



INTERNATIONAL JOURNAL OF
RESEARCH IN COMPUTER
APPLICATIONS AND ROBOTICS
ISSN 2320-7345

ETL FRAMEWORK DESIGN FOR NOSQL DATABASES IN DATAWARE HOUSING

¹Dumin Sahiet, ²PPG Dinesh Asanka

¹Pan Asia Banking Corporation PLC. Colombo, Sri Lanka. E-Mail: dumin.net@gmail.com

²Pearson Lanka (pvt) Ltd. Colombo, Sri Lanka. E-Mail: Dineshasanka@gmail.com

Abstract: - With the development of Web 2.0 significantly semi-structured, unstructured data has been generated. Typical RDBMS are lack of scaling and inefficient of handling Big data So the NoSQL Databases have the capabilities of handling those kind of data. Even though NoSQL databases are there, RDBMS are still remaining in the industry since some of features are not yet supported by the NoSQL databases, such as ACID properties. The typical Data warehouse consists of repository of data and those are non-volatile and extracting from different heterogeneous data sources. Since the central repository of data allows make strategic decisions of an organization. ETL is the process which used in extracting data, transform data to suit to data warehouse environment and load the data to the targeted data warehouse database. There were similar researches on for extracting data from NoSQL databases. However, those have limitations such as specific only for one vendor and only extraction is considered and transformation and loading to data warehouses with ETL characteristics are not addressed. Aim of this research on NoSQL ETL will be vital since currently data warehouses are consists of structured data and this will leads to store unstructured data in data warehouse and allows making strategic decisions, data analysis on top of it.

Keywords—*NoSQL, RDBMS, ETL, Data Ware House*

1. Introduction

With the development of Web 2.0 significantly, semi-structured, unstructured data has been generated. Typical RDBMS are lack of scaling and inefficient of handling Big data So the NoSQL Databases are have the capable of handling those kind of data it has been get popular among the IT community. Typical data warehouse are consists of structured data which allows to make strategic decisions. Extracting of these unstructured data will definitely allow making correct decisions since 80% of the data in an enterprise is unstructured data [1]. In high level Extract Transform and Loading process are responsible for extract the data from source data sources and the transformation phase data will be transformed, cleansed and homogenization are take place. Finally data will be loaded to the central data warehouse and its other parts such as DataMart and views. Typically data will be refreshed the data warehouse in the time period of the idle or low load in data warehouse (e.g. every night) and it has a specific time window to complete [2]. Literature review discusses the theoretical framework foundation of the investigation of the theories, concepts, design, implementation of the ETL between NoSQL Databases and data warehouse. The literature review also focus on ETL sub processes such as Extraction, Transformation, Loading which are main processes. Apart from those processes there are other processes of Data Cleansing, Optimization, security etc.

Oracle and Mongo DB is chosen as sample databases for implement this ETL Framework. Oracle is a dominant market player in the database industry [3]. MongoDB is an emerging NoSQL database follows under document databases. There were certain researches have done for bridging the gap between SQL and NoSQL databases. MSc Thesis "Middleware Layer for Replication between relational and NoSQL database" formalized a middleware for data replication between relational databases and NoSQL databases [4]. Thesis "Extracting Data from NoSQL Databases" [5] states the implementation of extraction data from NoSQL Databases but those researches do not have covered the ETL framework between No SQL and relational databases.

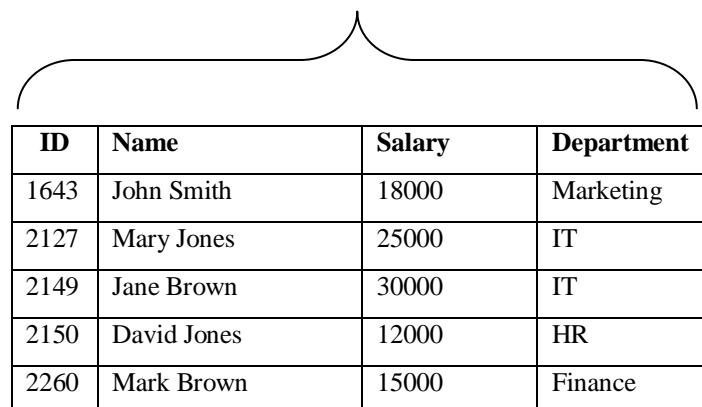
2. Literature Survey

Background study contains the literature review carried out in understanding the research goals.

