J.S. Arukgoda
100040X
CS 4532
Lab 1

Step 2

Case 1 :

Implementation	1 Thread		2 Threads		4 Threads	
	Avg	SD	Avg	SD	Avg	SD
Serial	0.0267	0.0015				
One Mutex	0.0316	0.0037	0.0471	0.0126	0.0436	0.0122
Read – Write Lock	0.0312	0.0038	0.0199	0.0030	0.0213	0.0049

Case 2 :

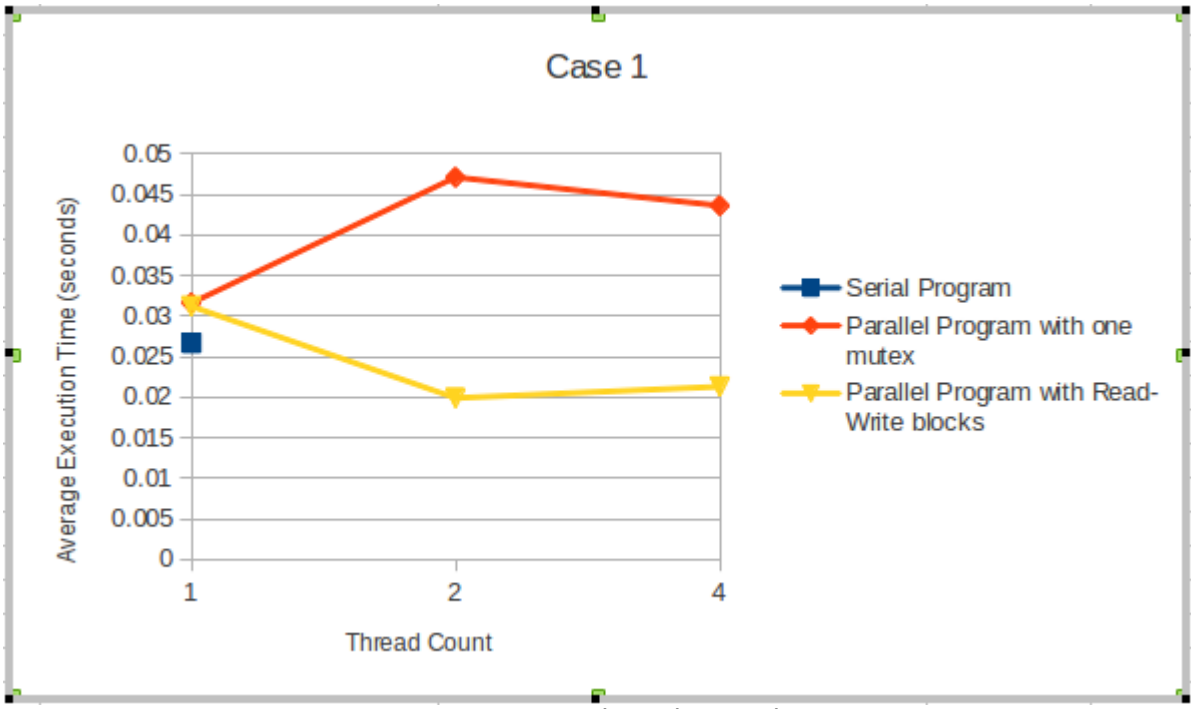
Implementation	1 Thread		2 Threads		4 Threads	
	Avg	SD	Avg	SD	Avg	SD
Serial	0.0380	0.0023				
One Mutex	0.0420	0.0040	0.0559	0.0139	0.0554	0.0120
Read – Write Lock	0.0428	0.0042	0.0326	0.0064	0.0317	0.0060

Case 3 :

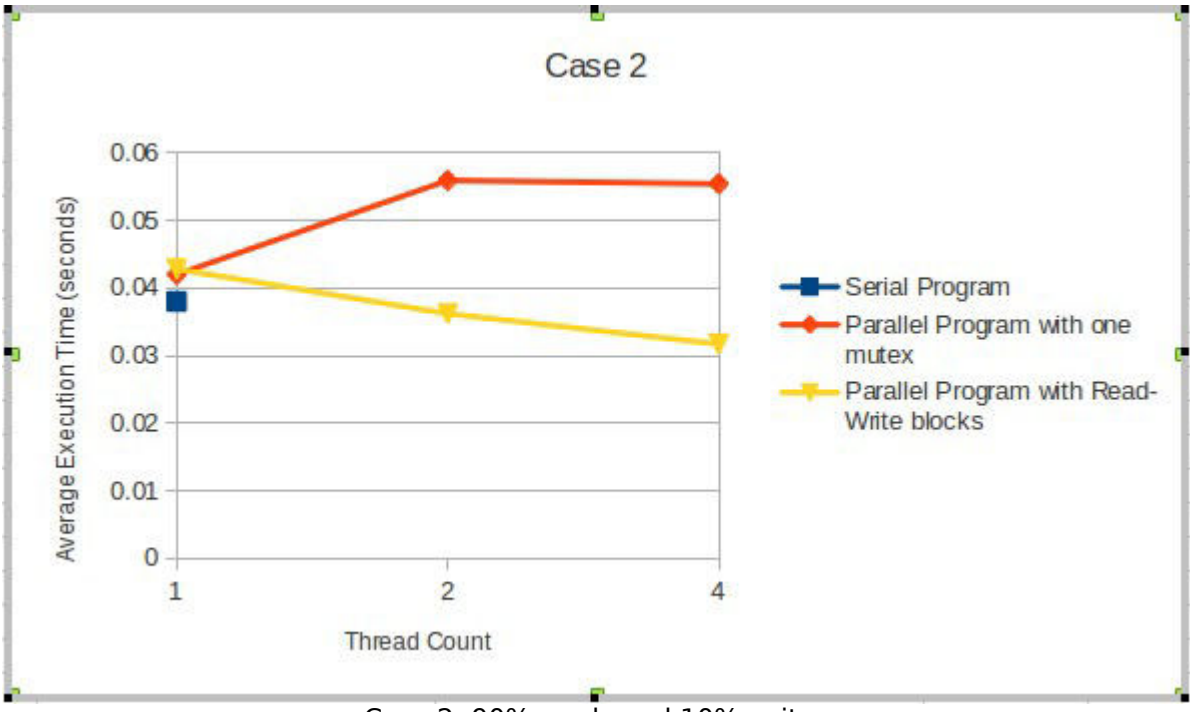
Implementation	1 Thread		2 Threads		4 Threads	
	Avg	SD	Avg	SD	Avg	SD
Serial	0.0793	0.0038				
One Mutex	0.0856	0.0058	0.1013	0.0167	0.0768	0.0131
Read – Write Lock	0.0865	0.0060	0.0945	0.0129	0.0761	0.0101

