BIG DATA: AN APPROACH ON APACHE HADOOP

Rodrigo Ramos Nogueira, Ezequiel Gueiber
Universidade Estadual de Ponta Grossa
E-mails: wrkrodrigo@gmail.com; ezequiel@uepg.br;

Abstract: Every eighteen months the volume of existing data in the world doubles in size, making it increasingly becomes difficult to store and query the information that is derived from multiple data sources. This paper introduces Apache Hadoop as a solution to problems involving Big Data. BIG Data is the term used to refer to the great mass of data, from different sources of information, stored in various locations, updated all the time and these are the three V's (Volume, Speed, Variety). Big Data is not a technology, it is a concept where voluminous and complex databases, and can be structured, semi-structured and unstructured communicate, but not always perform operations on the waiting time, making some impossible tasks using traditional storage technologies . The objective of this paper is to present a solution for Big Data storage, distribution and mining data from different sources, with a large volume of information and agile way. For the development of the research tool Apache Hadoop open source technology developed in Java and runs on Linux operating system was used. The main contribution of this research is to present an efficient and free solution for Big Data application in a distributed environment, with its benefits and specifying its easy use.

Keywords: Massive Data; Database, Hbase.

1. INTRODUCTION

According the Google Trends the overall volume of digital data came close to 8 zettabytes by the end of 2015. There are several information generated daily through text files, photos, videos, documents, transactions, among others, information that are stored on servers located throughout the world. In many scientific disciplines are becoming more computer based and data-driven, such as pysics, astronomy, oceanography and biology (BOLLIER, DAVID, 2010).

Big Data is the term used to describe the large volume of data generated at all times, which are come from various sources and utilizing different forms of storage.

Some of the major technology companies like Google, Facebook, Twitter and Amazon have their own servers and research teams in the area of Big Data in search of a better way to distribute and access this data as quickly as possible. Any undertaking or public or private, cannot do without information technology as a tool geared to the strategic plan, as it provides data for managers to subsidize them in their decision-making (REZENDE, DENNIS, 2011).

As Facebook reached 700 petabytes in 2011, in 2014 it crossed the hexabytes and has an important responsibility to hold with integrity that information and ensure timely access to all its users, which can theoretically reach anyone have a computer connected to the Internet.

The big problem is to solve the challenges posed by Big Data is not an easy task, after most of the solutions proposed involve high cost of information distribution and storage. In This context Apache Hadoop has emerged a free software platform open source which does not require large servers to manage large databases, it can be run from a group of low cost computers, collaboratively. Apache Hadoop is the subject matter presented in this paper.

2. BIG DATA CONCEPTS

Big Data, is also the term definded by IBM to determine the large amount of data generated by information systems. In the last five years is increasing business investment in Big Data and technologies resulting from this trend, and every day is generated a high volume of information around the world. Companies seek ways to ensure the integrity of such information, as well as quick access, aiding decision making, as isolated data does not have a great value, but the intersection of these can generate information relevant to a particular segment.

This interest is reflected in very interesting numbers, as about half a billion dollars invested by ventural capitalist in startups of Big Data in the last two years, although the Big Data market be concentrated in existing large authors today in the software industry, as IBM, HP, Oracle and Microsoft (TAURION,CESAR, 2013).

An example of the power of information in the case of a large network of US retailers that using data mining techniques, has led to a pattern of consumption where it was found that the disposable diaper buyers were potential consumers of beer, and to relocate products to stand side by side in stores diapers and sales of beers increased considerably. In the new millennium knowledge management has been considered a watershed, from the moment that the term and the concept of knowledege Management is defined as an area of study, research in the academic context, corporate and economic (REGENSTEINER, ROBERTO, 2013).

Still, this research states that buyers were parents with small babies at this stage of their lives ended up spending the weekend at home. After this action there was a considerable increase in sales of both (REGENSTER,ROBERTO, 2013).

According to IDC (Institute of Data Corporation) only in 2014 will generate 2.7 zettabytes of information, and this information is coming from different sources and need to be processed within the shortest possible time through of the 3V's (volume, variety and velocity), which are the basis of the concept of Big Data, for from these you can see that Big Data is present in information infrastructure solutions everyday and also future applications.

The first V which refers to Volume is the mass of the whole existing data are million Gigabytes generated every day in data centers distributed around the world. The volume is where it processes the Big Data Analytcs, a process that examines all the information stored and through this enables decision making. The volume aspect relates to the fact that the amount of available data in digital form is growing exponentially, from not only conventional systems, also from sources such as Facebook, tweeter, YouTube, RFID, embedded electronics, cell phones and the like , various types of sensors, and others (BRETERNITZ, VIVALDO, 2013).

