Predictive Performance Comparison Analysis of Relational & NoSQL Graph Databases

Wisal Khan
National University of Computer and Emerging Sciences,
Islamabad, Pakistan

Ejaz ahmed
National University of Computer and Emerging Sciences,
Islamabad, Pakistan

Waseem Shahzad
National University of Computer and Emerging Sciences,
Islamabad, Pakistan

***Abstract*—From last three decades, the relational databases are being used in many organizations of various natures such as Education, Health, Business and in many other applications. Traditional databases show tremendous performance and are designed to handle structured data with ACID (Atomicity, Consistency, Isolation, Durability) property to manage data integrity. In the current era, organizations are storing more data i.e. videos, images, blogs, etc. besides structured data for decision making. Similarly, social media and scientific applications are generating large amount of semi-structured data of varied nature. Relational databases cannot process properly and manage such large amount of data efficiently. To overcome this problem, another paradigm NoSQL databases is introduced to manage and process massive amount of unstructured data efficiently. NoSQL databases are divided into four categories and each category is used according to the nature and need of the specific problem. In this paper we will compare Oracle relational database and NoSQL graph database using optimized queries and physical database tuning techniques. The comparison is two folded: in the first iteration we compare various kinds of queries such as simpler query, database tuning of Oracle relational database such as sub databases and perform these queries in our desired environments. Secondly, for this comparison we will perform predictive analysis for the results obtained from our experiments.**

***Keywords*—Big Data; Hadoop; MapReduce; Relational Databases; NoSQL Databases; Decision tree**

I. Introduction

Now-a-days, data is being expanded rapidly in the industry. The nature of data is varied and diversified such as unstructured, semi-structured and structured data. The issue is not only how to store and access such amount of big data but also need to extract meaningful knowledge from such data rapidly. Relational databases are being used for such data for the past more than three decades. Many organizations have been using the traditional databases for handling and analyzing structured data efciently. Traditional or Relational databases require declarative language such as SQL to manipulate structured data. Relational databases are based on data consistency and can process the data at certain limit [9]. To manage large datasets using relational databases, organizations are required to increase their system capacity such as RAM, Disk, optimized methods of accessing data etc. The systems are also mostly supported limited by capacity.

Data is stored in the form of punch tables or punch cards. It is not an appropriate way to read and understand data [11]. It is impossible to index cross reference data by eliminating data inconsistency. Therefore relational databases are used to store data in the form of tables. Sometimes, theses tables were human readable. But as soon as to normalize (1NF, 2NF, 3NF etc.) the tables to eliminate the duplication, inconsistencies many elds start referencing and generating referential integrity and data becomes difcult to understand and maintain without complicated join queries. The ACID property of a relational database describes that we can trust once the data is committed. It would be accessible to future queries If it is expensive to nd data, we add indexes, sub databases, partitioning and query optimization which make data access faster. If we do a bunch of joins then we have to perform query time index lookup for each and every join. It works well but if we have 20 or more tables in a query, it is really expensive and becomes more expensive as data size grows and joins increase. To overcome such issues, researchers used parallel databases and distributed databases approaches to process large sparse dataset efciently and can be used for decision-making purposes. This approach can also handle the dataset at certain size which is varied and large in nature.

Now-a-days many organizations are rely on unstructured data such as emails, blogs, audios, videos, images and such data is generated at very high speed. Big data means when the dataset is large enough, cannot be processed by traditional databases efciently [4]. For example, data is expanded due to machine generated data, scientic experimental data, Facebook scale dataset and Google BigTable. Basically, big data has three characteristics: (1) Volume: Data is in huge and large amount. (2) Velocity: Data is generated or accessed at very high speed. (3) Varied: The generated data is varied in nature.

Keeping in mind such issues of big data, NoSQL databases are born. NoSQL stands for Not Only SQL and the word was used for the first time in 1998. NoSQL databases are used for processing and analyzing big data efciently. NoSQL databases are based on BASE (Basically Available State and Eventually Consistent) property [9]. NoSQL databases are horizontal scalable databases while relational databases are vertical scalable databases. To process large, sparse, irregular and connected dataset, new technology and storage methods are raised. Graph database is one of the NoSQL databases type and is used to process the large connected data set perfectly. For example, Facebook scale dataset and Google+ dataset which consist of billions of edges, millions of update rates per second and require complex storage system.

Graph databases are designed for connected data and are used in many applications such as Facebook, Amazon, LinkedIn and many more. The literature review presented the comparison between relational database (MySQL) and NoSQL graph database (Neo4j) [12]. The graph database performed better than relational database as shown by the researcher contribution.

