Towards a New Architecture for Data Multilevels Interactive Visualization in Big Data Domains

Moustafa Sadek Kahil
LAMIS Laboratory
Larbi Tebessi University
Tebessa, Algeria
moustafa-sadek.kahil@univ-tebessa.dz

Abdelkrim Bouramoul
MISC Laboratory
Constantine2 University
Constantine, Algeria
abdelkrim.bouramoul@univ-constantine2.dz

Makhlouf Derdour
LRS Laboratory
Larbi Tebessi University
Tebessa, Algeria
makhlouf.derdour@univ-tebessa.dz

***Abstract*—The progress of the Big Data technology in the different domains continues to reveal, day after day, so many issues affecting this field. The data presentation is a part of these issues, because it is directly related to the user that requires visualization to facilitate the data collection as well as the analysis and the utilization of results as needed. However, implementing visualization requires a low latency. Therefore, many problems regarding the Big Data major characteristics, namely volume, variety and velocity arise and should be addressed. Knowing that each Big Data domain has specific peculiarities, it remains far from thinking of a generic solution to the visualization problem. This latter is manifested in different kinds. One of these kinds that is prevalent in Big Data domains is hierarchical visualization, which is based on presenting data by levels, while respecting the relations between them. In this paper, we propose an architecture composed of modules that cover the Big Data visualization process, taking into account the hierarchical relations between data. In order to valorise our proposal, we implement a prototype that supports the modules composing our architecture which are developed using different tools such as Scrapy and D3js libraries. The experimentation, which was carried out in the educational domain, showed its applicability to real systems.**

***Keywords*—Big Data, Interactive Visualization, Data Scraping, Hierarchical Visualization, Scrapy, JSON**

I. Introduction

The rapid evolution of Big Data technology in different fields such as economic, academic, government, IT fields... has made exponential growth of data size and, as a result, multiple challenges are engendered such as data analysis, storage, searching... And because the data analysts are more comforted with visual representations, the data interactive visualization became obvious. In [1] and [2] the benefits of visualization tools are highlighted by taking into account decision-making, ad-hoc, data sharing and other features that continue to mark the growing need for these tools. In the first section, we present the imposed constraints to validate the data visualization in Big Data contexts. In the second section, we mention the most common related work to visualization, followed by a comparative study between them according to the criteria that are related to the presented constraints. Subsequently, we present in the third section our proposed architecture with a detailed description of each of its components. We also describe in the fourth section the experimentation to show the applicability of this architecture on real systems.

978-1-7281-2642-5/19/$31.00 ©2019 IEEE

II. Agreed Terminology

Before addressing the state of the art of the Big Data visualization and presenting our proposal, it would be useful to clarify the terms that will be used in our proposal.

- Domain: This is the application domain that is part of the Big Data domains.
- Domain class: The set of Big Data domains with similar features in term of visualization.
- Visualization levels: In the class of domains where data are viewed hierarchically, a visualization level is a level in which the data have the same degree of visualization.
- Pattern: It is an instance of a set of data that belong to the same visualization level.
- Preliminary visualization: It is the visualization of the first time where there is not yet interaction with the user to customize or perform the operations that are allowed.
- Entity: A component of the visualization system that can be active or inactive.
- Active entity: An entity that can directly change the state of the visualization system.

III. Visualization Constraints in Big Data Contexts

In order to implement a visualization system in a Big Data context, constraints that exist in the literature must be satisfied. But before engaging in our proposal, it must be emphasized that the constraints satisfaction one of the combinatorial optimization problems. These latter, not always having the same importance, belong to two main axes: hard constraints and soft constraints [16]. The hard constraints mean those that are essential to validate the visualization, and the non-satisfaction of one of them implies the invalidity of the system. While the soft constraints, they represent the constraints of preference. And depending on the field of application, any kind of constraints is designated. Indeed, the visualization domain can contain these two axes of constraints: (1) essential constraints to ensure to user the visualization and the interaction, and

warrant the dynamicity of the system facing changes or modifications, while respecting latency. The validity of visualization requires the satisfaction of any constraint contained in this axis, (2) other constraints to simplify the user's customization of visualization, exploration and research. And even there is a violation of some constraints that belong to this axis, the visualization could still be validated.

