CPSC 3620
Distributed Computing
Assignment 4
September 22, 2015

Assignment Report

Team 6 -
Akshit Gupta: Coding the static_mpi.c and dynamic_mpi.c.
Steven Williamson: Coding the static_mpi.c and dynamic_mpi.c.
Rohit Gupta: Execution and Analysis of the Program.
Sameer Singh: Line & Column plotting and Assignment Report.

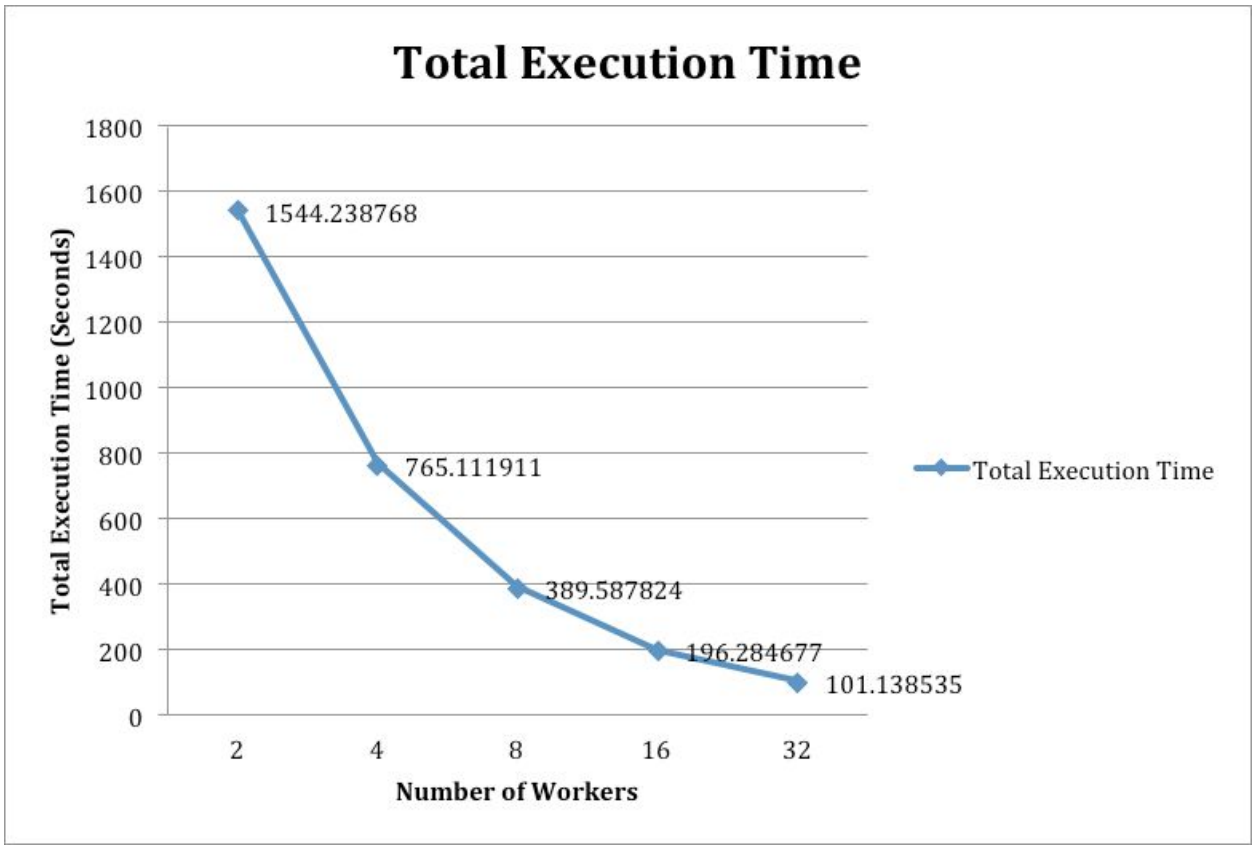
Collaboration Statement -
I worked on this assignment with Akshit Gupta, Rohit Gupta and Steven Williamson. My role in completing the assignment was making the assignment report, while Akshit Gupta, Steven Williamson and Rohit Gupta's role in completing the assignment was to code the static_mpi.c, dynamic_mpi.c and execution and analysis of the program, respectively. We consulted related material that can be found at the course website.

