COMPARATIVE ANALYSIS OF MONGODB, ORACLE NOSQL AND ORACLE 12C

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Software Engineering

Submitted By

Prabhjot Kaur

(Roll No. 801331019)

Under the supervision of:

Dr. Seema Bawa

Professor

Thapar University, Patiala

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Comparative Analysis of Mongodb, Oracle Nosql and Oracle 12c*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Seema Bawa* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Prabhjot Kaur

Signature:

(Prabhjot Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Seema Bawa 15/7/15

(Seema Bawa)

Professor,

CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

Abstract

Computer memory is an important functionality. In the present scenario, most of the computers have the necessary hardware for processing the information and store it in a safer mode to be accessed in future. For these operations, database management systems work in collaboration with the application programming interfaces at the low level. For different types of problems, different DBMS is used. In this thesis the aim is to provide an overview of the various DBMS. Oracle 12c, Oracle NoSQL, MongoDB has been discussed in several sections.

NoSQL is providing different type of databases to cater the requirements for data available in current digital world. There are various factors such as velocity, volume and variety which are increasing the complexity of data. To address the demand created by such factors, there are different type of databases are coming into market, which are commonly referred as NoSQL databases and amongst them two are being discussed.

Acknowledgement

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. It is a great privilege to express my gratitude and admiration towards my respected supervisor **Dr. Seema Bawa** Professor Computer Science & Engineering Department. She has been an esteemed guide and great support behind achieving this task. This work would not have been possible without the encouragement and able guidance of her. I also thank my supervisor for her time, patience, discussions and valuable comments. Her enthusiasm and optimism made this experience both rewarding and enjoyable. I am truly grateful to her for extending her total co-operation and understanding whenever I needed help and guidance from her. I am also heartily thankful to **Dr. Deepak Garg**, Associate Professor and Head, Computer Science & Engineering Department and **Dr. Damandeep Kaur**, PG coordinator, for motivation and providing uncanny guidance and support throughout the preparation of the thesis report.

I will be failing in my duty if I do not express my gratitude to **Dr. S. S. Bhatia**, Senior Professor and Dean of Academic Affairs, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable. Last but not least, I would like to thank my family for their wonderful love and encouragement, without their blessings none of this would have been possible.

Prabhjot Kaur

(**801331019)**

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Overview

NoSQL is providing different type of databases to cater the requirements for data available in current digital world. There are various factors such as velocity, volume and variety which are increasing the complexity of data. To address the demand created by such factors, there are different type of databases are coming into market, which are commonly referred as NoSQL databases. In the conventional database systems, i.e. Relational1 databases, the challenges posed by scalability and agility of data were not being handled which are commonly faced in the modern applications, moreover the relational databases doesn't provide cheap storage and processing power.

NoSQL are not only scalable and high performance systems but also, the data model caters the needs of modern applications which were not in scope of relational databases.

In NoSQL, there is no need to define schemas prior to the storage of data, which was essential in relational databases [1].

MongoDB is an implementation of a key-value store that supports the single value abstraction JSON. It uses API unlike SQL in relational databases but it has a query capability which tends to the functionality used in relational databases. MongoDB brings flexibility in schemas whereas in SQL one must need to define and declare a schema before actually storing the data in tables. Collections used in MongoDB don't enforce the structure for a document. The nature of flexibility provided in it helps in mapping of documents to an object. Even a significant variation in the data can be handled by the document through matching the data fields of the represented object. Further normalizing the data would divide it into multiple collections, and need numerous write operations not granular in nature.

Oracle NoSQL Database in many aspects implements Berkeley DB Java Edition as the basis for its storage and replication technology like NoSQL Database implementation. It is a Java based key-values store implementation which supports a value abstraction layer currently implementing Binary and JSON types. Its key structure is designed in a manner which leverages the large scale distribution and storage locality using range based

traversing. It supports built in cluster load balancing and transactional connotations from ACID to relaxed consistency. Oracle NoSQL is also integrated with open source technologies like Hadoop/Map Reduce.

Oracle Database 12c introduces a new multitenant architecture that makes it easy to consolidate many databases quickly and manage them as a cloud service [6]. Oracle Database 12c is high performance systems as it includes in-memory data processing capabilities. Over successive innovations, it has set new benchmarks in factors such as efficiency, performance, security, and availability. Designed for the cloud, it operates with lowered IT costs making maximum utilization of resources.

1.2 Mongo DB

1.2.1 Overview and Introduction

MongoDB is an open-source document database. It provides high performance, high availability. A record in MongoDB is a document, which is a data structure composed of field and value pairs [7].

Key Features

- *Performance* For simple queries, it provides good performance, as all the related data are in single document which eliminates the joined operations. MongoDB provides high performance data persistence [3].
- *Availability* To provide high availability. MongoDB's replication facility, called replica sets as well as data redundancy.
- *Scalability* It is horizontally scalable i.e. workload can be reduced by increasing the number of servers.

1.2.2 MongoDB Architecture

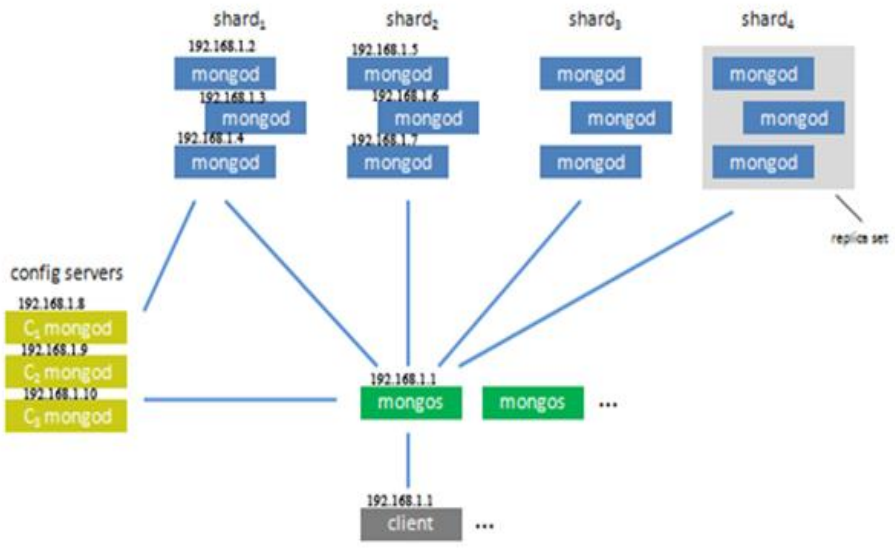

Figure 1-1 : MongoDB architecture [7].

Yellow boxes are VERY important. It contains all the configuration of the MongoDB deployment. If a config server falls down, you cannot add new shards to the infrastructure, but you can read and write data in the databases [7].

Green boxes are the schedulers. You can have all you want and it will be the client's access point. The client can be in the same machine or in other.

Blue boxes are the machines where the DB data is going to be stored.

1.2.3 MongoDB Data Model

- Data as Documents

  Data is stored in documents as data BSON objects. Key advantage of using BSON is efficiency because of its binary format. Documents are contained in 'collections' [14]. Collections can be considered equal to relational database tables. MongoDB

documents have entire data of a record in one document whereas in case of relational database data is spread on different tables.

- MongoDB Dynamic Schema

  Documents in MongoDB vary in structure. Each document is identified by unique ID which can be manually given by the user while creating the document or are automatically provided by the database. Fields vary from document to document [2].

1.2.4 MongoDB Query Model

- Drivers.

  Queries are expressed as JSON objects and are sent to MongoDB by database drivers. Drivers provided by MongoDB include Java, .NET, Ruby, PHP, JavaScript, node.js, Python and others.

- Indexing

  MongoDB supports many secondary indexes that are declared on any filed in document. Indexes types supported by MongoDB are:

- Unique indexes
- Compound indexes
- Array indexes
- Text search indexes
- TTL(Time to Live) indexes
- Geospatial indexes

Mongo shell and Query types

- Mongo Shell: It is a popular way to interact with MongoDB for Different operations for storing and retrieving data. In order to run queries Mongo shell needs to be running as without it no operations can be performed.

- Query Types: Applications using complex queries and secondary indexes can be build. MongoDB support many types of queries which is key factor of its flexibility. Queries supported are:

- Key-value queries
- Range queries
- Geospatial queries
- Text search
- Aggregation Framework

1.2.5 MongoDB Data Management

- Auto-sharing for linear scalability
- Pluggable storage architecture for application flexibility
- Storage efficiency with compressio

1.3 Oracle NoSQL

1.3.1 Overview and Introduction

Oracle NoSQL Database provides multi-terabyte distributed key/value pair storage that offers scalable throughput and performance. That is, it services network requests to store and retrieve data which is accessed as tables of information or, optionally, as key-value pairs [5]. Oracle NoSQL Database services these types of data requests with a latency, throughput, and data consistency that is predictable based on how the store is configured.

Oracle NoSQL Database offers full Create, Read, Update and Delete (CRUD) operations with adjustable durability guarantees. Oracle NoSQL Database provides performance scalability [4].

1.3.2 Oracle NoSQL Architecture

The following illustration depicts the typical architecture used by an application that makes use of Oracle NoSQL Database [9]:

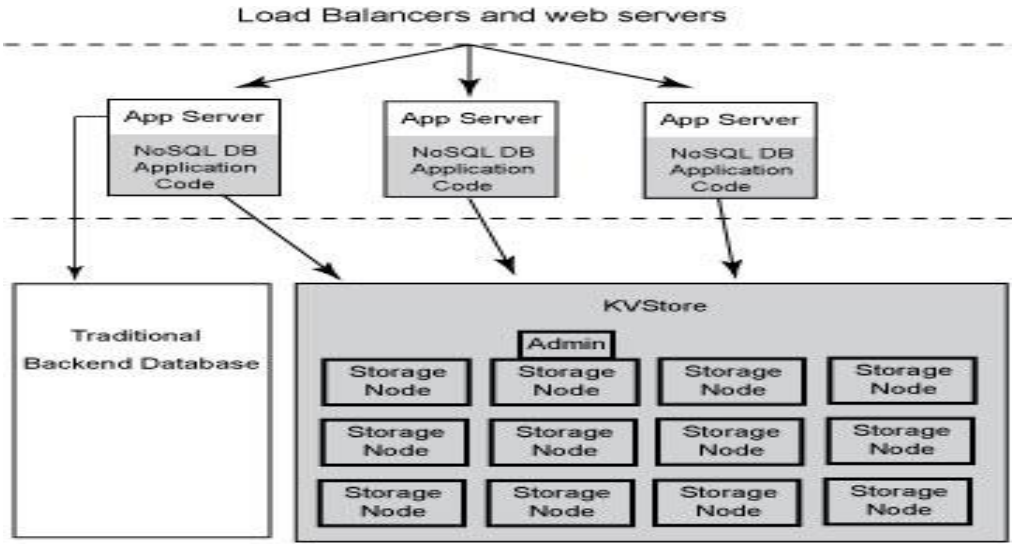

Figure 1-2 : Oracle NoSQL Architecture [9].

- *Replication Nodes and Shards*

  The master node copies all writes to the replicas. These replicas are used to service read-only operations [9].

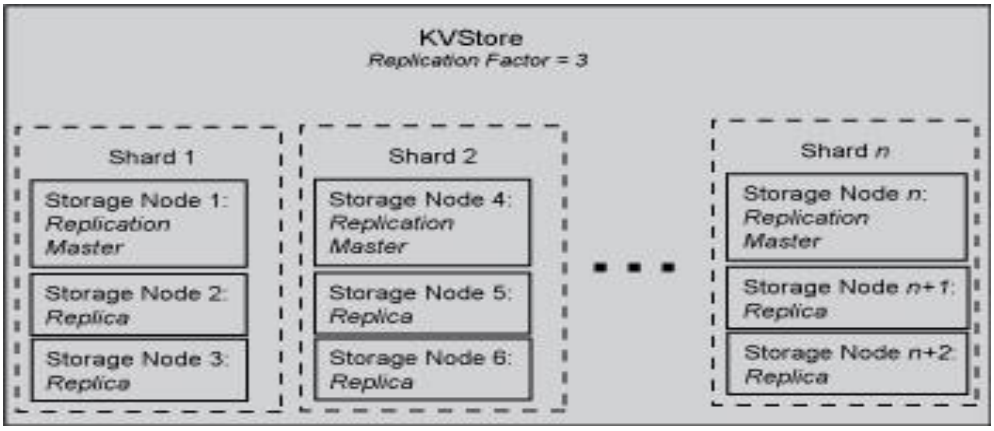

Figure 1-3 : How the KVStore is divided into shards [9].

- *Partitions*
- *Zones*
- *Topologies*

1.3.3 Data Models

You can model your data in Oracle NoSQL Database by using Tables, JSON schemas or a raw key-value interface. Tables are the easiest way to model data. They provide the highest level of abstraction; they are simple to model and should be familiar to any developer. This model also supports secondary indices and table evolution [9].

You can use JSON to model data for your JSON centric applications. If secondary indices or strongly typed keys are not a priority and, if keys are going to be modeled manually, then this is a good choice.

Finally, if you want to serialize the data, manage the key structure, manage secondary indices through index views, manage evolution and security through your client code,

1.3.4 Administration

The Administration command line interface (CLI) is the primary tool used to manage your store. The command line interface is accessed using the following command:

java -Xmx256m -Xms256m -jar KVHOME/lib/kvstore.jar runadmin

KVLite

It can be configured, and run using CLI. It runs a single process with no administrative interface. It has a single node which can't be replicated.

