ISSN 2347 - 3983
Volume 7, No.7 July 2019
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter03772019.pdf
https://doi.org/10.30534/ijeter/2019/03772019

An Evaluation and Comparative study of massive RDF Data management approaches based on Big Data Technologies

Mouad Banane[1], Abdessamad Belangour[2]
[1]Hassan II University, Morocco, mouad.banane-etu@etu.univh2c.ma
[2]Hassan II University, Morocco, belangour@gmail.com

ABSTRACT

Today, many researches are focused on the use of Big Data technologies for managing large volumes of Semantic Web data. The use of Big Data technologies such as NoSQL database management systems guarantees the scalability and high availability of Web data. The amount of data Web increases day after day and in a way excessive. To fully exploit these data and its metadata. This requires this technology of the Semantic Web to be a scalable and high performance at the level of the storage of these Web data. Many research efforts have been devoted to create and develop a distributed RDF data management system. To achieve the scalability and high performance, these systems are implemented on the basis of the Management technology of Big Data called NoSQL. In this paper, we present a comparative study of existing systems dedicated to managing large volumes of RDF data, which are generally based on NoSQL database management systems.

Key words : RDF, SPARQL, Semantic Web, NoSQL, Big Data.

1. INTRODUCTION

Today, many researches is focused on the use of Big Data technologies for managing large volumes of Semantic Web data[21,22]. the use of Big Data technologies such as NoSQL database management systems guarantees the scalability and high availability of Web data. RDF the Resource Description Framework [20] is a graph model whose unit of information is the triplet, used to describe any type of resource on the Web. It is at the base of the Semantic Web. The RDF syntax standardizes the descriptions to allow machines to sort and exchange more efficiently the metadata specific to each digital resource (article, table, chart, digital book, photo, animation, sound file, video, software ...).

An information system can be perceived along three axes: data, processing and communications. For fifty years, the whole of these axes is the subject of intensive research to improve processing times, management of physical space, research and the transmission of information. For the past 20 years, some have identified a fourth axis of representation of the information system which is the semantic axis. This axis, independent of the physical constraints of information management attempts to solve the problems of semantic heterogeneity We speak then of semantic graph, ontology and triplestore.

The Resource Description Framework (RDF) is developing for the enrichment of web-based resources with detailed descriptions and it increases the ease of automatic processing of web resources. Descriptions may be characteristics of resources, such as the author or the content of a website. These descriptions are metadata. Enriching the Web with metadata allows the development of what is called the Semantic Web. The RDF is also used to represent semantic graphs corresponding to a modeling of specific knowledge. RDF files are usually stored in a relational database and manipulated using SQL or derived languages like SPARQL[25].

The structure of an RDF document is very simple. it contains triples and each triple contains three elements (subject, predicate and object):

- The "subject" represents the resource to be described;
- The "predicate" represents a type of property applicable to this resource;
- The "object" represents a datum or another resource: it is the value of the property.

The subject, and the object in the case where it is a resource, can be identified by a URI or be anonymous nodes. The predicate is necessarily identified by a URI. RDF documents can be written in different syntaxes, including XML. But RDF itself is not an XML dialect. It is possible to use other syntaxes to express triples. RDF is simply a data structure consisting of nodes and organized into graphs. Although RDF / XML - its XML version proposed by the W3C - is only a syntax (or serialization) of the model, it is often called RDF. Abuse of language refers to both the triple graph and the XML presentation associated with it. An RDF document thus formed corresponds to a labeled oriented multigraph. Each triplet then corresponds to an oriented arc whose label is the predicate, the source node is the subject and the target node is the object.

NoSQL databases have been designed to solve the problems of volume, multi-source and multi-format data processing in big data environments. What we want to do with NoSQL is to reduce the complexity of the query language, simplify the

database architecture, and find a way to store the database on as many computers as possible. inexpensive depending on the needs. Thus, a NoSQL database is a distributed database for distributing computing and data load dynamically, nonrelational, preferring the management of a gigantic table to that of many interdependent tables. NoSQL databases can be categorized into four categories: Key/value-oriented database, column-oriented database, graph-oriented-database and document-oriented database.

This document is organized as follows: after the first section introduction, the second section exposes some existing related works this topic, the section tree presents and exposes RDF data management systems based on the NoSQL. Section IV presents a Review of RDF data management systems that use NoSQL database. Finally, section V concludes this work and suggests some future works.

.

