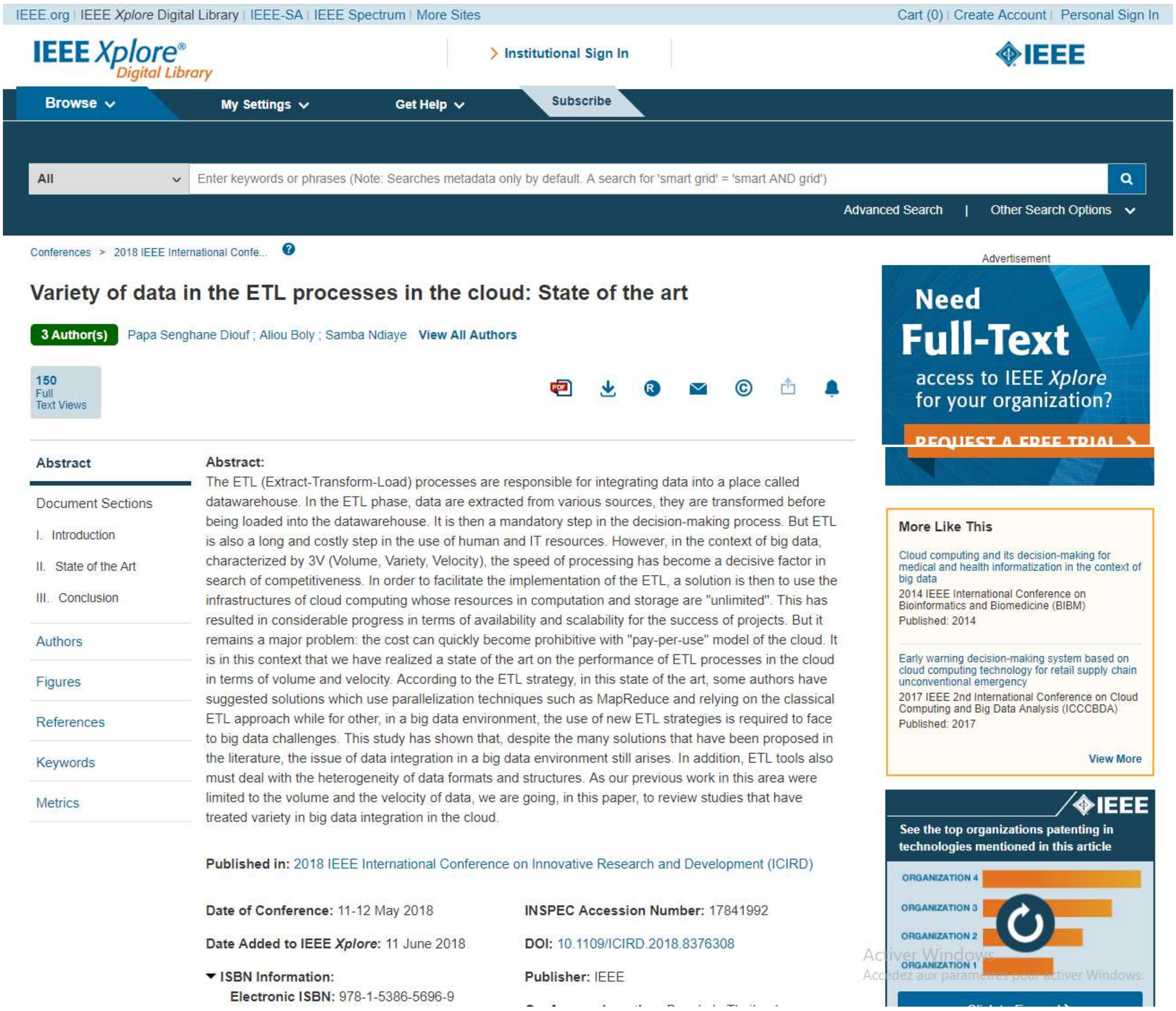
Publication pour la présente candidature LAFPT