1.3.5 Monitoring

Oracle NoSQL Database has admin console developed using HTML. Using browser, the administrator can access the machine where process is running.

It offers following functional controls:

- Topology: To monitor all the nodes which are installed on the store.
- Plan & History. To view plans executed recently on the system.
- Logs. To view the contents of the store's log files. One can download the contents of the log files from this screen.

1.3.6 Access and Security

KVStore can be access in the following ways:

- Using Java APIs that the application developer to interact with the Oracle NoSQL Database Driver.
- Using a command line interface or a browser-based GUI.

1.4 Oracle 12c

1.4.1 Overview and Introduction

Oracle Database 12c provide real time analytics on enterprise data using cloud, it provides such experience using following features [9]:

- Databases consolidation and enabling Database as a Service with Oracle Multitenant
- Optimization through in-memory operations.
- Compression of data as per use, and automated process for data storage optimization.
- High availability using oracle's Maximum availability Architecture.
- More security to enterprise data.
- Enterprise Manager Cloud Control for more efficiency.

1.4.2 Oracle Database Architecture

Oracle database server comprises of the following:

- Oracle Database
- Database Instance (at least one instance on a server).

Oracle Database

Set of files, that is located on disk and store data. This file doesn't have any dependency with a database instance.

Database instance

Internal memory structure SGA also known as System or Shared Global Area is shared pools which requests and responses to the queues (shares server) [9].

Program or Process Global Area (Internal memory structure PGA) are used for individual process and are shared with other existing processes) [12].

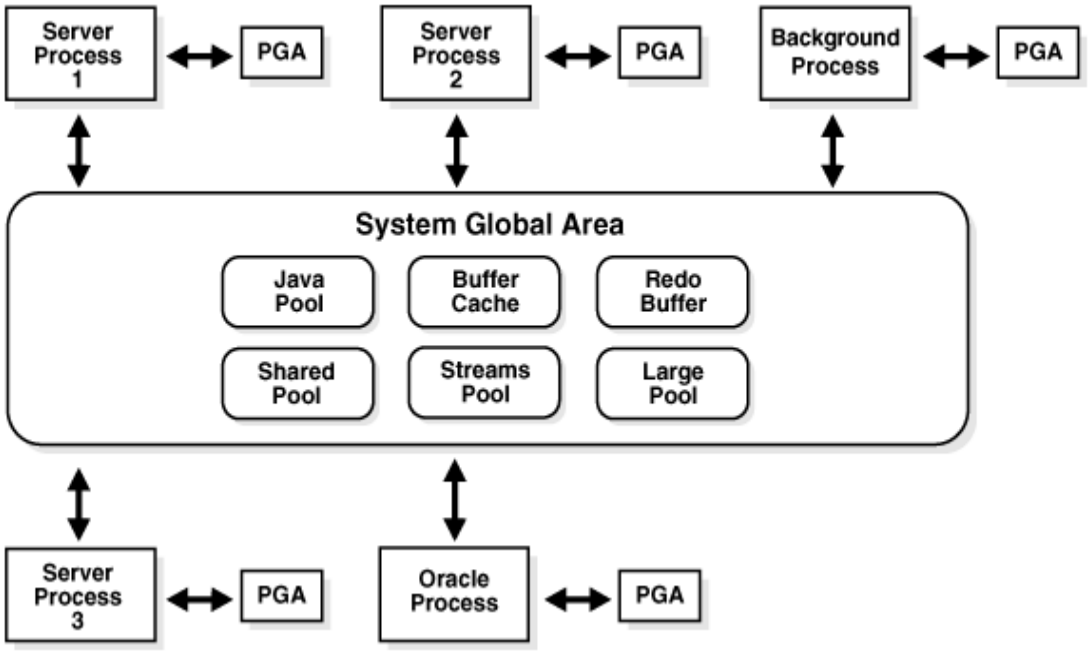

Figure 1-4 : Oracle Instance and Oracle Database [9].

1.4.3 New Multitenant Architecture

Oracle Database 12c comes up with a new concept which support the usage of multiple databases inside a single database, commonly referred “sub databases” and “super database”.

Terminology denotes the “super database” as CDB (container database); and the “sub database” as PDB. None of the database support CDB prior to Oracle Database.

CDB comprises of the following:

- One root
- One seed PDB
- Zero or more user-created PDBs.

Below shown in the graphical representation is a CDB with its four containers: the root, seed, and two PDBs [10].

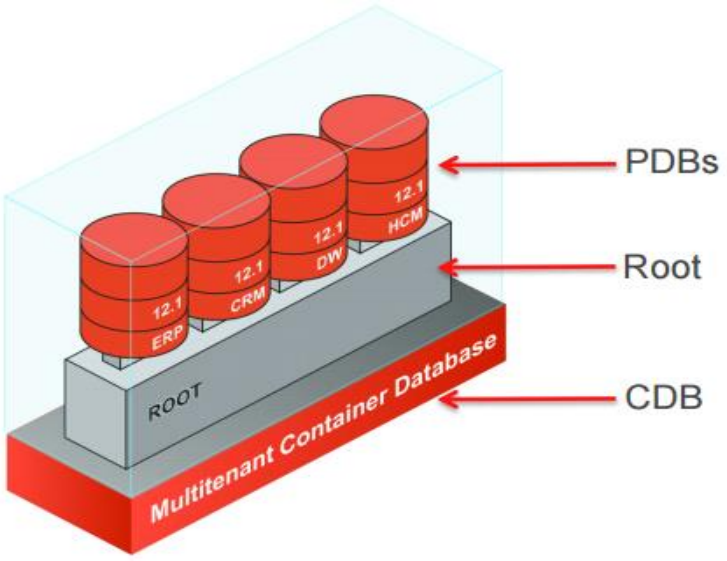

Figure 1-5: Oracle 12c Multitenant Architecture [13].

1.4.4 Memory Management

Architecture

On initialization of an instance, memory area is allocated by oracle database and background operations/process is started. Following information resides in allocated memory area:

- Program code
- Session information.
- SQL Query status information which is required during program execution.
- Lock data information which is shared between different concurrent processes.
- Cached data information, which also exists on disk [9].

Structures

The basic memory structures associated with Oracle Database include:

- System global area (SGA)

Shared memory structures grouped as known as SGQ, it contains data and control information an instance. It is shared by the background processes running on the server, e.g. cache data blocks and shared SQL [11].

- Program global area (PGA)

  PGA is a non-shared which is exclusively used by a given oracle process, it is a memory which holds data and control information. When a process starts, the oracle database creates a PGA; scope of each PGA is limited to a specific process. When the multiple individual PGAs are collected, this collection becomes a PGA total instance. The initialization parameter sets the size of an instance PGA.

- Software code areas

  It is the memory used for the software coding purpose. It is located at a different location from enterprise related programs in a more protected location.

- User global area (UGA)

  Memory allocated to a user session is known as UGA/

Oracle Database Memory Management

Oracle Database Memory Management governs the size of oracle instance memory structure with developments on the enterprise database. Memory management is performed in the following three ways:

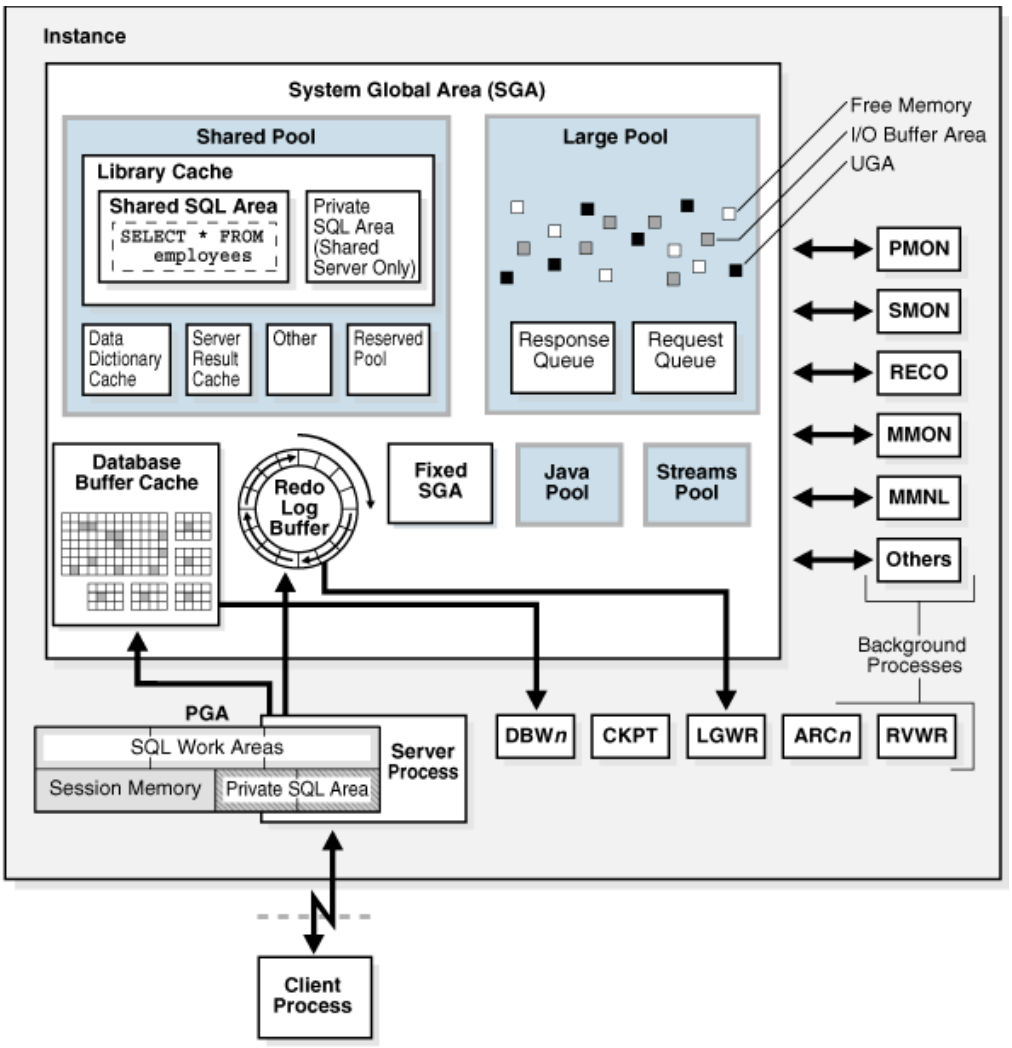

Figure 1-6 : Oracle Processes and the SGA [11].

1.4.5 Significance

- The concept of Multitenant architecture helps in consolidation on cloud without making changes to an existing application.
- It comes with more security as compared to older versions. It includes advanced encryption in security system.
- Adaptability: Applications run unchanged
- Oracle Enterprise Manager Cloud Control 12c integration helps in easy administration of the database lifecycle.

Chapter 2 Literature Survey

In the following chapter, analysis is performed on various databases which are MongoDB, Oracle NoSQL and Oracle12c. These studies are done from multiple such as performance and scalability. Below are abstracts of those analyses performed by various research fellows.

2.1 Evolution of NoSQL Databases

DataStax Corporation has conducted various experiments in order to guide lining top NoSQL databases. Different reasons for deploying NoSQL databases are discussed in this paper. Different range of workloads are performed on Amazon Web Services(AWS) with the help of End Point Corporation which is an open source consulting company have been carried out so as to achieve a canon for top NoSQL databases including Apache Cassandra , Apache HBase along with MongoDB. Documentation is presented on the basis of the outcomes provided by the company about performance of each database. Conclusion presented by this paper is that lot of work needs to be performed by the IT professionals so as to guarantee that they are working on the right database as per their application requirements [19].

Nyati et al. [20] in their paper have performed several performance analysis of NoSQL unstructured databases on premises of various standards including database querying, time required for inserting the data into database, their performance time, different iterations are carried out to examine them with different entries. On this basis they have carefully explained several NoSQL databases. Different NoSQL databases were explored. On examining all of the databases discussed, Nyati has chosen MongoDB as foremost choice for their analysis purpose. Reason of choosing MongoDB given is that it has the capability to handle small data along with large data capacity. MongoDB and MySQL databases were compared on different standards which concluded that MongoDB is way faster than MySQL in inserting and searching data [21].

2.2 MongoDB

Govindaraju et al. [16] narrates the various entertainments of MongoDB and Hadoop. In order to recognize the correct background or situation for scientific data analysis, this

paper has presented various methods for analyzing the performance, scalability as well as fault-tolerance operating MongoDB with Hadoop.

Matei et al. [17] yields a standard on entertaining the production of MongoDB with respect to the cloud environment. Database used in this paper which is MongoDB is preceded into the cloud so as to analyze and demonstrate the outcomes in terms of its production of MongoDB. In the next sections of this paper various likelihoods of low cost virtual machines are explored. Different choices of cloud services to opt from are provided and performance being tested for everyone.

Arora et al. [18] describes in this paper with the help of Class diagram and JSON format that using different data models provided by MongoDB along with support to various queries types, Modeling and Querying went henceforth with these supports provided in MongoDB database. Different types of queries supported along with how to process them in MongoDB is described in this paper along with various constructs including no JOIN support.

