Implementation of the Service-Oriented Architecture to manage the Big Data with Hadoop

Mint Mohamed Lemin Zeinebou, Noura AKNIN, Salihi Mohamed Lemin

Abstract— Owning to the fast evolution of the Digital WorldThe data do not cease to increase which obligate us to manage Zettabyte of data, therefore we are requested to discover a new design that can adapt with the new volume of Big Data and can make our existing tools evolve .In this article we present a conception that combine several Hadoop with the help of the Service-oriented architecture, to be able to manage the Big Data.

Index Terms— Big Data, Hadoop, HDFS, MapReduce, HBase, SOA, Datanodes, Namenode, Secondary Namenode.

1 Introduction

THE frequent growth of the volume of data in the world is summarized by the term Big Data which constitutes a real challenge for the majority of society. Since 2009 each time that we talk about Big Data we necessarily talk about Hadoop the framework that is the most efficient and the most famous in this area. Hadoop is a framework written in the Java language, very operational, that has as purpose to manage the Petabyte data, it is composed of several bricks which of the most important are: the file system HDFS, the algorithm MapeReduce and the database HBase (Fig .1), to store, analyze and make queries on a huge field of data.

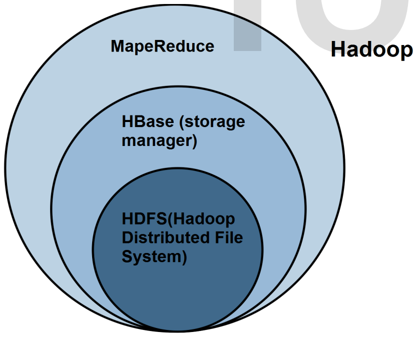

Fig. 1: The Hadoop components

The HDFS component which is used for the storage represents a Hadoop distributed file system. HDFS is one of the most popular open source data-intensive file systems and is an important file system of Hadoop that is inspired by Googles work [1]. HDFS is a typical representative of Internet service file systems running on clusters, and represent a few characteristics, such as, work in commodity clusters with hardware failures, access with streaming data, deal with large datasets, employ a simple c herency model, and portable across heterogeneous hardware and software platforms [2].The HDFS divides the data on several blocks; which are available from Datanodes. Datanodes represent the entities where is the blocks that contain the subdivisions of data. But, this situation causes a major problem; each block must respect a precise size between the two following values either 64 MB or 128 MB [3] at the maximum. However Hadoop MapReduce is a software framework for distributed processing of large data sets on compute clusters. It runs on top of HDFS, thus collocating data storage with data processing [4]. The MapReduce functionality is designed as a tool for deep data analysis, the transformation of very large data sets is based on the concept of parallel programming with high speed. MapReduce programming model works by the processing in two phases which are Map and Reduce phase [3]. When the algorithm MapeReduce remains the only way to explore the data, MapeReduce uses two managers of base that are MAP and REDUCE. MAP puts the input data in the form of a suite key/value in order to collect the data in combination with these keys. At the same time that it has disseminated the input data in several parts, it must execute the processing Map on each part of these data. But if we are working on Zettabyte of data processing MAP we can be totally blocked because it has a huge number of Suites key/value. In addition, the treatment Reduce must be applied on each value associated to a key grouped by MAP. Therefore the REDUCE method has the same problem with the operation MAP. Also HDFS stores the data by dividing them on blocks under the condition that the volume of each block is at a maximum of 128MB 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 Fig .2 [5]. 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. Also, it should be noted that the algorithm MapeReduce uses some programming languages that can make working heavy and slow especially if it must analyze Billions or more blocks.

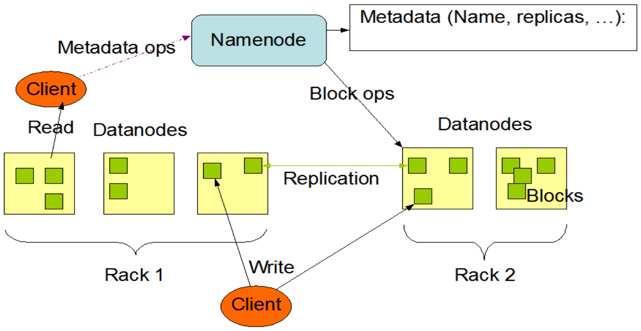

Fig. 2: Description of HDFS

Apache HBase is an open-source, distributed, versioned, non-relational database modeled after Google's Bigtable: A Distributed Storage System for Structured Data by Changet al. Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS[6].Data are stored in HBase by following [key, value] structures.In such pairs, the key represents the row identifier and thevalue contains the row attributes. The [key, value] pairs are stored using the equivalent to well-known primary indexes for RDBMS, which physically sort rows on disk and build a treeon top of it [7].

