

REPORT

Threads	Runtime	CPU rate
1	586.41s	99%
2	481.47s	196.2%
4	239.32s	387.1%
8	132.11s	750.3%
16	102.46s	1185.9%
24	87.91s	2137.5%
32	94.13s	2148.9%
64	94.17s	2152.4%
128	94.22s	2164.0%
256	97.33s	2082.3%

For this project, the server I used is `cycle2.cs.rochester.edu`. The CPU has 6 cores and totally 24 threads or hyper-threads. Also, for this computation-heavy work, the major of runtime is spent on matrix multiplication. Thus, we use “time” command to measure the total runtime which roughly refers the CPU time.

While running in sequential mode (threads = 1), the runtime is 586s with a CPU rate of 99%, referring just one CPU thread is working throughout the operation. As the number of threads gets more and more, speed would also get faster and faster until the threads number reaches the number of CPU-threads, i.e. 24. After that, more threads won't lead to a speedup any more as the CPU rate is already 2200%, nearly all 24 CPU threads are occupied. Certainly, the speedup can't reach the same rate of threads number increasing. The reason may be more threads require more time of management and synchronization, as well as the time for memory allocation and input/output is a fixed value.