Zac Sells

CS300

9/5/23

Dr. Johnson

Non-Rational Data Storage and Retrieval Systems

The technology industry is comprised of a countless amount of information, and this immense amount of information needs to be sufficiently stored. One way to help solve this problem is the use of databases. There are several types of database types that can be used. Two of the newest types are “NoSQL” and “graph” databases, which were developed in the pasty fifteen to twenty years. NoSQL and graph databases are considered non-rational and are less common rational databases that use SQL.

To understand what is meant by “NoSQL” and the difference between SQL and NoSQL, you must first understand how rational databases (SQL) are structured. Rational databases store large amounts of information in tables constructed of rows and columns. Rational databases are not limited to just a single table and may be constructed of several tables which may or may not be related in some way. To manage and retrieve the information from the tables a structured query language is used. According to MongoDB.com “SQL databases tend to have rigid, complex, tabular schemas and typically require expensive vertical scaling.” Popular examples include Oracle, MySQL, Microsoft SQL, and PostgreSQL. Non-rational databases such as “Not only SQL” or “NoSQL” store information in a variety of ways and may or may not support a structured query language. MongoDB.com defines a NoSQL as “non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.” Examples of NoSQL databases would be MongoDB, CouchDB, DynamoDB, and HBASE.

NoSQL databases were designed to streamline and simplify scaling of databases due to the ever-growing amount of information that needs to be stored. Due to their flexible and dynamic schemas NoSQL databases are perfect for applications in growing and changing environments. This allows the characteristics of the database to be modified quickly and often without affecting the use of an application. SQL databases have a predefined and rigid schema making it difficult to adapt and scale. NoSQL solve scaling issues as well due to their ability to be scaled horizontally. Horizontal scaling allows the database to scale out across a multitude of servers, meaning that when the database becomes insufficient more machines can be added to improve performance. According to bmcblogs.com “NoSQL databases are horizontally scalable, which means that they can handle increased traffic simply by adding more servers to the database. NoSQL databases have the ability to become larger and much more powerful, making them the preferred choice for large or constantly evolving data sets.” In contrast SQL databases scale vertically meaning they have a single machine or server that’s performance can be improved by upgrading components such as RAM, Storage (HHD, SSD, ect.), and CPU. According to touchstonesecurity.com small to mid-sized companies often use databases that require vertical scaling because it allows them to scale relatively quickly. The downside to this is that it poses a higher risk of downtime and outages. Whereas horizontal scaling allows changes to be made with relatively no downtime. According to MongoBD.com the initial cost of a horizontal scale-based database is significantly higher than a vertically based database, but as the need for scaling increases the cost of vertically scaling system becomes exponentially more expensive.

An example of a NoSQL product is MongoDB. MongoDB is a document-oriented databases meaning the database stores information in a data structure called a document instead of in rows and columns. MongoDB stores its documents in a file type called BSON (geeksforgeek.org, 2020). MongoDB is comprised of three main parts, the drivers, the shell, and the storage engine (geeksforgeek.org, 2020). Geeksforgeeks.org describes drivers as being “present on your server that are used to communicate with MongoDB. The drivers support by the MongoDB are C, C++, C#, and .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid.” The shell servers as an interface for quires, data updates, and allows administrative operations to be performed (geeksforgeek.org, 2020). Geeksforgeeks.org defines the engine as “an important part of MongoDB which is generally used to manage how data is stored in the memory and on the disk.” These three parts operate on two layers known as the application and data layers. The application layer is broken down into two sub- categories know as the frontend and backend (geeksforgeek.org, 2020). The frontend is where users interact with the database through either webpages or mobile applications (geeksforgeek.org, 2020). The backend is used to preform server-side logic and interacts with queries (geeksforgeek.org, 2020). The data layer consists of servers and the storage engine. Queries are sent to receive by the servers and sent to the storage engine. The storage engine is responsible for reading and writing data and also manages data (geeksforgeek.org, 2020).

