A Comparative Study of Elasticsearch and CouchDB Document Oriented Databases

Sheffi Gupta
Deptt of Computer Science & Engg.
Thapar University, Patiala
sheffi.gupta@yahoo.com
+91-9463780232

Rinkle Rani
Deptt. of Computer Science & Engg.
Thapar University, Patiala
raggarwal@thapar.edu
+91-9915554748

***Abstract*—With the advent of large complex datasets, NOSQL databases have gained immense popularity for their efficiency to handle such datasets in comparison to relational databases. There are a number of NOSQL data stores for e.g. Mongo DB, Apache Couch DB etc. Operations in these data stores are executed quickly. In this paper we aim to get familiar with 2 most popular NoSQL databases: Elasticsearch and Apache CouchDB. This paper also aims to analyze the performance of Elasticsearch and CouchDB on image data sets. This analysis is based on the results carried out by instantiate, read, update and delete operations on both document-oriented stores and thus justifying how CouchDB is more efficient than Elasticsearch during insertion, updation and deletion operations but during selection operation Elasticsearch performs much better than CouchDB. The implementation has been done on LINUX platform**

Keywords—NoSQL; Elasticsearch; Apache CouchDB; Performance Analysis.

I. INTRODUCTION

In the database community, the term NoSQL is identified as "Not Only SQL". It is considered to be the store of various unstructured databases including key-value databases, column family databases and document databases [1]. The need of NoSQL arises from the incompatibility of relational databases to handle gigantic volume of data over internet and to cope with new trending technologies like cloud computing, big data etc. The very basic requirements of cloud computing technology are dynamic scalability and handling large data easily, but relational databases require more complex and costly hardware to fulfill these requirements because they are designed to work on single server itself to handle its ACID properties in an efficient way [2]. Several NoSQL databases have been designed so far like MongoDB [3], Oracle NoSQL[4], Cassandra[5], HBase[6] etc . These databases are designed by taking on the risk of not following the most important ACID properties of traditional databases. Instead, they support their own property called BASE (Basically Available, Soft state, Eventual consistency) in order to fulfill the need of scaling up the increasing traffic over Web and to make possible to run on large clusters which the traditional databases were unable to do.

A. Advantages of NoSQL Databases

The various advantages of NoSQL databases in comparison to traditional databases are listed as follows:

1) Horizontally scalable: NoSQL databases distributes the load evenly to each host and helps to improve the performance by horizontally scaling up the data.

*2) Schema-free:*In NoSQL, there is no need for prior establishment for storing data fields and no need to define any fixed structure to the data sets as was in the case of relational databases.

3) Low cost : In NoSQL databases, the cluster can be expanded easily by adding new node and these don't have expensive licensing costs as most of NoSQL databases are open source software and run on server cluster whose expansion is cheap.

4) Integrated Caching Facility: NoSQL databases cache data in system memory to raise their performance and increase data output.

In this paper, we have introduced the concept of NoSQL databases and have discussed two NoSQL databases under the category of document-oriented databases on the basis of performance testing.

This paper is organized as follows: Section II presents the overview and detailed architecture of Elasticsearch and CouchDB. Section III provides a performance evaluation of the techniques. Finally, Section IV presents the concluding remarks.

II. OVERVIEW OF DATABASE MODELS

The present NoSQL distributed storage systems have greatly impacted on how data is stored, represented and accessed in large infrastructure. Features like scalability and availability have made these systems prominent in industrial practice and research.

A. Elasticsearch

1) Introduction of Elasticsearch

Elasticsearch is a real time distributed social analytics engine mainly designed to organize the data in order to make

it easily accessible. It is built as an open source on top of Apache Lucene [7] which is a full text search engine.

It is a distributed document store, it stores all objects as JSON documents. These documents are indexed by default and are schema free, so that we don't have to define fields for data types before adding data.[8] Indices in Elasticsearch can be considered as databases in Relational database management system. Using this similarity from the SQL world, indices are collection of JSON documents as databases are collection of tables. It handles the fault tolerance by redundantly coping the data, and maintaining the high availability of data. It also provides the feature of multitenancy for querying multiple indices independently. Communication with Elasticsearch is done through HTTP REST API.