Titre	Variety of data in the ETL processes in the cloud: state of the art
Auteurs	Papa Senghane Diouf ; Aliou Boly ; **Samba Ndiaye**
Référence	2018 IEEE International Conference on Innovative Research and Development (ICIRD)
Editeur	IEEE
Pages	95-99
Année	2018
DOI	10.1109/ICIRD.2018.8376308
URL	https://ieeexplore.ieee.org/document/8376308
Index	https://www.scopus.com/authid/detail.uri?authorId=6701604512
ISBN	978-1-5386-5696-9
Encadreur	Oui
Extraire d'une thèse	Non


IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites
Cart (0) | Create Account | Personal Sign In
IEEE Xplore® Digital Library
> Institutional Sign In
IEEE
Browse
My Settings
Get Help
Subscribe
All
Enter keywords or phrases (Note: Searches metadata only by default. A search for 'smart grid' = 'smart AND grid')
Advanced Search | Other Search Options
Conferences > 2018 IEEE International Confe...

Variety of data in the ETL processes in the cloud: State of the art

3 Author(s) Papa Senghane Diouf ; Aliou Boly ; Samba Ndiaye View All Authors

150 Full Text Views

Abstract
Document Sections
I. Introduction
II. State of the Art
III. Conclusion
Authors
Figures
References
Keywords
Metrics

Abstract:
The ETL (Extract-Transform-Load) processes are responsible for integrating data into a place called datawarehouse. In the ETL phase, data are extracted from various sources, they are transformed before being loaded into the datawarehouse. It is then a mandatory step in the decision-making process. But ETL is also a long and costly step in the use of human and IT resources. However, in the context of big data, characterized by 3V (Volume, Variety, Velocity), the speed of processing has become a decisive factor in search of competitiveness. In order to facilitate the implementation of the ETL, a solution is then to use the infrastructures of cloud computing whose resources in computation and storage are "unlimited". This has resulted in considerable progress in terms of availability and scalability for the success of projects. But it remains a major problem: the cost can quickly become prohibitive with "pay-per-use" model of the cloud. It is in this context that we have realized a state of the art on the performance of ETL processes in the cloud in terms of volume and velocity. According to the ETL strategy, in this state of the art, some authors have suggested solutions which use parallelization techniques such as MapReduce and relying on the classical ETL approach while for other, in a big data environment, the use of new ETL strategies is required to face to big data challenges. This study has shown that, despite the many solutions that have been proposed in the literature, the issue of data integration in a big data environment still arises. In addition, ETL tools also must deal with the heterogeneity of data formats and structures. As our previous work in this area were limited to the volume and the velocity of data, we are going, in this paper, to review studies that have treated variety in big data integration in the cloud.

Published in: 2018 IEEE International Conference on Innovative Research and Development (ICIRD)

Date of Conference: 11-12 May 2018

Date Added to IEEE *Xplore*: 11 June 2018

ISBN Information:
Electronic ISBN: 978-1-5386-5696-9

INSPEC Accession Number: 17841992

DOI: 10.1109/ICIRD.2018.8376308

Publisher: IEEE

More Like This

Cloud computing and its decision-making for medical and health informatization in the context of big data
2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)
Published: 2014

Early warning decision-making system based on cloud computing technology for retail supply chain unconventional emergency
2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)
Published: 2017

View More

Performance of the ETL processes in terms of volume and velocity in the cloud: state of the art

Papa Senghane Diouf, Aliou Boly, Samba Ndiaye
Cheikh Anta Diop University
Dakar, Senegal
{papasenghane.diouf&aliou.boly&samba.ndiaye}@ucad.edu.sn

***Abstract*—The ETL (Extract-Transform-Load) consists of extracting data from various sources, transforming and loading them into a place called datawarehouse. ETL is a mandatory step in the projects which implement decision-making information systems or knowledge management systems within organizations. But it is also a long and costly step in the use of human and IT resources. However, in the context of big data, characterized by 4V (Variety, Velocity, Volume and Veracity), the speed of processing has become a decisive factor in search of competitiveness. In order to facilitate the implementation of the ETL the solution is then to use the infrastructures of cloud computing whose resources in computation and storage are unlimited. This has resulted in considerable progress in terms of availability and scalability for the success of projects. But it remains a major problem: the cost can quickly become prohibitive with "pay-per-use" model of the cloud. So, in this case, how to find ETL solutions built on the cloud at a lower cost? A great deal of suggestions have been made. In this article, we have reviewed these works by highlighting the performance aspects of data processing in terms of volume and velocity.**