2. RELATED WORK

The development of the Web has led to the explosion of the volume of accessible data. The Web can be seen as a huge source of unstructured data. The relational model turns out to be unsuitable for the management of this data[16].

The exploitation of resources on the Web involves the development of new technologies. The goal of these technologies is to make the information contained in web documents usable by software programs and agents, through a formal metadata system. All of these technologies and the data they structure are called the Semantic Web[23].

RDF data is stored in a specific database named triplstore. A triplestore is a database specifically designed for storing and retrieving RDF data. Like a relational database, a triple store stores data and retrieves it through a query language. But unlike a relational database, a triplestore stores only one type of data, the triple, in the form of (subject, predicate, object). A triplestore does not need an initialization phase to save new data, that is, it does not need to create tables like in a relational database. In addition, a triple store is optimized for storing a large number of triplets and for retrieving these triplets using the SPARQL query language.

Recently, much of the research is focused on the use of Big Data technologies for managing large volumes of Semantic Web data, we mention Jena-HBase [1], Rainbow [2], Hive-HBase [3], MapReduce-HBase [4], CumulusRDF [5], Rya [6] , H2RDF [7], SPIDER [8], SHARD [10].

Therefore, this paper presents our work that compares and evaluates the NoSQL stores for RDF data management, the first work done by Mauroux et al [9], but it has only evaluated five RDF stores based on NoSQL: CumulusRDF, 4store, Hive+HBase, Jena + HBase, and Couchbase [9], using standard RDF benchmarks on two deployments modes single-machine and distributed deployments. We also cite the survey [10] that classifies RDF data management systems as: 3store, 4store, Virtuoso, RDF-3X, Hexastore, Jena Apache, SW-Store, BitMat, AllegroGraph, and Hadoop/HBase. In this article we will compare 10 RDF stores based on NoSQL technology. In [16] the authors reviewed the storage of large RDF data in NoSQL systems based on the different NoSQL models. In the SPARQL2Hive[23] approach, which presents a technique for transforming SPARQL queries into an Apache Hive program, this solution can be used for RDF data stored in NoSQL databases. In this paper, we present a comparative study of existing systems dedicated to managing large volumes of RDF data, which are generally based on NoSQL database management systems.

3. DISTRIBUTED MANAGEMENT SYSTEM FOR LARGE RDF DATA

The In this section, we provide a detailed description of each RDF data storage management system.

3.1 Jena-HBase

Jena-HBase is a triplestore implemented on the HBase[13] NoSQL database. This triplestore uses the Jena Framework for querying RDF data. HBase is a distributed, column-oriented, scalable and fault-tolerant NoSQL[19] database where workload in terms of memory and computation (CPU) as well as storage is distributed on all machines in the HBase cluster. This NoSQL system is inspired by the BigTable's work [12], led by Google. It is expanded on top of the HDFS(Hadoop Distributed File System)[24] file system. It allows random read / write access in real time to a very large data set. In HBase Columns are grouped into column families, HBase provides an extra dimension to each cell named timestamp. Jena-HBase builds a structure of three index tables that contains: OSP, SPO and POS. For example, the SPO table is used to store triples (S?), and (SP?), This technique allows fast joins and reduces read/write on disk-level. The Jena framework is the SPARQL query engine, Jena by default provides a technique that allows fast join.

3.2 Rainbow

Also based on HBase, Rainbow is a scalable triplestore, for indexing Rainbow also uses the same technique of three triples index SPO, POS, and OSP. It benefits from the advantages provided by the HBase system such as distributed memory storage, high availability, and fault tolerance. The Rainbow query layer is independent of the choice of SPARQL query processing engines, but the processing mechanism used in Rainbow is Sesame [12].

3.3 Hive-HBase

Hive-HBase is an RDF triplestore also implemented on the basis of HBase. In this system the subject used as the row key for each row of the table, the predicate is the column, and for the object, its value is stored in the corresponding row and column. The querying tool used in this approach is Hive [14,23], the work consists of converting SPARQL queries into

HiveQL queries which the query language of Hive. The use of Hive has. a speed advantage at join level, in addition, Hive allowing analysis and synthesis of data. Hive is used in the Hadoop ecosystem to speed up requests, it's faster than running MapReduce jobs.

