An Introduction to NoSQL Databases and Applications

Fangyuan Huang
University of Rochester
fhuang5@ur.rochester.edu

Ning Zhang
University of Rochester
nzhang15@ur.rochester.edu

ABSTRACT

This paper introduces mainstream NOSQL databases that can be divided into four categories in terms of database structure, namely, Column-Oriented, Document-Oriented, Key-Value Store, and Graph-Oriented. It investigates the different trade-off strategies over choosing two out of the three main database properties (Consistency, Availability, and Partition Tolerance) according to the CAP theorem. The paper further explores and compares the characteristics of four major NoSQL databases from four different categories and their applications in industries, specifically, BigTable, MongoDB, DynamoDB, and Neo4j.

1. INTRODUCTION

The term NOSQL is generally interpreted as "Not Only SQL"[1], which implies that it contains a few basic properties of traditional relational database systems (SQL) with some extra modifications. Unlike SQL databases that emphasize on consistency, NoSQL databases often accept a weaker consistency level to guarantee availability and partition tolerance.[2]

To understand NOSQL database, it is helpful to start with Standard Query Language (SQL) database systems. The SQL databases are designed under the structures of tables, columns, rows, upon which data are created, read, updated or deleted. SQL databases have strict requirement on the structure of the data that they are managing such that they are able to meet the requirement of immediate consistency and reliability.

SQL databases meet most needs raised by businesses up to 1990s, but the whole story changed its direction in the advent of Web 2.0 era in 2000s, in which surfing on Facebook and Twitter, driving with Google Maps and Waze, and shopping on Amazon and Ebay had become daily life of people. According to a survey conducted by International Data Corporation, out of billions of gigabytes of data that are generated by businesses and people every day around 80 percent are unstructured.[3]

With hundreds of gigabytes of unstructured data generated by commercial companies, the requirements of handling data in web-scale and big data systems have caused increasing needs for database systems with large scalability and high performance, known as NoSQL database systems.

2. Problem Statement

NoSQL database has been created and adapted for decades to meet the new challenges for database systems at the 21st century. With an increasing amount of successful implementation of NoSQL database, it is important to know why NoSQL database systems have achieved big success, how NoSQL database systems work, and the difference between popular NoSQL systems.

3. CAP Theorem

CAP theorem, proposed by Dr. Eric Brewer in 2000, addresses three important properties of a distributed system, namely Consistency, Availability and Partition-tolerance. Consistency means that the nodes will have copies of a replicated data items visible for various transactions; availability means that each read or write request for a data item will either be processed successfully or will receive a message that the operation cannot be completed; partition-tolerance means that the system can continue operating if the network connecting the nodes has a fault that results in two or more partitions, where the nodes in each partition can only communicate among each other[4].

The CAP theorem states that it is impossible for any distributed system to satisfy more than two properties. In another word, every distributed system has to make a trade-off strategy on which two CAP properties it prioritizes. Based on its various strategies, a distributed system can be characterized as either an AP, CP, or CA system. RDBMs are all categorized as CA system, while NoSQL systems may either be in AP or CP system.

4. What is NoSQL Database

NoSQL is a type of database management system that emphasizes high availability, scalability, and high performance. According to CAP theorem, it is impossible to guarantee all three of the desirable properties at the same time in a distributed system with data replication.

NoSQL is one approach where a weaker consistency level is often accepted to guarantee the other two properties. In practice, a form of consistency known as eventual consistency is often adopted in NoSQL systems. However, some NoSQL databases adopt extra approaches and techniques to make the database comply with ACID model, such as Neo4j.

Scalability: In NoSQL systems, horizontal scalability is generally used to achieve scalability. Horizontal scalability refers to expanding the system by adding more nodes for data storage and processing as the volume of data grows.
Availability: One motivation of NoSQL is the increasing demands of continuous system availability. To accomplish this, data is replicated over many nodes, so as to ensure data availability even if some nodes fail.
Performance: In many NoSQL applications, it is necessary to find individual records from millions of data records. To achieve high performance, two techniques are frequently used in NoSQL systems, called hashing and range partitioning on keys.
Consistency: The eventual consistency is usually applied and then realized by two different replication models, called master-slave and master-master replication. Master-slave model has better consistency and master-master model has better accessibility. [5]

