CS 300-ON Database Management Systems

Greg Gutierrez

9/5/2022

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 relational database management system does. There are different types of NoSQL databases based on their data model. According to MongoDB, the main types of NoSQL databases are document, key-value, wide-column, and graph. They provide flexible schemas and scale quickly with large amounts of data and high user loads. ("What is NoSQL? NoSQL Databases Explained.") According to an online education tutorial called "NoSQL database," from IBM, NoSQL databases can offer more "flexibility" than traditional relational databases. The tutorial elaborates 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 NoSQL attribute database, meaning that data is copied and stored on different servers in different geographical locations, or the data is held over a network of connected computers. ("NoSQL-Databases") This ensures that if one of the servers that stores this data goes down, the database can continue, running stand types of problems NoSQL databases were created to solve. It's essential to look back at the history of databases to goer court on the issue.

In the 1970s, a computer scientist of the national database systems in a paper wrote the idea on how to store, analyze, and process storing, the process of and processing stored in a column and row form. This process was a massive spring in innovation and made the traditional file-based system seem obsolete. Each program system was grossly inefficient in the file-based system, where duplicate data was often on different files. There was no central control. The relational database management system (RDMS) fixed this. After this, IBM created the SQL language,

which made it possible to query data from an R, which makes it possible to query data from an RDMS very quickly with easy-to-type.

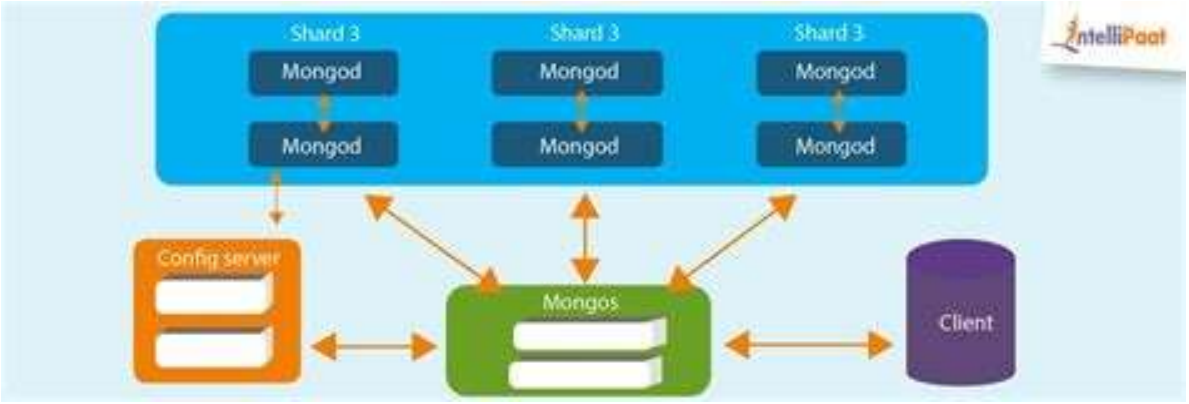
In an article entitled “The Evolution of NoSQL,” Jonathon Lacefield explains that there are a few critical steps in which RDMS are ineffective. One major limitation is inefficiency in processing small amounts of structured data—like names or zip codes (Lacefield). Lacefield explains that with the vast amounts of data from the new internet users, relational database management systems have become 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 generated daily. (Lacefield) The increase in the book of SQL positions as a database could be a more effective model than the RDMS. Some other advantages of the NoSQL system are that these systems do well when they can 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: when d: when data volume continuously increases, most associated with upgrading RDMS servers increases leverage commodity server clusters which utilize large numbers of utilizable servers for parallel computing to obtain the most excellent computations for the least cost. Lacefield also mentions that NoSQL database systems are much easier to maintain than RDMS.

Some of the thanes are equipped with auto-repair capabilities. (Lacefield) The author points out the disadvantages of a NoSQL database in auto-repairNoSQL—The Bad Parts” by Rajeev Sharma. Some of the crucial 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?” where he explains that “ JSON can be defined as a JavaScript 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 Sharm, the lack of incorporation of JSON is vast because many engineers typically use JSON to solve normalization and data representation issues. Sharma continues by saying there are currently no reliable standards for NoSQL databases, meaning that no two databases are equal in terms of 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 of 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. These arise when transactions are needed in the database. (Ex: deleting, modifying data).