★ Line Plot for the total execution time as a function of the number of workers.

★ **Static_mpi.c**

Line Plot for Palmetto:

Number of Workers	2	4	8	16	32
Total Execution Time (Seconds)	1544.238768	765.111911	389.587824	196.284677	101.138535



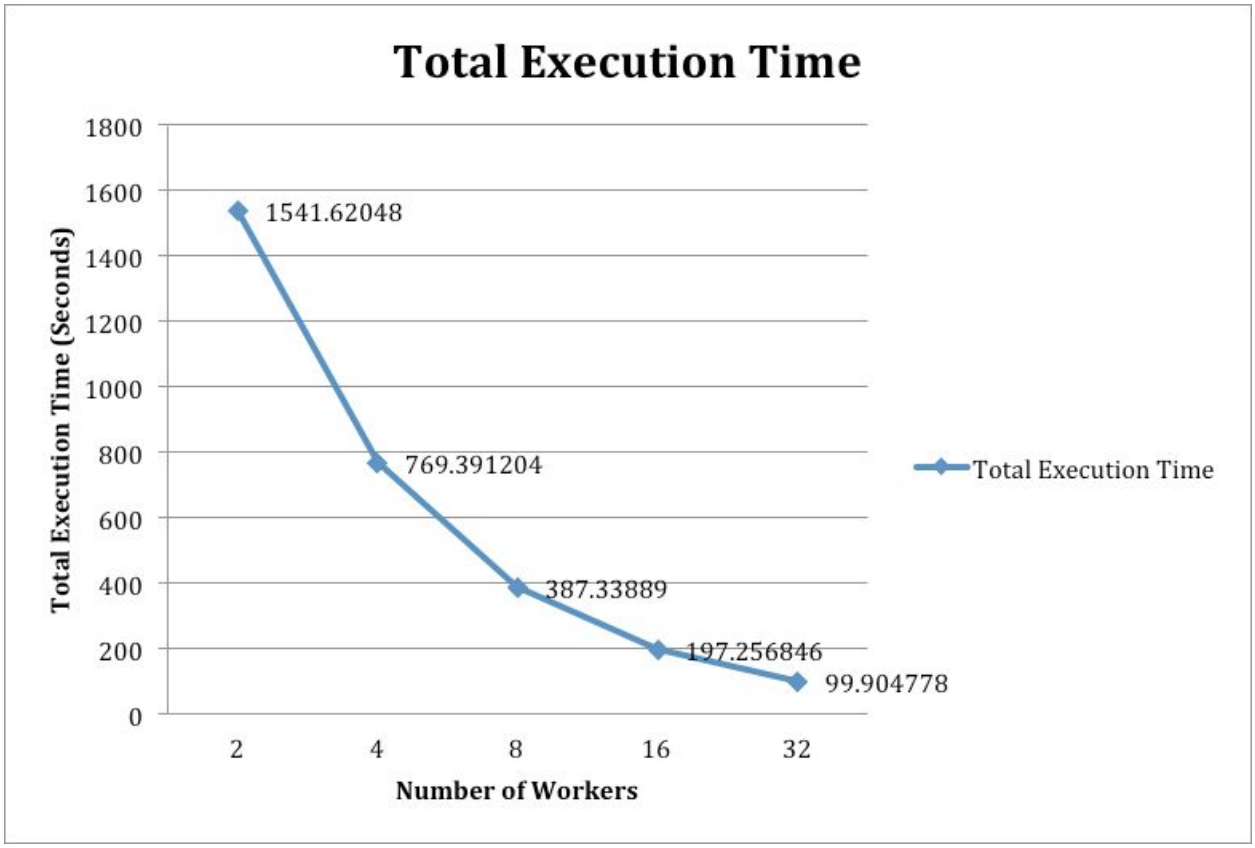
Graph 1

****Line Plot for CloudLab:**

★ **Dynamic_mpi.c**

Line Plot for Palmetto:

