BigDimETL: ETL for multidimensional Big Data

Hana Mallek, Faiza Ghozzi, Olivier Teste, and Faiez Gargouri

MIR@CL Laboratory, University of Sfax, BP 242, 3021 Sfax, Tunisia,
Université de Toulouse, IRIT 5505, 118 Route de Narbonne, 31062 Toulouse, France
{mallekhana,jedidi.faiza,faiez.gargouri}@gmail.com
Olivier.Teste@irit.fr

Abstract. With the broad range of data available on the World Wide Web and the increasing use of social media such as Facebook, Twitter, YouTube, etc. a "Big Data" notion has emerged. This latter has become an important aspect in nowadays business since it is full of important knowledge that is crucial for effective decision making. However, this kind of data brings with it new problems and challenges for the decisional support system (DSS) that must be addressed. In this paper, we propose a new approach called BigDimETL (Big Dimensional ETL) that deals with ETL (Extract-Transform-Load) development. Our approach focus on integrating Big Data taking into account the MultiDimensional Structure (MDS) through MapReduce paradigm.

Keywords: ETL, Data Warehouse, BigData, Twitter, MapReduce, Parallel processing, Multidimensional structure

1 Introduction

The Decision Support System (DSS) [11] delivers useful data for supporting business decisions. It maintains permanently generated data in order to be stored and collected in the Data warehouse (DW). This integration process referred as ETL (Extract-Transform-Load), which is responsible for integrating data at regular intervals such as daily, weekly or monthly. Nowadays, the DSS is facing a huge volume of data due to the exponential growth of new and heterogeneous data sources such as web, social networks and ubiquitous applications (Smart-phones, GPS, etc.). According to IBM[1], 2.5 trillion gigabytes of data are generated every day. In fact, this huge amount of data introduces the concept of "Big Data", which is often defined by the 3Vs characteristics, i.e. Volume (quantity of data), velocity (the speed of the generating, capturing and sharing of data), and variety (range of data types and sources). This data evolution creates big challenges in the decisional domain in order to handle, store, transfer and analyze it, at the right time. Big data requires several set of new integration technologies such as Hadoop [22] MapReduce [6], Hive [18], etc. These technologies present a new opportunity to process and analyze data for DSS. Therefore, supporting new requirements and technologies of big data is a challenging issue for the conventional

[1] http://www.ibmbigdatahub.com/infographic/four-vs-big-data

ETL tools [2]. In this paper, we propose a new approach called BigDimETL for ETL process in Big Data context. It aims to adapt the Extraction and the Transformation phases of ETL processes with MapReduce (MR) technology, in order to minimize time consuming through the parallel data processing. Although, MR technology treats a huge amount of data, without taking into account the MultiDimensional Structure (MDS) of the DW. Thus, our solution is based on dividing the input data vertically according to the MDS metadata. The MDS is considered as a high level DW/ETL specific constructs [13] and it is dedicated to Online Analytical Processing (OLAP) and Business Intelligence applications. Consequently, decision makers are not concerned to learn a new language to analyze big data any more. As mentioned previously, our approach focuses on extraction and transformation phases. Accordingly, in the extraction phase, a division method is used in order to minimize the overload into the transformation and loading phases. While in the transformation phase, we choose to focus on the "select operation" since it is one of the most frequently used query. It was picked from different primordial operations like projection, conversion, etc. This query will be used in treating and filtering data steps. To validate our approach, we carried out a series of theoretical experiment which validate the feasability of our proposed approach. The rest of this paper is structured as follows: A state of the art of studies dealing with ETL processes, for big data and data warehouse, is presented in section 2. In section 3, we present our platform as well as several key concepts. In section 4 and 5, algorithms of our proposed extraction and transformation approaches are presented. Finally, in section 6, we describe our contribution and different analysis of the proposed algorithms complexity for testing the feasibility our approach.