As examined and analyzed from various papers discussed above it can be concluded that though introductory work is done about using queries in databases etc but still there is no previous work that has been done on implementing NoSQL-MongoDB for developing a blog web application. Because of the expanding usage of data being processed in daily routine including academic along with commercial sector, usage of NoSQL databases is enlarging which has led me to conclude that lot of more work is required to be performed on implementation of NoSQL-MongoDB so as to use them as back end storage for various web based applications [16].

Brust A. [22]: A decision for Oracle termed Janan Dash and NoQSL which is a RDBMS phenomenon of IBM is been discussed by author in the paper. For usersround the sphere, with swamped of the terms like NewSQL, Database Appliance, NoQSL, and Big Data etc. offers numerous problems for RDBMS. As now is the period for energetic schematic and therefore to meet the altering necessities of new data model, if it is not met on hourly basis, then the updating of the records must be replicated on everyday basis so that the desired requirements are met. Further agreeing to the author, swapping into NoSQL technologies will be an intellectual idea, if some organization deals with such types of

anxieties. Some concerns like scale-in (horizontal scaling) or scale out (vertical scaling) can be commenced by precisely advanced and intended NoSQL Technologies. MongoDB API, query semantic and wire protocol has been executed by IBM, thus creating a way for numerous new applications that could be linked with enterprise DB's such as the grid Web Sphere extreme Scale data grid and IBM's DB2 relational DB. On case-by case basis NoSQL offers various solutions of relaxed consistency which are very essential. For NoSQL databases in forthcoming the enactment and various features will endure to development and advancement over time.

Kaur et al. [23] modeling data and the methods to enquiry them in NoSQL DB's were described in the existing work. They also momentarily expounded the several classes of NoSQL DB's. Two classes which are a 'document DB- MongoDB' and a 'graph DB-Neo4j' were reserved under deliberation out of the four classes of NoSQL databases that were conversed. The alteration was considered between the aptitude to work with capacity and diversity of the data and what are the distinctions from one another amongst the syntaxes and query semantics. To operate with the mounting data storage needs MongoDB, document database was explicitly premeditated and developed but it has no endowment for joins. It is essentially used to regain, stock and handle the semi-structured data, this data was chiefly kept in the document arrangement. On the other hand, the intact database as a network assembly teeming with connotation between nodes was demonstrated by graph databases. Nodes were used to accumulate the objects and the edge involving nodes used to function as articles and interactions, correspondent to relational database architecture. MongoDB has its individual query language. Neo4j graph database use to query with Cypher which is a declarative graph query language. Graph database is best in denoting the categorized data and also, ACID complaint has the competency of stowing semi-structured evidence. Furthermore with the help of instances query formats are profoundly explained.

Aggarwal et al. [24] in their paper they have explicated the data modeling and query implementation in MongoDB NoSQL database and further they also proofs it with the help of the class diagrams. They have also particularized about the additional feature i.e. no JOIN support. JSON format and precise class diagrams were taken in anxiety for proving modeling of representation of the database. When the correlated data is stocked

up into a single document that are in JSON-style format made of key and value duos it is known as embedding of a document and it is also identified as the technique to store the data in MongoDB in the deformalized form. Enlightenment of key points or rubrics for creating documents and the assortments were conferred in the further part of the study. Query format was expanded explicitly by providing comparative queries in MySQL by means of little sample collections so that there is a better sympathetic and trepidation of the query formats.

The study that was offered next was based on the judgment of two types of databases i.e. NoSQL document oriented database management system and relational DB [25]. The authors Laura I.A., Florin R., Alexandru B. distinguish the MongoDB and Oracle database. Numerous restrictions were taken into account out of all those few were circulation and system requirements, constructions, constraints, query and addition times, hypothetical modifications, veracity and architecture.

Alexandru et al. [25] as per the authors there are four aspects on which MongoDB stresses which include power, speed, elasticity and ease of use. With the help of different drivers provided to MongoDB it further helps yielding maintenance for several programming languages. MongoDB also provides structures as indexing as well as simulated servers. Oracle is relational database model working on MySQL whereas in case of MongoDB there is no need to define the schema prior i.e. it is schema-less database. Huge amount of data can be fetched using MongoDB which may include data up to the range of 16MB whereas Oracle database can handle magnitude of 4KB only. Major drawback of Oracle database is duplication which is removed in case of MongoDB. Conclusion presented by this paper is that in order to have fast database along with high elasticity, one must opt for MongoDB. Also, if the case is such that it has no concern for swiftness and relations amongst tables is required than relational database i.e. Oracle Database can be chosen.

Coe, B [26] in this artifact a simple conception of MongoDB is provided. As per author, two important features of MongoDB are discussed which includes its schema less nature and deprived of joins. The two factors have marked a change between MongoDB and other SQL databases. The role played by tables and rows in relational databases are

played by collections and documents in case of MongoDB. According to the author in MongoDB collections two documents may contain different fields also it may have similar fields having different data types. Another important feature discussed in this artifact was its sharding. In case of SQL tables where columns are provided indexing, where as in case of MongoDB it allows to provide indexing to fields in documents. Sharding technique was debated by the author. At last it was concluded that NoSQL databases provides faster outcomes as compared to relational databases.

Arora R [27] in their suggested proposed an algorithm which can be used for converting relational data into non relational data which in this case is MongoDB's document based structure. As per author with the development of various NoSQL technologies many consumers along with businesses want to switch to NoSQL databases. Because of such emerging needs there is great need for switching data from relational databases to non relational databases. The course used for changing the data was developed with the help of NetBeans and Pentaho. First step conducted was establishment of link with MySQL server and then the database was to be chosen from the list provided whose data has to be moved into the MongoDB's document model. This process of moving data results in formation of embedded documents. Pentaho DI tool was being used which on receiving test files would generate the output of MongoDB collection. From all the studies carried out by the author it was recommended to switch to the NoSQL solutions that are available in the market.

Banker K [28], have discussed various aspects of MongoDB which may include its background and different situations about how MongoDB was organized and making it different from other NoSQL databases. As huge amount of data is being processed in daily routine therefore scalability has become a basic necessity which is delivered by MongoDB as it has the feature associated with it of secondary indexes and dynamic queries. Other features discussed in this paper were atomic updates, composite aggregations and sharding facility provided by MongoDB. In the other chapters simple CURD operations and queries are discussed.

Chodorow K [29] in her book has elucidated about sharding and its ideology. According to the author dataset is formed by collections which include one or many servers may

also be referred as shards. In this book three types of progressions are being discussed. First progression is about how different processes of mognos works in order to direct the requests regarding the data. Second progression discusses about the shards for storing the data. And finally the third progression elucidates about the config servers understanding the cluster. To spread the data over different shards the author has proposed setting of cluster. Next chapters include about MongoDB clusters.

Liu et al. [30] in their paper they have discussed about different methods of auto sharding in MongoDB along with several principles. And based on different aspects they have proposed an algorithm to express the distribution of data. The scheme discussed by them is to distribute the data amongst several shards. Also the scheme provides the process which depicts the growth of clusters. The imbalance between the algorithm proposed and the technique used for auto sharding is performed using read and write operations used for accessing clusters. A test was conducted in order to calculate its coinciding performance. Huge amount of data was inserted during the test and the better results were presented with the help of graphs. Productivity of the results were verified using directing experiments.

Zugic G [31] In this artifact, horizontal and vertical scaling were discussed. Different sharding keys used to communicate were evaluated and then the process for selecting the one amongst them. A proper explanation is provided so as to choose an appropriate sharding key. According to this artifact range shard key is brought into picture when application requires lesser write ascendable whereas query separation is critical. On the other hand hashed shard key works when application requires huge write ascendable but query separation is pivotal.

Huang et al. [32] in their paper they provided an outcome for virtualized auto-scaling that could be used in order to meet the requirements of SLA. The solution that was delivered was obtained studying the work of performed on cloud computing individualities along with sharding techniques used in MongoDB. Auto scaling process was used in the initial stage and this particular process belonged to route server. According to the outcomes obtained the result was such that the time taken by auto-scaling DB resolution to provide reaction was 4.3 seconds where as it was 7.1 for non-

scaling. In the next half, a shard data transfer algorithm was proposed which helped in lessening the effect which was caused in moving data from one machine to new virtual machine. The outcome provided by the algorithm was used by auto-scaling DB so as to set the number of VM that are required to be used in addition. Also the algorithm helped in determining in what data is to be transferred to VM's that are added later on.

Wang et al. [33] in their paper have highlighted the importance of MongoDB. Reason provided for highlighting this is the ability of MongoDB in providing proficiency in automatic load balancing. In automatic load balancing, the load is spread over various secondary nodes if the load over primary node is concentrated. Where as in case of auto sharding the work performed was of decreasing the load on one node and this is done with the help of distributing the data over chunks and in further process that data was moved to other nodes. After studying about the structure of the implementation process used by MongoDB in processing automatic load balancing, the author has enlightened about such technique which uses heat diffusion produced by clusters such that it is cheaper. This technique is referred as dynamic load balancing. Using dynamic load balancing outputs observed were improved. From the studies conducted by the author and from the outputs obtained, the conclusion that was presented was to make use of dynamic load balancing process as the outcomes of load balancing technique based on amount doesn't produced adequate results.

2.3 Oracle NoSQL

Re Lai [34] in this paper, the author Oracle NoSQL Database is used to discover the application development. Additional micro blog sample application is being presented here i.e. Kvitter which is fairly comparable to well-known micro blog Twitter. In order to cultivate the NoSQL samples Twitter-clone is currently a preferred theme which is frequently used by the developers. Two precise goals were kept in concern while building the sample application which are: The first one was to platform the oracle NoSQL Database and the second important goal was to use the accustomed perceptions like Java Contextual and Dependency Injection (CDI), Java enterprise design outlines and Java Server Faces (JSF) that were used by most of the Java Enterprise designers while building the application.

Oracle NOSQL database white paper [35]: Significant enactment and initiative quality storage was provided to the extensively circulated and exceedingly presented NoSQL environment by Oracle's NoSQL database. Exceptional performance, consistency as well as sturdiness is precisely delivered by the write-optimized storage structure and it is also recognized commercially by its realistic results. In addition to its performance, it also embraces "No Single Point of Failure" design which guarantees that data always endure accessible even after any number of catastrophes while processing and the system also remains modest and continues to run in every case.

All over the coming years, a per the author Red Monk's O'Grady, the features that can be enhanced by using NoSQL databases will be focused which includes upgrading the compatibility of various applications with the databases and organizing the tools. NoSQL databases because of their capability to handle large data and also handling unstructured data will be in high demand and use as stated by the author Dave Rosenberg. O'Grady also said that the NoSQL databases will be embraced for certain projects because of their capacity to manage large chunks of data or because of its scalability. In addition he also stated that, NoSQL databases will provide better storage management and help specific applications but they will not substitute the relational databases. Basho's Sheehy further stated that though the tools used by relational databases are suitable but there are various other tools present which have significant value in certain areas [41].

As per the requirements there will be many such applications that will make use of both the databases i.e. relational databases as well as NoSQL databases. So the process of choosing the correct database with respect of their application holds great importance depending upon what essential feature is required for e.g. its scalability or consistency. It has become a pivotal process to examine the various aspects of technologies used by databases so as to reach at the right decision [42].

2.4 Oracle 12c

White Paper Oracle Database 12c – Built for Data Warehousing [36]:High-level framework is offered by author in the initial part of this paper. An ephemeral discussion about Oracle's Information Management Reference Architecture is delivered and explained, further ahead the author contributes an indication regarding how finest we

should edifice are data for data warehousing. The maximum of the remaining discussion in the paper is committed to essential key technical encounters and how the functionality of Oracle Database 12c is used to discourse all the challenges. Precisely, the paper shields all the noteworthy points like: Assortment of data warehouse policy, outline for Oracle Exadata Database machine, control and keeping up huge volumes of data efficiently, distinguishing for manageability, stowing of data with efficacy by use of firmness, transmuting and stacking data proficiently, stuffing near real-time and in batches, using row and set based processing, augmenting scaling with parallelism and response times, use of peripheral tables for a flexible and consistent approach, elevating query performance, isolating for better performance, structures for optimizing the physical data model, managing an EDW, apportioning and managing possessions with Database Resource Management, using graphical interfaces for observing and managing the database which are Oracle Enterprise Manager Express or Oracle Enterprise Manager Cloud Control and lastly, for optimal query performance by managing optimizer figures.

Oracle Partitioning was first announced in Oracle 8.0 in 1997, and in Oracle database it is one of the most normally used and effective functionalities. Enormous tables and catalogs are subdivided into reduced parts, performance is also improved, and availability for thousands of applications is precisely enabled in Oracle Partitioning. For diverse mission perilous systems of any figure like OLTP, data warehousing, or varied workloads of any extent – from hundreds of Gigabytes to Peta bytes, queries and preservation procedures are hustled up by an directive of magnitude. Virtually unconstrained data clipping is permitted by break down of partitions and tables into minor regions, with new functionality in Oracle Database 12 Release 12.1.0.2 on engineered systems along with zone maps. By help of partitioning some of the hardest issues which are created by cutting-edge applications are being tackled by database administrators and developers. Partitioning is also an important tool for developing systems that involve extreme high requirements or in other words multi-terabyte systems. Furthermore, it can significantly reduce the total cost of data proprietorship, with addition to “tiered achieving” tactic which keeps older important information still online on low cost storage devices in the most optimum compacted format. When Heat Maps and Automatic Data Optimization is used along with the renewed functionality i.e. partitioning, it delivers a programmed and

modest way to contrivance an Information Lifecycle Management (ILM) strategy. Oracle Database 12c announces the 10th generation of Oracle partitioning which have facilitated all the users more than ever to take advantage of it for their own personal desires and for over a decade it also enhanced the accessibility, manageability and performance for more than tens of thousands of customers by meeting their necessities [37].