There are mainly four types of NoSQL databases: Key value, Document, Column and Graph databases. (1) Key-value databases are based on hash table. Hash table uses unique key and a pointer. Key and Pointer both are used to refer to particular item. Hash table is suitable for processing large number of records. (2) Document databases are used to store data as key/value but different from key/value database. We can search document by key as well as by the contents of a document. The document is stored in XML, JSON (JavaScript option notation) and BSON (Binary JSON) forms. MongoDB and CouchDB are the examples of Document Database. (3) Wide-column databases follow hybrid architecture; means it uses characteristics of relational databases and stores schema of key-value databases. These are suitable for distributed data in cluster environment. Examples are Hbase, Cassandra and Accumulo. (4) Graph databases are the main focus of this document are designed to process and analyze connected data efciently. Graph databases not only store information about objects but also store their relationship. Neo4j, Pregel, ArangoDB and OrientDB are the example of graph databases. The store data is in much more logical fashion. The graph databases represent the real world and prioritized presentations, discoverability and maintainability of data relationship. [19], particularly native graph databases are designed and optimized for storing and managing graphs. Native graph databases provide a natural adjacency index and hence do not heavily depend on indexes. The main benefits of native graph databases are performance and scalability. The relationship in a native graph databases attached to a node established a direct connection naturally to other related nodes of interest. Due to such direct connection or locality, the traversing of a graph by using graph queries become much easier by chasing the pointer. Therefore, native graph databases can traverse millions of nodes per seconds in contrast to joining data through global indexes and it is too slow in relational databases.

[4] the volume of data grows 20% annually of the world data and will be 50 times by 2020. In the near future, the market value of big data will be 16.9 billion while the same value was 3.2 billion in 2011. With the rapid growing of data, 2020 data production will be 44 times larger than it was in 2009. According to survey, Walmart database performs 1 million database transactions and approximately generate more than 2.5 PB (Peta byte) of data each hour. By the end of 2011, International Data Corporation (IDC) indicated that 1.8 ZB (Zeta byte) of data was created and 2.8 ZB of data will be created by the next few years. IDC also estimates the growth rate of the following technologies: (1) by 2020, enterprise data will reach 40 ZB. (2) By 2020, internet Business to Business (B2B) and Business to Customer transactions will reach 450 billion per day. Big data is generated by various resources such as Internet of things (IoT), self-quantified multimedia and social media data.

[18], the decision trees are being used in various domains such as data mining, engineering and artificial intelligence etc. Mainly there are two goals of decision tree: (1) yield perfect classifier and (2) provide the problem predictive structure. Decision trees are simple, easy to understand and generate the results in symbolic and visual terms that communicate very well. In the breast cancer prediction, the decision tree J48 algorithm has the highest sensitivity than all other algorithms (Logistic regression model, Artificial Neural Network (ANN), Nave Bayes etc.). The decision tree J48 returns 85.6% accuracy in the breast cancer prediction.

For our experiment we used MedCare (Medical Diagnostic System) dataset. Our experiment will describe performance comparison analysis of relational database (Oracle 11g) and NoSQL graph database (Neo4j). The Medcare database is our own in-house developed schema as case study of hospital healthcare system. The Medcare schema consists of main large data tables such as Patients, Patient visit, Dependent, Medical staff, Patient IssueMed, Patient history, Patient Appointment and Patient History etc.

The rest of the paper is organized as follows. Section 2 explains related work and concepts; Section 3 describes our designed research methodology; Section 4 explains our experimental framework; Section 5 discusses the analysis of our experiment. Finally section 6 describes conclusion and future work.

II. Related Work

A. Big Data Transaction Approach

Figure 1 shows the transaction of big data coming from different sources. The small block on the bottom left represents the Enterprise Resource Planning (ERP), in this phase a different types of data is collected about an organization i.e. purchase details, purchase records, payment records etc. As this includes structured data, therefore, this data is not as much bigger in size. This data is further handled by CRM (Customer Relationship Management), which collects data about an organization, all the entities directly and indirectly linked with organization like emails, chats, database, telephone, etc. As CRM involves data from databases and other resources ,therefore, this has comparatively bigger size which can cover multiple Gigabytes. The third layer is about Web. Web collects data from different CRMs and joins multiple networks and branches of organization, therefore, the size of data increases up-to multiple terabytes while handling Webs. The fourth layer in Figure 1 represents the Big-Data, which covers data from different resources like data coming from different organizations, mobile webs, social networks, machine generated data, scientific applications etc. This type of data is scaled up to multiple Petabytes.

B. Big Data Approaches

[1], The data governance techniques become more popular to take the important decisions with the passage of time for the business communities across the globe. Organizations are required to maintain good quality of their data for effective data governance by putting more efforts on its data. [6], There are many open issues and research challenges in analyzing huge amount of information regarding data warehouse and OLAP research. Analyzing large amount of data requires complex strategies to extract valuable knowledge stored in archives. Two techniques are proposed in the above mentioned sentences. First, how to extract the hidden structure from the

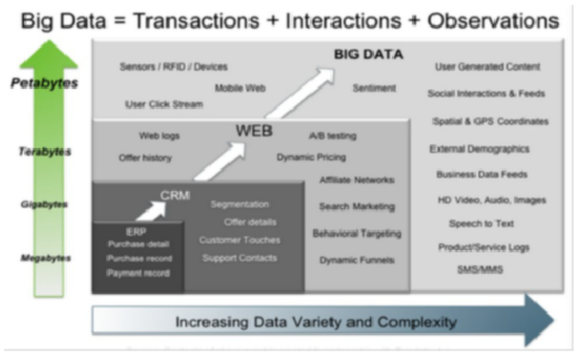


Fig. 1. Big Data Transactions with Interactions and Observations. http://hortonworks.com/blog/7-key-drivers-for-the-big-data-market/