2 MOTIVATION

The problem of Big Data occurs in different domains. It does not concern only social networks or large commercial enterprises, but also other areas such as sport, health, education, and the Telecom Operators .etc. Also the Big Data influences ont he quality of use of the Internet for the different populations of the world. Furthermore, when there is an access to the Internet the increase in the volume of data is inevitable. The principle of the new proposed design, which is intended to make evolve Hadoop to take account of the increase in the volume of data,is to define a new architecture based on the use of several Hadoop together and to manage the syntactic and semantic interoperabilities by allowing systems Hadoops to exchange between them by messages to facilitate broadcasts and the collections of data in the light of the concept of Service-Oriented Architecture (SAO).

2.1 Related work

The field of exploitation of Big Data is full of Frameworks that are very important and powerful as Hadoop, also many articles have already been published on Hadoop and its components especially HDFS and MapReduce. Most of these articles detail the way of management of Big Data via Hadoop. And proof has demonstrated significant ability to process Terabyte and Petabyte sized data sets in a timely and cost effective manner using a scale out approach [8]. Other Articles talk about the Big Data and indicate Today's measurement tags will be inadequate as data sets continue to surge in size. The size of the digital universe in 2013 was estimated at 4.4 Zettabytes (1 Zettabyte is equivalent to 250 billion DVDs (Cisco, 2014); by 2020, the digtal universe is expected to reach 44 Zettabytes (IDC, 2014) [9].The studies also on the HDFS specify that the size of blocks is btween 64MB and 128 MB, which shows the difficulties for Hadoop to manage the Big Data, that is why Hadoop has been adapted with Petabyte of data at the maximum.
In addition, other news articles demonstrate that the SOA reprsents a software architecture that implements business processes or services by using a set of loosely coupled, black-box components orchestrated to deliver a well-defined level of service [10]. In SOA, a software system implementing the business process or its part is decomposed and distributed in the form of autonomous but cooperative components known as services. This architecture has many advantages, such as better scalability and fault-tolerance. Moreover, SOA principles, such as loose co pling, statelessness, or reusability, allow easy runtime modifications of a composed system by changing its particular components [11]. This implies that this architecture represents a Digital World soft and flexible as it provides an environment of movement, exchange and unrest between the services. The study also focus on the Big Data and the SOA to show the need to link these two terms and to benefit of different characteristics of the SOA but in a different way from what others represented.

2.2 Contribution

We have put together two fundamental terms in the field of informatics to know the SOA and the Hadoop, in seeking to take advantage of the benefits of each concept, we have presented a design of the Service Oriented Architecture and a new use of this architecture to respond to the problems mentioned in the top and benefit of all the advantages of the service-oriented architecture. In fact, we combine several Hadoop together that each of them can manage Petabyte of data to respond to scenarios where we need to manage theincrease in volume of Big Data.

3 DESCRIPTION

The Service Oriented Architecture (SOA) implementations in the business domain typically incorporate an Enterprise Service Bus (ESB) or Message Broker which orchestrates the overall service process execution by mediating between individual services and routing messages [12]. The application of SOA will be assessed in each of these areas, in particular the opportunities it gives for enhancing and increasing the use of finite element analysis [13].The service oriented architecture is used initially to redress the heterogeneity of the various components of the information system in managing all types of interoperability. This architecture is composed of several services which constitute the core of the previous concept.

- *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, MoroccoEmail: aknin@uae.ma* and *Salihi Mohamed Lemin ,SYSCM Laboratory,Faculty Of ScienceNKTT, UniversityNouakchott, MauritaniaEmail: mlsalihi@gmail.com*

The SOA provides a strong communication between the consumer and the producer of the services using the Bus. It also offers opportunities for the exchange of messages between the services. It is based on several protocols very important such as SOAP (Simple Object Access Protocol) to invoke the services; WSDL (Web Services Description Language) is the descriptor of services, UDDI (Universal Description Discovery and Integration) is the discoverer of registry that contains the requested service. The SOA paradigm allows for the development of services implementing other generic, and domain-independent features such as user profiles, knowledge databases, dialog services, etc. [10]. SOA supports variability through dynamic service retrieval and binding [14]. Each service in the SOA must respect a contract which determines and details its way to operate. For our design we must describe a contract which obligates each service to represent a Hadoop and we describe also in the contract the data sent to each Hadoop. This allows us to earn much of the time in the case where we are looking for the data store on the blocks during treatment. The Hadoop's role is to manage the data received and if there is a relationship between the other data in the other Hadoop, it offers possibilities to exchange messages between them due to the SOA. In our design we can manage more Zettabytes of data because it gathers several Petabytes in harmony and the several Hadoop will be used to manage Petabyte of different data ,which helps societies and beneficiaries without exceeding these limits and without finding a problem nor for the distribution or for the collection of data. Our results show that the big data arrives up to of Zettabyte and that Hadoop initially created to manage Petabyte and other problems that can meet with Hadoop MapReduce as well as HDFS in the case of management of Zettabyte data, by comparing several searches made on the increase in the volume of Big Data and other on the managements of Big Data via Hadoop ,and that if we compare our design with the other it is found that the environment that offers the service oriented architecture is an exceptional environment and can be used up to the infinite because each time that the volume of data increases we can add another Hadoop in the form of a service .