- Data - Mainly Digital Data can be categorized as structured data and unstructured data [6].
- Structured Data - Structured data is best known as relational data, it has predefined set of columns and which belongs to a specific data types. In a relation it provides a well-defined mathematical structure with rules and standards for accessing and manipulating it. The relational model is a database model proposed and formulated based on first-order predicate logic by E.F Codd [7]. In the relational model of a database, all data is represented in terms of tuples, grouped into relations and concerns about following main three things.
 1. Data Structure (How the data is organized) In the relational data model, data is stored as relations (tables), each relations has a scheme (heading). The schema defines the relation's attributes (columns) and data takes form of tuples (rows).
 2. Data Integrity (What data is allowed) Data integrity of a relation will be controlled by using following.
 - Domains restrict the possible values a tuple can assign to each attribute.
 - Candidate and primary keys identifies the tuples in a relation
 - Foreign keys are link relations each other
 3. Data Manipulation (What operations can do with data) SQL (Structured Query Language) is a popular Data Manipulation Language (DML) that is used to retrieve and manipulate the data in relational Databases.

Figure 1 illustrates an example of relational model.

Schema is {ID, Name, Salary, Department}
Attributes are ID, Name, Salary, Department



ID	Name	Salary	Department
1643	John Smith	18000	Marketing
2127	Mary Jones	25000	IT
2149	Jane Brown	30000	IT
2150	David Jones	12000	HR
2260	Mark Brown	15000	Finance

Figure 1 Example of Relational Model

- Unstructured Data - The structure of this type of data is not clearly pre-defined and it is changing the frequently. Basically unstructured data is just any binary data E.g.: Documents, images, videos, blogs, and audios. With the development of Web2.0 has led to generation of large volumes of unstructured data [8].
- Data warehouse - The term Data Warehouse was coined by Bill Inmon in 1990, which he defined in the following way: "A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process" [9]. Data is extracted from multiple heterogeneous

systems and platforms and loaded into a centralized environment. Data warehouse mainly provides a framework for the top decision makers to make strategically decisions since data warehouse consists of historical data and integrated data. As the data warehouse is separated from operational databases users queries do not cause any impact on these operational systems. Data warehouse contents the data which are from different relational databases, multivalued databases, flat files, csv files, xml files, excel files etc. Typically, data warehouses are allowed to extract and store the structured data. A research carried out by IBM mentioned that 80% of the data in an enterprise is unstructured Data [1]. So extracting unstructured data and loading to data warehouse are crucial for decision making.

- NoSQL Databases - NoSQL is a term used in database management systems depart from the traditional relational database management systems. Though many would think that NoSQL means that doesn't support SQL actually it means "Not Only SQL" [10]. These databases have their own API, Libraries and preferred languages to interact with their data. The main characteristic of these DBs are that they are not relational and they are used best for unstructured and semi structured data or data that changes form and size frequently. The Research Paper "Scalable SQL and NoSQL Data Stores" has mentioned main six key features in NoSQL Databases [11].

1. Horizontally scaling of throughput among Servers
2. Replicate and partitioning data among multiple Servers
3. A Simple call level interface (in contrast to SQL Binding)
4. No ACID Support/ Supports for BASE
BASE stands for basically available, soft state and Eventual consistency. Basically available indicates that the system does guarantee availability, in terms of the CAP theorem. Soft state indicates that the state of the system may change over time, even without input. This is because of the eventual consistency model. Eventual consistency indicates that the system will become consistent over time, given that the system doesn't receive input during that time.
5. Efficient use of distributed indexes and RAM support for storage
6. Ability of adding new attributes for a records

The Thesis "Extracting Data from NoSQL Databases" [5] delivers valuable ideas and suggestions for when to choose NoSQL or RDBMS with Business and Engineering dimensions. It has stated NoSQL not a replacement and it will be a good option for a special type of databases where the RDBMS is not natively supported. That Thesis has mentioned since the NoSQL is not supported ACID properties but typical RDBMS is supported ACID which can be used by the required application. E.g.:- Financial Applications.

