

New Database Constraints and Modern Applications

Author

Naseer Ganiee

Research Scholar

Department of Computer Science Engineering & IT, JIT University

Email: naseer@rationaltabs.com

-----ABSTRACT-----

Database is an essential part of application and the relational model is most widely used and acceptable but in present era we need new data management requirements for modern applications that are not supported by traditional databases. To handle these new data management requirements we need some new database approaches like NOSQL approach which is open source, distributed and supports horizontal scaling.

The aim of this paper is to identify the new data management requirements of modern applications and how NOSQL approach is efficient for modern applications.

Keywords – NOSQL, RDBMS, Cloud Computing, CAP, BASE.

1. INTRODUCTION

Enterprises use different types of databases for data management operations and relational databases are most acceptable and used by developers from decades. The use of database for a particular application also changes with time and in present era applications need some new data management requirements that are not furnished by traditional databases. Applications based on cloud computing and social networking support millions of users and require new data management facilities like horizontal scaling that enable us to dynamically add and remove nodes instead of upgrading the existing machines. Keeping these new data management considerations in mind experts proposed a new database architecture known as NOSQL. NOSQL approach is a strong competitor to the relational model because it supports high scalability. The famous CAP theorem describes that not any database system supports all the three attributes but only two of three is possible. Relational databases support only consistency and partition tolerance properties and the NOSQL databases support the last two means availability and partition tolerance for high availability and partitioning of data and give origin to a new model known as BASE (Basically Available, Soft-state, Eventually consistent). NOSQL database are successfully deployed by Google, Facebook, Amazon for scalability and high availability of data.

2. NOSQL CATEGORIES

There are different categories of NOSQL databases and are describes as under:

2.1 KEY-VALUE STORES

The key-value stores are the simple NOSQL databases where a key points a value that is normally a string. The lookup or indexing process is used to find the values associated with

a key. The famous products under this category are SimpleDB, BigTable, Redis, Tuplespace and Memcachedb.

2.2 DOCUMENT STORES

This NOSQL category is designed for applications that use documents. This database can hold documents of any length and document content is used to for retrieval process. Document stores are like XML documents, good alternative for services like messages, blogs and comments. The famous products under this category include MongoDB, CouchDB, Apache Jackrabbit, Terrastore and OrientDB.

2.3 WIDE COLUMN STORES

This type of NOSQL category is comparable to a column oriented database which stores data by column not by row. These databases contain one extendable column not a table with rows and columns like in relational model. These databases also manage key-value pairs, the key are referred as columns and are stored in structures known as tables. The famous products under this category include Hadoop, Cassandra and Hypertable.

2.4 GRAPH DATABASES

Nodes, edges and properties are used to represent and store data in graph databases. The nodes are like objects and edges represent the relationship between the objects. The index free adjacency technique is used for traverse records or in other words each node contains a pointer which points to the adjacent node. The famous products under this category include Neo4J, InfoGrid, Sonex, VertexDB and AllegroGraph.

3. DATABASE CONSIDERATIONS FOR MODERN APPLICATIONS

There are some important factors to keep in mind when choosing a database for modern applications:

3.1 SCALABILITY

Scalability is a key standard for databases used in modern applications that help in balance the resources with full functionality. The database scaling needs to be on demand because it is very difficult to identify when an application needs to scale and also it is possible to lower the resource utilization when system is inoperative and the important thing is to scale a database without disturbing an application. NOSQL databases are known for scalability, in Cassandra NOSQL database nodes can be added impeccably as we need more capacity. The new resource utilization is an automatic process in the cluster. MongoDB also has a very good scalability options like automatic sharding or partitioning of data across servers, distributed reads and writes. In Couchbase nodes can be added and removed with a click from the cluster with no changes to the application and to achieve faster data access cross data center replication is also possible.

3.2 PERFORMANCE

A database needs to provide low latencies in spite of the data size consistently that is actually the performance. In NOSQL databases data is shared across nodes present in the cluster that results in low read and write latency.

3.3 AVAILABILITY

Modern applications need highly available databases. To achieve high database availability there is a need of system that perform rare tasks like, easy to setup the node in the

cluster, easy removal of node for maintenance, online upgrades, regular backups and most important thing is to support disaster recovery in case there is a problem with the entire data center. In Cassandra when a node fails to store a write operation is automatically routed to another node that store the write request with a message known as hint that contains information of the node that fails to store the write operation. The hint contained node examines the entire cluster for the recovery of the node and if the failed node is back active, the recovered node gets the hint message back for proper write operations at proper locations. From Cassandra case study it is clear that NOSQL databases are perfect for modern applications with high scalability, performance and availability of data.

3.4 SIMPLE STRUCTURE

Relational databases have no flexible schema like NOSQL databases and if application changes database schema also changes. NOSQL databases have very flexible schema and when new fields are added there is no need to modify the existing structure. This results in very simple structure.

4 CONCLUSION

This paper describes the databases of modern applications, the importance of NOSQL databases in modern application development. This paper also describes the different NOSQL categories and the major factors like scalability, availability and performance to keep in mind when choosing a database for modern applications.

REFERENCES

JOURNAL PAPERS

[1] Ameya Nayak, Anil Poriya, Dikshay Poojary (2013), Type of NOSQL Databases and its Comparison with Relational databases, International Journal of Applied Information Systems, Vol.5, No.4

[2] Indu Arora, Dr Anu Gupta (2012), Cloud Databases: A Paradigm Shift in Databases, International Journal of Computer Sciences, Vol.9, No.3

[3] J M Tauro, Aravindh S, Shreeharsha A.B (2012), Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases, International Journal of Computer Applications, Vol. 48, No. 20

[4] Nishtha Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria, Dishant Gosain (2012), A Survey of Relational and Non-Relational Database, International Journal of Engineering Research & Technology, Vol. 1

[5] Rabi Prasad, Manas Ranjan Patra, Suresh Chandra (2011), RDBMS to NOSQL: Reviewing Some Next Generation Non-Relational Database's, International Journal of Advanced Engineering Sciences and Technologies, Vol. 11

[6] Uma Bhat, Shraddha Jadhav (2010), Moving Towards Non-Relational Databases, International Journal of Computer Applications, Vol. 1, No. 13

[7] Vatika Sharma, Meenu Dave (2012), SQL and NOSQL Databases, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2

WHITEPAPRES

[1] CouchBase (2011), NOSQL Database Technology

[2] DataStax Corporation (2013), Evaluating Apache Cassandra as a Cloud Database

[3] DataStax Corporation (2013), Implementing a NOSQL Strategy

[4] DataStax Corporation (2011), NOSQL in the Enterprise

IJSP