**Abstract**

**A Theoretical Analysis of the Backup-Process for MongoDB with an Implemented System**

In this work, we have aimed to find a theoretical and practical method to manage restoring backup, while at the same time allowing the normal functioning flow of the system. After researching an initial theoretical analysis, surveying the various modes of backup that have been described, we reached our preliminary conclusion: This was that an implementation of a two-staged backup process, based upon the available tools of the MongoDB system, was feasible. In order to validate the primary theoretical conception, achieve proof of concept, and attain an actual and practical running system we have built a fully implemented system. The presented system enables the possibilities of running the backup in different modes of deployment: standalone, replica set, and sharding. The system is capable of performing the two-stage backup without locking the records and without constraint, in a running state parallel to the flow of the system, for the modes of replica set and sharding.

In order to obtain optimal results, we have researched elaborately the MongoDB system, while paying attention to the advantages and disadvantages of the supplied processes. The present day state of the art of relational data base systems is such that the relational databases are presently incapable of handling efficiently the massive amounts of data requirements from modern organizations. It was for this reason that non-relational databases with NoSQL have been researched and implemented in the last decade. Especially noteworthy in this class is the MongoDB system. We have achieved our results by elaborately researching in detail the MongoDB system, while judging the advantages and disadvantages of the various options available for the backup process supplied by the system.

The theoretical presentation analyzes in detail the functioning components of the system, and points out the advantages of the two staged backup process, which functions in coordination with these different components. These components include: the replica set and sharding. The subcomponents include: journal, locking, oplog, balancer, Mongos, configuration server, and split and migration. The function of these components is described in our work. Replica set is composed of several servers that hold the same data and function in a master slave relationship. Sharding is the horizontal partitioning of the data in different servers.

The second part of this work demonstrates in a practical working system the MongoDB system with three types of deployment for backing up the system. These different types are dependent on the system being either a standalone system, a replicas set with multiple servers of identical data, and sharding, i.e., the distribution of data across many servers. The implemented system supplies a different API for connecting and for management of the multiple collections of the databases. The system allows the online entry of data with the subsequent backup, without a requirement of locking the system. In order to validate proof of concept, one may delete the data and recover the collection from the backup file. The data may be viewed at each stage with the accompanying GUI interface.