Taking into account the two axes of the mentioned constraints, and after having studied the constraints related to the data visualization in the Big Data domains, we propose to classify these constraints into four classes: (1) basic visualization constraints, (2) interactivity constraints, (3) scalability constraints and (4) constraints of structure.

- Basic visualization constraints are those that are essential to validate the visualization in any classical or modern domain. They can be summarized in two essential constraints: expressiveness and efficiency [10]. The first requires the ability of the visualization system to express a set of data by visual significants such as points, circles, squares... While the second, it is relative to the user: the data visualization system must ensure that the user understands the idea this graphic representation means. For example, in a tree representation, the visualization should highlight a meaning for the root, branches, and leaves.
- Interactivity Constraints represent the requirement of the functionalities that the visualization system must provide to the user to ensure interaction. The latter can manifest in several features such as zoom in and zoom out, selection, pan, fish-eye, filtering, search, detail on demand [2] [3].
- Scalability constraints are imposed to ensure the dynamicity of the visualization systems. In the Big Data domains, these constraints are often hard. They are summarized in three constraints [20]: (1) perceptual scalability which means the ability of visualization system to adapt with the perception of new input data that are massive and heterogeneous, (2) Realtime Scalability, the ability of this system to evolve in real time so as to achieve the velocity, and (3) Interactive Scalability which consists of adapting the system with not only interaction with the user, but also while introducing changes in the system such as adding new features that are related to interactivity.
- The structuring constraints are related to the optimization of the various tasks in the visualization process such as the ease of operation and the data integration.

IV. Overview and Related Work

The data visualization in the Big Data contexts has increasingly become a complex topic. This complexity is justified by the fact that each domain has its own particularities. For example, in the finance, data are often only statistical, the interest of viewing data is limited to representing results in diagrams, curves or other statistical representations. As for a marketing domain, other types of data other than statistical data, such as image and text, should be visualized. As a result, the visualization of data, that are characterized in the Big Data domains by their high volume, variety, and the need for high velocity, strongly depends on the application domain or class of domains to develop a solution. This does not make the generation of visualization an aspect of first priority.

To cite the relevant related work of data visualization, it is interesting to focus on two dimensions: (1) a methodological dimension which consists of studying the domain specificities, the relations between data and its component entities, and (2) an applicative dimension that focuses on the technical side, i.e. it includes the techniques and tools that can be used for the implementation of visualization systems. The realization is done essentially by taking advantages each technique or tool presents.

A. Methodological Dimension

In this dimension, we take the classification that was proposed in [13] which includes classes of methods for the data visualization namely: *Data Reduction*, *Hierarchical Exploration*, *Incremental and Adaptive Processing*, *Caching and Prefetching*, and *User Assistance*. There are work that optimize a task in the data visualization process. For example, in [8], the interest was to optimize the user queries before they are processed, so as to reduce the complex queries to minimize the response time. In [3], to accelerate the visualization process, it is parallelized by distributing tasks in the cloud. In [11], the user complex queries are processed incrementally, the result is, therefore, presented progressively to the user until the end of the query processing. Similarly, in [15], the user queries processing is done gradually in the cloud. The work that is proposed in [14] improves the exploration and analysis of data through the hierarchical visualization according to the relations between levels and nodes composing the hierarchical structure. There are also other work that aim to improve the interaction with the user by offering several features [12] such as drilling, panning, flipping and so on. The table I presents the advantages and disadvantages of each cited work.

