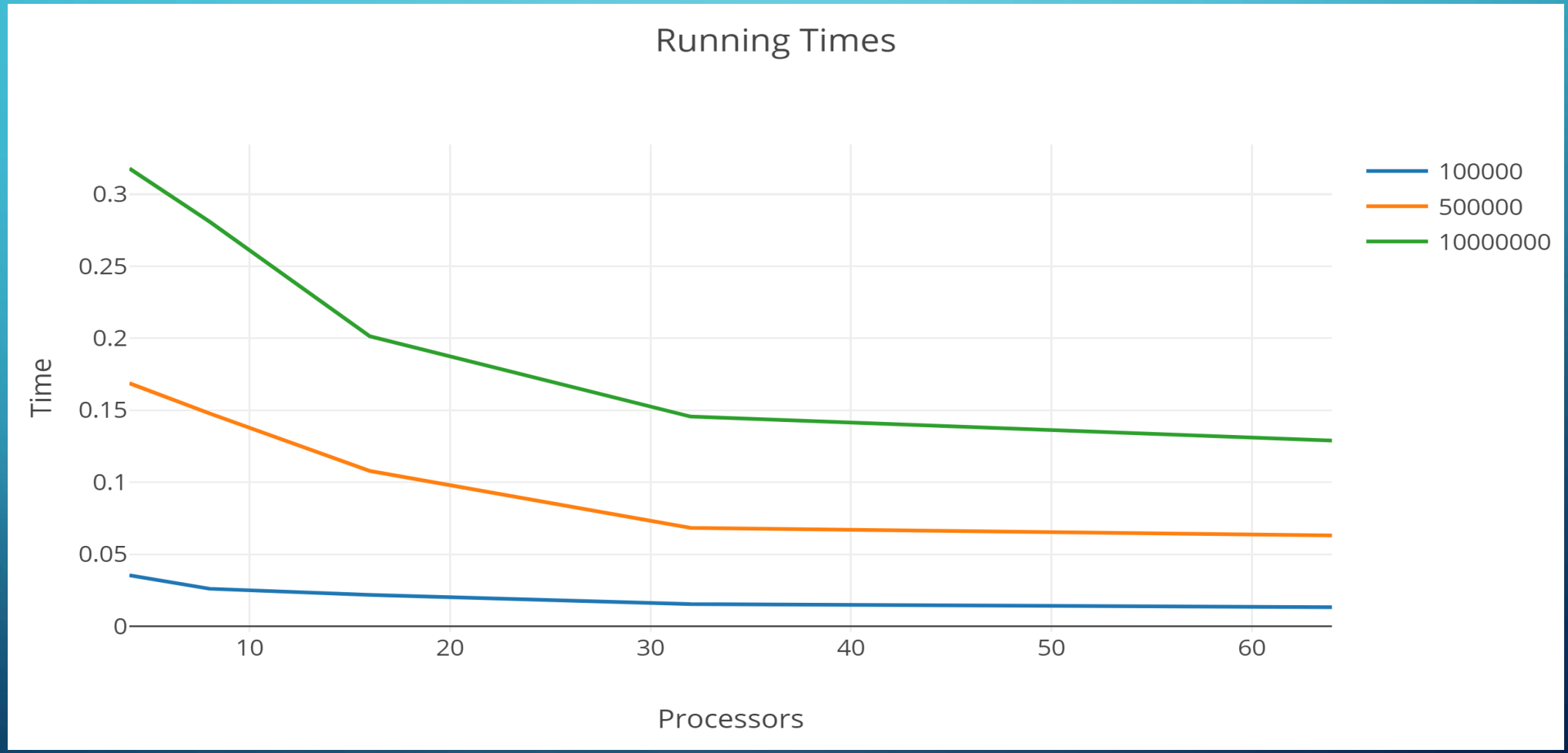
PARALLEL IMPLEMENTATION

- Each processor generates random data for the computation of the convex hull.
- Each processor parallelly computes the convex hull of the data generated.
- Now each processor will send it’s data to the next processor in power's of 2.
- After receiving the data each processors merges the hulls and sends on the new convex hull to the next processor.