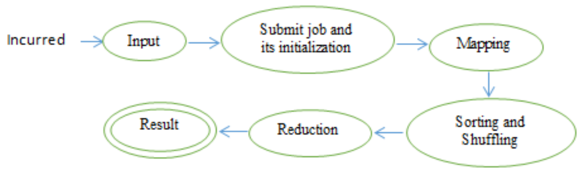


Fig. 2. MapReduce Tasks process flow [17]

massive amount of data, that is varied in nature (e.g., legacy frameworks, Web, exploratory information stores, sensor and stream databases, informal organizations). Second, once the structure is extracted then how it will be plotted on charts, dashboards for decision making purpose?

[16], Hadoop is used for large data scale analytics. Hadoop is not a single tool rather it is a framework that supports data-intensive parallel applications. It can work with 1000s on compute nodes. Hadoop is open source software and works on distributed model. Hadoop has no scalability problem because it divides computation into smaller pieces on different nodes in a cluster environment. At very high level it has two main components HDFS (Hadoop file system) and MapReduce. The objective of Hadoop is to support running applications on big data. Dean and [7], MapReduce is a programming model used to process large data sets in the distributed environment. MapReduce process is distributed and replicas of data over the shared nothing cluster by using two main functions i.e. Map Function and Reduce Function. Inside at Google over the recent years more than ten thousand unmistakable MapReduce programs have been implemented, and a normal of one hundred thousand MapReduce tasks are executed on Googles bunches (cluster) each day, handling a sum of more than twenty petabytes of information for every day. Figure 2 describes the process flow of MapReduce tasks.

[2], Hadoop++ is used to collocate the related data by performing heavy weight changes. It creates Trojan File from the co-grouping of the two input files. In Hadoop++, users are required to reorganize their input data and therefore, Hadoop++ provides a static solution. However, this approach does not modify the core architecture of Hadoop. Only the two files can be collocated in Hadoop++, when two files are created by the same job. In Hadoop++ data is reorganized and loaded from the scratch when new data is added incrementally to the files. Hadoop++ is not suitable for the applications where

TABLE I. HADOOP USAGE

SN#	Specified Use	Used By
1	Recommendation system	Facebook
2	Data warehouse	Facebook
3	Searching	Yahoo, Amazon
4	Log Processing	Facebook, Yahoo
5	Analysis of videos and images	New York Times

data is added to the files incrementally like log processing file. [8], to balance the load, HDFS data placement policy placing the blocks randomly. HDFS does not consider any data characteristics. Particularly, in HDFS there is no way to collocate (arrange) same data on the same node. Cohadoop is used to address the above short coming. CoHadoop is the extension of Hadoop with light weight mechanism. Definition of CoHadoop: CoHadoop is the extension of Hadoop infrastructure, where:

- HDFS accepts hints from the application layer to specify related files. Hints are:
    1) Collocating log files with reference file for joins.
    2) Collocating partitions for grouping and aggregation.
    3) Collocating index files with their data files.
    4) Collocating columns of a table.
- Based on these hints, HDFS tries to store these files on the same set of data nodes.

[15], BDAS (Berkeley data analytics stack) is developed at AMPLAB in UC Berkeley. BDAS is used to analyze big data. BDAS is integrating software components to understand and analyze big data. Big data analytics research lab, developed and designed at Frankfurt, is also used to analyze big data and unify research activities regarding huge information. AT UC Irvin, ASTERIX project has been developed and is also used to tackle and examine the big data. CSAIL is the MIT big data Laboratory. CSAIL is emphasizing on developing the new technologies for handling big data challenges of next generation. Its focus is developing scalability, easy to use and easy to implement across various platform to handle big data.

[4], The organizations were unable to process and manage vast amount of data by using the existing tools in the past. By handling big data, new technologies are implemented to improve performance and decision-making support. Big data technologies are depended on the three milestones. The milestones are 1) minimize hardware cost 2) before committing significant company resources, check the value of big data and 3) reduce processing costs. Various technologies are used to handle big data like 1) Batch based processing technologies such as Apache Hadoop, Skytree Server, Talend Open Studio, Jaspersoft, Dryad, Pentaho, Tableau and Karmasphere. 2) Technologies based on stream processing such as Storm, Splunk, S4, SAP Hana, SQLstream s-Server and Apache Kafka. 3) Big data processing methods such as Bloom Filter, Hashing, Indexing and Parallel Computing. Table I describes hadoop usage [17].

C. Handling Structured data in Hadoop

[3], HadoopDB database systems are using Hadoop as the task coordinator and network communication layer. HadoopDB

connected multiple single nodes with database systems. MapReduce framework is used to parallelize the queries across the nodes. Moreover, much of possible work of a single query is stored within the corresponding node databases. To handle fault tolerance and function in heterogeneous environment, HadoopDB inherited job tracking and scheduling implementation from Hadoop. HadoopDB uses database engine for managing much of query processing to gain parallel database performance. HadoopDB includes four components to the core architecture of Hadoop. 1: The Database Connector, 2: Meta Information, 3: Data Loader and 4: SQL to MapReduce to SQL (SMS) Planner.

