Non-Relational Data Storage and Retrieval Systems

Daniel Cannon

CS-300-ON: Database Management Systems

Professor Osvaldo Garcia Almaguer

October 29, 2023

Non-Relational Data Storage and Retrieval Systems

SQL is the main tool for querying questions from a database, but other structures exist. Called a fourth-generation database management system, NoSQL, which can stand for Not Only SQL, was designed to get around the limitations of the relational model and has been recently increasing in popularity. It can also stand for "non-relational SQL," according to some (Smallcombe, 2023). NoSQL databases are a new database system that has emerged in response to the challenges and limitations of traditional relational databases (MongoDB, 2023). Relational databases have been the primary database model for many years but aren't appropriate for every problem (RavenDB, 2023).

Large companies rely on SQL databases to store data and run queries quickly. Relational database examples are MySQL and SQL Server, which store data efficiently and consistently. However, maintaining the relationships is intensive and consumes memory and power. SQL queries can be slow and inefficient when dealing with large amounts of data or complex joins across multiple tables. SQL also does not support some common queries in modern applications.

As data amount increases, relational databases have performance and availability issues. Relational databases typically scale vertically by adding more resources to a single server, which can be expensive and limited. On the other hand, NoSQL databases scale horizontally by distributing data (DPAD, 2023). This can be cheaper and more scalable. NoSQL databases use sharding, replication, and partitioning techniques for horizontal scaling, improving fault tolerance, and load balancing (Siddique, 2023). Relational databases are good at scaling vertically, while NoSQL databases quickly scale vertically and horizontally (nOps, 2023). Vertical scaling means that you are adding more power and storage to a machine.

In contrast, horizontal scaling involves adding additional nodes. In order to handle large amounts of traffic, horizontal scaling is used to manage the tasks and applications, which NoSQL databases are proficient in compared to traditional databases. Vertical scaling obviously has its limitations and can only be executed to a certain extent since a single machine cannot infinitely increase in power. In NoSQL, every item stands on its own and only has two fields, which are known as keys and values. If the volume of queries is too high, the workload can be split across more servers, which is horizontal scaling.

NoSQL databases were designed to solve several issues. Many applications these days deal with data that doesn't fit well into relational structures (i.e. text, images, videos). NoSQL databases don't require data to be changed from that original format. They don't impose any constraints, which makes it easier for users to manipulate the data since they can use it in its appropriate form. NoSQL does have some issues of its own, such as being limited in data retrieval options. Some examples of NoSQL database products are Apache Cassandra, MongoDB, and Couchbase, which will be discussed further.

NoSQL Database Products

Apache Cassandra is a DMS that companies use for various applications. It has a "masterless design," which means that there's no bottleneck in the system (Leontiev, 2019). Every node in the cluster is identical and can perform any operation. Data is replicated across multiple nodes and the replication factor can be configured for each keyspace, which is a group of tables. Cassandra uses "hashing" to distribute data among nodes and uses a "gossip protocol" to maintain cluster membership and state information (Singhal, 2019). Cassandra also supports tunable consistency, which allows the client to be specific about the level of consistency needed for each read or write operation.

A table in Cassandra consists of rows and columns where a primary key identifies each row (Tseng et al., 2020) – this is how NoSQL databases function. Each column has a name, a value, and a timestamp associated with it. A table can have partitions, which are subsets of rows sharing the same partition key (the first part of the primary key), and each partition can have one or more clusters (the remaining portion of the primary key). Apache Cassandra supports CQL (Cassandra Query Language), which bears similarities to SQL but has limitations. With CQL, clients can. Modify query tables as well as perform data insertion, update, deletion, and selection operations. However, CQL does not support joins, subqueries, aggregations, or transactions. Additionally, it requires that the key or secondary indexes restrict queries and that the clustering key determines the order of results.

MongoDB is an open-source NoSQL database system that's freely available (MongoDB, 2023). MongoDB uses documents as its data structure, similar to JavaScript Object Notation (JSON) objects. Documents are dynamic, so they can have any number and type of fields and change over time. They are organized into

collections that are similar to tables in traditional databases (Kobielus, 2018). MongoDB employs sharding and replication techniques to distribute and replicate data across nodes. Sharding allows for workload distribution and improved query performance, while replication ensures data consistency and durability in case of node failures. Another advantage of MongoDB is its compatibility with CQL, making it suitable for applications requiring fast and flexible data access. Additionally, MongoDB works with programming languages like Python, Java, Ruby, and more.

Couchbase is a cloud-based database platform. It was created in 2011 through the merger of two open-source projects, Membase and CouchOne (Aslett, 2021). Membase was a distributed key-value store, whereas CouchOne focused on document-oriented databases. Its distributed architecture enables the handling of large-scale data across nodes or servers.