***Index Terms*—etl, big data, cloud, performance, cost.**

I. Introduction

Business Intelligence (BI) enables the production of relevant information and knowledge within organizations. That is why their use has now become widespread. A key step in the process of implementing a decision-making system is the design and implementation of a datawarehouse. In this process, data from various sources are first and for most extracted, then cleaned and eventually standardized before being stored in the datawarehouse. This very important step in the decision-making process is called ETL (Extract - Transform - Load). Nowadays, organizations produce large amounts of data in a wide variety of formats at a rapid rate [1] [2]: this is the era of big data. All this makes the ETL step difficult, time consuming and costly. The latter must adapt to the context of the big data because the traditional approaches prove to be inadequate or even impossible to implement [3] [4] [5]. This involves using cloud features such as infinite availability and resource elasticity [6] [7], usage billing, data remoteness, and so on. In this paper, we have reviewed the works that have emphasized on the improvement of the performance of the ETL processes, particulary the volume and velocity aspects of data.

II. State of the art

Several studies have already addressed the issue of the improvement of the performance of the ETL processes in a big data context [8] [9] [10] [11][12] [13] [14] [15] [16] [17]. According to the ETL methodology used for this purpose, they can be divided into two categories:

1) traditional approaches, based on the traditional ETL architecture;
2) new approaches based on new ETL methods that we describe as "ETL approaches oriented big data" in this study.

A. Traditional ETL approaches

Fig.1 describes the ETL process

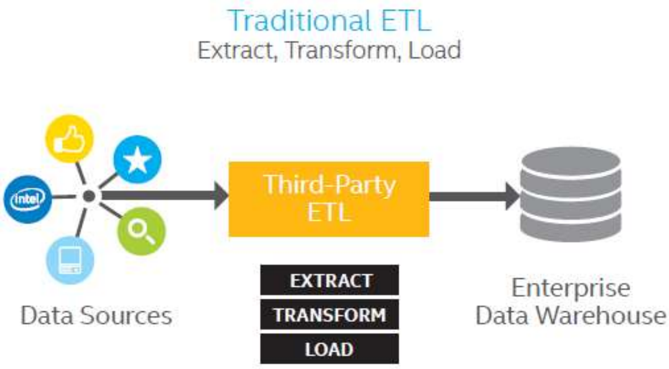
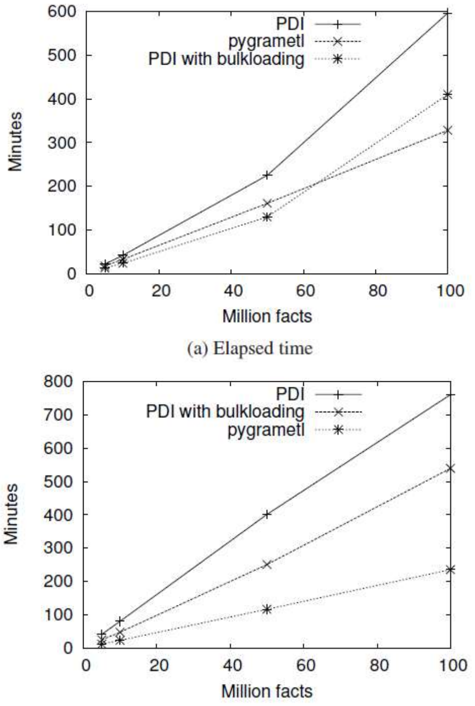


Fig. 1. Traditional ETL approach [17].

