Evaluation of Hadoop/Mapreduce Framework Migration Tools

Trust Odia, Sanjay Misra, Adewole Adewumi
Department of Computer and Information Sciences,
Covenant University,
Ota, Nigeria
{trust.odia, sanjay.misra, wole.adewumi@covenantuniversity.edu.ng}

***Abstract*— In distributed systems, database migration is not an easy task. Companies will encounter challenges moving data including legacy data to the big data platform. This paper reviews some tools for migrating from traditional databases to the big data platform and thus suggests a model, based on the review.**

***Keywords*—hadoop; mapreduce; CAP theorem; NoSQL**

I. Introduction

Business Intelligence deals with carrying out analysis on business information with analytical tools that spun profitable business actions and decision making. Most companies are business oriented. Intelligence in business is a step into the future. In business you want to know what, when, why and how of a customer. Business Predictions, decision making, Information gathering and analysis are key features in Business Intelligence. It involves detailed analytics. Big Data Analytics plays a major role in Business Intelligence. Business Intelligence has evolved from the use of decision support systems, data warehouses to Online Analytic processing (OLAP) and many more. As the years go by, organizations see the need to be more alert with business methods and ideas in order to meet customer needs, competition, make profit and meet business goal. Business Intelligence uses tools for gathering, extracting, processing and analyzing business information. The Business Intelligence technologies are known as Business Analytics.

Organizations have seen the need to employ analytics in business, others are yet to discover. For those companies that have the Business Intelligence Analytics mindset, there is more to it. There are numerous Business Intelligence Analytics technologies, but have focused on Big Data Technology that is based on Hadoop/MapReduce platform. Companies who are investing in Big Data technologies are at large advantage over other companies, because these big data technologies offer analytics that help business decisions and inferences. Legacy systems migration is a critical issue. Companies running Legacy data and applications installed decades ago will be skeptical about migrating. Migrating to a Big Data platform is not a day's job; imagine relocating to a new apartment. There is financial cost, compatibility, security, safety, new skilled IT persons to operate this appliance or training of old staffs, time etc. Most companies or organizations might not be interested because of loss of data, data duplication, time and financial cost most especially. Hadoop is one of the Big Data technologies, and it runs on a platform totally different from traditional database systems like MySQL. Migration is inevitable, for companies interested in Big Data technology.

The study reviews existing tools for migrating to the Hadoop/MapReduce framework. Migration to the Hadoop/MapReduce involves a lot of tools. These tools have different performance rate, cost effect, speed etc. Based on the tools that will be highlighted and reviewed, a model will be suggested that seems cost effective, with high performance rate. Consistency, Availability and partitioning (CAP theorem) as stated by Eric Brewer, was cited and explained by (Gilbert & Nancy, 2007) [15]. The study reviews existing models and the tools used, judges them based on CAP theorem and other criteria used for migrating to the Hadoop Framework.

The study shows that data replication is a way of migrating across databases with benefits. Data replication can be an extraordinary solution for developing applications, for improving performance and for better experience for users if applied in the right circumstances (Cristian, 2010) [1]. There are different types of database replication in relation to data. Transactional replication is more users (customer) beneficial. Data integration is also a criteria for migration and the informatica tool can be used. Data migration results from introduction of news systems, big data technology has been introduced, companies want to employ it in business. Enterprises build data warehouses and use business intelligence to identify and develop new opportunities Zandi, 2013 [2]. Apache Hadoop is an open source software that enables the distributed processing of large data sets across clusters of commodity servers. Hadoop operates on a MapReduce framework and has Hadoop Distributed File System. It is a component in big data analytics appliances like oracle exadata. Organizations tend to begin with minor test of a NoSQL database in their organization, which makes it possible to develop an understanding of the technology. Basically NoSQL databases are open source, which is cheap to use. The open source nature of these NoSQL databases makes it faster for development and more customer satisfaction to achieve from a larger percentage of users experience

II. LITERATURE REVIEW

A. Existing Models for Hadoop/Mapreduce

The existing models for big data technology are described below. Basically the Hadoop and Mapreduce models implemented in big data.