2 Related work

On the DSS field, various studies consider ETL modeling processes as an efficient solution to acheive data warehousing projects. The objective of modeling is to reduce the costs of DW implementation, and produce useful information in multidimensional structure through OLAP cubes [16]. Some researchers have focused on business data sources to model ETL processes. These processes are modeled as new specific tasks which are implemented as a framework called ARCTOS [21] and EMD [9]. However, working with new conceptual constructions proves to be significant, but the non-standardization is still a limit since it is a vital asset in terms of modeling. Therefore, the standardization appeared with [19] which presents ETL processes as activities using Activity Diagram. In same context, [8] uses BPMN language to design ETL process. With the works of [7] and [4] a semantic design of ETL processes is presented, based on RDF (Resource Description Framework) graph in ordet to generate semantic DW. In other hand, several studies focused on the development of ETL functionalities model which are considered as important sub-processes of ETL. For example, [17] designs CDC functionality (Changing Data Capture) as colored network (CPN) describing and presenting the behavior of ETL processes, and [15] presents SCD (slowly

Changing Dimension) and SKP (Surrogate key Pipeline) through a coordination Reo language to improve the synchronization between the entities that interact with each other. Today, Information System researchers are facing a big explosion of web data, especially with social media (Facebook, Twitter, Youtube,). Hence, Traditional Database Systems (TDS) cannot support this big volume of data. Indeed, computing industry starts looking at new options, namely parallel processing to provide a more economical solution. In this case, other lines of works deal with data integration using new technologies introduced to handle big data context, such as Hadoop, MapReduce [6], Hive, etc. Consequently, it is very crucial to use these technologies to manage big amount of data. In fact, previous works have shown the power and the gain in time for processing and storing large volumes of data using Hadoop framework. Thus, [14] proposed a web service called TAREEG that collect geographic data using MR paradigm to represent real spatial data. However, few studies have focused on the exploitation of DSS with the MR paradigm. Moreover, the amount of data pushes the experts to adapt ETL processes in order to support Big Data requirements. It raises several problems related to Big Data 3Vs characteristics, which must be considered while modeling ETL process. In fact, authors in [2] and [13] use MR paradigm to add the parallelism concept to these processes in a physical level of integration. Similarly we find [14] which uses MR and Hive system to define a Framework called CloudETL that supports the representation of DW Star schema. The multidimensional structure is moreover, treated in the transformation phase. [2] proposed a Framework called P-ETL which presents a parallel ETL through the MR paradigm. This work employ the parallelism strategy while integrating data, without considering the multidimensional structure, which is considered as the paramount stage for further analysis operations. [1] uses Hive and Hadoop to support DW in cloud computing environment and to build OLAP cubes. In fact, the multidimensional structure is designed to solve complex queries in real time [11]. According to [3], the performance of OLAP queries can be advanced if the data warehouse is perfectly selected and integrated. Besides, several works try to adopt querying OLAP in Big Data context [5]. In fact, these previous studies are very interesting in the data warehousing context. Although, these works share some similarities with ours, the pretreatement phase, including the extraction and the partitioning, aren't covered. As a conclusion, our goal is to reduce the cost of DW implementation and produce relevant information for decision support. Among the possible solutions, the MR paradigm proves a powerful solution for parallel processing of massive data. Hence, we propose to handle the extraction and the transformation phase according to a multidimensional structure from the first step using vertical partitioning.