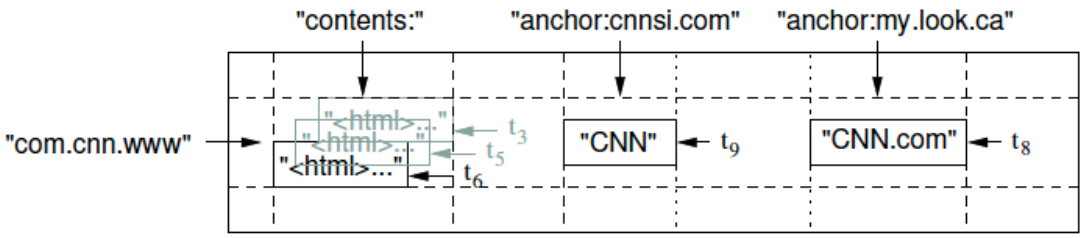

Figure 1: A slice of an example table that stores Web pages. The row name is a reversed URL. The contents column family contains the page contents, and the anchor column family contains the text of any anchors that reference the page. CNN's home page is referenced by both the Sports Illustrated and the MY-look home pages, so the row contains columns named anchor:cnnsi.com and anchor:my.look.ca. Each anchor cell has one version; the contents column has three versions, at timestamps t3, t5, and t6. [6]

5. Four Main Categories

Here we introduce the application of NoSQL based on four different database structures, namely column-oriented, document-oriented, key-value store, and graph-oriented.

5.1 Column-Oriented

A column-oriented DBMS is a database management system that stores data tables as columns rather than rows. The column-oriented systems usually is multidimensional and have several components: table name, row key, column, and timestamp. Column is typically composed of two components: column family and column qualifier. By storing data this way, the database can more precisely access the data it needs to answer a query rather than scanning the discarding unwanted data in rows. As a result, query performance is often increased, especially in very large data sets.[7]

One example is BigTable by Google. A Bigtable is a high performance column-oriented distributed NoSQL database service for large analytical and operational workloads (in petabytes). According to Google, Bigtable has achieved wide applicability, scalability, high performance, and high availability.

In Bigtable, data is indexed using row and column names that can be arbitrary strings. A bigtable can also be understood as a sparse, distributed, persistent multidimensional sorted map. This map is indexed by a row key, column key, and a timestamp. The reason BigTable can be qualified as a column-oriented database is because they applied column keys, which are named using syntax family:qualifier, and grouped into sets called column families. Column families form the basic unit of access control for BigTable system.

The Bigtable system is widely used by more than sixty Google products and projects, including Google Analytics, Google Finance, Orkut, Writely, etc. [8]

5.2 Document-Oriented

Document-oriented NoSQL systems typically store data as collections of similar documents. All documents are specified as self-describing data, thus there's no requirement to specify a schema. Although documents in a collection should be similar, they can have different data attributes. A popular language to specify documents is JSON (JavaScript Object Notation.) [9]

For example, MongoDB is a schema-free document-oriented NoSQL database system, where documents are stored in BSON (binary JSON.) It's developed by MongoDB Inc., formerly known as 10gen Inc., and released initially in 2009. In MongoDB, documents with similar structure are organized as collections, and each document can have different fields. A new field can be added to a document without having to update the central system catalog and affecting other documents in the system. This technique is called dynamic schema. Since MongoDB doesn't rely on pre-fixed schema, this distinguish itself from relational database structure and allow it to have a flexible scalability. MongoDB has an interactive JavaScript shell that can be used to issue all the supported commands including administrative processes.

MongoDB is an open source software and has powerful supplement on query model, indexing, sharding, map-reduce, and others. MongoDB is the most popular document-oriented database and has been used by plenty companies such as Facebook, Craigslist, eBay, and Foursquare.[10]

5.3 Key-value Stores

As the name of this type of NOSQL databases indicates, this data model stores data in the key-value structure. The key is a unique object associated with a data item and is used to locate the data item on physical disks at high speed.

The value is the data item itself. This key-value structure accomplishes high-speed read and write processing in real time operations. Furthermore, data can be horizontally partitioned and replicated across a cluster to get scalability and availability. However, complex query languages can not be implemented on the structure. [11]

For example, DynamoDB is a cloud based NoSQL database based on key-value store techniques and implemented by Amazon. DynamoDB is schemaless except for primary key, which means there is no predefined schema such as table name, column name, and data types. An attribute in an item is a name-value pair that can be single-valued or multi-valued set.

DynamoDB is highly scalable with low latency and high performance because of the use of key value pairs that are designed with simpler and less constrained data models than RDBMS.[12] It's high performance is realized by two different types of primary keys. First is Hash Type. This type of key is made up of just hash attribute.

An unordered index is built by DynamoDB based on this attribute. Second is Hash and Range Type, where a primary key is made up of two attributes: a hash attribute and a range attribute. Besides an unordered index built based on hash attributes, a sorted range index based on the range attribute is also created.[13]

DynamoDB has been used by Amazon, EA Sports, Nordstorm, Shazam, New York Times and many others.

