EMPIRICAL ANALYSIS

Locking Schemes implemented in Part 1 :

1. Spin Lock
2. Sequential locks
3. Read-write lock
4. RCU
5. Custom Read-write

For the empirical comparison of all the locking schemes, these are the parameters :

1. No. of threads
2. No. of operations per thread
3. Percentage of read operations

Experimental Setup :

- So for this analysis keeping No. of threads and No. of operations per thread constant for each of the locking schemes and varying the percentage of read operations to observe the behaviour of each locking scheme.
- No. of CPUs = 3
- Operating System : Ubuntu (64-bit)
- RAM = 4 GB
- No. of Threads = 10
- No. of operations per thread = 5000000
- Percentage of read operations = ? (They'll vary!)

Observations :

1. **SPINLOCK**

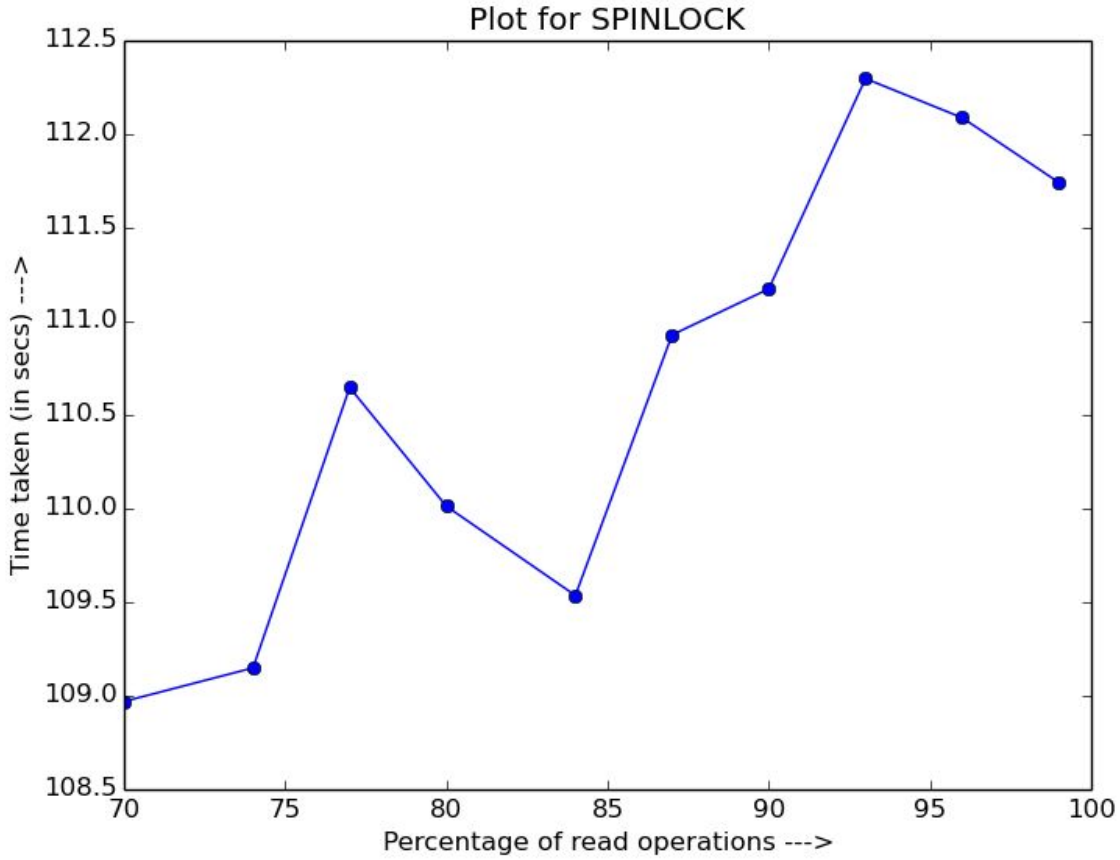
(%) Read Ops	Time Taken (in Secs)
70	108.96971082687378
74	109.14900493621826
77	110.64693355560303
80	110.01253867149353
84	109.53605699539185
87	110.92817759513855
90	111.17355728149414
93	112.29709148406982
96	112.0876796245575
99	111.7416064739227