3 BigDimETL architecture overview

Figure 1 presents our proposed BigDimETL architecture which adapts the three main steps of typical ETL processes to MapReduce paradigm. Thus, our main concepts are defined in the following:

- Multidimensional structure composed of a list of dimensions of analyses (D), where, $D = \{D_1, D_2, ...\}$ and fact (F) where, $F = \{F_1, F_2, ...\}$ to present the subject of analysis. Each dimension D_i is composed of several attributes $Att = \{att_1, att_2...att_k\}$. Each fact composed of numerical measures $M = \{m_1, m_2, ..\}$. A fact is linked to associated dimension.
- MapReduce is a framework developed by Google described as a programming model used for a parallel processing of massive data sets. The input data must be divided in several partitions according to the default size (64MB). Each split is browsed separately by several jobs to perform transformation tasks. Each job has to be specified by the means of two functions: The first function Map takes key/value pairs from splits as input. The resulted key/value pairs play the role of input value for the second function Reduce.
- ETL process is described as three principal phases (Extraction, Transformation, Loading), considered as the dynamic and responsible part dealing with the workflow synchronization and the transmission of data to be loaded into the DW [20]. It is a big challenge, for this typical process to deal with a big amount of data, and generate multidimensional data into the DW. According to [20], ETL processes present several critical operations needed for the transformation phase in order to model ETL. In our case, we will classify these operations in two main classes according to the type of operation: Elementary Operations (EO) or Complementary Operations(CO). While EO requires as input one operand (Select, Project, Conversion operations), the CO requires two or more operands as input. To remap our operations according to map and reduce functions while conserving key/value structure, we find that EOs can be running as map or reduce functions and COs need both functions of MapReduce. In this paper, we focus on the "select operation" for the BigDimETL that will be executed as MapOnly function.

BigDimETL has three important phases described as follow: The extraction phase takes the case of capturing the data and then do the correspondence between the MDS metadata and the input data. In this phase we need to split our input data into sub logical parts in order to work in a distributed and parallel aspect. We also need to ensure the vertical partitioning of column oriented data according to Facts and Dimensions described in the XML metadata. Hence, each separate part represents a group of columns related to a specific dimensions and facts. This phase faces two main challenges: i) the large volume of the dataset and ii) the time consumed when data is extracted. Details of the data extraction phase are presented in section 4. Then, the data are transferred to the next phase which is the transformation phase. This step covers several operations of data processing such as Selection, projection, conversion, etc... Thus, the transformation phase corresponds to Map and Reduce functions which cover all types of the transformation operations of ETL processes. This phase will present EO and CO. Finally, the loading phase feeds the processed data into the final database following multidimensional structure where the result of Reduce or Map phase is loaded.

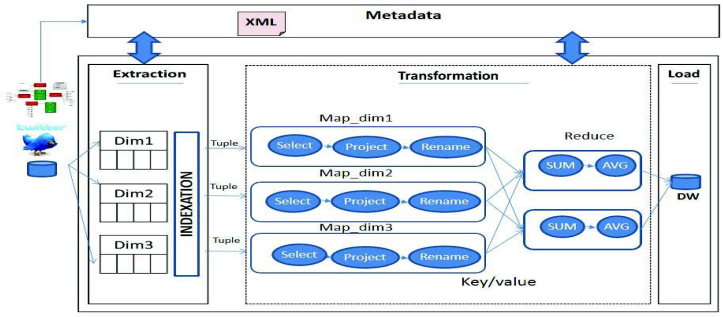

Fig. 1. BigDimETL approach

4 Using dimension partitioning in Extraction phase with Map Reduce

As mentioned in the introduction, Big Data is characterized by 3Vs (Volume, Velocity, Variety). "Twitter" shares the same Big Data characteristics which is a popular social network used by millions of users to exchange news, informations, opinions in different domains such as political, commercial and educational domains. Therefore, the processing of these data is considered as a big challenge and it is hard to extract useful information for experts and decisions makers. Thus, to guide ETL processes to work efficiently, the metadata will be used as a dictionary guide for the input data content described by all the data elements. The multidimensional structure is described in a metadata XML file, including the global structure of the DW which should be given in the beginning. Figure 2 shows a part from our XML file which describes the star schema of Tweets [12]. This file will be saved as a file system in HDFS (Hadoop Data File System), in order to be accessible during ETL steps. Therefore, we have at first, the description of facts, which is composed of identifier "id-fact", foreign key of dimensions, and a list of measures. At the sight, we find a description of the dimensions which is composed of a list of attributes identified with "id-dim". So, we have as input the metadata of our Data Warehouse and the Twitter data presented in a JSON format as a tree of objects.