Number of Workers	2	4	8	16	32
Total Execution Time	1541.620480	769.391204	387.338890	197.256846	99.904778



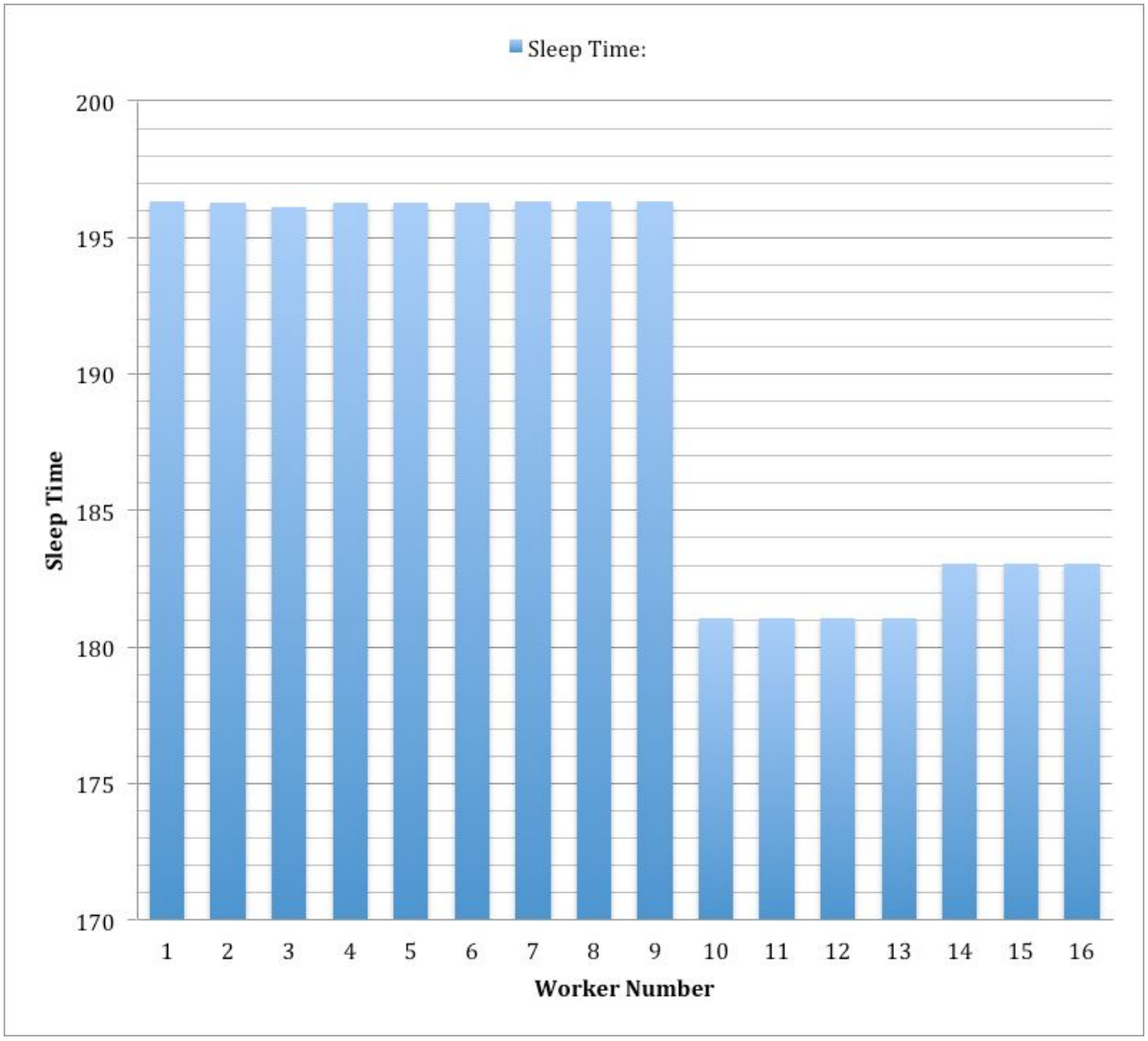
****Line Plot for CloudLab:**

★ **Column plot with each column presenting the total sleep time (processing time) for each worker for the case of 16 workers.**