A universal approach is mandatory to appliance database consolidation and DBaaS so that approaches could be designed to discourse one set of objective (for instance, reducing systems footprint) does not create new experiments along other proportions of a consolidated atmosphere (for instance, HA, management costs, data protection, or performance). The obligatory explanation for well-organized, steadfast database consolidation and DBaaS is provided by Oracle enterprise manager, Oracle MAA best practices and Oracle Database 12c with Oracle Multitenant. While these resolutions permits DBaaS and consolidation on any platform, the total life-cycle costs are also diminished and by means of the Oracle engineered systems finest service is accomplished with full comprehension [38].

A simplified path is delivered for business customers and SQL designers by Oracle's analytical SQL which provisions the most significant operative and business astuteness broadcasting requirements. Business customers and SQL designers can respectively benefit from enhanced query performance across vast range business intentions and from the increased production by moving the processing inside the database which will assist them in their field of work.

The systematic features of Oracle's in-database afford various advantages to the business users and Information Technology teams, some of which are as follows:

- Increases manageability.
- Augmented developer productivity.
- Curtails learning struggle.
- Offers speculation fortification.
- Supplies amplified query speed.

It becomes a substantial working tool for all the SQL users: data warehouse, DBAs, application developers, when the supremacy and tractability of Oracle's diagnostic functions is united with their observance to intercontinental SQL principles [39].

The client is left unaware of the transaction whether it is committed or not, while the Transaction Guard is not used, because the commit communication that is sent back to the client is not long-lasting, if a transaction has been already underway and commit has been allotted. If the non-transactional state is previously committed or if it is inappropriate than the transaction cannot be authentically resubmitted. Further, resubmission can lead to the transactions in the improper state, out of order, or applied more than once, when there is an absenteeism of comprehensive evidence or consistent commit. Transaction Guard is harmless and accomplishes well with lower disbursements than home-grown resolutions for a recognized conclusion. Various equivocal errors which can further lead to customer assist calls, abundant opportunities and user frustrations can be eluded by making use of Transaction Guard [40].

The purpose-built nature of most NoSQL DBMS allows them to benefit from advances in other technology areas, such as networking, faster. Traditional relational database management systems have a longer history and need to maintain backward compatibility, hence, may need some time to adopt new technologies, while they eventually will adapt accordingly.

Most of the leading RDBMS vendors listed in Gartner's recent Magic Quadrant (MQ) have already solved the problems that nearly all NoSQL databases will have to solve when hitting certain thresholds as a result of growth. Growth and requirements of business critical applications will force almost all NoSQL DBMS vendors to provide comprehensive solutions for data consistency, security and overall management, cutting into the cost benefits they seem to provide at first glance.

Concluding, the leading relational DBMS providers will not rest to provide the same benefits that NoSQL databases provide. They will add the additional benefits they already contain and offer the integrated solution at a competitive price point. First in line: Oracle with its rich DBMS solution portfolio, including its flagship solution, the Oracle Database 12*c*. Welcome to the future [6]

Chapter 3 Problem Statement

Growing global market in last 20 years brought exponential growth in data. For the business players to compete in the fast changing business scenarios, decision making portfolios should be as strong as the customer end portals else. As the decision support systems are based on facts, and source of collecting such facts is the data generated by these business applications. This data is not in conventional form like relational databases, flat files but a far more complex in nature.

Depending on the business scenarios or the subject under analytics velocity, variety and scalability are three main features demanded from a data analytics system.

In the current scenario selection of the database for any project is very crucial as we know that RDBMS behaves poorly with agile development approaches, because each time we complete new features, the schema of your database often needs to change. In present scenario there's also no way, to store data that's completely unstructured or unknown in advance sing a relational database. RDBMS doesn't enforce the coherent table structure to be implemented by database designers, inexperienced may create un-necessary complexity by not choosing correct data types. As RDBMS relates tables using shared keys, choosing different data types to link the keys could corrupt the data.

To fulfill the demand created by the limitations of RDBMS and to cater the demand for a system which could store more complex data, databases such as NoSQL databases are built to allow the insertion of data without a predefined schema.

Business application being developed are majorly web oriented, data intensive and large scale. These applications can be accessed through various devices including mobiles. Big data sizes vary greatly from dozens of terabyte to many petabytes all in a single set of data. It is difficult to capture, manage and process such huge data.

Because of the increasing demand in the data being processed and the various issues related to their storage, NoSQL databases came in picture. This led to evolution of the popular word these days known as "Big Data". Various factors associated with this are the mainly known to us in form of 5V's of Big Data which are: volume, velocity, value,

variety and veracity. Volume refers to the fact of storing data in huge amount may be in terabytes, velocity represents the processing speed of the system, value refers to the statistical, events or correlations related, veracity refers to its origin, availability and finally the factor considered in this thesis which is variety which refers to storing of structured and unstructured data.

After looking into the database systems which satisfies the demand created by more complex data, we studied three database systems, two out of which support unstructured data storage, i.e. Mango DB, Oracle NoSQL and one being advanced Oracle relational database i.e. Oracle 12c.

MongoDB is database systems used for the document storage. It is designed to address the need of high performance systems providing high availability and also automatic scaling. It saves the documents in a BSON format i.e. binary JSON and field values aside from the usual JSON types can include other documents, arrays and arrays of documents. Every field in MongoDB can be indexed and queried. It block all the operations including read using has a write. It supports dynamic consistency in which guarantees success in each write operation. Some of the additional features of MongoDB are geospatial indexing which allows location based queries and GridFS for supporting big files. It can be used as primary storage for CMS content.

Oracle NoSQL Database is a distributed key-value database. It is developed to address the needs of reliable, high available and scalable data storage across a configurable set of systems that function as storage nodes. Data storage is done in the form of key-value pairs, written to particular storage node(s), on the basis of primary key which is a hashed value. Storage nodes keep on getting replicated to ensure high availability of the database system, on event of a node failure rapid failover is done and load balancing of queries is done in optimal manner. Applications are designed using Java/C API to perform read and write data operations.

Oracle 12c provides a few SQL optimizing features which are automatically enabled on the usage demand. It has high consolidation density, rapid provisioning and cloning using SQL by using it a pluggable database from one container DB and can be plugged into another. It comes up with new paradigms for rapid patching and upgrade, the time

invested on patching one container database results in patching all of its many pluggable databases.

A NoSQL database extends a system in the forms where it was limited while using RDBMS. A main difference being, no need to define explicit schema unlike RDBMS. It is able to scale out and make use of new nodes as per the storage needs. NoSQL databases are high availability systems as these are designed with an assumption of hardware failures. These are high performance systems with low latency. Data models in NoSQL database systems are broadly categorized as document stores, column-oriented stores, Graph databases, Key-value stores. Document stores are used to store semi-structured document object data and metadata. These stores can be queried like relational database but doesn't adhere to strict structure of database table. Some e.g. of these databases may include MongoDB, CouchDB. Key-value stores are the simplest model and fastest way to get data by known key, but without the flexibility of more advanced querying. Example of Key value store is Riak, Amazon S3. Column-Oriented stores is advanced Key-value store data model and each storage block contains data from only one column. Example of column oriented includes HBase and Cassandra. Graph database allows objects to link and be linked by other objects so constructing a graph like structure and e.g. of this database includes Neo4J. NoSQL database systems should be used to address specific demands from an enterprise.

As we saw that NoSQL database are extending features which were absent in relational database, So we want to do the performance analysis of the three databases MongoDB which is NoSQL database, Oracle NoSQL is also a NoSQL database and Oracle 12c which is a latest Oracle RDBMS database. Study was performed on these database systems and their performances were further analyzed and compared. We compared theses three databases on similar dataset and on similar infrastructure and record the performance that can help someone to choose the database for their respective projects. As the selection of database depends on major two scenarios such as the data that we are going to store contains unstructured data or just structured data. So we have divided the problem and compared the databases on different dataset.

We first compared the three databases MongoDB, Oracle NoSQL and Oracle 12c on structured data and then compared the two databases MongoDB, Oracle NoSQL for unstructured data.

This study relieved the facts about performance these respective database systems delivers for the conventional datasets which is relational data, and also when used to store unstructured data.

The various advantages provided by NoSQL databases as analyzed from the literarture survey part are as follows:

1. Non relational and schema less data models.
2. High scalability.
3. High performance.
4. Large variety of data can be stored.

In this thesis work performed is on these three databases. Three of the databases are compared and the outcomes obtained from different testing phases are used to compare their performances. The NoSQL are databases chosen are because of increasing demand of the data in the daily routine.

In the initial phases all the three databases were installed and implemented on the system. In this case the system used was of 64 bit. In order to process any data on these databases, al l the three databases are required to be working or running. For MongoDB different commands are run in command prompt to start its working. Same is the case with Oracle NoSQL. For Oracle 12c one need to check whether its services are running or not. Connection is formed with the database using Oracle Developer.

In next stages the data used to perform the test was the structured data. Structured data was inserted in all the databases. The dataset used was same for the three databases. Various numbers of iterations are performed. In our case 10 iterations were conducted. The coding was in java and with the help of the command; systems time was taken as the output which is provided to us in form of nanoseconds. The data folder is created in the system in which the entire data is being stored which is inserted during different iterations. The path is specified of the data folder. In the data folder we have db folder

which has all the data. The structured data is stored in form of local files as specified in the folder. Analysis was performed for three different types of queries which include:

1. Time taken by the database to insert the data
2. Time taken by database to retrieve data
3. Time taken by databases to delete data

Performing the tests, the outcomes obtained are then stored in the form of table as well in the form of graphs. Graphs provide the better view of the time taken by the different databases.

Further the databases are compared using unstructured data. Now further along with the data images are also stored. In this case only two of the databases were compared which are MongoDB and Oracle NoSQL as the other database i.e. Oracle 12c is not compatible with storing the unstructured data. Dataset for both the database is same. The data inserted during the iterations are stored in the data folder whose path is specified. As this dataset is different and it contains images, its data is stored in form of grid files as specified in the folder. The results obtained from this testing are then stored in form of tables and graphs to depict the clear picture of their performances.

In this thesis the analysis is performed using the structured data and unstructured data (in which images are added in addition). Further analysis can be performed using large datasets which may include videos or emails.

Chapter 4 Implementation Details

This chapter contains the implementation details of research performed. It is documented in the following three categories providing different level of details or different phases of implementation.

- Installation
- Implementation
- Snapshots from the implementation performed

Implementation is performed on Windows 64 bit platform.

4.1 Installation

4.1.1 MongoDB Installation:

a. First we downloaded MongoDB and installed it in our system.

b. After installation we need to create a directory "data\db" in our system where MongoDB will store all the data.

c. We start MongoDB by running command Mongod.exe. and pass the data directory we created with –dbpath argument. Now our MongoDB is start and we will connect it.

d. We run Mongo.exe and it will connect to MongoDB.

4.1.2 Oracle NoSQL Installation:

a. First we downloaded Oracle NoSQL and installed it in our system.

b. After installation we start it Oracle NoSQL by running command " java -jar lib/kvstore.jar kvlite ". Now our Oracle NoSQL is start.

c. It will show success connection as "Created new kvlite store with args:

d. -root ./kvroot -store kvstore -host myhost -port 5000 -admin 5001".

4.1.3 Oracle 12c Installation:

a. First we downloaded Oracle 12c and installed it in our system.

b. After successful installation we can see that various services has been started in our system that will manage the Oracle12c database.

c. Oracle12c store all the database data in " OracleHomeUser" folder in our system. "OracleHomeUser" is the user we created during the installation.

4.2 Implementation

4.2.1 Mongo DB Implementation

a. We did all the implementation with the help of java.

b. Java is installed in the system and eclipse tool is used to make project where all the coding is done.

c. We make a project of MongoDB and import all the jars that are required to run MongoDB with Java.

d. We make the connection from java to MongoDB.

e. All the coding is done using MongoDB Java API

f. We created a dataset of approx 1 GB and then perform the three operations i.e. insertion of new data, retrieval of data and deletion of data.

g. We used timestamps from Java to measure the time taken by these operations. We measured the time in nanoseconds and recorded it.

h. For each operation (insertion, retrieval, deletion) we did ten iterations each.

4.2.2 Oracle NoSQL Implementation

a. We did all the implementation with the help of java.

b. Java is installed in the system and eclipse tool is used to make project where all the coding is done.

c. We make a project of Oracle NoSQL and import all the jars that are required to run Oracle NoSQL with Java.

d. We make the connection from java to Oracle NoSQL.

e. All the coding is done using Oracle NoSQL Java API

f. We created a dataset of approx 1 GB and then perform the three operations i.e. insertion of new data, retrieval of data and deletion of data.

g. We used timestamps from Java to measure the time taken by these operations. We measured the time in nanoseconds and recorded it.

h. For each operation (insertion, retrieval, deletion) we did ten iterations each.

