The key feature analysis of Graph Databases

Md Haris Uddin Sharif[1], Krishna Chaitanya Potlapally[2],
Nandun Chiranjeeva Edirisinghe[3]

[1,2,3]Department of Information Technology, University of the Cumberlands

ABSTRACT

The paper discusses different graph database platforms. Several competing platforms were identified and analyzed. They identified platforms are neo4j, Dgraph, and NoSQL. Key features were identified within the paper, and several business uses for them were analyzed. This analysis also attempted to identify the skills needed by database administrators and data administrators to use these databases at an organizational level. In addition to this research, we tried to find out those company or vendor who already has their product, enterprise system with that graph database, enterprise system. And a short overview of basic functionality, technical highlights. Moreover, we are tired of finding out the competitors of those graph databases.

Keywords: NoSQL, Neo4j, PostgreSQL, Dgraph DBS, Dgraph, Graph db uses, Graph db administrator.

1. INTRODUCTION

Graph databases are designed specifically for storing and navigating relationships. Relationships are first-class residents in graph databases, and they consider most of the database's value. Nodes are utilized to save data entities, while edges keep relationships between things in graph databases. Advantage contains a start node, an end node, a type, and a direction, and it may be used to define parent-child connections, actions, and ownership, among other things. The amount and types of relationships that a node can have are limitless.

A graph can be navigated along with specified edge types or over the whole chart in a graph database. Because the associations between nodes are not computed at query time but stored in the database, traversing the joins or relationships in graph databases is fast. When you need to construct linkages between data and query these associations fast, graph databases are helpful for use cases like social networking, recommendation engines, and fraud detection.

A social network graph is depicted in the graph below. You may figure out who a person's "friends of friends" are by looking at the people (nodes) and their relationships (edges)—for example, Howard's pals.

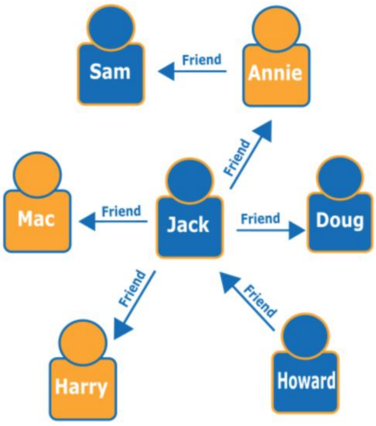


Fig: Social-Network-Graph [10].

The graph database Neo4j is an open-source, NoSQL, local design data set that gives an ACID-agreeable value-based backend for your applications. Starting improvement started in 2003. However, it has been openly accessible since 2007. The app's source code, written in Java with Scale, is accessible free of charge on GitHub or an easy to work and download.

Comcast is a vast media communications organization that gives satellite TV, ISP, items for home security, cell phones, and communication. The Xfinity Home is another administration period that allows clients to control or screen surveillance cameras, smoke alarms, indoor regulators, lighting, and movement sensors. Through an in-home touch screen console or distantly through a Web-based entryway or Internet-associated device, for example, a cell phone or tablet. Comcast has set out on making and finishing its xFi intelligent home prototype, which – given exploration – they knew needed to incorporate associated devices, associated individuals, rich interfaces, and computerization to be helpful. A few items that discarded this model include apparatuses like Kidwatch, which advises you when your kid shows up home from school, and Porchcam, which shows the individual who just rang your doorbell on your TV screen. Find how Xfinity Comcast depends on Neo4j for client personalization in its xFi innovation [1].

There are numerous benefits of utilizing Neo4j; for example, The Oracle Investment Keeps Paying Off, and the graph database Neo4j is getting mainstream step by step. There are situations where using an RDBMS versus a diagram information base bodes well, and the other way around. The organized information, while unstructured, significantly associated, or highly unique information, chains of importance and systems can be moved out of Oracle and Neo4j.

Using Neo4j and Oracle also empowers associations to improve application execution by offloading questions that influence associated information. Neo4j has the most significant and energetic local area of diagram information base fans adding to the Neo4j environment. Exceptionally Performant Read and Write Scalability, Without Compromise.

Using Neo4j, we can query any data with a depth of millions or tens of millions of combinations by a second per computer core, equivalent to a relational database world of millions of JOIN executions per square second core difficult. There's a big difference in speed, and this variation increases the tighter our database relationships are and the more data we will have.

Applying Neo4j, the quickest path query on data with tens of billions of nodes and connections can take one or two milliseconds to run on. A similar SQL query could run multiple thousands of times slower if an application were singly using an RDBMS such as Oracle [2].