3.4 CumulusRDF

The underlying storage component used in this system is Apache Cassandra[17] which is a NoSQL database belonging to the family of column-oriented databases. Cassandra is a project launched by Facebook. Cassandra is widely used by many applications managing large amounts of data and a large number of queries. The storage nodes in Cassandra are organized as a peer-to-peer network in which a node can both issue and respond to requests. In order to query stored data, a client can connect to any node in the system. For the indexing structure of CumulusRDF, it uses four tables index which are the following triple patterns: SPO, PSO, OSP, CSPO. Key/value data storage and key hashing-based search technology replace the use of dictionaries to map RDF terms. At query level CumulusRDF uses Sesame, which is a SPARQL query processor, and its role is to store and translate SPARQL queries into index searches on indices in the Cassandra database.

3.6 Rya

Rya is a distributed and scalable RDF data management system. Based on Accumulo[18], Rya enables the management of billions of triplets using its distributed column storage technique and indexing scheme. The indexing scheme of Rya is the three tables index: SPO, POS, OSP. Accumulo is open source, distributed, scalable and column-oriented NoSQL database management system. Accumulo is developed based on Bigtable as HBase. In the query part Rya uses OpenRDF Sesame, this query mechanism allows quick and easy access to RDF data.

3.7 H2RDF

H2RDF is a distributed RDF data store, at the H2RDF processing level uses the MapReduce [22] framework and in the storage layer, it uses a NoSQL system. Multiple joins are among the advantages of this solution for the speed of query responses, but generally, the execution of MapReduce jobs takes a longer time than the execution of the same query, but in HiveQL.

3.8 SPIDER

SPIDER is a scalable query processing system for RDF data, implemented on the MapReduce framework, Spider distributed data to servers using MapReduce. SPIDER does not take all the functionality of SPARQL but it converts SPARQL queries into MapReduce jobs.

3.9 SHARD

The SHARD triplestore is considered a distributed, scalable and robust RDF store. At query level SHARD uses the MapReduce parallel/distributed computing framework. And for storage SHARD retains the data in flat files in the Hadoop Distributed File System (HDFS), so that each line of this file at three represents all the triples associated with a different subject.

3.10 MapReduce-HBase

This scalable RDF triplestore is based on the HBase database. This solution adopts the Hexastore [15] approach and benefits the advantages provided by HBase. To ensure better indexing, all combinations of possible RDF triple are indexed through the creation of an HBase table for each triple pattern, so the tables are: PSO, SOP, SPO, POS,OPS and OSP. For querying RDF data this approach uses the MapReduce processing framework, i.e. SPARQL queries are converted to MapReduce jobs.

3.11 Couchbase

This RDF data store is based on Couchbase which is an open source project, distributed and is a document-oriented database. Couchbase is a robust and advanced solution that manages concurrent access to documents, scaling, replication, load balancing, fault tolerance, and backups. Documents inserted in Couchbase are stored in JSON format. The language created by Couchbase to query the database is a language inspired by SQL, called N1QL. The architecture of a Couchbase server is composed of two large parts, the first manages access and storage of data, while the second manages the administration of the Couchbase cluster. The technique used here is to load map RDF data into JSON documents. For indexing Couchbase uses the following three indexes: (? P?), (? O), and (? PO). At the query level, and for a distributed query Couchbase provides MapReduce views that are created over stored JSON documents, it implements the Jena framework interface to execute SPARQL queries.

4. REVIEW OF NOSQL-BASED RDF DATA MANAGEMENT SYSTEMS

Now, we review some of RDF systems which are based on NoSQL databases for storing a massive RDF data including : Jena-HBase[1], Rainbow [2], Hive-HBase [3], MapReduce-HBase [4], CumulusRDF [5], Rya [6] , H2RDF [7], SPIDER [8], Couchbase[9], SHARD [10]. The comparison is based on some criteria of database software such as :NoSQL database, NoSQL database model, database Licence, Index structure, Querying, Index triples, join optimization method, SPARQL Translate, and the Execution time. Table 1 illustrates a review of RDF data management systems that use NoSQL database.

Table 1: RDF stores loading time.