1) Parallel programming model

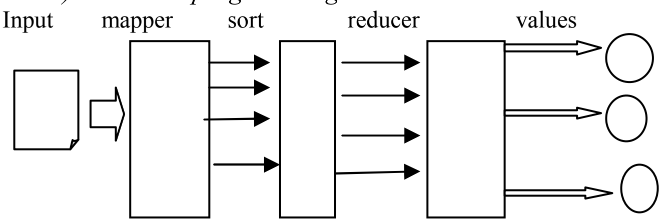

Fig. 1. Mapreduce dataflow *(taken from Dominique A. Heger, datanubes, DHTechnologies)*

According to (Jeffrey & Sanjay, 2008) [3] mapreduce is a programming model that handles processing of large dataset. Mapreduce is the framework of hadoop,executing its task with a parallel, distributed algorithm on a cluster. From the figure 1 there are two components of mapreduce; mapper and reducer. Mapper filters and sorts jobs while the reducer generates a summary of the output. The mapreduce model is based on parallel programming. Hadoop/mapreduce architecture can be viewed as a cluster computer architecture. (Silva & Buyya) [4] proposed a cluster computer architecture for parallel programming models. The diagram is shown below.

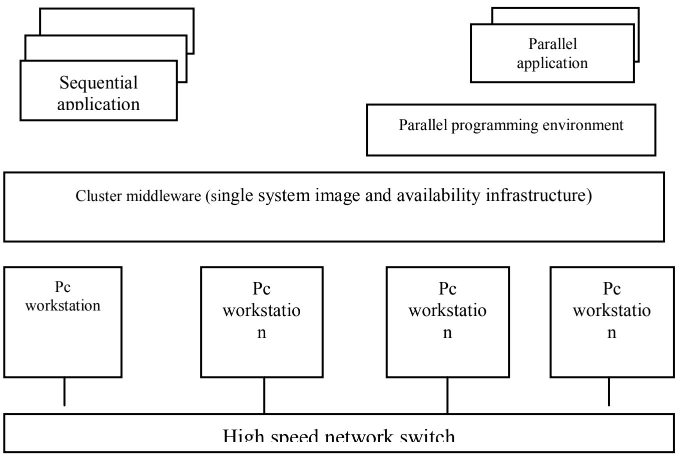

Fig. 2. Parallel programming models and paradigms, pg6

The Hadoop MapReduce framework is based on centralized master/slave architecture. The single master server since we are dealing with databases handles input jobs and passes to several slaves for further execution. Master server manages the map and reduces task assignments with the slaves. The master server sends instructions to the slaves to execute. Mapreduce construct are basically in the form of a directed acyclic graph. The following are Mapreduce algorithms; Sorting, Searching, Breadth First Search (BFS), PageRank, etc (Silva & Buyya) [4] discussed the master/ slave principle and the diagram below explains the hadoop/mapreduce master/slave architecture.