COMPETITORS OF NEO4J

Neo4j's principal competitors include Tiger Graph, Stardog, Memgraph Riversand, and Dgraph Labs. First, let's discuss their statics based on online records.

Stardog is a tech company with a graph database that gives an enterprise information combination stage based on intelligent diagram innovation. The total funding raised $ 23.3m. Alexa Website Rank is 342628. The level of tweets with commitment (most recent 30 days) is 66.67% [3].

Riversand creates and conveys multi-space Master Data Management and Product Information Management organizations. The total funding raised $ 45m. Alexa Website Rank 190124. The level of tweets with commitment (most recent 30 days) is 90.91% [3].

Tiger Graph is a versatile diagram data set for the endeavor. The total funding raised $ 171.7m. Alexa Website Rank 295666. The level of tweets with commitment (most recent 30 days) is 90% [3].

Memgraph is an innovation organization promoting a value-based and logical diagram data set. The total funding raised $ 2.1m. Alexa Website Rank 184754. The level of tweets with commitment (most recent 30 days) is 100% [3].

Dgraph Labs is an organization that encourages an open-source and low-dormancy data set. The total funding raised $ 14.5m. Alexa Website Rank 205100. The level of tweets with commitment (most recent 30 days) is 90.7% [3].

The NoSQL graph database is a popular technology for managing, designing, and manipulating enormous data sets of structured, unstructured, or semi-structured data. It supports many companies accessing, analyzing, and integrating data from multiple sources, supporting them besides their big data and social media analytics.

Marriott International is a large hotel business company that deploys NoSQL for its reservation system that accuses $38 billion yearly. Marriott International is converting its legacy systems with NoSQL [4].

It was the best time to replace the legacy/old infrastructure to better race in the digital economy [4].

But that simple choice requests several complex issues, including open source versus restrictive systems and NoSQL versus open-source. Marriott International Inc was determined to modernize its infrastructure to an open-source platform that allows Marriott to expand new applications quicker [4].

Moving the original reservation system, which Marriott affectionately calls Marsha, has not without risk. Marsha usually processes 38 billion dollars in reservations yearly, running on an individual mainframe that has taken 160 GB of memory [16].

Marriott prefers to go open source and NoSQL; there are other circumstances in the determination to move Marsha off the mainframe:

Going from mainframe to distributed system helps the sourcing of possible technical skills.

Taking benefit of open source leadership decreases Marriott's application development expenses. For example, it was very costly to make modifications or enhancements to the mainframe software.

They reduced infrastructure expenses - it's also very costly to manage Marsha on an old/legacy system [4].

Cloud moves - moving from the mainframe indicated a more reliable solution to Marriott's cloud-based forward procedure, where web clients will be served from restricted data centers.
NoSQL is more desirable for object-oriented development - it was nice to have a data store anywhere they could go from Java object-oriented right into the database and back, without a significant translation [17].

Another issue is scalability with commodity hardware in the enterprise [4].

Here is another critical issue for downtime. Excellent availability with no downtime for version upgrades and support can be performed online. The rest is undoubtedly a big deal, particularly with multiple relational databases. When any team/company says this can take a 12 hours outage for the upgrade. However, using NoSQL solutions, we can do on-the-fly upgrades [4].

That company also considers the performance, speed, and fast response times - when customers are on their website, they want a quick answer with a smart device or laptop answer. NoSQL indeed gives that answer quickly to the customer [4].

COMPETITORS TO THE PLATFORM

MongoDB is a popular database that provides variations of NoSQL (scalability, flexibility, performance) and forms based on relational databases. Redis Enterprise is a distributed cloud service for hosting and managing the Redis dataset with predictable performance. This company is also a big competitor of NoSQL. IBM Db2 is the database that allows enterprise-wide extracts managing high-volume workloads. It is optimized to produce industry-leading offerings while lowering prices. Ninox is an easy no-code policy for everyone. They create business applications that suit any workflow. Additionally, Ninox encourages partner collaboration functionalities and combines the example of the most prevalent services: Google and others. This company is also the biggest competitor of NoSQL [15].

Redis is an open-source, ahead key-value store. It is often connected to a data structure server that keys can contain hashes, strings, lists, and sets. The Amazon Relational Database Service is a web service and becoming popular that performs it easy to set up, manipulate, and compare a relational DB with the cloud: PostgreSQL, My, Amazon Aurora, SQL, MariaDB, Oracle, and Microsoft SQL Server. The Amazon Relational Database Service is also another competitor of NoSQL. NoSQL is better than others because NoSQL is an excellent choice for these companies encountering rapid increases with no clear schema descriptions. NoSQL offers more flexibility than a relational database. It is a reliable option for businesses that must investigate large quantities of data or whose data structures they operate are variable. Dgraph is another open-source, fast, and horizontally scalable graph database that is written entirely in Go. Dgraph's goal is to provide Google product scale and throughput, with sufficiently low latency to work real-time user queries, up to terabytes of structured data. Dgraph sustains GraphQL-like query syntax and returns in the protocol, JSON, and buffers over GRPC and HTTP [14].

