International Journal Of

Recent Scientific Research

ISSN: 0976-3031

Volume: 7(3) March -2016

ANALYSIS OF HADOOP OVER SAP SOFTWARE SOLUTIONS

Veena V Deolankar., Nupoor Deshpande and Mandar Lokhande

THE OFFICIAL PUBLICATION OF
INTERNATIONAL JOURNAL OF RECENT SCIENTIFIC RESEARCH (IJRSR)
http://www.recentscientific.com/ recentscientific@gmail.com

Available Online at http://www.recentscientific.com

ISSN: 0976-3031

International Journal of Recent Scientific Research
Vol. 7, Issue, 3, pp. 9212-9215, March, 2016

International Journal of Recent Scientific Research

RESEARCH ARTICLE

ANALYSIS OF HADOOP OVER SAP SOFTWARE SOLUTIONS

Veena V Deolankar[1]*., Nupoor Deshpande[2] and Mandar Lokhande[2]

[1,2] Department of Computer Science Engineering, MGM's JNEC, Aurangabad, MS, India

ARTICLE INFO

Article History:

Received 06th December, 2015
Received in revised form 14th January, 2016
Accepted 23rd February, 2016
Published online 28th March, 2016

Keywords:

SAP, HADOOP, Software and business requirement.

ABSTRACT

Every startup should know the process of gathering the software requirement from client, analyze them and document them which is the main motto for any business requirement. The goal is to maintain and develop sophisticated description of system requirement specification document(SRS). This SRS will define software interaction with hardware, external interfaces, response time, speed of operation, quality and security. This study is very useful for a software engineer, how to implement and select the proper software as per the requirement for large data process. In this paper, attempt has been made to prove that Hadoop can be used for large data processing as compared to SAP software solutions. Further it is also shown that for the parameters like flexibility, cost effectiveness, BIG DATA capable and portability gives a edge for Hadoop over SAP.

INTRODUCTION

In today's world rising startups have made the business world more competitive. So to improve their productivity and optimize their manufacturing process these startups are using software's to analyze and manage their Business requirement document (BRD). Business requirement document (BRD) includes documentation of customer needs and expectations. The objective of BRD is, first, to agree with the stakeholders. Second, to communicate with technology service provider which satisfy the customers and business need's and third, to describe the inputs and outputs which are associated with each process function. The BRD distinguish between business and technical solutions.

One of the solutions of BRD can be SAP and HADOOP. The SAP solutions is a processes which gives support from front office to back office, including prospects, students, alumni, financials, operations, procurement, human capital Management, reporting and analytics. SAP Campus Management is integrated with finance and human management solutions without interfaces of data structures. This streamlining saves money as well as time during the implementation phase and maintenance. Hadoop is a popular open source implementation of MapReduce. Hadoop Distributed File System is designed and manufactured to reliably store large files across machines in a large cluster. It is extended by the Google File System.

Hadoop Distributed File System – Goals

1. Store large data sets
2. Cope with hardware failure
3. Emphasize streaming data access

Hadoop and HANA (SAP) can be friends. The data processing in Hadoop is working with unstructured or semi-structured data, refining it, trying to get an output. These output many times end up in database like MySql, HANA, SQL. (VMware, 2012).

MATERIALS AND METHODS

A detailed analysis of SAP and Hadoop for using large data process is shown below

Description of Hadoop and Its Functions

"Running Hadoop" defines running set off daemons. These daemons can be active either on one server or many multiple server.

Corresponding author:* **Veena V Deolankar
Department of Computer Science Engineering, MGM's JNEC, Aurangabad, MS, India

Hadoop cluster

Hadoop uses 40 notes and 1000-4000 nodes in clusters. It has 1 Gbps bandwidth in each rack and 8 Gbps out of rack. Rack switches are connected to another tier of switches performing uniform bandwidth and forms a cluster like structure. Majority of racks will be slave nodes with local disk storage and moderate CPU and DRAM amounts. Some may be master nodes which have slight change in configuration or may have less local storage.

Typical Workflow

Firstly, the input local data is inserted into clusters (HDFS writes). Then analyze the input data (MapReduce). These input data are stored as a result into clusters (HDFS writes). Read out the results, that is, the output from the clusters (HDFS read).

Writing Files to HDFS

Clients directly consults with namenode. Without interfering of namenode, client directly writes block to one of the datanode. The function of datanode is to replicate, therefore it replicates each block. This process or cycle repeats for each block or next block. Here the client breaks the input block into smaller blocks and stored in different machines throughout the clusters. (Tutorials point, 2014)

Preparing HDFS Writes

Here initially data node sends “read” message to client on the same TCP pipeline as an acknowledgement.