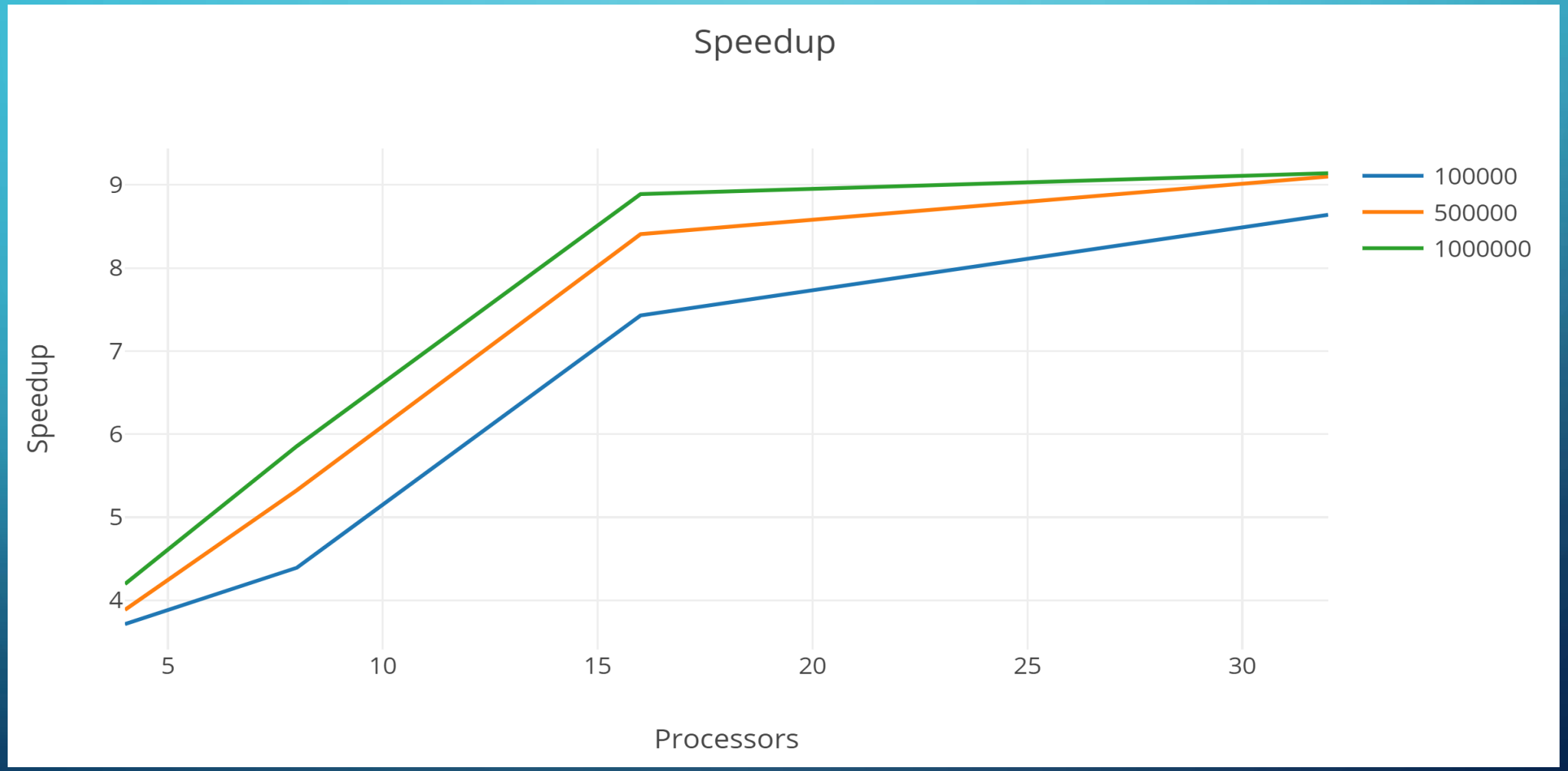
MPI COMMANDS USED

- MPI_WTIME
- MPI_SENDRECV
- MPI_TYPE_CREATE_STRUCT
- MPI_COMMIT

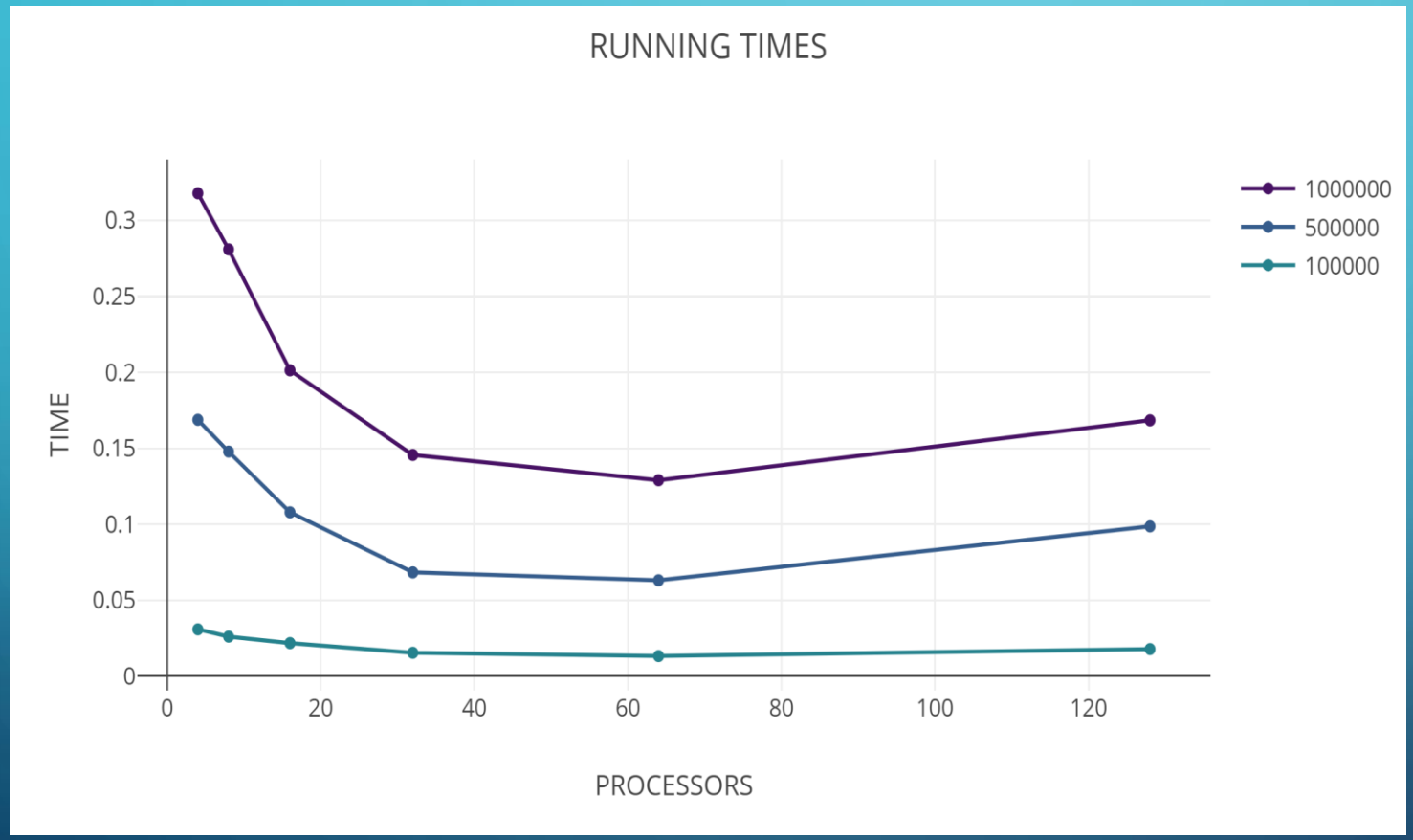
RUNNING TIME WITH READING FROM FILES



SPEEDUP WITH READING FROM FILES

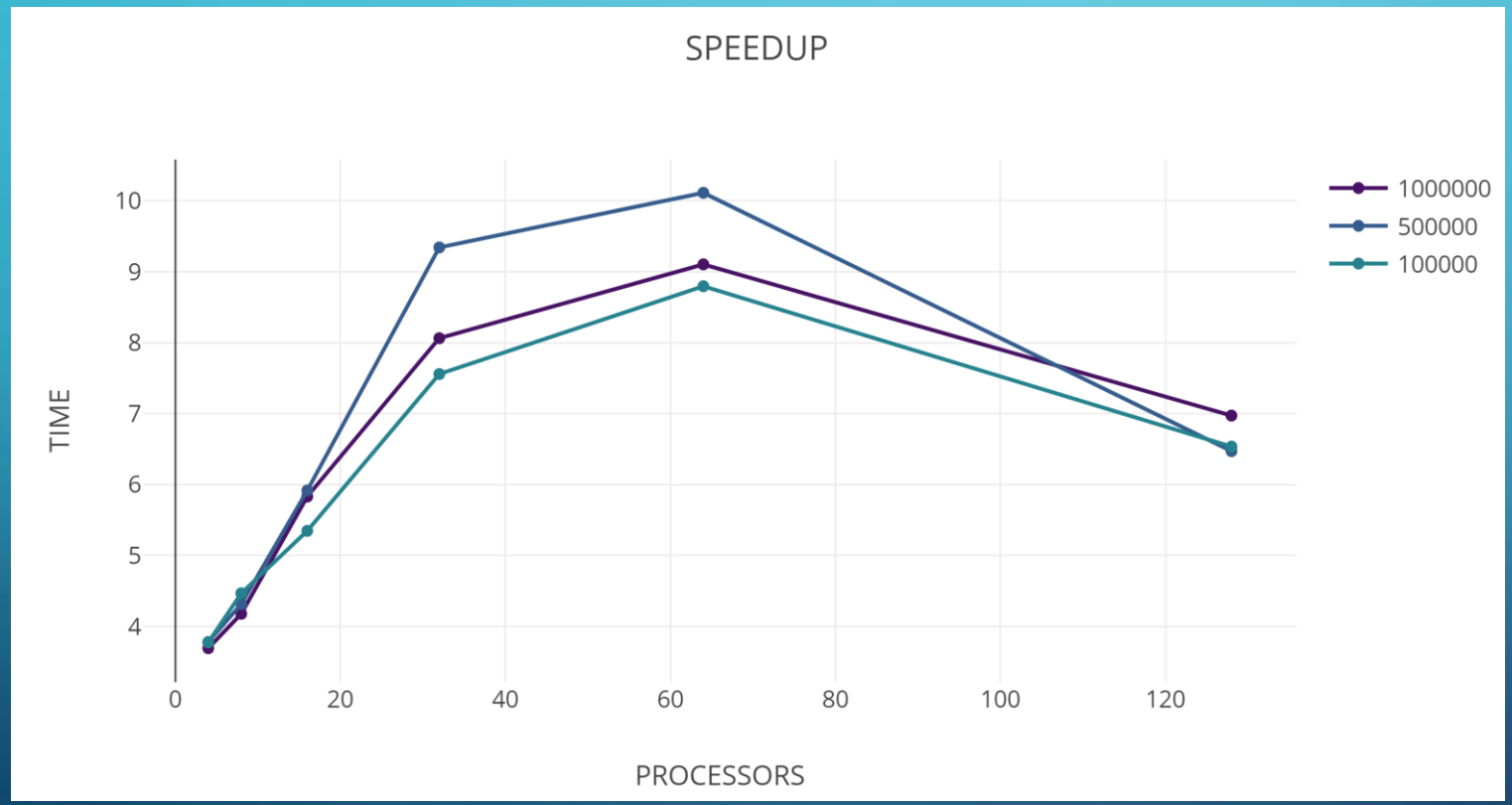


RUNNING TIME WITHOUT READING FROM FILES



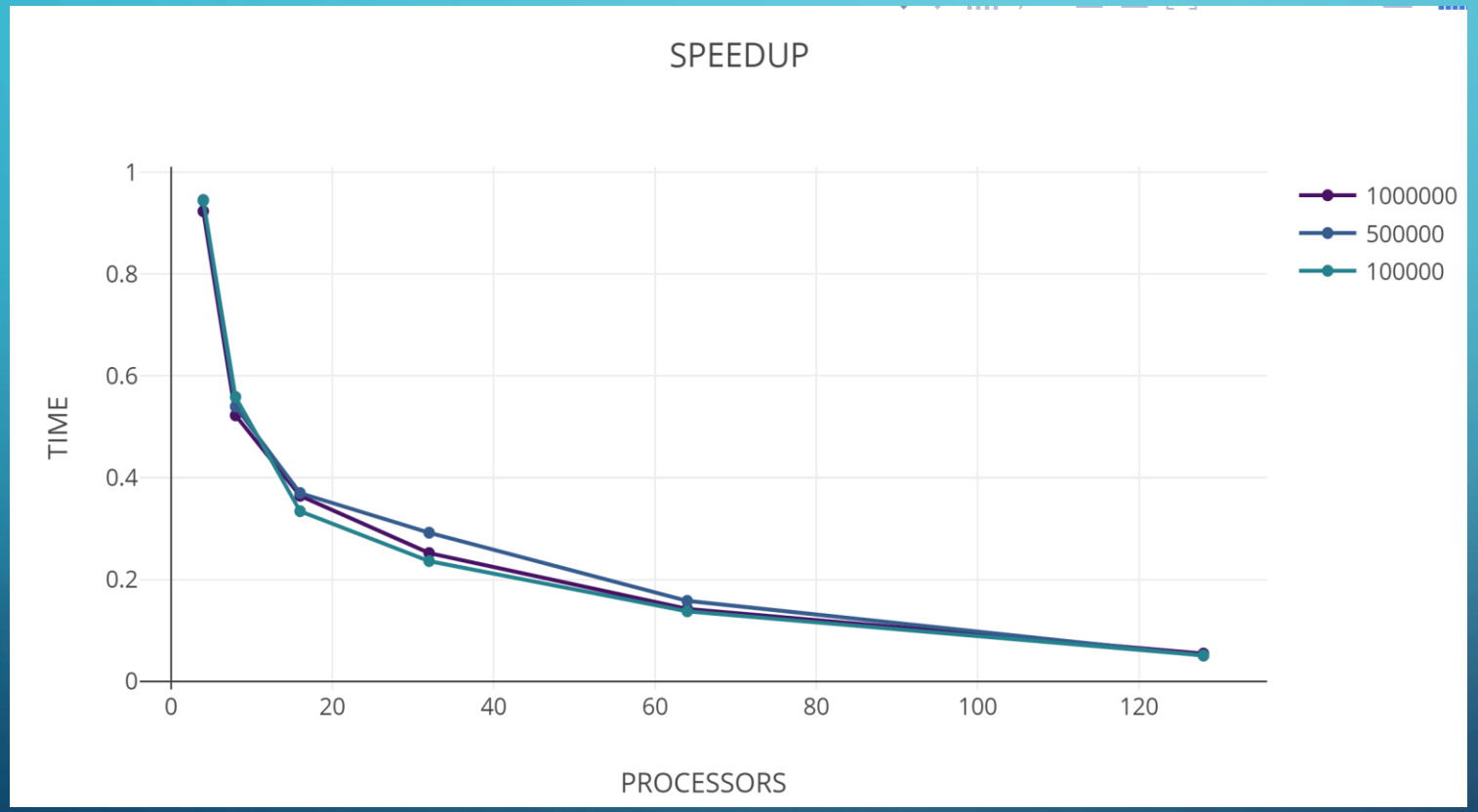
Data Size			
	100000	500000	1000000
4	0.03085	0.16876	0.31786
8	0.02608	0.14782	0.28094
16	0.02179	0.10786	0.20134