Since the NoSQL has capabilities of handling large volume of data "Bigtable: A Distributed Storage System for Structured Data" research paper states how the large volume of Google earth, web indexing and Google finance applications has used the NoSQL Technology [12]. There are different types of data store categories find in NoSQL Databases [11].

1. Key-value Stores: These systems store values and an index to find them, based on a programmer defined key. e.g: SimpleDB
2. Document Stores: These systems store documents. Indexing and simple query mechanism is provided. e.g.:MongoDB
3. Extensible Record Stores: These systems store extensible records that can be partitioned vertically and Horizontally across nodes. e.g.: Cassandra
4. Graph Stores: This kind of database is designed to store data those have relations represented by Graphs. Example of these types of data is transportation links, network topologies. E.g.: Neo4j

Brewer's Theorem which states in a distributed environment it's impossible to achieve the consistency, availability and partition tolerance properties simultaneously [13]. Some NoSQL databases are concerns on consistency and availability. Consistency has achieved using concept called eventual consistency that every change has to be propagated to entire database but some may nodes doesn't have the latest data a particular time [14]. Research paper "Scalable transactions in the cloud: Partitioning revisited" states they have developed a middleware on top of Elastrans Cloud database expanding the consistency to number of partitions and allows to increase the availability mentioned in Brewer's Theorem [15].

Research paper "MongoDB vs. Oracle - database comparison" [16] evaluates the performance of the RDBMS and NoSQL for CRUD operations. According to the results of the paper NoSQL database has high performance than the traditional RDBMS. But they have mentioned if you need a more complex database, with relations and

a fix structure, RDBMS is a reliable database and even though it moves slower since RDBMS allows to have multiple relations which have one-one, one-many, many-many relationships and allows to join relations and make complex queries. Research paper “MyStore: A High Available Distributed Storage System for Unstructured Data” [17] describes the new methodology of integrated NoSQL Database technologies called MyStore. Their objective is to integrate the advantages and uniqueness of different types of NoSQL in to one platform. As per the paper, MongoDB provides flexible query performance and Casandra provides a high availability and scalability.

ETL as Concept

Extract, Transform and Loading are the processes are the core processes running on background on data warehouse architecture. The data which can be extracted from heterogeneous sources such as Online Transaction Processing Systems (OLTP), text files, spread sheets, CSV files, XML files, JSON files, web pages, streaming data. ETL is identified and estimated as 80% of time in DW project has been spent for the ETL [18]. Using an ETL tool has following benefits.

- When there are many source systems to be integrated
- When source systems are in different formats
- When the processes need to be run repeatedly e.g. (Hourly, Daily, and Monthly)
- Auditing. ETL tools assist with auditing because of their repositories and their ability to preserve Versions.
- Visual flow and self-documentation, ETL tools provide graphically design the flows and logic of the data extraction process.
- Advance Data Transformation and Cleansing Functionalities, ETL tools consist of richest of data transformations tools and techniques.
- High Performance
Because of parallel processing and micro batch ETL packages works in high performance

A Journal paper “A Survey of Extract–Transform–Load Technology” [19] states following core process.

1. Extraction the appropriate data from data sources
2. Transport the data to special area called staging area to minimize the transformation load on production Systems
3. Transformation of data and computation of new values which is accepted by data warehouse
4. Cleansing and integrity check of data whether those are comply with business rules and database Constraints.
5. Loading the data to appropriate relations and reorganize / rebuild the indexes

Data Extratction

Extraction of the data is the hardest part of refreshment of data warehouse. The extraction software must not effect to source systems in runtime and off peak hours. Other thing is extraction software must install at source end with minimal effects to source configuration [19]. Extracting of data achieved in certain ways. Research Paper “Efficient Snapshot Differential Algorithms for Data Warehousing” [20] experimented a new algorithm called window algorithm and it is quite efficient and safe snapshot differential algorithm and experimental results have proved that this is a realistic assumption.

