A REVIEW ON DOCUMENT ORIENTED AND COLUMN ORIENTED DATABASES

Jaspreet kaur
Student of MTech (CSE)
Sri Guru Granth Sahib World University,Fatehgarh sahib, India

Harpreet Kaur
Student of MTech (CSE)
Sri Guru Granth Sahib World University,Fatehgarh sahib, India

Kamaljeet kaur
Assit. Prof. in CSE dept.
Sri Guru Granth Sahib World University,Fatehgarh sahib, India

ABSTRACT

Non-relational databases are growing these days. Non-relational databases consists of various types of databases which are used to handle different types of data and have different features like Key value stores, document oriented databases and column oriented databases, graph databases and many others. In this paper we study the two popular non-relational databases document oriented databases and column oriented databases, describe their types, advantages and disadvantages.

KEYWORDS: Non-relational, Document oriented databases, Column oriented databases, Key value stores.

1. INTRODUCTION

Non-relational databases are very popular and in use these days because of their various advantages over the relational databases like handle various types of data like graph, object, semi structured and structure data. These databases can handle very large amount of data and also provide greater scalability that is why these are very useful to use in distributed environment like in cloud and grid computing applications[1].Non relational databases has many categories like Key value stores, Document oriented databases, Column oriented databases, graph databases and many others[1]. In this paper we will discuss the document oriented databases and column oriented databases.

2. DOCUMENT ORIENTED DATABASES

Document oriented databases are one of the main categories of non-relational databases. Document oriented databases are used to store, manage and retrieve the structured or semi-structured data in the form of a document. The main concept in these types of databases is “document” which is like a record in relational databases but it is different from records in many aspects like it is less rigid and use different format to store data. The document oriented databases store data in JSON, BSON or XML format and many others [8]. A document in document oriented database is look like following example:

{
Emp_Name:”Seema”.
Emp_id:”1345”
Date_of_birth:”15-03-1985”
}

There are many databases which come under this category and are listed below [9]:

- Mongodb
- Couchdb
- Ravebdb
- SimpleDB
- OrientDB
- Jackrabbit
- IBM Lotus Domino

- Couch base server

We will describe Mongodb, Couchdb and ravendb in following sections.

2.1 Mongodb

MongoDB is an open source document-oriented database system developed and supported by 10gen. It is part of the NoSQL family of database systems. Instead of storing data in tables as is done in a "classical" relational database, MongoDB stores structured data as JSON-like documents with dynamic schemas making the integration of data in certain types of applications easier and faster. It is written in C++. MongoDB is a scalable, high-performance, open source NoSQL database Written in C++ [3].

Mongodb features are listed below:

- It storesJSON-style documents with dynamic schemas which provide simplicity [3].
- It provides full index support. We can index any attribute for high performance [3].
- Mongodb provides replication for fault tolerance and high availability [3].
- Mongodb provides horizontal scalability which is less complex and less expensive than SQL databases [3].
- It also supports Map/Reduce features [3].

2.1 Couchdb

CouchDB is a “NoSQL” database, categorized in document stores. While this term is a rather generic characterization of a database, or data store, it does clearly define a break from traditional SQL-based databases. A CouchDB database lacks a schema, or rigid pre-defined data structures such as tables. Data stored in CouchDB is a JSON document. The structure of the data, or document, can change dynamically to accommodate evolving needs. It is written in Erlang [4].

The main features of couchdb are given below:

- CouchDB stores data as "documents", as one or more field/value pairs expressed as JSON [5].
- CouchDB provides ACID semantics. It does this by implementing a form of Multi-Version Concurrency Control [5].
- The stored data is structured using views. CouchDB can index views and keep those indexes updated as documents are added, removed, or updated [5].
- Couchdb also provides Map/Reduce functionality [5].
- Couchdb has distributed architecture with replication [5].
- CouchDB guarantees eventual consistency to be able to provide both availability and partition tolerance [5].
- CouchDB can replicate to devices (like smart phones) that can go offline and handle data sync for you when the device is back online. CouchDB also offers a built-in administration interface accessible via web called Futon [5].

2.3 Ravendb

RavenDB is a transactional, open-source Document.Database written in .NET, offering a flexible data

model designed to address requirements coming from real-world systems. RavenDB allows you to build high-performance, low-latency applications quickly and efficiently. Data in RavenDB is stored schema-less as JSON documents, and can be queried efficiently using Linq queries from .NET code or using Restful API using other tools. Internally, RavenDB make use of indexes which are automatically created based on your usage, or were created explicitly by the consumer. RavenDB is built for web-scale,offering replication and sharding supp ort out-of-the-box [6].

Ravendb main features are listed below [6]:

- It is a schema free database like other document oriented databases.
- Sharding, Scaling, replication and multi-tenancy are supported in Ravendb.
- ACID transactions are fully supported, even between different nodes.
- RavenDB is a very fast persistence layer for every type of data model.
- Ravendb is easily extended by bundles.
- It also support map/reduce functions.