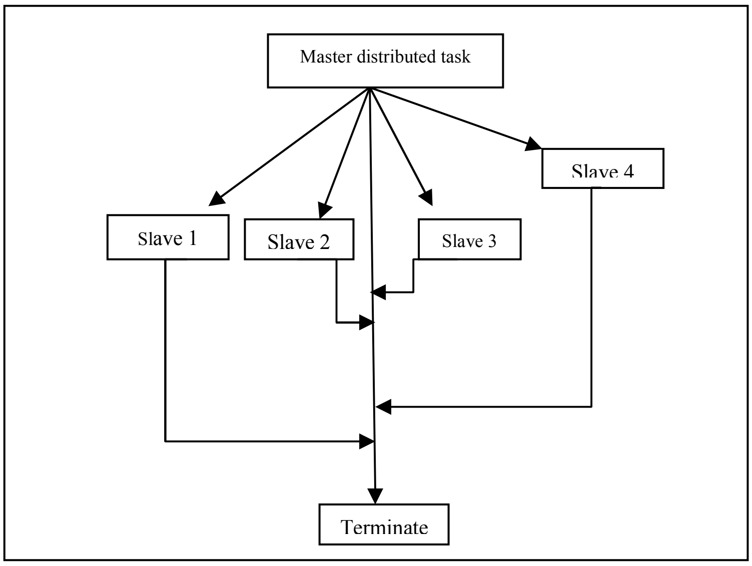

Fig. 3. A static master/slave structure source (*Silva & Buyya)*

a) Oracle big data architecture(Existing model example)

This is an example of an existing model. The oracle big data is one design model that is popular and widely used in corporate businesses. According to (Oracle, 2012) [5], Oracle has big data appliances already being deployed in businesses. A review on oracle's approach or architecture for a suggested migration model is carried out. These appliances have embedded Hadoop tools in them. The diagram below is information architecture for big data technology design.

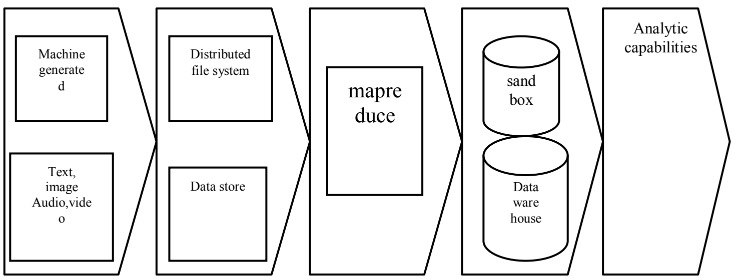

Fig.4. Big data information architecture *(taken from Oracle white paper in enterprise architecture, 2012, [5])*

2) Extract, Transform and Load (ETL) Model

The big data can be designed from the extract, transform and load (ETL) model. According to (Simitsis & Vassiliadis,

2014) [6] ETL involves extracting data from different dources, cleansing and customizing and inserting into data warehouses. There are ETL tools; for example data moving tools and many more, they can be used collectively in the big data design. The ETL model by (Simitsis &Vassiliadis, 2014) [6] is shown below and big data design is also based on this.

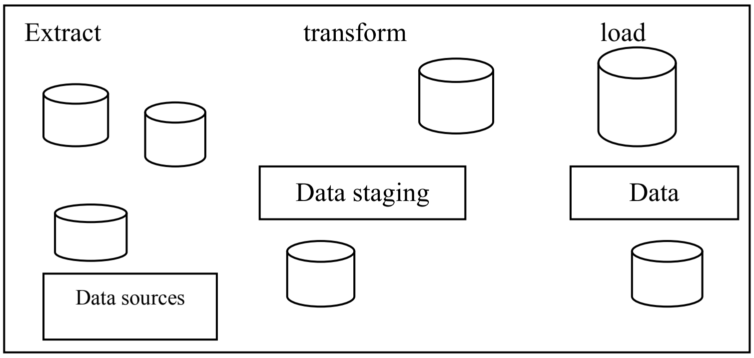

Fig. 5. extract, transform, load process model *(Simitsis & Vassiliadis, 2014).* A Methodology for the Conceptual Modeling of ETL processes

(Reynaud et al, 2009) [14] postulated a diagram for data extraction, transformation and integration guided by ontology and it is similar to that of figure 5.