According to our search, we found only 14 companies that use the Dgraph in their tech stacks, including PlayNet, XXXX-trial, and Leader Tech Stack. So, when Dgraph operates with businesses that need more support and combination for analytics or knowledge management, they can leverage their partnership (with Capventis) to provide corporations with the insights they want [5]. Capventis review and analytics teams can help plan, design, and improve end-to-end solutions and services. Capventis cooperates with its customers to increase customer commitment and experience management over the business from the original method to large-scale integrations or migrations [5]. Dgraph stores its state in a specific directory and write-ahead log in other locations (w). So if you want to check the on-disk files, we can write our data and shut down the Dgraph, pushing everything in memory to disk [6].

Neo4j stores data in nodes joined by directed typed with relationships with properties on both. It's a high-performance graph store with all the characteristics required of a sophisticated and robust database [6]. This Neo4j is a competitor of the Dgrahp database. Titan is also a scalable graph database that optimizes for saving and querying graphs holding multi-billions of vertices and edges distributed over a multi-machine cluster. Thus, Titan is a very close competitor of the Dgrahp database. Cayley is another open-source graph database encouraged by the graph database behind Freebase and Google's Knowledge Graph. One of the main goals is to be a portion of the developer's toolbox wherever any Linked Data with graph-shaped data are concerned. However, this graph database is also a strong competitor of the Dgraph database [13].

KEY FEATURES OF DATABASE

NO-SQL

Although the NoSQL databases have been about since the 1960s, it wasn't until the early twenty-first century that businesses began to embrace them, particularly for big data and real-time online and cloud applications. Since then, the NoSQL database has grown in popularity and utility, albeit relational databases remain useful. But what should you look for in a NoSQL solution when you first start looking? Here are the five most important characteristics to look for in a NoSQL database:

A variety of models: Whereas relational databases require data to be organized into tables and columns before being accessed and analyzed, NoSQL databases are incredibly flexible for data management. They can easily consume structured, semi-structured, and unstructured data, whereas relational databases are inflexible and only handle structured data. Different data models address specific application requirements. To more readily manage diverse, agile application development needs, developers and architects pick a NoSQL database. Graph, document, wide-column, and key-value data models are all popular data models. Supporting numerous data models is good since it allows you to utilize the same data in various data model classes without managing a separate database.

It's simple to scale: It's not that relational databases can't grow; it's that they can't produce EASILY or CHEAPLY. This is because they're created using a classic master-slave design, which implies scaling UP via more and larger hardware servers rather than scaling OUT or, worse, via sharding. Sharding is splitting a database into smaller bits and distributing it over numerous physical servers rather than a single giant server, which causes operational management issues. Instead, seek a NoSQL database with a peer-to-peer architecture with the same nodes on all nodes. This enables cloud applications to scale quickly in response to data volume and complexity. This scalability also enhances performance, allowing for fast read/write rates and constant availability.

Flexible: Whereas relational databases require data to be organized into tables and columns before being accessed and analyzed, NoSQL databases may handle data in various ways thanks to their multi-model capabilities. Relational databases, on the other hand, are designed to operate primarily structured data and can readily process structured, semi-structured, and unstructured data.

Distributed: Look for a NoSQL database built to distribute data worldwide, which means it can write and read data from many places, including numerous data centers and cloud regions. In contrast, relational databases rely on a centralized application that is location-dependent (e.g., a single site), particularly for write operations. Because data is spread with numerous copies where it needs to be, adopting a distributed database with a master less design allows you to ensure continuous availability.

There will be no downtime: Zero downtime is the final but certainly not least significant critical attribute to look for in a NoSQL database. A master less architecture enables this by allowing numerous copies of data to be maintained across various nodes. If one node fails, no worries: another node has a duplicate of the data for quick and simple access. This is a major attention when you examine the expense of downtime.

NEO4J

Flexible Schema: Neo4j uses a data model known as the graph model. The graph is made up of nodes that are linked to one another. Properties, which are key-value pairs, are used to store data in nodes and relationships [7].

ACID Property: Neo4j can handle all ACID characteristics (Atomicity, Consistency, Isolation and Durability).

Scalability: Neo4j makes it easy to expand your database by increasing the number of reads/writes and volume without compromising data integrity or query processing time.

