Andrew Gordon

Dr.Johnson

CS 300 Introduction to Database Management Systems

3/9/2022

NoSQL and Graph Database Overview- Assignment 1

NoSQL is commonly referred to as a non-relational database. According to an article on MongoDB entitled "What is NoSQL?", some people believe the term NoSQL means "non-SQL" while others believe it to mean "not only SQL". NoSQL databases do not store data like how a relationship database management system does. There are different types of NoSQL databases based on their data model. The three main types of noSQL databases according to MongoDB are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.("What is NoSQL? NoSQL Databases Explained.") According to an IBM education tutorial called "NoSQL databases", noSQL databases can offer more "flexibility" than traditional relational databases. The IBM tutorial goes on to elaborate that NoSQL databases store data within one structure, such as a JSON document. Since noSQL does not require a schema, it allows for rapid scalability to house large unstructured data sets. NoSqL is a distributed database, meaning that, data is copied and stored on different servers in different geographical locations, or the data is stored over a network of connected computers.("NoSQL-Databases") This ensures that if one of the servers that stores this data goes down, then the database can continue to run.

To understand the types of problems NoSQL databases were created to solve its important to look back at the history of databases in order to get the proper context of the problem. In the 1970s a computer scientist by the name of Edgar Cobb created the idea of relational database management systems. In a paper Cobb wrote , Cobb expressed this new idea on how to store, analyze, and process business information. The data being stored would be put in a column and row form. This process was a massive spring in innovation and made the traditional file-based system seem obsolete. In the file -based system each program defines and manages its own data. The file-based system was a grossly inefficient system, where duplicate data was often stored on different files. There was no central control. The relational database management system(RDMS) fixed this. After this, IBM created the SQL language this made it possible to query data from a RDMS very quickly with easy to type commands. Still to this day, RDMS are used in many companies around the world. According to Jonathon Lacefield, in an article entitled "The Evolution of NoSQL", he explains that there are a few critical areas in which RDMS are not effective. One major limitation is that they are only capable of processing small amounts of structured data—like names or zip codes(Lacefield).

Lacefield goes onto explain that with the introduction of the internet, and the huge amounts of data coming from the new internet users, that relational database management systems became overwhelmed. Lacefield comments on the effects of the internet on the RDMS model, saying that:

> the RDBMS model either broke or became very challenging to shard correctly. Relational databases also required a tremendous amount of maintenance. A database of a few thousand objects may handle things decently, but as you scale up, performance declines. This is a big problem—especially considering the massive volume of unstructured data that is being generated on a daily basis.(Lacefield)

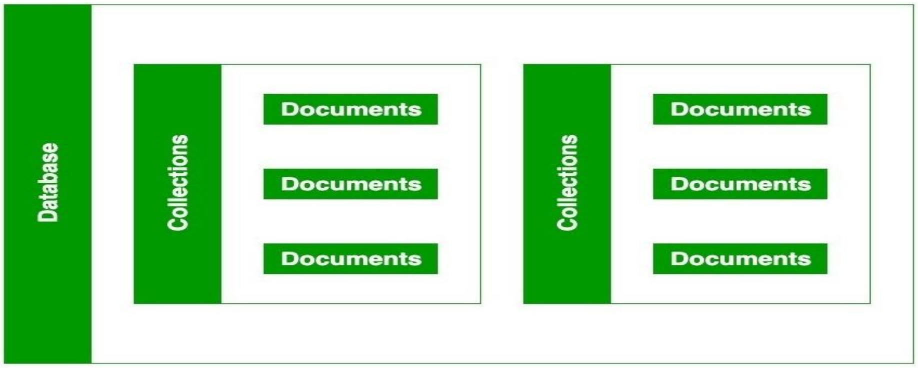
With the increase in volume of unstructured data NoSQL positions itself as database system that could be a more effective model than the RDMS in this regard