Total time for all operations (in secs) - 1106.5423574447632

Avg Time (in secs) - 110.6542357444763

Variance (in secs^2) - 1.3046346592563303

Standard Deviation (in secs) - 1.142206049387031



Inference :

As the percentage of read operations increases more the time taken increases for Spinlock.

2. SEQUENTIAL LOCKS

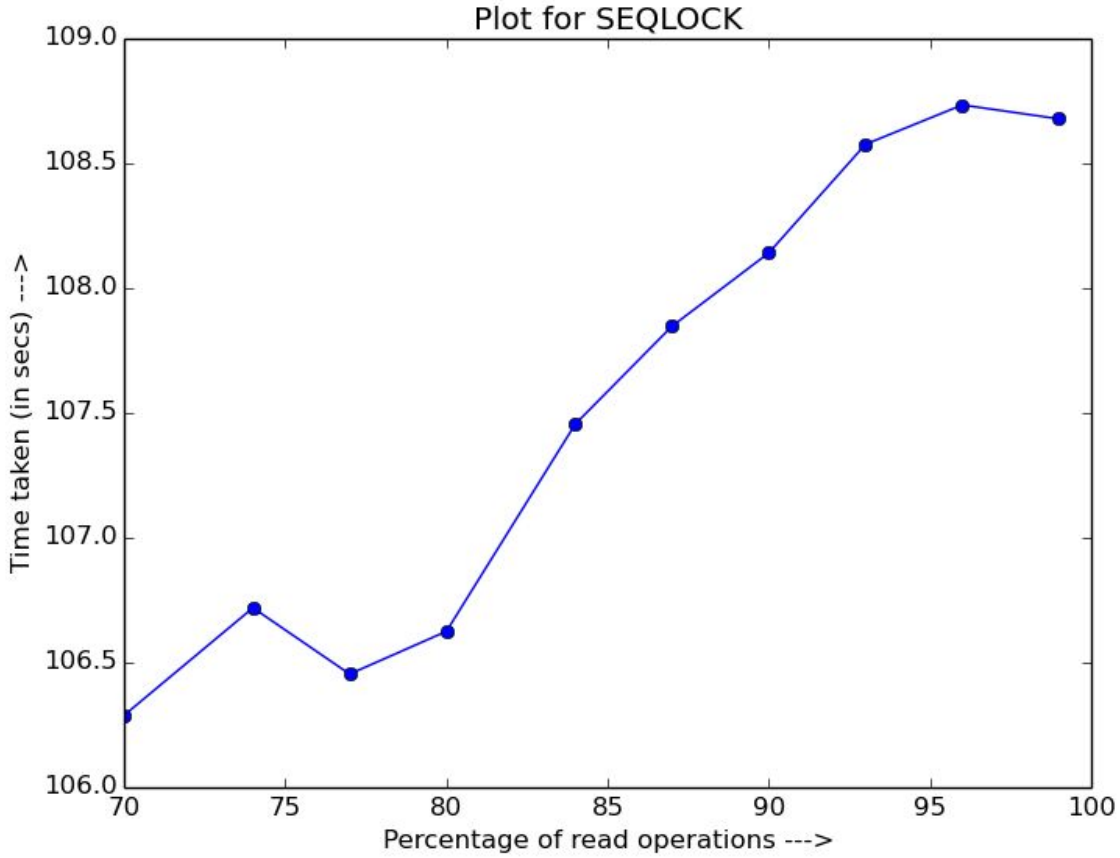
(%) Read Ops	Time Taken (in Secs)
70	106.28844332695007
74	106.71938347816467
77	106.4540331363678
80	106.62510561943054
84	107.45589303970337
87	107.84782147407532
90	108.14147925376892
93	108.57653045654297
96	108.73416519165039
99	108.6784365177154

Total time for all operations (in secs) - 1075.5212914943695

Avg Time (in secs) - 107.552129149436

Variance (in secs^2) - 0.85147591061543

Standard Deviation (in secs) - 0.92275452348684



Inference :

As the percentage of read operations increases more the time taken increases for Seqlock.