B. Applicative Dimension

We target via this dimension the technical side, by answering the question: What do we want to visualize? And to answer to this question, three strategies proposed in [9] are to be considered: (1) *data-type visualization*: the visualization of a specific data type like text, image... (2) *dataset visualization*: the visualization of a dataset where the data types can vary but these data concern a specific subject and (3) *Special topic visualization* that consists to visualize an entire domain or even a class of domains having common characteristics. The table II illustrates these three strategies.

From a practical view, the data-type visualization would not be of great importance, because one of the essential Big Data characteristics is the variety, data are always heterogenous.

The data visualization follows one of the widespread techniques that are listed in [1] and [2]: *TreeMap*, *Circle Packing* and *Sunburst* for hierarchical visualization, *Parallel Coordinate* to visualize heterogenous data, *Streamgraph* for visualization around central axis, *Circular Network Diagram* to

TABLE I
SYNTHESIS OF RELATED WORK

Class	Technique Or Approach	Advantages	Disadvantage
Pre-processing	Binned Aggregation [3]	Structured visualization, Scalability (perceptual, interactive)	Used for visualizing medium size data, Real time constraint not satisfied
Real time processing	Parallel coordinate (5Ws model) [4], [9]	Scalability (perceptual, interactive)	High complexity, Unstructured visualization, Limited application domains
	imMens [6]	Scalability (real time, perceptual, interactive)	Risk of losing patterns, High complexité
	Effective BD Visualization [8]	Scalability (real time, perceptual, interactive)	Unstructured visualization, Complexity is very relative to the query.
Hybrid	RHadoop [7], HadoopViz [5]	Scalability (real time, perceptual, interactive)	Limited application domains, Slow process in the real-time case.

TABLE II
EXAMPLES OF DOMAINS RELATED TO DIFFERENT CLASSES OF DATA VISUALIZATION

Class	Examples of application
Special topic visualization	Network traffic, Health, Smart cities, Education, Physics, Astronomy, Atmospheric sciences, Bioinformatics, Business Intelligence, Marketing
Data-type visualization	Text, Video, Image
Dataset visualization	Health and medical data, Metrological data, Social networks

visualize objects with circles and link them according to the existing relations, and so on.
The appropriate technique can be chosen according to the specificities of domain while being based on the comparative study that is made in [1] and [2] where the criteria are: volume, variety and dynamicity.

V. AN APPLICATIVE ARCHITECTURE FOR DATA HIERARCHICAL VISUALIZATION IN BIG DATA DOMAINS

We are interested in our work to propose an approach allowing the data interactive visualization in the class of domains where the data require a hierarchical presentation, while supporting the data large size, veriety and offering high velocity to process the different queries. For this purpose, we list the characteristics and constraints that are related to this class in order to choose the methods that should be used in the development of our approach. We can summarize the visualization characteristics in this class of domains in the following points:

- Hierarchy: the data presentation is realized by levels. We can distinguish in the hierarchy one higher level, many intermediate levels and one lower level.
- Exploratory research of information: the interest of the data visualization is to facilitate the data research and exploration. This is achieved by providing to user a presentation with which he can interact to explore data he seeks to find.
- Scalability: Visualization must take into account the aspect of scalability that includes (1) Interactivity, (2) Real Time, and (3) Perception that we discussed in the Constraints section.

From the hierarchical domains' characteristics and constraints, we propose an approach that will take into consideration the following aspects:

- Provide to user an interactive visualization: define meaningful graphics that are understandable to him, provide him adjustment features such as: zoom, selection, customization
- Ensures the shapes' homogeneity for a quicker grip and familiarity to the user
- Ensure the availability of data whose patterns are displayed. This through a data updating process
- Provide criteria to adapt each user's own visualization according to his needs

The architecture that we propose aims to meet the needs mentioned above, i.e to wrap the visualization process in an educational field: on the one hand, it is an instance of the class of domains where data require a hierarchical presentation, and, on the other hand, it is one of the domains for which data visualization is still recent, while considering the constraints of dynamicity and scalability. The figure 1 presents the architecture we propose. It covers the process of data visualization that can be viewed hierarchically in the Big Data domains, while taking into consideration the user aspect as an actor (active entity) in this process. It is composed of four (04) non-sequential and complementary modules that take care of retrieving, storing and updating data, handling the visualization criteria, visualizing data and interacting with the user through this visualization. We describe the functioning of each component of this architecture to explain the progress of tasks composing each module. We also mention some tools, namely frameworks, libraries or web services that are used in each task.