Reliability: Neo4j's replication ensures data safety and dependability.

Cypher Query Language: Neo4j offers the Cypher Query Language, a sophisticated declarative query language. It is used to establish and retrieve data relationships without the need of sophisticated queries such as joins.

Built-in Web applications: Neo4j includes a built-in Neo4j browser web application for creating and retrieving graph data.

Graph DB: Property Graph Data Model is implemented in Neo4j.

DGRAPH

Binary backups: Binary backups are full Dgraph backups that are saved to cloud storage like Amazon S3 or any Minio storage backend. Backups can also be stored to a network file system shared by all Alpha systems on-premises. These backups may be used to restore a new Dgraph cluster to its condition when the backup was taken. Binary backups, unlike exports, are Dgraph-specific and may be used to swiftly recover a cluster [7].

Change Data Capture: Change data capture (CDC) can be used with a Dgraph to track data changes over time, such as database mutations and drops. Dgraph's CDC implementation allows you to store CDC updates broadcast by Dgraph Alpha leader nodes in Kafka or a restricted file as a sink. Dgraph broadcasts events for all **set** and delete mutations; excluding those that affect password fields, as well as any drop events, when CDC is enabled. The CDC records Live Loader occurrences, but not Bulk Loader occurrences.

Changes to Raft logs trigger CDC events. So, if the Alpha leader node can't contact the sink, Raft logs grow as events are gathered on that node until the sink becomes available again. To minimize disruptions in the transmission of CDC events, you should enable CDC on all Dgraph Alpha nodes.

Encryption of Rest: Encryption at rest is the process of encrypting data that is physically stored in any digital format. It ensures that any user or application cannot access sensitive data on drives without a valid decryption key. As an enterprise feature, Dgraph offers encryption at rest. If encryption is enabled, Dgraph encrypts and secures data using the Advanced Encryption Standard (AES) algorithm. The encryption key file had to be present on the local file system prior to v20.07.0. We've implemented support for encryption keys stored on Vault servers starting with version 20.07.0. This provides an alternative method of configuring the encryption keys required to encrypt data at rest.

DATABASE TECHONOLOGY
NOSQL

Many industries now adopt NoSQL database technology for critical business applications, replacing their relational database deployments to gain flexibility and scalability. Previously only used by Google, Amazon, and Face book, many industries are now adopting NoSQL database technology for critical business applications, replacing their relational database deployments to gain flexibility and scalability. Here are three corporate use cases that NoSQL excels at:

Internet of Things: Over 20 billion items are now connected to the Internet, ranging from smart phones and tablets to household appliances and systems in automobiles, hospitals, and warehouses. The expansion of semi-structured and continuous digital telemetry is increasing the volume, velocity, and variety of machine-generated data. The three well-known difficulties from large data IoT applications that relational databases struggle with are scalability, throughput, and data diversity. On the other hand, NoSQL enables businesses to extend concurrent data access to millions of linked devices and systems, store massive amounts of data, and fulfill mission-critical infrastructure and operations performance needs.

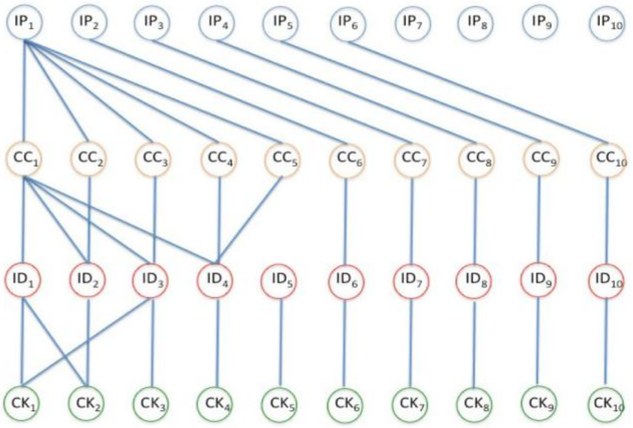
Fraud Detection: Fraud detection is critical for financial service firms to reduce earnings loss, minimize financial exposure, and comply with laws. Customers anticipate instant confirmation when they pay with a credit or debit card. The procedure has an influence on both the company and its consumers. Data – detection algorithm rules, client information, transaction information, location, time of day, and more – is used to identify fraud at scale and in under a millisecond. While traditional databases struggle to fulfill this low latency criteria, elastically scalable NoSQL databases can consistently satisfy it.

Digital Communications: In a business setting, digital communication might take the form of online contact through direct messaging to assist visitors in finding a product or completing the checkout process. The program, like mobile text messaging, may need to accommodate millions of website visits. Relational databases have limitations in terms of responsiveness and scalability, but NoSQL databases, due to their distributed architecture, provide the sub-millisecond responsiveness and elastic scalability that digital communication applications demand.