3. Read-Write Lock

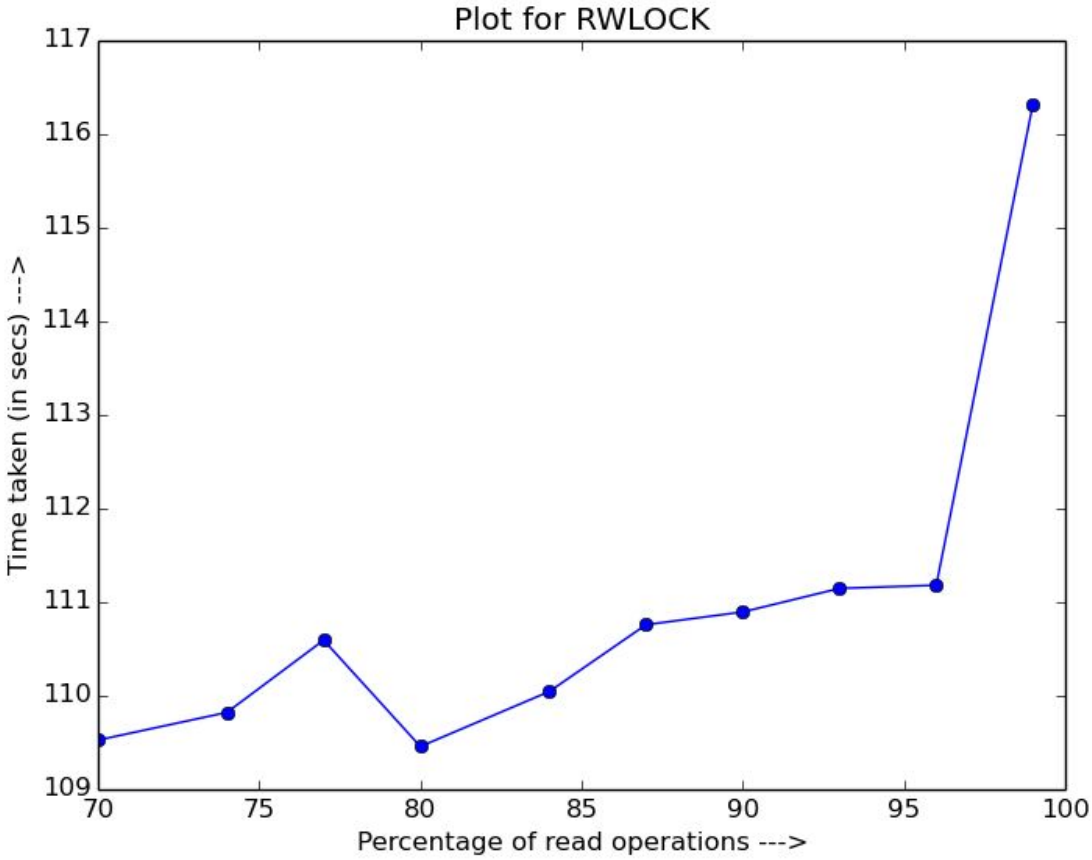
(%) Read Ops	Time Taken (in Secs)
70	109.52646851539612
74	109.82247400283813
77	110.59447121620178
80	109.4583523273468
84	110.04602003097534
87	110.75957417488098
90	110.89567041397095
93	111.14750814437866
96	111.18127846717834
99	116.3210074901580