Name	**Jena-HBase**	**CumulusRDF**	**Hive-HBase**	**Couchbase**	**H2RDF**	**Rya**	**MapReduce-HBase**	**SPIDER**	**SHARD**	**Rainbow**
NoSQL DB	HBase	Cassandra	HBase	Couchbase	HBase	Accumulo	HBase	HBase	HDFS	HBase
NoSQL model	Column	Key/Value	Column	Document	Column	Column	Column	column	column	column
Licensce DB	Apache 2.0	Open source	Apache 2.0	Freemium	Apache 2.0	Apache 2.0	Apache 2.0	Apache 2.0	Open source	Apache 2.0
Index strecture	Three table Index	Four table Index	Two table Index	Three table Index	Three table Index	Three table index	Six table Index	-	Native indexing	Three table Index
Querying	Jena	Sesame	Hive	Jena	MapReduce	Open RDF sesame	MapReduce	MapReduce	MapReduce	Sesame
Index triples	SPO, POS, OSP	SPO,PSO, OSP, CSPO	SP?, ?P?	?P?, ??O, ?PO	SPO, POS, OSP	SPO, POS, OSP	SPO, PSO, OSP, SO, POS, PSO	-	-	SPO, POS, OSP
Join optimisation method	Jena-join-strategy	Sesame-join	Hive-tables-join	Jena-join-srtategy	None	Sesame-join	None	None	None	Sesame-join
SPARQL Translate	Yes	Yes	Yes	Yes	None	Yes	None	None	None	Yes
Execution time	Medium	Medium	Low	Medium	High	Medium	High	High	High	Medium
NoSQL DB	HBase	Cassandra	HBase	Couchbase	HBase	Accumulo	HBase	HBase	HDFS	HBase

We evaluate the loading time of the RDF store data presented in this study, for this we used the "DBLP 2017" dataset[21] with a size of 1GB and 882 Million Triples. We used the basic configuration of these systems that are available. Figure 1 presents the results obtained.

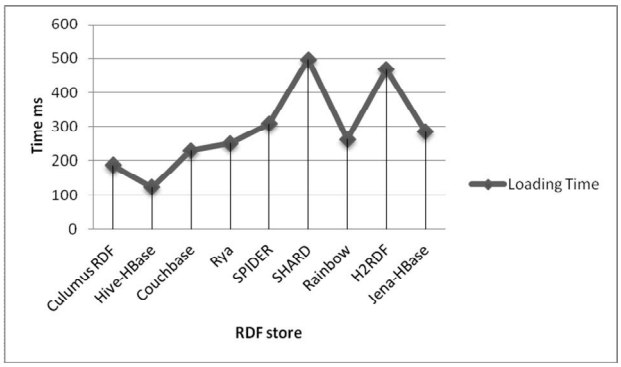

Figure 1: RDF Stores loading time.

Note that systems that use the MapReduce framework directly are longer at the query execution level than other systems that use HiveQL or Sesame, for example. At the indexing level most of the systems presented use an index structure of three tables: OSP, POS and SPO and especially systems based on column-oriented databases like HBase and Accumulo. The dynamism, the simplicity of a column-oriented NoSQL system like HBase is the reasons why the majority of these RDF data management systems are based on HBase.

5. CONCLUSION

Today, many researches is focused on the use of Big Data technologies for managing large volumes of Semantic Web data[21,22]. The use of Big Data technologies such as NoSQL database management systems guarantees the scalability and high availability of Web data. We presented this article with a comparison and evaluation of distributed RDF data stores based on NoSQL systems. The comparison revealed The comparison first revealed that NoSQL systems are a very good solution for the management of large volumes of RDF data, and secondly this variety of systems gives other technical advantages to the application and diploid level of these systems.