32	0.01542	0.06832	0.14563
64	0.01325	0.06312	0.12898
128	0.01783	0.09862	0.1684

SPEEDUP WITHOUT READING FROM FILES.



Data Size			
	100000	500000	1000000
4	3.777	3.781998	3.694079
8	4.4685	4.317751	4.17954
16	5.3483	5.917393	5.831926
32	7.5577	9.342067	8.062899
64	8.7954	10.11169	9.103737
128	6.5361	6.471811	6.972684

EFFICIENCY WITHOUT READING FROM FILES



Data Size			
	100000	500000	1000000
4	0.94425	0.9455	0.92352
8	0.558563	0.539719	0.522443
16	0.334269	0.369837	0.364495
32	0.236178	0.29194	0.251966
64	0.137428	0.157995	0.142246
128	0.051063	0.050561	0.054474

OBSERVATIONS

- Sequential performance was as predicted.
- Parallel Speedup was not true to prediction in all cases.
- There is an ideal no of processors to be used for which we will have the maximum benefit, after which there will be minimal decrease in the time taken for the program to run.

CHALLENGES FACED

- DATA GENERATION.
- I tried to create my own data and read from the file. It took too long to read from files, so I started generating data in each node of random number's within a given size.
- DEBUGGING PROGRAMS
- I had to write the data to a file to debug.
- MEMORY LIMITATIONS
- Sometimes I ran out of memory when using a large number of points.

REFERENCES

- Miller, Russ, and Laurence Boxer. Algorithms, sequential & parallel: A unified approach
- https://www.mpich.org/
- https://en.wikipedia.org/wiki/Convex_hull

THANK YOU