Total time for all operations (in secs) - 1109.7528247833252

Avg Time (in secs) - 110.97528247833252

Variance (in secs^2) - 3.5439392359267

Standard Deviation (in secs) - 1.88253532129


Plot for RWLOCK
Time taken (in secs) --->
Percentage of read operations --->
117
116
115
114
113
112
111
110
109
70
75
80
85
90
95
100

4. RCU

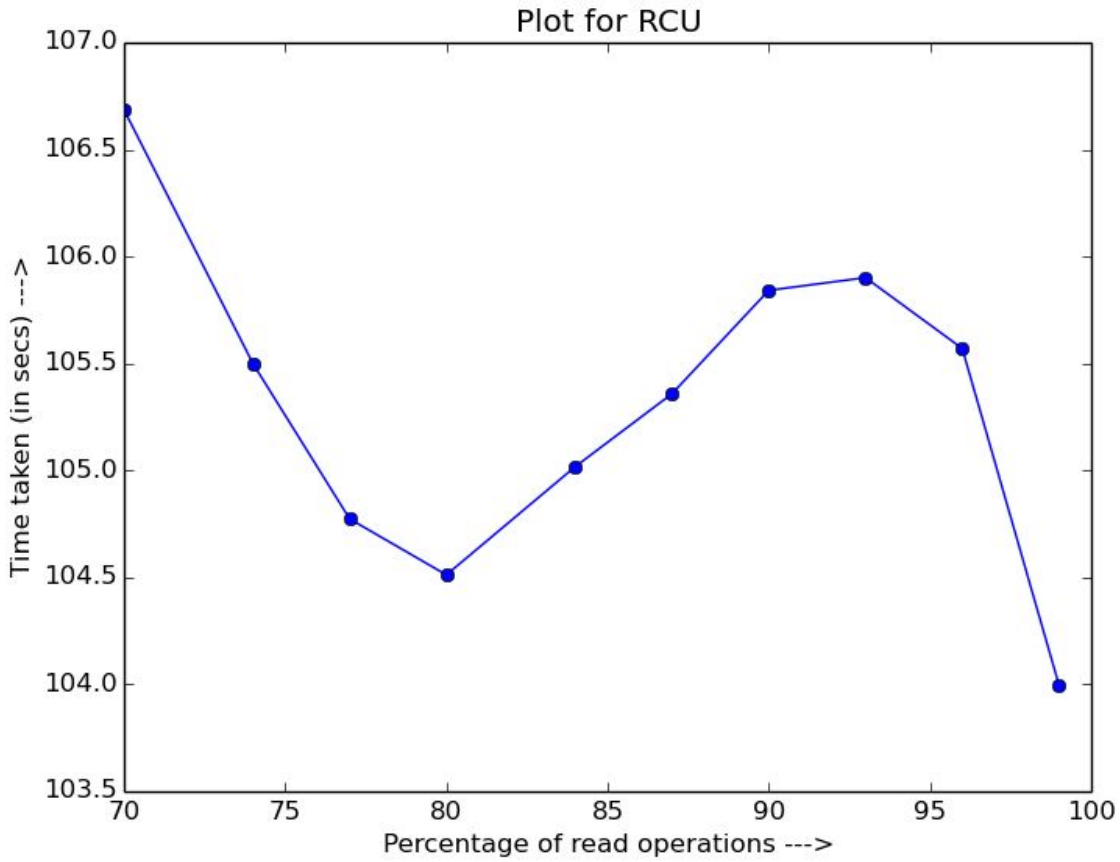
(%) Read Ops	Time Taken (in Secs)
70	106.6860454082489
74	105.49990487098694
77	104.7736406326294
80	104.51208758354187
84	105.01820158958435
87	105.35898351669312
90	105.8419599533081
93	105.90128803253174
96	105.57035899162292
99	103.9979147911071

Total time for all Operations (in secs) - 1053.160385370

Avg Time (in secs) - 105.3160385370

Variance (in secs^2) - 0.5363048808079389

Standard Deviation (in secs) - 0.7323283968329638



The above graph shows the decrease in time taken to execute 10 threads when percentage of read operations increase while using RCU lock but only till a percent of read operations, then it again starts to increase and decrease again after a certain percetage.