1) Research works done by Thomsen et al.[13]: For Thomsen et al.[13], the graphic tools used to make ETLs, despite the easiness of their use, have limitations. Indeed, for certain specific case to an organization, these tools are not capable of finding a solution. It is necessary in that situation to develop specific solutions. This is often complex, costly and time consuming to implement. To overcome this type of limitation, Thomsen et al.[13] suggest the code programming of the ETL process. These authors propose Pygrametl, a framework based on the Python language. Pygrametl facilitates ETL programming by providing features such as the access to data sources, powering dimension and fact tables, and so forth. Thus, Pygrametl makes it possible to create customized ETL solutions. Experiments are made by authors to compare

Pygrametl with Pentaho DataIntegration (PDI), a graphical ETL solution commonly used in organizations. The Fig. 2 below charts the obtained results:



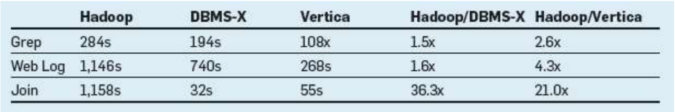
(a) Elapsed time

(b) CPU time

Fig. 2. Performance of Pygrametl [13].

Results have proved that Pygrametl is faster than PDI (CPU time) except in the case of small amounts of data bulkloaded. In conclusion, Thomsen et al.[13] have shown that ETL development with Pygrametl is more efficient [13] than ETL with graphical tools. Nevertheless, Pygrametl is implemented in Python. It should be emphasized that Pygrametl needs to develop in Python language, which can be constraining.

2) Research works done by Stonebraker et al.[16] : Stonebraker et al.[16] propose to compare the use of the MapReduce processing paradigm with that of the parallel databases in the datawarehousing process. This comparison is made on each step of the process. Specifically, MapReduce on Hadoop is compared to two parallel DBMSs that are line-oriented DBMS-X and column-oriented Vertica.



	Hadoop	DBMS-X	Vertica	Hadoop/DBMS-X	Hadoop/Vertica
Grep	284s	194s	108x	1.5x	2.6x
Web Log	1,146s	740s	268s	1.6x	4.3x
Join	1,158s	32s	55s	36.3x	21.0x

Fig. 3. Comparison of the performance of systems built on parallel databases and those based on MapReduce [16].

For the ETL process, the authors have proved that the use of MapReduce is more efficient than that of parallel DBMS. This can be explained by the large capacity of MapReduce to load and process large masses of data rapidly [18] [16]. In addition, MapReduce is fault-tolerant and can be run in heterogeneous environments.

3) Research works done by of Liu et al.[10] [11] : Based on the Pyrametl approach proposed by Thomsen et al. [13] and the MapReduce paradigm advocated in the ETL process used by Stonebraker et al.[16], Liu et al.[11] suggest the ETLMR solution, which is actually the MapReduce version of Pygrametl. ETLMR improves the Transformation and Loading phases of the ETL process [11] by implementing distribution strategies. Data are partitioned into elements of equal size before being processed through Map / Reduce tasks. ETLMR was compared with PDI. The results, illustrated in Fig. 4, have shown that ETLMR is more efficient than PDI, especially when the number of tasks increases.

Tasks	4	8	12	16	20
ETLMR	246.7	124.4	83.1	63.8	46.6
PDI	975.2	469.7	317.8	232.5	199.7

Fig. 4. Processing time [11].

Liu et al.[10] have proposed CloudETL as a result of ETLMR. The latter lean on Hadoop as the ETL execution platform and Hive as a warehousing system. The use of Hive enables to take advantage of the power of the MapReduce paradigm using HQL queries (HQL being very close to SQL).