ADVANTAGES OF DOCUMENT ORIENTED DATABASES:

- Document oriented databases are schema less, so the complexity is less [8].
- Document oriented databases can handle unstructured, semi structured and structured data [1].
- Document oriented are highly scalable [1].
- Document oriented databases can handle very large amount of data and provide horizontal scalability and sharding that is why best suited for distributed environments [1].
- Document oriented databases has high flexibility while storing data. No need to predefine data type of data [1].

DISADVANTAGES OF DOCUMENT ORIENTED DATABASES:

- Reliability is lesser than relational databases[2].
- Some document oriented databases has less security than the relational databases [7].

3. COLUMN ORIENTED DATABASES

In traditional database management systems introduced the concept of row oriented databases. Row oriented database is the database which stores data in rows. It has high performance for the OLTP i.e. online transaction processing. But there are many problems with row oriented database like data compression, difficult to handle complex read queries, creates very large I/O burdens and perform low operations, increase the no of actual disk reads to satisfy a query, storage of multiple indexes so to overcome from these problems, column oriented databases come in to existence in non relational databases [9].

Column rather than the row. In Column-stores each database table store column separately, with attribute values belonging to the same column as compared to traditional database systems that store entire records (rows) one after the other. It is mainly used in OLAP (online Analytical Processing), Data Mining operations. It supports for large-scale, data-intensive applications (especially data warehousing and business intelligence), Customer

Relationship Management (CRM).-oriented database systems (column-stores) have scaled a lot of attention in the past few years as opposed with traditional databases. A column oriented DBMS is a database management system that stores its content by column rather than row [11].

The only main difference between row and column stores is physical storage and query optimization [10].

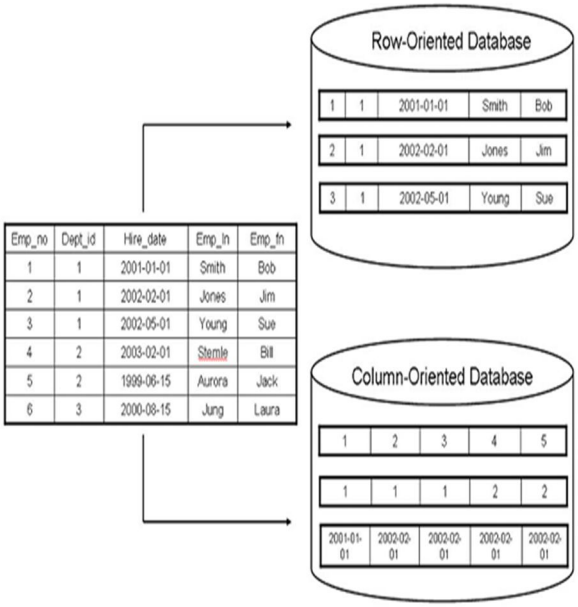

Fig.3.1 COLUMN V/S ROW ORIENTED DATABASE [12]

Columnar database column-based structure [10]

The column-based DBMS stores all of the values from one column of a table in a contiguous data set. This allows the reading and writing of parts of records. It conserves I/O bandwidth by transferring only the values that may be used in the query.

The first table represents the physical storage of the record-based structure in RDBMS, which is the simplest DBMS. The data is stored in the table as simple DBMS.

Cust ID	Name	City	State	Region
1222	ABC	Delhi	MN	Central
1893	DEF	Ldh	MN	North
3737	IJK	Asr	MN	
2788	XYZ	Agra	MN	Central
5467	MNP	Chd	MN	South

TABLE 1 [10]

The second table represents the physical storage of a column-based structure in columnar databases. Each of the separate blocks in this diagram represents separate storage areas, either as separate files or separate areas of a large block of storage managed by the DBMS.

Cust ID **Name** **City**

Record	Value	Record	Value	Record	Value
1	1222	1	ABC	1	Delhi
2	1893	2	DEF	2	Ldh
3	3737	3	IJK	3	Asr
4	2788	4	XYZ	4	Agra
5	5467	5	MNP	5	Chd

TABLE 2[10]

There are many databases which are categorized in column oriented database [14]:

- Apache HBase
- MonetDB
- Hyper table
- Mnesia
- Apache Accumlo
- LucidDB
- Sybase
- Vertica(HP)
- Cassandra
- Google's Big table

Some of the column oriented databases using column approach are described below:

3.1 MonetDB

MonetDB is an open source high-performance database management system developed at the National Research Institute for Mathematics and Computer Science in the Netherlands. It was designed to provide high performance on complex queries against large databases, e.g. combining tables with hundreds of columns and multi-million rows. MonetDB has been successfully applied in high-performance applications for data mining, OLAP, XML Query, text and multimedia retrieval. MonetDB is one of the first database systems to focus its query optimization effort on exploiting CPU caches [11].

3.2 LucidDB

LucidDB tables are column store tables.Data in LucidDB is stored in Operating System in a file name as db.dat.Column store table consists of set of clusters. Each column maps to single cluster. A single cluster page, therefore, stores the values for a specific set of RowIDs for all columns in that cluster. Each cluster also has associated with it a Btree index. The Btree index maps rid values to pageIds. The rid values correspond to the first rid value stored on each page within a cluster, and the cluster pages are identified by their pageIds[11].

