Hong Yuk Sing 3035927257   COMP3230 Prog assignment 2 report

1. Measure the performance

For the test below, the same seeds(42) is used and for each thread number, 10 tests are run to exact the best performance

Thread Numbers	Speed (tok/s)	User Time	System Time	Use Time/System Time
0 (Sequential)	53.940160	4.7361	0.0560	84.5732143
1 (1 (child)Thread)	48.057068	5.2260	0.1785	29.2773109
2	80.376766	5.4888	0.2197	24.9831589
4	124.031008	5.7973	0.2978	19.467092
8	169.536424	6.3563	0.5503	11.5506088
10	173.677069	6.9440	0.7105	9.77339901
12	170.439414	7.8074	0.7906	9.875284594
16	168.754120	8.6029	1.0979	7.83577739
32	143.901068	12.1065	2.7884	4.3417372

2. Data analysis

In general, the multithreading approach can increase the speed of a program (note that a 1-thread setting is not a multithreading approach) and achieve better performance compared to the sequential approach.

Concerning speed, it increases as the number of threads increases from 0 to 10. This suggests that the LLM task can be speed up when the number of threads used increases, which allows the computing task to be divided among different threads. However, when the number of threads increases from 10 to 32, the speed starts to decrease. This can be explained by the fact that when there is a large number of threads, the program needs to devote resources to those threads, which might outweigh the benefits of parallelism. Therefore, the ideal number of threads for this program is around 10.

Both user time and system time increase when the number of threads increases. The increase in user time can be explained by the increase in threads, which results in more time spent allocating the task to different threads and synchronizing them using semaphores (which includes post() and wait() semaphore operations). The system time, which represents the CPU time spent in system mode, also increases with the number of threads due to the additional overhead of managing more threads and synchronizing them using semaphores.

The ratio of user time/ system time decreases as the number of threads increases. This reflected that the programs spends more time of system operation than user task (e.g. computation) when the number of threads increases. The relative cost of system operation will be higher when number of threads increases.

3. Conclusion

In conclusion, while multithreading can significantly improve the performance of a program, there is a trade-off for the cost of system overhead. Therefore, it is important to set a suitable number of thread for multi-threading program in order to optimize the performance.