Aws DynamoDB is an example of a key-value database and is designed to reduce the complexity between tables by combining objects into collections based on common themes. Influxdata.com describes key-value databases as having “sets of key-value pairs, where the key is the identifier, and the value is the data in question. A key-value pair is very similar to the various different implementations of hash tables in many programming languages, such as dictionaries with Python and objects in Javascript. The major difference with a key-value database is that your data is persisted and managed via the database you are using. Under the hood, key-value databases work by keeping an in-memory data structure that is mapped to the data stored on disk. RAM is much faster than accessing data from disk so most databases will have some sort of algorithm to keep frequently accessed data in RAM and only fallback to disk if the index isn’t already stored in memory.” Specifically, DynamoDB consists of a table, secondary index, primary key, sort key, item, attribute, streams, queries, scan, and filter (dashbird.io, 2020). DynamoDB stores items in a hash table; each item has a “key-value” that is used as an identifier.

Hbase is an example of a column-oriented database which means the data is stored in individual columns and indexed by a unique row key. According to aws.amazon.com the architecture structure of a column-oriented database “allows for rapid retrieval of individual rows and columns and efficient scans over individual columns within a table. Both data and requests are distributed across all servers in an HBase cluster, allowing you to query results on petabytes of data within milliseconds. HBase is most effectively used to store non-relational data, accessed via the HBase API.” According to towardsdatascience.com Hbase uses “zookeeper” to manage clusters. HBase’s underlying architecture consists of Master, multiple region servers, and HDFS (towardsdatascience.com, 2019).

The use of NoSQL has a variety of advantages, according to MogoDB.com, the most notable being the ability to handle enormous amounts of data without sacrificing speed. As previously mentioned, NoSQL databases are able to handle large quantities of data due to being flexible, dynamic, and scaled horizontally. These attribute along with the fact that SQL databases are upgraded using hardware gives NoSQL and advantage when handling large amounts of data. A second advantage of NoSQL is its ability to store data using a variety of structure formats from unstructured to semi-structured or even structured. This allows applications to present information in a variety of ways and allows the information to be presented in way that more accurately represents how the data would be used. A third advantage of using NoSQL is the ability to easily update schema and data. NoSQL allows for updates to be made quickly and significantly reduces the threat of downtime. The final advantage of NoSQL is that it is a developer friendly system. According to [www.adservio.fr](http://www.adservio.fr) “Many NoSQL database management systems require only a few lines of code, which is ideal for developers who want to get started quickly. For example, MongoDB is a NoSQL database management system that allows developers to store data in flexible structures and retrieve it with code written in the language of their choice.” Although there are several advantages to NoSQL there are some drawbacks as well.

A major disadvantage of using a NoSQL is the lack of standardization and the fact that NoSQLs have only been used since the 2000s. Variation in how these databases are implemented creates an uncertain future for their use. It also creates space for confusion; if there is no standard model then users may have difficulty navigating their systems. The lack of maturity also limits the amount of support that will be available on a large scale. A second disadvantage is the issue of backup, while specific NoSQL provide tools for backup, they lack maturity. It is difficult to say weather or not the tools are sufficient to completely back up the large amounts of data. The third and final disadvantage of NoSQL is that queries lack adjustability. Due to a low level of adjustability, you cannot use different types of standard queries.

A graph database is a platform for creating and manipulating graphs (Oracle, 2022). Graph databases are built to store and navigate information surrounding relationships (asw,amazon, 2022). According to techtarget.com “A graph database, also referred to as a semantic database, is a software application designed to store, query and modify network graphs. A network graph is a visual construct that consists of nodes and edges. Each node represents an entity (such as a person) and each edge represents a connection or relationship between two nodes. Graph databases have been around in some variation for a long time. For example, a family tree is a very simple graph database. The concept of using databases to map relationships digitally started seeing popular usage in business around 2015 when increased compute power, in-memory computing, and agreed-upon standards moved the concept from academics to real-world uses in business and enterprise computing.” There are two different types of graph databases “property” and “RDF or Resource Description Framework” (Oracle, 2022). Property graph focus on mapping relationships of data and allow query and data analytics based on the relationships created (Oracle, 2022). RDF graph databases focus on data integration (Oracle, 2022). Graph databases are traditionally classified as a type of NoSQL database (techtarget, 2022). Graph databases use case examples include social networking, recommendation engines, and fraud detection (aws.amazon, 2022).