NEO4J

Fraud Detection: As our lives become more digital, we are increasing financial transactions online. Fraudsters have swiftly responded to this trend, devising ingenious ways to swindle internet payment systems. While criminal gangs may and do engage in this sort of activity, even a single well-informed fraudster may construct a significant number of fake identities and carry out big-scale scams.

Consider the following identifiers in an online transaction: user ID, IP address, geo location, tracking cookie, and credit card number. Typically, these identifiers should have (nearly) one-to-one correlations. Individuals that use several computers, shared devices, and families that share a single credit card number are all examples of natural variations.
Fraud should be regarded a significant probability as soon as the correlations between these variables reach a fair amount. The larger the links between identifiers, the more reasons for concern. Large, closely-knit graphs are particularly strong evidence of fraud

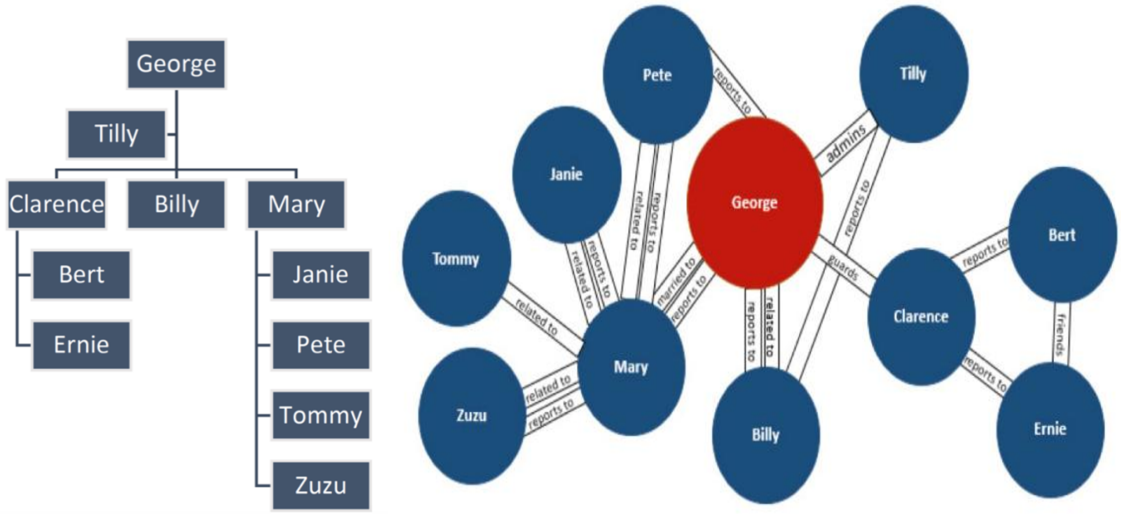

IP_1 IP_2 IP_3 IP_4 IP_5 IP_6 IP_7 IP_8 IP_9 IP_{10}
CC_1 CC_2 CC_3 CC_4 CC_5 CC_6 CC_7 CC_8 CC_9 CC_{10}
ID_1 ID_2 ID_3 ID_4 ID_5 ID_6 ID_7 ID_8 ID_9 ID_{10}
CK_1 CK_2 CK_3 CK_4 CK_5 CK_6 CK_7 CK_8 CK_9 CK_{10}

A graph of a series of transactions from different IP addresses with a likely fraud event occurring from IP_1, which has carried out multiple transactions with five different credit cards.

Fig: A graph of series of transaction from different IP [11].