Some other advantages of the NoSQL system are that these systems do well when they have the ability to scale outward to new nodes. This is much more efficient and affordable than trying to scale out with relational database management systems. This brings up the next point is that when data volume continuously increased the cost associated with upgraded and maintaining RDMS servers increases as well. NoSQL systems leverage commodity server clusters which utilizes large numbers of readily available servers for parallel computing to obtaining the greatest amount of useful computations for the least cost. Lacefield also mentions that NoSQL database systems are much easier to maintain in comparison to RDMS. Some of these systems are equipped with auto- repair capabilities. (Lacefield)

In the article “NoSQL—The Bad Parts”, written by Rajeev Sharma, the author points out some of the disadvantages of a NoSQL database. Some of the important disadvantages include the lack of JSON, lack of standardization, lack of maturity, and its narrow focus. (Sharma) JSON, according to an article written by Priya Pedamaker entitled “What is JSON?”, explains that “ JSON can be defined as a Java Script Object Notation file format that is used for sending, receiving and storing the data from the same or different systems in a network.”(Pedamaker) According to Sharma the lack of incorporating JSON is huge because many engineers typically use JSON to solve normalization and data representation issues. Sharma continues saying that there are really no reliable standards for NoSQL databases currently. Meaning that no two databases are equal in terms or performance and standardizations. Support could end up spotty in comparison to RDMS. This is shown in the lack of core developers and administrators with core knowledge in NoSQL. Although NoSQL is known to be administrator or developer friendly this means nothing if administrators do not know how to use the technology in the first place. The last major issue that NoSQL databases can’t address is the narrow focus that NoSQL has on scalability and storing data, the issue arises when transactions are needed in the database.(ex: deleting, modifying data) When transactions or queries are needed then RDMS are still superior.(Sharma)

Some Examples of NoSQL databases would be MongoDB, Redis,Amazon DynamoDB. According to geeksforgeeks.org, the website explains that MongoDB works by giving you an environment that a user can create a server that the user can create multiple databases on, the data is stored in collections and documents. Hence the database, collection, and documents are related to each other as shown below:(“What is MongoDB – Working and Features”)

.



www.geeksforgeeks.org

Another popular NoSQL database is Redis. According to Alexander Nnakwue , a software engineer, he states that Redis is an open-source non-relational kind of database and caching server. It works by mapping keys to values with a predefined data model. (Nankwue) Another example of a NOSQL database is DynamoDB by Amazon. Every user on the AWS platform can access DynamoDB. According to serveress.com DynamoDB like many other databases, stores data in tables. Each table has a set of items, and each item has a set of fields or attributes. Each table must have a primary key, and this key can be a single attribute or a combination of two. You can reference specific items in a table by using the primary key, or by developing indices and using keys from those indices. The data is distributed across many machines, this means scalability, but also means you cannot connect to a database host a query your data directly("Amazon DynamDB and Serverless")

The development of graph data stemmed from the lackluster ability to analyze the relationships between attributes and data fields in the RDMS and a traditional NoSQL database. A graph database is a type of NoSQL database management system, and it gives users the flexibility to store nodes and relationships instead of tables or documents. According to an article written by Todd Blaushaka, entitled "What is a Graph Database and Why Should You Care", Blaushaka explores the reasons why graph databases came about and are so popular. In order to analyze the relationships between business entities (customers, order, product, etc..) in relational database management system databases require join queries and become computationally more expensive as data volume grows. NoSQL requires, according to Blaushaka, that all of its data is stored in a single table. In order to evaluate a relationship between different business entities, users would need to scan millions and even billions of rows depending on how large the database is. (Blaushaka) This makes NoSQL databases very cumbersome to use to analyze relationships. Graph databases are built for the sole responsibility to analyze relationships between different entities. Graph databases do not require multiple tables joins or multiple scans across a large table, thus making graphs very efficient in studying relationships between entities. Other than making it extremely easy to analyze relationships between entities, what are other advantages to graph databases? As companies grow and business changes commence, the need to change how a business needs to store its data will need to change. Graph databases offer the flexibility of updating or creating new nodes or delete existing ones all while maintaining the functionality of the database. In an article "Reasons to Use a Graph Database" written by Jason Camaye lists one other reason advantage of the graph database management system. He states:

