An Approach for Semantic Mapping of Heterogeneous Big Data Based on Domain Ontology and the Hadoop HDFS/SPARK Ecosystem

Mohamed OUBEZZA1, Ali EL Hore2, Jamal EL Kafi3

1 LAROSERIE Laboratory Department of Computer Sciences
Faculty of Sciences, El-Jadida, Morocco

2 DIS Laboratory Department of Computer Sciences
Faculty of Sciences, El-Jadida, Morocco

3 LAROSERIE Laboratory Department of Computer Sciences
Faculty of Sciences, El-Jadida, Morocco

IJASR 2019
VOLUME 2
ISSUE 6 NOVEMBER - DECEMBER **ISSN: 2581-7876**

Abstract – Semantic interoperability of big data is a challenge for researchers today. Massive data are characterized by 5V: Volume, Variety, Velocity, Veracity and Visualization. These aspects make data processing very resource-intensive and time-consuming for the preparation, processing, verification, visualization and analysis of results.

The fourth V, Veracity, recently added to the aspects of Big Data, remains essential for effective data exploitation, especially for social data and media data where not all data to be processed is always true, taking into consideration the veracity of data requires even more processing and therefore more time.

Current research focuses on a single aspect and deals with it, regardless of other aspects, so the purpose of our work is to provide a system for semantic mapping of heterogeneous massive data from multiple sources, cleaning of this data, construction of knowledge in RDF format, verification of its veracity and finally the construction of the knowledge base and inference of new knowledge.

In this paper we present an evolving, incremental and distributed Framework for the processing of massive heterogeneous structured, semi-structured or unstructured data.

Keywords: Semantic interoperability, Domain Ontology, Hadoop, SPARK, SPARQL, IDIM, RDF, OWL

1 INTRODUCTION

Today, data are everywhere, expressed in different ways, but of importance to decision-makers. Internet users, for example, generate a large amount of data every second in social media or in personal blogs, this data, generated in several formats, must be analyzed in real time.

The semantic interoperability of massive heterogeneous data from multiple sources and in multiple formats has always attracted researchers, especially in the fields of Social Business Intelligence.

In this article, we will provide an incremental and distributed Framework that ensures the semantic interoperability of big heterogeneous data based on domain ontology and parallel data processing. Our approach is based on the Hadoop HDFS and SPARK ecosystem.

Our solution consists of three subsystems which are, in order: 1) Import data: input data can be structured according to a relational schema expressed by UML, semi-structured according to an XML schema or unstructured in flat text format. 2) Preparation of the triplets: the imported data will be transformed into RDF triplets and processed to simplify the inference. 3) Inference: query processor to execute SPARQL queries to extract information from the system.

The article is divided into four parts, the first part presents preliminaries and tools used in the architecture, the second part presents the state of the art, the third part presents the proposed solution and its subsystems; and the last part con-cludes the work and presents some perspectives.

2 PRELIMINARIES

2.1 Ontology

Several definitions are provided for ontology, the one retained is Gruber's [1]: an ontology is an explicit specification of a conceptualization.

Domain ontology facilitates the understanding of a domain and the sharing of knowledge in that domain

2.2 Ontology Web Language

Ontology Web Language (OWL) [2] is the standardized language for representing an ontology. OWL is derived from RDFS and allows, in addition to structuring knowledge, to represent advanced and complexe concepts such as restrictions, joins, union.

OWL is written in XML and facilitates the integration of heterogeneous information from different sources.

OWL thus makes it possible to describe the meta-data, this description is relative to the desired level of expressiveness of OWL, which gave rise to the 3 under language of OWL:

- OWL-lite: the lowest level of expressiveness, thus providing support for users basic needs.

- OWL-DL: the maximum level of expressiveness ensuring complementarily and decidability.

- OWL-Full: the highest level of expressiveness.

The most appropriate sub-language for our case is OWL-DL (Description Logics).

2.3 Ressource Description Framework (RDF)

Semantic web knowledge is represented in RDF. RDF [3] describes the resources and specifies the relationship between them. Each data that has an URI (Uniform Resource Identifier) can be stored in RDF.

The unit of an RDF is the triplet; a triplet is made up of: a subject, a predicate and an object.