A. Data Parsing and Storage Module

In a Big Data application domain, data can be structured, semi-structured and / or unstructured. And since our goal is to visualize them to facilitate analysis and exploration, it is useful to preprocess them to simplify the visualization task, it is a matter of formatting them. This module is dedicated to do that purpose: it takes the heterogeneous data retrieved via the semi-structuring tools to give them a format that allows them to be easily processed. This task is called *data scraping*. Among the tools dedicated to this task, we find:

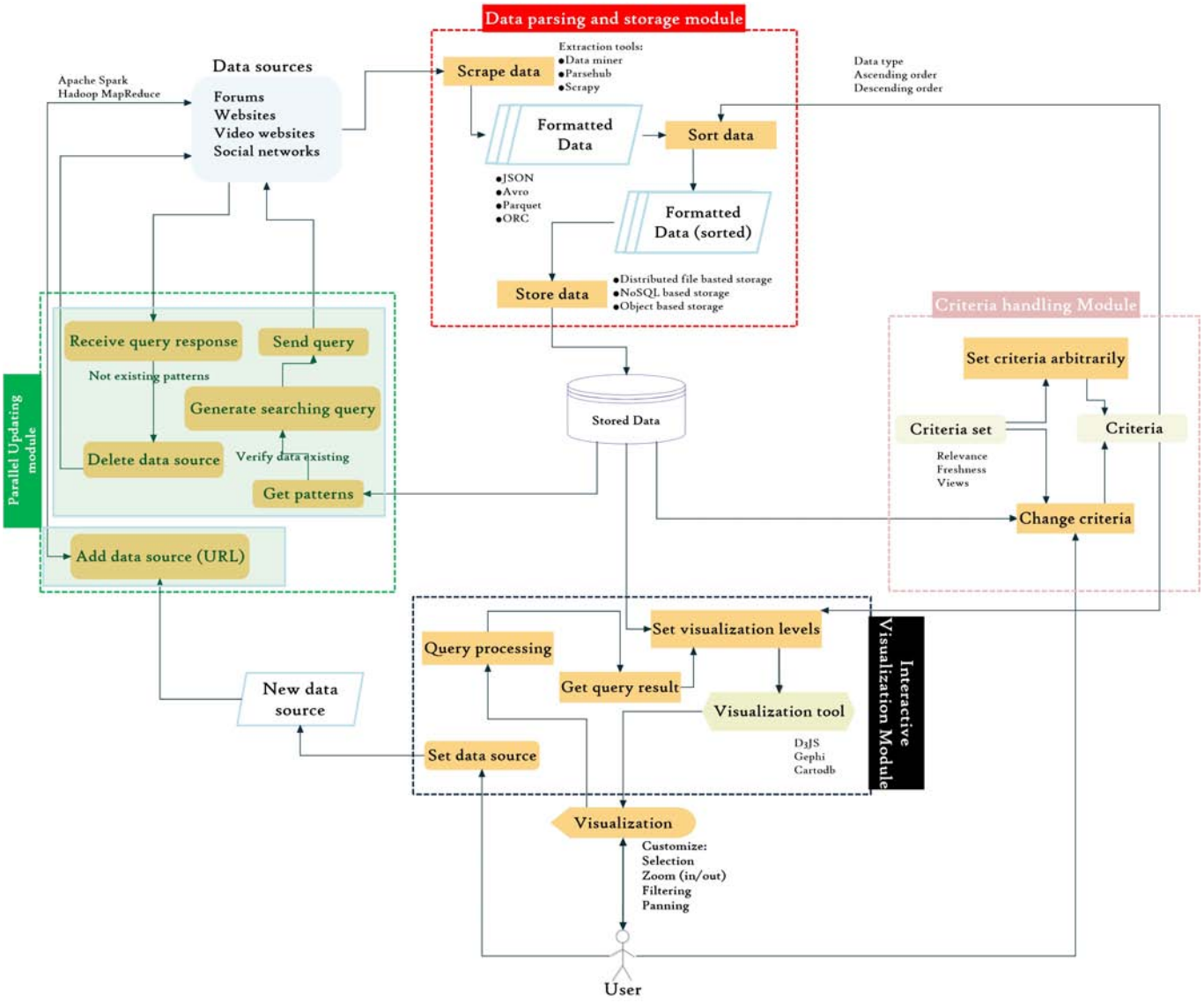

Fig. 1. The proposed architecture for data hierarchical interactive visualization

- Web services: Data Miner [17] [18]
- Desktop applications: Parsehub
- Libraries: Scrapy of Python [24] [23], Sparkler (Spark+Crawler) [22]

The result of formatting data by these services is an operational format such as csv, json [19].

Once the data have been formatted, they are sorted according to one of the existing criteria. These are defined before: according to studying the application domain, they are designated according to its characteristics and the requirements the visualization must respond to user. The last step of this module is to store the formatted and sorted data by one of the storage types presented in [25]: (1) distributed file system such as HDFS, BSF, (2) cloud such as Google Cloud Storage, Amazon... or (3) a NoSQL database such as Cassandra, MongoDB, Redis,... The storage type is chosen depending to the application domain. The pseudo algorithm 1 describes the process of this module.

B. Criteria Handling Module

This module handles the criteria that are used to sort data so as to ensure the customization of visualization according to the user needs. A criterion of sorting the formatted data is chosen from the criteria set. The choice is made either: (1) arbitrarily when the preliminary visualization starts or when there is no interaction with the user, or (2) responding to a user query to change the viewing criteria. The pseudo algorithm 