International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 6 1057 – 1059

Overview of Indexes Used in NOSQL Databases of MongoDB Architecture

Ms. Poonam Verma, *Email Id: newjimsgn@gmail.com*
Ms. Arpana Chaturvedi, *Email Id:pcord.bca@jagannath.org*

Abstract: The present day transactions result into petabytes of data collected and the credit almost goes to the booming ICT industry. The data received is able to detect the hidden patterns for the enterprises and research industry and help them to improve their traditional methods. However the data is unstructured and requires new innovative technologies to be implemented in the architecture handling big data. In the present paper, we have tried to explore the NOSQL database handling techniques and specifically the indexes that help to reduce the time complexity to handle the unstructured data. This paper is divided into four sections where the first section compares the DBMS and DSMS followed by the literature review on this technique and introduction to the MongoDB architecture and overview of NOSQL databases. The fourth section handles the types of the databases, it's index types. Fifth section describes the performance comparison of the various MongoDB with RDBMS.

Keywords: *DBMS,DSMS, NOSQL, MongoDB, Indexes*

*****

I. Introduction

DBMS is one of the most popular research area that has been able to handle the traditional transactions. Transactional data in todays' ICT era amounts to petabytes. This data cannot be recognized as big data as it consists of only structured data. However DBMS does not deal with streaming data also known as DSMS. DSMS is data streaming media system and it is able.to handle the streaming data. Only a summary of data is stored on the DSMS. Various slices on the storage are maintained in order to keep a track of the streaming data in a particular sequence.

In the present scenario the streaming data flow is huge and it is also at a point considered to be unbounded .this permits the research in the area of temporal data models. In the present article, we provide an overview of the basic operations of MongoDB that is able to handle the complex queries

II. Literature Review:

Abhijeet Raipurkar, G.R. Bamnote (2013) focus is given on Query processing in a distributed system requires the transmission of data between computers in a network. Two cost measures, response time and total time are used to judge the quality of a distribution strategy. They presented various algorithms are used that derive distribution strategies which have minimal response time and minimal total time, for a special class of queries.

Xiao feng Li, Dong Li, et al (2010) is based on study of some common optimization algorithm based on multi relation semi join is put forward to apply to this situation that takes buffer zone of distributed database system as the final assembly station of intermediate result query. The experiment proves that query that query optimization algorithm based on multi relation semi join reduces the data volume of intermediate result and effectively decreases the overall cost of network communications.

III. Overview of NOSQL and Mongo DB

NOSQL has a bridge connecting between the traditional application modes and latest data models.The traditional framework of RDBMS permits only predefined structure to store the data. This feature causes an error as it requires to handle unstructured data which is also streaming. RDBMS is also unable to handle large data in video formats.

NOSQL requires the databases on the same platform and applications are able to send data, post queries etc. NOSQL permits to store data in the form of graphs and edges

connecting various nodes define the pattern or relationship between this data. For RDBMS data should be in the ACID compliance in order to achieve consistencies.

NOSQL databases permits the following features:

- The response to every data is almost in immediate mode.
- Append only file permits to recover data even during crash.

Most popular NOSQL database is MongoDB that has been a buzzword in most of the technologies. It is able to create exact replicas or copies of data in all the clusters at the horizontal pattern of the nodes in a network. While selecting the database, following criteria should be kept in mind:

A) Suitability of a solution that suits the required criteria, which would be possibly either document based, real time data and data for analytics.

B) Moreover it becomes necessary that the scalability, persistence should also be fulfilled. MongoDB follows a document based databases where data exists in key value pairs.

In MongoDB there exists GRIDFS that converts data into binary objects into chunks and stores them in different nodes. The main attraction of MONGODB is the map reduce model which collects the data from the storage layer and provides various functions of map & reduce written in java and c# that further helps to transform data in different formats. The system may achieve parallelism by partitioning data and processing each partition individually at parallel level, along with functions such as load balancing in different clusters, detection of data format or values that lie outside the boundaries, it further helps in recovery of data after an abrupt failure.

MONGODB is followed in distributed applications. The data existing might be redundant. MongoDB deals with collection than dealing with the tables.

IV. NoSQL Database Types:

There are basically 4 types of NoSQL that have a popular reach and they are as following:

a) Document Databases: These databases can consist of different key-value pairs or nested documents.
b) Key Value Stores: This consists of the items in the databases with an attribute name along with its value.
c) Graph Stores: These are used to store information about the various social networks of data.
d) Wide Column stores large columns of data and only the primary key can be queried.

V. Types of Indexes :

Document databases provide various types of indexes and they are:

i. Compound Indexes
ii. Geospatial Indexes
iii. Sparse Indexes
iv. Time To Live Indexes
v. Text indexes
vi. Unique indexes

i. Compound Indexes: Compound Indexes permit to include data from multiple fields. When the database indexes are being created, it blocks the operations such as querying to be carried out on the databases. However NoSQL, offers a feature of creating a Compound Index in the Background. However the background indexing causes the blocking of the administrative operations of repairing databases. Background construction of indexes for large volume of data can be expensive.

ii. Geospatial Indexes: The real-time data of the earth can be stored and searched using the Geospatial indexes in the MongoDB. The Location data can be stored as the GeoJSON objects and the Legacy coordinate pairs. The basic operations of the calculations of the 2Dsphere on the sphere.

iii. Sparse Indexes: Documents with merely a indexed field is accepted in the sparse indexes, even if the index consists of null value. Sparse index does not include all documents of a collection.

iv. Time To Live Indexes: Sometimes , it becomes

necessary for removing the data automatically like machine generated event data, logs, and session information that is required to persist in a database for a finite amount of time. Compound Indexes do not support TTL indexes and ignores such indexes.

v. Text indexes: Text Indexes can include any field which is an array of string elements. A Wildcard text index permits to search the documents on multiple fields. However one collection consists of only one text index.

vi. Unique indexes: These indexes permit to use unique values and do not permit duplicate values. Unique index cannot be implemented on Hashed Index.

VI. Comparison of MongoDB with RDBMS:

In MongoDB, the data is stored in the form of a single document as a single collection. Only one lookup operation on the aggregated database is to be carried out to find the result of a complicated query. The insert and select operations can be thus carried out faster than the Traditional RDBMS, as the Traditional RDBMS, requires one index lookup operation to be operated on the root table with a clustered index and lookup is to be implemented on large number of the data in each cluster. Different tables might reside in non-contiguous form on the disks.

VII. Conclusion:

In this paper, we have presented an overview of the working operations in the popular database of MongoDB architecture. We have further presented the different types of indexes that can be implemented on the databases for faster query handling. The basic comparison has been shown between the MongoDB with RDBMS has been also presented. In the next paper, we will be presenting the experimental tests proving the performance of MongoDB with RDBMS.

References:

[1] Chen, Ming-Syan, and Philip S. Yu. "Using join operations as reducers in distributed query processing." Proceedings of the second international symposium on Databases in parallel and distributed systems. ACM, 1990.

[2] Kim, C., Shim, K. 2015. Supporting set-valued joins in NoSQL using MapReduce, IS Journal, 49, 52-64.

[3] Ntarmos, N., Patlakas, I. Triantafillou, P. 2014, Rank Join Queries in NoSQL Databases, PVLDB, 7(7), 493-504.

[4] MongoDB High+Performance+Benchmark+White+Paper_final paper March 2015.

[5] MongoDB Books http://mongodb.org/display/doc/

[6] Lior Okman, Nurit Gal-Oz, Jenny Abramov,"Security Issues in NoSQL databases",IEEE, 978-0-7695-4600-1, 2011