When analyzing the data source information systems, it is common that are not derived from the same source, this is the variety, the second V. This diversity of devices and data sources, which are from the management systems, sensors, GPS, social networks, among others even the most diverse, that they will generate data in various formats. The variety also is related to how data are stored and can be structured, semi-structured and unstructured.

As for speed, the third V, is a challenge because it encompasses the speed under which the data is persisted and analyzed due to performance issues of relational databases to manage the vast amount of data produced. Before this problem arose storage and distribution technologies like NoSQL (Not Only SQL) and Map Reduce, distribution technology developed by Google, described in more detail later in this paper.

Among the processes that allow extracting information from a database, turning into relevant information for decision making, statistics stand out and mining data, where there is knowledge discovery through a data set and there is interest to obtain results from large sets of data in a short time.

Big Data apply in different scenarios, and from now on will be addressed problem solving ways and proposed solutions for Big Data, to start the MapReduce distributed computing model, under which the apache hadoop is based.

3. MAP REDUCE

A challenge within Big Data is the fragmentation and distribution of bases. In order to contribute to solving this problem MapReduce model was created. MapReduce is a programming model that parallel was developed by Google™ to solve the big problem of partitioning data volume and complexity.

The MapReduce model has already been implemented in other programming languages such as LISP and Haskell, also existing MapReduce libraries for Java, C ++, Python, and other languages.

According (DEAN, Ghemawat, 2008) more than 10,000 programs have used this model, addressing the most diverse problems, and algorithms for word processing, statistical analysis, genetic analysis, among others.

The MapReduce model simplifies parallel processing by abstracting away the complexities involved in working with distributed systems, such as computational parallelization, work distribution, and dealing with unreliable hardware and software. With this abstraction, MapReduce allows the programmer to focus on addressing business needs, rather than getting tangled up in distributed system complications (HOLMES, ALEX, 2012).

The MapReduce makes short work of those who use the model, making all communication control between us, contingency and competition in the distribution system. Thus, it is up to those who only use the development of Map and Reduce functions. The MapReduce functions always return the data to form pairs, said (key, value).

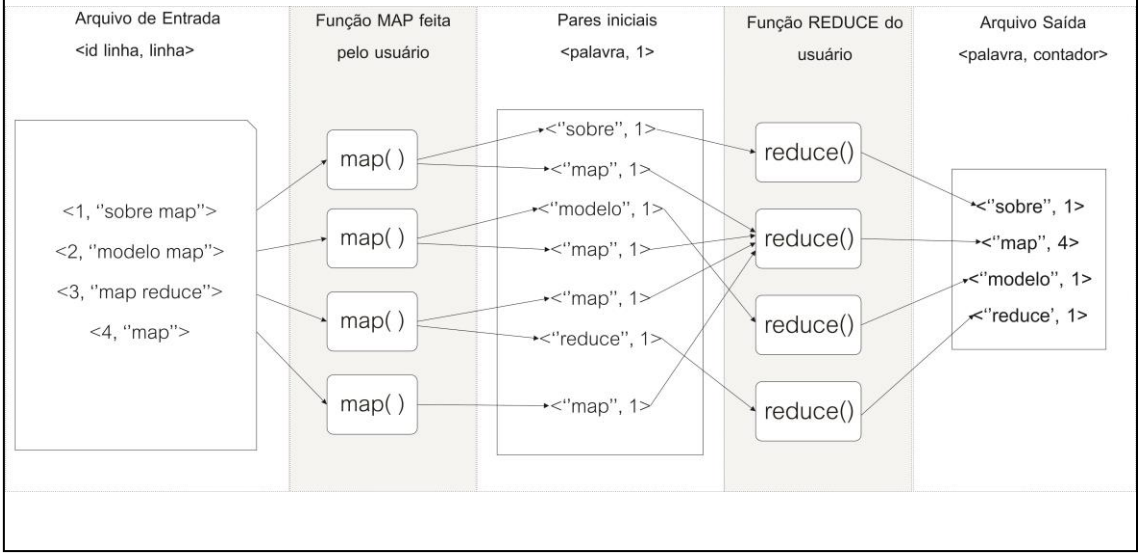
The Frame 1 presents a example of the a Map algorithm for use in thee Map Reduce model. It consists of the word count in a particular file. For implementation of the Map function were used as two input parameters and key value fields where value refers to a particular text file line and a line key identifier. The map function breaks the last line of reference words which are emitted one by one, along with the constant 1 which further serve to effect counting of words.

```
function map (Int key, String value){
    arrayWords = split( valor );
    for each palavra in arrayWords
        emit(word, 1);
}
```

Frame 1 – Map function

The implementation of the Reduce function is in the Frame 2, it receives as parameter the key, which refers to a specific word, and the value is associated with that key. In our case all the values have the constant 1, indicating its occurrence. The Reduce function is the number of occurrences of the count and sends each key (word) and the number of occurrences of the same.