Data Transformation

Data transformation is natively built with SQL and relational algebra. But in ETL scenarios there are certain problems and which has been addressed by researches. Research Paper “One-to-many Data Transformations through Data Mappers” [21] address the problem of expressing one-to-many data transformations that frequently arise in ETL scenarios using the mapper operator. They have defined the data mapper operator as a computable function mapping the space of values of an Input schema to the space of values of an output schema. Identify the set of source data items of materialized view in a data warehouse leads to data lineage problem [22].

Research paper “Practical Lineage Tracing in Data Warehouses” [22] formalize the problem using a linear tracing algorithm with aggregation. This algorithm proposed several schemas to storing of auxiliary views that allows to drilldown through to exact source tuples.

Data exchange problem occurs when the source schema and target schemas are different. It requires the materialization of the result at the target subsequently so it doesn't need to refer back to source.

Research paper "Data Exchange: Getting to the Core" states number of problems occurs when the data exchange problems arises. They have come with solution called universal solution. Universal solutions have the good property of having exactly the data needed for the data exchange and can be computed in polynomial time via the chase.

Data cleaning is a sub process of Transformation stage. When consider about textual fields cleaning of it a challenge since these data consists lots of arbitrary values while data entry. Apart from that identification of duplication of data requires efficient algorithms. Data de-duplication is the process of identifying the duplicate data using different methods and eliminates applying pointers to the data instead of repeating data in multiple times. In relational database duplicate data are minimum and almost nonexistent due to the normalization.

The thesis "Data de-duplication in NoSQL databases" [14] led to the types of database key-value, which can be used as a preliminary step in the backup process. This allows easy integration with backup tools available, rather than having to develop new ones. The experimental results proved a higher DD ratio and a better performance of DDNSDB for DB files with more structural information available and higher percentage of duplicate data.

Data Loading

Data loading to data warehouse is typically take place in a periodical fashion. Initial loading of the data is carried at the very first time and subsequently data is loaded incrementally during every day. The new insertions, deletions, updates are identified at the extraction phase and transformed and cleansed. Research paper "Extraction transforming and Loading" states issues of data loading to data warehouse. DBMS typically supported with declarative way and simple SQL commands are not sufficient for open loop fetch technique. So data is inserted one by one and its extremely slow large volume of data. Other main issue state is administrators are defined indexes and materialized views for enhance the performance of queries. Because of that while loading data automatically incurs an overhead of maintaining the indexes and materialized view [23]. Conference Paper "Bulk Loading a Data Warehouse Built upon a UB-Tree" stated two bulk loading approaches for the UB Tree it's a multidimensional index structure. At the One for initial loading which creates a new UB Tree and for incremental loading which adds data to existing UB index. They have demonstrated the algorithm is minimizing the CPU and I/O cost for large datasets and it can be integrated with existing RDBMS as well [24].

Other Approaches in ETL

Execution time of an ETL job is really crucial since ETL job need to be completed in minimal period of time. The traditional optimization techniques are not applicable for ETL. "State-Space Optimization of ETL Workflows" Research Paper models the problem of optimization using state-safe problem. They have configured each ETL workflow as state and they have constructed the search space. The optimal state is chosen from cost model criteria are which they have given. The authors propose a method that produces states that are equivalent to the original one via transition from one state to another reconstructing using these ways [2].

- SWA- interchanges two activities in workflow
- FAC - replace homologous task in parallel flows into with equivalent task flow
- DIS-divide task of joint flows to clones to parallel flows
- MER/SPL – merge and split group activities

This architecture is shown in Figure 2.

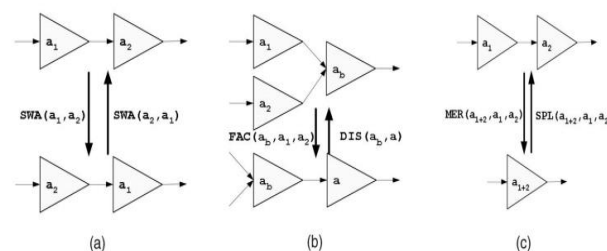


Figure 2- Examples of Transitions (a) Swap (b) Factorize and Distribute (c) Merge and Split [2].