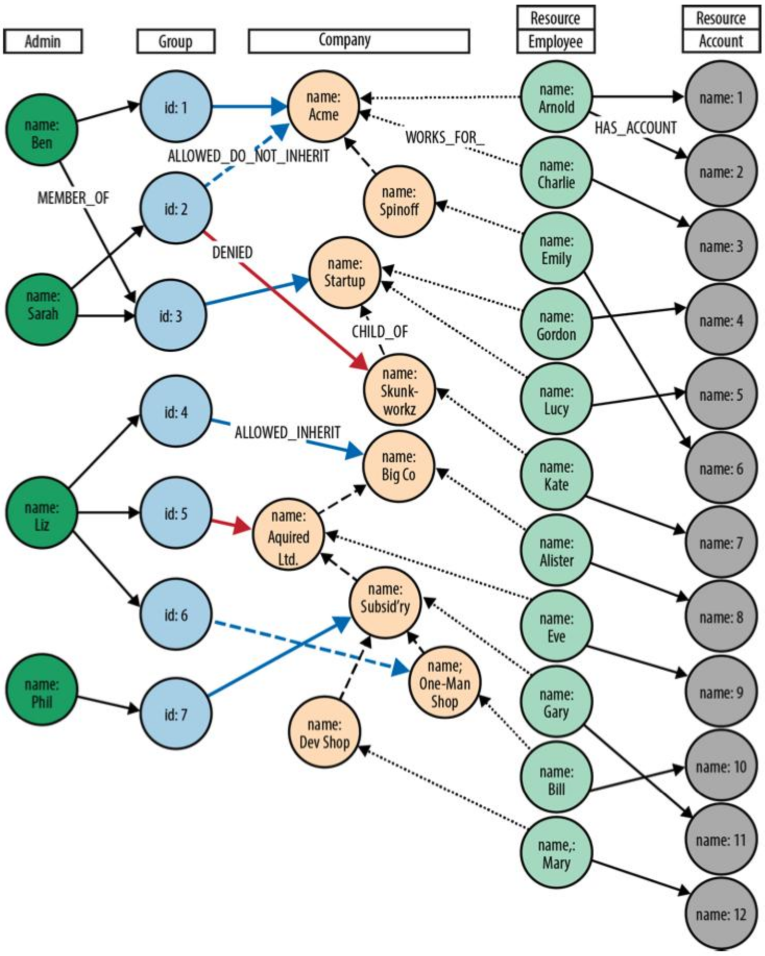
Such schemes can be discovered before they may do considerable damage by putting checks in place and linking them with the proper event triggers. Logging in, making an order, or enrolling a credit card are all examples of triggers that might cause a transaction to be reviewed against the fraud graph. Complex graphs can be detected as a suspected fraud incident even if fan-out is skipped.

Solutions for Master Data Management: A hierarchy in your master data is any structure in which nodes have other nodes above and below them, with various branches. Employee reporting and supervisory structures, such as the one on the left, are examples of a master data hierarchy. A modest ranking like the one on the left is simple to represent and maintain in a relational database. However, as the number of people we model grows, retrieving and keeping the data becomes more expensive. For example, if an employee is promoted, every link in the hierarchy in which the person engages must be reset.



Of fact, in the actual world, such clear hierarchies are uncommon. Employees frequently report to a number of persons, and some reporting relationships exist only for the purpose of transition (such as job shadowing or coverage). Most organizational hierarchies, in reality, are networks with real-life intricacies and a variety of interactions. As company demands evolve, traditional hierarchies must be recast as easier and more flexible networks to describe with a graph database. While the example given concerns personnel reporting linkages, master data networks apply to product listings, document relationships, and sales or customer data as well.

Identity & Access Management: Telenor Norway is a Norwegian telecommunications business with a global presence. It has allowed its largest commercial clients to self-service their accounts for numerous years. Administrators inside each of these client companies can add and delete services on behalf of their workers using a browser-based application.



The application uses a complex identity and access management system that allows privileges to millions of users over tens of millions of product and service occurrences to ensure users and administrators see and change only those parts of the organization and services they are authorized to manage.

Telenor opted to replace its previous IAM system with a graph database solution due to speed and responsiveness difficulties. Their first solution relied on a relational database with recursive JOINs to describe complicated organizational structures and product hierarchies.

Their most essential queries were unacceptably sluggish due to the join-intensive methodology. On the other hand, Telenor recognized the performance, scalability, and adaptability required for addressing their identity and access management demands after implementing a graph database solution, lowering searches that formerly took minutes to milliseconds.

DGRAPH

Faster Time-to-market (Mooncamp): Customers at Mooncamp have been astonished and thrilled. In one case, Mooncamp was able to assist a customer with a Microsoft Teams integration that took only three days to complete (versus a quite lengthy roadmap offered by a competitor). Customers are astounded by Mooncamp's ability to create and offer solutions so quickly.

Because Mooncamp works directly with its clients to build solutions, the ability to rapidly and efficiently incorporate customer input, requests, revisions, and new features was critical. Companies pick Mooncamp because they trust the company's ability to offer features without sacrificing quality or performance, and Dgraph has played a key role in fostering that trust. Mooncamp uses Dgraph's capabilities to display speed and performance outcomes that Mooncamp produces over a 30-day period while providing product demos. This allows potential clients to get a firsthand look at Mooncamp's system capabilities.

Fully Distributed (Actually) But Cost-Effective: (Boxall) sought a genuinely distributed database because FactSet sought to develop a durable and internationally co-located database system. This means being able to achieve global consistency across various datacenters in Amazon Web Service's compute and storage cloud by horizontally sharding on the fly (rather than creating entire replica sets). Many graph databases can't shard and have large infrastructure footprints. FactSet sought to avoid complications in the future on both fronts. While FactSet has not yet implemented sharding, it is certain that as Symbology's use expands and needs evolve, sharding can be rapidly and simply done using Dgraph.