2) Overall Architecture of Elasticsearch

Fig.1 shows the relationship among Cluster, RestClient Node and the Elasticsearch service.

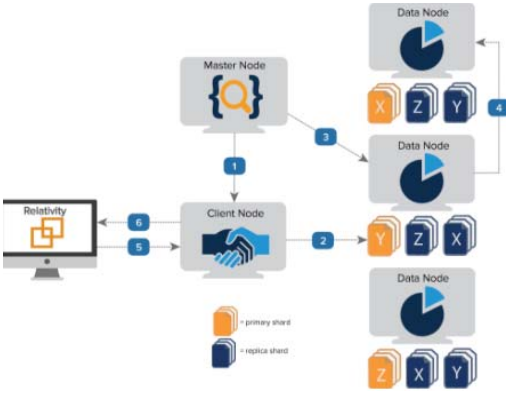

Figure 1: Elasticsearch Architecture

The Elasticsearch architecture is based on following concepts [9]:

- An index in Elasticsearch is created and gets split into one or more shards which resides on different nodes.
- Node is a running instance of Elasticsearch. When a node is started, it searches for a cluster to join with it.
- Cluster is a group of nodes. Each cluster is connected to a single master node which is chosen automatically.
- A master node handles Primary shard which is the first place where the document is stored when it is indexed. After indexing the document in primary shard, Replicas of the primary shard will get copy as well.
- Replica shard is just a copy of primary shard. So, it provides a back-up plan if primary shard goes down and also, replica shards has the ability to enhance the performance of Elasticsearch.

B. Apache CouchDB

1) Introduction of CouchDB

CouchDB is a NoSQL database designed to modularize, scale up and completely embrace the increasing demands of revolutionary world. It is a cross platform document oriented database created by Damien Katz presented in key-value maps using JavaScript Oriented Notation (JSON)[10]. With the increasing data, requirement of efficient data models is important. CouchDB is one such platform that handles a variety of structured, unstructured, image, audio, email data. The data stored in any format can be easily queried to retrieve relevant data. At a time, CouchDB has the capability to connect with large number of databases which further consists of large number of documents in JSON format.

Unlike Elasticsearch, it also includes features like Document level ACID semantics through the application of Multi-Version Concurrency Control [11] with eventual consistency and incremental MapReduce. Talking about similarity to Elasticsearch, CouchDB too possess some of its advantages like fault tolerance and handled failures by providing multi master replication, schema-less and Restful API.

2) Overall Architecture of CouchDB

Fig.2 shows relationship among CouchDB server, views and HTTP RestClient request.

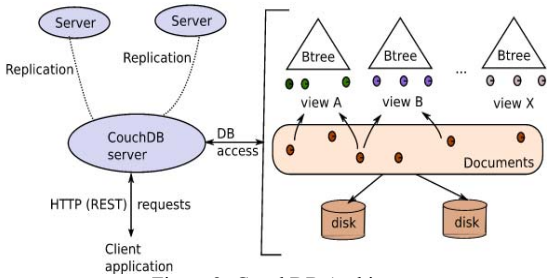

Figure 2: CouchDB Architecture

The CouchDB architecture is based on following concepts [12]:

- There exists two predefined functions in CouchDB: map function and reduce function, collectively called MapReduce.
- For query language, it uses JavaScript with MapReduce. These functions are the real reason for success of CouchDB in handling variations in unstructured data, independently indexing each document store and processing them in parallel by generating key-value pairs corresponding to each document.
- View results from the combination of map reduce function: list of key/value pairs.
- Views are assigned individual keys and are stored in B-trees to provide fast retrieval of rows by keys
- B-trees streams the rows according to key range and helps in fast searching of rows through these keys.

