Chapter 4

MonogDB: Data Management in NoSQL

Sonali Tidke
SPPU, India

ABSTRACT

MongoDB is a NoSQL type of database management system which does not adhere to the commonly used relational database management model. MongoDB is used for horizontal scaling across a large number of servers which may have tens, hundreds or even thousands of servers. This horizontal scaling is performed using sharding. Sharding is a database partitioning technique which partitions large database into smaller parts which are easy to manage and faster to access. There are hundreds of NoSQL databases available in the market. But each NoSQL product is different in terms of features, implementations and behavior. NoSQL and RDBMS solve different set of problems and have different requirements. MongoDB has a powerful query language which extends SQL to JSON enabling developers to take benefit of power of SQL and flexibility of JSON. Along with support for select/from/where type of queries, MongoDB supports aggregation, sorting, joins as well as nested array and collections. To improve query performance, indexes and many more features are also available.

INTRODUCTION

Today is era of computing system which are generating enormous data every day through various web and business applications. This data need to be stored in easy to access and retrieve format. A huge amount of such data is processed by RDBMS (Relational database management system). Relational model was proposed by E.F. Codd in 1970. This model is well suited for storing data in structured format for client-server programming. But with increase in generation and complexity of data, relational model is being inefficient to fulfill today’s need.

NoSQL provides support for such large applications. NoSQL is schema less database management model which does not restrict user to store data in fixed type. MongoDB is one such DBMS developed using NoSQL concepts. MongoDB is a document based database management system.

Few reasons why NoSQL is better for data management than SQL are:

DOI: 10.4018/978-1-5225-2486-1.ch004
1. SQL cannot scale up to the current requirements of big data storage. RDBMS stores data using relations. Data is correlated with some common characteristics between stored data which results in the schema of RDBMS. This schema does not allow SQL to store unpredictable and unstructured information. Big data applications require to store data without fixed structure which NoSQL supports. NoSQL avoids major join queries making access to data faster and easy to store.

2. RDBMS requires data to be normalized into logical tables to avoid data redundancy and duplication. Though it helps in managing data efficiently, increase in data complexity hampers performance of SQL. In contrast, NoSQL stores data in the form of collections which makes data access faster. NoSQL processes complex data in real time.

3. SQL works on ACID (Atomicity, Consistency, Isolation, Durability) properties while NoSQL supports CAP (Consistency- Availability – Partition tolerance) theorem and works on BASE (Basically Available, Soft State, Eventually Consistent) properties.

Considering all this points, objective of this chapter is to focus on why and how MongoDB is useful for various business requirements. This chapter also gives various details about managing data using MongoDB.

**BACKGROUND**

**NoSQL**

NoSQL (Moniruzzaman & Hossain, 2013) is a non-relational database management system which does not restrict data storage in predefined schema. Websites like Google or Facebook collects terabytes of data from various sources for their users (Ricardo & Susan, 2016). It is not feasible to store such data in fixed schema and typically requires horizontal scaling. In such situation, NoSQL databases has become alternative for relational databases by providing flexible, horizontally scalable, schema-less data model.

**Use of NoSQL**

Now a days, many third parties are capturing user data easily for their use and analysis. Data like user personal information, social graph, location details, log file are few examples where data has been increasing tremendously. SQL databases are designed for fixed schema data with limited size. NoSql databases are evolved and becoming popular where such type of data is required to handle properly.

NoSQL is in used from more than a decade, it is facing various challenges. Following are few challenges which can be considered before choosing NoSQL for enterprise applications:

1. **Maturity:** RDBMS is in use from a long time and it is very stable and having rich in functions provided for handling data. In contrast to this, most of the NoSQL variations are still in development phase where many key features not yet implemented.

2. **Support:** RDBMS vendors provide assurance and support of data recovery for any kind of failure. Since most of the NoSQL systems are open source projects, guaranteed support cannot be assured for these systems.