Querying and Modeling Data Using Mongo DB

Asst Prof. Ravindra Sonawane[1], Asst Prof. Dipali Khairnar [2]

Department of Computer Engineering

[1,2] D.Y. Patil College of Engineering, Ambi, Pune.

Abstract- *With the uninterrupted growth of data volumes, the storage of information, support and maintenance have become the biggest challenge. Map reduce is method which can be used to reduce the data set to reduce query processing time and improve system throughput, In the Proposed system we are going to mine the big data this Hadoop and Mongo db and we will try to mine the data with sorted or double sorted key value pai ,for and analyze the outcome of system. Relational database products fall behind to scaling the applications according to the incoming traffic. Due to huge data storage and scaling demands, growing number of developers and users have begun turning to NoSQL databases. This paper describes data modeling and query execution in MongoDB Document database. This paper shows how data is retrieved from MongoDB Document database without using JOIN.*

Keywords- Mongo DB, Key Oriented, Document, Scalability, JSON, Relational, Graph.

I. INTRODUCTION

RELATIONAL databases have been used for decades for general data storage in web and business applications with millions of reads but few writes requirements.The data stores for these applications needs to provide good horizontal scalability. Horizontal scalability means the ability to distribute both the data and simple read/write operations over many servers. The relational databases have little capabilities to horizontal scale over many servers. NoSQL databases were developed to deal with such large scale data needs vertical scalable. The term “NoSQL” was first used by Carlo Strozzi [1] in 1998 for his RDBMS, Stozzi NoSQL. Recently, the term NoSQL (Not Only SQL) has been used for databases which don’t use SQL (Structured Query Language) as its query language and which don’t require fixed table schema. A key feature of NoSQL systems is “Share nothing” horizontal scaling-replicating and partitioning data over many servers [2]. Due to this feature, NoSQL systems can support a large number of simple read/write operations per second. NoSQL systems don’t provide ACID (Atomicity, Consistency, Isolation, Durability) guarantees but follow BASE. BASE is acronym for Basically Available, Soft state and Eventually consistent.Basically available means that most data is available most of the time [8].

1.1 Key-Value databases

In key value data stores, data is stored in the form of keys and values [6]. Each key is unique and keys are used to retrieve the values. The query speed of these databases is higher than the relational databases. Amazon’s Dynamo and Riak are famous key value data stores. These are used in applications where schema is continuously evolving.

1.2 Column-oriented databases

In column oriented data stores data is stored by columns and columns of related data is stored in same file which are called column families. These data stores are mostly used in read intensive applications. Base and Cassandra are famous column oriented database.

1.3 Document databases

Document data stores are similar to key value data stores but the value is stored in JSON or XML format. It is used for applications in which data is changed occasionally like Customer Relationship Management System.

1.4 Graph databases

Graph Database uses graph structure with nodes, edges and properties of the edges to store the data. They are suited for the applications in which there are more interconnections between the data like social networks. OrientDB and neo4j are popular open source graph databases.

The objective of this paper is to show how schema is modeled and data is queried in MongoDB. The paper is organized as follows: In section 2 document\oriented databases are described. Section 3 shows class diagram and JSON representation of MongoDB schema. In section 4 describes how queries are executed in MongoDB.

II. DOCUMENT DATABASES

Document database stores data in the form of documents rather than as normalized relational table in relational databases. Data format of these documents can be JSON, BSON or XML [7]. Documents are stored into collections. The relational equivalent of document and collection are record (tuple) and relation (table).But like relation collection does not enforce fixed schema. It can store documents with completely different set of attributes. Documents can be mapped directly to the class structure of programming language but it is difficult to map RDBMS entity relationship data model. This makes easier to do programming with document databases. There is no need of JOINS in document databases as in RDBMS due to embedded document and arrays. That is why today a growing number of developers are moving to document databases. CouchDB by Apache Software Foundation and MongoDB by 10gen are open source databases built for scalability and ease of use. MongoDB is an open source NoSQL document database, initiated by 10gen Company. It was designed to handle growing data storage needs. It is written in c++ and its query language is JavaScript. MongoDB stores data in the form of collections. Each collection contains documents. MongoDB documents are stored in binary form of JSON called BSON format. BSON supports Boolean, float, string, integer, date and binary types. Due to document structure, MongoDB is schema less. It is easy to add new fields to a document or to change the existing structure of a model. MongoDB offers a technique named Sharding to distribute collections over multiple nodes. When nodes contains different amount of data, MongoDB automatically redistribute the data so that load is equally distributed across the nodes. MongoDB also support Masterslave replication. The slave nodes are copies of Master nodes and used for reads or backups

III. DATA MODELING IN MONGODB

The example database which has been used for querying MongoDB contains three collections- User, Tag and Post.

Data in MongoDB has a flexible schema. Documents in the same collection. They do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

```
{
  _id: POST_ID
  title: TITLE_OF_POST,
  description: POST_DESCRIPTION,
  by: POST_BY,
  url: URL_OF_POST,
  tags: [TAG1, TAG2, TAG3],
  likes: TOTAL_LIKES,
  comments: [
    {
      user:'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    },
    {
      user:'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    }
  ]
}
```

IV. JSON REPRESENTATION

JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange. The JSON format was originally specified by Douglas Crockford, and is described in RFC 4627. The official Internet media type for JSON is application/json. The JSON filename extension is .json.

