A SOA implementation to combine Spark and HDFS

Mint Mohamed Lemin Zeinebou, Noura AKNIN, Salihi Mohamed Lemin

Abstract— The lack of speed (running slowness) present a big problem especially when we have to analyze an incredible volume of data like Big Data. To avoid this situation we should choose the fastest and the most efficient tools and combine this tools, at the same time, in an heterogenic architecture that manages the exchange between the combined tools. Each of these tools must comply with a contract agreement to minimize Big Datas management delays and ensure a well-crafted analysis and data exploitation. In this paper we propose a new implementation of the service-oriented architecture to put Spark and Hadoop HDFS all together and make them compatible with the volume and variety of the analyzed data (Big Data).

Index Terms— Big data,DataNode , Hadoop, HDFS ,Hadoop MapReduce , Mule , NameNode , Namenode_Secondaire , REST ,SOA ,Spark.

1 INTRODUCTION

A Great evolution of the distributed application was recognized under the title of the service-oriented architecture. This architecture, that allows in the beginning to manage the interoperability and makes them syntactic or semantic and puts the heterogeneity to the different components of the information systems of any company, also decomposes the functionality into several functions which are services and manage all intersections between them, thus, allow a realization of a moving information system. Despite the difficulty of fulfilling a technical implementation of this architecture, several tools success this task. For example, MULE is an effective implementation of this architecture and also REST gives an achievement of the SOA using web services. But, in every implementation we cannot avoid the redundancy of services, which can be considered as a great opportunity for us. The fact of maintaining Frameworks is very important and necessary. Regarding the subject of management of the volume of Zettabytes, we must use Spark, MapReduce and HDFS several times as a service and replicate this usage according to the volume of Big Data and the variety of it, considering that Spark, MapReduce and HDFS cannot exceed the size of the data Petabytes. We will talk about each keyword: A.HDFS (Hadoop distributed file system) is a distributed storage system for reliably storing and streaming petabytes of both unstructured and structured data on clusters. HDFS has three classes of nodes in each cluster: 1. NameNode: responsible for managing the whole HDFS metadata like permissions, modification and access times, namespace and disk space quotas. The most important role is to support the Web-HDFS access from the client via the clusters public hostname..

• *Mint Mohamed Lemin Zeinebou ,LIROSA Laboratory ,Information Technology and ModelingSystems Research , Abdelmalek Essaadi University, Morocco,Email: zeinebou.l7aj@gmail.com* • *Noura AKNIN ,LIROSA Laboratory ,Information Technology and ModelingSystems Research ,Abdelmalek Essaadi University, Morocco Email: aknin@uae.ma and Salihi Mohamed Lemin ,SYSCM Laboratory,Faculty Of ScienceNKTT, UniversityNouakchott, MautaniaEmail: mlsalihi@gmail.com*

2. Secondary Namenode: responsible for check- ing the NameNodespersistent status and periodically down- loading current NameNode image and log files; it cannot play the role of the primary NameNode. 3. DataNode: responsible for storing the unstructured file data or other structured data such as spreadsheets, XML files, and tabseparated-value files (TSV) in which the geotagged datasets have been stored. HDFS stores these files as a series of blocks (the unit of storage), each of which is by default 64 MB (or 128 MB) in size [1]. Where there will be a huge number of blocks if we are working on the Big Data or on the Zettabyte. In addition, HDFS is used for not losing the data and must replicate each blocks according to a factor of replication that is often 3, which also increases the number of blocks. Otherwise there is an only one Namenode to manage the diffusion and replication of the blocks on Datanodes, although the Secondary Namenode takes the place of main Namenode if it is unavailable. So if we lose the Namenode and the Secondary Namenode we can no longer manage the different blocks or access to them, which involves difficulties to find the data [2] .B.MapReduce was developed and patented by Google to process extremely large datasets over a commodity computing cluster. It abstracts the difficulties of developing scalable distributed applications, such as fault tolerance and locality-aware data distribution. Such features, along with its simplistic programming approach, allow it to be used by any programmer. MapReduce works on a set of key/value pairs as an input. The programming task is simplified into two processing stages. The Map stage processes the input set and produces an intermediate set of key/value pairs. The key/value pairs are then grouped to be processed by the Reduce stage, which generates another set of pairs. The Map and Reduce stages can be as simple or complex as required, also composing chains of computations. MapReduce can be applied to a wide range of applications, Google Web Search being a notable example [3]. The figure 1 brings and describes the work of HDFS and MapReduce together.