According to (Trujillo & Lujan-Mora, 2003) [12] ETL processes are responsible for the extraction of data from heterogeneous data sources, their transformation and their loading into Data warehouses. This is the model that Hadoop/MapReduce framework is based upon also. Data is extracted from traditional databases regardless of the type of database. Structured Querying Language (SQL) is the most standard form for relational databases. There are different types of databases though; hierarchical, relational, extended Markup Language (XML) databases etc. there are several data migration technologies or models but ETL is preferred most, based on its advantages. It handles data migration, including terabyte scale datasets, multi pass data transformation, interoperability with data quality tools, data profiling and many to many data integration capabilities. Migrating to the Hadoop/MapReduce platform involves a lot of technologies cooperating together. Hadoop is an opensource software that has a file system called Hadoop Distributed File Sytem and carries out preprocessing of data through Mapreduce on jobs. Most big data appliances operate on Hadoop/MapReduce. Hadoop works with servers and data machines. ETL model consists of a workflow of activities to perform appropriate filtering, intermediate data staging, transformations and loading. The different categories of ETL and their examples will be discussed below.

a) Extract data (And its database examples)

It deals with filtering data from their source and sending to the ETL workflow for further processing. Data extraction is a part of data integration. (Grolinger, 2013) [13] Classified data into several categories and data can be extracted from these sources for transformation. They are;

- Web application data: examples are yahoo mail and goggle maps
- Analytical data:
- Unstructured data:
- Semi-structured data
- Transactional data: for example automated teller machine application or software
- Enterprise Data Warehouse data:
- Structured data: data can be extracted from relational databases like oracle, Microsoft SQL etc.

A relational database according to (Loganathan et.al, 2014) [16] is defined as any structured collection of data which stores information in an organized way such that the desired information can be quickly selected from the database. Relational database management systems are XML enabled databases; IBM database 2, Microsoft SQL server, Oracle database and PostgreSQL. An example of a relational database is XML database; it is an important relational database that has the mechanism for representing data in XML format. They are efficient in terms of cost and conversion of data. The documented-oriented database has some similarities with the XML database, so data in XML format is highly relevant to the NoSQL data stores. Document- oriented database is type of No Structured Query Language (NoSQL).

b) Transform data (Example of tools used)

It involves processing (data staging), analyzing and customizing extracted data. This is where Mapreduce programming comes in. Cloudera sqoopis a software tool used to import data from relational databases to Hadoop. It is an open source software that supports parallel import and export between Hadoop and various relational databases. Its default implementation is Java Data Base Connectivity (JDBC-based).The transformation phase involves data preprocessing and analysis. Mapreduce does the analysis of data. The use of the document oriented database in transformation is advantageous. There is no need for schema migrations which causes a lot of challenges in the relational databases world. Compared to key value stores, data can be evaluated more sophisticatedly terms of views calculation.

c) Load data

After extraction and transformation, data is loaded into a data warehouse. Oracle's exadata or teradata are examples of data warehouses.

B. Existing Tools and Migration Methods

The existing tools used for big data design and implementation are numerous. They are data stores used with hadoop as connectors or adapters. They include;

- Redix: is an open source, distributed database categorized as a key value data store.
- MemcachedDB: is a distributed memory caching system that operates with database driven websites for storing frequently accessed objects or data. It has its own key value database.
- Berkeley database: provides a database for key value data at a high performance.

- Voldemort: is a distributed database key value data store popular with Linkedin for high scalability storage.
- Riak: is an open source, fault tolerant key value NoSQL database based on principles from Amazon's Dynamo paper with heavy influence from Dr. Eric Brewer's CAP Theorem. Riak uses consistent hashing in a simple key value scheme (buckets) for data distribution.
- Cassandra: Apache Cassandra is an open source distributed database designed for large amounts of data across many commodity servers, with no single point of failure.
- Hbase: is an open source non-relational distributed database. It is developed as part of the Apache Hadoop project and runs on top of Hadoop providing big table solutions for Hadoop.
- Dynamo database (DB): Amazon DynamoDB is a fully managed proprietary NoSQL database.
- Amazon simple database: Amazon SimpleDB is a proprietary nonrelational data store that offloads the work of database administration. Developers' store and query data items via web services requests.
- Mongo database (DB): is an open source NoSQL cross platform document oriented database system. Jason like documents with dynamic schemas. MongoDB uses the BSON format.
- Couch database (DB): it is an open source NoSQL database that uses Jason to store data. It uses java script for query via Mapreduce. It carries out master-slave replication.
- Couchbase server: it is an open source NoSQL document oriented database, optimized for interactive applications. It provides low latency high throughput key value or document access.
- Neo4j: it is a native graph database.
- Hypergraph database (DB): it is an open source intelligent system oriented database. It has object oriented capabilities.
- Allegro: it is a graph based database, efficient in memory utilization and disk management.