4.2.3 Oracle 12c Implementation

a. We did all the implementation with the help of java.

b. Java is installed in the system and eclipse tool is used to make project where all the coding is done.

c. We make a project of Oracle 12c and import all the jars that are required to run Oracle 12c with Java.

d. We make the connection from java to Oracle 12c.

e. All the coding is done using Oracle 12c Java API

f. We created a dataset of approx 1 GB and then perform the three operations i.e. insertion of new data, retrieval of data and deletion of data.

g. We used timestamps from Java to measure the time taken by these operations. We measured the time in nanoseconds and recorded it.

h. For each operation (insertion, retrieval, deletion) we did ten iterations each.

4.3 Snapshots of Implementation

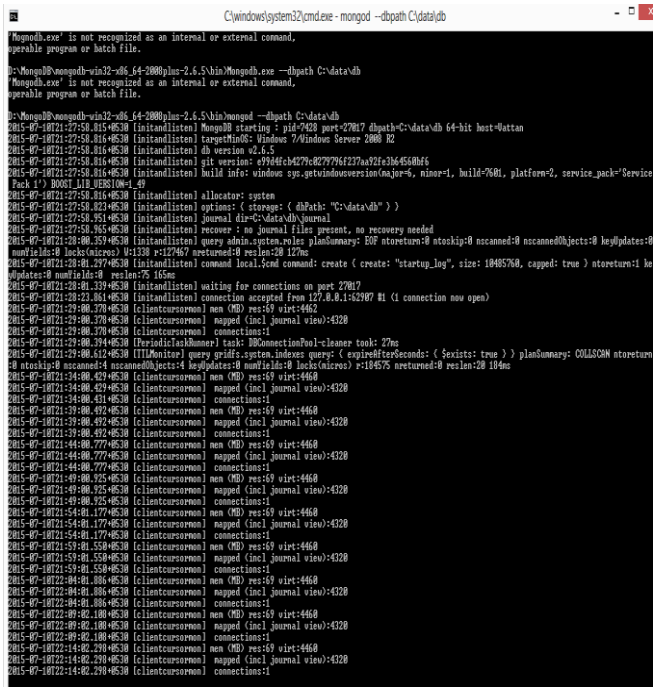

Figure 4-1. MongoDB run.

In this figure we can that MongoDB is started with data directory path (C\data\db).We started it by running Mongod.exe. We can see it is waiting for connection at port 27017 and when me start MongoDB it will make connection to it.

Figure 4-2.MongoDB start and connection

In this figure it is clear that MongoDB is started. We started it by running Mongo.exe. When it will run it will make connection as shown in fig1. It will connect to test.

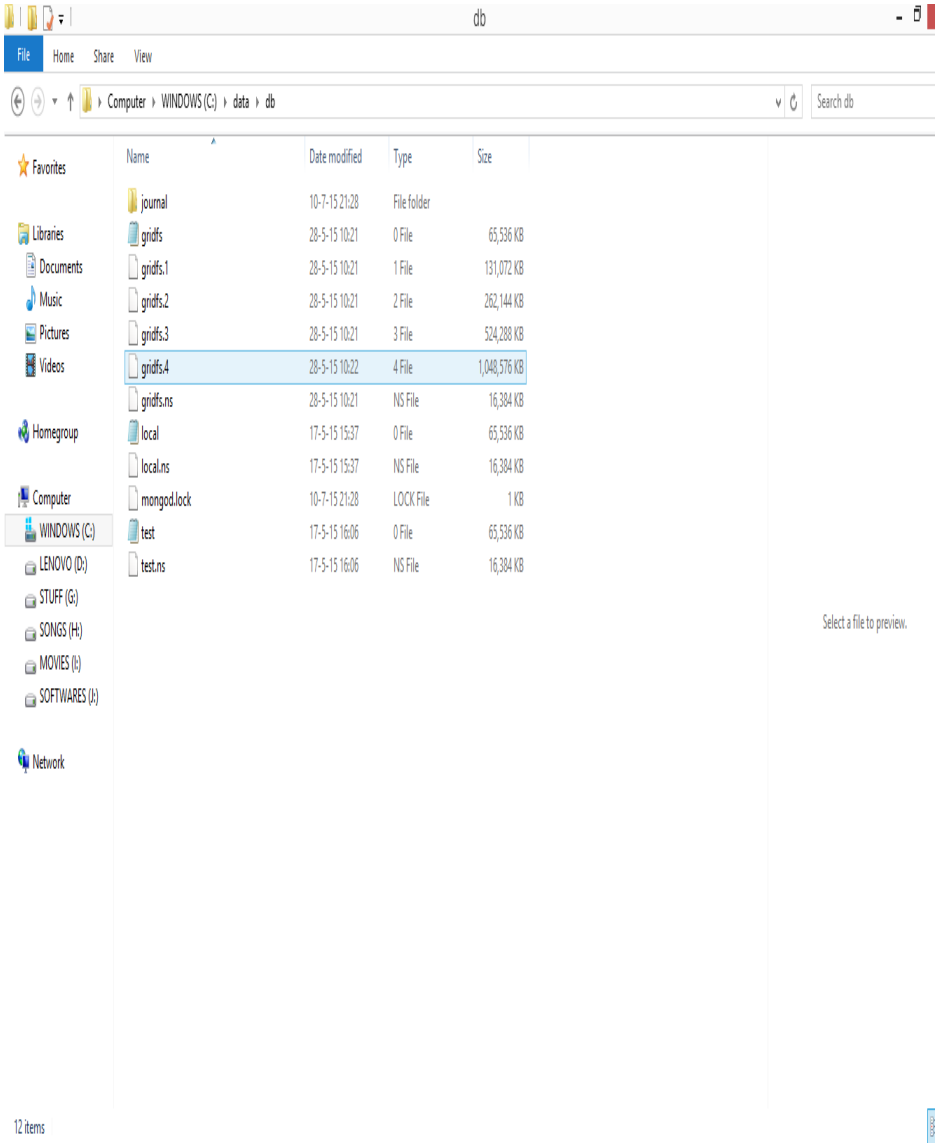

Figure 4-3.MongoDB Data Storage

In this figure we can see the storage directory of MongoDB that we given during the running.

Figure 4-4.Oracle Start

We can see in this figure that we are starting the Oracle NoSQL using KvLite that is also a part of oracle NoSQL. It will make storage node and replica node.

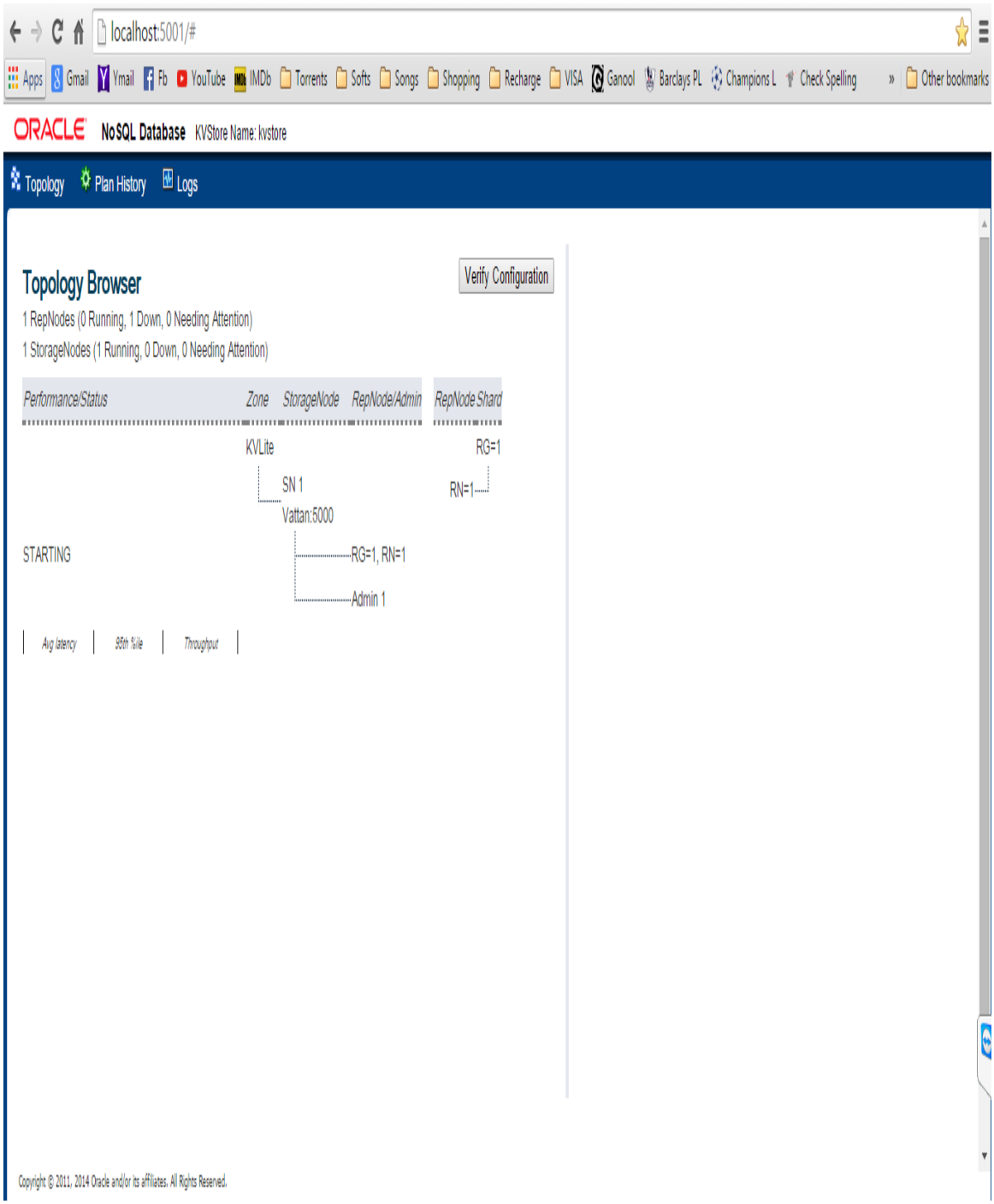

Figure 4-5.Oracle NoSQL Topology

In this we can see the oracle topology. It is showing that one storage node and one replica node is created on the machine.

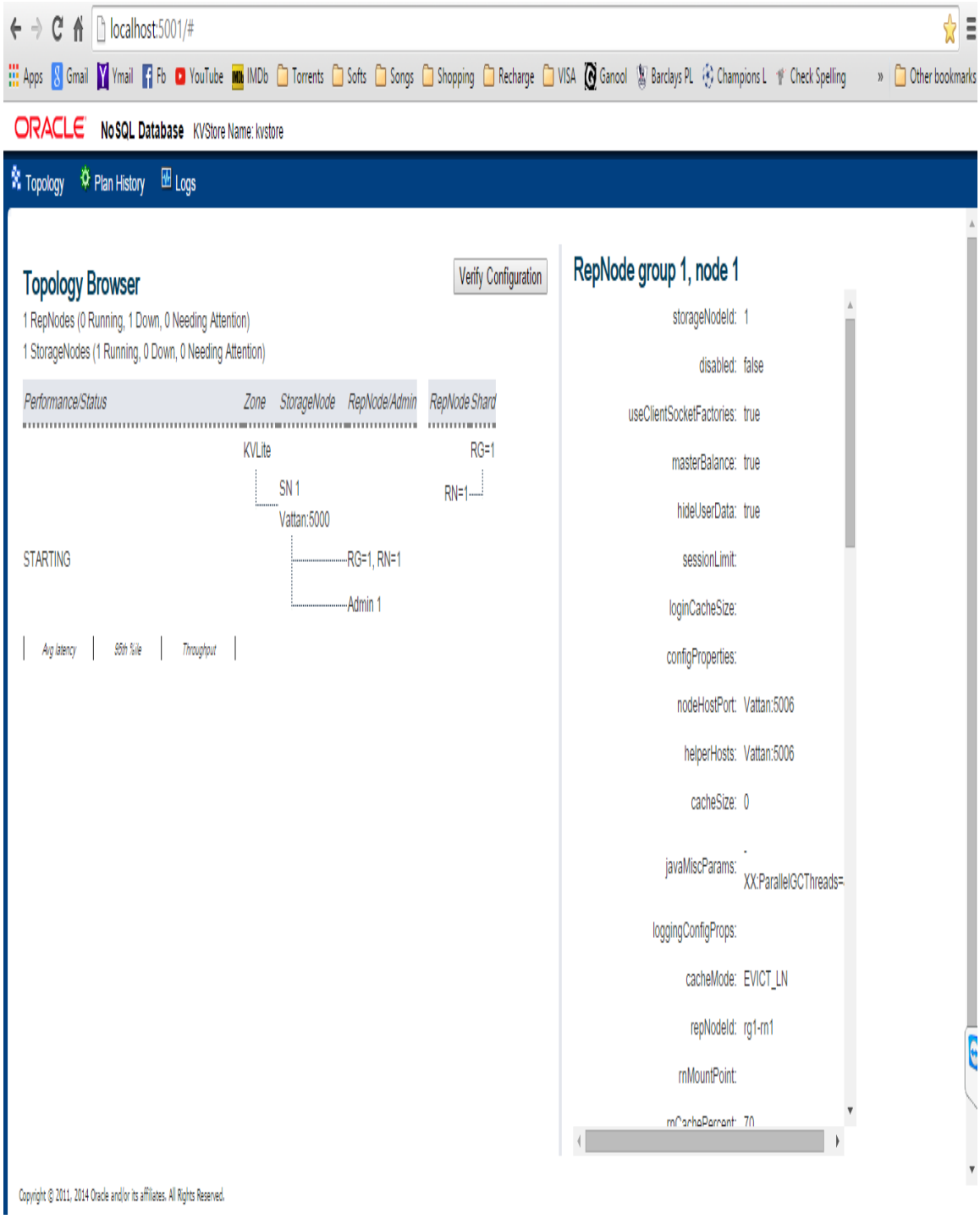

Figure 4-6 Oracle NoSQL Nodes.