4.1 Algorithm for extracting phase with BigDimETL

In our work, we partitionate the input data for distributed Database vertically. The work of [10] considers vertical partitioning as partial loading data in order to ameliorate row data processing. Vertical partitioning is very significant, which leads us to use this principle in our case, especially, to create the MDS. In addition, Vertical partitioning reduces the amount of data that have to be accessed by queries that operate on a small subset of columns since only the required

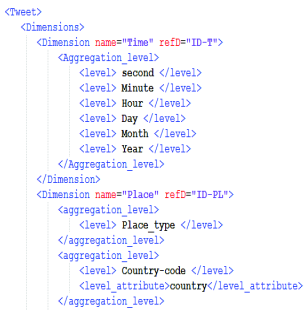

Fig. 2. Multidimensional structure of Twitter

columns have to be scanned [10]. Vertical partitioning of CSV file is similar to the partitioning of relational schema splits. Thus, we propose to convert the tree structure of JSON (object oriented) into columns structure (Columns oriented) (CSV). In this latter structure, the first line is conserved to present keys of elements and its values are reserved for the rest of file, in which each line is a record (tuple) that presents a tweet. The main advantage is that CSV structure is oriented columns which can represents different dimensions and facts clearly. Also, the structuring into CSV format is a compliant with Map Reduce paradigm [2]. In our case, the process to convert JSON structure to CSV file is based on a Metadata structure of our DW. The algorithm 1 shows the details of the whole processes of converting, partitioning and indexing data with MR paradigm. The operations in the first step are reserved to parse our XML metadata, then the second step is reserved to parse JSON file and to compare the correspondence between XML elements and JSON keys. The map function Key value in this case receives the Json key attached with its nested element, then the value of the map function will receive its value. Next, with the use of the reduce function we group all values which have the same key and we make it as CSV column. After the partitioning phase, we need to immediately index each partition in order to reduce the partitions search area by indexing directly the desired partition. Browsing sequentially big amount of data is a fastidious operation, especially for query processing. The aim of indexing phase is to identify each partition contents to facilitate parsing the information. In our context, the indexing phase is based on Dimensions identifiers "id_dim".

5 Algorithm for Selection operation for BigdimETL

The transformation phase is responsible for handling the extracted data. Typically, this phase is composed of series of operations (Select, Project, Conversion ...) applied on the extracted data. The huge volume of data requires a new methodology to be treated in the nick of time. So, the integration of MapReduce principal in the classical operations can minimize a lot of time consuming. In our approach BigDimETL integrates MapReduce paradigm into DSS, and

Algorithm 1: Extract and partitioning data

Input: MDD:Metadata document, FJ: File JSON, FCSV : CSV file
Output: Key list , Value list

```
Parsing Metadata
Extract dimensions
Extract facts
Parsing JSON Input data
Compare extracted dimensions (datt) and facts (fatt) according to Keys in
    JSON file
foreach Dimension and Fact do
    if datt = metadata dim or fatt= metadata fact then
        Key = dimension_name first line of CSV= Key
        Value= list of value of the same key Put values in the rest of CSV file
Create CSV column Index Facts and Dimensions : give index for each group of
    columns (fact, dimension)
```