Figure 1: Flow of Map Reduce




```
function reduce (String key, Interator value){
count = 0;
for each num in valores
        count += 1;
emit(chave, count);
}
```

Frame 2 – Reduce funcion

In addition, one Map Reduce algorithm execution scheme is shown in Figure 2. The flow observed in the first stage the input file, which contains four pairs of <key/value>, each representing one line. Each pair generates a process for Map function, which performs the broken lines in words sending the data pairs <key/value>, where in this case the word itself becomes a key. In Reduce phase, the function performs the counting of the data, then sending these values associated with the key.

The Map Reduce was developed in a way to process large volumes of distributed form of files between computers on a network using parallel, in general, the Map Reduce lets break a big problem in several other sub-problems, solve them simultaneously, allowing you to find a more rapid solution to solve one by one.

With Map Reduce we are not just fighting with clean code and easy to maintain, but also with the performance of a work that will be distributed to hundreds of nodes for computing over terabytes and even petabytes of data. In addition, this work is potentially competing with hundreds of others on a shared cluster of machines

With Map Reduce not working only with clean code and easy to maintain, but also with the performance of a work that will be distributed to hundreds of nodes in a cluster, manipulating data at the level of terabytes. Also it can work concurrently with hundreds of machines in a shared cluster (MINNER, SHOK, 2013).

For the operation of Map Reduce data in the mapping phase should be stored in a distributed system, so the stored files will be divided and distributed, using a scheduler tasks processes

will be distributed among the nodes in the cluster and declare the Map function the user set which data will be used.

After explaining the concept of Map Reduce model, the next steps demonstrate apache hadoop, described in the next section.

4. APACHE HADOOP

Within the context of Big Data is inherent in expanding the volume of data and the need for high availability information, however, conventional architectures physical constraints imposed by the topology attached to them. To fill this gap appeared Apache Hadoop, a framework for processing and storage data in large-scale.

The Hadoop project was created in 2005 by Doug Cutting, who put the name of Hadoop was in honor of his son, as this was his son's teddy bear's name. Doug Cutting developed a framework of distributed files based on papers provided by Google on Map Reduce (aborted in this paper) and GFS (Google File System), soon after the project was Yahoo's investment faced problems to handle the large number of references for websites, from there Hadoop came into being as an independent project of the Apache Software Foudation.

In January 2008, Hadoop has become a high relevance of the Apache project, confirming its success and its diverse active community. At this point, Hadoop was being used by many other companies in addition to Yahoo, like Last.fm, Facebook, and the New York Times (WHITE, TOM, 2013).

Apache Hadoop has been implemented in Java and has its open source code. Also supports the storage and access to the large volume of data, which may physically be on a single computer or even on different computers, which are referred to us. Running on Linux environment the main objective of Hadoop is to store large-scale information in a distributed environment enabling rapid access and overcome the major challenges of Big Data.

The use of Apache Hadoop apply the idea of master/slave in a network environment of computers connected by an Ethernet network, where computers, also called nodes within a cluster have different functions. The master and slave nodes work in sync, allowing the operation of Hadoop. Apache Hadoop is designed to be implemented at any level of clusters, whether it consists of a network of distributed high-end servers or even on a home network of personal computers.

Using the capabilities of Apache Hadoop is possible to integrate several databases, whether relational, non-relational, structured or unstructured. The structure of Hadoop allows you to store and integrate all these diversities.

The Apache Hadoop framework involves not only the concept of Big Data and database, but also distributed computing, parallel computing, file systems, operating systems, diversity of information (structured and unstructured data) and file handling through its own file system HDFS (Hadoop File System), allowing through its various features centralize the operation through the framework.

Apache Hadoop has several components, which began as Hadoop subprojects and today are independent projects of the Apache Foundation:

- Hadoop Streaming: This component that allows coding applications in many languages, not only in java;

- HDFS (Hadoop Distributed File System): It is the data management system distributed Hadoop, this manages data storage and distribution and will be explained in more detail later in this paper;
- Hive/Hue: It is as the manipulation of data within Hadoop uses the Map Reduce, these allow commands to be executed in SQL (Structed Query Language);
- Pig: This component allows the commands to run at a higher level than the Map Reduce and scripts are created distributed across the cluster;
- Map Reduce: It is a framework that implements the Map Reduce programming model, which aims to share the information to process in separate blocks and competitors, Map Reduce will be explained in detail in the sequel;
- Sqoop: It is a tool creates an Apache Hadoop interface with relational databases and data warehouse tools;
- Zooqueper: The component is the coordinate distribution service apache hadoop, designed to work in clusters which several other components;
- Hbase: This is the database management systems that is native within the Apache Hadoop, it is a NoSQL column oriented database.