How Dgraph Assists Delhivery in Delivering More Packages in Less Time: Delhivery required a database that could easily capture the changing elements of a logistics-focused business, such as iterative data modeling, low-latency queries, and cheaper maintenance costs. The team rapidly understood that they would require a straightforward, easy-to-operate NoSQL backend, such as a graph database, to meet their use case.

Dgraph allows data to be exchanged through files and APIs utilizing non-proprietary and open JSON/RDF formats. With Dgraph, Delhivery was able to map their use cases quickly and easily, allowing them to focus on business-critical elements like performance and scalability.

REQUIRED SKILLS FOR DATABASE ADMINSTRATORS

To identify the critical technical skills required for database administrators using graph databases, this paper will first look at the general skills database administrators need before looking at few of the requirements needed by several platforms. A database administrator is required to execute many activities and tasks as part of maintaining the databases. The following list shows the tasks that a database administrator may conduct [8].

- Making sure if they are any notifications from the SQL server.
- Database schema and contents are contrasted and harmonized
- Maintain verification and encryption for the backup of the databases.
- Author documentation and create scripts to execute undertaking [8].

A database administrator should have increase skills and the capabilities to utilize the required tools to execute the tasks. The utilization of the tools may include the automation of repetitive and/or scheduled tasks. The following list shows skills a database administrator should always be looking to sharpen [8].

- Execution of optimization.
- Storage of data.
- Ability to read and review query methods.
- Build scripts.
- Interactions of operating systems.
- Improve knowledge of SQL.
- Retrieval and back up.
- Improve knowledge of trigger and storing procedures.

Base knowledge of SQL should never be overlooked. This is the basics. All database administrators need to increase their data manipulation language (DML) and Data Definition Language (DDL). As an admin increase their knowledge on this, they will realize that there are methods to automate functions, which will increase efficiencies [8].

Utilization of tools such as SQL squirrel can obtain assistance to issue SQL commands. Tools such as SQL squirrel can be employed with many database servers including neo4j, and NoSQL [8]. Another aspect database administrators should be familiar with is task such as Indexing or identification of tables. As such tasks are very routine, the skill is vital. This, too, will assist in automation. Some tools assist in modifying, importing, exporting, generating databases, records, indexes, and tables [8]. All database admins need to be familiar with how the data is retained in the database. The efficiency depends on storage media is utilized. For example, if the data is stored in the cloud due to network limitations, recalling data may be slowed [8]. This is due to the network bottle capping the flow of data. The local disk may make the data unsafe as it can be easily erased [8].

A knowledgeable database administrator will have the tools to monitor the operations per second, the inputs, and the outputs. Tools can be obtained to do monitoring, such as SQL queries [9]. Database admins to effectively manage databases needs to be familiar with query plans. Database administrators should be satisfactorily fluent enough to debate for or against alterations in the Indexing. Database administrators need to have enough experience in reviewing and reading query plans so that when altering, compensation is giving so that the operational efficiency of the database is not reduced [8]. Using the correct tools will reduce the workload by assisting the database administrator. Utilized tools can assist with generating and altering triggers and other timed events. In addition, it will assist with optimizations, moving data, and the repair of databases [8]. The interaction between the database management system (DBMS) and operating system (OS) is vital to upholding the integrity of the data. A database administrator's critical skill is in-depth knowledge of Light Directory Access Protocol (LDAP), naming conventions, active directory, and security settings [8]. The utilization of tools will assist database administrators by modifying settings; determine the required protocol for the administrator services, and naming convention. Such a tool is DB Visualize. Database administrators reduce doing repetitive tasks by letting scripts do them. As a graph database administrator, the administrator can build scripts to do repetitive work. This will automate functions such as the termination of servers. Database administrators with knowledge of script building can reduce time and increase their productivity or efficiencies [8].

Nearly all database administrators should generate triggers and stored protocols while maintaining the required logic for deployment. Triggers and protocols are more accurate than the standard SQL and are more capable when the correct logic is utilized. Database security is vital. In many industries, high visibility is given to the security of the database and its ability to maintain data safely. Database administrators should be able to have a good plan of data retrieval in case of failure [8].