In this figure we can see the detailed information about the nodes that has started.We can see that it has been started at port 5006.

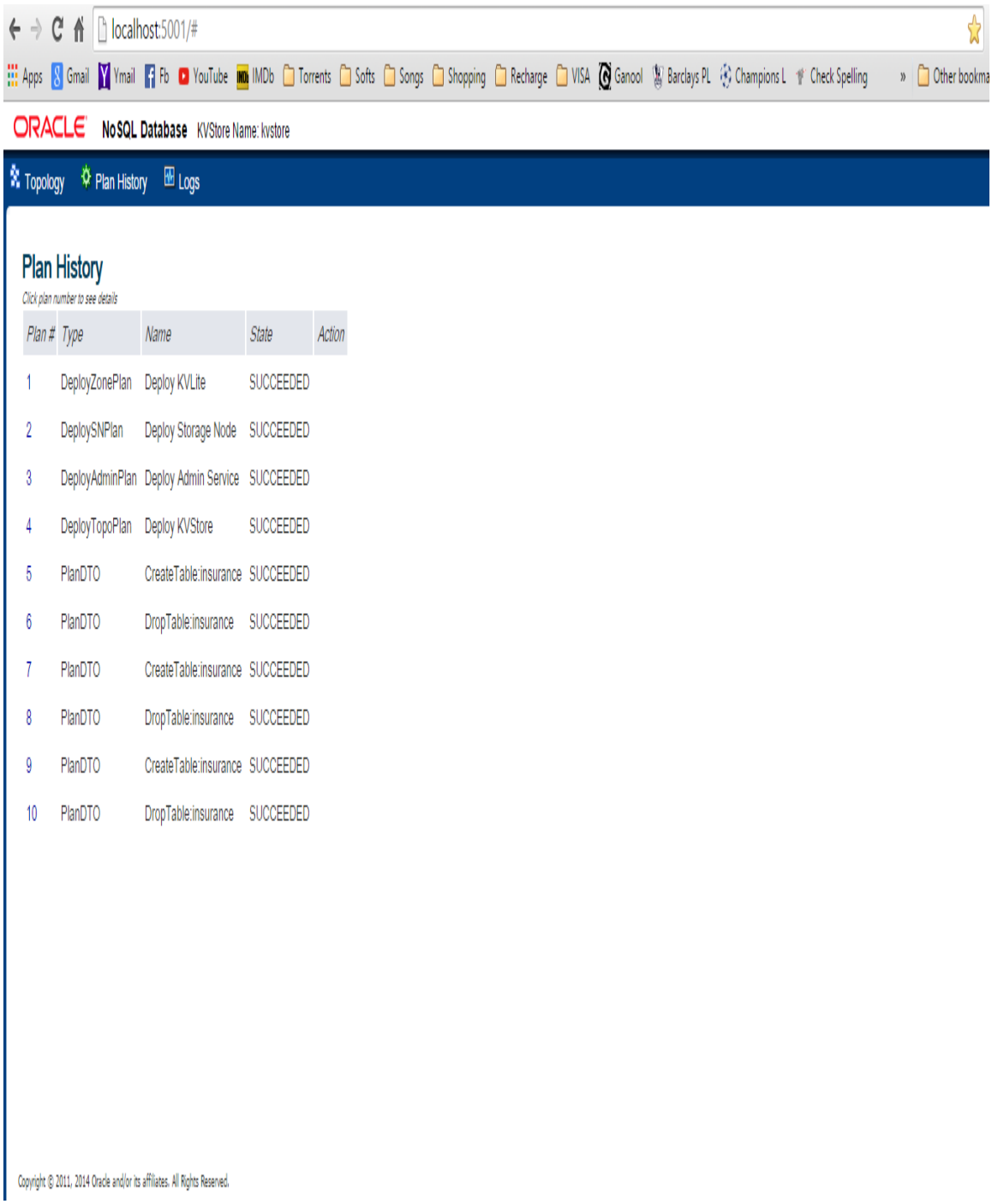

Figure 4-7.Oracle NoSQL Plan History

In this figure we can see the NoSQL Plan History we can see all the information about the Oracle NoSQL operations such as deploy of Kvlite, creation of table etc.

Figure 4-8.Oracle NoSQL Storewide Logs

In this figure we can see the Storewide Logs of OracleNoSQL, at which node what operation is taking place

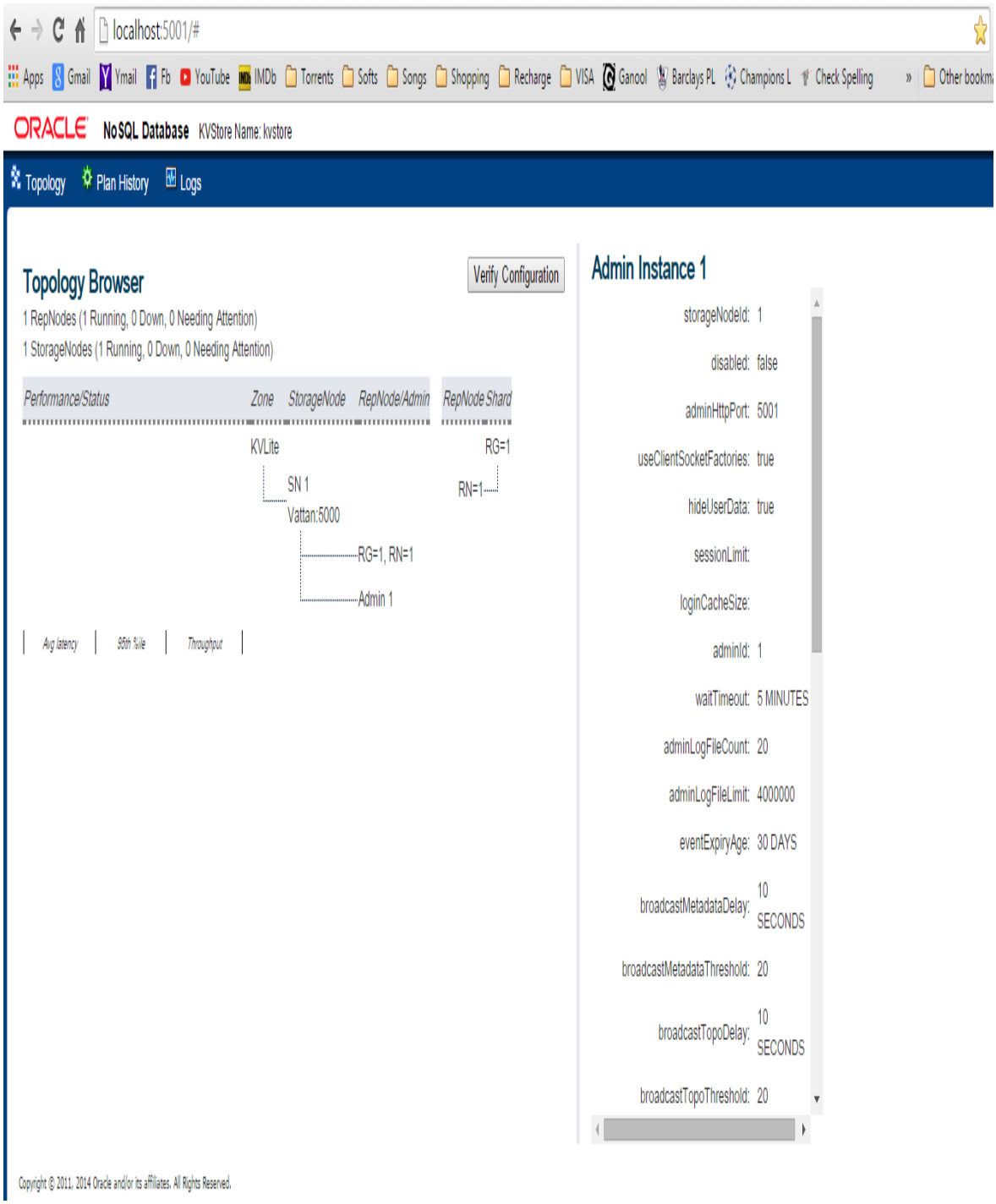

Figure 4-9.Oracle NoSQL Admin Instance

In this figure we can see the Admin Instance. We can see that the admin port is 5001.We can manage all the information from this port.

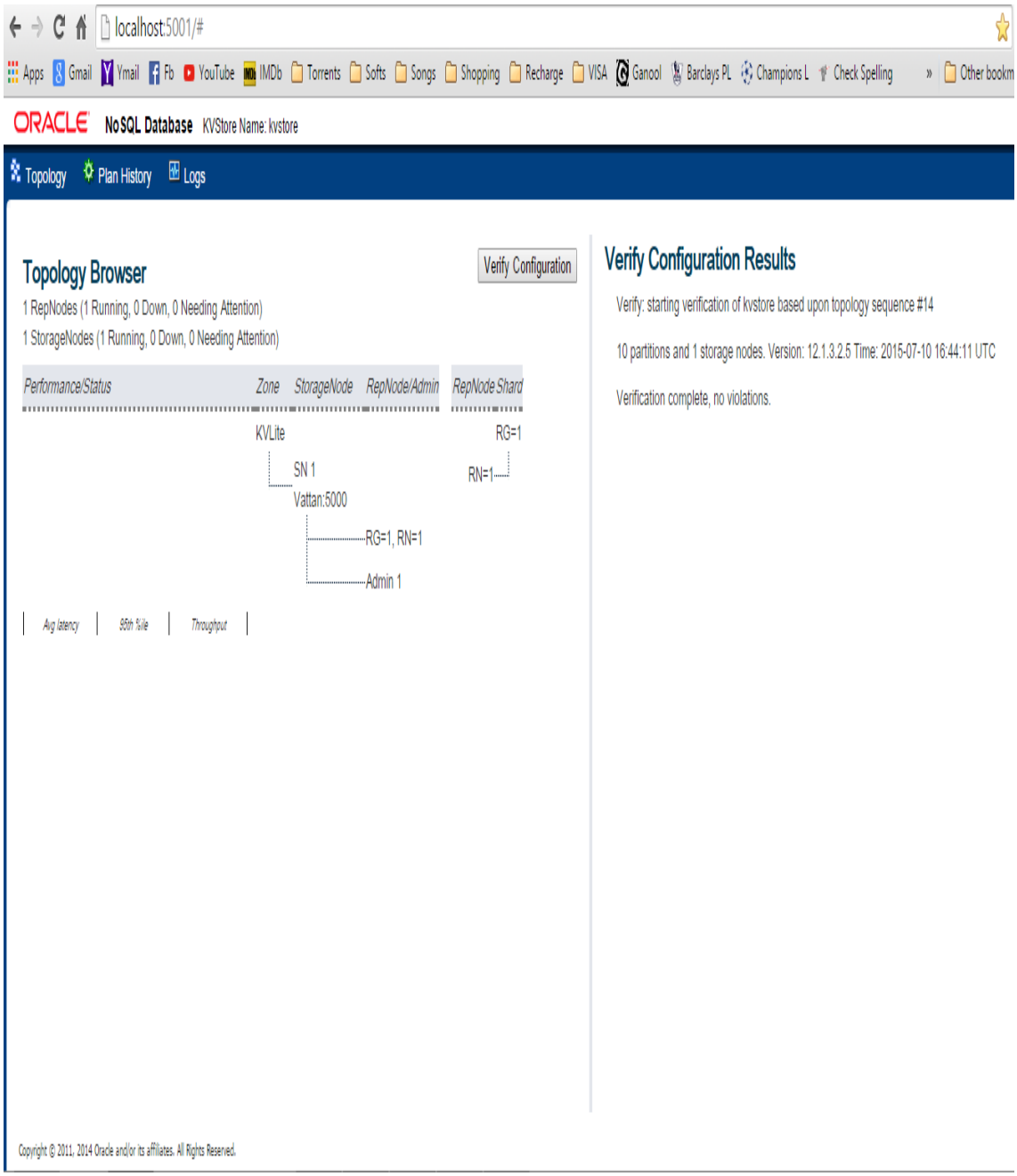

Figure 4-10.Oracle NoSQL Verify Configuration

In this figure we can see the verify configuration for our nodes. We can see that the partitions are in limits and there is not any violation.