[5], Map reduce is used to manage and analyze large unstructured data efficiently and the dominating architecture for handling big data on cluster. HiveQL is used in Hadoop for handling structured data and a data warehouse infrastructure tool that creates interaction between user and HDFS. HiveQL is SQL-Like query language. HiveQL generates query execution plan by using naive rule based optimization techniques and does not guarantee efficient query plan. There are many ways to execute map reduce operations such HiveQL is used to process structured data for map reduce. Clydesdale is the new approach to handle structured data efficiently and does not bring any changes in the core architecture of Hadoop. Clydesdale follows many techniques such as columnar storage, star join and block iteration. Clydesdale is suitable when the workloads fit the data as star schema. The experiments have shown that Clydesdale performed approximately 83x faster than hive by using star schema benchmark.

D. Relational Databases and NoSQL Databases Comparisons

[9], Relational databases users are required to increase the system capacity like CPU and RAM to handle the large amount of data of a certain limit. Relational databases are vertically scalable databases. To handle massive amount of semi-structured data and unstructured data, NoSQL databases are used. NoSQL databases follow the principle of BASE (Basically, Available, Soft state and Eventually consistent). Relational Databases provide better data integrity, security and trustful transactions. While NoSQL databases are suitable for large volume of data of various format. NoSQL Databases handle big data at lower cost and required minimum overhead. NoSQL databases are horizontally scalable databases just by adding new server in the cluster environment. Commodity hardware is used to store big data in the cluster.

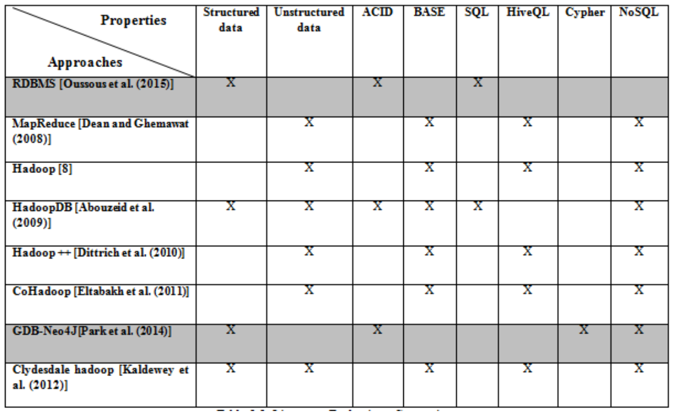
[10], The CAP theorem is presented by Eric Brewer and stands for Consistency, Availability and Partition Tolerance. Today CAP is implemented and adopted by large companies e.g. Amazon. In CAP: Consistency means after performing some writes operation by the system, how a system will be in a consistent state. Availability means system must be designed in a way in which updated data is always highly available to the users after performing the writes operation. Partition Tolerance means the system must be able to continue its operations if data is distributed over various nodes in the network. Traditional databases' focus is on the consistency and partition tolerance. While NoSQL databases follow availability and partition tolerance. NoSQL databases are commonly used to handle big data (large data sets). Amazons Dynamo follows availability and partition tolerance of CAP theorem.

[11], In RDBMS, the data is stored in the form of tables (rows and columns). To avoid repetition of records RDBMS uses primary key concept. Therefore, data is consistent and reliable. NoSQL databases follow different approach. The NoSQL databases split the data on different systems to accelerate the processing and perform the task fast and efficiently. RDBMS follows the rigid schema and becomes mature with the passage of time. It is hard but possible to bring the changes in the mature schema if required. The same is not true for NoSQL databases. NoSQL databases schema, developed gradually and is flexible for stored data in row including NULL values problem. In RDBMS NULL values problem occur persistently.

[12], Typical data structure for storing data provenance information is the Directed Acyclic Graph (DAG). For the development of a data provenance system whether the fundamental innovations like traditional databases (MySQL) and NoSQL databases (Neo4j), would be more viable or not. In software engineering and in computer science, Graph is one of the key information reflection (abstraction). We have various types of applications of graph and each and every every graph application required to store and query the graph. Many social network sites such as Facebook, Google and LinkedIn are using graph databases for storing their huge amount of data. Most commonly, graph is the appropriate data structure for modeling objects' interactions.

[13], Healthcare systems (public and private) in United States generating more data and requiring new technology to handle data analytics effectively. Data driven approach is used to handle data analytics in healthcare systems by using two independent tasks, data management and data services. Here, data management means storing the data with minimal redundant structure and error free. Data services describe various analytics queries such as join, search and statistical queries. The problem appeared due to the gap between data management and data services in relational databases. To overcome this problem, they presented an approach to convert third normal form (3NF) of relational databases in equivalent graph of Graph database. A graph database uses Denormalized forms. A graph database does not require creating more tables and replicating them unlike relational databases. For example, Neo4J is suitable in OLTP (online transaction processing) environment. Pregel is used where high latency and high through put have high priority. Their experiments have shown that Graph database performed better than relational database (MySQL) in heterogeneous environment of healthcare systems of United States in OLTP.