REFERENCES

1. Khadilkar, Vaibhav, et al. "**Jena-HBase: A distributed, scalable and efficient RDF triple store**." *Proceedings of the 11th International Semantic Web Conference Posters & Demonstrations Track, ISWC-PD*. Vol. 12. 2012..
2. Gu, Rong, Wei Hu, and Yihua Huang. "**Rainbow: A distributed and hierarchical rdf triple store with dynamic scalability**." *In 2014 IEEE International Conference on Big Data (Big Data)*, pp. 561-566. IEEE, 2014. https://doi.org/10.1109/BigData.2014.7004274
3. Haque, Albert, and Lynette Perkins. "**Distributed RDF triple store using hbase and hive**." *University of Texas at Austin* (2012): 139.
4. Sun, Jianling, and Qiang Jin. "**Scalable rdf store based on hbase and mapreduce**." *In 2010 3rd international conference on advanced computer theory and engineering (ICACTE)***,** vol. 1, pp. V1-633. IEEE, 2010. https://doi.org/10.1109/ICACTE.2010.5578937
5. Ladwig, Günter, and Andreas Harth. "**CumulusRDF: linked data management on nested key-value stores**." *In The 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2011)*, vol. 30. 2011.
6. Punnoose, Roshan, Adina Crainiceanu, and David Rapp. "**Rya: a scalable RDF triple store for the clouds**." *In Proceedings of the 1st International Workshop on Cloud Intelligence*, p. 4. ACM, 2012. https://doi.org/10.1145/2347673.2347677
7. Papailiou, Nikolaos, Ioannis Konstantinou, Dimitrios Tsoumakos, and Nectarios Koziris. "**H2RDF: adaptive query processing on RDF data in the cloud**." *In Proceedings of the 21st International Conference on World Wide Web*, pp. 397-400. ACM, 2012. https://doi.org/10.1145/2187980.2188058
8. Choi, Hyunsik, Jihoon Son, YongHyun Cho, Min Kyoung Sung, and Yon Dohn Chung. "**SPIDER: a system for scalable, parallel/distributed evaluation of large-scale RDF data**." *In Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 2087-2088. ACM, 2009. https://doi.org/10.1145/1645953.1646315
9. P. Cudré-Mauroux et al., « **NoSQL Databases for RDF: An Empirical Evaluation** », in *The Semantic Web – ISWC* 2013, vol. 8219, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, et K. Janowicz, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, p. 310‑325.
10. K. Rohloff et R. E. Schantz, « **High-Performance, Massively Scalable Distributed Systems using the MapReduce Software Framework: The SHARD Triple-Store** », p. 5.
11. J. Broekstra et A. Kampman, « **Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema** », p. 15.
12. F. Chang et al., « **Bigtable: A Distributed Storage System for Structured Data** », *ACM Trans. Comput. Syst.*, vol. 26, no 2, p. 1‑26, June 2008. https://doi.org/10.1145/1365815.1365816
13. N.Dimiduk , A. Khurana « HBase in Action »,2013.
14. « Apache Hive TM ». [Online]. Available on: https://hive.apache.org/. [Consulted on: 31-|July-2018].
15. C. Weiss, P. Karras, et A. Bernstein, « **Hexastore: sextuple indexing for semantic web data management** », *Proc. VLDB Endow.*, *vol*. 1, no 1, p. 1008‑1019, août 2008. https://doi.org/10.14778/1453856.1453965
16. M. Banane, A. Belangour, et L. E. Houssine, « Storing RDF Data into Big Data NoSQL Databases », **in Lecture**

Notes in Real-Time Intelligent Systems, 2017, p. 69‑78.
https://doi.org/10.1007/978-3-319-91337-7_7

17. M.Brown « **Learning Apache Cassandra : build an efficient, scalable, fault-tolerant, and highly-available data layer into your application using Cassandra** ». 2015.
18. « Apache Accumulo ». [Online]. Available on: https://accumulo.apache.org/. [Consulted on: 31-juill-2018].
19. Erraissi, A., & Belangour, A. (2018). **Data sources and ingestion big data layers: meta-modeling of key concepts and features**. *International Journal of Engineering & Technology*, 7(4), 3607-3612. https://doi.org/10.2139/ssrn.3185342
20. O.Lassila, R.Swick , « Resource Description Framework (RDF) Model and Syntax Specification». 1999.
21. Available on http://www.rdfhdt.org/datasets/, consulted 23/07/2018 download from http://downloads.linkeddatafragments.org/hdt/.
22. M. Banane , A. Belangour. RDFMongo: **A MongoDB Distributed and Scalable RDF management system based on Meta-model**. *International Journal of Advanced Trends in Computer Science and Engineering* Vol(8) 3. 734 – 741. https://doi.org/10.30534/ijatcse/2019/62832019
23. M.Banane, A.Belangour. **New Approach based on Model Driven Engineering for Processing Complex SPARQL Queries on Hive**. *International Journal of Advanced Computer Science and Applications(IJACSA)*, Volume 10 Issue 4, 2019.. https://doi.org/10.14569/IJACSA.2019.0100474
24. Erraissi Allae, and Abdessamad Belangour. « **Hadoop Storage Big Data Layer: Meta-Modeling of Key Concepts and Features** ». *International Journal of Advanced Trends in Computer Science and Engineering* 8, nº 3 (2019): 646-53. https://doi.org/10.30534/ijatcse/2019/49832019
25. J. Pérez, C.Gutierrez, M. Arenas « **Semantics and Complexity of SPARQL** ». *lecture Note in Computer Science, Springer 2006.* https://doi.org/10.1007/11926078_3