2 describes the cases of choosing the sorting criterion. Whenever the user changes the data display criterion, the hierarchy is modified, i.e the sequence of levels composing the hierarchy changes. The latter is performed according to patterns, types of data, sub domains or other criteria that characterize each domain. For example, in the case of the education field, the

Algorithm 1: Parsing and storage algorithm

```
criteria-set: the set of all defined criteria
data-sources-set: the set of the exesting data sources
formatted-data: the data that are serialized
selected − criteria ← first criterion in criteria-set
for each data-source in data-sources-set do
    formatted − data ← parse data-source
    add formattedData in formattedData-set
end
get selected-criteria
sort formatted-data-set by selected-criteria
for each criteria from criteria-set do
    add criteria column to formatted-data-set
    store formatted-data-set by storage type
end
```

Algorithm 2: Handling criteria Algorithm

```
selected-criterion: instance of criteria-set
selected_criteria ← firstcriteriaincriteriaSet
if no changing-criterion-query then
    Set selected-criterion arbitrarly from criteria-set
else
    selected − criterion ←
     executequeryoncriteria − set
end
```

C. Data Parallel Updating Module

This module is responsible for two essential tasks:
(1) Verification: As indicated in the pseudo algorithm 3, in order to check if the data related to the visualized patterns still exist, it retrieves the patterns from the formatted and sorted data, generates for each pattern a query and run it on the data sources. If a pattern no longer exists, a query to delete the sources of the relative data is generated and executed.
(2) Adding data: If new sources are to be added (from the visualization module) so as to view them, an adding query is generated at this module to add these sources to the data sources set.
After one of two cases, the data are updated. Formatting data as well as the visualization must, therefore, be redone via resetting the modules which are associated to them.