★ **Static_mpi.c**

Column Plot for Palmetto:

Worker number:	Sleep Time (Processing Time):
1	196.283987
2	196.282996
3	196.112922
4	196.283264
5	196.283191
6	196.283587
7	196.283962
8	196.283922
9	196.284264
10	181.044209
11	181.044657
12	181.044443
13	181.044740
14	183.037474
15	183.037517
16	183.036801

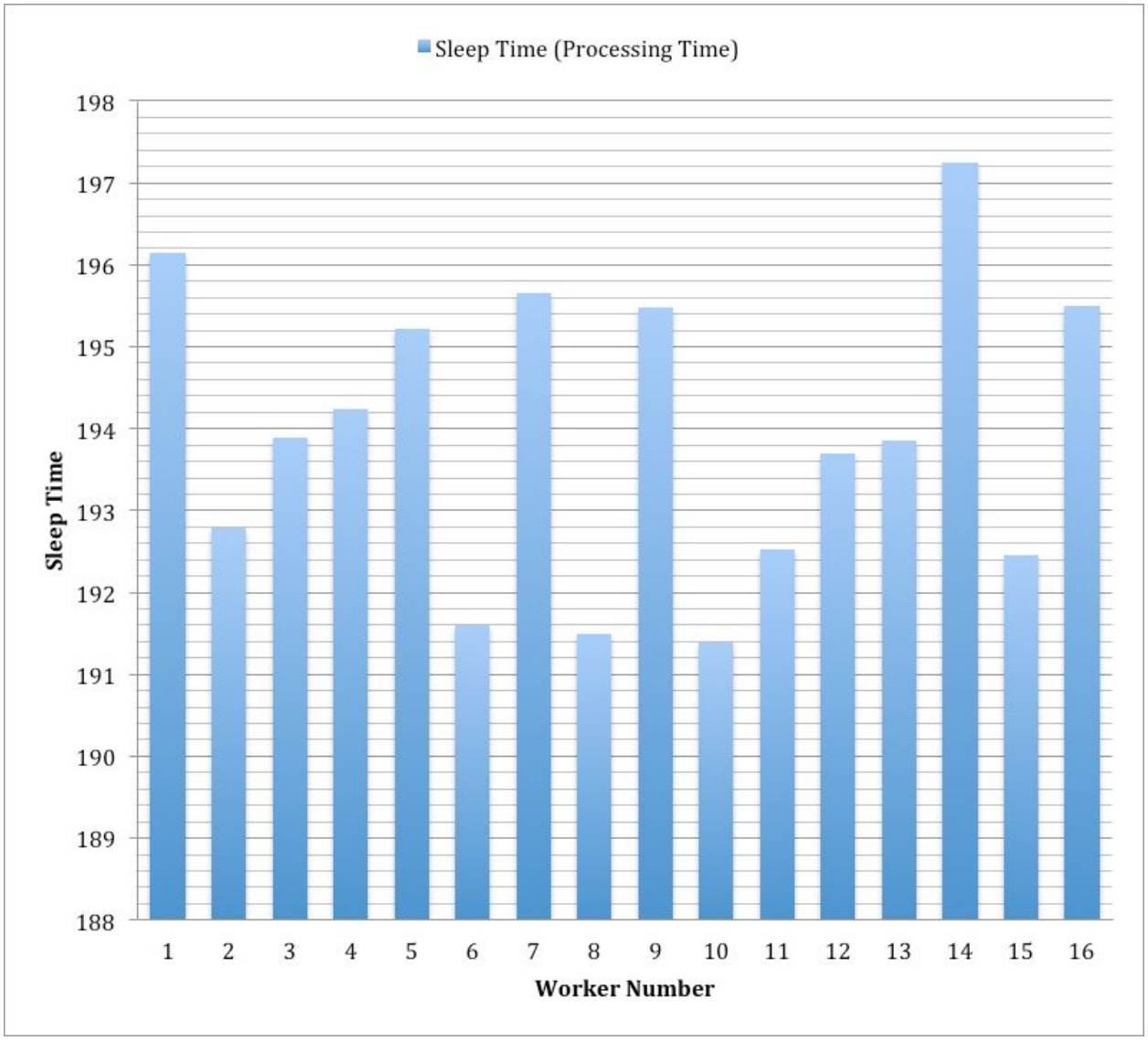


****Column Plot for CloudLab:**

★ **Dynamic_mpi.c**

Column Plot for Palmetto:

Worker Number	Sleep Time (Processing Time)
1	196.134113
2	192.783123
3	193.890252
4	194.230556
5	195.210499
6	191.601311
7	195.656081
8	191.480473
9	195.473719
10	191.376014
11	192.528673
12	193.684862
13	193.846020
14	197.249654
15	192.441388
16	195.491522



****Column Plot for CloudLab:**

★ **Calculation of Variance** for each count of workers of each program on each executing platform, the variance is in differences between total sleep time of each worker using the following formula:

$$\mathrm{Var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2.$$

with n is the number of workers, x_i is the total sleeping time of worker i, and μ is the average sleep time for all the workers.

Static_mpi.c

On Palmetto-
Calculated Variance for:
Worker count 2 - 2359750.23 seconds
Worker count 4 - 576672.024 seconds
Worker count 8 - 145153.187 seconds
Worker count 16 - 35024.3563 seconds

On Cloudlab-
Calculated Variance for:
Worker count 2 -
Worker count 4 -
Worker count 8 -
Worker count 16 -

Dynamic_mpi.c

On Palmetto-
Calculated Variance for:
Worker count 2 - 2359155.03 seconds
Worker count 4 - 581004.381 seconds
Worker count 8 - 145390.296 seconds
Worker count 16 - 36450.46475 seconds

On Cloudlab-
Calculated Variance for:
Worker count 2 -
Worker count 4 -
Worker count 8 -
Worker count 16 -

★ Discussion about which workload balancing program

Workload balancing is used to distribute computations fairly across processors in order to obtain the highest possible execution speed. Imperfect load balancing leads to increased execution time.