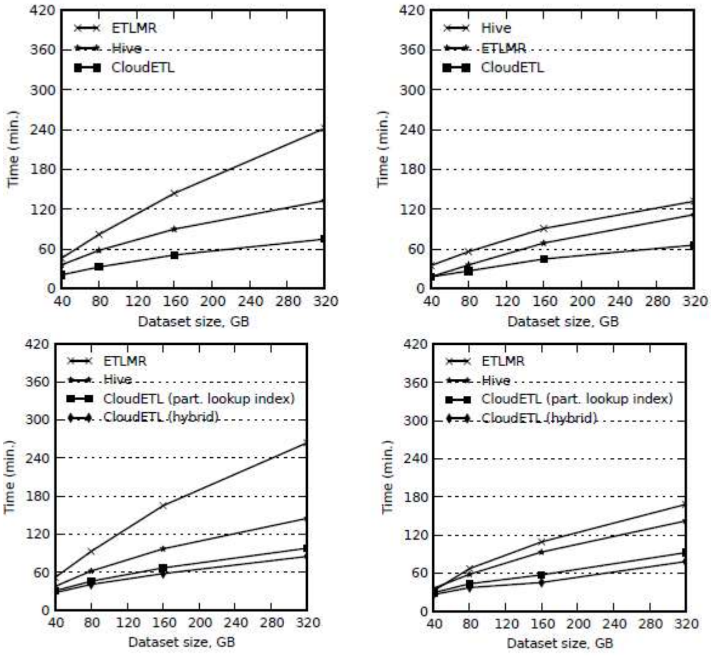
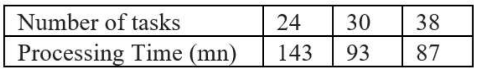


Fig. 5. Processing time [10].

The experiments in Fig. 5 show that CloudETL performs better than ETLMR.

4) *Research works done by Bala et al.[9] [14] :*

Bala et al.[9] have noticed that the use of ETL tools such as Pygrametl, ETLMR and CloudETL requires some expertises. Therefore, they offer a new tool which is more accessible to non-IT professionals while still performing well: P-ETL (Parallel-ETL), which was developed in the Hadoop/Mapreduce environment. With this in mind, data mapping tasks are assigned to the mappers. The mapper processes the data in a transformation tunnel (T1, T2,) where by each Ti performs a specific operation. The role of the reducer is to synthesize data from the mappers. The P-ETL method differs from the traditional approaches of ETL with graphical interface because not only does it allow to visualize the process but also to make settings on the parallel environment [19]. The performance of P-ETL has been demonstrated through some experiments. Thus, for data of 300 GB, the use of P-ETL shows that the processing time decreases as the number of tasks increases (Fig. 6). This study also shows that there is a threshold, in terms of the number of parallel tasks, beyond which time saving is no longer significant. We thus find that the parallelization of the ETL process ([8] [9] [10] [11]) improves its performance. However, this parallelization is only limited to the process level. Bala et al. [14] proposed to extend this parallelization at the functional level. Therefore, a so-called BIG-ETL approach is proposed by the authors. By always using MapReduce, these authors have proposed the BIG-ETL solution. In BIG-ETL, functionalities and processes are distributed in two directions: horizontal, vertical. Experiments were conducted by the selection of the Change Data Capture (CDC) functionality to evaluate BIG-ETL. In the experiments, the P-ETL platform is the one used. This is explained, according to the authors [14] by the fact that BIG-ETL allows the distribution of data and the parallelization and distribution of the ETL process.



Number of tasks	24	30	38
Processing Time (mn)	143	93	87

Fig. 6. Processing time [9].

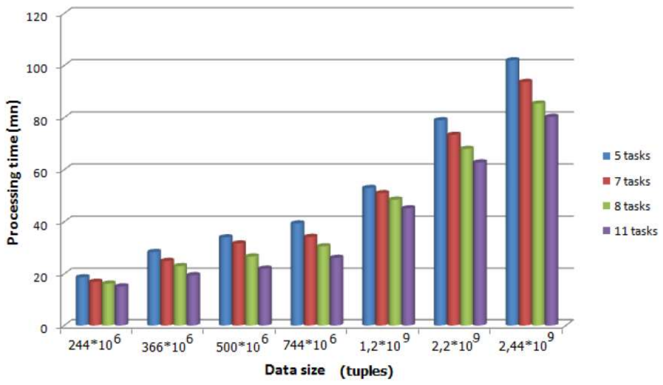
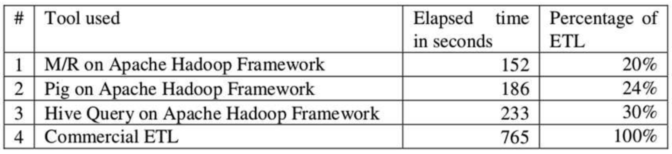


Fig. 7. Data processing time [14].

Fig. 7 showcases the results of the experiments carried out by the authors.