For running an application using Apache Hadoop are needed two mainliny components: HDFS and Map Reduce framework (VIEIRA, MARCOS, 2012). Knowing the importance of these components, these are explored in the coming sessions.

4.1. Hadoop File System

Hadoop File System - HDFS is the Hadoop file manager system, HDFS is for distributed storage environment as well as NTFS (New Tecnology File System) is for Windows 9x family and Ext2 (Second Extended File System) this for some versions of Linux.

A functional component of Hadoop ecosystem is the Hadoop Distributed File System (HDFS). HDFS is the mechanism that stores the amount of data distributed by the cluster of computers, the data that is stored but which are not read long ago will also be analyzed (LUBLINSKI,BORIS, et al., 2013).

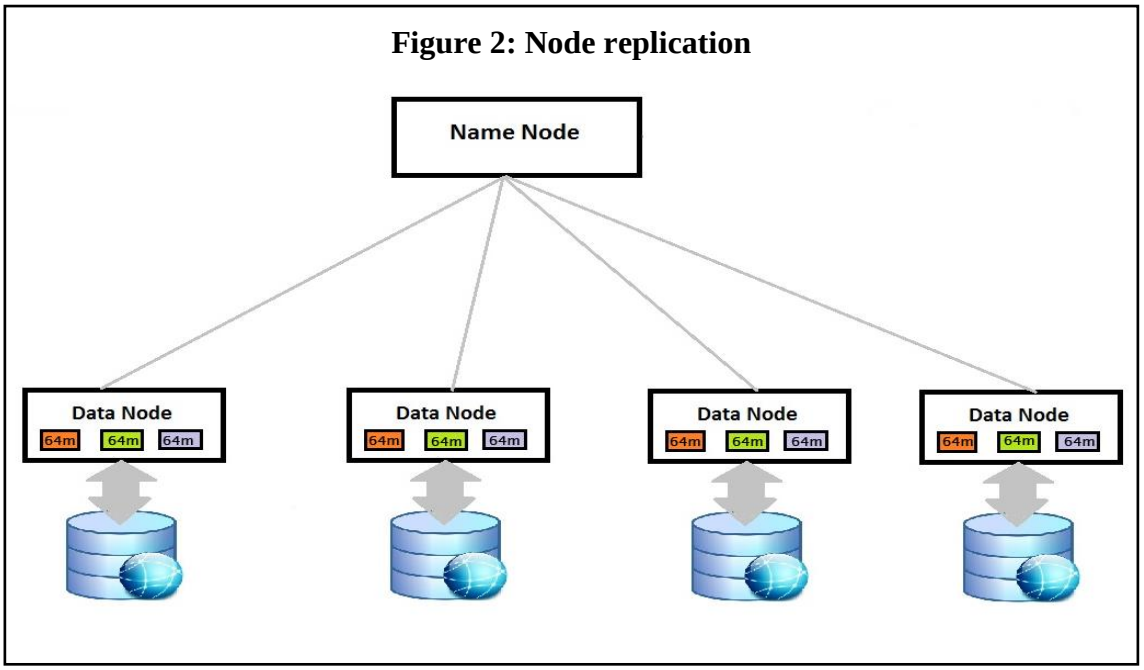
The HDFS is responsible for one of the main features of Apache Hadoop, the ability to run on commodity hardware, ie is responsible for making a group of computers with diverse and most simple configurations, work together as a single cluster . A distributed file system that works with low computational cost (WHITE, TOM, 2013).

Like any file system manages the storage, organization, retrieval, protection, sharing and permission files, including all these functions through an appropriate interface, facilitating the developer's work that does not care about the architecture and use only the file system framework.

Besides the responsibility of a traditional file manager, HDFS brings with it the responsibilities of a distributed file management system, allowing file remote access as if on the same node, as well as providing the same performance of a traditional file system, ensuring thus scalability.

So, as the Apache Hadoop framework was based on the paper published by Google ™ on Map Reduce, HDFS had extreme influence by the model presented by an paper about Google File System (GHEMAWAT, SANJAY, et al.,2003).

Figure 2: Node replication



Google File System it is a distributed file system from Google. Initially it was designed to be a system used to meet needs related to storage and processing of large data sets. It is used for several purposes and developed in clusters that allocate hundreds or even thousands of nodes (VIEIRA, CELSO, 2011).