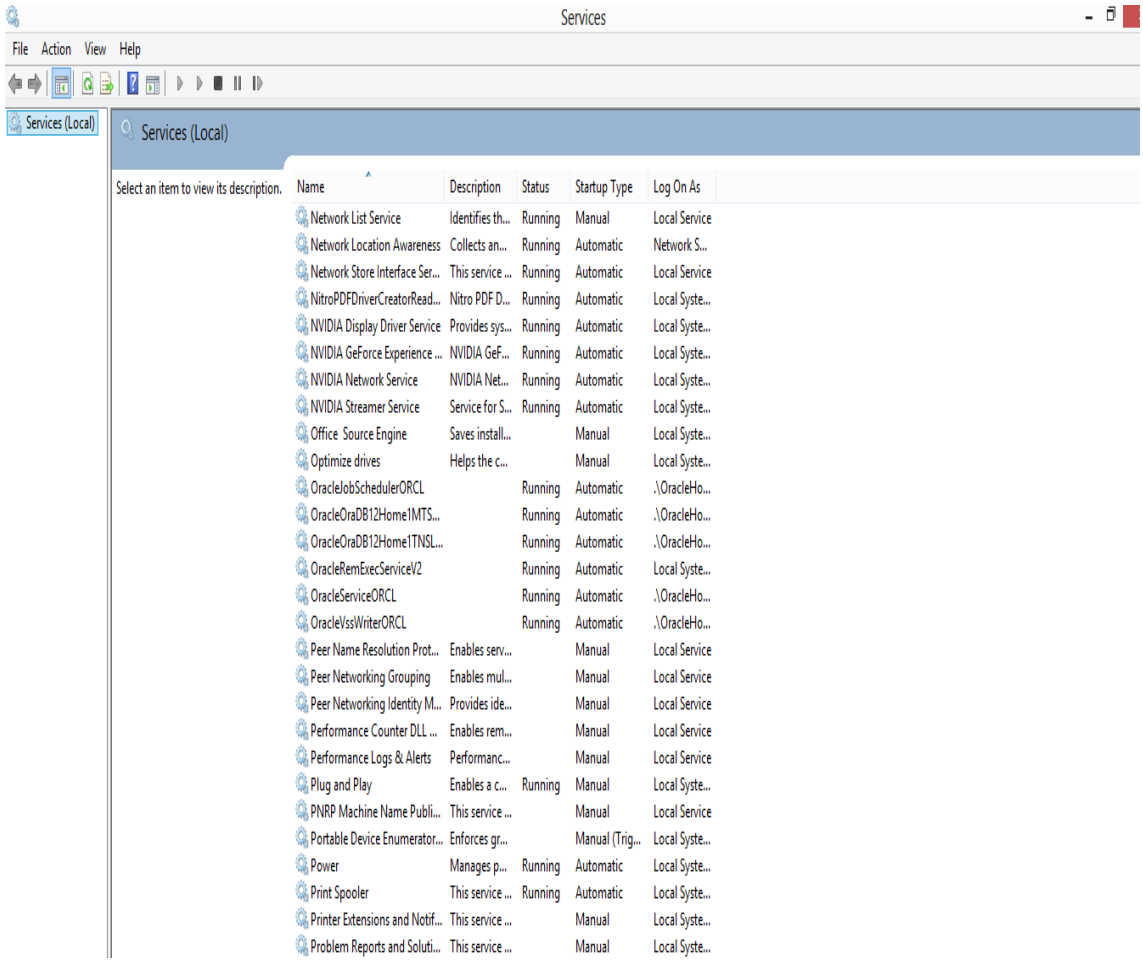

Figure 4-11.Oracle 12c Services

In this figure we can see the Oracle 12c Services that are running. Oracle 12c has services that run in our system and do all the necessary operations that are required.

Chapter 5 Testing and Results

The code for the project has been done in Java, and the simulations have been done on Windows 64 bit machine. The data insertion for small data sets i.e. data set of 10 was done manually. A large set of data (1.5 GB approx) has been taken from free data sources, just in order to save time and effort of creating huge data for analysis. The time for operations: insertion, selection and deletion from data have been recorded.

5.1 Comparison of MongoDB, Oracle NoSQL and Oracle 12c

We did comparison of these three databases on structured data. We created projects for all the three databases in Eclipse (Development Tool) as Java Project and we used the Java APIs of all the three databases to communicate with them and finding the results.

We did the operations of insertion, selection (retrieval) and deletion of data from the databases and recorded the time taken by them in nanoseconds.

Table 5-1.INSERTION TIME (IN NANOSECOND)

MongoDB	Oracle 12c	Oracle NoSQL
89955093	9739968	1408435
87899927	9681967	1683552
88558977	9406337	1215443
91906064	9467417	1434121
88972679	7146886	1501339
92253553	8323833	1324612
85882743	8454720	16325454
87180824	9600870	1532415
87365605	9849810	1426583
89698967	8396205	1325412

In the table first column is of MongoDB, second column is of Oracle 12c and third column is of Oracle NoSQL and the values are of time taken in nanoseconds

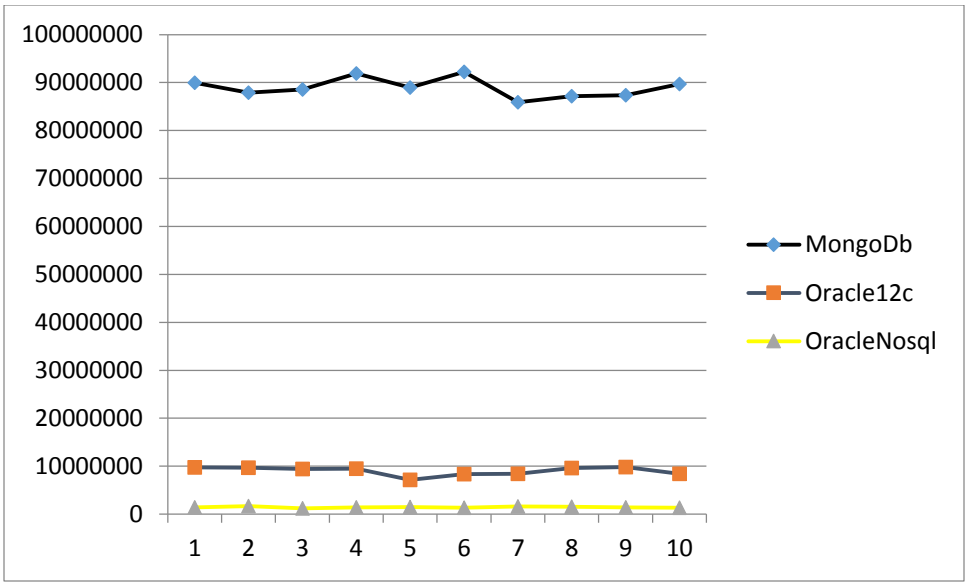

Figure 5-1.Graph showing the trends of insertion time

In this graph x-axis is showing the no of iterations and y-axis is showing the time taken in nanoseconds. From this graph it is clear that the MongoDB has taken maximum time and Oracle NoSQL has taken minimum time.

Table 5-2.RETRIEVAL TIME

MongoDB	Oracle 12c	Oracle NoSQL
88371152	211497843	44209169
77705195	213083873	48878461
91195686	202773139	47500823
90544850	206102776	43170007
84389618	224027992	43012505
98168572	210193090	46254224
82411442	206818286	47251891
93512112	20427506	45771650
85672813	210501057	42310523
82782543	225827033	47215967

In the above table first column is of MongoDB, second column is of Oracle 12c and third column is of Oracle NoSQL and the values are of time taken in nanoseconds.

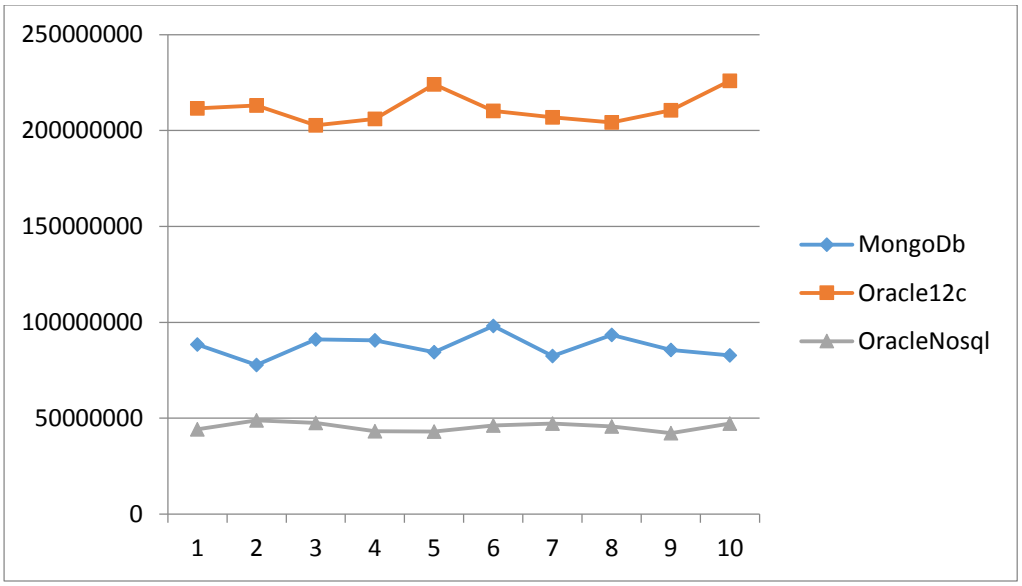

Figure 5-2.Retrieval Time

In this graph x-axis is showing the no of iterations and y-axis is showing the time taken in nanoseconds. From this graph it is clear that the Oracle12c has taken maximum time and Oracle NoSQL has taken minimum time.

Table 5-3.DELETION TIME

MongoDB	Oracle12c	Oracle NoSQL
4354139	12390024	1979715
5210288	11987613	2475541
6673643	11065766	1998193
5169739	13260544	1789288
4883842	13535147	1953247
4427538	12588149	1936523
5399687	15174045	1862451
4533274	12085166	2015382
4532247	12419280	1863774
7992255	11396317	1853216

In the table above first column is of MongoDB, second column is of Oracle 12c and third column is of Oracle NoSQL and the values are of time taken in nanoseconds.

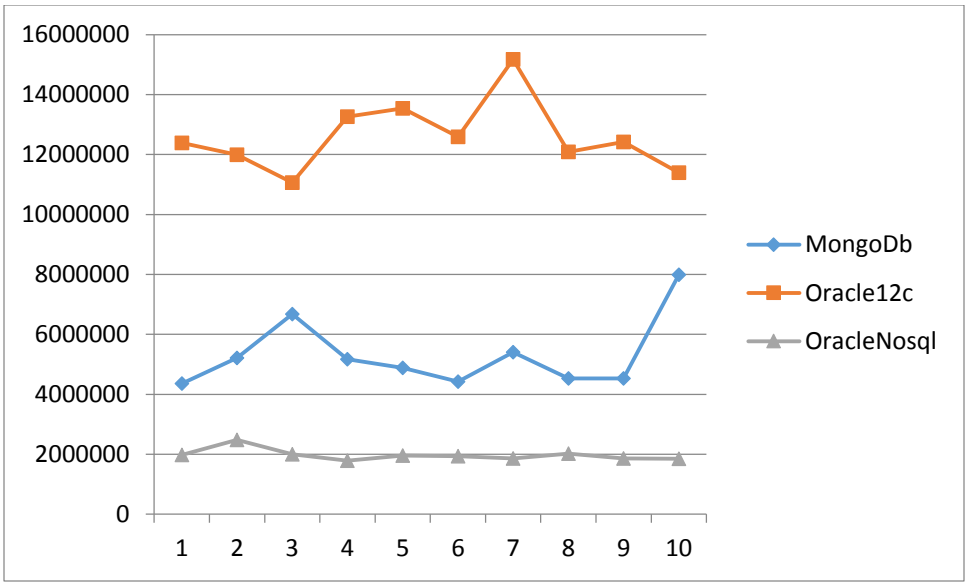

Figure 5-3.Graph representation of Deletion Time

In this graph x-axis is showing the no of iterations and y-axis is showing the time taken in nanoseconds. from this graph it is clear that the MongoDB has taken maximum time and Oracle NoSQL has taken minimum time.

5.2 Comparison of MongoDB and Oracle NoSQL

We did comparison of these two databases on unstructured data. We created projects for all the two databases in Eclipse (Development Tool) as Java Project and we used the Java APIs of all the three databases to communicate with them and finding the results.

We did the operations of insertion, selection (retrieval) and deletion of data from the databases and recorded the time taken by them in nanoseconds

Table 5-4.INSERTION TIME FOR MULTIMEDIA DATA

MongoDB	Oracle NoSQL
83777053	47355268
90998805	47509611
91050120	45251703
83506262	52110689

88499316	57031112
88626817	42229581
97399104	48598696
83372446	45740785
84677453	46160393
7755204	21031337

In the table first column is of MongoDB, second column is of Oracle NoSQL and the values are of time taken in nanoseconds

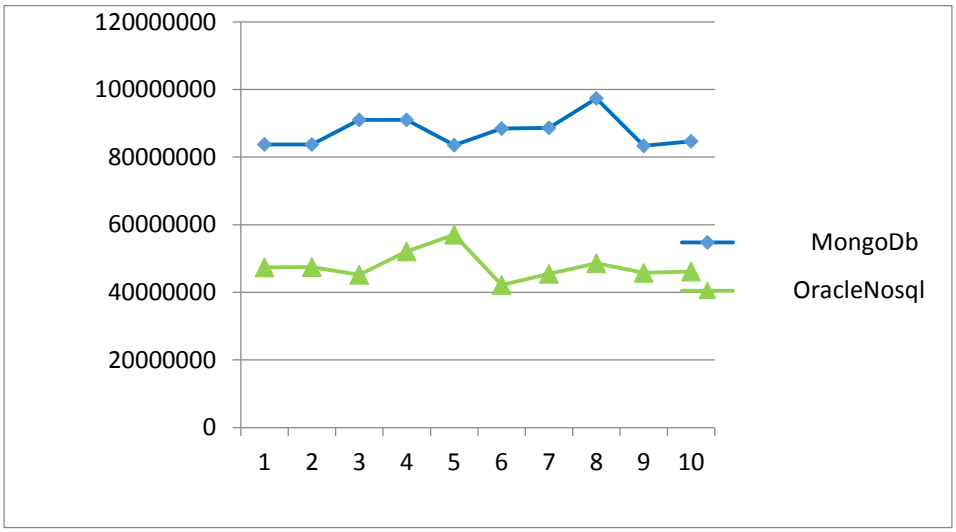

Figure 5-4.Chart depicting insertion time for multimedia data

In this graph x-axis is showing the no of iterations and y-axis is showing the time taken in nanoseconds. From this graph it is clear that the MongoDB has taken maximum time and Oracle NoSQL has taken minimum time.