2.4 Hadoop Distributed File System (HDFS)

Apache Hadoop [4][5][6] is a platform for distributed processing of very large datasets in clusters. It allows applications to run with thousands of nodes and pytabytes of data.

Hadoop is derived from Google's MapReduce programming model and Google File System (GFS).

Hadoop Distributed File system (HDFS) is Hadoop's distributed file system, providing fault tolerance and a very high rate of access to application data. HDFS allows data to be stored on thousands of servers, following the Master/Slave architecture, the large data is automatically divided into pieces managed by the different Hadoop cluster nodes.

In our work, the MapReduce framework is replaced by the SPARK framework.

2.5 Apache Spark

Apache SPARK [7] is a big data processing framework for performing complex analyses on a large scale. According to Apache [7], SPARK exceeds Map Reduce in terms of speed; it is 100 times faster for in-memory processing and 10 times fast-er for disk processing.

The secret of SPARK's speed is that it executes programs in memory and not on disk as Map Reduce does.

The experiments performed [8][9] also give SPARK the advantage, with a rate that can reach 5X. SPARK discards data in Resilient Distributed Datasets (RDD), an RDD is a distributed memory abstraction that allows in-memory computing to be performed in a fault-tolerant manner.

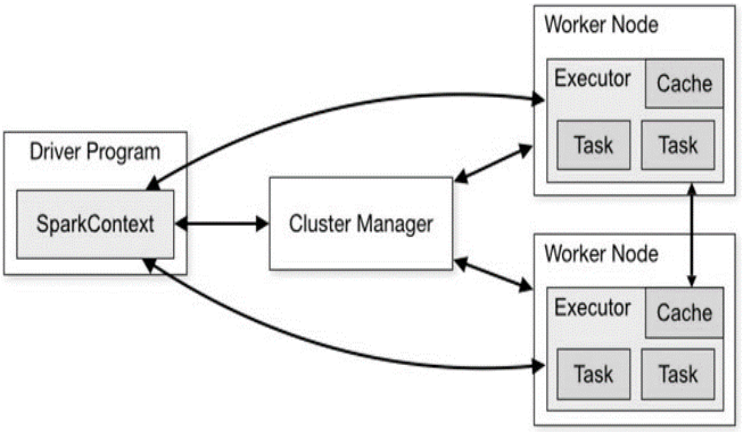

Fig 1 Apache Spark Architecture [7].

Spark consists of one master node and several worker nodes in a cluster. When the master node receives the task, it loads the corresponding data in RDD format. A dataset constructed in the RDD format consists of partitions divided into several nodes. Partitioned partitions are stored in each worker node memory and operations are performed. Fig 1 shows the Apache Spark architecture.

2.6 Apache HBase

Apache HBase [10] is a distributed and scalable database, which allows non-relational data to be stored and provides real-time cluster access to very large tables (Billions of rows and millions of columns)

The choice of HBase is justified by its functionalities in linear scalability, automatic spitting support, and backup manage-ment.

We will use HBase for RDF storage and as a target for SPARQL queries.

2.6 SPARQL

SPARQL Protocol And RDF Query Language [11] is a query language and protocol that allows to search, add, modify or delete RDF data available in an RDF stock. Since 2013, SPARQL has become an official recommendation of W3C.

3 STATE OF THE ART

3.1 Data import and preparation

Knowledge extraction depends on the types of data to be processed, so several approaches exist depending on the level of data structuring:

- Structured data in relational database: Several approaches exist for extracting knowledge from relational databases, the most appropriate and recommended by W3C is RDB direct mapping [15].

- Semi-structured data via XML/XMLS: to convert semi-structured data into RDF, ReDeFer [16] was adopted as the most widely used solution.

- Unstructured data: Extracting knowledge from unstructured data requires the use of NLP techniques, the approach that will be adopted and the one presented in the work of Sudip Mittal et al[17] which passes through the modules: 1) Extractor, 2) Assessor and 3) Knowledge Base Creator.

3.2 Reasoning and inference

Existing work on the reasoning process uses RDF closure Web PIE [14] this method has the disadvantage of allowing the use of duplicate values, consumes more disk space, and requires more time for calculation.