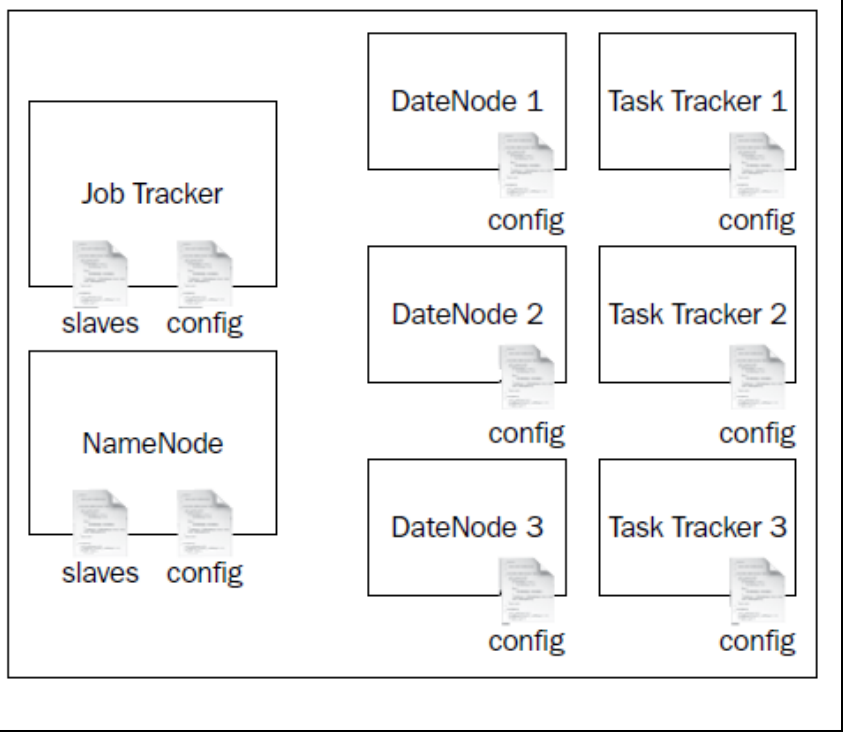
The HDFS is a major contributor to the fault tolerance of the Hadoop Apache because as the number of larger machines will be the likelihood of failure from the data replication can be guaranteed that the information cannot be accessed will be available in another computer.

The hard drives data into blocks and replicate the nodes of the cluster ensures consistency and availability of information, the provision of the blocks at the nodes can be seen in Figure 2.HDFS in your distributed architecture, divides the data into blocks, which by default has 64MB size, but can be configured according to the need, may have larger blocks, smaller and even with mixed sizes. These data are distributed to us, if at any time any node is not available one block will be replicated to another node, and can be accessed. This mechanism of dividing data into blocks and replicate the nodes of the cluster ensures consistency and availability of information, the provision of the blocks at the nodes can be seen in Figure 2.

In HDFS architecture can the four major components for storing and distributing files, these work together and are illustrated in Figure 3:

- NameNode: manages files stored in HDFS, cited performs a function in the Map Reduce, because it is where the information will be mapped and divided into blocks, storing location information.
- DataNode: where the data actually will be stored as the goal is a distributed application may occur several instances of a DataNode distributed in blocks.
- TaskTracker: is responsible for performing tasks or submit progress reports to the JobTracker.
- JobTracker: is responsible for managing what is being done by TaskTracker if a task fails he is responsible for relocating the task in another TaskTracker.

Figure 3: HDFS Ecosystem



The four components cited above can be displayed immediately on start Apache Hadoop, it is so relevant that in each process performed within the Hadoop ecosystem, at least one of the four is used.

The fact that the management of the Hadoop File System be done in all clusters allows us to decrease redundancy (TURKINGTON, GARRY, 2013).

It has now been addressed the concept of system architecture in which the distributed information, will now be seen database technology used.

4.2. Hbase

An important component when it comes to Big Data is the NoSQL (Not Only SQL - Not only SQL), which arose from increased scalability in writing and reading data especially after the rise of technologies focused on cloud.

