

Approaches used in efficient migration from Relational Database to NoSQL Database

Arati Koli

M.E. Student, Dept. of IT, Pimpri Cinchwad College of
Engineering, Pune, India
aratikoli77@gmail.com

Swati Shinde

Professor, Dept. of IT, Pimpri Cinchwad College of
Engineering, Pune, India
swaatii.shinde@gmail.com

Abstract—now a days, the emerging software applications producing the large volume of data rapidly. Hence, sufficient management of such a large data is major challenge. The traditional relational databases are inefficient to handle such a dynamic data which would be in any form (i.e. structural, semi-structured or hybrid). The ever increasing data have pushed the centralized databases like oracle, MySQL, SQL server to their limits. So, the NoSQL databases are the new generation databases for tackling such a problem with terabyte to petabyte of datasets. Hence, many leading companies such as facebook, Amazon, eBay, foursquare and many more using NoSQL databases. Some companies' still using relational databases and they wish to move towards new technologies, this motivated us to study the approaches used for migration of databases from relational database to NoSQL database. There are different techniques in the literature introduced by people for such migration. This paper surveys different approaches used for Migration of structured database to un-structured database.

Keywords—Relational database, MySQL, SQL server, Oracle, NoSQL, Migration

I. INTRODUCTION

In recent years the increasing volume of data (i.e Millions of queries per second) makes query processing, data retrieval speed slower and limits the traditional relational databases like MySQL, Oracle, SQL server. The relational database stores the data into tabular format and they follow specific structure, in the form of rows and columns. Ever increasing growth in data limits the traditional relational database and makes query processing, data retrieval speed slower. The new generation databases such as different NoSQL databases (i.e. Cassandra, MongoDB, Redies, Hbase, CouchDB) can tackle this problems and provide efficient way to handle such Big Un-Structured data which would be in the form of images, audio or video. The NoSQL is Database Management System is also referred as “Not-Only SQL”. The NoSQL make it easy for application to build, scale and deploy.

II. RELATIONAL VS NOSQL

The relational database and NoSQL database are different in their data-structure. The relational database is suitable only for

structural data whereas the NoSQL databases could have structural as well as un-structural data. The relational databases follow strict ACID rules wherein if business requires large volume of data which don't need to be up-to-date and not all node in the cluster follow strict ACID property then the NoSQL is better fit.

The NoSQL databases mainly developed to serve the purpose of performance, high availability, scalability availability, scalability (i.e. horizontal scaling).

That means the NoSQL database can allow any number of nodes to be added parallel to the distributed cluster. Wherein, the relational database allow vertical scaling in this we can add hardware at centre to serve more number of users this increases the complexity of system.

III. COMPARISON BETWEEN NOSQL DATABASES

Relational Database MySQL suffers when dealing with millions of query per second. Hbase is better at write speed where the Cassandra is best fit for read speed. The MongoDB is write-capable which provides the dynamic schema documents of many form can be easily added within the same collection.

Today's leading companies like Facebook, Google, Twitter, Amazon, eBay moving towards the new technologies, they have influenced the use of NoSQL database. The facebook introduced the Cassandra for their inbox search. Now, they are using Hbase for same purpose. The facebook instagram switched to Cassandra from Hbase as instagram is a facebook family now. Different NoSQL databases are introduced with their own specific feature and purpose, Example. The MongoDB is Document oriented data structure which stores the data in the form of documents and collections.

MongoDB isn't replacement for relational databases. It can serve some different needs. MongoDB works well if we want to write data a lot. Apache Cassandra is right choice for those who want high availability without affecting performance. Similarly, Hbase is data store uses columnar family. The NoSQL databases discussed above can process billions of

rows and columns per second which is not possible with relational databases.

The new generation software producing the large volume of data which can be in the form of Structured, Un-Structured or Semi- Structured. So, the inability of relational Databases to handle such a rapidly growing data motivates to study the approaches used transformation from relational database to NoSQL databases.

The data migration contains three steps ETL (Extract, Transform and Load)

- (1) Extract data from source database, (2) Transformation data, (3) Migration of data to the target database[6].

The first step extracts the data from homogeneous as well as heterogeneous database(i.e. Relational database) the next step is to transforms the data for storing in proper format for further querying. Third step migrates the data to the target databases (i.e. NoSQL database) the migration process contains many challenges; first one is large volume of data.

The second is representation of existing relationship have to be correct. The cost of adapting new database with existing applications needs to be addressed and there are different techniques in the literature introduced by people for migration of relational database to NoSQL database.