When transactions or queries are then required, RDMS is still superior. (Sharma) Some Examples of NoSQL databases would be MongoDB, Redis, A, and Amazona DynamoDB. According to geeksforgeeks.org, the website explains that MongoDB works by giving you an environment where the users can create a seron which the user develops multi databases that are stored in collections and documents. Hence the data display records are related to each other, as shown in :("What is MongoDB – Working and Feat."



Another popular NoSQL database is Redis. Alexander Nnakwue, a software engineer, is an open-source non-relational database and caching server. It works by mapping keys to values with a predefined data model. (Nankwue) Another example of a NoSQL database is Dynamo DB by Amazon. Every user on the AWS platform can access DynamoDB. According to serveress.com, Dynamo, DB like, and many other databases, too, test data in tables. 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 using the primary key by developing indices and keys from those indices. The data is distributed across many of these scalabilities, but also, you cannot connect to a database host data on 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 NoSQL database manager that allows users to store nodes and relationships instead of tables or documents. Todd Blaushaka, in his work titled "What is a Graph Database and Why Should You Care," states why graph databases came about and are so popular. In Todd, the paper business becomes computationally more expensive

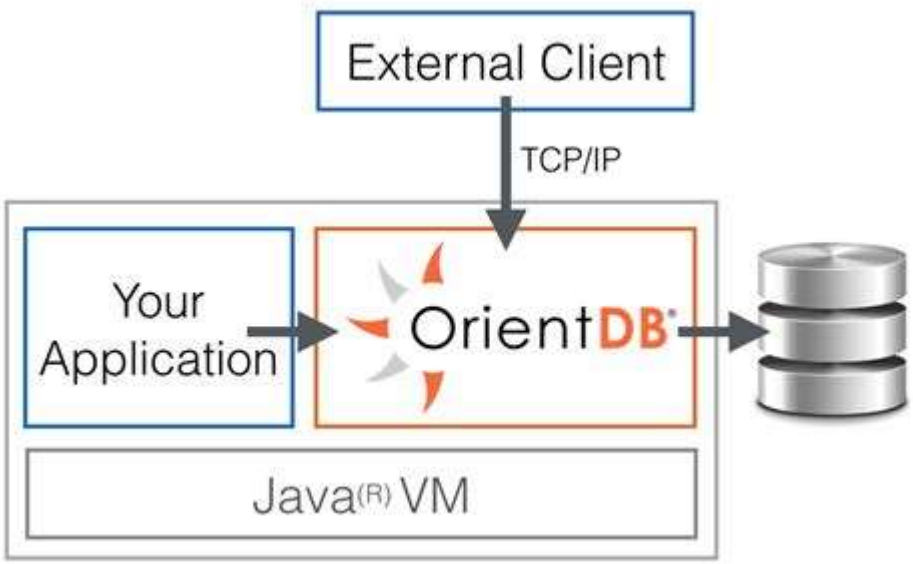
as data volume grows. NoSQL requires, according to Blaushaka, that all its data is stored in a single tablet to-go a relationship between different business entities; users would need to scan millions and even billions of rows depending on how extensive the database is. (Blaushaka) This makes NoSQL databases very cumbersome to use to analyze relationships. Graph databases are built with the sole responsibility of analyzing ships between different entities. Graph databases do not require multiple tables or 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 deleting existing ones while maintaining the database's functionality.

The article “Reasons to Use a Graph Database” by Jason Camaye lists one other 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 quickly correct the quick 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 comprehensive, analyst-comprehensive 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 called Neo4J. According to their website, Neo4J is an open-source native graph database. Neo4J connects each data record or node containing containers to other forms. 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. It’s 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 Orient DB.

Orient DB, unlike Neo4J, is a multi-model NoSQL DBMS that supports graphs 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 into the backend. Orient DB can utilize a graph database structure like Neo4J. Still, unlike Neo4J, it supports the integration of document, key-value, and object-oriented storage found in other typical NoSQL database management systems.



Reference

- IBM Cloud Education. "NoSQL-Databases." IBM, 9 Aug. 2019, https://www.ibm.com/cloud/learn/nosql-databases.
- Jason Camaya, "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." Log Rocket Blog, 16 Jan. 2020, https://blog.logrocket.com/guide-to-fully-understanding-redis/.
- Lacefield, Jonathan. "The Evolution of NoSQL." 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
- "What Is JSON? How It Works: Advantages & Disadvantages: Examples." EDUCBA, 5 Feb. 2022, https://www.educba.com/what-is-json/.
- "What Is Mongo dB - 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/explaine