Database administrators need to develop soft skills to be effective in their job. The following list shows some of the crucial soft skills they need to develop. Legal skills are required to avoid any legal issues that may arise. Data of people is given significant importance, and managing it without understanding may cause legal issues. Project Management Skills. These skills allow the management of resource to achieve the required output with the minimum input. Understanding the how to generate PERT chart will reduce unwanted effort. Also work in bigger organization learning to read gnat charts will show the progress of projects and how timely the administrators need to be. Teamwork Skills. These skills allow better interaction with colleagues. This allows increased efficiencies. Different individuals’ strength can be used to their advantage. Communication Skills. Communication skill is key for database administrators, just like with everyone. This helps in obtaining information faster. It also helps understanding colleagues better, which will improve efficiencies. Ability to write is important. When the database administrator is not employed the successor must be able to read reports. It is also important that they be able to write well so emails and communications are understood better [8].

SKILLS FOR GRAPH DATABASE ADMIN

In comparing the skills needed between graph and neo4j, neo4j has a clear disadvantage as the developers will require files to be converted into CSV files or another supported, which is currently limited [7]. Dgraph allows the importation of files in RDF. The task to convert specific files may require the developer to have soft skills such as being highly organized and having patience as one RDF file may turn out to be multiple CSV files [9]. To utilize dgraph database administrators and developers must know graph query languages that responds in JSON. For neo4j developers will require to know cipher [11]. Both dgraph and neo4j allows the usage of API’s. To utilize this need for developers and database administrators with skills in API is required [12]. Dgraph is better for developer and database administrators as it is compatible with multitude of additional products. This allows the database administrators not to be specialized.

CONCLUSION

Even though both dgraph and NoSQL are great platforms, it was clear that the intent to use at an enterprise level in commercial application neo4j was the best. It had most of the features as NoSQL, including the ability to scale. Dgraph had logging of audit trails, but this could be achieved in neo4j as well. During our research, it was also seen that there were not man commercial uses of dgraph. Neo4j had other competitors that relied heavily on NoSQL, but none were as advanced as neo4j. Neo4j would be hard to adapt as some training and improvement of technical skills will be required for the database administrators and data administrators, but this can be overlooked due to the added functionality and advantage of neo4j.

REFERENCES

[1]. Lowing,J.(2018).Xfinity xFi & User Personalization with Graphs. Retrieved from:
[2]. Neo,J.(2021).Why Neo4j? Top Ten Reasons. Retrieved from: https://neo4j.com/top-ten-reasons/
[3]. Craft,C.(2020).Neo4j competitors. Retrieved from: https://craft.co/neo4j/competitors
[4]. Reed,J.(2015). Why Marriott is transforming their legacy systems with NoSQL. Retrieved from:
[5]. Center,N.(2021).Dgraph and Capventis Partner to Advance DT.
[6]. StacS,I.(2021).Dgraph Alternatives & Comparisons. Retrieved from: https://stackshare.io/dgraph

[7]. Rawal, P. (2017, January 23). Neo4j vs Dgraph - The number speak for themselves. Dgraph Blog. https://dgraph.io/blog/post/benchmark-neo4j/.

[8]. Thomas, S. (2016, February 5). 7 Essential Skills and Tools for Database Administrators. Designrshub. https://designrshub.com/2016/02/database-administrators-skills.html.

[9]. DataStax. (2021, January 27). The 5 Features to Look for in a NoSQL Database. The 5 Features to Look for in a NoSQL Database | Datastax. https://www.datastax.com/blog/5-features-look-nosql-database

[10]. Stephan,T.(2015, October 30). 10 use cases where NoSQL will outperform SQL. Network World. https://www.networkworld.com/article/2999856/10-use-cases-where-nosql-will-outperform-sql.html

[11]. Sadowski, G. (2016).How Graph Databases Stop E-commerce Fraud in Real Time

[12]. https://neo4j.com/blog/graph-database-ecommerce-fraud/

[13]. Neo4j. (2021, May 28). Top 10 Reasons for Choosing Neo4j of Your Graph Database. Neo4j Graph Database Platform. https://neo4j.com/top-ten-reasons/

[14]. Burton, A. T., & Burton, T. (2020, July 29). The Top 10 Enterprise Use Cases for NoSQL. The Couchbase Blog. https://blog.couchbase.com/top-10-enterprise-use-cases-for-nosql/.

[15]. Encryption at Rest. dgraph. (n.d.). https://dgraph.io/docs/enterprise-features/encryption-at-rest/.

[16]. Graph Database Use Cases & Examples. Dgraph. (n.d.). https://dgraph.io/case-studies.

[17]. Kerner, S. M. (2021, April 20). Dgraph GraphQL database users detail graph use cases. SearchDataManagement. ttps://searchdatamanagement.techtarget.com/news/252499595/Dgraph-GraphQL-database-users-detail-graph-use-cases.