2c) Discuss your observations from the plot. Explain to what extent the increase in the number of threads improves the performance, and why the run time does not show significant decrease after reaching a certain number of threads.

Ans) In the mentioned plot, initially the execution time decreases suddenly with more parallel threads but the change in decrement keeps reducing till number of threads equals around 11. After that, the performance saturates with an increase in the number of threads. The run time does not show significant decrease after a certain number of threads cause the program is not memory bound but rather compute bound. With more increase in number of threads, the threads run in multiple cores in different NUMA nodes which might be far from where the data is allocated inside a particular cache within a NUMA node. The memory access time for such data induced by NUMA hops doesn't scale well with increase of parallel threads.

[Extra Credit] 2d)

Ans) To optimize the program mentioned in task2, the change is made to run parallel threads closer to the cores which are closer to the NUMA node where the data is allocated. Since the parallel threads are no longer spreads across multiple cores and sockets, due to the concept of data affinity, the memory access time for such data operation decreases which decreases the run time of the program.

The following changes are made to the environment variables in the script as follows:

```
export OMP_PROC_BIND = close
```

The change in run time is also reflected in the mentioned plot, with the optimized changes, the run time does show decrease even after reaching a certain number of threads compared to the non-optimized version of the program.