Since the ETL process large amount of Data due the complexity for transforming there are situations can failures can occurred. But due to the time window of ETL processes there should be mechanisms for resumption of ETL processes. Conference paper "Efficient Resumption of Interrupted Warehouse Loads" [25] states an

algorithm called DR algorithm to resume the interrupted ETL workflows without restarting the workflow from beginning. The resumption algorithm consists of two phases.

- Design Phase - Constructs a workflow customized to execute the resumption of the original workflow.
- Resumption Phase –Based on previous characteristics and invoked in the event of failure. Multiple Times can be executed

As per the results of the research they have defined DR algorithm as an efficient lightweight recovery algorithm that can be used for complex distributed processing.

NoSQL and Other Related Works

Thesis “Extracting Data from NoSQL Databases” [5] states the implementation of extracting data from NoSQL databases to a platform called Spotfire. Author has used Cassandra and Neo4j as NoSQL databases for this research. Figure 3 will illustrate the overview of the architecture of Cassandra tool has implemented.

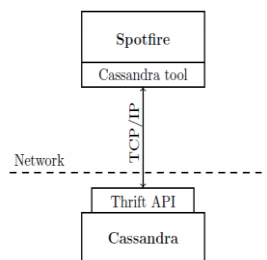


Figure 3 - Architectural overview of the Cassandra tool and its surrounding Entities [5].

Cassandra tool directly communicates via the Thrift API. Thrift allows for several different transport protocols. Typically, it is used on top of the TCP/IP stack using streaming sockets. For the Neo4J NoSQL database it was quite different and figure 4 illustrates its overview architecture.

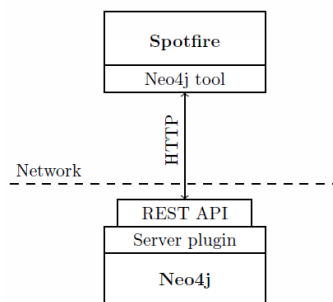


Figure 4 - Architectural Overview of Neo4J tool and its surrounding entities [5].

There is an additional server plugin written in Java and deployed as JAR. All the communication between Neo4J tool and Neo4J instance is done through REST API. The REST API expose to outside the functionalities implemented by server plugin. Even though this research is evaluate the results of time taken to load pages, no of optimal clusters. It doesn't address the CPU utilization, I/O utilization and memory and network utilizations. Basically this research focus on loading data SpotFire data source so it doesn't look on other RDBMS. This research which found at least minor relationship to build NoSQL ETL and covers the of extracting of NoSQL data. This research doesn't cover the Transformation, Loading stages of ETL processes.

The research paper “Data Migration between Document-Oriented and Relational Databases” [26] proposed a new approach to data migration between document oriented database and RDBMS using XML documents for storing data of document oriented databases. During the Data loading to RDBMS schema is automatically created from collection of XML documents. Even this research also concern about only for extracting of NoSQL data and didn't apply the characteristics of the ETL tools such as transformation, Cleansing, Parallel processing and bulk loading. So NoSQL ETL has lack of researches conducted till now.

3. Methodology

Data extracted from the NoSQL Databases and loaded in to the staging area. At the Staging area all the transformation and cleansing will be happened. This is because to reduce the processing the data in NoSQL database to reduce performance of issues and because of the transformation happened in staging area it reduces impact on data warehouse as well.

NoSQL Data Extraction

Data will extract by the ETL Framework using Application Programming Interfaces (API) which has produced by the No SQL databases. To enhance the performance Parallel extraction jobs will be executed.

Performance

ETL framework will be able to schedule the Jobs as per the convenient time. So the relevant jobs will be executed at predefined time. Apart from that ad-hoc execution of ETL jobs also possible. This is due to loading the data in nonpeak hours.

Parallel ETL jobs would be able to configure from the framework to reduce the time of loading of the data to data warehouse. Priority would be able to define in ETL framework by the user. So priority based algorithm will be used to execute the ETL sub processes.