5) *Research works done by Misra et al.[8] :*

ETL software suites are often proprietary and costly. According to Misra et al.[8], the use of open-source ETL tools based on MapReduce can be a good alternative. The advantage of these tools is their scalability, their fault tolerance and their low cost. In order to better support their arguments, a comparative study regarding the commercial and open source tools has been done by the authors. The comparison criteria used by these authors are mainly cost and performance.



#	Tool used	Elapsed time in seconds	Percentage of ETL
1	M/R on Apache Hadoop Framework	152	20%
2	Pig on Apache Hadoop Framework	186	24%
3	Hive Query on Apache Hadoop Framework	233	30%
4	Commercial ETL	765	100%

Fig. 8. Comparison between ETL tool build under MapReduce and Commercial ETL tool [8].

Fig. 8 illustrates the comparison between commercial ETL tools and Hadoop / MapReduce based approaches. Misra et al.[8] have come to the conclusion that the open source solutions based on Hadoop / MapReduce are more efficient.

6) *Research works done by Cao et al.[20] :*

In the ETL process, the extraction time is relatively long. It is in this context that Cao et al. [20] suggest a parallel approach for the extraction of data by using the Hadoop/Mapreduce platform. A method of data segmentation is then proposed by these authors in order to make the distribution of the data more uniform at the level of each map. In the experiments, their solution is compared with that of the ETL tool Sqoop, which also leans on MapReduce. These two tools are distinguished by the data segmentation method used. The authors rely on the data segmentation algorithm RPDS while Sqoop uses a query to segment data (select split column), max (split column) from $< tablename >$. The experimental results are shown below in Fig. 9.

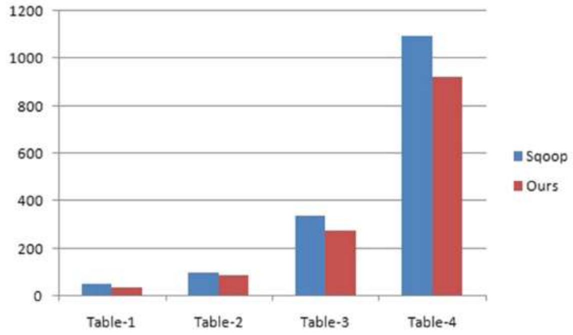


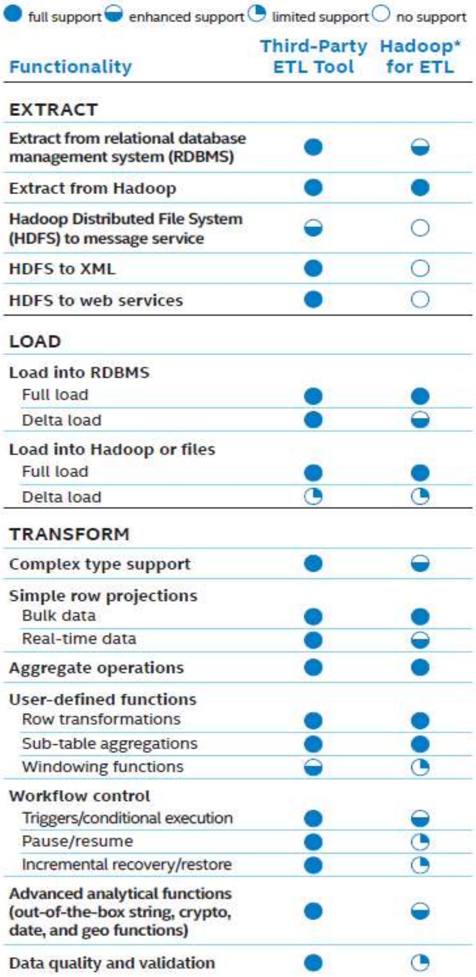
Fig. 9. Comparison with Sqoop [20].

Nevertheless, one of the limitations of this approach is that data come from relational databases. In a context of unstructured data, the performance of this method remains to be checked.

B. ETL approaches oriented big data

New methods have been proposed to improve the performance of the ETL process in the big data context. These methods are described in the following work.