D. Interactive Visualization Module

At this level the hierarchy rules are defined: the relations between levels are determined after the definition of the

Algorithm 3: Verifying data Algorithm

```
patterns-set: the set of patterns existing in the
 formatted data
patterns − set ←
 retrieveallpatternsfromformattedData − set
for each pattern from patterns-set do
    generate search query (pattern-search-query)
     execute query in dataSources-set
    if no result then
        delete dataSource containig pattern
        go to parsing algorithm
    else
        continue
    end
end
```

visualization levels as well as the patterns that compose them as shown in the pseudo algorithm 4. After that, the patterns are visualized via a specialized tool by specifying the number of levels to display. Data visualization tools are numerous and can be classified into three classes [20]: (1) platforms, (2) services and (3) libraries.
In this module, the interaction with the user includes: (1) adding data sources by sending them to the update module, (2) generating queries to change the visualization criteria, (3) answering the change queries for the adjustment parameters such as zoom and selection and (4) generating the search queries and visualizing the results.

Algorithm 4: Defining levels Algorithm

```
patternlevel: the level at which the current pattern
 belongs
sub-pattern: the level below the current level
for each pattern in the top level do
    if pattern has sub patterns then
        create new patternlevel
        nbr − of − levels ← nbr − of − levels + 1
    else
        continue
    end
    for each sub-pattern do
        add sub-pattern to patternLevel
        defineLevels
    end
end
```

VI. Experimentation and Discussion

In order to implement our proposed architecture, we choose the educational domain, a domain where data are usually represented hierarchically, coming from different types of educational web sources: websites, forums, videos... The purpose is to realize a visualization that meets the needs we have listed. The experimentation includes the implementation of any module that composes this architecture, basing on the results

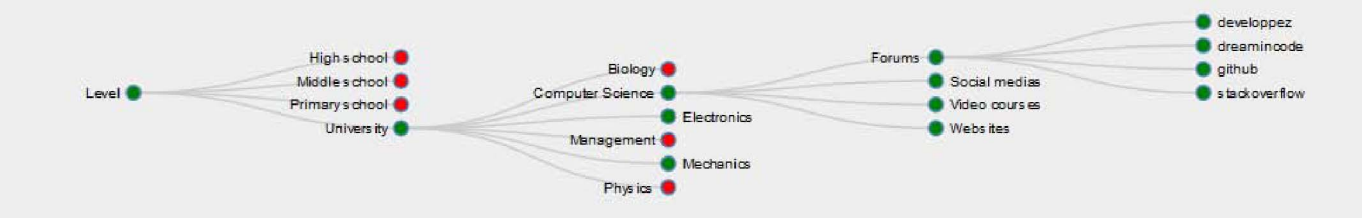

Fig. 2. Overview of the hierarchical visualization produced by the framework supporting our architecture

these modules present. For that, we measure the result of each module separately from the others.

In the parsing and storage module, we choose as an example the forums of the educational website: www.developpez.net as well as the videos of the website: cosmolearning.org. The data scraping, which is done through the library Scrapy of python and xpath1.0, gives a result formatted in CSV or JSON, which is semi-structured. This facilitates the tasks of handling criteria and visualizing data. Retrieving data includes collecting discussions from all the forums under the URL: http://developpez.net/forums/, and the video courses from the URL: https://cosmolearning.org/videos/. To format the data that are retrieved in the JSON file, we first designate two attribute-pairs: (discussion-title and discussion-url) and (video-title abd video-url). With a computer whose CPU (core i7 4600u) is of frequency of 2.4 GHz and the capacity of RAM is 8 GB, under the operating system Linux Ubuntu 18.04, and a connection internet of 1 MB of flow, we obtain the results that are shown in the table III. with such computer features, these latter are accpetable. The data scraping speed mainly depends on the processor and the internet speed. For the reasons that we list below, we add for "discussion" the attributes: name-of-forum (under which this discussion appears), number-of-responses and number-of-views, and for "video" the attributes: topic and number-of-views.