> A graph database management system has advanced problem-solving capabilities. Using specific keywords in queries, you can easily get the right information within the blink of an eye. A targeted and semantic query algorithm doesn't touch or alter unrelated information in the system. Hence, it is an excellent choice for retrieving real-time data from a wide, analytical set of databases. (Camaye)

Some disadvantages of graph DBMS are that there is no standardized query language. The language is dependent on the platform used. Another disadvantage is that graph databases are not appropriate for transactional-based systems. Graph databases are not good at handling queries that extend the entire database. Much like NoSQL database management systems, another downside to implementing graphs databases is that the user-base is small, thus making it hard to find support when a user encounters a problem.

One of the more popular graph databases in the market right now is the is called Neo4J. According to their website, Neo4J is an open-source native graph database. Neo4J works by connecting each data record or node contains pointers to other records. These pointers are called relationships. All the information needed to find the next node in the sequence is available in the node itself. The native storage is a connected graph. Because of this Neo4J does not need to compute the relationships between a user's data at query time its already built into the database architecture. According to the Neo4J website, this database architecture speeds up data retrieval and queries, saying that "the queries of densely connected data are orders of magnitude faster. Other databases don't save direct pointers between records. They need to compute connections by searching through a separate data structure, called an index. This makes them inherently slower than Neo4j for relationship-intensive queries." (Neo4J) Neo4J claims that this database is used extensively for governments, businesses, and other organizations to detect fraud, enhance artificial intelligence technology, manage supply chains, and come to a better understanding of the relationships between data.

Another key NoSQL graph database management system in the market right now is OrientDB. OrientDB, unlike Neo4J, is a multi-model NoSQL DBMS that supports of graph, documents, key-value, and object-oriented storage. What is a multi-model DBMS?A multi-model database is a database that allows for the support of multiple data models that are then integrated in the backend. OrientDB can utilize a graph database structure similar to

Neo4J, but unlike Neo4J, it supports the integration of document, key-value, and object-oriented storage that are found are in other typical NoSQL database management systems.

Works Cited

IBM Cloud Education. “Nosql-Databases.” *IBM*, 9 Aug. 2019, https://www.ibm.com/cloud/learn/nosql-databases.

-, Jason Camaya, et al. “Reasons to Use Graph Database.” *Proche*, 29 Aug. 2021, https://www.theproche.com/2021/08/29/reasons-to-use-graph-database/.

“Amazon Dynamodb and Serverless.” *Serverless*, https://www.serverless.com/guides/dynamodb.

“A Guide to Fully Understanding Redis.” *LogRocket Blog*, 16 Jan. 2020, https://blog.logrocket.com/guide-to-fully-understanding-redis/.

Lacefield, Jonathan. “The Evolution of Nosql.” *DataStax*, DataStax, 12 Aug. 2021, https://www.datastax.com/blog/evolution-nosql.

Sharma, Rajeev. “NoSQL Database - the Bad Parts.” *Ezeelive.com*, Publisher Name Ezeelive Technologies Publisher Logo, 24 May 2018, https://ezeelive.com/nosql-database-bad-parts/#:~:text=NoSQL%20Database%20%E2%80%93%20the%20bad%20parts%201%20 1.,...%207%207.%20Removal%20of%20Duplicate%20Data%3A%20.

“What Is JSON?: How It Works: Advantages & Disadvantages: Examples.” *EDUCBA*, 5 Feb. 2022, https://www.educba.com/what-is-json/.

“What Is Mongodb - Working and Features.” *GeeksforGeeks*, 6 June 2021, https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/.

“What Is neo4j?” *Neo4j Graph Data Platform*, 13 Mar. 2022, https://neo4j.com/videos/what-is-neo4j-3/.

“What Is Nosql? NoSQL Databases Explained.” *MongoDB*, https://www.mongodb.com/nosql-explained.