We can find many solutions in the literature that offers methods for migration from relational database to NoSQL database A paper presents NoSQLayer presented for migration of data and model from relational database (i.e. the MySQL) to NoSQL database (MongoDB NoSQL database) while keeping the semantic of the original database w.r.t manner the data is modelled and the way programmer write the source code to retrieve data from database [1]. A method presented by author which creates semi-automatically two logical levels over physical data [2].

A virtualization system which allow users to retrieve data from both sql and NoSQL database by using source specific API[3] A paper on schema conversion model for transforming SQL database to NoSQL proposed a method of graph transforming algorithm which can provide high performance of join query with nesting relevant tables, and for containing all required content of join query in a table by offering correctly nested sequence [4].The research in which the Methodology presented for migration from RDBMS to NoSQL databases. The author used MySQL as RDBMS and MongoDB as NoSQL database. Author discussed detailed analysis on NoSQL databases and its comparison with other NoSQL databases such as Cassandra, MongoDB, HBase etc in [7].

Now a days, the cloud computing area highly impacting the use of NoSQL databases, the emerging areas in cloud such as cloud computing threats such as categorization of threats in cloud computing [8] is hot topic in research area which

increasing the popularity of cloud computing and indirectly the cloud technologies influencing the use of NoSQL database

Many companies such as facebook which using the Cassandra and MongoDB to improve respective read and write performance, wherein redis NoSQL database is used by twitter and instagram, HBase database is used by yahoo and adobe. Some companies still using the relational databases and wants to use the NoSQL databases.

Fig 1 illustrates the conversion from relational database source that is oracle or MySQL to Destination NoSQL databases would be MongoDB, cassandra, couchdb or redis.

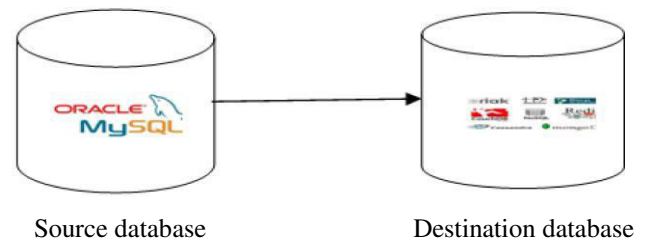


Fig.1.Migration from relational database to NoSQL

II. METHODS USED IN MIGRATION

Following are some approaches used for migration from Relational database to NoSQL:

The paper [1] presents the NoSQLayer framework. The framework consists of two modules: Data Migration Module and Data Mapping Module. In data migration module the elements for example, column, rows are identified from source database and then they are mapped automatically into NoSQL model. At last complete data will get migrated.

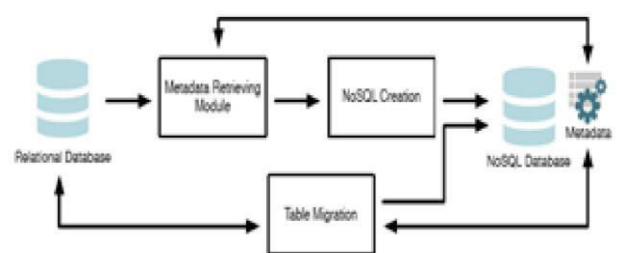


Fig.2. Migration Module Working Diagram

The migration module uses the Java Database MetaData API, which contains the methods and classes useful for retrieving metadata e.g. `getColumns()`, `getTables()`, `getMetaData()`. Next step is creating new schema for NoSQL database. the NoSQLayer supports MongoDB[1] and the MongoDB is document store. So, the records which are retrieved previously stored in document format in MongoDB database.

Last step of this module migrates data which is retrieved from relational database by using `select * from sql query` and insert into MongoDB documents.

Data mapping module: data-mapping module provides interface between user application and NoSQL source. It uses mediator which is open source MySQL parser which acts as an interface which converts user request into source specific format.

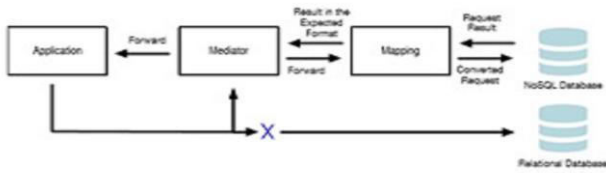


Fig.3. Mapping Module working Diagram

In [2] paper the nested relationships are handled efficiently by using mapping algorithm and also used index migration to support faster query search. The HBase database is used at backend in this approach. The virtualization architecture allow user to query and join from both databases i.e. from NoSQL and relational SQL using single SQL statement.