5. CUSTOM Read-Write Lock

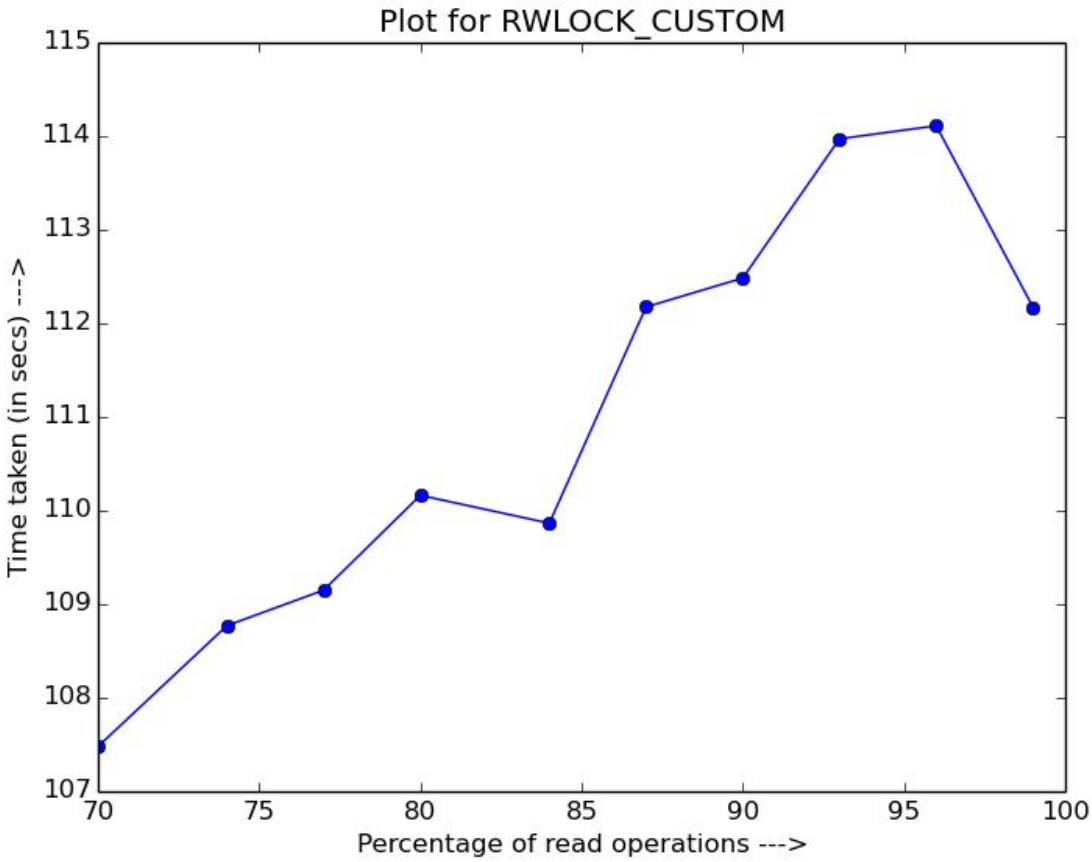
(%) Read Ops	Time Taken (in Secs)
70	107.47928953170776
74	108.77041912078857
77	109.14989399909973
80	110.16282439231873
84	109.86498785018921
87	112.17513275146484
90	112.48456954956055
93	113.9715142250061
96	114.11097526550293
99	112.1747448444366

Total time for all Operations (in secs) - 1110.344351530075

Avg Time (in secs) - 111.03443515300751

Variance (in secs^2) - 4.6239556166305

Standard Deviation (in secs) - 2.15033848885



Inference :

As the percentage of read operations increases more the time taken increases for Custom Read write lock, but the time taken also starts to decrease after a certain percentage of read operations.

