Chapter 6

Conclusions and Future Directions

6.1 Contributions

- A taxonomy of data management that highlights different approaches and various techniques are presented. This thesis explores notable prior work in the area of scalable transactions in cloud data stores. Comparison of proposed work with the existing work is also presented.

- The architecture of system is proposed to process scalable transactions on the partitions which are distributed among a cluster of low cost commodity servers.

- The design of the workload-driven partitioning, which forms the partitions based on data access patterns of web application is introduced. The proposed partitioning scheme uniformly balances the load among all partitions which in turn increases the throughput of the overall system. Demonstration of how this workload-driven partitioning can be used to limit the transaction to one partition is explained.

- A Mathematical Formulation of workload-driven partitioning is presented. Data partitioning strategy is introduced which finds out all possible combinations of partitions using mutation and selects the partitions with optimized load and association.

- Workload-driven partitioning algorithm for achieving scalability is proposed. The proposed algorithm reorganizes the application data based on data access patterns of web applications. Performance evaluation is conducted through experimentation over data stores such as Amazon SimpleDB and Hadoop HBase. Experimental and analytical results are observed and it clearly shows that the
proposed scheme outperforms the well known schema level partitioning in terms of throughput, response time, efficiency.

- Demonstration of detailed experiments that show the effectiveness of workload-driven partitioning scheme in forming partitions that balance the workload among the partitions is explained.

- A new metric for finding efficiency of workload-driven partitioning algorithm is introduced. Comparison of static, dynamic and workload-driven partitioning is explained in detail.

- Statistical model and analysis of workload-driven partitioning in Amazon SimpleDB is presented.

- The workload-aware elasticity algorithm is demonstrated by adding or removing the domains of Amazon SimpleDB based on average load. A framework for workload-aware elasticity is proposed. The framework, is introduced to build elastic and scalable data store in the cloud. Experimental results clearly show that throughput increases linearly while performing scale-in and scale-out operation.

- A partitioning evaluator is proposed which finds relationship between throughput, number of users and number of fragments. It shows the impact of these factors on throughput. An analysis of workload-driven partitioning, schema level and graph partitioning is performed. From the analysis, it is proved that these two factors affect the throughput. It also evaluates throughput with the input workload.

The contributions of this work as follows:

- Design of workload-driven partitioning based on data access patterns.

- Mathematical Formulation of workload-driven partitioning and workload-driven partitioning algorithm which reorganizes application data, based on data access patterns.

- Workload-aware elasticity framework for building scalable and elastic data store.

- Workload-aware elasticity algorithm for performing scale-in and scale-out operation.
6.2 Conclusions

Workload-driven partitioning that restructures application data based on data access patterns of web applications is presented and evaluated. Analytical and experimental results are observed and it clearly shows that workload-driven partitioning outperforms the existing tree based schema level partitioning in terms of throughput, response time and efficiency. Workload-driven partitioning is one of the first partitioning schemes which forms the partitions based on data access patterns of web applications. Workload-driven partitioning algorithm forms optimized partitions and also shows load balancing capabilities for transactional workloads is demonstrated. Mathematical formulation of workload-driven partitioning is also presented.

A taxonomy of data management in the cloud which highlights different approaches and various techniques is presented. Comparison of proposed work with the prior work is also provided.

Workload-aware elasticity framework for building elastic and cloud data store is presented. Workload-aware elasticity algorithm which performs scale-in and scale-out operation is demonstrated in Amazon SimpleDB.

Partitioning evaluator which finds the relationship between throughput, number of fragments and number of concurrent users is proposed. It also evaluates the throughput with the given input workload as number of fragments and number of concurrent users. These factors influence throughput in workload-driven, graph and schema level partitioning is proved.

6.3 Future Directions

In this section the areas of future work are discussed.

- Workload-driven partitioning algorithm - Performance of workload-driven partitioning can also be evaluated using the TPC-W benchmark.

- Workload-aware elasticity algorithm - It can be demonstrated in Hadoop HBase.
• Partitioning Evaluator - Evaluation of the different partitioning algorithms is difficult because there is no common equation or objective function for evaluating and comparing different partitioning algorithms.