This architecture allows the applications to use jdbc API to interact with NoSQL as well as SQL databases while preserving the user way to query the database.

[3].

A. SQL Query Parser

- The parser checks syntax and also validates relations. It can be implemented using java CC supports standard SQL-92 syntax for SELECT, INSERT, UPDATE, and DELETE statements including inner and outer joins and GROUP BY, ORDER BY, and HAVING[3].
- The parser validates for correct SQL syntax and throws exceptions if syntax of user query is not correct.

B. SQL Query Translator

- The query translator takes the input tree which is generated by parser and validate fields by checking relational schema for database source.

C. Schema Generator

- In this step the NoSQL schema is generated here they considered NoSQL database as MongoDB so by considering data from all collection the schema is generated.

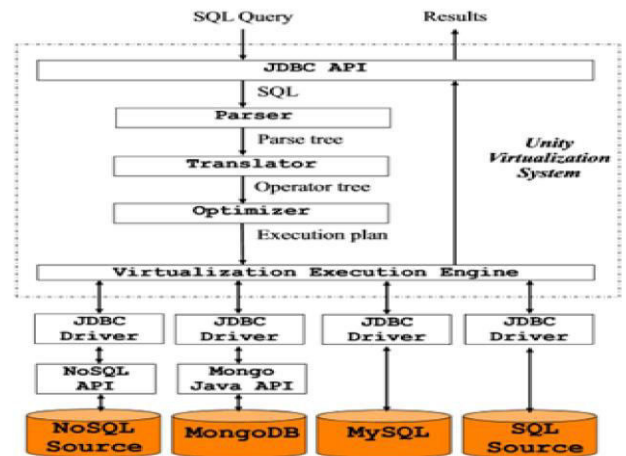


Fig. 4. Unity Architecture

D. Query Optimizer

- The main goal of optimizer is to perform all operations on the single database the optimizer also use different filters for faster query retrieval and execution.

E. Execution Engine

- The execution engine consist of implementation logic for different operators such as selection, different joins , projections, grouping etc.

F. MongoDB SQL/JDBC

- The authors constructed the MongoDB drivers for accepting the SQL queries.

G. MongoDB SQL/JDBC

- The Virtualization Engine contains different dialects database that perform mapping and in this step the regular syntax are identified.

The author introduced two logical levels over physical data for transformation from structural database into a Document store [4].

A. Physical level of data

The authors given example of university application which can store data in structural database which may contain logical objects such as Course, Teacher, Student, Lecture, Examination, Grade, and Attendance.

Fig. 4. illustrated the metadata of relational database. The relational database metadata is useful in conversion from relational database to NoSQL database.

B. First Logical level of data

The following Figure shows the metadata of relational database in this entity contains the table like structure which consist of view and table also the constraints like foreign keys and primary keys which play very important role in migration. The view and table is sub entity of the Table-like-structure and which having one to many relationship with attribute and primary key and foreign key are sub entities of constraint entity which has unique name, the unique key also comes under the PK_UK .

C. Second Logical level of data

The Table-like-structure (TLS) is taken as a basis for the document.

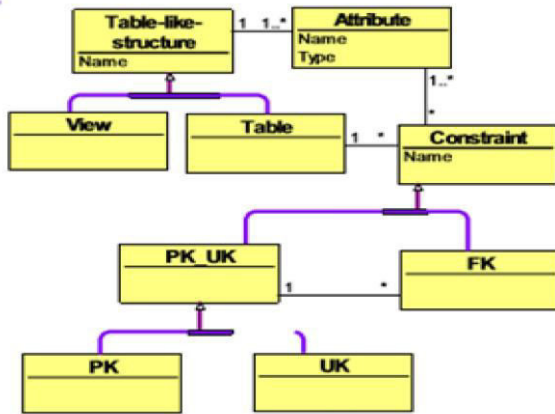


Fig. 5. Metadata of Relational Database

First, tree is created for Table-like-structures. Second, the new table-like-structures is added and others are eliminated. At last, authors chose attributes that are excluded from document.

D. Data Transformation

In this step the actual data is transformed and document is generated based on previously created template. All queries which are requested from application are executed and generated based on the template and metamodel transformation.