The IDIM Method presented in [12][13], based on the TIF and EAT sub graphs, saves execution time and storage space. But it only processes RDFS Meta data and not OWL.

All these methods do not process OWL Data, therefore do not benefit from the level of expressiveness provided by OWL. Also they do not process the veracity of the data, and therefore false data enters the inference process, which implies unrelia-ble results.

4 PROPOSED APPROACH

4.1 Incremental and Distributed Inference Method

An inference method is called incremental if it has the ability to take into account the incoming data in real time, i.e. when the new data arrives it will not need to redo the entire calculation to take these data into account.

Distributed property reflects the ability to distribute processing over multiple nodes. A method that is incremental and distributed in this way provides a reasonable processing time.

4.2 Transfert Inference Forest (TIF)

The TIF inference transfer forest presented in Liu's work[12][13], are RDF sub graphs of the global ontology, so when the query processor looks for an element, it looks in the TIF sub graphs, if the element is not present in the TIF tables but present in OWL, the processor adds this element to the appropriate TIF sub graph. So the next processing will require less time, since the processor will only search in the TIF sub graphs.

The TIF contains only the ontological triples, i.e. the triples on which inferences can be derived. The TIF sub graphs (or tables) depend on the type of the OWL relationship, so the TIF sub graphs set up are: PTIF (Property TIF), CTIF (Class TIF) and DRTF (Domain / Range Transfer Forest).

4.3 Transfert Inference Forest (TIF)

The EAT subgraph [12][13] is also a subgraph of the global ontology, but it contains only the triple affirmation, a triple is called affirmation if it is not ontological, which cannot be derived from the other triple affirmation.

EAT is also separated into PEAT (Property EAT) and CEAT (EAT Class).

4.4 Data Flow of the proposed architecture

Our solution consists of three essential steps: 1) Data integration, 2) Triple preparation and verification of veracity and 3) Reasoning and Inference.

The data flow of our approach is as follows:

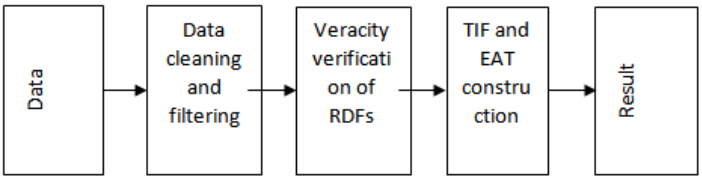

Fig 2 Data Flow of the proposed architecture

The system input data is heterogeneous and can be in different formats: structured (DBMS), semi-structured (XML) or un-structured.

The method of integrating these data depends on their type. Thus 3 modules have been implemented: Structured data integration module, semi-structured data integration module and unstructured data integration module.

Data integration is done according to domain ontology, creates raw RDFs triple, i.e. they still need to be processed to prepare them for the query pocessor.

These RDFs triple will then be filtered, cleaned, validated according to their veracity and indexed by replacing the URI with an internal Identifier unique in the system, to reduce the storage space required for the triplets.

Once validated, the triples are classified into ontological and affirmation triples, and according to the type of triple will be stored in TIF tables or EAT tables.

SPARQL queries will be launched on the triples stored in TIF and EAT tables.

The RDF triplets at their storage will have an internal ID formed by the source ID and the triplet ID, each data source has its identifier stored in the HBASE database.

The verification of the veracity of RDF triplets is based on the expert's inputs, the forms of the expert's intervention in our system are:

1. The expert provides information about the classification of sources in the HBase database, so if two RDFs are in con-flict, the one with the highest level of source will be given first priority. And the other one will be removed from the inference base. The source of an RDF during processing will be identified by the beginning of the RDF ID.

If the sources have the same level, the number of occurrences of each RDF is carried out, the RDF with the highest number of occurrences will remain for further processing..

2. Experts also enter a threshold for judging RDFs, so if the threshold is set at 5, any RDF present less than 5 times will be judged as uncertain.

3. A blacklist of RDFs is also available in the HBase database. Each RDF in the list will be automatically ignored.

4.5 Global Architecture

The architecture of the solution is presented in Fig 3. In this architecture, there are two different and complementary processes: 1) data import and preparation and 2) inference on the RDF stock. Both processes execute their processing on the Hadoop HDFS/SPARK framework.