4 RESULTS AND ANALYSIS

The figure below represents a design of the core of the SOA which is made up of services but in our design each service plays the role of a Hadoop. The red lines indicate the exchanges between the services that is to say the messages sent between them (Fig .3), due to these messages The work is performed in a very short time , and in this new architecture we can manage Petabyte of data several times which resolves the problems relating to volumes. Each service manages Petabyte of data. The data are divided at the beginning of the treatment and collected at its end. Then this design also allows us to manage the Big Data by Hadoop regardless of the size of the data collected.

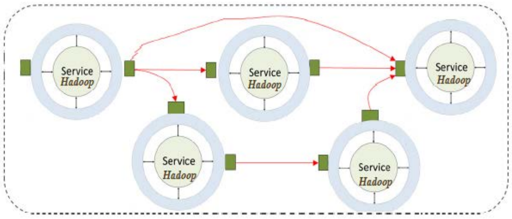

Fig. 3: Nucleus of our Service Oriented Architecture

Due to the bus The User may request a new task to a service or Hadoop during the processing of data and for this the SOA must amend the contract of this service to make it compatible with the request asked (Fig .4).

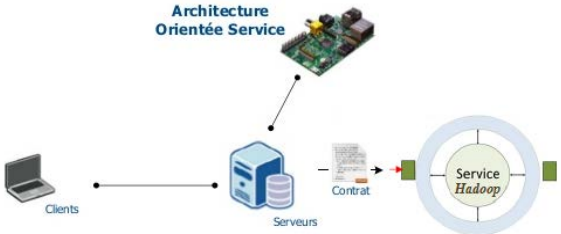

Fig. 4: Use of bus

5 CONCLUSIONS AND FUTURE WORK

We have noted the point of weakness of a framework that is very important as Hadoop and we have proposed the use of the SOA to complete and correct the limits of this software. We have evolved Hadoop in taking advantage of all benefits of the Service Oriented Architecture by making Hadoop capable of managing the Big Data with new sizes of volumes. We have suggested a use of the Service-Oriented Architecture to maintain together a set of softwares or frameworks for the management of Big Data , which can be other than Hadoop, we concentrated on Hadoop because it is the most used for the moment, but we have indicated the possibility to use this implementation with other frameworks .This design can maintain in reality for any team to the laboratory concerned by the management of large volumes of data ,evidently it is a very interesting idea in the future .We believe that this design will achieve the goal of targeting dozens of companies.

REFERENCES

[1] 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.

[2] F. Tian, T. Ma, B. Dong, and Q. Zheng, "PWLM3-based automatic performance model estimation method for HDFS write and read operations," Futur. Gener. Comput. Syst., vol. 50, pp. 127–139, 2015.

[3] C. Uzunkaya, T. Ensari, and Y. Kavurucu, "Hadoop Ecosystem and Its Analysis on Tweets," Procedia - Soc. Behav. Sci., vol. 195, pp. 1890–1897, 2015.

[4] S. Ibrahim, T. D. Phan, A. Carpen-Amarie, H. E. Chihoub, D. Moise, and G. Antoniu, "Governing energy consumption in Hadoop through CPU frequency scaling: An analysis," Futur. Gener. Comput. Syst., vol. 54, pp. 219–232, 2014.

[5] A. Ben Ayed, M. Ben Halima, and A. M. Alimi, "MapReduce Based Text Detection in Big Data Natural Scene Videos," Procedia Comput. Sci., vol. 53, pp. 216–223, 2015.

[6] HBase – Apache HBaseHome, ⟨https: / / hbase.apache.org/ ⟩ , [O nline;accessed 05-March-2014].

[7] H. Garcia-Molina,J.D.Ullman,J.Widom,DatabaseSystems—The Complete Book,2nded.PearsonEducation,UpperSaddlerRiver, NewJersey(USA),2009.

[8] 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.

[9] 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.

[10] M. B. Carvalho, F. Bellotti, R. Berta, A. De Gloria, G. Gazzarata, J. Hu, and M. Kickmeier-Rust, "A case study on Service-Oriented Architecture for Serious Games," Entertain. Comput., vol. 6, pp. 1–10, 2015.

[11] 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.

[12] A. Cameron, M. Stumptner, N. Nandagopal, W. Mayer, and T. Mansell, "Rule-based peer-to-peer framework for decentralised real-time service oriented architectures," Sci. Comput. Program., vol. 97, Part 2, no. 0, pp. 202–234, 2015.

[13] R. I. MacKie, "Application of service oriented architecture to finite element analysis," Adv. Eng. Softw., vol. 52, pp. 72–80, 2012.

[14] M. Galster, P. Avgeriou, and D. Tofan, "Constraints for the design of variability-intensive service-oriented reference architectures - An industrial case study," Inf. Softw. Technol., vol. 55, no. 2, pp. 428–441, 2013.