Step 3

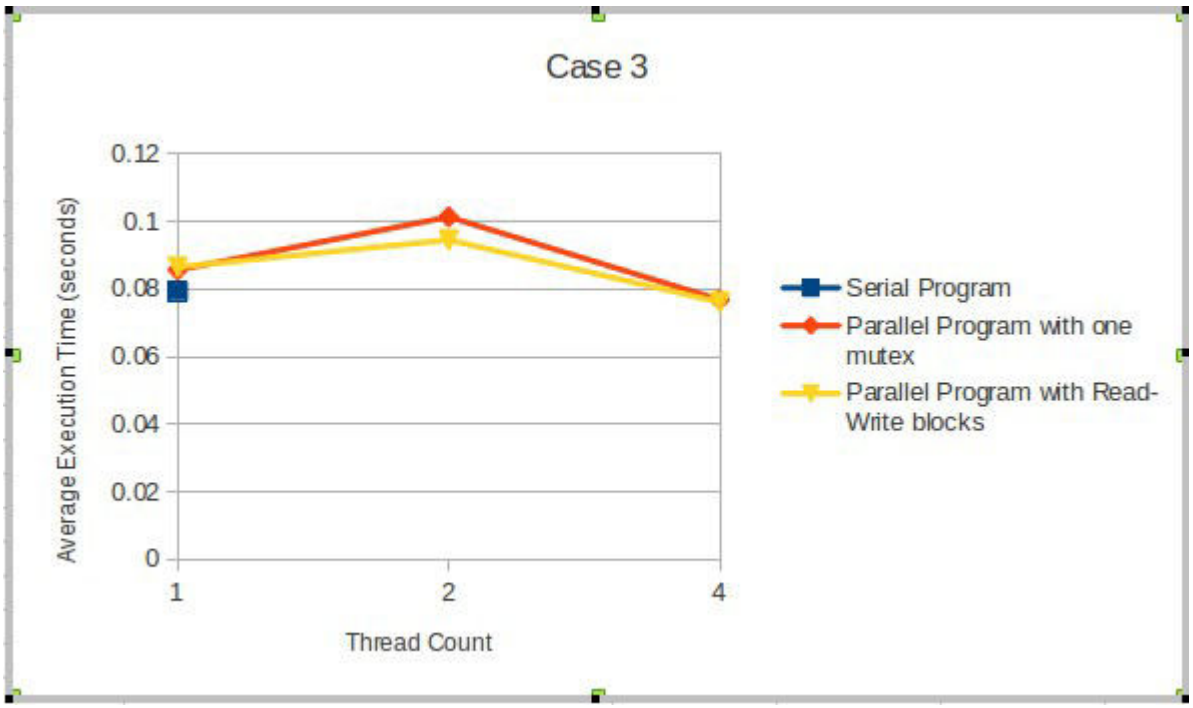
Average execution time is plotted against the number of threads.



Case 1: 99% reads and 1% writes



Case 2: 90% reads and 10% writes



Case 3: 50% reads and 50% writes

Step 4

Observations :

- Case 1 contains 99% read operations and 1% write operations. Case 2 contains 90% read operations and 10% write operations. Case 3 contains 50% read operations and 50% write operations. It is observed that when the portion of write operations increase (Inserts and Deletes), the operating time increases with it regardless of the implementation. Therefore it can be concluded that write operations are more expensive than read operations.

- In the single threaded implementations of all three cases, the serial program performs better than parallel programs. This can be explained by the overhead of parallel programming. Running a single thread in a multi-threaded environment is more expensive than in a single-threaded environment.

- Parallel Programming with one mutex for the entire linked list performs worse than the serial program except for 4-thread instance of case 3. Since the use of a single mutex for the entire linked list restricts the access to the linked list for only one thread at a time. threads have to go through the overhead of resource sharing and the advantage of multi-thread execution is lesser than that overhead.

- Parallel Programming with Read-Write lock performs better than both the serial program and the parallel program with one mutex when the read

operation portion is larger than that of write operations. This is due to the ability of multiple threads to access the linked list during the read operations (Member).

System Specifications :

- Intel Core i5 processor 430M (2.26 GHz, 1066 MHz FSB, 3072 KB x 4 Cache)
- 4 GB DDR3 memory (1067 MHz)
- Ubuntu 12.04 32-bit, Kernel Linux 3.2.0-55-generic-pae, GNOME 3.4.2