GUI based System for Data Migration

NIKHILESH CHAUDHARI, ADITYA BHANDARE , UMAIR BAIG

BE(Computer Engineering),Sinhgad Academy of Engineering,Pune

Abstract - *These days we are looking at a transition in the ways of storage of data. Traditionally, the data was stored in a relational schema. The relational model is a tabular representation of data stored as records. The relational model was a very successful model for storage,analysis and representation of data. It is still predominant for databases where transactions happen. Nowadays NoSQL databases are becoming increasingly popular backends and are being widely accepted in software industry. Current technology frameworks such as Internet of Things (IoT),Hadoop framework have added to the increasing significance of NoSQL,also most of these NoSQL databases are open source backends like the MongoDB or CouchDB platform. They have various different styles and ways of how the data will be stored on the platform. For example, CouchDB is a document-based platform. The data in CouchDB is represented in the form of JSON document. JSON stands for JavaScript Object Notation. Its schema is of form key-value pair. Another database platform Apache Cassandra,which is a column oriented database, where the columns can be dynamically added or increased as per requirements of the user. Today, we face issues where data is not only migrated from SQL platform only, but also from a NoSQL platform to another NoSQL platform. No database platform is suitable for all type of requirements. Organisations select the database platform as per requirement of the clients. For the legacy data, to derive the benefits of the NoSQL platform, the respective relational data needs to be modelled to the newer platform schema. The NoSQL platform provides benefits like scalability, schema-less data storage, no complex joins, tuning, faster access to data etc. which the relational schema cannot provide . We have designed a GUI based open system to migrate the data from relational database platform to the NoSQL platform. The system provides facility to allow migration to multiple platforms as the client requires it. The client of the system is required to select which platform he needs the data to be migrated.Hence it automates the process of data migration.*

Keywords— JSON, ACID, BASE, NoSQL, SQL, DATA TRANSLATOR,HADOOP,COUCHDB,JAVASCRIPT.

1.INTRODUCTION

High usage of internet, social media websites and cloud computing has challenged the relational model of data storage in aspects of scalability. Heavy amount of data poses the question whether the relational model is still capable of handling itself or not. Rate at which data is being generated these days and also the amount of data generated everyday plays an important role as to how and where the data should be stored and managed. The data generated by social websites, cloud storage and mobile phones in unstructured and hence complex to handle. This huge and complex data cannot be handled by the relational models in an efficient manner but the cost associated for handling and maintaining the data with it is very high and consume resources in ad-hoc manner. The rigid schema nature of the relational schema is the major reason for such a high resource consumption.

SQL is an acronym for Structured Query Language .It is a language used for accessing and manipulating relational databases. It was the first commercial language for Edgar Codd's relational model. SQL is an ANSI standard, however different versions of SQL are available in the market. They follow the ACID properties. ACID stands for Atomicity, Consistency, Isolation and Durability. The relational platforms are dominant in systems where data consistency is the major requirement. It is still heavily used in systems involving transactions.

The NoSQL databases outperform SQL databases in most of the features except for data consistency. The NoSQL databases compromise data consistency in order to gain high availability, scalability and performance. But a large and dominant number of organizations do not require data consistency in their requirements.

The NoSQL databases dont have a particular schema. Hence,they are efficient in handling unstructured data. It is a really a myth whether NoSQL platform is only useful for websites where only unstructured data is to be stored. But this is not the reality. The websites and organizations using relational database for this purpose will soon feel the necessity and have the requirement of NoSQL databases as their clients and data starts growing. The major factors for the popularity of NoSQL databases are:

1. Low cost
2. Performance
3. Scalability.
4. Availability.
5. Schemaless

1.1 CAP THEOREM

The RDBMS systems support ACID properties whereas the NoSQL systems follows BASE properties.Base stands for Basically Available,Soft State,Eventually Consistent Horizontal scalability can be easily achieved by using NoSQL platform. Implementation of database plaform in a distributed environment makes it difficult to achieve the three major properties of databases like Consistency, Availability and Partition-Tolerance. This is the Brewster's theorem also known as the CAP theorem [2].

Consistency means that same data is visible at all the client nodes in the system, even while performing concurrent updates. Availability means that all database clients have access to some version of data in a cluster even if a node in the cluster fails. Partition tolerance means that the system keeps performing even when a node in a cluster fails to communicate with other nodes [2].