Micro batch ETL jobs will be configured to load the data in specific intervals to improve the performance of data loading instead of one time data loading. This leads to low execution time of ETL jobs and data will be available quickly. Because of micro batch ETL jobs will reduce the interval time of refreshing the data in typical data warehouses.

Security

Security is a key area to address in every database. Methodology is focus on the transmitting of the data since each NoSQL and Data warehouses has their own data security mechanism to protect data. The Framework considers the transmission of aspects of the data as well.

Auditing

Audit mechanism will be implemented to log the all events triggered by the ETL framework on each ETL processes. ETL jobs auditing section will include the ETL jobs execution details such as start time, end time, status of the job, and exception details. This will allows monitoring the failed or success processes and would be able to re-run the failed process as when needed.

4. Experiment and Performance

Any ETL Framework has a performance impact in several aspects. Proposed ETL Framework performance will be evaluated against the following metrics.

- Process Utilization
- Memory Usage
- Time
- Network Bandwidth Utilization

Table 01 show the list of “Perfmon.exe” in Windows Operating System performance counters will be used for the testing with their definition and preferred values.

Object	Counter	Preferred Value	Description
Memory	Available Mbytes	>100 MB	Available MBytes is the amount of physical memory available to processes running on the computer, in Megabytes.
Process (oracle service %dbname %)	%Processor Time	< 80%	% Processor Time is the percentage of elapsed time that all of process threads used the processor to execution instructions. This is for Oracle process.
Process (mongod)	%Processor Time	< 80%	% Processor Time is the percentage of elapsed time that all of process threads used the processor to execution instructions. This is for MongoDB process.
Physical Disk/Logical Disk	Disk Read Bytes/sec	-	Used for determine bandwidth utilization for Read Ops (Extracting from MongoDB)
Physical Disk/Logical Disk	Disk Write Bytes/sec	-	Used to determine bandwidth utilization for Write Ops (Loading to Data warehouse)
Network Interface	Bytes Received/sec	< 80-90%	Bytes per second but counts only Received.(at Loading Area)
Network Interface	Bytes Sent/sec	< 80-90%	Bytes per second but counts only send.(at Extraction Area)

Table 1 – Performance Counters

5. Conclusion

Since the typical RDBMS are lack of scaling and inefficient of handling big data it's vital to extract transform and loading the unstructured data from NoSQL Databases to data warehouse and this will leads makes strategic decisions, data analysis on top of it.

There were similar researches on for extracting data from NoSQL databases. But those have limitations such as specific only for one vendor. Only Extraction is addressed and transformation and loading to data warehouses with ETL characteristics are not addressed.

The research question was to identify and implement an efficient framework for NoSQL Databases. In the Methodology, it is realized that it would take a lot more time and effort to create an enterprise level application for the NoSQL ETL Framework. However, the research gap has been addressed substantially with the Framework introduced in the research.

6. REFERENCES

- [1] Dirk deRoos, Chris Eaton, George Lapis, Paul Zikopoulos, and Tom Deutsch, Understanding Big Data : Analytics for Enterprise Class Hadoop and Streaming Data.: Mc-Graw Hill, 2012.
- [2] Panos Vassiliadis, Timos Sellis Alkis Simitsis, "State-Space Optimization of ETL Workflows," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 10, pp. 1404-1419, October 2005.
- [3] Donald Feinberg, Roxane Edjlali, Merv Adrian Mark A. Beyer, "Magic Quadrant for Data Warehouse Database Management Systems," Gartner, 2013.
- [4] Susantha Bathige, Dinesh Asanka, "Middleware Layer For Replication between Relational and Document Databases," Sri Lanka Institute of Information Technology, Colombo, Msc Thesis December 2012.
- [5] Nasholm Petter, "Extracting Data from NoSQL Databases - A Step towards Interactive Visual Analysis of NoSQL Data," Chalmers University of Technology, Göteborg, Sweden, Master of Science Thesis 2012.
- [6] Marcelle Kratochvil, "What is Unstructured Data?," in Managing Multimedia and Unstructured Data in the Oracle Database.: Packt Publishing, 2013, ch. 01.
- [7] E.F. Codd., "A Relational Model of Data for Large Shared Data Banks," in Communications of the ACM., 1970, pp. 377-387.
- [8] Shidong Huang, Lizhi Cai, Zhenyu Liu, and Yun Hu, "Non-structure Data Storage Technology-An Discussion," in IEEE/ACIS 11th International Conference on Computer and Information Science, 2012, pp. 482-487.
- [9] W. H. Inmon, "What is a Data Warehouse?," Prism Tech Topic, vol. 1, no. 1, 1995.