Graph databases are used to navigate deep hierarchies, find hidden connections between distant items, and discover inter-relationships between items. These broad uses translate seamlessly into real would applications such as fraud detection, recommendation engines, and social media analysis. Graph databases have become a staple in the finance industry (Oracle, 2022). The first line of defense against fraud protection is the use of pattern recognition (Oracle, 2022). The finance industry can use graph database to track spending patterns of credit card users. These patterns can be classified in a number of ways and then connected through a graph database (Oracle, 2022). Classifications would be purchase location, frequency, types of stores, and other things that fit a user profile. The info would be gathered in a database and supplemented with machine learning in detecting against fraud (Oracle, 2022). Recommendation engines use graph databases to map data, for example a recommendation engine for purchases would store a user’s consumer interests, friends’ purchases, and purchase history (aws.amazon, 2022). The engine would then use this information to find other products in similar categories (aws.amazon, 2022). According to Oracle.com “social networks make the ideal use case as they involve a heavy volume of nodes (user accounts) and multi-dimensional connections (engagements in many different directions). A graph analysis for a social network can determine how active are users, (number of nodes) which users have the most influence, (density of connections) who has the most two-way engagement, (direction and density of connections). However, this information is useless if it has been unnaturally skewed by bots. Fortunately, graph analytics can provide an excellent means for identifying and filtering out bots.”

An example of a graph database is neo4j. According to bbvaapimarket.com “Neo4j is a property graph database, and it uses graphs to represent data and the relationships between them. A graph is defined as any graphical representation that consists of vertices (shown by circles) and edges (shown with intersection lines)”. A Neo4j property graph is a weighted graph, that uses labels where property can be assigned to both nodes and relationships. Graph databases update the node and the relationships based on searches and quires (bbvaapimarket.com, 2015).

A second example of a graph databases is Amazon Neptune. Neptune is a graph database service that is optimized to store billions of relationships based on highly connected datasets (docs.aws.amazon.com, 2022). Neptune supports many popular query languages and is used for recommendation engines, fraud protection and knowledge graphs (docs.aws.amazon.com, 2022). Neptune has three main components, the primary DB instance, the Neptune replica, and cluster volume. The primary DB instance is responsible for reading and writing data and preforms all data modifications. The replica connects to the same storage locations, but only supports read functionality. The cluster volume is compiled of copies of data across multiple zones.

Using a graph database in certain scenarios provides several advantages including group by aggregate queries, combine and hierarchize multiple dimensions, AI infrastructure, make powerful recursive path query Easily accessible. In addition to group by queries graph databases allow for group by aggregate due to their flexible grouping restrictions (tigergraph, 2020). This allows for a wider range of grouping which would aid systems in recognizing more distant patterns (tigergraph, 2020). Another feature that aids in pattern recognition is combine and hierarchize multiple dimensions. Graph databases are able to manage data in a manner that groups time series, demographic, geo-dimensions, etc. with a hierarchy of granularity on different dimensions (tigergraph, 2020). Graph databases allow situations where groups of people can be segmented on multiple different levels (tiger, 2020). Graph databases also pair well AI infrastructure. According to TigerGraph.com “Graph databases serve as great AI infrastructure due to well-structured relational information between entities, which allows one to further infer indirect facts and knowledge. Machine learning experts love them. They provide rich information and convenient data accessibility that other data models can hardly satisfy.” Finally graph databases make powerful recursive path query easily accessible (tigergraph, 2020). Graph databases queries translate easily to real life situations by finding direct and indirect connections in a graph. According to TigerGraph.com “we can extend the single-pair-vertex reachability queries with multiple reachability queries sharing some common vertices. A set of reachability path queries can be bundled together to constrain each other to form an interesting subgraph pattern.” This allows applications to create a subgraph based on patterns formed from the main graph (tigergraph.com).