NoSQL database platform are classified as per the properties,data model they provide or give priority to. We here, are focused on NoSQL platforms namingly MongoDB, Cassandra and CouchDB. MongoDB provides consistent, partition tolerant systems (CP). Cassandra and CouchDB provides under available, partition tolerant systems (AP) and Consistency-Availability respectively.

1.2 Example - MODELING MONGODB

Zhao [1] has proposed about modeling the MongoDB with the relational model. He has also given the comparison of semantic expression of both models and analysing the feasibility of the migration of a relational database to MongoDB and its evaluation. As said, MongoDB is a document oriented database. The collection in MongoDB is equivalent to a table on SQL platform.The document model in MongoDB is the same with the relational model in the relational database systems. One row/record of the table is equivalent to a document in MongoDB, the keys are similar to fields in tables, and lastly, value in the table is equivalent to the value of the key.

The diagram given below defines how the modeling is done in the system. Thus, provides the scheme of migration from relational database to MongoDB platform.

CouchDB [3] is also a NoSQL document based platform by Apache software foundation. It also stores data in JSON format. The explanation and modeling given above about MongoDB migration is, hence, also applicable for CouchDB

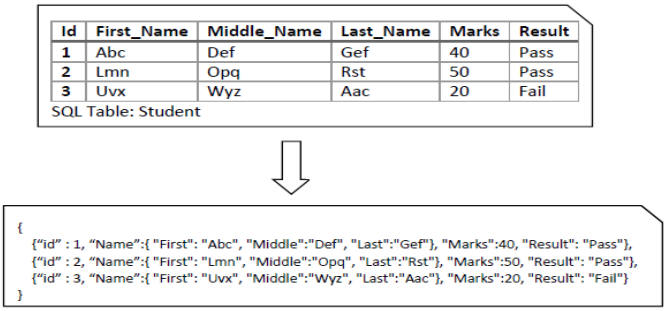


Figure 1. **Modeling MongoDB as SQL Model**

MODELING CASSANDRA

Cassandra [4] is another column-oriented database platform by Apache software foundation. It is a distributed platform and has no single point of failure. In Cassandra's terminology, node is a place where data is stored. Data center is a group of related nodes, while cluster is a component that contains one or more data centers. Cassandra works on Gossip protocol which facilitates the nodes to communicate with each other and detect any faults in nodes or in the cluster. Unlike relational tables where a column family's schema is fixed, Cassandra never forces individual rows to have all the columns.

NoSQL TO NoSQL MIGRATION

While the platform is selected on the basis of requirement, organizations have felt the need for migration of data from one NoSQL platform to another. This is, particularly, because any platform cannot provide consistency, availability and partition tolerance all at the same time. As the organizations feel of adding some new functionalities or features to their products, the database may require features that the present platform is unable to provide it. Collaborations among organizations many a times require data sharing among

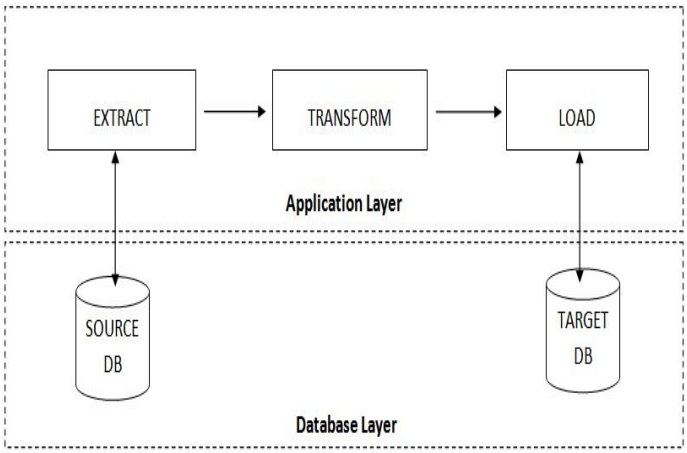
organizations, and hence, require migration of data. Clients many a time face problems, or many times are not happy with features of their working platform. For ex, many a time organization have migrated from their existing platform to another because of scalability problems. [7]

EXISTING SYSTEMS

Mongify [5] [11] is one of the existing systems which migrates data from SQL to NoSQL MongoDB platform. However,it allows migration only to MongoDB platform. It is based on self-defining routes. Tungsten Replicator is another such tool, but specifically used only for migrating to MongoDB platform. These systems work for only one specific platform and does not give the client any other options that client may possibly wish to migrate.