	Consistency	Availability	Partitioning Scheme
BigTable	Eventual consistency	High Availability	Range Based
MongoDB	Eventual consistency	High Availability	Consistent Hashing
DynamoDB	Eventual consistency	High Availability	Consistent Hashing
Neo4j	Immediate consistency	High Availability	Not supported

Table 1: comparison between four popular NoSQL database systems based on CAP theorem

5.4 Graph-Oriented

The design of graph-oriented database takes advantage of the simple structure of node-edge relationship, and develops it into a well defined database system.

In a graph database, the data is represented as a collection of nodes and edges. Both nodes and edges can be labeled to indicate the types of entities and relationship they represent. As in RDBMs, data relationships are typically stored in data and retrieved by invoking SQL keyword 'join'. However, when a database has many relationships, its performance gets slow as joining tables incur high computational costs. In contrast, graph databases explicitly store relationships such that they allow simple and fast retrieval of complex hierarchical structures.[14]

An example of graph databases is Neo4j. Neo4j is a graph oriented database developed by Neo Technology Inc. and initially released in 2007. Neo4j is an open source system implemented in Java. The system is composed of two elements, a node and a relationship. Each node represents an entity and each relationship signifies association between two nodes. Both nodes and relationships can have properties.

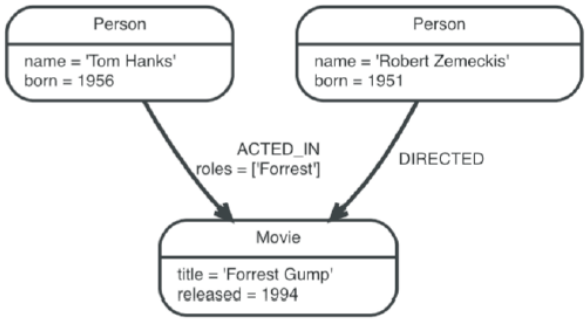

Fig. 2: An example of nodes and relationship in Neo4j. Where 'Person' and 'Movie' are optional labels for node. [15]

In Neo4j, nodes with same label are grouped into a collection that identifies a subset of the nodes in the database graph for querying purposes. Relationships are directed, and each relationship must have a start node and end node as well as a relationship type.

Neo4j has a graph visualization interface, so that a subset of the nodes and edges in a database graph can be displayed as a graph. It implements master-slave replication method to ensure consistency. The data and indexes are fully replicated on each node in the cluster.[16] Various ways of synchronizing the data between master and slave nodes can be configured in the distributed cluster. Neo4j has many industry customers such as eBay, Walmart, Cisco, Lufthansa, and Tom Tom.

6. Conclusion

This paper described the main characteristics and realization of four different categories of NoSQL systems. Different trade-off strategies on CAP were demonstrated. The results were summarized in table 1 (above).

Based on the information collected, we conclude that the choice of database systems highly depend on the specific application domain and the required functionalities. Sometimes it is impossible to clearly identify the best database choice given the complexity of application. This paper shall have given readers a brief understanding about NoSQL database systems.

7. Acknowledgement

This paper would not have been made possible without the help and encouragement of our 'database' class instructor Dr. Tamal Tanu Biswas.

8. REFERENCES

[1] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 883.

[2] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 889.

[3] Chakraborty, Goutam. "Analysis of Unstructured Data: Application of Text Analytics and Sentiment Mining", SAS. Retrieved June 24, 2016.

[4] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 889.

[5] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 886.

[6] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 900-901.

[7] Bigtable: a distributed storage system for structured data. Fay Chang, et al. OSDI'06 Seventh Symposium on Operating System Design and Implementation, Seattle, WA, November 2016. Section 2.

[8] Bigtable: a distributed storage system for structured data. Fay Chang, et al. OSDI'06 Seventh Symposium on Operating System Design and Implementation, Seattle, WA, November 2016. Section 8.

[9] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 890-891.

[10] GC, Deepak, "A Critical Comparison of NOSQL Databases in the Context of ACID and BASE" (2016). Culminating Projects in Information Assurance. P56-57.

[11] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 895-897

[12] Doble, W. (2014, September 1). Comparing the Use of Amazon DynamoDB and Apache HBase for NoSQL. Retrieved from Amazon Web Services Docs.

[13] GC, Deepak, "A Critical Comparison of NOSQL Databases in the Context of ACID and BASE" (2016). Culminating Projects in Information Assurance. P27-29.

[14] Elmasri, Ramez, and Sham Navathe. 2016. Fundamentals of Database Systems. Page 903-905.

[15] Why Graph Databases? – Neo4j Graph Databse. (2015) Retrieved from Neo4j: http://wwww.webcitation.org/6hAHWGLqa

[16] GC, Deepak, "A Critical Comparison of NOSQL Databases in the Context of ACID and BASE" (2016). Culminating Projects in Information Assurance. P41-42.