The major disadvantages of graph databases are the level of difficulty to scale as designed as one-tier architecture and the absence of a uniform query language (ionos.com, 19). Graph databases were originally designed to work on a one tier architecture structure. Meaning they were designed to work on a single server. According to itworldcanada.com vendors have started to offer what is known as “sharding.” Sharding is the ability of a graph databases to be distributed across multiple servers (itworld.canada, 2020). Sharding could be a viable solution for the scaling issue surrounding graph databases, but not all vendors offer this functionality currently. The second main issue is the lack of a standard language (itworldcanada.com, 2020). All vendors use a unique language or syntax regarding updates and queries (itworldcanada.com, 2020). The lack of a standard language presents several issues such as causing difficulty when trying to migrate from one product to another. This also means companies will need to either higher staff who is already familiar with the product or the company will be required to spend time and money training staff (itworldcanada.com, 2020).

Bibliography

5, M., Shiff, L. &amp; Rowe, W., 2018. SQL VS nosql databases: What's the difference? BMC Blogs. Available at: https://www.bmc.com/blogs/sql-vs-nosql/#:~:text=In%20contrast%2C%20NoSQL%20databases%20are,or%20constantly%20evolving%20data%20sets. [Accessed September 6, 2022].

Anon, Column-oriented databases, explained. KDnuggets. Available at: https://www.kdnuggets.com/2021/02/understanding-nosql-database-types-column-oriented-databases.html [Accessed September 6, 2022].

Anon, Document database - nosql. MongoDB. Available at: https://www.mongodb.com/document-databases#:~:text=A%20document%20database%20(also%20known,for%20developers%20to%20work%20with. [Accessed September 6, 2022].

Anon, 2022. Graph database: How it works, its uses &amp; benefits. InfluxData. Available at: https://www.influxdata.com/graph-database/ [Accessed September 6, 2022].

Anon, 2020. How mongodb works ? GeeksforGeeks. Available at: https://www.geeksforgeeks.org/how-mongodb-works/ [Accessed September 6, 2022].

Anon, NoSQL vs SQL databases. MongoDB. Available at: https://www.mongodb.com/nosql-explained/nosql-vs-sql [Accessed September 6, 2022].

Anon, What are the pros and cons of nosql. Adservio. Available at: https://www.adservio.fr/post/what-are-the-pros-and-cons-of-nosql [Accessed September 6, 2022].

Anon, What is a key-value database? MongoDB. Available at: https://www.mongodb.com/databases/key-value-database [Accessed September 6, 2022].

Contributor, T.T., 2020. What is graph database? - definition from whatis.com. WhatIs.com. Available at: https://www.techtarget.com/whatis/definition/graph-database [Accessed September 6, 2022].

Clay, E., 2021. Horizontal vs vertical scaling: What you need to know. Touchstone Security. Available at: https://touchstonesecurity.com/horizontal-vs-vertical-scaling-what-you-need-to-know/#:~:text=The%20main%20difference%20between%20scaling,or%20RAM%20to%20existing%20machines. [Accessed September 6, 2022].

Wu, M., 2022. Graph database advantages: Using graph query languages. TigerGraph. Available at: https://www.tigergraph.com/blog/what-are-the-major-advantages-of-using-a-graph-database/ [Accessed September 6, 2022].

“Neo4j: What a Graph Database Is and What It Is Used For.” BBVA API\_Market, https://www.bbvaapimarket.com/en/api-world/neo4j-what-graph-database-and-what-it-used/#:~:text=Neo4j%20uses%20property%20graphs%20to,%2Drelational%20(NoSQL)%20databases.

Vogt, G. (2000). Neptune. Amazon. Retrieved September 6, 2022, from https://docs.aws.amazon.com/neptune/latest/userguide/intro.html