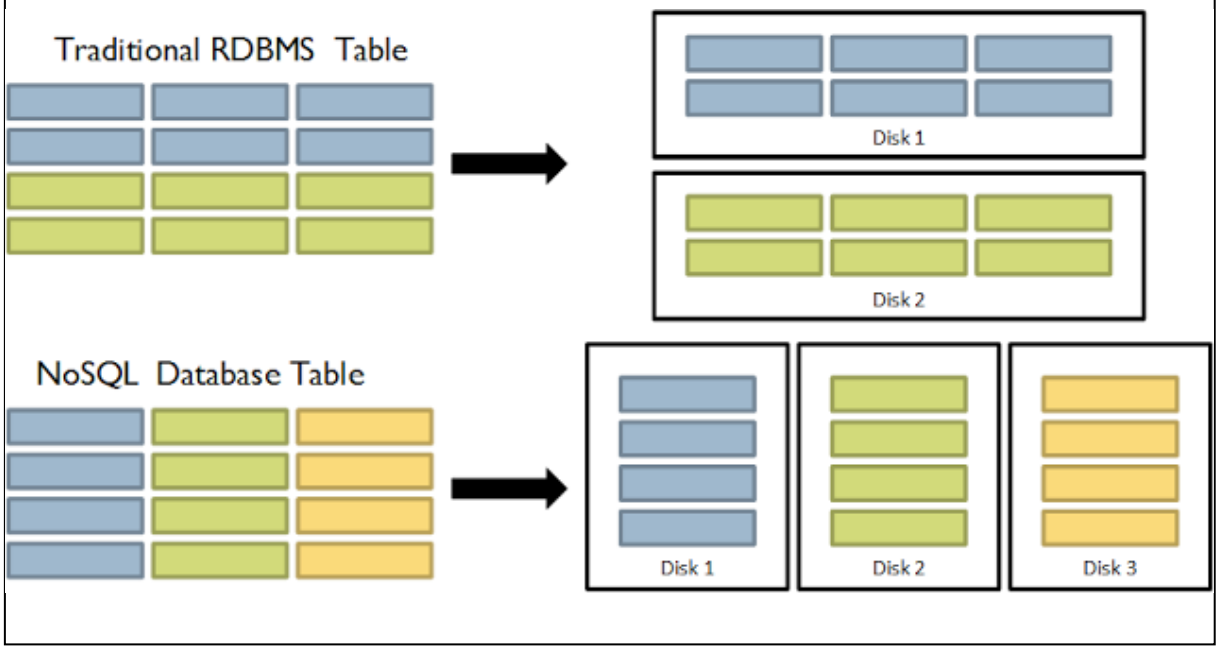
The NoSQL terminology defines the non-relational databases, a trend that grows in the course of the past years, mainly for pioneering the use of such technology by big companies like Facebook. NoSQL promotes various innovative solutions for storing and processing large amounts of data (VIEIRA, MARCOS, 2013).

From NoSQL database is possible large-scale data access, getting results in a short amount of time when compared to the same operation using relational database. During the database using NoSQL we encounter various forms of data storage: document-oriented, key/value, columns orientes,oriented graphs and others.

The Apache Hadoop have it own database, this name is Hbase (Hadoop Database). It is a DBMS (Database Management System) NoSQL, oriented column that runs on top of HDFS. It is well suited for sets of sparse data, because the fact of having a structure based tables and columns that allows to integrate better the model of distributed computing used by Apache

Hadoop, the fact of being in columns facilitates their fragmentation and distribution. The Figure 4 shows the comparison between relational database and NoSQL.

Figure 4: Comparison of Traditional RDBMS and NoSQL (BROWN, PATRICK,2013)

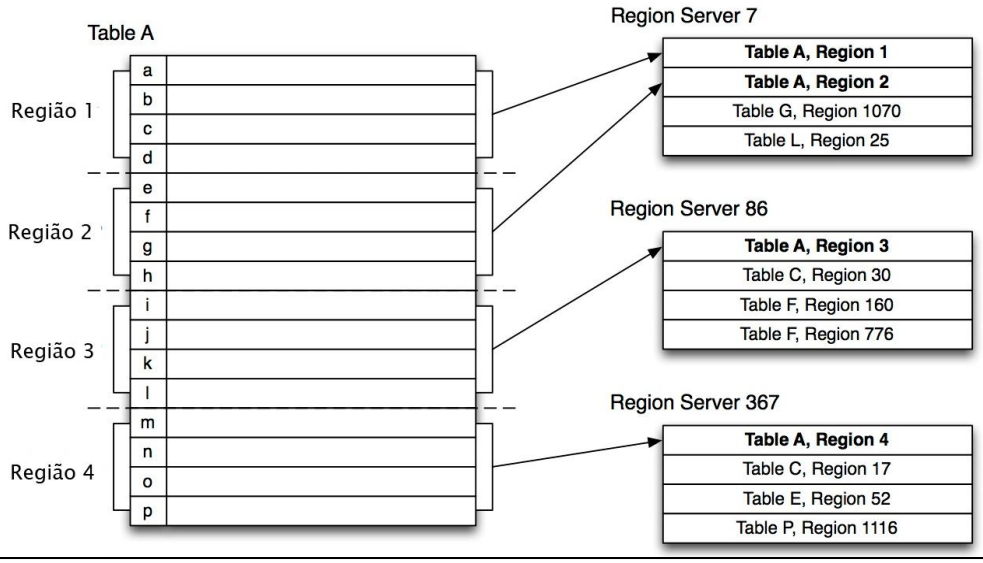


The Hbase is the open source implementation of Google's Big Table architecture. Similar to traditional relacionas database management systems (RDBMSs), the data in Hbase are organized in tables. Unlike conventional RDBMSs, because the Hbase supports several lost definitions schemes and does not provide support for joints, query languages, or SQL (LUBLINSKI,BORIS, et al., 2013).