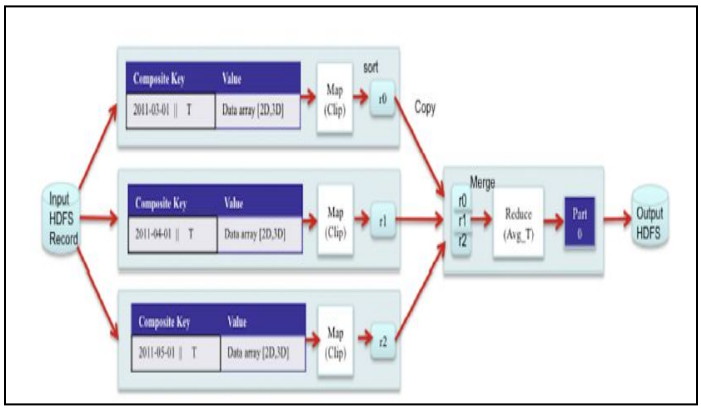

Fig. 1 HDFS and MapReduce

C. Apache Spark is a Framework open source of treatments of Big Data. Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing. It run pro- grams up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk. Spark offers over 80 high-level operators that make it easy to build parallel apps, Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming. It runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3.You can run Spark using its standalone cluster mode, on EC2, on Hadoop YARN, or on Apache Mesos. Access data in HDFS, Cassandra, HBase, Hive, Tachyon, and any Hadoop data source [5]. We want to take advantage of these frameworks thanks to the characteristics of the service-oriented architecture, D. The Service Oriented Architecture should be utilized as a business transformation tool for solving larger business needs, rather than strictly as an IT architectural initiative. Moreover, SOA should be seen as a means to drive organizational strategy that focuses on the alignment of business and technology for agility[6]. As well the business processes automation and IT support is often implemented by means of service-oriented architecture (SOA) [7] . Service Oriented Architectures (SOAs) define standard interfaces and protocols that allow developers to encapsulate information tools as services that clients can access without knowledge of, or control over, their internal workings. Thus, tools formerly accessible only to specialists can be made available now for all. In SOA, data and functional- ity are decoupled yielding to minimal dependencies among the service requesters and providers. The service oriented approach suits well the industrial operational environment as well as with service maintenance. Services that do not themselves hold any significant amounts of data, but transform it, can be implemented by the external service providers [8]. E. Big Data is defined as any collection of data sets which volume and complexity make data management and processing difficult to perform using traditional tools (i.e. handling N-dimensional data sets using plain text files and/or SQL databases). Those problems invest Big Data monoliths as much as ecosystems of small data causing major concern for most private and public data providers for which small quantities do not equal simpler management. Even though Big Data is usually associated with the LOD concept, it is generally comprised of linked and non-linked data, open and private data, and, as such, it is characterized as being composed of the three Vs: significant growth in the volume, velocity and variety of data. In this review, we include in the above definition of Big Data also the collection of technologies that cope with the effects of this abundance and heterogeneity, proposing solutions to meet the needs of a modern scientific community. Using Big Data involves many challenges. [3]. this 4V definition is widely recognized because it highlights the meaning and necessity of Big Data. Big data is a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex and of small quantities do not equal simpler management". Even though Big Data is usually associated with the LOD concept, it is generally comprised of linked and non-linked data, open and private data, and, as such, it is characterized as being composed of the "three Vs": significant growth in the volume, velocity and variety of data. In this review, we include in the above definition of Big Data also the collection of technologies that cope with the effects of this abundance and heterogeneity, proposing solutions to meet the needs of a modern scientific community. Using Big Data involves many challenges. [3]. This 4V definition is widely rec-ognized because it highlights the meaning and necessity of Big Data. Big data is a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex, and of a massive scale (Fig.2) [9]. Fig. 2 4V of Big DataThere is the challenge of managing large amounts of data (Big Data), which is getting increasingly larger because of cheaper storage and evolution of digital data and information collection devices, such as cell phones, laptops, and sensors. For example, Facebook, a social networking website, is a home to 40 billion photos, and Wal-Mart handles more than 1 million customer transactions every hour, feeding databases estimated at more than 2.5 petabytes [10]. A wide variety of technologies and heterogeneous architectures have been applied in the implementation of the Big Data use cases. The publications have mainly concentrated on describing architectures of individual contributions by large big data companies such as Facebook or LinkedIn [11]. The volume of Big Data is currently measured in petabytes, exabytes, or zettabytes, by 2020, the digital universe is expected to reach 44 zettabytes [12].