- [10] Eric Evans. (2013, August) NoSQL: What's in a name? [Online]. http://blog.sym-link.com/2009/10/30/nosql_whats_in_a_name.html
- [11] Rick Cattell, "Scalable SQL and NoSQL Data Stores," 2011.
- [12] Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Fay Chang, "Bigtable: A Distributed Storage System for Structured Data," Google Inc.,
- [13] N. Lynch S. Gilbert, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," SIGACT News, vol. 33, pp. 51-59, June 2002.
- [14] W. Vogels, "Eventually consistent," Commun ACM, vol. 52, pp. 40-44, January 2009.
- [15] J. Armendáriz-Iñigo, M. Ruiz-Fuertes, R. Oliveira F. Maia, "Scalable transactions in the cloud: Partitioning revisited," International Conference on on the Move to Meaningful Internet Systems, pp. 785-797, 2010.
- [16] Florin Radulescu, Laura Ioana Agapin Alexandru Boicea, "MongoDB vs Oracle - database comparison," in Third International Conference on Emerging Intelligent Data and Web Technologies, Bucharest, Romania, 2012, pp. 330-335.
- [17] Lei Zhang, Weizhong Qiang, Hai Jin, Yaqiong Peng Wenbin Jiang, "MyStore: A High Available Distributed Storage System for Unstructured Data," in IEEE 14th International Conference on High Performance Computing and Communications, 2012, pp. 233-240.
- [18] Christian Thomsen and Torben Bach Pedersen, "A Powerful Programming Framework for Extract-Transform-Load Programmers," ACM, November 2009.
- [19] Panos Vassiliadis, "A Survey of Extract-Transform-Load Technology," International Journal of Data Warehousing & Mining, pp. 1-27, July-September 2009.
- [20] W., Garcia-Molina, H. Labio, "Efficient Snapshot Differential Algorithms for Data Warehousing," VLDB, pp. 63-74, 1996.
- [21] Helena Galhardas, Antonia Lopes, Joao Pereira Paulo Carreira, "One-to-many Data Transformations through Data Mappers," University of Lisbon, Lisbon,.
- [22] Jennifer Widom Yingwei Cui, "Practical Lineage Tracing in Data Warehouses," Computer Science Department, Stanford University,.
- [23] Alkis Simitsis Panos Vassiliadis, "Extraction Loading and Transform," Encyclopedia of Database, no. Springer, 2009.
- [24] Akihiko Kawakami, Volker Mark, Rudolf Bayer, Shuichi Osaki Robert Fenk, "Bulk Loading a Data Warehouse built upon a UB Tree," in International Database Engineering and Applications Symposium (IDEAS 2000), Yokohama Japan, 2000, pp. 179-187.
- [25] Janet L. Wienerz, Hector Garcia-Molina, Vlad Gorelik Wilburt Juan Labioy, "Efficient Resumption of Interrupted Warehouse Loads," in ACM SIGMOD International Conference on Management of Data , Dallas, Texas, USA, 2000, pp. 46-57.
- [26] Cyril Klimes Bogdan Walek, "Data Migration between Document-Oriented and Relational Databases," in World Academy of Science, Engineering and Technology , 2012.
- [27] Phokin G. Kolaitis, Lucian Popa Ronald Fagin, "Data Exchange: Getting to the Core," IBM Almaden Research Center, 2005.
- [28] Nicoleta C. Brad, "Data De-Duplication IN NoSQL Databases," University of Saskatchewan, University of Saskatchewan, Msc Thesis 2012.