Types of data migration methods

- Consolidation (many –to- one)
- Migration (one- to- one)
- Upgrade (one –to – one)
- Integration (many- to – many)

Focus is on migration and integration. The other methods are usually executed with integration models. Migration involves heterogeneous diversity of data models. Usually there is a movement (abandonment) of old database or legacy systems to a new platform. Integration is extremely heterogeneous and includes any subset of data. These two methods will be used together to achieve optimized migration to the Hadoop/MapReduce platform.

III. HADOOP/MAPREDUCE FRAMEWORK

It is based on centralized master/slave architecture as stated above in figure 3.Apache Hadoop is an open source software as explained above for storage and large scale processing of datasets on clusters of computing machines. A Mapreduce construct basically reflects a special form of a Directed Acyclic Graph (DAG) according to (Heger) [8].

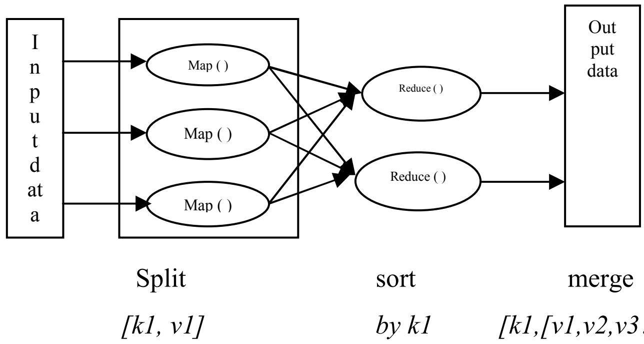

Fig. 6. Representation of a mapreduce DAG taken from (Heger) pg 5

Algorithm for map and reduce function according to (Heger) [8].

```
Map function             reduce function
map (input_record) {     reduce(key values){
…                        while(value has_next){
emit(k1,v1)              aggregate=merge(value.next)
…                        }
emit(k2,v2)              collect(key,aggregate)
…                        }
}
```

The Apache Hadoop framework is composed of the following modules:

- Hadoop Common- it contains libraries and utilities needed by other Hadoop modules.
- Hadoop Distributed File System (HDFS) is a distributed file system that stores data on commodity machines; provides high aggregate bandwidth across the cluster.
- Hadoop YARN - is a resource management platform responsible for managing computer resources in clusters and using them for scheduling of users' applications.
- Hadoop MapReduce -is a programming model for large scale data processing.

IV. REVIEW OF EXISTING NOSQL TOOLS

Below is a detailed description of existing NoSQL tools to buttress the explanation. (See Table 1 below on page 8)

A. No Structured Query Language (NoSQL)

According to (Loganathan et al. 2014) [16] the NoSQL architecture was designed to address the problems of massive horizontal scalability and agility. A direct opposite of relational databases, that is featured in the Hadoop/MapReduce framework. It is used in big data design. It is a database.

B. NoSQL Data Store Types

- Graph stores: These are NoSQL databases focused on the relationship between entities rather than entities themselves. They are popular in social media. Examples include Neo4j, Sones, Orient database, flock database etc
- Key value stores: these are schema-less NoSQL databases, they store values as key. Other NoSQL databases are built upon the principle of this kind of NoSQL database. Examples include Amazon simple database, Berkeley database etc
- Document databases: is a NoSQL database that stores data as documents as opposed to rows and columns. It uses JASON to store data. It is similar to key value database and schema-less. examples include couch database, mongo database, Terastore etc
- column family stores: This is not schema-less, they are semi Structured databases which means you need to specify a group of columns in these database. Examples are Apache Hbase, Cassandra, big-table etc.