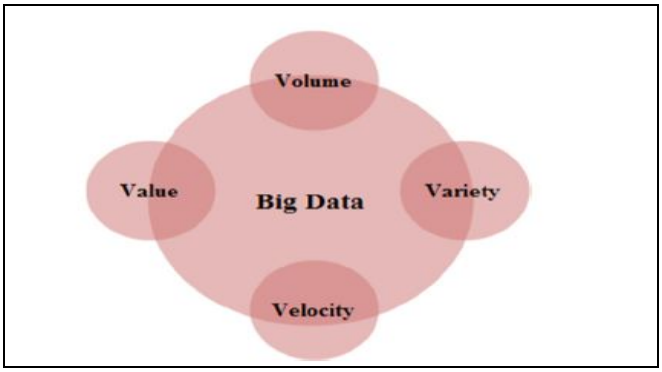

Fig. 2 4V of Big Data

2 MOTIVATION

Spark is much faster than Hadoop MapReduce but Spark still needs HDFS for data storage, The fact that we have to encounter the same Hadoop project problems for managing Big Data also with Spark the fact that we need to handle more data Pettaocte. At the same time we need Spark for some type of data variety such as graphs and type of batch source or real-time stream. In this paper, we propose a new implementation of the service-oriented architecture to be able to analyze the different types of data and considering a remarkable increase in volume of Big Data and the management time of this volume. We propose An SOA design such that each service must be HDFS and Spark to be able to handle the different types of data and any volume collected. In this design, exchanges between the two types of service are ensured thanks to the SOA feature. For the working period, we benefit firstly from the speed of Spark and its capacity to manage different varieties of data that MapReduce cannot manage. We put in place several services together each able to handle data petabytes and solve the problem and that of Big Data volumes.

2.1 Related work

In our previous article we have already proposed a new use of SOA to handle Big Data with the size of Zettabytes. This architecture is in the form of several Hadoop together more specifically MapReduce and HDFS. Each Hadoop is able to

handle Petabytes of data without exceeding these limits, we have just divided the collected data or the Big Data between the sets of Hadoop and thanks also to the characteristics of the SOA it is ensured that there is an exchange between the whole of Hadoop which make the short working time [2]. But for several other types of varieties this architecture cannot be considered. The previous article groups several Hadoop together while this one gathers several Spark together in order to be able to handle Big Data Regardless of its size.

2.2 Contribution

For any management of Big Data we need the very advanced news frameworks. The fact that Spark is the fastest at the moment and also provides several other possibilities like the management of the other types of Data source, makes the use of it much needed in this area. It is good that Hadoop MapReduce does the treatments in two phases Map and Reduce but for Spark all this will be done in a single step.

3 DESCRIPTION

It is true that the first architecture [2] that we proposed is able to handle the huge increase in data volume but Spark can handle other varieties more than MapReduce and in a shorter time period. The, the idea to improve this design and replace each Hadoop (MapReduce + HDFS) by Spark + HDFS. We need to combine several Frameworks together either Hadoop HDFS and Spark since all these software cannot exceed the size of the data Petabytes, we need to manage data Zettabytes that is the current size of Big Data, this combination must be in an architecture that Manages the intersections between the-services and ensures exchanges between them to facilitate the collection of data at the entry of the kernel of this architectu-reand to collect them at the end of processing, so that we will be able to manage several Petabytes of data until reaching Zettabytes. If we replace each Hadoop composed of two parts MapReduce and HDFS by Spark and HDFS (Fig. 3), this architecture is much faster than the first one, and it is also capable of handling very special types of data sources than the first architecture. At the same time that we want to take advantage of the speed of Spark and its ability to manage several types of source, Hence the idea of designing a new architecture.

