Mini Project 2

Assignment 1

Neo4j

Invention of Neo4J began sometime back in 2000 when the creators encountered one of the major performance bottlenecks with their RDBMS application. While they were able to construct multiple data models in RDBMS they experienced challenges to retrieve the data with queries, as it showed overhead on the performance. The objective of project Neo created by the creators of the database was to retain the consistency of the ACID properties with the speed, flexibility in schema, speed of the queries, ease of creation in the data models for retrieval (Jordan, 2014).

Key features of Neo4j:

- J in Neo4J stands for Java (Jordan, 2014).
- It requires Java development kit (JDK) to run Neo4j (Jordan, 2014).
- It can be installed on Windows, Linux, and Unix operating systems (Jordan, 2014).
- Neo4j comes with web browser utility tool to provide data visualization and other features such as speed (Jordan, 2014).
- The browser can also display the relationships between the data models and graphs. It also provides options for configuration settings (Jordan, 2014).
- Cypher is the interactive query language in Neo4j that is much similar to structured query language for performing database operations (Jordan, 2014).
- Data modeling can be architected in Neo4j. The differences between entity-relationship model and Neo4j data modeling are, in an entity relationship model there are entities, relationships, and attributes, whereas in Neo4j, there are nodes, relationship, and properties (Jordan, 2014).

- Neo4j has all or nothing approach to transactional commits and rollbacks when performing the database operations. It completely updates the database or it does not update at all to keep the integrity and consistency of the software (Jordan, 2014).
- SQL database operations such as INSERT, CREATE, SELECT, UPDATE, SET, DELETE, ORDER BY, SKIP, LIMIT, MATCH, and WHERE clause can be leveraged (Jordan, 2014).
- Neo4j supports importing data from the external sources (Jordan, 2014).
- Neo4j supports the development environments of .NET, Java, PHP, Python, Ruby, and Spring Data Neo4j (Jordan, 2014).

MongoDB

For several decades, RDBMS databases have reined the world with performance and speed. However, as the volume and variety of data formats grew the one-size-fits-all approach did not work for variety of databases and data formats. The inventors of MongoDB wanted to specialize in the document store database with high scalability, flexibility, eventual consistency, and lightning speed, so that they were not creating the similar NoSQL databases other vendors created (Membrey, Hows, & Plugge, 2014).

Key features of MondoDB:

- MongoDB can store the documents in open standard BSON format (Binary JavaScript Object Notation). With BSON format MongoDB can extract the data with extreme speed and without the requirement of conversions of the internal code. However, BSON document format may take more storage space than JSON format (Membrey, Hows, & Plugge, 2014).
- Though BSON consumes more space to store the data, it's speed and ability traverse the data back and forth, makes MongoDB as a choice for many organizations dealing with petabyte scale of documents or text-based data. MongoDB can be retrieved through several programming languages

such as Java, C, C++, Ruby, C#, and Python. BSON format is highly compatible for most of the programming languages, so there is no need for doing conversions on JSON file formats (Membrey, Hows, & Plugge, 2014).

- MongoDB supports dynamic queries without requiring the database format to be fixed in some schema before running the query. (Membrey, Hows, & Plugge, 2014).
- MongoDB can index the documents to speed up the query extraction time. This will aid to accelerate the execution time on the database. MongoDB can also build geospatial indexes for location-based data (Membrey, Hows, & Plugge, 2014).
- Most of the databases maintain multi-version concurrency control for the consumers to visualize different versions of the data to avoid any further updates to the data while other updates are in progress or maintain versioning of the database. MongoDB adopts the technique information-in-place with instant writes to the database (Membrey, Hows, & Plugge, 2014).
- MongoDB stores the binary data up to 4 MB with GridFS approach per document (Membrey, Hows, & Plugge, 2014).
- MongoDB performs replication of the data with master-slave architecture ensuring there is only database with eventual consistency (Membrey, Hows, & Plugge, 2014).
- MongoDB has the auto-sharding capabilities by performing the resplitting and recombination of the data (Membrey, Hows, & Plugge, 2014).
- MongoDB can perform map and reduce functions by mapping and filtering all the documents that meet particular criteria and finally processes the data for reduction (Membrey, Hows, & Plugge, 2014).

Key differences:

- Both Neo4j and MongoDB are NoSQL databases. However, Neo4J is a graph database (Membrey, Hows, & Plugge, 2014).
- MongoDB is schemaless and provides holistic view of the data with eventual consistency with update information-in-place principle (Membrey, Hows, & Plugge, 2014).

- MongoDB has aggregate framework, and Neo4J does not have aggregate framework (Membrey, Hows, & Plugge, 2014).
- Graph database like Neo4J comply ACID framework with complete fine-grained atomicity of the transactions (Membrey, Hows, & Plugge, 2014).
- MongoDB provides flexibility, high scalability in a distributed environment. Therefore, it requires eventual consistency of the database than instant ACID compliance (Membrey, Hows, & Plugge, 2014).
- MongoDB does not create relationships between the database models, as each data set stored in the document store of the database is disaggregated and independent. A graph system requires handling the complex relationship of the database, while NoSQL database does not require handling the complex associations between the data models (embrey, Hows, & Plugge, 2014).
- Neo4J enables navigation through the graphs as a tree, where MongoDB cannot provide visualization of the document stores as graphs (Membrey, Hows, & Plugge, 2014).

References

Edward, S. G., & Sabharwal, N. (2015) (MongoDB image). Practical MongoDB: Architecting, Developing, and Administering MongoDB (1 ed.). New York City, NY: Apress .

Jordan, G. (2014). Practical Neo4j. New York City, NY: APress.

Membrey, P., Hows, D., & Plugge, E. (2014). MongoDB Basics. New York City, NY: APress.

The above are from the following article https://medium.com/@gp_pulipaka/the-differences-between-mongodb-and-neo4j-babd234ac1f7