Table 5-5.RETRIEVAL TIME FOR MULTIMEDIA DATA

MongoDB	Oracle NoSQL
2224355	7270070

2563041	8637966
2644358	7564042
2789226	7193364
2793174	8878219
2707910	7493727
2333697	7915703
2531067	8527943
2158450	7516622
2680278	7424950

In the table first column is of MongoDB, second column is of Oracle NoSQL and the values are of time taken in nanoseconds

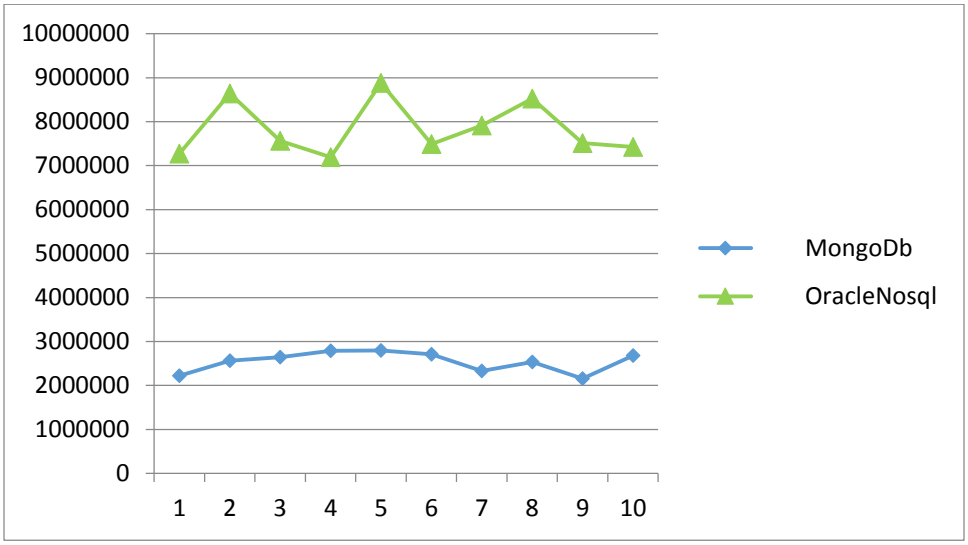

Figure 5-5.Retrieval time for multimedia data

In this graph x-axis is showing the no of iterations and y-axis is showing the time taken in nanoseconds. From this graph it is clear that the MongoDB has taken maximum time and Oracle NoSQL has taken minimum time.

Table 5-6.DELETION TIME FOR MULTIMEDIA DATA

MongoDB	Oracle NoSQL
7224120	17167202
9562724	22459469
8911915	18029707
7948861	21060514
9559791	19328794
7327055	17968918
8929788	22635918
7083534	24413033
8398469	18723659
7755204	21031337

In the above table first column is of MongoDB, second column is of Oracle NoSQL and the values are of time taken in nanoseconds

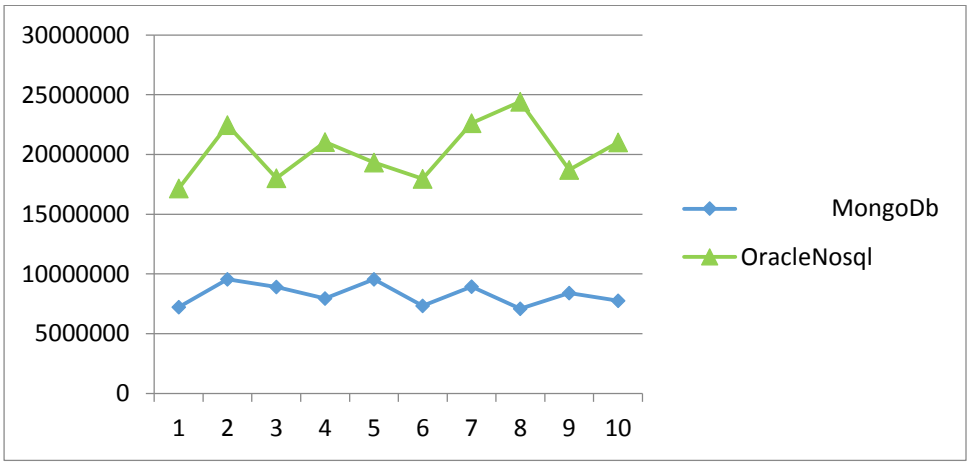

Figure 5-6.Deletion time for multimedia data

In this graph x-axis is showing the no of iterations and y-axis is showing the time taken in nanoseconds. From this graph it is clear that the NoSQL has taken maximum time and Oracle MongoDB has taken minimum time

Chapter 6 Conclusion

6.1 Conclusion

In our Thesis we compared the databases both RDBMS and NoSQL using structured data set and then compared NoSQL databases using multimedia data set or we can say unstructured data on their query processing time in specific insertion time, retrieval time and deletion time. Insertion and selection is the most frequent operations that happen on a database so we kept that in mind and we found the performance results. By analyzing the above observations we conclude the following:

1. For structure data the insertion time for NoSQL is better than ORACLE 12c and MongoDB.
2. For structure data the retrieval time of NOSQL is also better than Oracle 12c and MONGO DB.
3. For structure data the time to delete data using NOSQL ORACLE is less than Oracle 12c which is less than MONGO DB.
4. For multimedia data, insertion, deletion and retrieval time taken in Oracle NO SQL is better than MONGO DB.

6.2 Future Work

Database is wide field and there is a lot opportunity in the field of research as the data is increasing exponentially. In future we can compare these databases on large dataset and different type of dataset that includes large video files and other multimedia files.

References

[1] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," *in Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, Oct. 2011, pp. 363 –366.*

[2] K. Chodorow and M. Dirolf, "MongoDB: The Definitive Guide. O'Reilly Media," *September 2010.*

[3] H. Kaur, "Improvement in Load Balancing Technique for MongoDB Clusters," *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 8– No.4, February 2015, pp 31- 35.*

[4] B. Tudorica and C. Bucur, "A comparison between several NoSQL databases with comments and notes*," in Roedunet International Conference (RoEduNet), 2011 10th, june 2011, pp. 1 –5.*

[5] An Oracle White Paper, "Oracle NoSQL Database," *November 2012.*

[6] An Oracle White Paper, "Back to the Future with Oracle Database 12*c*," *April 2014*

[7] MongoDB official guide. Available: http://www.mongodb.org.

[8] Kristina Chodorow and Michael Dirolf, "MongoDB: The Definitive Guide,"

ỎReillyMedia, September 2010. p135.

[9] Oracle official guide. Available: http://www.oracle.com

[10] An Oracle White Paper, "A Technical Overview of New Features for Automatic Storage Management in Oracle Database 12c", *January 2013.*

[11] Oracle Corp., white paper, "Optimizer with Oracle database 12c," (accessed 2014/07/14) . [Online]. Available: http://bit.iyllmPOVkV.

[12]Oracle Architecture Overview: Available: http://www.cs.utexas.edu/~cannata/dbms/Class%20Notes/02%20oracle-architecture.pdf

[13] Oracle Database 12c: Oracle Multitenant. Available: http://www.rmoug.org/wp-content/uploads/QEW_Presentations/2013/5%20-%20Consolidation%2012c.pdf

[14] K Bhoola, k Kruger, J Peick, P Pio and NA Tshabalala, "Big Data Analytics," *Presented at Acturial society os South Africa.*

[15] Shalom and Nati, "The Common Principles Behind The NOSQL Alternatives," *December 2009. – Blog post of 2009-12-15.*

[16] Govindaraju M., Dede E., and Gunter D., "Performance Evaluation of a MongoDB and Hadoop Platform for Scientific Data Analysis," (accessed 16 Aug 2014). [Online]. Available: http://datasys.cs.iit.edu/events/ScienceCloud2013/p02.pdf

[17] Matei T., "MongoDB Performance in the Cloud," (accessed 20 Aug 2014). [Online] Available : http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1300&context=etd_projects

[18] Arora R. and Aggarwal R., "Modeling and Querying Data in MongoDB," (accessed 22 Aug 2014). [Online]. Available : http://www.ijser.org/researchpaper%5CModeling-and-Querying-Datain-MongoDB.pdf, 2013

[19] Datastax Corporation 2013, "Benchmarking Top NoSQL Databases A Performance Comparison for Architects and IT Managers," (accessed 25 Aug 2014). [Online] Available: http://www.datastax.com/wpcontent/uploads/2013/02/WP-Benchmarking-Top-NoSQL-Databases.pdf

[20] Nyati S.S., Pawar S and Ingle R., "Performance Evaluation of Unstructured NoSQL data," *in International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, 2013.*

[21] NoSQL Databases Explained. Available: http://www.mongodb.com/nosql-explained. Retrieved: http://www.mongodb.com

[22] Brust A. and Dash J., "RDBMS vs. NoSQL: How do you pick?," Available: http://www.zdnet.com/rdbms-vs-nosql-how-doyou-pick-7000020803/.

[23] Kaur K and Rani R., "Modeling and querying data in NoSQL databases," *in IEEE International Conference on Big Data, Silicon Valley, CA, 2013.*

[24] Aggarwal R and Arora R, "Modeling and Querying Data in MongoDB," *in International Journal of Scientific & Engineering Research, vol. Volume 4, no. Issue 7, pp. 141-145, 2013*

[25] Alexandru B., Florin R. and Laura I. A., "MongoDB vs. Oracle - database comparison," *in Third International Conference on Emerging Intelligent Data and Web Technologies, Bucharest, 2012.*

[26] Coe, B., "To MongoDB, or Not to MongoDB," [Online]. Available: http://www.codemag.com/Article/1309051.

[27] A. R. Arora R, "An Algorithm for Transformation of Data from MySQL to NoSQL (MongoDB)," *International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE), vol. 2, no. 1, pp. 6- 12, 2013.*

[28] Banker K, "MongoDB in Action," *Shelter Island, NY: Manning Publications Co. pp. 3-126, 2012.*

[29] Chodorow K., "Scaling MongoDB," *L. M, Ed., Sebastopol, CA: O'Reilly Media, Inc. pp. 1-45, 2011.*

[30] Liu Y, Wang Y and Jin Y., "Research on the improvement of MongoDB Auto-Sharding in cloud environment," *in The 7th International Conference on Computer Science & Education, Melbourne, VIC, 2012.*

[31] Zugic G, "Selecting a MongoDB Shard Key," Available: https://goranzugic.wordpress.com/2014/05 /29/selecting-a-mongodb-shard-key/.

[32] Huang, Chao-Wen, Hu. Wan-Hsun, Shih CC, Lin BT and Cheng CW, "The improvement of auto-scaling mechanism for distributed database: A case study for MongoDB," *Network Operations and Management Symposium (APNOMS), 15th Asia-Pacific, Hiroshima, Japan,* 2013.

[33] Wang XL, Chen H and Wang Z, " Research on Improvement of Dynamic Load Balancing in MongoDB," *Dependable, Autonomic and Secure Computing (DASC), 2013 IEEE 11th International Conference Chengdu.*

[34] Oracle NOSQL. Available: http://www.oracle.com/technetwork/articles/bigdata/oracle-nosqldb-appdev-1891870.html, *December 2012*

[35] An Oracle White Paper, "Oracle NoSQL Database," *September2011.*

[36] An Oracle White, "Oracle Database 12c – Built for Data Warehousing," *February 2015.*

[37] An Oracle White Paper, "Oracle Partitioning with Oracle Database 12c Efficient Data Management and Performance Acceleration for every System," *SEPTEMBER 2014.*

[38] An Oracle White Paper, "High Availability Best Practices for Database Consolidation," *July 2014*

[39] An Oracle White Paper, "Analytical SQL in Oracle Database 12c," *February 2015.*

[40] An Oracle White Paper, "Transaction Guard with Oracle Database 12c," *MAY 2015.*

[41] N. Leavitt, "Will NoSQL databases live up to their promise?," *Computer, vol. 43, no. 2, pp. 12 –14, Feb. 2010.*

[42] Cory Nance, Travis Losser, Reenu Iype and Gary Harmon, "NOSQL VS RDBMS - WHY THERE IS ROOM FOR BOTH".

List of publication

Prabhjot kaur and Seema Bawa, "Comparative analysis of MongoDB, Oracle NoSQL and Oracle 12C," in 1st International Conference on Next Generation Computing Technologies (IEEE) (NGCT) , (2015). [Communicated]

Video Link

https://www.youtube.com/watch?v=DpXOvVC82AA&feature=youtu.be