III. Results and Discussion

In our research, we have compared the performance of two databases: Elasticsearch and Apache CouchDB. The implementation has been done in LINUX. The tool used for implementation is Eclipse IDE. The coding has been done using JAVA. The Elasticsearch [13], CouchDB[14] and Maven java libraries have been used to make the connection

and perform all the required operations for the comparison. Rest Client API is used to hit the service of Elasticsearch and CouchDB. Insertion, Selection, Updation and Deletion operations have been performed and the results have been compared on the basis of time taken by both the databases to perform these operations. We have taken a sample dataset consisting of 20000 documents.

The results of executed queries are shown in this section. The tables below depict the time taken for data (in milliseconds) respectively. In all the Figures, X-axis represents the number of iterations and Y-axis represents time in milliseconds.

A. CouchDB insert vs Elasticsearch insert

Results are calculated on the basis of different number of attributes. The outputs of insert operation are as shown in Table I and Fig. 3.

TABLE II
INSERTION TIME

CouchDB(msec)	Elasticsearch(msec)
147	279
208	425
35	127
142	439
99	510
25	496
39	185
27	552
26	138
71	114

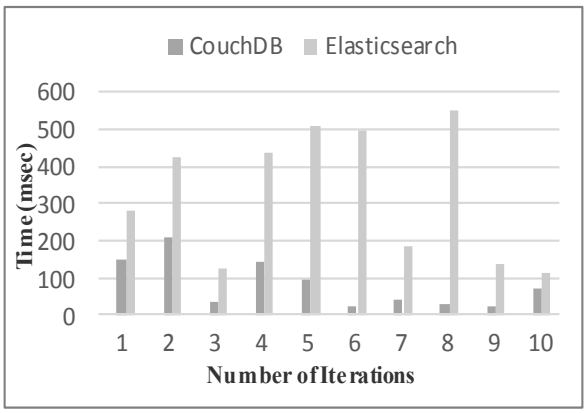

Figure 3: Time taken for Insertion

B. CouchDB select vs Elasticsearch select

Results are calculated on the basis of different number of attributes. The outputs of select operation are as shown in Table II and Fig. 4.

TABLE II
SELECTION TIME

CouchDB(msec)	Elasticsearch(msec)
240	275
170	115
151	95
89	24
154	51
225	84
183	46
85	36
116	39
110	60

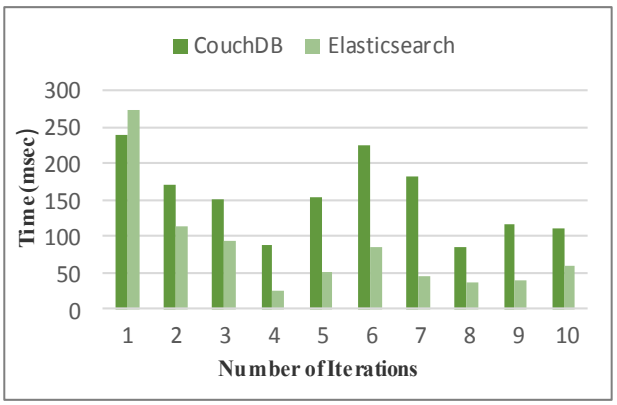

Figure 4: Time taken for Selection

C. CouchDB update vs Elasticsearch update

Results are calculated on the basis of different number of attributes. The outputs of update operation are as shown in Table III and Fig. 5.

TABLE III
UPDATION TIME

CouchDB(msec)	Elasticsearch(msec)
330	430
46	328
47	125
124	181
29	175
28	231
24	144
77	164
155	297
62	248

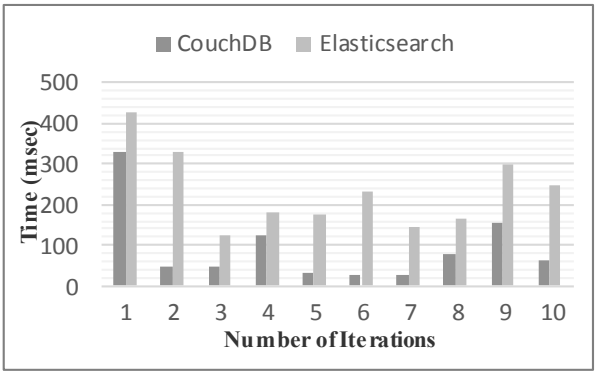

FIGURE 5: TIME TAKEN FOR UPDATION

D. *CouchDB delete vs Elasticsearch delete*

Results are calculated on the basis of different number of attributes. The outputs of delete operation are as shown in Table IV and Fig. 6.