[14], the comparison of relational data model and graph data model has been discussed by the author. According to him, the data model consists of three properties: integrity rules, data structures and query operators. From last few years, most of the systems natures have become more and more connected. The connected nature of data is not easily handled by the relational data model. The graph data model is the appropriate choice for such systems such as geographical systems, biological systems and social networks. In graph data model, the relationship is stored at individual level while the relationships are handled at the conceptual level by the relational data model. No additional computing is required when adding new relationship in the graph data model while the same is not true for relational data



Approaches \ Properties	Structured data	Unstructured data	ACID	BASE	SQL	HiveQL	Cypher	NoSQL
RDBMS [Oussous et al. (2015)]	X		X		X			
MapReduce [Dean and Ghemawat (2008)]		X		X		X		X
Hadoop [8]		X		X		X		X
HadoopDB [Abouzeid et al. (2009)]	X	X	X	X	X			X
Hadoop ++ [Dittrich et al. (2010)]		X		X		X		X
CoHadoop [Eltabakh et al. (2011)]		X		X		X		X
GDB-Neo4J [Park et al. (2014)]	X		X				X	X
Clydesdale hadoop [Kaldewey et al. (2012)]	X	X		X		X		X

Fig. 3. Literature Evaluations Comparison

model due to its rigid schema property. In graph data model nodes are used to represent entities, edges are used to create relationship among nodes, while both relationship and nodes have properties in the form of key-value pairs. RDF (Resource Description Framework) is the semantic graph data store and represents information as subject-predicate-object and is uses on different systems for highly connected data. For example, Oracle database, Social Networks applications, medial, life sciences and intelligence communities. Graph databases are used in various types of applications such as Master Data Management, Graph-Based Search, IT and Network Operations, Real-Time Recommendation System and Social Data Analysis. Figure 3 shows the approaches to handle big data.

III. Research Methodology Process Flow

This section describes the design of our proposed methodology. It mainly consists of relational databases (ORACLE) with its default settings and NoSQL graph database (Neo4j). Secondly, we will perform the physical database tuning of Oracle database. With the physical database tuning we can improve the performance of Oracle database. Tablespaces are one of the physical database tuning techniques of Oracle Database. Thirdly, we will run a large data set on both layouts to conclude the results in a simple (Standalone System) and in a client/server environment where our performance measures are how much time a query can take to return its result. Secondly, how much the results are accurate. Figure 4 describes our desired research methodology or framework.

IV. Design of Experiment

A. Experimentation Framework

To evaluate the proposed research methodology of figure 2, we setup an experiment on a medical data set Medcare case study on Oracle 11g Enterprise Edition and Neo4j 3.03 Community Edition. We define parameters such as sub-schema, network speed and the number of records returned by a query. Experiment compares the effectiveness of both databases.

B. Schema

The Medcare database schema is a case study of hospital healthcare system. The Medcare schema includes of main table such as Patients, Patient visit, Dependent, Medical staff, Patient

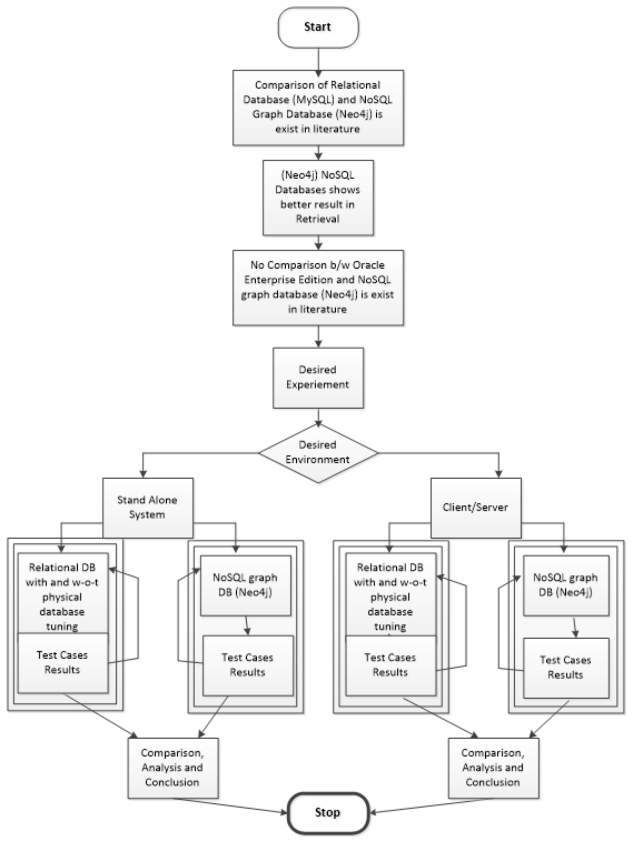

Start
Comparison of Relational Database (MySQL) and NoSQL Graph Database (Neo4j) is exist in literature
(Neo4j) NoSQL Databases shows better result in Retrieval
No Comparison b/w Oracle Enterprise Edition and NoSQL graph database (Neo4j) is exist in literature
Desired Experiement
Desired Environment
Stand Alone System
Client/Server
Relational DB with and w-o-t physical database tuning
NoSQL graph DB (Neo4j)
Test Cases Results
Relational DB with and w-o-t physical database tuning
NoSQL graph DB (Neo4j)
Test Cases Results
Comparison, Analysis and Conclusion
Comparison, Analysis and Conclusion
Stop

