Experiment Setup:

- KVM hypervisor, virt-manager, Ubuntu 16.04, 2GB RAM, 2 CPUs
- Avg workload:
  - CPU1: ~15%
  - CPU2: ~15%
  - MEM: ~50%
- We have used following set of parameters for comparison
  - % read=100, threads=8, ops/thread=50000
  - % read=99, threads=8, ops/thread=50000
  - % read=95, threads=8, ops/thread=50000

Result:

- Spinlock:
  - As the %write increases, avg time remains statistically same. This is because the spinlock does not differentiate between read and write operations
  - Avg deviation also does not provide any pattern because of the same reason
- RWlock:
  - For no writer case, avg time is very less as all readers can read simultaneously. Writers may starve if there are many-many readers therefore consuming time. But more no. of readers means saving of time because of parallel readings.
  - Avg deviation is large for the case when possibility of writer starvation is higher. Hence, it has high average deviations.
- Seqlock:
  - As the no. of writers increases, avg time taken increases. This is because writer can write without any constraint causing readers to read data again and again.
  - This lock solves the writer starvation. Hence, lesser avg deviation
- RCU lock:
  - Very efficient lock for the case where no. of writers are minimal. But with the increase of %write ops, avg time taken increases with the positive second derivative
  - Avg deviation is large because of allocating and copying the data again and again
- Custom RWlock:
  - Avg. time increases with the increase of writers
  - Writer may starve hence large avg deviation
- For the no/few writers case, RCU lock is most efficient
- As the no. of writers increases up to a certain point, Seqlock becomes most efficient avoiding writers starvation

Insights:

- RCU lock becomes exponentially inefficient as the no. of % writes increases. This is because of the fact that, in write call, RCU copies the shared data to a new location with large overhead subjected to the size of shared data.
- Spinlock is most fair lock as it does not differentiate between writers and readers
- Seqlock is the most unfair lock as it allows every writer to write whenever they deem necessary causing the active readers to reread the data again