SYSTEM DESIGN

As we see that existing system, they allow migrating only to a specific NoSQL platform. They also do not provide GUI for the system and thus they are complex to handle for end users .Here,we provide options for organisations to migrate to possibly multiple platforms. Multi-threading is implemented in the system. Thread level parallelism is implemented in the system. One thread selects one table from the source platform and and performs its migration . Hence, concurrency is used to its core.



System Design

As we have seen the modeling of various schemas and migration feasibility above, a system where migration to multiple NoSQL database platform is possible. The system allows client to select which database he needs his data to migrate to. The system shows a complete monitoring of the migration process over the GUI. The monitoring shows the number of records in source database to be migrated, number of records migrated successfully, estimated time remaining and time of start for migration. The system is made dynamic so as to add more platform options to migrate.

Given above is the architecture of the system. It works essentially as a 2-tier database system .When the client gives the platforms of the source and target databases, the specified platforms are connected to the system by using the appropriate drivers. The client then gives the name of database to be migrated along with its authentication details. Now, as a connection is established between two platforms at the application layer of the system, a new database is created at the target platform. Now, the phases that take place are :

1. EXTRACTION OF DATA FROM SOURCE.
2. TRANSFORMATION OF DATA TO NoSQL FORMAT.
3. LOADING OF DATA AT DESTINATION.

These are exactly the same activities which are also carried out during data integration and data warehousing processes. Extraction phase of the thread copies one record at a time from the source database into the application layer and passes it to the transformation phase. The transformation phase is actually where the modeling of the data takes place, and creates a new data entry that can be loaded into the destination platform. Loading phase is one where the new record generated at the transformation phase is copied into the target database. This process is repeated till every record of the source database is migrated to the target database. The system should be open for adding up new platforms for migration.[6]

CONCLUSION

Data migration is currently an important practice of importing historical data to a new platform. It is a fairly time consuming process, if done manually. The number of NoSQL databases are increasing day by day [8]. Many of them are open source. Certainly, the tool allows migration to database platforms quickly and as per client organisations requirement. The automation of this practice certainly will be required to save time and deliver quickly to client's requirement. Also having seen the advent of Internet of Things and other technology frameworks, there is a high need of migration techniques. Since SQL and NoSQL, both databases have their own advantages, it is likely that there is a need to use both databases simultaneously. Hence, we have come up with a solution for migration of data which helps both the databases to exist in correlation with each other. However with NoSQL database platform now coming up with features of ACID properties, scalability and distributed nature. Probably, NoSQL will be the platform of choice for most of the clients and organization in the near future.

REFERENCES

[1] Schema Conversion Model of SQL Database to NoSQL, P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2014 Ninth International Conference on Gansen Zhao; Qiaoying Lin; Libo Li; Zijing Li IEEE 8-10 Nov. 2014.
[2] "Brewer's CAP Theorem", julianbrowne.com, Retrieved 02-Mar-2010
[3] Brewers CAP Theorem on distributed systems: 2010. Available: http://www.royans.net/arch/brewerscap-theorem-on-distributed-systems/comment-page-1/
[4] S. CouchDB the definitive guide: 2013.Available:http://guide.couchdb.org/draft/consistency.html#figure/3.
[5] Getting started with Cassandra: 2010. Available: http://nosql.mypopescu.com/post/573604395/tutorial-getting-started-withcassandra.
[6] ETL available at http://www.dataintegration.info/etl
[7] Mongify available at http://mongify.com/
[8] Mughees Thesis on Data Migration From Standard Sql To Nosql Muhammad Mughees http://hdl.handle.net/10388/ETD-2013-11-1342
[9] Making the Shift from Relational to NoSQL: 2013. e_Whitepaper Transitioning Relational to NoSQL. http://www.couchbase.com/
[10] NoSQL: 2013. Available: http://en.wikipedia.org/wiki/NoSQL
[11] Making Shift from Relational to NoSql http://www.couchbase.com/sites/default/files/uploads/all/whitepapers/Couchbase_Whitepaper_Transitioning_Relational_to_NoSQL.pdf
[12] MongoDb - The Definitive Guide at http://mongify.com/
[13] Modeling MongoDB with Relational Model - Gansen Zhao , Weichai Huang , Shunlin Liang , Yong Tang