To divide the data based on a hash function, Couchbase utilizes a technique called sharding. It also uses replication to copy data across nodes for fault tolerance and load-balancing purposes (Objelean, 2023). The database offers flexibility when it comes to storing and querying data by supporting JSON documents as the underlying data structure.

Couchbase provides a query language called N1QL, enabling users to perform CRUD operations (create, read, update, delete) on documents as well as execute complex queries. Additionally, Couchbase offers full-text search and analytics. Moreover, Couchbase includes eventing capabilities that allow users to do custom logic when there are changes in the data by utilizing JavaScript functions.

Graph Databases

There are several NoSQL databases - the four most used are key-value, document-based, graph-based, and wide-column-based databases (Dancuk, 2023). Graph databases are a NoSQL database type that uses structures in the form of graphs to store and query data (Expeed Software, 2023). Graph structures are made of nodes representing objects and edges representing relationships between the nodes. These nodes and edges can also have properties or attributes that provide information. Unlike types of NoSQL databases, which don't focus as much on data relationships, graph databases excel in handling complex and dynamic data relationships. They

find applications in fields such as social networks, recommendation systems, fraud detection, and route optimization. One of the advantages of using graph databases is their ability to model data naturally and intuitively while allowing for fast and flexible data queries. They distribute data and workload across nodes to ensure availability and fault tolerance (GeeksforGeeks, 2023).

Additionally, they offer query languages optimized explicitly for their data models and use cases. Some examples of graph query languages are CQL and N1QL. There is also Cypher, which is a language used for querying graphs, and Gremlin, an imperative language used for graph traversal (NebulaGraph Database, 2022).

When it comes to handling relationships between data, graph databases offer performance and flexibility compared to relational databases (Amazon Web Services, 2023). In graph database relationships, they're stored as parts of the database itself, eliminating the need for joins when querying them. What makes graph databases appealing is their ability to add or modify data without impacting the structure. While not a recent development in technology, graph databases have gained popularity recently due to their adoption by major tech companies. Prime examples of graph databases are Neo4j and Amazon Neptune.

Neo4j uses sharding to partition data based on replication (which is a hash function method) to copy data across nodes (Neo4j, 2023). Neo4j also supports automatic failover, automatic rebalancing, and cross-data center replication for availability and disaster recovery. Another advantage of Neo4j is that it supports Cypher (LangChain, 2023). Cypher also provides support for data consistency and integrity. Neo4j is well suited for many applications involving complex and dynamic relationships between data, like social networks, recommendation systems, fraud detection, and route optimization.

Amazon Neptune is a cloud-based graph database service offered by Amazon Web Services (AWS). It supports two graph models: property graphs and RDF graphs. Property graphs utilize query languages, like Gremlin and openCypher, while RDF graphs use SPARQL (TerminusDB, 2022). Property graphs are more straightforward compared to RDF as they employ identifiers known as simple strings. Neptune is a service that takes care of all the tasks related to managing databases, including hardware setup, software updates, configuration, data backup, and recovery (TaggedWeb, 2023). Users can focus on their data and queries without worrying about the technicalities.

Neptune stores data in a cluster volume to ensure its durability and availability. In case of any node failures, it automatically handles failover and recovery. One of the advantages of Neptune is its support for serverless computing. Users are not required to provision or manage servers or instances; they only pay for the resources they actually use. Additionally, Neptune offers a "global database" feature that allows for fast local reads and writes across different regions.

When compared to databases or other NoSQL databases, graph databases have several benefits. They excel at handling dynamic relationships between data naturally and intuitively. Graph databases allow users to model their data as graphs that closely resemble how humans think and reason about information. Moreover, these databases enable users to add or modify data without impacting the schema or structure of the database. They also provide flexible querying capabilities on the stored data since relationships are treated as essential elements within the database. They don't need to use joins to query them. Graph databases also have query languages that are optimized for their data models and specific use cases, allowing for expressive queries. To ensure availability and fault tolerance, graph databases employ techniques like sharding, replication, and partitioning to distribute data and workload across nodes.

However, graph databases do have some drawbacks in exchange for performance gains. They adhere to the BASE model, which means they may not always guarantee that all nodes have an identical view of the data at any given time. BASE stands for basically available, soft state, and eventual consistency. This can potentially result in data inconsistency or loss in scenarios (Balusamy et al., 2021). Additionally, graph databases lack standardization or interoperability and don't possess a universal protocol for their data models or query languages. This can lead to compatibility issues when using graph database systems or integrating with others. Moreover, since graph databases have concepts and terminologies compared to relational databases, users might need to acquire new skills. Furthermore, optimizing performance in graph databases requires different techniques than those employed by relational databases, so users may need to adjust their strategies (Synametrics Technologies, 2022).