1) Data Integration and knowledge generation:

This process is generated by the existence of data to be imported, as soon as the system receives data, it will automatically pass to the integration module, and depending on the type of data the appropriate sub-module takes care of the data integra-tion. The integration module is based on domain ontology, stored in OWL, to extract concepts and relationships between them.

The import uses Hadoop HDFS and SPARK for data processing.

Then the RDF duplicates will be removed, once the RDFs are cleaned, they will move to the veracity module which verifies using external resources and expert veracity entries, this module is presented in detail in [19].

Finally the verified RDFs will be compressed by replacing the URI, which is in String format, with a system-internal ID, the URI mapping and ID is stored in HBase's URI_ID table.

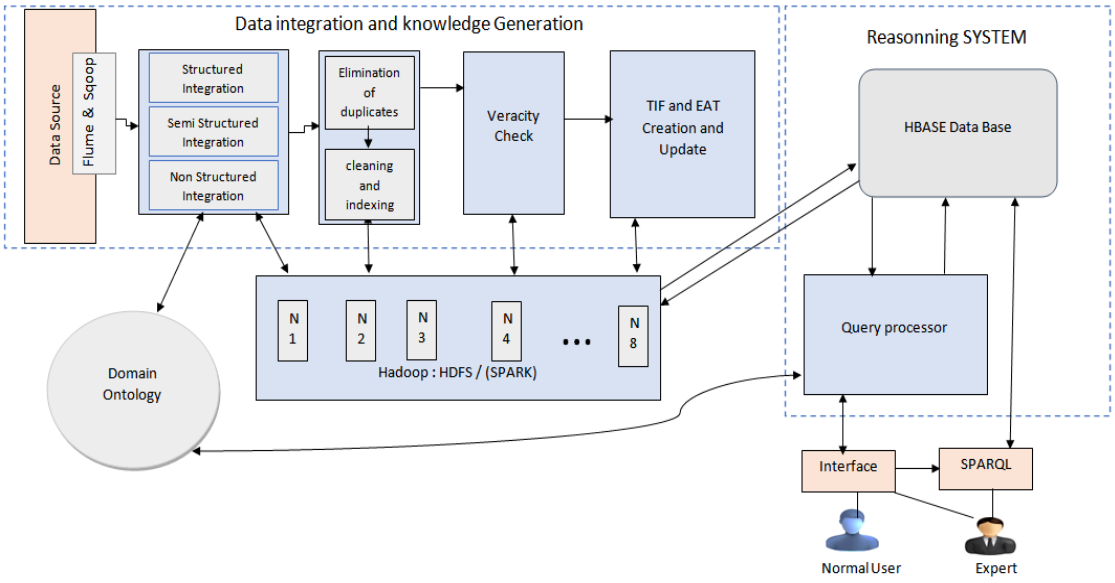

Fig 3 the global architecture of our approach

The resulting RDFs will be used to create or update the TIF and EAT sub graphs.

1) Reasoning System:

The reasoning and inference process is generated by a request from the end user; the inference method used is already presented in our previous work [18], the request can be:

1) SPARQL request entered by the expert, this request will be executed directly on the Apache HBase database.

2) A request stored in the system and launched on the graphical user interface by a normal user, this request will follow the same path as the previous one.

3) The last type of the query is the one entered by a user and which requires be parsing and converting by the query processor. Before being executed in the HBase database. The query processor is based on domain ontology for inference rules, constraints, and junctions.

5 CONCLUSION AND PERSPECTIVES

In this paper we have presented a scalable, incremental and distributed Framework based on the Hadoop HDFS and Apache SPARK infrastructure, which allows the semantic mapping of heterogeneous structured, semi-structured and unstructured data. This mapping is ensured by a domain ontology, which makes it possible to create RDF knowledge. The RDF triplets will then be cleaned from duplicates, and indexed. Then we check the veracity of the data, once the RDF triplets are verified, we create the TIF and EAT trees from these triplets. These trees will be stored in the HBase database, this database will be the target of SPARQL queries initiated by the expert or built from the interface by the query processor.