Fig. 4. Research Methodology

TABLE II. Medcare Schema Object Size

SN#	Object Name	Number of Records	File Size in MB
1	Patient	27952	04 MB
2	Dependent	19036	2 MB
3	Patient_Visit	625721	320 MB
4	Patient_IssueMed	869666	80 MB

IssueMed, Patient history, Patient Appointment and Patient History. MedCare schema has been updated yearly and all the updated records are saved into their respective tables. The following table describes the number of records present in each table and size of each object. The table II describes the objects' name, number of records in each object and their size in MB.

C. Queries Schema

The designed methodology is used to evaluate and compare objective benchmarks of ORACLE 11g Enterprise edition and Neo4j 3.0.3 Community edition. The objective benchmark consists of the following properties:

- Disk space requirements
- Set of predefined Queries
- Scalability characteristics

For the preliminary analysis, five various queries are performed on the objects of mentioned schema using both tools. The queries are given in the following figure 5. Figure 6 represents the characteristics and complexity level of each performed query.

Query#	Oracle 11g Enterprise Edition	Neo4j 3.0.3 Community Edition
1	Select count(*) From Patient_Visit p,Patient_Issuemed i where P.patient_visitno=I.patient_visitno and P.depend_sno=I.depend_sno and P.patient_id=I.patient_id;	MATCH (PATIENT_VISIT)-[r:has_med]->(PATIENT_ISSUEMED) RETURN COUNT(*)
2	Select count(*) From patient p, dependent d Where d.patient_id=p.patient_id;	MATCH (dep:DEPENDENT)<-[:has_dependent]-(pd:PATIENT_DATA) RETURN COUNT(*)
3	Select count(*) from Patient p, Dependent d, Patient_visit pv where d.patient_id=p.patient_id and pv.depend_sno=d.depend_sno and pv.patient_id=d.patient_id;	MATCH (visit:PATIENT_VISIT)<-[:VISITS_ARE]- (dep:DEPENDENT) OPTIONAL MATCH (dep)<-[:has_dependent]-(pd:PATIENT_DATA) Return count(*)
4	select count(*) from patient_visit pv where pv.depend_sno in(select d.depend_sno from dependent d where d.depend_sno=pv.depend_sno) and pv.patient_id in (select p.patient_id from patient p where p.patient_id=pv.patient_id)	MATCH (visit:PATIENT_VISIT)<-[:VISITS_ARE]- (dep:DEPENDENT) OPTIONAL MATCH (dep)<-[:has_dependent]-(pd:PATIENT_DATA) Return count(*)
5	select count(*) from patient_issuemed pi where pi.patient_id in (select p1.patient_id from patient_visit p1 where p1.patient_id=pi.patient_id) and pi.depend_sno in (select p2.depend_sno from patient_visit p2) and pi.patient_visitno in(select p3.patient_visitno from patient_visit p3	MATCH (PATIENT_VISIT)-[r:has_med]->(PATIENT_ISSUEMED) RETURN COUNT(*)

Fig. 5. Sample Tested Queries

Query #	Simple Query	Joined Query	Subquery & Correlated Query	No# of Tables/Subquery/Joins
1		X		2/-/3
2		X		2/-/1
3		X		3/-/3
4			X	3/2/2
5			X	3/3/1

Fig. 6. Represents the Characteristics and Complexity Level of Each Performed Query

- Query 1: count records from Patient Visit and from Patient IssueMed tables.
- Query 2: count records from Patient and from Dependent Tables.
- Query 3: is used to count records from three tables. The tables are Patient, Dependent and Patient Visit.
- Query 4: is same as query 3 but uses the concept of correlated subquery.
- Query 5: is same as query 1 but uses the concept of correlated subquery.

D. Default Configuration of Oracle 11g and Neo4j on Local System

The experiments have shown that Neo4j performs well with reasonable time in all queries for the Medcare data set. The time of each query in seconds is given in table III.

When the dataset size increases, the graph database performs much better than relational databases, [12]. Relational database does not store the relationship of data in the database whereas graph databases hold the relationship of information and also is used for connected data so it enhances the performance of graph database. The above five queries are performed when the system is in the normal state.

