Abstract

In a company, it's important to have a database management system, which is capable of containing every document that is owned by their employees. The data will become very big, which makes it impossible to put the whole data into a single server. To solve this problem, that huge data can be distributed to clusters. In this distributed database environment, the implementation of MapReduce will give more contribution to the system. MapReduce is an operation, or a method that utilizes the divide-and-conquer alghorithm. As the name suggests, MapReduce consists of a mapping process and a reduce process, which will aggregate useful information from the dataset.

In this final project, a research will be conducted to observe MapReduce's performance in a distributed database environment. First of all, it is needed to make sure that every computer that is used in the system has *the exact* specification of hardware, software, and database management system. After that, import the document-oriented dataset into the system's *master*'s database. Then, by using a method called *sharding,* each of the computers will be given roles, so that: *master* as the *router,* one computer as a config server, and the rest as the *shard servers*; only after that that the homogenous distributed document-oriented database is built. The dataset will then be distributed across the shards. Finally, the MapReduce query will be executed and observed under a single server and 3 different architectures of distributed database environment.

The overall result of this research shows that MapReduce performs better in a distributed environment than in a single server. The conclusion is that the distributed database environment improves the performance of MapReduce.

Keywords: *MapReduce*, *document-oriented database*, *distributed database system*, *distributed database management system, homogeneous distributed database*