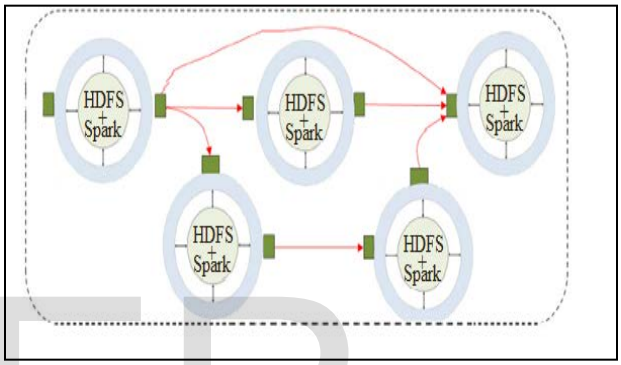

Fig. 3 The core a new SOA for Spark and HDFS

4 RESULTS AND ANALYSIS

We propose a new implementation of service- oriented architecture, to bring together several Spark based on HDFS for data storage. Each service in our design plays the role of a Spark + HDFS and we repeat each service according to our need and the data volume. Thus the novelty in this implementation (Fig. 3), comes from the use of Spark + HDFS for the types of data sources that the MapReduce cannot manage, and in addition we take advantage of the Speed of Spark. The two architectures [2] and (Fig. 3) do not exceed the limits of HDFS or Spark or MapReduce, and they combine the management of several Petabytes of data until the arrival of Zettabytes. So we find this design which mixes the speed of Spark and its ability to manage several other types than MapReduce (real-time flow) Since each service in a service-oriented architecture must respect a contract in which one determines the data sent to the services (HDFS + Spark) according to its varieties and complexities, it is sent to each service a Petabytes data and the services are repeated until the management of all the size and types of data collected.

5 CONCLUSIONS AND FUTURE WORK

We are working on several terms necessary in the field of Information and Communication Technology (ICT) and in all

areas relating to access to the Internet as the Big Data which has become a fundamental term everywhere. in addition to the latter it was the essential parts of the project as Hadoop MapReduce and HDFS, the Framework Spark that represents a term of news in the field of Big Data, all in a Service Oriented Architecture which was considered as a solution to the problems of information system only but now it regards as a great solution for combining these framework to manage the big data regardless of its size or its variety. It make these framework capable of managing more than petabytes. The Size of Big Data for the moment is of Zettabytes and latter, will likely exceed this size in the near future.

A new use of the service-oriented architecture in the Big Data domain, was discussed. For the future, we want to use this architecture but in other areas. We can conclude that this architecture has many opportunities that are not used until today.

REFERENCES

[1] S. Gao, L. Li, W. Li, K. Janowicz, and Y. Zhang, "Constructing gazetteers from volunteered Big Geo-Data based on Hadoop," Comput. Environ. Urban Syst., pp. 1–15, 2014.
[2] M. Mohamed, L. Zeinebou, N. Aknin, and S. M. Lemin, "Implementation of the Service-Oriented Architecture to manage the Big Data with Hadoop," vol. 7, no. 10, pp. 1615–1618, 2016.
[3] C. Vitolo, Y. Elkhatib, D. Reusser, C. J. A. Macleod, and W. Buytaert, "Web technologies for environmental Big Data," Environ. Model. Softw., vol. 63, pp. 185–198, 2015.
[4] J. L. Schnase, D. Q. Duffy, G. S. Tamkin, D. Nadeau, J. H. Thompson, C. M. Grieg, M. A. McInerney, and W. P. Webster, "MERRA Analytic Services: Meeting the Big Data challenges of climate science through cloud-enabled Climate Analytics-as-a-Service," Comput. Environ. Urban Syst., 2014.
[5] A. O'Driscoll, V. Belogrudov, J. Carroll, K. Kropp, P. Walsh, P. Ghazal, and R. D. Sleator, "HBLAST: Parallelised sequence similarity - A Hadoop MapReducable basic local alignment search tool.," J. Biomed. Inform vol. 54, pp. 58–64, 2015.
[6] E. Hustad and C. de Lange, "Service-oriented Architecture Projects in Practice: A Study of a Shared Document Service Implementation," Procedia Technol., vol. 16, no. 2212, pp. 684–693, 2014.
[7] V. Mates, M. Rychlý, and T. Hruška, "Modelling of Context-adaptable Business Processes and their Implementation as Service-oriented Architecture," Procedia Econ. Financ., vol. 12, no. March, pp. 412–421, 2014.
[8] A. Doulamis and N. Matsatsinis, "Visual understanding industrial workflows under uncertainty on distributed service oriented architectures," Futur. Gener. Comput. Syst., vol. 28, no. 3, pp. 605–617, 2012.
[9] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'big data' on cloud computing: Review and open research issues," Inf. Syst., vol. 47, pp. 98–115, 2015.
[10] H. Demirkan and D. Delen, "Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud," Decis. Support Syst., vol. 55, no. 1, pp. 412–421, 2013.
[11] P. Pääkkönen and D. Pakkala, "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems," Big Data Res., vol. 2, no. 4, pp. 166–186, 2015.
[12] S. Erevelles, N. Fukawa, and L. Swayne, "Big Data consumer analytics and the transformation of marketing," J. Bus. Res., vol. 69, no. 2, pp. 897–904, 2015.