Report of HW4

First, we use threadpool to ensure that fixed size threads can be fully utilized. And because of the characteristics of threadpool, we can allocate each request well and ensure efficient work. Second, we use synchronized when interacting with the database. Also, we minimize the granularity of the lock. More efficiently ensures the efficiency between threads. For example, two threads can be processed at the same time without waiting. For key parts, such as updating data to the database and order matching, we use synchronized specifically to ensure that dirty reads and data updates don't happen.

By observing the increased response time and cpu usage for threads and kernel, we found that increasing the number of cores significantly improved performance. As the number of requests per thread increases, we will be able to process more requests per unit of time, resulting in less user wait time.

All of these measures have greatly increased the scalability of the server. This ensures that even if many requests are sent at the same time, the server can handle them well. Because threadpool will put requests that did not get thread into the query. Once a thread is released, a request in the query is assigned to it.