TABLE III. QUERIES EXECUTION TIME IN SECONDS

Query#	Oracle 11g	Neo4j 3.0.3
1	4.515 Sec	0.346 Sec
2	0.172 Sec	0.216 Sec
3	3.531 Sec	0.452 Sec
4	3.469 Sec	0.452 Sec
5	10.391 Sec	0.346 Sec

ORACLE 11g										
Query#	Local System					Server System				
	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th
1	0.953	0.391	0.422	0.469	9.359	6.093	0.359	5.328	4.907	15.094
2	0.422	0.047	0.094	0.094	9.265	5.828	0.125	4.750	4.672	14.844
3	0.407	0.031	0.109	0.078	9.172	6.375	0.094	4.782	4.844	15.219
4	0.407	0.016	0.109	0.079	9.238	5.781	0.375	5.938	4.391	14.703
5	0.406	0.047	0.094	0.094	9.219	5.781	0.062	4.765	5.047	18.187

Fig. 7. Results(Time in Seconds) of Queries on local and on server systems.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	1.000	0.960	1.000	0.980	0.000	0.042	0.920	LS
0.000	0.000	0.000	0.000	0.000	0.000	0.042	0.040	RS

Fig. 8. Results return by weka tool.

E. Configuration of Oracle 11g on Local System and on Server System with partitioning

While performing experiment, the Oracle 11g database of the local system and the server was in the consistent state before query executions. The results of the desired experiments are given below in figure 7 and each query is tested 5 times.

For query number 2 the performance of server system and local system is almost same. But for all other queries the performance of a local system is better than server system. The network speed at the time of experiment was 0.557 mb/s at peak.

We also processed the results of figure 7 in Weka tool by using J48 algorithm. J48 [18] has the following advantages:

- J48 algorithm is suitable where dataset is not changed rapidly.
- Represents decisions about data in alternatives possibly rules and tree.
- Can easily modify a decision tree as new information available.
- The decision trees are self-explanatory

The above figure 8 shows the results of the dataset of table 4.5 return by weka tool. J48 (C 0.25 -M 2) classifier is used for the classification with 10-fold cross validation. In the figure 8, the class local system (LS) always performed better than remote system (RS). The classifier J48 classifier returns 96% accuracy of the table 4.5 dataset.

V. Predictive Analysis of Comparison

As the dataset size increases the graph database performs much better than relational databases, [12]. A relational (oracle) database follows rigid schema structure and is difficult to manage the changes when there are more than 20 tables due to constraints. For efficient data retrieval relational database (Oracle) uses indexing. In relational (Oracle) database, whenever any schema (user) and its objects (Tables, Views, and

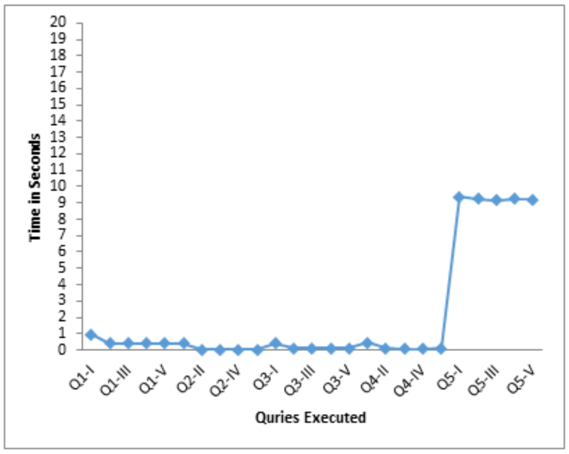


Fig. 9. Queries Execution Time on Local System in seconds.

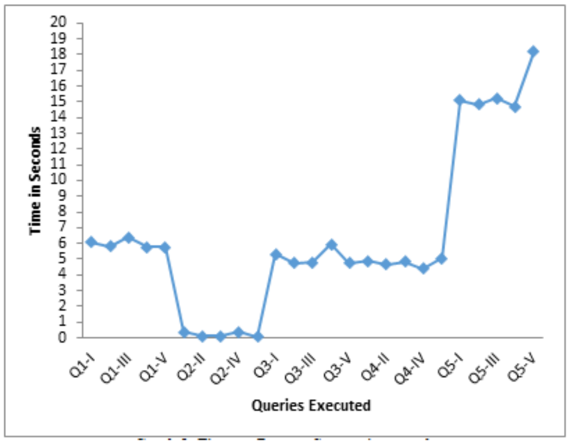


Fig. 10. Queries Execution Time on Remote System in seconds.

Procedures) are created, by default database objects are stored in a User tablespace (Logical folder). Therefore, we created Medcare schema by using oracle default configuration and also loaded data of the same schema in Neo4j. Neo4j performed better than Oracle due to the following reasons:

- Oracle database does not perform well when the data is more connected.
- Oracle database is heavily dependent on constraints and indexing.
- Neo4j uses graph algorithm (Dijkstra and Floyd Warshall) to find the shortest paths and save them. Therefore it always takes constant time.
- Neo4j also stores the relationship while Oracle database does not.

While performing the queries of figure 5 on a local and on a server system with separate tablespaces for large tables such as patient_visit and patient_issuemed. Whenever the query is executed for the first time, the Oracle database reads data from the disk by finding the appropriate segment (Table etc.). After finding the segment, Oracle finds the appropriate extent (Where data is located). At the end, oracle selects the particular block (portion of data) from the selected extent and stores it in the Database buffer in memory. For the next time Oracle database reads data from the buffer. Local system (LS) performed well than remote system (RS) except for the query 2 when executed the first time as shown in figure 7.