Comparing RWLOCK and Custom RWLOCK

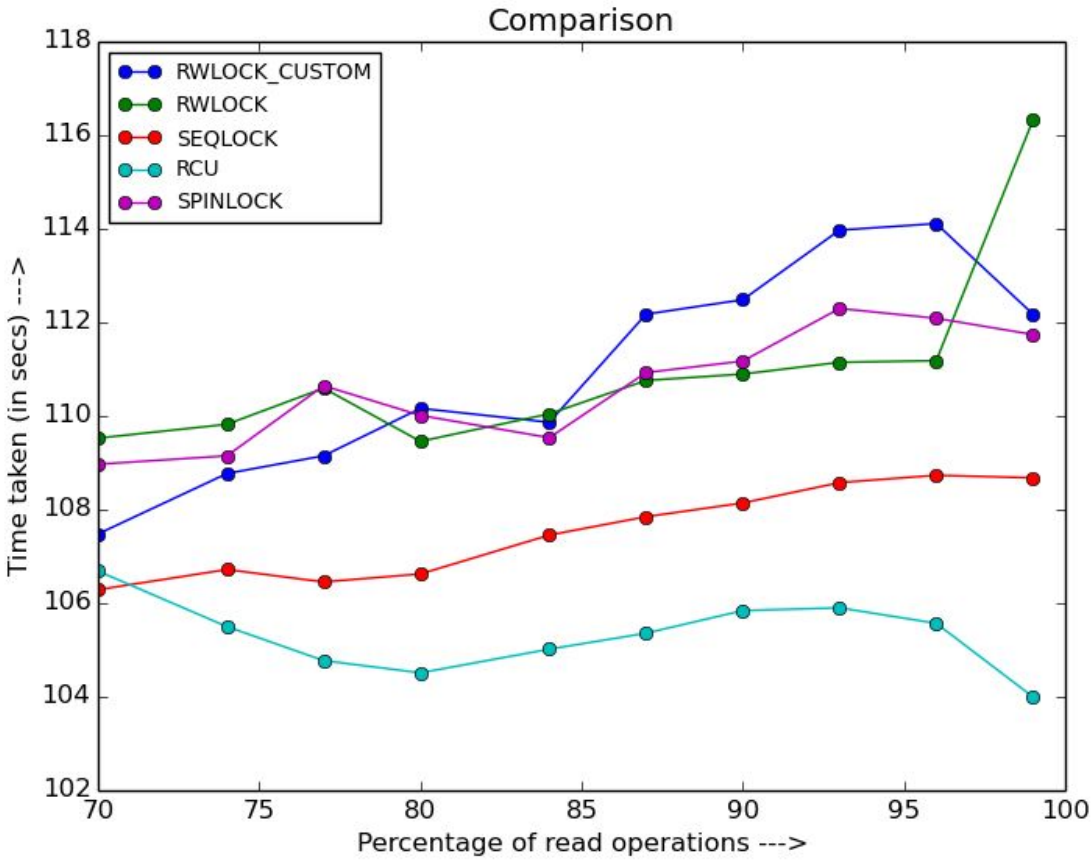
RWLOCK		**CUSTOM RWLOCK**	
(%) Read Ops	Time Taken (in Secs)	(%) Read Ops	Time Taken (in Secs)
70	109.52646851539612	70	107.47928953170776
74	109.82247400283813	74	108.77041912078857
77	110.59447121620178	77	109.14989399909973
80	109.4583523273468	80	110.16282439231873
84	110.04602003097534	84	109.86498785018921
87	110.75957417488098	87	112.17513275146484
90	110.89567041397095	90	112.48456954956055
93	111.14750814437866	93	113.9715142250061
96	111.18127846717834	96	114.11097526550293
99	116.3210074901580	99	112.1747448444366

