Neo4j is a NoSQL database. It is highly scalable and schema-free. It is the world's leading open-source graph database[1]. Its architecture is designed for traversal of nodes and relationships between the nodes. It provides optimal management and storage. It aimed at storing and querying graphs rather than tables. It is scheme free. The graph contains nodes. Nodes are connected to each other if there exists a relationship. Nodes and relationship store data in the form of key-value pairs. It is not required to declare a primary key or foreign key to declare the relationship. It can be deployed as a standalone application. It is available in different editions. Enterprise edition supports more features than community edition.

There are many advantages of using Neo4j alongside RDBMS[2]. We use oracle as it is a trusted and well-established technology. We may have created many applications using an oracle. We can use this without interruption and gain the strategic capabilities of the Graph database by adding Neo4j to your arsenal. By using them both we manage both structured and unstructured, highly connected dynamic data. Therefore we can improve our application performance. We use JOINS to combine or perform analysis between to tables in RDBMS. These operations are memory-intensive which also affects the performance of our application. If the graph database is used we can analyze complex data fast. The depth of connections will affect the query significantly. If we use Ne04j to query data, it queries depth of millions or tens of millions of connections per second per computer core.

Features:

It supports ACID(atomicity, consistency, isolation, durability) transactions.

It provides a query language called Cypher graph query language for creating and retrieving the relationship between data eliminating complex queries like joins[3].

It provides language drivers(API) for programming languages like C#, Java, JavaScript, and python.

It provides graph algorithm libraries to support AI initiatives.

It provides a built-in browser and terminal which can be used to create and retrieve graph data.

It provides offline backups and bulk import tool for importing bulk data

It supports exporting data to JSON and XLS format.

It provides high-performance native API and cypher API to develop java applications

It provides high-performance caching supporting scalability.

Significance:

It has a well-developed UI with the natural association and built-in learning like training materials, expert-authored books which helps us easy to learn[4].

Cypher, the worlds most powerful graph query language, libraries for various programming languages, java native API makes it easy to use

Key investors like Forrester, Gartner, and others recognize the neo4j graph dataset to have enough production applications to warrant considerations in the report.

We need to worry about the volume of the data while importing because it has an amazing stacking rate of colossal information sizes, with exceptionally low memory impressions.

We can make changes as the business requirement changes because of its whiteboard-friendly data modeling which simplifies the development cycle

We can pick the license and group that you need and include bunching and data replication capacities that bode well for your deployment and your association. This reduces expenses.

Protects data integrity performing fast read and write operations by combing native graph storage, optimized scalable architecture, and acid compliance.

It supports both structured and unstructured data which helped NASA[5] users advance the mars program by 2 years because Nasa focuses more on unstructured data for the mars program and it is also incredibly fast at reading

It provides the locking support which helps in the privacy of our data and it can also be embedded into our application so that we can directly access that database without the help of cypher query language

Limitations:

No sharding support: Sharding is one of the key features every NoSQL databases should process because sharding helps us break large dataset and distribute into a number of typical replicated databases called shards[6].

There is a limitation on the number of nodes, relationships, and properties on the Community version of neo4j. Being a free version it does not support all the features supported by enterprise edition.

A master-slave architecture is used in Ne04j where writes are always done on the master even if we write on the slave. Updates we write on the master will be written on slaves but not immediately visible on all the other slaves. One best thing is even if the master dies another slave is selected as a master. This architecture is somewhat complex [7].

The queries that perform some kind of grouping and aggregation are summarization queries. But only k-neighborhood and summarization are not supported by Neo4j[8].

The system will slow down when your data-sets in Neo4j becomes larger than the available RAM of your system because all the data is stored in the system. You should have data in the primary memory if you want to retrieve data faster or avoid expensive data seek.

There is an alternative approach to present this dataset using another database called ArongaDB.

ArongaDB:

It is a multi-model database system[6]. In this database, data is stored in key or value pairs, documents or graphs. This can be accessed by AQL query language. It helps in keeping the product updated which required by any developing product. A huge amount of storage memory which is the problem for neo4j has been simplified by ArangoDB. It also simplifies fault tolerance and increases performance and flexibility. Master-slave Architecture which is a problem in Neo4j is replaced by Master/master CP(Consistent and Partition) architecture. CP indicates the presence of network partition. It is a distributed multimodel database where cluster continues to serve even the machine fails. Sharding which is a limitation in Neo4j is also provided in ArangoDB. It provides HTTP API to manage our database. It also provides a cloud that is already ready on the Azure Cloud.

References:

[1]"Learn Neo4j Tutorial - javatpoint", *www.javatpoint.com*, 2020. [Online]. Available: https://www.javatpoint.com/neo4j-tutorial. [Accessed: 25- Feb- 2020].

[2]Gabe Stanek and Stefan Kolmar, "3 Advantages of Using Neo4j Alongside Oracle RDBMS", *Neo4j Graph Database Platform*, 2016. [Online]. Available: https://neo4j.com/blog/3-advantages-neo4j-alongside-oracle-rdbms/. [Accessed: 26- Feb- 2020].

[3]"Chapter 1. Introduction - The Neo4j Operations Manual v4.0", *Neo4j.com*, 2020. [Online]. Available: https://neo4j.com/docs/operations-manual/4.0/introduction/. [Accessed: 25- Feb- 2020].

[4]"Top 10 Reasons for Choosing Neo4j for Your Graph Database", *Neo4j Graph Database Platform*, 2020. [Online]. Available: https://neo4j.com/top-ten-reasons/. [Accessed: 25- Feb- 2020].

[5]"NASA - Neo4j Graph Database Platform", *Neo4j Graph Database Platform*, 2020. [Online]. Available: https://neo4j.com/users/nasa/. [Accessed: 25- Feb- 2020].

[6]D Fernandes and J Bernardino, "Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB", *Scitepress.org*, 2018. [Online]. Available: https://www.scitepress.org/papers/2018/69102/69102.pdf. [Accessed: 26- Feb- 2020].

[7]Rolando and M. Hunger, "Is Neo4j 2 replication architecture a master-slave or multi-master?", *Database Administrators Stack Exchange*, 2015. [Online]. Available: https://dba.stackexchange.com/questions/95841/is-neo4j-2-replication-architecture-a-master-slave-or-multi-master. [Accessed: 26- Feb- 2020].

[8] F. Holzschuher and R. Peinl, "Performance of graph query languages", *Proceedings of the Joint EDBT/ICDT 2013 Workshops on - EDBT '13*, 2013. Available: 10.1145/2457317.2457351 [Accessed 26 February 2020].