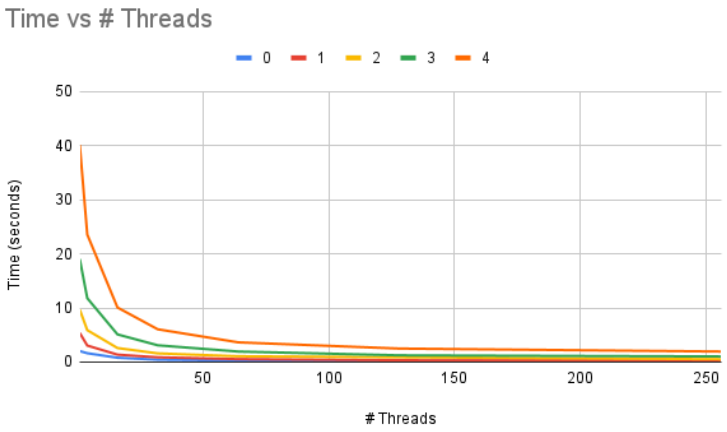
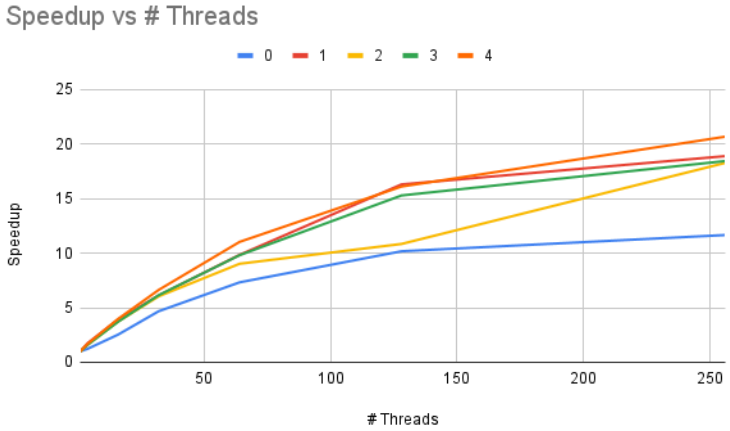
The experiment was conducted on a server with 48 Intel E5-2696 CPUs with clock speed 2.4 GHz via ssh connection. The program was run against 5 test cases of increasing size. Within each test case, we tested 7 different levels of threading: 1, 4, 16, 32, 64, 128, and 256 threads, and recorded the time taken to execute the program.

In the given test cases, the time taken to compute the hash value always decreases when the number of threads increases, with time being approximately proportional to 1/log(n).



The speed-up achieved when the number of threads increased was approximately 17 times more than when using a single thread, when using 256 threads to compute the hash.



Theoretically, this level of speed-up may not be the case with sufficient threading, as the amount of cost of creating threads nears the amount of computation in each thread, however this was not observed in our experiment.

Time vs #Threads	0	1	2	3	4
1	2.026872	5.374225	9.678767	19.035684	40.156938
4	1.631021	3.049452	5.895397	11.802433	23.559551
16	0.797677	1.36733	2.589422	5.110608	10.081833
32	0.433352	0.877338	1.601914	3.096145	6.056116
64	0.276659	0.545802	1.071899	1.939266	3.639262
128	0.199203	0.32967	0.892516	1.245059	2.493421
256	0.173794	0.284446	0.529653	1.03211	1.942123

Speedup vs #Threads	0	1	2	3	4
1	1	1	1	1	1
4	1.242701351	1.76235763	1.641749826	1.612861009	1.704486558
16	2.540968337	3.930452049	3.737809828	3.7247396	3.983098907
32	4.677195444	6.125603815	6.042001631	6.148188796	6.63080727
64	7.32624639	9.84647363	9.029551292	9.815922107	11.03436301
128	10.174907	16.30183214	10.84436245	15.28898149	16.10515753
256	11.66249698	18.89365644	18.27378869	18.44346436	20.67682531