This tutorial will help you understand JSON and its use within various programming languages such as PHP, PERL, Python, Ruby, Java, etc.

```
{
  "book": [


    {
      "id":"01",
      "language": "Java",
      "edition": "third",
      "author": "Herbert Schildt",
    },


    {

      "id":"07",

      "language": "C++",

      "edition": "second",

      "author": "E.Balagurusamy",

    }


  ]

}
```

V. QUERY MODEL

Queries for MongoDB are bidding in a JSON like syntax and are forward to MongoDB as BSON altar by the database driver. The concern archetypal of MongoDB allows queries over all abstracts central a collection, including the anchored altar and arrays. Through the acceptance of predefined indexes queries can dynamically be formulated during runtime.

Not all aspects of a concern are formulated aural a concern accent in MongoDB, depending on the MongoDB disciplinarian for a programming language, some things may be bidding through the abracadabra of a adjustment of the driver. { “employees":[{ "firstName":"Manmit","lastName":"Zala" }, {"firstName":"Pradip" , "lastName":"Chavda" },{"firstName":"Nilay", "lastName":"Parekh" }]} The query model supports the following features:
1. Queries over documents and embedded subdocuments
2. Comparators (<;_;_;>)
3. Conditional operators (equals, not equals, exists, in, not in ...)
4. Logical perators: AND
5. Sorting by multiple Attributes

5.1 Query: - Find the tag names which have been used in the post under which a particular user has commented. (user=’a1’)

MySQL mysql> select t.name from comment c,post p,user u,tag t where p.pid=c.postid an d c.userid=u.uid and p.pid=t.pid and u.username='a1'; output: +--------+ | name | +--------+ | space | | apple | | google | | apple | +--------+ MongoDB > var u=db.user.findOne({username:"a1"}) >var tag1=db.post.find({"comments.userid":u._id}) > db.tag.find({_id:tag1.tagid},{_id:0,name:1})

5.2 Query:- Find the tag(s) which have been used in the posts of each user

mysql>select u.uid,t.tagid from user u,post p,tag t where u.uid=p.uid and p.pid =t.pid;

MongoDB db.post.find({},{_id:0,uid:1,tagid:1})

5.3 Query: - Given a cid of a comment find its related post, parent comment and tags associated with the post.

MySQL mysql> select c.cid, c.postid, c.parentid, t.tagid from comment c,post p ,tag t where c.postid=p.pid and p.pid=t.pid and c.cid='c2'; MongoDB db.post.find({"comments.cid":"c2"},{"comments. cid":1, "comments.parentid":1,tagid:1});

5.4 Query: - Find the time of the post under which some user has commented. (user=’a1’)

MySQL mysql> select p.time from comment c,post p,user u where p.pid=c.postid and c. erid=u.uid and u.username='a1'; MongoDB > var u=db.user.findOne({username:"a1"}) > db.post.find({uid:u._id},{time:1,_id:0})

VI. CONCLUSION

In this paper data modeling in MongoDB has been shown by using Class diagram and JSON format. It also has been shown that how queries are written in MongoDB. MongoDB does not use JOINs to relate documents like Relational Databases. In this all the data is stored in Single document or if needs to store in different documents then documents are related by using reference fields. Now a day data increases day by day the storage, retrieval and analysis of bigdata in structured databases like Oracle and Mysql is not possible so we have presented many Nosql system among them Mongo db is preferable for as an alternate for Mysql , still it is an Active search are for data mining to mine knowledge from bigdata.In future we are interested in batter method and system for efficient mining of bigdata.

REFERENCES

[1] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: a distributed storage system for structured data. In Proceedings of the 7th symposium on Operating systems design and implementation, OSDI.

[2] Olli Sutinen,” NoSQL – Factors Supporting the Adoption of NonRelational D Databases” M.Sc thesis, Dept. Comput. Sci.,tampere Univ., Finland, 2010.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4. Jyoti Nandimath , Ankur Patil , Ekata Banerjee , Pratima Kakade :”Big Analysis using Apache Hadoop “ In SKNCOE Pune India,2013.

[4] J. Wang, ―Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),‖ IEEE J. Quantum Electron., submitted for publication.

[5] Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin” MongoDB vs Oracle - database comparison “2012 Third International Conference on Emerging Intelligent Data and Web Technologies,2012

[6] Suyog S. Nyati, Shivanand Pawar ,Rajesh Ingle: ” Performance Evaluation of Unstructured NoSQL data over distributed framework” 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI),2013

[7] Lior Okman,Nurit Gal-Oz,, Yaron Gonen,, Ehud Gudes ,Jenny Abramov:” Security Issues in NoSQL Databases” 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11,2011

[8] Chanchal Yadav, Shuliang Wang , Manoj Kumar: Algorithm and approaches to handle large Data- A Survey. IJCSN International Journal of Computer Science and Network, Vol 2, Issue 3, 2013

[9] Ruxandra Burtica, Eleonora Maria Mocanu, Mugurel Ionut Andreica, Nicolae Tapus: Practical application and evaluation of no-SQL databases in Cloud Computing , ©2012 IEEE

[10] Romain Fontugne,,Johan Mazel, Kensuke FukudaHashdoop: A MapReduce Framework for NetworkAnomaly DetectionCERN – European organization for nuclear Research 2014 IEEE INFOCOM Workshops: 2014 IEEE INFOCOM Workshop on Security and Privacy in Big Data,2014