CHAPTER V

Conclusions and Future Works

This chapter summarizes the performance of proposed algorithms. In addition, discussion and future works are presented in Section 5.2 – 5.3.

5.1 Conclusions

This thesis describes five co-allocation strategies for fragmented replicas which are based on dynamic co-allocation strategy [10]. Two of the strategies -- Random and Round-robin algorithms -- are used as baselines for comparison to other three algorithms - Random-with-weighted-probability, Biggest-remaining-first and Fewest-replicas-first algorithms. Performance of these algorithms is measured on simulated grids.

In this thesis, the performance of the proposed algorithms are not compared to the co-allocation strategy for fragmented data is presented in [9] using one by one algorithm. The reason is that the one by one co-allocation strategy is static while the dynamic co-allocation is dynamic. One by one co-allocation strategy uses the estimated completion time which is obtained at the beginning of the transmission to determine the fragment assignment. The fragments are assigned at the beginning of the transmission and are not changed if the network behavior changes. On the other hand, the algorithms proposed in this thesis are adapted from dynamic co-allocation strategy [10] which automatically adjusts workload of each server according to capability of the server. So, it is not reasonable to compare the performance of the proposed algorithms to one by one algorithm.

Furthermore, the workload of servers and failure of link between client and servers are not considered in this thesis. However, the proposed algorithms can adapt well in these situations because the co-allocator automatically assigns more work to faster servers.

5.2 Discussion

From the experiment, it is found that Fewest-replicas-first algorithm performs best in term of the average completion time when the number of replicas for each fragment is different. However, this algorithm does not work well in the situation that the number of replicas for each fragment is equal because Fewest-replicas-first algorithm selects fragments randomly. In this situation, Biggest-remaining-first algorithm should be adopted. This algorithm considers the amount of remaining data to be sent for the fragment for selecting replicas.

Random-with-weighted-probability algorithm does not perform well because it uses the original fragment size to select the replicas. This information is static and defined before starting fragment transmission. It does not reflect the real-time situation, and this algorithm cannot adjust well when the available bandwidth is changed. As a result, its performance is not good.

The effect of fluctuation of the available bandwidth is also studied in this thesis. It is found that all five algorithms are not sensitive to the variance of the transmission rate because of the characteristic of dynamic co-allocation [20]. The dynamic co-allocation can automatically adjust the workload according to the network bandwidth. That is, a faster server gets more work while a slower one gets less.

From the experiment, it shows that Fewest-replicas-first and Biggest-remaining-first algorithms work well in the different situations. To achieve the best performance from both algorithms, these two algorithms and Random algorithm are combined, as shown in Figure 25.

Combined_algorithm(server *i*).

IF there is only one fragment whose number of replica is fewest

F_a = *Fewest-replicas-first* (server *i*).

ELSE

IF there is only one fragment whose size is biggest

F_a = *Biggest-remaining-first* (server *i*).

ELSE

F_a = *Random* (server *i*).

ENDIF

ENDIF

Figure 25: Combined algorithm

The combined algorithm yields the best performance of Random, Fewest-replicas-first, and Biggest-remaining-first algorithms because the combined algorithm considers both fragment size and the number of replicas to select the fragment to the server. As a result, the combined algorithm works well in most situations.

5.3 Future Works

In this study, one grid framework is studied. The proposed algorithms need to be implemented in different grid topology to study the performance.