In summary, NoSQL databases are non-relational data storage systems that provide flexibility, scalability, and high availability. Although traditional databases are the foundation of many applications, NoSQL databases

address various problems with relational databases. They enable users to much more easily deal with large amounts of data. Although not appropriate for every issue, they are another effective tool to be used.

References

Amazon Web Services. (2023). *What is a graph database?* AWS. Retrieved October 25, 2023, from https://aws.amazon.com/nosql/graph/

Aslett, M. (2021, December 30). Couchbase modernizing relational database workloads. *Ventana Research*. Retrieved October 26, 2023, from https://mattaslett.ventanaresearch.com/couchbase-modernizing-relational-database-workloads

Balusamy, B., Abirami, N., & Kadry, S. (2021). NoSQL Database. *Big Data: Concepts, Technology, and Architecture*. https://doi.org/10.1002/9781119701859.ch3

Dancuk, M. (2023, June 21). *What is a graph database?* Knowledge Base by phoenixNAP. Retrieved October 26, 2023, from https://phoenixnap.com/kb/graph-database

DPAD. (2023, March 19). Decentralization: characteristics, structural examples and use cases. *Medium*. Retrieved October 28, 2023, from https://dpadofficial.medium.com/decentralization-characteristics-structural-examples-and-use-cases-9412b52d2933

GeeksforGeeks. (2023, August 5). *What is a graph database*. Retrieved October 28, 2023, from https://www.geeksforgeeks.org/what-is-graph-database/

Kobielus, J. (2018, July 3). *MongoDB drives NoSQL more deeply into enterprise opportunities*. Wikibon Research. Retrieved October 27, 2023, from https://wikibon.com/mongodb-drives-nosql-deeply-enterprise-opportunities/

LangChain. (2023). *Neo4j*. Retrieved October 29, 2023, from https://python.langchain.com/docs/integrations/providers/neo4j

Leontiev, S. (2019, December 6). 3 reasons you need a masterless architecture. *DataStax*. Retrieved October 26, 2023, from https://www.datastax.com/blog/3-reasons-you-need-masterless-architecture

MongoDB. (2023). *MongoDB: the developer data platform*. Retrieved October 27, 2023, from https://www.mongodb.com/

Neo4j. (2023, June 8). *Neo4j graph database & analytics*. Graph Database & Analytics. Retrieved October 26, 2023, from https://neo4j.com/

nOps. (2023, September 20). *Horizontal vs vertical scaling: an in-depth guide*. Retrieved October 29, 2023, from https://www.nops.io/blog/horizontal-vs-vertical-scaling/#:~:text=The%20primary%20difference%20between%20horizontal,.)%20to%20an%20existing%20machine

Objelean, A. (2023). *Introduction to Couchbase - NoSQL document database*. Today Software Magazine. Retrieved October 23, 2023, from https://www.todaysoftmag.com/article/1506/introduction-to-couchbase-nosql-document-database

RavenDB. (2023, August 2). *Relational database integration*. RavenDB NoSQL Database. Retrieved October 25, 2023, from https://ravendb.net/why-ravendb/integration-with-relational-databases#:~:text=RavenDB%20gives%20you%20the%20ability,Microservices.

Siddique, R. (2023, March 28). Architecting a scalable and fault-tolerant system for a million users. *Medium*. Retrieved October 24, 2023, from https://levelup.gitconnected.com/building-a-fault-tolerant-system-architecting-for-scalability-d56c5426d5db

Singhal, S. (2021, December 11). Implementing Cassandra's gossip protocol. *Medium*. https://medium.com/@swarnimsinghal/implementing-cassandras-gossip-protocol-part-1-b9fd161e5f49

Smallcombe, M. (2018, November 29). *SQL vs NoSQL: 5 critical differences*. Integrate.io. https://www.integrate.io/blog/the-sql-vs-nosql-difference/

Synametrics Technologies. (2023). *Top 10 performance tuning tips for relational databases*. Retrieved October 26, 2023, from https://www.synametrics.com/SynametricsWebApp/WPTop10Tips.jsp

TaggedWeb. (n.d.). *Amazon Neptune*. Retrieved October 25, 2023, from https://www.taggedweb.com/software/amazon-neptune

TerminusDB. (2023, July 24). *Graph database fundamentals*. Retrieved October 23, 2023, from https://terminusdb.com/blog/graph-database-fundamentals/

Tseng, L., Pan, H., & Wu, Y. (2020). Tutorial: deep dive into Apache Cassandra: theory, design, and application. *2020 IEEE International Conference on Pervasive Computing and Communications Workshops*. https://doi.org/10.1109/percomworkshops48775.2020.9156261