offers the capabilities to manage ETL Operations. Hence, the transformation phase will be treated as Map and Reduce functions. In this paper, we focus on SELECT operation since it is the most frequently used query. It is responsible of restricting specific data to be loaded into the DW. Moreover, this operation is the basic query to retrieve data, and enables the selection of one or many columns from oriented columns structure. The result of SELECT statement is a set of records from one or several sources. In our context, it is very interesting to parallelize the treatment of this operation. The basic syntax of SELECT statement in relational algebra is as follows: $\sigma(Q)_P$, Where Q is a selection condition or a predicat. This predicat is applied independently to each individual tuple t in a partition P. For example, we need to extract all tuples from Dim_Place where the attribute "Place" is not NULL.
$t = \sigma(Place <> NULL)_{Dim_Place}$.
In our case, we need to translate the "select operation" statement according to MapReduce structure(key, value). Thus, each Map operation retrieves specific dimension "Dim_place" which is identified in the indexed file. Then, we apply the predicate verification $Q = Place <> NULL$. The algorithm 2 proposed in this work, provides the affectation of key and value of received data: Hence, we have as input key each tuple t from the input data. The input value is the same tuple t as the key, where the predicate P is improved else the value will take 0 as value. Thus, we will take as input the partitioned CSV according to the dimensions and facts. As a result we find (key, value) in the (t,t) structure if the predicate is satisfied else we obtain (t,0). In this case, Selections really do not need the full power of MapReduce. We can conserve the output data without the shuffle and the combination phases which are considered as time consuming phases.

Algorithm 2: Select operation

Input: $key = t$, $P = Predicat$, $value = t$, index= List of indexed partition
Output: key , Value

```
1 Select(P)
2 begin
3     Get partition index (direct access)
4     foreach tuple t in Dimension D do
5         if Predicat=true then
6             Value= tuple
7         else
8             Value = 0
9 return(Key, value)
```

6 Theoretical experiments

As mentioned above, the twitter data were used as input for our approach. The default JSON format of these data leads to several causes such as: i) Huge time consuming during the parsing process because of the nested elements (object tree structure). ii) In the case of nested elements, it is difficult to identify the next one, so we need to determine the start of the next JSON element. Based on these reasons, we propose to convert the JSON into CSV format and to execute the following select operation: $\sigma(Place <> NULL)_{Dim_{Place}}$ on these two formats with MapReduce algorithm in two different cases. The first case using Horizontal Partitioning (HP) data and in the second case, using Vertical Partitioning (VP) data according to multidimensional structure. In this section, we aim to analyze the feasibility of the proposed approach. Therefore, in table 1 presents complexity of the select operation execution using HP and VP. Using the Horizontal partitioning, we notice that the complexity of the select operation is equal to $O(n^2)$ for the JSON and CSV formats. This complexity is due to the parsing of all elements with nested loops and sub elements to extract "Place element" for the JSON format. While for the CSV format, the complexity is the result of parsing all tuples and all columns in order to verify the predicate P. Consequently, the collect of the treated data on the reduce function for each next query job (corresponding to a join) presents a time consuming task. In the other side, we notice that the complexity of the select operation using the vertical partitioning is equal to $O(n) + O(n)$. The oriented columns structure (CSV) of tweets is composed by several partition (dimension or fact content). Hence, it is easy to select the reference of Place partition using the index file. As a result, the access will be straight to the suitable partition. As a conclusion, for the theoretical point of view, the cost of the same queries is estimated by the number of loops, and research time.

Table 1. Algorithm Complexity of Select operation

Input	Partitioning	Select operation Complexity
JSON	horizontally	$O(n^2)$
CSV	horizontally	$O(n^2)$
	vertically	$O(n) + O(n)$

7 Conclusion

In this paper we have introduced the first and the second part of our approach BigDimETL. Our approach leads to integrate Big Data with conserving the multidimensional structure of DW. The integration of the parallelism aspect of ETL processes through MapReduce paradigm is very useful in our case. We concentrate to adapt the extraction and Transformation phase with MapReduce paradigm. Hence, to distribute our input data we have used vertical partitioning according to dimensions of multidimensional structure described in the metadata. As a future work, we aim to implement the ETL phases, and to support the various functionalities of ETL (CDC, SCD).

Acknowledgments. This work is dedicated to the soul of my supervisor, Dr. Lotfi Bouzguenda, who left us in juin 2016. We are very grateful for his help, his advice and his prestigious remarks. May his soul rest in peace.