NoSQL databases are categorized in data store type. keyvalue stores has only two columns (key and value).Complex information may be stored within the value columns. Document databases stores data all in a single document in (JavaScript Object Notation) JSON, extended Markup Language, or another format, which can nest values hierarchically.

Table 1 is evaluted based on database partitioning, replication and consistency control features. These requirements are relevant for effective migration to the Hadoop/MapReduce framework.

C. Motivation

Based on the review of existing NoSQL tools in section IV and Table 1 comparison of NoSQL databases, a motivation arose to propose a model. The suggested model was born out of a need to display the efficiency, performance of selected compared tools in relevant conditions. There is a need to review migration tools to Hadoop, it highlights the key features and efficiency of these tools. These tools are meant to be separate and independent for flexibility and user ability. Most legacy systems might be difficult to migrate to Oracle big data platform, because of interoperability issues, vendor-dependence and hardware. According to (Gilbert & Nancy, 2012) [15] the CAP theorem of Dr. Brewer states that it is impossible to fulfill consistency, availability and partition-tolerance for distributed computing, there must be a trade-off. In relation to this fact a combination of CAP theorem, replication and concurrency control factors are used for a proposed model

V. SUGGESTED ETL MODEL (ARCHITECTURE)

Below is a detailed explanation of the proposed model.

A. Introduction

The model is designed based on the comparison made among the migration tools in table 1. The comparison made among the migration tools was based on the CAP theorem explained by (Gilbert & Nancy, 2007) [15].

B. Components of the model

The model consists of reviewed migration tools which are;

MySQL: an open source popular relational database management system. It should be installed in a server operating system on a server preferably. It is used in this proposed model as an extraction source of data in relation to the ETL model in section II.A.2 and figure 5

OracleDB: it should be installed in a server operating system on a server most preferably. It is Oracle's database, also a proprietary relational structured database management system. It is used in this proposed model optionally in a situation where MySQL is not used. It is also an extraction source of data from the ETL model in section II.A.2 and figure 5.

Cloudera Sqoop: is a software tool used to import data from relational databases to Hadoop, as stated in II.A.2b. It is an open source software that supports parallel import and export between Hadoop and various relational databases. It is mandatory for this proposed model for cheap and effective function as similar to oracle big data machines. It serves as an intermediary between relational databases and the target (Hadoop/Mapreduce).

Voldemort: is a distributed data store that is designed as a key-value store used by LinkedIn for high-scalability storage. It is used in this proposed model because of good performance in partitioning, replication and consistency among other compared data store tools.

Hbase: is a non relational, distributed database that is open source, it is used in this proposed model because of good performance in partitioning, replication and consistency among other compared column family stores.

MongoDatabase: it is used in this proposed model because of good performance in partitioning, replication and consistency among other compared document stores.

HyperGraph: Database: it is used in this proposed model because of good performance in partitioning, replication and consistency among other compared graph stores

Hive: developed by Apache, installed on top of Hadoop, so it's mandatory to be in the proposed model. It is used for analysis of data transformed and exported for loading and summarization.

A. Operation of the model

The first compartment of tools represents the extraction tools, data is extracted from the relational databases that legacy data, applications reside using these tools. Companies are likely to have their data in relational databases like MySQL or OracleDB. Extraction symbolizes a connection (remotely or physically) to and from the relational databases and the server operating system that hadoop is resided in. The second compartment of tools consists of the transformation and loading tools. SQL server connector can be used (optionally) to connect sqoop. A connection is made from Cloudera Sqoop to Voldermort, Hbase, Mongo Database, hypergraph database by installing these data stores on your linux/osx operating system. Voldemort and hbase reside in hive when hive is installed in your server operating system. A connection on Linux preferably is made from either of them to hive. Hive should be installed on apache Hadoop. A connection is made from mongo and Hypergraph database to hive in your server operating system.