The Figures 9 and 10 graphically represent the execution time of each query on local system and on remote system respectively. Queries are plotted on x-axis and the time consumed by each query is represented on y-axis. In both graphs Query 5 takes considerably more time than other queries because in query 5 we have fetched data from three large tables and this query is using three sub-queries and a join condition too.

VI. Conclusion and Future Work

Relational databases are being designed to managed structured data. Now-a-days, many organizations are heavily dependent on unstructured data and generate enormous amount of data such as Facebook, Google, Yahoo, Google+ and Amazon etc. To handle such large data, Hadoop and NoSQL databases are used. Our experiment has shown whenever data becomes more and more connected (large number of joins) and large in size, relational databases show worse performance than NoSQL graph database. Relational databases (Oracle 11g Enterprise Edition) used constraints, indexes and do not store any relationship information. While NoSQL graph database stores relationship information among various nodes. Graph database (Neo4j 3.0.3) use native graph storage. Native graph is optimized and designed for storing and managing graph. NoSQL graph database uses index-free adjacency. The connected nodes physically point to each other in the graph database due to index-free adjacency characteristics. Our experiment describes NoSQL database performance are significantly better than Oracle 11g with and without partitioning. The results returned by the Weka tool also present that Oracle 11g on Local system with partitioning is performed better than Oracle 11g with partitioning on Server system.

In future we will try to compare relational database (Oracle 11g) and graph database (Neo4j) for a remote system. It is not 100% solution for enhancing performance in row based database management systems but performance will be checked by applying partitioning under umbrella of physical database paradigm. There is very less amount of work done in this area thats why it is considered as future work. Our proposed techniques also need some improvement in terms of high performance that we will deal all the issues related to elapsed time in the future works and how to build schema of structured and unstructured data.

References

[1] Rifaie, M., R. Alhajj, and M. Ridley (2009). Data governance strategy: a key issue in building enterprise data warehouse. In Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, pp. 587-591. ACM.

[2] Dittrich, J., J.-A. Quiane-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad (2010).Hadoop++: making a yellow elephant run like a cheetah (without it even noticing). Proceedings of the VLDB Endowment 3(1-2), 515-529.

[3] Abouzeid, A., K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz, and A. Rasin (2009). Hadoopdb: an architectural hybrid of mapreduce and

dbms technologies for analytical workloads. Proceedings of the VLDB Endowment 2(1), 922-933.

[4] Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B., & Vasilakos, A. V. (2016). Big data: From beginning to future. International Journal of Information Management, 36(6), 1231-1247.

[5] Kaldewey, T., E. J. Shekita, and S. Tata (2012). Clydesdale: structured data processing on mapreduce. In Proceedings of the 15th international conference on extending database technology, pp. 15-25. ACM.

[6] Cuzzocrea, A., I.-Y. Song, and K. C. Davis (2011). Analytics over large-scale multidimensionaldata: the big data revolution! In Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP, pp. 101-104. ACM.

[7] Dean, J. and S. Ghemawat (2008). Mapreduce: simplied data processing on large clusters. Communications of the ACM 51(1), 107-113.

[8] Eltabakh, M. Y., Y. Tian, F.Ozcan, R. Gemulla, A. Krettek, and J. McPherson (2011). Cohadoop: exible data placement and its exploitation in hadoop. Proceedings of the VLDB Endowment 4(9), 575-585.

[9] Oussous, A., F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih (2015). Comparison and classication of nosql databases for big data. In Proceedings of International Conference on Big Data, Cloud and Applications.

[10] Strauch, C., U.-L. S. Sites, and W. Kriha (2011). Nosql databases. Lecture Notes, Stuttgart Media University.

[11] Zafar, R., M. F. Zuhairi, E. Ya, and H. Dao. Big data: The nosql and rdbms review.

[12] Vicknair, C., M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins (2010). A comparison of a graph database and a relational database: a data provenance perspective. In Proceedings of the 48th annual Southeast regional conference, pp. 42. ACM.

[13] Park, Y., M. Shankar, B.-H. Park, and J. Ghosh (2014). Graph databases for large-scale healthcare systems: A framework for efcient data management and data services. In Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on, pp. 1219. IEEE.

[14] Ali, N. M. and T. Padma (2016). Graph database: A contemporary storage mechanism for connected data. system 5(3).

[15] Fang, H., Z. Zhang, C. J. Wang, M. Daneshmand, C. Wang, and H. Wang (2015). A survey of big data research. IEEE network 29(5), 6.

[16] Apache Hadoop. http://wiki.apache.org/hadoop

[17] Khan, N., et al. (2014). Big data: Survey, technologies, opportunities, and challenges. The Scientic World Journal, 2014, 18.

[18] Cristina Petri, Cluj Napoca, (2010). Decision Trees.

[19] Robinson, I., Webber, J., & Eifrem, E. (2015). Graph databases: new opportunities for connected data. " O'Reilly Media, Inc.".

[20] Saba Luqman and Ejaz Ahmed(2016). Systematic Mapping: Database Tuning Progress in a Decade, Journal of Basic and Applied Scientific Research (JBASR), ISSN: 2090-24x, Indexed in Copernicus, Vol. 6(11), pp. 15-25.