References

1. Arres, B., Kabachi, N., Boussaid, O.: Building OLAP cubes on a cloud computing environment with mapreduce. In: ACS International Conference on Computer Systems and Applications, AICCSA. pp. 1–5 (2013)
2. Bala, M., Boussaïd, O., Alimazighi, Z.: P-ETL: parallel-etl based on the mapreduce paradigm. In: 11th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA. pp. 42–49 (2014)
3. Bellatreche, L., Schneider, M., Mohania, M.K., Bhargava, B.K.: Partjoin: An efficient storage and query execution for data warehouses. In: DaWaK. pp. 296–306 (2002)
4. Berro, A., Megdiche, I., Teste, O.: Graph-based ETL processes for warehousing statistical open data. In: Proceedings of the 17th International Conference on Enterprise Information Systems. pp. 271–278 (2015)
5. Chung, W.C., Lin, H.P., Chen, al.: Jackhare: a framework for sql to nosql translation using mapreduce. Autom. Softw. Eng. 21(4), 489–508 (2014)
6. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008)
7. Deb Nath, R.P., Hose, al.: Towards a programmable semantic extract-transform-load framework for semantic data warehouses. In: Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP. pp. 15–24 (2015)

8. El Akkaoui, Z., Mazón, J.N., al.: Bpmn-based conceptual modeling of etl processes. In: Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery. pp. 1–14. DaWaK'12 (2012)
9. El-Sappagh, S.H.A., Hendawi, A.M.A., El Bastawissy, A.H.: Original article: A proposed model for data warehouse etl processes. J. King Saud Univ. Comput. Inf. Sci. 23(2), 91–104 (2011)
10. Jaspreet Kaur, K.K.: A new improved vertical partitioning scheme for non relational databases using greedy method. International Journal of Advanced Research in Computer and Communication Engineering 2 (2013)
11. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. John Wiley & Sons, Inc., 2nd edn. (2002)
12. Kraiem, M.B., Feki, J., Khrouf, K., Al.: Modeling and olaping social media: the case of twitter. Social Netw. Analys. Mining 5(1), 47:1–47:15 (2015)
13. Liu, X., Thomsen, C., Pedersen, T.B.: Etlmr: A highly scalable dimensional etl framework based on mapreduce. Trans. Large-Scale Data- and Knowledge-Centered Systems 8, 1–31 (2013)
14. Liu, X., Thomsen, C., Pedersen, T.B.: Cloudetl: scalable dimensional ETL for hive. In: 18th International Database Engineering & Applications Symposium, IDEAS. pp. 195–206 (2014)
15. Oliveira, B., Belo, O.: Using reo on etl conceptual modelling: A first approach. In: Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP. pp. 55–60. DOLAP '13 (2013)
16. Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., Silvestri, C.: Trajectory data warehouses: Design and implementation issues. JCSE 1(2), 211–232 (2007)
17. Silva, D., Fernandes, J.M., Belo, O.: Assisting data warehousing populating processes design through modelling using coloured petri nets. In: 2013 - Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies. pp. 35–42 (2013)
18. Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., Murthy, R.: Hive: A warehousing solution over a map-reduce framework. Proc. VLDB Endow. 2(2), 1626–1629 (2009)
19. Trujillo, J., Luján-Mora, S.: A UML based approach for modeling ETL processes in data warehouses. In: 22nd International Conference on Conceptual Modeling, 13-16, 2003, Proceedings. pp. 307–320 (2003)
20. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for etl processes. In: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP. pp. 14–21. DOLAP '02, ACM, New York, NY, USA (2002)
21. Vassiliadis, P., Vagena, Z., al.: ARKTOS: towards the modeling, design, control and execution of ETL processes. Inf. Syst. 26(8), 537–561 (2001)
22. White, T.: Hadoop: The Definitive Guide. O'Reilly Media, Inc. (2012)