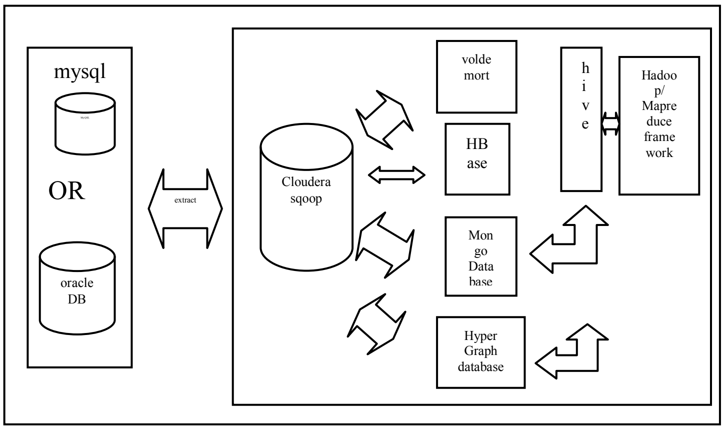

Fig. 7 Suggested model for migration

VII. DISCUSSION

The suggested model was based on the CAP theorem, data replication, data integration and consistency control. The four categories of data stores selected may be connected to Hadoop, for optimized performance of pre-processing, storage and analysis of data to Hadoop. These are the selected tools from the comparison with good results:

- Key-value stores-Voldemort
- Column Family- Hbase
- Document stores-Mongo database
- Graph database- Hyper Graph Database

Reasons are:

Hbase is a high performance data store used by facebook, oracle and other big data organizations. In terms of consistency Hadoop Distributed File System provides replication for HBase and thus handles most of the strong consistency guarantees that HBase needs for our usage.

Partitioning: in NoSQL involves the distribution of rows and columns in a database into segments on a server; horizontally and vertically respectively.

Replication: in NoSQL improves systems reliability, fault-tolerance and durability. It writes requests on nodes and propagates them to other nodes.

Consistency: based on CAP theorem, how data is seen among the server nodes after update operations.

The table below gives a basic description of the existing tools compared with the selected tools.

VIII. CONCLUSION

A review of migration tools is essential in big data technology; it gives a wide range of choices of the available tools to use. The focal point of the study is to harness the advantages of the tools reviewed. The suggested model has not being implemented, it is a suggestion. Future work should be to implement this model. Security was not used to judge the tools this is the major drawback in this study, so the tools might have security flaws when used in the suggested model or in other implementation.

REFERENCES

[1] M. M. Cristian, "Database replication," *database system journals,* vol. 1, no. 2, pp. 33-38, 2010.

[2] F. Zandi, "A concurrent group decision-making framework for selection of business interlligence-based data warehouse architecture," *Sage journals,* vol. 22, pp. 3-16, November 2013.

[3] D. Jeffrey and G. Sanjay, “Mapreduce: simplified data processing on large clusters,” vol. 51, Communications of the ACM, 2008, pp. 107-113.

[4] L. Silva and R. Buyya, “Parallel programming models and paradigms”.

[5] Oracle, "An oracle white paper in enterprise architecture," August 2012. [online]. available: http://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf.

[6] A. Simitsis and P. Vassiliadis, "A methodology for the conceptual modeling of etl processes," research gate, 2014.

[7] K. krishnan, “Data warehousing in the age of big data, Elsevier Inc,” 2013.

[8] D. A. Heger, "Dhtechnologies, datanubes," [online]. available: http://www.datanubes.com/mediac/HadoopArchPerfDHT.pdf.

[9] G. seth and L. Nancy A., "Perspectives on the cap theorem," *IEEE,* vol. 45, no. 2, pp. 30, 2012.