TABLE IV
DELETION TIME

CouchDB (msec)	Elasticsearch (msec)
401	616
110	408
167	533
69	389
114	472
68	166
84	376
101	584
115	551
240	352

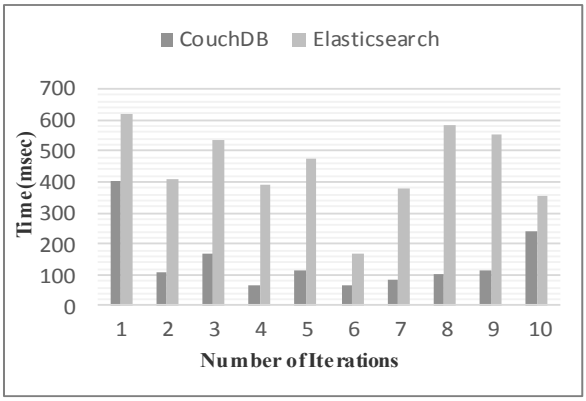

Figure 6: Time taken for Deletion

IV. CONCLUSION AND FUTURE WORK

There are many common features of distributed databases: Elasticsearch and CouchDB. On the same time there are certain differences also. Elasticsearch and CouchDB are NOSQL databases. Both the databases are used for managing excessively big data. For backing up of data, replication is done. In this paper we have implemented Elasticsearch and CouchDB on 20000 datasets. From the results, it has been observed that Elasticsearch takes much higher time than CouchDB in terms of insertion, deletion and updation of data, but for selection of data, Elasticsearch performs search more efficiently than the CouchDB.

As a future work we can integrate these databases with ETL (Extract Transform load) tool to analyze the live data such as data of social networking sites. We will then analyze the performance of these databases.

REFERENCES

[1] J. Pokorny, "NoSQL databases: a step to database scalability in web environment", Int J of Web Info Systems, vol. 9, no. 1, pp. 69-82, 2013.

[2] M. Allen, "Relational Databases Are Not Designed For Scale", relational-databases-scale, 2015.

[3] C. Gyorodi, R. Gyorodi, G. Pecherle and A. Olah, "A comparative study: MongoDB vs. MySQL", 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), 2015.

[4] An Oracle White Paper November 2012 Oracle NoSQL Database.

[5] G. Wang and J. Tang, "The NoSQL Principles and Basic Application of Cassandra Model", 2012 International Conference on Computer Science and Service System, 2012.

[6] V. Bhupathiraju and R. Ravuri, "The dawn of Big Data - Hbase", 2014 Conference on IT in Business, Industry and Government (CSIBIG), 2014.

[7] X. Li and Y. Wang, "Design and Implementation of an Indexing Method Based on Fields for Elasticsearch", 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2015.

[8] O. Kononenko, O. Baysal, R. Holmes and M. Godfrey, "Mining modern repositories with elasticsearch", Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014, 2014.

[9] "Advantages of Elastic Search", 3Pillar Global, 2016. [Online].Available: http://www.3pillarglobal.com/insights/advantages-of-elastic-search.

[10] S. Zhang, "Application of Document-Oriented NoSQL Database Technology in Web-based Software Project Documents Management System", 2013 IEEE Third International Conference on Information Science and Technology (ICIST), 2013.

[11] "Technical Overview - Couchdb Wiki", Wiki.apache.org, 2016. [Online]. Available: https://wiki.apache.org/couchdb/Technical%20Overview

[12] "Finding Your Data with Views", Guide.couchdb.org, 2016. [Online]. Available: http://guide.couchdb.org/draft/views.html.

[13] "Elasticsearch | Elastic", Elastic.co, 2016. [Online]. Available: https://www.elastic.co/products/elasticsearch

[14] "Welcome to The Apache Software Foundation!", Apache.org, 2016. [Online]. Available: http://apache.org/.