The **dynamic_mpi** is the dominant program on the Palmetto supercluster than the dynamic_mpi because it takes advantage of faster IPC than CloudLab provides. It is the best minimizing the total work of the program. The **dynamic_mpi** is best at taking advantage of the sleep time. It keeps all processes busy until the entire workload is complete and then will finish with the remaining processes. This will result in some variance in the sleep time of the processes but will generally be lower execution time. While some processes will have more execution than others, it is the most efficient in the end.

The **static_mpi** is stronger on the CloudLab cluster because there is minimal communication and since the CloudLab node communication time will be slower on Palmetto, the **dynamic_mpi** will suffer from this. Likewise, with CloudLab having more communication time between nodes because some nodes are located in different locations, the **dynamic_mpi** will not be good on the CloudLab cluster. The **dynamic_mpi** will excel on the Palmetto cluster because it's communication takes little time and it optimizes the use of the processes and the sleep time. The **static_mpi** will divide the work evenly but will not divide the processing time perfectly. This happens because of the variation between process time cannot be accounted for in the **static_mpi**.

The **static_mpi** implementation is stronger than a usual implementation. It divides the number of type 0, 1, and 2 workloads evenly between all processes. This is different from dividing the total workload array evenly and distributing partitions without sorting the workload types. **Static_mpi** attempts to evenly distribute the processing time between the processes. Because of this enhanced implementation, the difference between the **dynamic_mpi** and **static_mpi** is slim. The **dynamic_mpi** is a much more simple and natural implementation and would be easier to support in the future. Therefore, **dynamic_mpi** makes the best overall implementation.

Based off the measurements, the **dynamic_mpi** performed better overall, while the **static_mpi** performed better at making the best use of all processors.