In this thesis, we will fix essential defects of a current solution to enhance the performance of data locality. This solution is based on two major algorithms, the Peeling algorithm and the Random Server algorithm. Furthermore, we are looking for a novel algorithm which provides better performance than Random Server algorithm.

The purpose of making data locality solutions is to facilitate data computation as well as any other forms of data replication among a huge cluster computing system of servers. Placing tasks as close as possible to relevant data chunks becomes very important for large-scale computation. In addition, an optimal algorithm which can provide better data locality solution will further enhance industrial application such as cloud computing and SNS performance.

The Random Server algorithm is to randomly choose an idle sever at every assignment step. This server will select tasks whose data is local to the chosen server. When a sequence of such tasks is completed, the server will then make a random selection in the rest of the tasks.

The Peeling algorithm is to give top priority to the idle server who has local data for only one task when every assignment step is taken. When no such idle servers are detected, the Peeling algorithm will stop and the Random Server algorithm will be again applied to take over the progress in the system.

In our recent performance analysis report, we are convinced that the Peeling algorithm outperforms pure Random Server algorithm when the throughput of the system is limited beneath an upper bound. However, in experiments simulating higher load upon the system, the performance of the Peeling algorithm converges to the Random Server algorithm. This problem is intended to be solved and a novel algorithm that substantially outperforms the Random Server algorithm is intended to be developed.