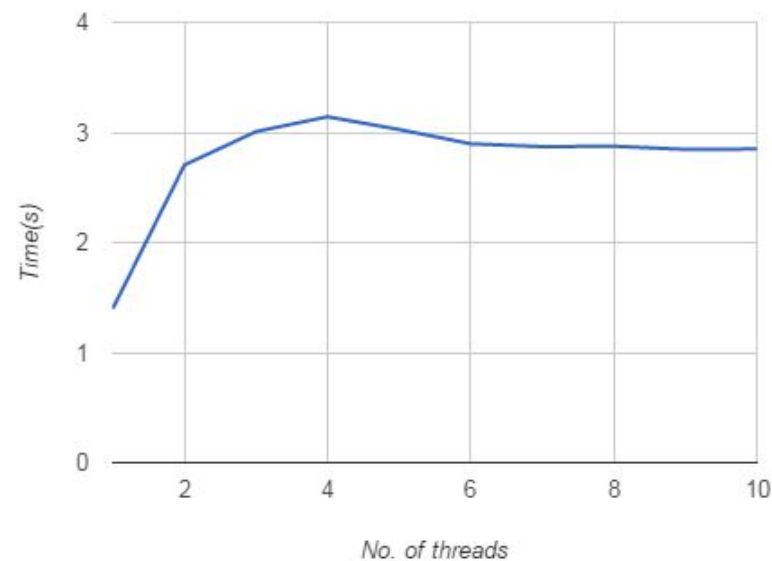


Plots for counter 1 and 2, and final counter value

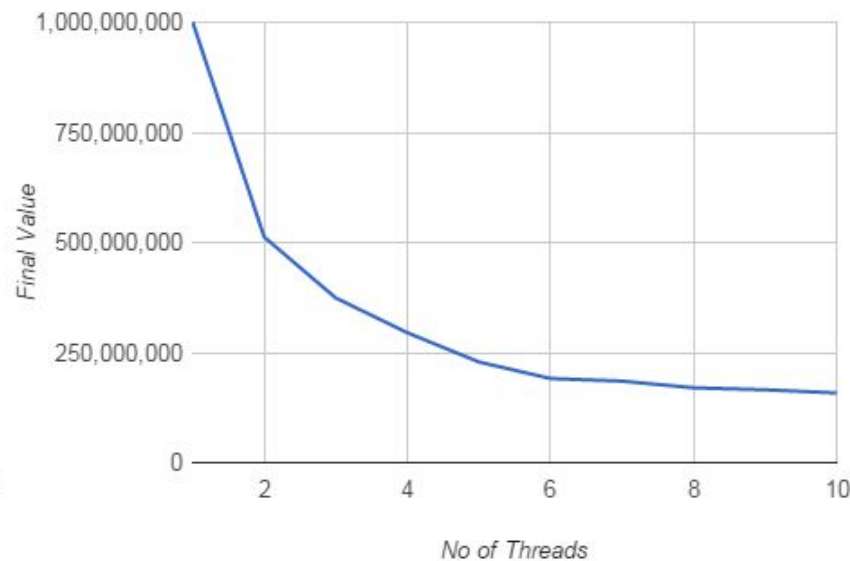
Palash Bansal, 2014072

Counter 1 - Non Safe

Non-Safe Counter Performance

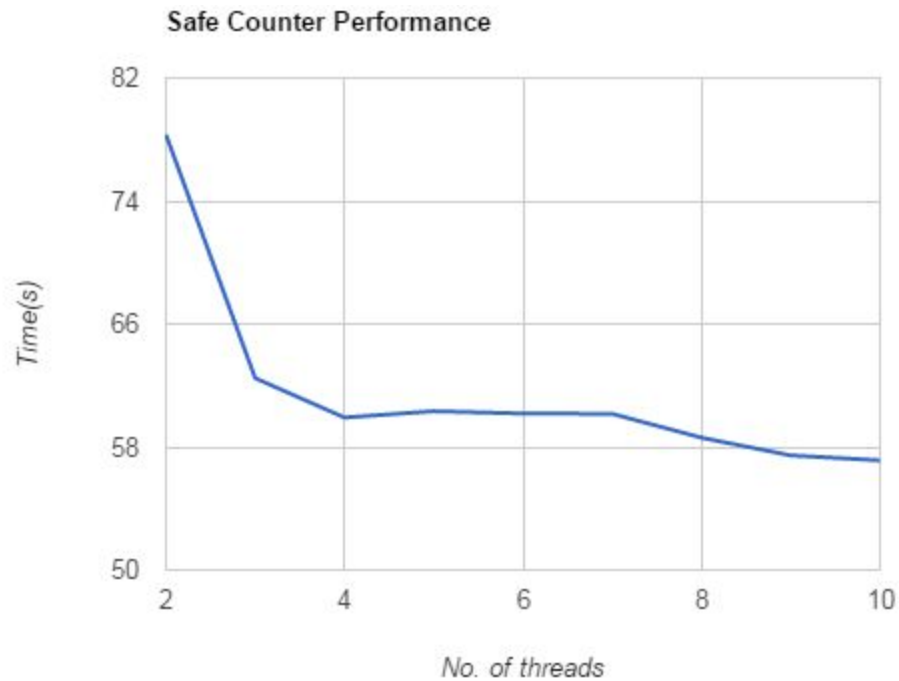


Non-Safe Counter - Final Value



- In the performance graph, it can be seen that, as we increase the number of threads, the performance decreases at first, then increases and remains almost constant from thread 6 onwards, which is also the no of cores in the VM the program is tested in. The performance decreases at first because we are writing to the same variable in all the threads and so it takes more time, but as we continue to increase the no of threads, the performance(comparatively) increases, as it is using more cores, and the overhead of writing to same location decreases a bit.
 - In the correction vs threads graph, it can be seen that the final value constantly decreases, and hence the error increases, but also the difference between errors in consecutive threads decreases as no of threads increases, this is because the extra thread will only contribute to a small percentage of the error. eg. from 9 -> 10 the error increased will be $1/10$, but in 4->5 the error increased is $1/5$.
-

Counter 2 - Safe



- It can be seen in the graph that the time decreases with the number of threads, i.e the performance increases with the increase in threads, though the time is still a lot more than it takes with 1 thread. The simple reason for the increase in performance is that as we increase the number of threads, more CPU is being used and the overhead because of locks decreases, and so there is an increase in performance.
- For the correctness, the final value that is obtained is correct, and is dot 1 billion in every case, the value is sometimes a slight less than 1 billion because of integer division of work between the threads, but we can say that the value obtained is always correct.