Using NoSQL database can get better performance on queries with large volumes of data. There are advantages when working with a subset of the available columns. For example, computing maxima, minima, averages and sums, specifically on large datasets, is where these column-oriented data stores outshine in performance(VAISH, GAURAV, 2013).

Bigtable is designed to scale to a large size. Bigtable is a bank oriented columns, which is treated as a multidimensional map and is designed to store large amounts of data. The structure of Big Table is similiar to a relational database when it is indexing, as it is indexed by a row key (RowKey) a column key (ColumnKey) and a timestamp, each value in the map is an array byte not interpreted. The Figure 5 shows the store scheme that used from Bigtable and Hbase.

Figura 5: Hbase Region(BERTOZZI, MATTEO,2013)



HBase is oriented columns and comprises a set of tables. Every table contains columns and rows as well as a traditional database. Each table should have a defined element as the primary key, and all attempts to access to HBase tables should use this primary key.

Working with Hbase relies heavily on another component of Hadoop, the Zookeper, as previously mentioned, the Zookeper is responsible for the distribution of data within the hadoop environment, this component is used to reduce errors and expands.

The ZooKeeper Apache Open Source is an API that allows distributed processes in large systems synchronize information among themselves without fail, so that all customers who make requests to receive consistent data.

With HBase you must set the table schema and specify the columns of families. However, it is very flexible in the new columns can be added to households, as well as leaving families, at any time, causing the flexible framework and therefore be able to adapt the new application requirements.

In HBase a master node manages the servers and clusters of tables and performs data management. Likewise HDFS some concerns have failed due to the availability of NameNode because HBase is also sensitive to the loss of a master node.

The Hbase automatically manages the tables in regions, through a server component named region (regional server), distributes each region a subset of data table rows, this process is illustrated in Figure 6.

The division into regions facilitates the storage and distribution of data by the HDFS, and is the process that makes the flexible storage and faster than in the relational database.

These divisions have been developed thinking in auto sharing, as offered by other systems. Zones allow fast recovery when a server fails and also load balancing, since they can be moved between servers when the server load that currently serves the region is under pressure, or if the server becomes unavailable due to a fault or because it is being removed from the cluster (LARS,GEORGE, 2013).

6. DISCUSSIONS

6.1 Applications

The main resources were described to run an application with Apache Hadoop. It became clear their applicability to BIG Data, employment collaborative, parallel computing, among others, in order to provide better performance, including lower costs by clustering architecture that provides. applies to cluster formation computers.

Thus, the model is highly relevant when dealing with a large volume of data for sample, however when applied to smaller volumes of data, for example, store data from an inventory management software, there will be a big cost of storage and further processing, assuming a small bank will have just a few tables, distribute this information using the resource distributed model can bring the opposite effect, the worse the computational performance.

6.2 Database

The structure of the Apache Hadoop is relatively dynamic when it comes to the database, with the only restriction to work with a database and that this bank runs on the Linux operating system.

Within the Apache Hadoop it is possible to use from conventional relational databases, as well as NoSQL, however, as seen previously there will be a considerable increase in performance when using a nosql database, the template NoSQL allows for better fragmentation and distribution of data by us, thus obtaining a better solution for Big Data volume problems.

6.2 Distribution model and data storage

The data distribution model using the Map Reduce architecture, is certainly the main reason for the Apache Hadoop is increasingly gaining ground and solving the most diverse problems involving Big Data, as well as the fact run on a commodity hardware.

The two factors mentioned together with allow the Apache Hadoop is implemented in any computing environment, from large corporations to servers and large structure, even in small laboratories with conventional hardware.

When used Map Reduce allows that operations are carried out with a large volume of data at a low time, paralleling operations and obtaining an expected result.

6.3. Advantages

Apache Hadoop is considered one of the main tools for Big Data and data manipulation on a grand scale, among the main benefits we can highlight:

- Scalability: As in other tools is difficult to work with changes, additions and cluster computers removals in some cases even being necessary to redo the encoding. In the case of hadoop this change is made in only one configuration file, which simplifies the work of developers.
- Open Source: You can not say it's the best advantage, but it sure is the main reason for the expansion of Hadoop technology, being an open source project allowed the spread and evolution of technology, not only in the Apache project Hadoop, but also in new Hadoop technologies that have been developed in different environments and operating systems.