- The scraping process is parallel. The formatted data are, therefore, not sorted.
- Adding entities in formatted data implies increasing the number of criteria. Because, as is mentioned in the description of handling criteria module, the change of a criterion requires the modification of the formatted data.

We find that by adding other properties, the time of data scraping does not change. This is justified by the exclusive dependency of the CPU once URLs are opened in the Scrapy program. In handling criteria module, the JSON data are modified through queries which are generated by jQuery. The modification here means the change of the attributes sequence.

The visualization, which is realized through the javascript library D3JS using the source code that was proposed in [26], is consistent with the formats we have used. It provides to user an interface with which he interacts via the provided functionalities: Selection, zoom in, zoom out and detail on demand.

The figure 2 presents a screenshot of the visualization that we implemented, which is preliminary in this case. This presentation is relized according to the criterion that we put by default which considers the sequence: (Level of study, Specialty, Type of data sources, Data sources, Data, Link). The pattern "Data" varies according to the type of data source. The table IV shows the different patterns that can associate each data type.

In updating data module, two functions have been defined: (1) The first function retrieves all the links from the formatted data to test if they still exist or not. For each https link, a query is generated to open it. If the link no longer exists, the function deletes all the formatted data to which they are linked. (2) The second function transmits the links which are added by the user to the data scraping and formatting module in order to scrape and format all the data that exist in these links and add them to the existing formatted data.

TABLE III
RESULTS OF DATA SCRAPING

	Number	Latency
Forums	97	11.2 seconds
Discussions	297796	4620 seconds
Videos	2613	327.7 seconds

TABLE IV
DATA SOURCES' DIFFERENT PATTERNS

Data source	Patterns
Forum	Discusssion
Website	Article
Video Courses	Video
Social Media	Video, Text, Image

VII. CONCLUSION

In this paper, we have proposed an architecture, that encompasses the process of interactive visualization of data that can be represented hierarchically in Big Data domains. One of the benefits of this architecture is that it is composed of four modules so as to ensure the modularity. Given the high complexity of data updating, there is interest to use tools that provide parallel processing. That's why a thinkable solution consists on implementing the tasks of the update module using Apache Spark, Apache Flume or other powerful frameworks. The other modules that compose our architecture can be optimized to reduce latency. We plan to optimize the data scraping and formatting process by parallelizing it either

by running it in the cloud or by using other framework-based tools that provide the distributed processing such as the Apache Spark sparkler platform. Similarly, the research task can be enhanced by providing other features such as semantic research.

REFERENCES

[1] S. M. Ali, N. Gupta, G. K. Nayak, and R. K. Lenka, "Big data visualization: Tools and challenges", in 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Greater Noida, India, 2016, pp. 656–660.

[2] L. Wang, G. Wang, and C. A. Alexander, "Big Data and Visualization: Methods, Challenges and Technology Progress", p. 6.

[3] Q. Fu, W. Liu, T. Xue, H. Gu, S. Zhang, and C. Wang, "A big data processing methods for visualization", in 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, China, 2014, pp. 571–575.

[4] J. Zhang, M. Huang, and Z.-P. Meng, "Visual analytics for BigData variety and its behaviours", Computer Science and Information Systems, vol. 12, no. 4, pp. 1171–1191, 2015.

[5] A. Eldawy, M. F. Mokbel, and C. Jonathan, "RadoopViz: A MapReduce Framework for Extensible Visualization of Big Spatial Data", p. 12.

[6] Z. Liu, B. Jiang, and J. Heer, "imMens: Real-time Visual Querying of Big Data", Computer Graphics Forum, vol. 32, no. 3pt4, pp. 421–430, Jun. 2013.

[7] Y. Xu, W. Zhou, B. Cui, and L. Lu, "Research on performance optimization and visualization tool of Hadoop", in 2015 10th International Conference on Computer Science and Education (ICCSE), Cambridge, United Kingdom, 2015, pp. 149–153.