[10] M. Anirban, G. Kazuo and K. Masaru. [online]. available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.8648&rep=rep1&type=pdf.

[11] N. IACOB, "Data replication in distributed environments," 2010.

[12] J. Trujillo and S Lujan-Mora, "A uml based approach for modeling etl processes in data warehouses," vol. 2813, pp. 307-320, 2003.

[13] K. Grolinger, W. A. Hagashino, A. Tiwari and M. A. M. Captrez, "Data management in cloud environments: nosql and newsql data stores," *journal of cloud computing,* vol. 22, no. 2, pp. 1-24, 2013.

[14] C. Reynaud, N. Pernelle, M.C. Rousset, B. Saffar and F. Sais, "Data extraction, transformation and integration guided by an ontology," 2009.

[15] S. Gilbert and L. Nancy, "Brewer’s Conjecture and the feasibility of consistent, available and partition-tolerant web services," Cambridge, 2007.

[16] A. Loganathan, "A systematic approach to big data exploration of the hadoop framework," *International Journal of Information and computation technology,* vol. 4, no. 9, pp. 869-878, 2014.

Table 1 Comparison of different NoSQL databases features based on the CAP Theorem

Nosql data stores types	Partitioning	Replication	Consistency	Concurrency control
Key value stores Redix	Not available (planned for redis Cluster release).it Can be implemented By a client or a proxy	Master-slave, Asynchronous Replication	Eventual consistency. Strong consistency If slave replicas are Solely for failover.	Application Can implement Optimistic(using the Watch command) Or pessimistic concurrency
Memcached	Client's responsibility. Most clients support Consistent hashing	No replication Repcahed can be Added to Memcached for Replication	Strong consistency (single instance)	Application can Implement optimistic (using CAS with Version stamps)or Pessimistic Concurrency control
Berkeley database	Key-range partitioning And custom partitioning Functions. Not supported By the C-sharp and java APIs at this time	Master-slave	Configurable	Readers-writer locks
Voldemort	Consistent hashing	Masterless, Asynchronous Replication	Configurable, based On quorum read and write requests	Mvcc with vector
Riak	Consistent hashing	Masterless, Asynchronous Replication	Configurable, based On quorum read and Write requests	Mvcc with vector
Column family stores Cassandra	Consistent hashing And range partitioning	Masterless, Replication	Configurable, Based on quorum Read and write requests	Client-provided Time-stamps are Used to determine The most recent Update to a Column

Hbase	Range partitioning	Master-slave or Multi-master, Asynchronous Replication.	Strong consistency	Mvcc
Dynamo database	Consistent hashing	Three-way replication Across multiple Zones in a region. Synchronous replication	Configurable	Application can Implement Optimistic or Pessimistic control
Amazon simple database	Partitioning is achieved In the database design Stage by manually adding Additional domains (tables).can't query Across domains	Replicas within a chosen	Configurable	Application can Implement Optimistic Concurrency control By maintaining a Version number Attribute and by Performing a Conditional put/delete Based on the attribute Value.
Document stores Mongo database	Range partitioning based On shrd key	Master-slave, Asynchronous	Configurable	Readers-writer locks
Couchdatabase	Consistent hashing	Multi-master, Asynchronous Replication	Eventual consistency	Mvcc
Couchbase server	A hashing function Determines to which Bucket a document Belongs	Multi-master	Within a cluster: strong Consistency. Across Clusters: eventual Consistency	Application can Implement Optimistic or Pessimistic Concurrency Control
Graph database Neo4j	No partitioning (cache sharding only)	Master-slave	Eventual consistency	Write locks are Acquired on Nodes and Relationships Until committed
Hypergraph database	Graph parts can reside In different peer-to-peer nodes	Multi-master, Asynchronous Replication	Eventual consistency	Unclear how Locking is Implemented 100% read Concurrency. Near full write. Concurrency
Allegro	No partitioning	Master-slave	Eventual consistency	