CUSTOM RWLOCK Avg Time (in secs) = 111.034435153007

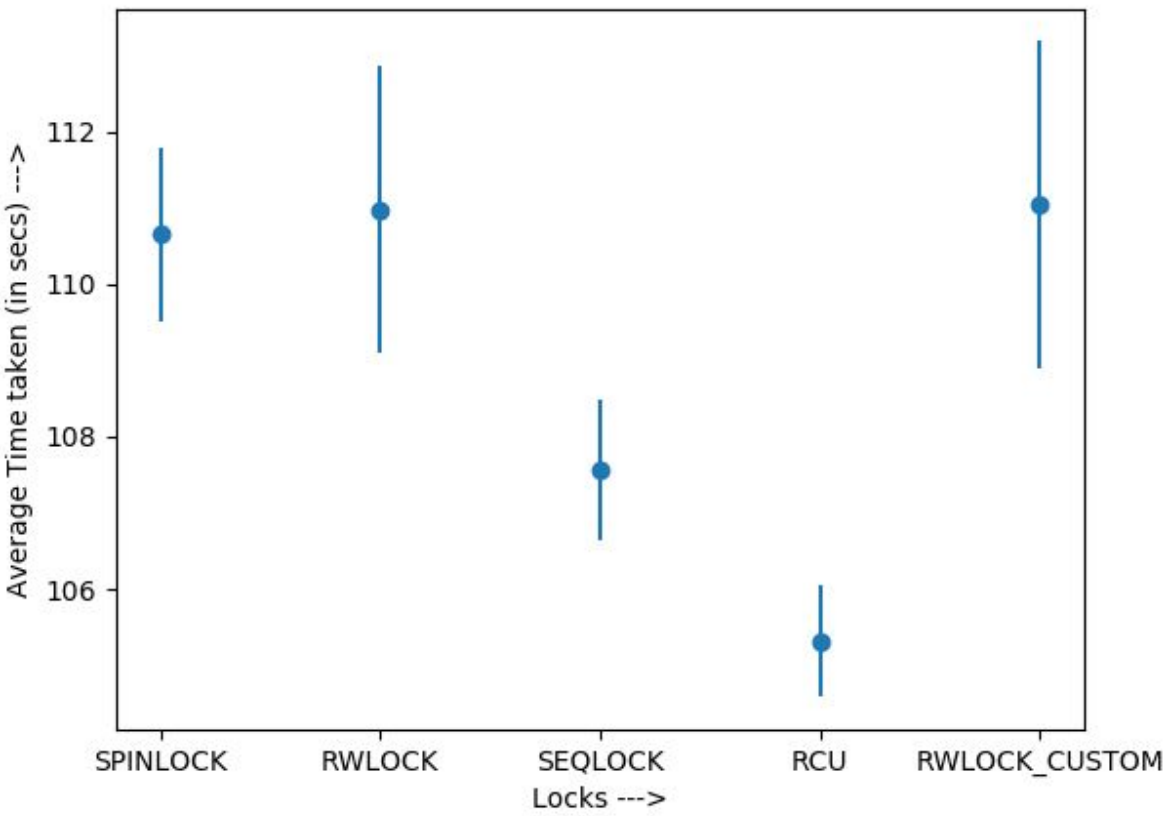
RWLOCK Avg Time (in secs) = 110.9752824783

So the Custom RWLOCK implemented is **slower** than the RWLOCK provided by linux kernel API.

Comparison :



This plot shows how all the locking schemes together vary in time taken when the percentage of read operations increase.



This plots the Average time taken by each lock along with the deviation.

Results Insights :

Fastest Lock among all these - RCU
Its average time taken (in secs) is - 105.31603853702545

Slowest Lock among all these - RWLOCK_CUSTOM
Its average time taken (in secs) is - 111.03443515300751