1) Research works done by Intel IT [17] :
INTEL has proposed a new tool based on the ELT (Extract Load Transform) approach and built on Hadoop. This approach involves the extraction, loading and processing of data. Thus, by moving the transformation step at the end of the process eliminates the need to use a separate ETL tool. Then INTEL compared this new tool with its traditional and proprietary ETL tool. The Fig. 10 illustrates this comparison. From the extraction and loading point of view, the study shows that the Hadoop-based ETL is not as mature as Intel's tool is.



● full support ◒ enhanced support ◔ limited support ○ no support

Functionality	Third-Party ETL Tool	Hadoop* for ETL
EXTRACT		
Extract from relational database management system (RDBMS)	●	◒
Extract from Hadoop	●	●
Hadoop Distributed File System (HDFS) to message service	◒	○
HDFS to XML	●	○
HDFS to web services	●	○
LOAD		
Load into RDBMS		
Full load	●	●
Delta load	●	◒
Load into Hadoop or files		
Full load	●	●
Delta load	◔	◔
TRANSFORM		
Complex type support	●	◒
Simple row projections		
Bulk data	●	●
Real-time data	●	◒
Aggregate operations	●	●
User-defined functions		
Row transformations	●	●
Sub-table aggregations	●	●
Windowing functions	◒	◔
Workflow control		
Triggers/conditional execution	●	◒
Pause/resume	●	◔
Incremental recovery/restore	●	◔
Advanced analytical functions (out-of-the-box string, crypto, date, and geo functions)	●	◒
Data quality and validation	●	◔

Fig. 10. Comparison of the functionalities of the proprietary tool and the one based on Hadoop [17].

The costs of using each technology were also compared. For these authors, after one year, the cost of the ETL under Hadoop is less compared to the use of the proprietary tool.

2) Research works done by Guo et al.[12] :
Guo et al.[12] proposed the so-called TEL (Transform, Extract and Load) approach. TEL introduces a virtual layer between the sources and the datawarehouse. Virtual tables are used to perform the data transformation before their extraction and thier loading. For Guo et al.[12], the use of a buffer zone in traditional ETL approaches penalizes the process. In the TEL approach, this field is deleted. Experimental studies have been carried out by Guo et al to validate their solution. The TEL solution was compared with that of the ETL Kettle (Pentaho Data Integration) tool. The experimental results (TEL-M [12]) show that the performances of TEL have exceeded on the whole those of Kettle [12]. Fig. 11 gives an overview of Kettle's performance comparison with that of TEL.

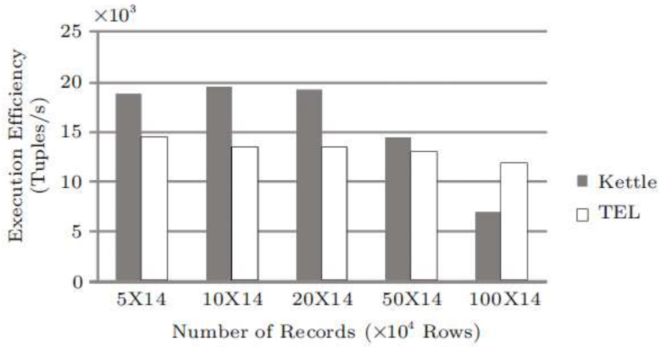


Fig. 11. Performance comparison between Kettle and TEL [12].

III. Conclusion

The ETL phase is an essential step in the datawarehousing process. This is by far the longest and most expensive phase in an implementation project. The adaptation of this ETL phase to the big data / cloud context, characterized by large volumes of heterogeneous and remote data, has therefore become a necessity. We are interested in the problem of performance management in terms of speed and cost. In this literature review, we have found that several approaches are proposed to address the issue. A first approach suggests new solutions based on dedicated programming environments, faster than the graphic tools used so far. For example, Thomsen et al. propose Pygrametl, a framework witch is implemented in Python language. The experiments carried out have proved an improvement in the ETL phase. A second variant of this approach exploits the advantages of task parallelization on Hadoop / MapReduce platforms to improve the performance of the ETL phase. New experiments have replaced Hadoop / Mapreduce by Spark, even more efficient. A second approach proposes new ELT (Extract - Load - Transform) or TEL (Transform - Extract - Load) architectures. The main interest of this new approach is the elimination of the "buffer zone" used in the ETL approach. Thus, in the case of the TEL architecture, experiments show that the use of disk space can be reduced to 50%. Studies have also showed that this approach, unlike traditional ETL methods, allows very large volumes of data to be processed very quickly. The performance in terms of data storage and execution time is improved. All these solutions have not yet taken into account the dynamicity or even the elasticity of the data in the cloud. Also, the issue of the cost of resource use has not been deeply studied in these different approaches. And, since the cloud billing model is in use, exploiting the elasticity of resources would reduce the cost. The implementation of ETL solutions that take these aspects of the cloud into account is a promising prospect.