ACKNOWLEDGMENT

This work has been supported by the National Centre for Scientific and Technical Research of Morocco.

REFERENCES

[1] T. Gruber "A Translation Approach to Portable Ontology Specifications". Knowledge Acquisition, 1993, 5, pp. 199-220

[2] http://www.w3.org/TR/OWL (accessed 07 November 2019)

[3] http://www.w3.org/TR/rdf-mt/ (accessed 07 November 2019)

[4] http://hadoop.apache.org (accessed 07 November 2019)

[5] Aditya B. Patel, Manashvi Birla et Ushma Nair "Addressing Big Data Problem Using Hadoop and Map Reduce" IEEE, 2012

[6] Vasantharaja V and Dr. M. Sai Baba "Parsing and Mapping of OWL Ontology Using MapReduce into Hadoop", ResearchGate, 2016.

[7] http://spark.apache.org/docs/latest/

[8] Satish Gopalani et Rohan Arora "Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means" International Journal of Computer Applications (0975 – 8887), 2015

[9] Juwei Shi, Yunjie Qiu, Umar Farooq Minhas, Limei Jiao, Chen Wang Berthold "MapReduce vs. Spark for Large Scale Data Analytics", VLDB 2015.

[10] http://hbase.apache.org (accessed 07 November 2019)

[11] http://www.w3.org/TR/sparql11-query/ (accessed 07 November 2019)

[12] Bo Liu, Kerman Huang, Jianqiang Li et MengChu Zhou "An Incremental and Distributed Inference Method for large scale Ontologies based on MapReduce Paradigm" IEEE, 2016

[13] Dibya Raj Ghosh et Poovammal E "Ontology Based Semantic Web on Hadoop Platform" IEEE 2016

[14] Jacopo Urbani, Spyros Kotoulas, Jason Maassen, Frank Van Harmelen, Henri Bal "WebPIE: A Web-scale Parallel Inference Engine using MapReduce", Elsivier 2011

[15] https://www.w3.org/TR/rdb-direct-mapping/ (accessed 07 November 2019)

[16] http://rhizomik.net/html/redefer/ (accessed 07 November 2019)

[17] Sudip Mittal, Karuna P. Joshi, Claudia Pearce and Anupam Joshi " Parallelizing Natural Language Techniques for Knowledge Extraction from Cloud Service Level Agreements." IEEE 2015.

[18] Oubezza, M., El Hore, A. and El Kafi, J. (2019) 'An incremental and distributed inference method for large-scale ontologies over SPARK', Int. J. Cloud Computing, Vol. 8, No. 2, pp.140–149.

[19] Oubezza Mohamed, Ali El Hore and Jamal EL Kafi (2019) 'An Enhanced Framework to Ensure Big Data Veracity in Social Business Intelligence', International Research Journal of Advanced Engineering and Science, Vol.4, No. 4.

BIOGRAPHY

Mohamed OUBEZZA is an Engineer in Information Systems from the National School of Applied Sciences of Tangier, Morocco. He has accumulated professional experience in the field of IT at INWI (Telecommunications Operator) as Head of the Intelligent Network Team. He is currently a trainer at the Office of Vocational Training and Work Promotion. A member of the LAROSERIE DIS laboratory team at the Faculty of Sciences of El-Jadida, his areas of interest are knowledge management, information extraction, semantics, knowledge discovery, semantic web technologies, rule-based reasoning.

Ali EL HORE is a Doctor from University Toulouse 1 Capitole –- France in 2006. He was previously a Professor at the Faculty of Toulon in France. He is currently a Professor at the Faculty of Sciences of the University of El-Jadida in Morocco and President of the DIS team. He is the organizer of several international scientific events. His research focuses on Knowledge management, text mining, information extraction, ontology, knowledge representation, semantics, knowledge discovery, semantic web technologies, and reasoning.

Jamal El KAFI is a Doctor of the University of Bordeaux – France in 1990. He is currently a Higher Education Professor at the Faculty of Sciences of El-Jadida. He is the Director of LAROSERIE Laboratory and Head of Master of Business Intelligence and Big Data Applications. He is an expert consultant in Information Systems Engineering and Complex Project Management. His areas of researches are Artificial intelligence, E-learning, knowledge management, and information systems management.