- Cost: Due to being an open source project, means that companies do not have the initial cost of investment in the platform, allowing companies to invest in the development of technology improvements applied to their respective needs.
- Ease of setup: Given the importance of their resources, Hadoop Apache is a tool relatively simple, based on XML configuration files for your main settings, managing important items such as parallel computing capabilities and scalability of information in delegating clusters the developer only responsibility to create solutions using the tool.
- Information Assurance: The fact of the Apache Hadoop was designed to run in a distributed database system, and always have the information saved in at least three nodes ensures that all information that is stored can be retrieved.
- Comodotie hardware: A large diferecial Apache Hadoop is that it was designed to run on both a more sophisticated hardware and for a simple hardware, this allows it to be deployed both in a great warehouses data with high-end servers as in a small computer network consisting of personal computers.

6.4. Disadvantages

Apache Hadoop is a framework that since its inception in 2005 has become an open source project that is in constant development since undergoing changes all the time.

- Heritage Parallel Computing: The Apache Hadoop was developed based on the concepts and applications of parallel computing and thus inherits many solutions, but also the problems with it problems that are not parallelizable can not be solved with Hadoop
- Small files processing: as well as other solutions for Big Data, Hadoop Apache is a solution aimed at large-scale data manipulation and do not get efficiency for small files, mainly because if there is a large processing to a small volume , resulting in additional costs, which is called overhead.
- Learning Curve: Although considered a relatively easy to use technology, there is a big learning curve for those who want to learn from the start and use the resources of Apache Hadoop framework, as to its implementation successfully occur several concepts are involved, as knowledge in Java, parallel and distributed computing, computer networks and database.

7. CONCLUSIONS

During research on Apache Hadoop can be seen immense advantages, but the first consideration to make this technology should only be used when really it was designed, for Big Data, need not necessarily be multiple sources of information, but to obtain a plausible result, need to have a Big Data scenario.

As for Apache Hadoop is an extremely powerful tool that covers many computing environments and can be applied from the academic environment even the multinational companies operating in several countries since its architecture is structured to receive a lot of information.

Although many features and various components to solve Big Data, when it comes to low volume data Apache Hadoop is inefficient, after its structure is made to handle large volumes, not small.

This research has forecast continued with the implementation of a network environment with Apache Hadoop for mining of agricultural data.

REFERENCES

VIEIRA, MARCOS, et al. ***Bancos de Dados NoSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data***. *Simpósio Brasileiro de Bancos de Dados* (2012).

TURKINGTON, GARRY. ***Hadoop Beginner's Guide*.** Packt Publishing Ltd, 2013.

VAISH, GAURAV. ***Getting started with NoSQL***. Packt Publishing Ltd, 2013.

BOLLIER, DAVID, and Charles M. Firestone. ***The promise and peril of big data***. Washington, DC, USA: Aspen Institute, Communications and Society Program, 2010.

TAURION, CESAR. ***Dados: A era do crescimento***. LINUX Magazine, page 34, November/2012

Wiki, Hadoop. ***PoweredBy***. (2011).

DEAN, JEFFREY, AND SANJAY GHEMAWAT. ***MapReduce: simplified data processing on large clusters.***Communications of the ACM 51.1 (2008): 107-113.

REZENDE, DENIS. ***Tecnologia da informação aplicada a sistemas de informação empresariais: o papel estratégico da informação e dos sistemas de informação nas empresas.*** Atlas, 2001.

HOLMES, ALEX. ***Hadoop in Pratice***.(2012).Manning Publications.

VIEIRA, CELSO. ***Processamento de dados em larga escala na computação distribuída*** (2011). Monografia

GHEMAWAT, SANJAY, HOWARD GOBIOFF, AND SHUN-TAK LEUNG. ***The Google file system.*** ACM SIGOPS operating systems review. Vol. 37. No. 5. ACM, 2003.

LUBLINSKY, BORIS, KEVIN T. SMITH, AND ALEXEY YAKUBOVICH. ***Professional Hadoop Solutions***. John Wiley & Sons, 2013.

BRETERNITZ, VIVALDO. "***O uso de Big Data em Computacional social Science: tema que a sociedade precisa discutir.***" Reverte-Revista de Estudos e Reflexões Tecnológicas da Faculdade de Indaiatuba 11 (2013).

REGENSTEINER, ROBERTO. ***Gerenciamento do conhecimento: origem, contexto histórico e gestão.*** Augusto Guzzo Revista Acadêmica 1.12 (2013): 141-153.

GEORGE, LARS. ***HBase: the definitive guide***. O'Reilly Media, Inc. 2011.