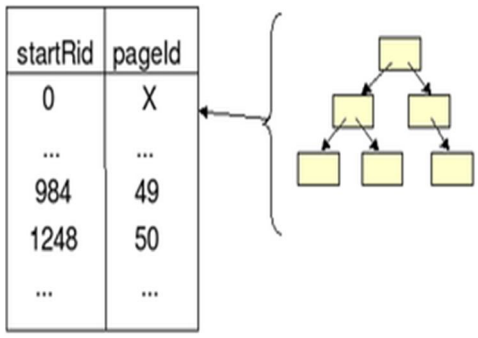

Fig.3.2.1: Rid -to –pageId Btree map[11]

It has advantages using lucidDB is that column values, by default, within a cluster page, are stored in a compressed format, which allows LucidDB to minimize storage requirements. Instead of storing each column value for every rid value on a page, it stores just the unique column values and then associates with bit-encoded vector.

3.3 Sybase IQ

Sybase IQ was the only commercially available column-oriented database for many years used by SAP Company. It deals with data partitioned, index based storage technology. The Sybase IQ Very Large Data Base (VLDB) provides partitioning and placement where a table can have a specified column partition key with value ranges. This partition allows data that should be grouped together and separates data where they should be separated. Use for business intelligence, analytics and data warehousing solutions on any standard hardware and operating system [12]. Key features are:-

- Web enabled analytics
- Communications & Security
- Fast Data Loading

- Query Engine supporting Full Text Search
- Column Indexing Sub System
- Column Storage Processor
- Multiplex Grid Architecture.

3.4 VERTICA (HP)

Vertica (HP) is an another type of column oriented databases recently used by Hewlett Packard (HP) used to enable data values having high performance real-time analytics needs. With extensive data loading, queries, columnar storage, MPP (massively parallel processing) and data compression features. The Vertica analytics platform uses transformation partitioning to specify which rows belong together and parallelism for speed according to its elasticity, scale, performance, and simplicity and provides optimal query execution plans acc to it [12].

ADVANTAGES OF COLUMN ORIENTED DATABASES:

- High performance on aggregation queries (like COUNT, SUM, AVG, MIN, MAX)[12].
- Highly efficient data compression and/or partitioning [12].
- True scalability and fast data loading for Big Data [12].
- Provides hard disk access and reduce disk space [13].
- Improved Bandwidth Utilization.
- Improved Code Pipelining [11].

DISADVANTAGES OF COLUMN ORIENTED DATABASES [12]:

- Transactions are to be avoided or just not supported.
- Queries with table joins can reduce high performance.
- Record updates and deletes reduce storage efficiency.
- Effective partitioning/indexing schemes can be difficult to design.
- Increased Disk Seek Time [11].

4. CONCLUSION

We have study the document oriented databases and column oriented databases, list various databases which come under these categories and also give their advantages and disadvantages as compared to relational databases. So, we concluded that these databases are very useful in handling large amount of data and are highly scalable and can handle semi structured and structured data in very efficient manner and also many other advantages over relational databases which makes them more useful and popular in future.

5. REFERENCES

[1] C. Strauch, "NoSQL Databases," February2011. [Online]. Available: http://www.christofstrauch.de/nosqldbs.pdf.

[2] Neal Leavitt, " Will NoSQL Databases Live Up to Their Promise?" IEEE Computer Society0018-9162/10/$26.00 © 2010 IEEE.

[3] MongoDB. Mongodb. [Online]. Available: http:// en.wikipedia.org/wiki/Mongodb/

[4] Apache CouchDB [online]. Available: http://couchdb.apache.org/.

[5] Apache CouchDB [online]. Available: http://en.wikipedia.org/wiki/Apache_Couch DB.

[6] Apache RavenDB [online]. Available: http://ravendb.net/.

[7] Okman, L.; Gal-Oz, N.; Gonen, Y.; Gudes, E.; Abramov, J. , "Security Issues in NoSQL Databases," *Trust, Security and Privacy in Computing and Communications (TrustCom),2011 IEEE 10th International Conference on* ,vol., no., pp.541-547, 16-18 Nov. 2011 doi:10.1109/TrustCom.2011.70.

[8]Document oriented databases [online].Available:http://en.wikipedia.org/ wiki/Document_oriented_database.

[9]Abadi j.Daniel, Maddan R.Samuel, Hachem Nabil, "*ColumnStores vs. RowStores: How Different Are They Really?*" SIGMOD'08, June 9–12, 2008, Vancouver, BC, Canada.

[10] Bhatia Anuradha, Patil Shefali, "*Column oriented DBMS an approach*" International Journal of Computer Science & Communication Networks, Vol 1(2), Nov 2011.

[11] Venkat Rakesh "Column oriented databases vs Row oriented databases"[ppt].

[12] Column oriented database technologies [online]: Available: http://dbbest.com/blog/column-oriented-database-technologies/.

[13] Rise of column oriented databases [online].Available:http://slideshare.net/srudra25/rise-of-column-oriented-database/.

[14] NOSQL databases [online]. Available: http://nosql-databases.org/