References

[1] Anureet Kaur. Big data: A review of challenges, tools and techniques. 2016.

[2] Caesar Wu, Rajkumar Buyya, and Kotagiri Ramamohanarao. Big data analytics= machine learning+ cloud computing. *arXiv preprint arXiv:1601.03115*, 2016.

[3] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 42–47. IEEE, 2013.

[4] Xiufeng Liu, Nadeem Iftikhar, and Xike Xie. Survey of real-time processing systems for big data. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 356–361. ACM, 2014.

[5] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of big data on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[6] Barrie Sosinsky. *Cloud computing bible*, volume 762. John Wiley & Sons, 2010.

[7] Jakobus S Van der Walt. Business intelligence in the cloud. *South African Journal of Information Management*, 12(1):1–15, 2010.

[8] Sumit Misra, Sanjoy Kumar Saha, and Chandan Mazumdar. Performance comparison of hadoop based tools with commercial etl tools-a case study. In *BDA*, pages 176–184. Springer, 2013.

[9] Mahfoud Bala, Oussama Mokeddem, Omar Boussaid, and Zaia Alimazighi. Une plateforme etl parallèle et distribuée pour l'intégration de données massives. In *EGC*, pages 455–460, 2015.

[10] Xiufeng Liu, Christian Thomsen, and Torben Bach Pedersen. Cloudetl: scalable dimensional etl for hive. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 195–206. ACM, 2014.

[11] Xiufeng Liu, Christian Thomsen, and Torben Bach Pedersen. Etlmr: a highly scalable dimensional etl framework based on mapreduce. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems VIII*, pages 1–31. Springer, 2013.

[12] Shu-Sheng Guo, Zi-Mu Yuan, Ao-Bing Sun, and Qiang Yue. A new etl approach based on data virtualization. *Journal of Computer Science and Technology*, 30(2):311, 2015.

[13] Christian Thomsen and Torben Bach Pedersen. pygrametl: A powerful programming framework for extract-transform-load programmers. In *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP*, pages 49–56. ACM, 2009.

[14] M Bala, O Boussaid, and Z Alimazighi. Big-etl: extracting-transforming-loading approach for big data. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, page 462. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.

[15] Xiufeng Liu and Nadeem Iftikhar. An etl optimization framework using partitioning and parallelization. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 1015–1022. ACM, 2015.

[16] Michael Stonebraker, Daniel Abadi, David J DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, and Alexander Rasin. Mapreduce and parallel dbmss: friends or foes? *Communications of the ACM*, 53(1):64–71, 2010.

[17] Intel it center, evaluating apache hadoop* software for big data etl functions. White Paper, August 2014.

[18] Xiufeng Liu, Christian Thomsen, and Torben Bach Pedersen. Mapreduce-based dimensional etl made easy. *Proceedings of the VLDB Endowment*, 5(12):1882–1885, 2012.

[19] Mahfoud Bala, Omar Boussaid, Zaia Alimazighi, and Fadila Bentayeb. Pf-etl: vers l'intégration de données massives dans les fonctionnalités d'etl. In *INFORSID*, pages 61–76, 2014.

[20] Lianchao Cao, Zhanqiang Li, Kaiyuan Qi, Guomao Xin, and Dong Zhang. An efficient data extracting method based on hadoop. In *International Conference on Cloud Computing*, pages 87–97. Springer, 2014.