Neo4j Technical Report

Akash Bharti
B00837970

Introduction

Neo4j[1] is an open-source project maintained by a private company Neo4j Inc. It is the SQL of the graph databases. It provides a graph database management system, a language to query the database called "Cypher" and also a visual interface known as the neo4j browser. It stores data as nodes and relationships both holding properties in the form of key-value form. Nodes are often used to represent entities. It also supports labels that can be used to group nodes, and each node can be assigned multiple labels. Each node and relationship can also have properties. This structure enables developers to model any scenario that is defined by relationships.

Neo4j also has a great and powerful visualization tool known as Neo4j Desktop which is a convenient way for developers to work with the local Neo4j database. Applications today face the challenge of handling large amounts of interconnected data. Neo4j allows users to build applications capable of handling this type of data. With neo4j, we can store the relationship between data and access those relationships really easily. The language Cypher is very similar to SQL and allows both simple and complex queries, to get what we want.

Significance

A relational database, although provides the ACID properties but has some limitations as well. They are not optimized to handle large amounts of data. They also cannot describe relationships other than the standard one-to-one, one-to-many, and many-to-many. Horizontal scaling is also very inefficient for relational databases. In order to overcome these limitations, a number of non-relational databases have been developed. Neo4j is one of them. Neo4j provides the following advantages [2][3]:

1. Performance
   Neo4j overcomes the performance limitation of relational databases. In Neo4j, performance will always remain high even if the data grows significantly unlike relational databases which suffer as the number and depth of relationships increase.

2. Flexibility
   The Structure and scheme of Neo4j database can be easily adjusted according to the situation. It is very easy to upgrade the data structure without affecting the existing functionality.
   Use Cases:
   1. Fraud Detection and Analytics
      It is already being used in detecting fraud in sectors like banking, insurance and e-commerce.

   2. Recommendation Engines
      It can easily handle recommendations, outperforming other relational and NoSQL databases.

3. Social Networks
   Neo4j is perfect for social networks. It can speed up the development of social network applications.

4. Identity and Access Management
   Neo4j makes it easy to manage constantly changing roles, groups, and identities for business.

5. Privacy and Risk Compliance
   It allows you to track where our private information is stored and who accesses it.

6. Knowledge Graph
   It can enhance our application's search capabilities to provide relevant results. It can improve keyword searches and deliver additional results related to the keywords.

3. Responsiveness
   One of the strengths of neo4j is managing data. If we want to push the boundaries of its capability, we would need a total of 34 billion nodes, 34 billion relationships between these nodes, 68 billion of properties and 32000 types of relationships.

Limitations

Neo4j is certainly one of the best graph databases that one can use for free, but it is still very raw. If the task is not very trivial, then we will get some overhead for making it work with Neo4j. Neo4j supports master-slave replication to use slaves for reads and writes, but the whole database has to be copied to each node. In addition to this, Neo4j has only hash indexes. It lacks range indexing which makes sorting essential but it can be a very expensive operation. Neo4j also does not scale horizontally as well as some NoSQL databases available.

Alternative approach

Amazon Neptune [5] is a cloud-based high-performance graph database that is available on AWS. It supports popular graph query languages such as Gremlin and SPARQL to query the connected data. Amazon Neptune provides great features such as provisioning, patching, backup, recovery, failure detection and repair. Neptune is closed source, unlike Neo4j which is open source [6]. Neo4j has a strong and customizable graph visualization tool known as Neo4j Desktop. Neptune, on the other hand, lacks proper graph data visualization. The main difference between Neo4j and Neptune is in security. While Amazon Neptune isolates the Graph Data in Virtual Private Clouds, Neo4j has user role management which is only available with Neo4j Enterprise Edition. Permissions, in Amazon Neptune, are managed by standard AWS IAM roles.

References

[1] “Neo4j Documentation,” *Neo4j Graph Database Platform*, 27-Apr-2018. [Online]. Available: https://neo4j.com/docs/. [Accessed: 27-Feb-2020].

[2] “Neo4j: What a graph database is and what it is used for,” *BBVAOpen4U*, 2015. [Online]. Available: https://bbvaopen4u.com/en/actualidad/neo4j-what-graph-database-and-what-it-used. [Accessed: 25-Feb-2020].

[3] Gleb B and Vlad L, “Capabilities of the Neo4j Graph Database with Real-life Examples,” *Rubygarage.org*, 05-Jul-2018. [Online]. Available: https://rubygarage.org/blog/neo4j-database-guide-with-use-cases. [Accessed: 25-Feb-2020].

[4] “How To Build A Recommendation Engine In Two Minutes Flat | Neo4j,” *Neo4j Graph Database Platform*, 26-Feb-2020. [Online]. Available: https://neo4j.com/developer/guide-build-a-recommendation-engine/. [Accessed: 25-Feb-2020].

[5] “Amazon Neptune - Fast, Reliable Graph Database built for the cloud,” *Amazon Web Services, Inc.*, 2017. [Online]. Available: https://aws.amazon.com/neptune/. [Accessed: 27-Feb-2020].

[6] solo, “Graph Data and GraphQL API Development—Leap Graph,” *Graph Data and GraphQL API Development—Leap Graph*, 2011. [Online]. Available: https://leapgraph.com/aws-neptune-vs-neo4j. [Accessed: 25-Feb-2020].