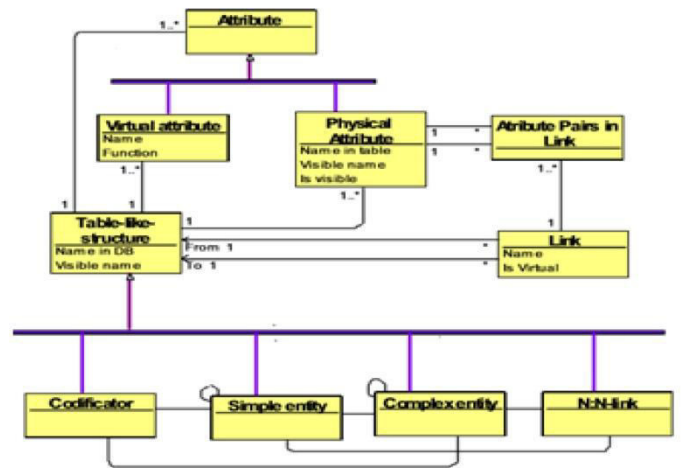


Fig. 6. Logical Levels of Data

The NoSQL databases not support the join queries. While migrating from relational database to NoSQL database the join queries brings poor performance. The authors proposed the graph based schema conversion model [5].

The paper [8] research focus is on the schema translation; more specifically set of rules are used to assist in translation process.

III. COMPARATIVE ANALYSIS OF MIGRATION APPROACHES

Table 1 shows the comparative analysis of the five approaches In paper [1] the NoSQLayer framework is used which preserves the semantics of original database and suitable to handle large volume of data. The authors [2] provided multiple nested approaches to improve the query performance and also contain migration of index for faster query execution. The unity virtualization architecture [3] allows us to use both the relational databases and NoSQL

TABLE I. COMPARATIVE ANALYSIS OF DIFFERENT APPROACHES USED IN MIGRATION

| Parameters | Paper[1] | Paper[2] | Paper[3] | Paper[4] | Paper[5] |
|-----------------|---|---|---|---|--|
| Approach/Method | NoSQLayer | Multiple Nested Schema | Integration and Virtualization | Structure Demoralization and Transformation | Graph based |
| Database used | MongoDB | HBase | MongoDB | Document-Oriented | MongoDB |
| Application | Web application | Coloumner database | UnityJDBC | Tool DigiBrowser | MySQL MongoDB migration system |
| Future scope | Will Extend NoSQLayer For other applications using different technologies | New algorithms will be used to identify table types and defining document structure | Parallelizing the Virtualization | New algorithms created for recognizing table types, defining document structure | Will use different techniques which are used to remove space problem |
| Advantages | Suitable to handle large volume of data | Improve query performance | Minimal overhead in SQL translation process | Easy to implement | Improve query performance |
| Disadvantages | Less efficient for small data | Consume more memory | Current system not supporting parallelism | - | Consume more memory |

databases. The migration from a Relational DB into a document-oriented DB and authors introduced two logical layers over physical data [4]. The graph based model helps in improving query performance by nesting related tables.

IV. COCLUSION

In this paper, we have studied different NoSQL databases and comparison of relational database to NoSQL databases. We found some approaches used for migration from relational database and compared different approaches used in migration from traditional relational database to NoSQL databases. The framework approach can be used when the data volume is large and need to preserve the semantics of relational database. The Hbase schema for migrating of relational database to NoSQL supports multiple nested and the Data migration features. Wherein the virtualization architecture enable users to access both the structural and un-structural database using single sql request. The schema conversion model nested the relevant tables together to improve the performance. Each approach has their own strengths and weakness. One or two approaches can be used based on the best fit to the application requirements. In future we will implement one or more approaches and compare its performance with relational database for handling big data.

V. REFERENCES

- [1] Leonardo Rocha, Fernando Vale, "A Framework for Migrating Relational Datasets to NoSQL", 2015 ICCS International Conference on Computational Science.
- [2] Gansen Zhao 1, Libo Li 1, 2, "Multiple Nested Schema of HBase for Migration from SQL", 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.
- [3] Ramon Lawrence, "Integration and Virtualization of Relational SQL and NoSQL Systems including MySQL and MongoDB", 2014 International Conference on Computational Science and Computational Intelligence.
- [4] Girts Karnitis and Guntis Arnicans, "Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation", 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN).
- [5] Gansen Zhao 1, Qiaoying Lin 1,2, "Schema Conversion Model of SQL Database to NoSQL", 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.
- [6] Mohamed Hanine, abdesadik bendarag, "data migration methodology from relational database to NoSQL databases", 2015 International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:9, No:12.
- [7] Nawal N. Moatassem, "A Study of Migrating Biological Data from relational database to NoSQL databases", 2015.
- [8] R. Ouanouki, D. Sygusch, "Demystifying NoSQL schema design using basic SQL Analogy", 2015.