[8] M. Mani and S. Fei, "Effective Big Data Visualization", in Proceedings of the 21st International Database Engineering and Applications Symposium on - IDEAS 2017, Bristol, United Kingdom, 2017, pp. 298–303.

[9] J. Zhang, M. L. Huang, W. B. Wang, L. F. Lu, and Z.-P. Meng, "Big Data Density Analytics Using Parallel Coordinate Visualization", in 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China, 2014, pp. 1115–1120.

[10] P. Godfrey, J. Gryz, and P. Lasek, "Interactive Visualization of Large Data Sets", IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 8, pp. 2142–2157, Aug. 2016.

[11] C. D. Stolper, A. Perer, and D. Gotz, "Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics", IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 12, pp. 1653–1662, Dec. 2014.

[12] N. Elmqvist and J.-D. Fekete, "Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines", IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 3, pp. 439–454, May 2010.

[13] N. Bikakis, "Big Data Visualization Tools", arXiv:1801.08336 [cs], Jan. 2018.

[14] N. Bikakis, G. Papastefanatos, M. Skourla, and T. Sellis, "A Hierarchical Aggregation Framework for Efficient Multilevel Visual Exploration and Analysis", arXiv:1511.04750 [cs], Nov. 2015.

[15] B. Chandramouli, J. Goldstein, and A. Quamar, "Scalable progressive analytics on big data in the cloud", Proceedings of the VLDB Endowment, vol. 6, no. 14, pp. 1726–1737, Sep. 2013.

[16] C. M. Li and F. Manya, "MaxSAT, Hard and Soft Constraints," in Handbook of satisfiability, vol. 185, Amsterdam, The Netherlands, 2009, pp. 613–631.

[17] V. Bharanipriya and V. K. Prasad, "WEB CONTENT MINING TOOLS: A COMPARATIVE STUDY", p. 6.

[18] A. K. Sharma and P. C. Gupta, "Study and Analysis of Web Content Mining Tools to Improve Techniques of Web Data Mining", vol. 1, no. 8, p. 7.

[19] S. Ahmed, M. Usman, J. Ferzund, M. Atif, A. Rehman, and A. Mehmood, "Modern Data Formats for Big Bioinformatics Data Analytics", International Journal of Advanced Computer Science and Applications, vol. 8, no. 4, 2017.

[20] R. Agrawal, A. Kadadi, X. Dai, and F. Andres, "Challenges and opportunities with big data visualization", in Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems - MEDES '15, Caraguatatuba, Brazil, 2015, pp. 169–173.

[21] M. Hussain Iqbal, and T. Rahim Soomro, "Big Data Analysis: Apache Storm Perspective", International Journal of Computer Trends and Technology, vol. 19, no. 1, pp. 9–14, Jan. 2015.

[22] C.-H. Su, W.-C. Huang, V.-D. Ta, C.-M. Liu, and S.-L. Peng, "Exploiting a Cloud Framework for Automatically and Effectively Providing Data Analyzers", in 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), Kanazawa, 2017, pp. 231–236.

[23] D. Myers and J. W. McGuffee, "Choosing scrapy", Journal of Computing Sciences in Colleges, vol. 31, pp. 83–89, 2015.

[24] J. Wang and Y. Guo, "Scrapy-Based Crawling and User-Behavior Characteristics Analysis on Taobao", in 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Sanya, China, 2012, pp. 44–52.

[25] M. Acheli and S. Khouri, "Big Data : définition, applications et outils" Developpez.com. [Online]. Available: https://mehdiacheli.developpez.com/tutoriels/bigdata/introduction-definitions-applications-outils. [Accessed: 10-Jan-2019].

[26] Bl.ocks.org. (2019). "Interactive Tree Structure in d3.js". [online] Available at: http://bl.ocks.org/alexalittle/e2f58a7c28fe51162e8f [Accessed 17 Feb. 2019].