Hadoop Rack Awareness

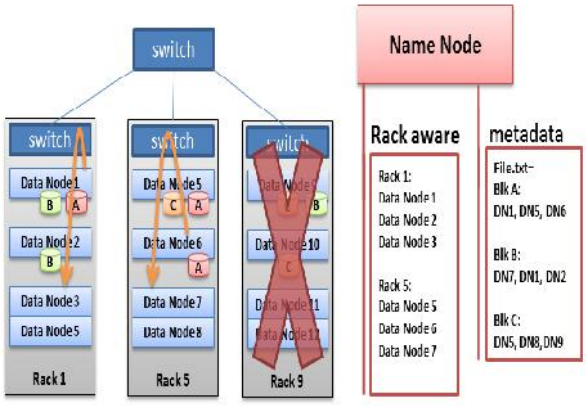

Figure 1 Data are distributed in the form of blocks. These blocks are structured in rack system.

Each block of data will replicate to multiple machines, so that if one of the rack fails the whole data will not lost. As giving the input to rack it creates all mess about storing the data. For this somebody should know where datanode is located and this somebody is nothing but the namenode.

The data in the rack have higher-bandwidth and low latency. Bulky flow of data is necessary throughout the rack.

Re-Replicating Missing Replicas

If there are missing heartbeats it signifies loosing of data. Namenode search and consults with metadata by finding affected data. It consults to rack awareness script. Then namenode request to datanode to re-replicate.

Client Reading Flies From HDFS

For each block the data node list are received at client side . Sequentially each data node is read by client.

There are two main features of Hadoop

MapReduce

MapReduce framework specifies the user jobs as "map" and "reduce" function. The fault tolerance and work distribution is managed by MapReduce framework . Mappers are placed on same node or same rack as the input block. Mappers save user output to the local disk before reducing the data. If the reducer crashes, mappers allow it to recover. It allows more reducers than nodes. (Matei Zaharia, 2009).

Fault Tolerance in MapReduce

If a Task crashes: As there are multiple nodes, if any of the task crashes then it retrieves on another node. This is due to no dependency. If the same task repeatedly fails, the job discards the input block. If a node crashes: As there are multiple other nodes, if a node crashes it re-launches the current task on other nodes. While re-launching the current task it also re-launches any maps to the particular node. This is due to loss of files along with the crash node. If a task is going slow: A second copy of task must be launched on another node. Significant for performance in large clusters (Matei Zaharia, 2009).

HDFS (Hadoop distributed file system)

Distributed file system (HDFS)

For entire cluster it exists only single namespace. Data is replicated 3x times for fault tolerance. 128mb of blocks are splited for 1 file. HDFS stores application data and meta data separately. HDFS supports multiple operations such as read, write, delete files, delete directories etc. When a user reads a file, the HDFS client asks the namenode for the list of datanode. Namenode directly request to the datanode and transfer the desired block. (Konstantin *et. al*, 2010).

Table 1 Functions of Hadoop

HDFS	Distributed file system
MapReduce	Distributed computation framework
HBase	Column oriented table service
Pig	Data flow language and parallel execution framework
Hive	Data warehouse infrastructure
Zookeeper	Distributed co-ordination service
Chukwa	System for collecting management data
Avro	Data serialization system

Architecture of HDFS

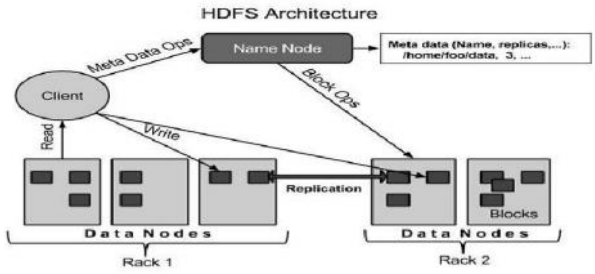

Figure 2 Process between the namenode and datanode.

HDFS Design

HDFS is designed like file system for storing very large files such as

Streaming data access

The most efficient data processing pattern in HDFS is built for write-one, read-many-times pattern. Dataset are copied and generated from source file and various analyses are performed and the dataset over time.

Commodity hardware

Hadoop does not use the expensive hardware, instead it is designed to run on clusters of commodity hardware.

Low-latency data access

HDFS has low-latency data access as it is optimized for delivering a high throughput of data.

Lots of small files

The limit to the number of files in a file system is governed by the amount of memory on namenode as namenode holds file system metadata in memory. (Tom white, 2012).

Non-Flexible: Vendor packages may not fit in a company's business module and customization is expensive. Risk of project failure is increased by implementing SAP.

SAP (HANA) costs very high. HANA can be used in cloud but you can buy hardware for as cheap as $100k. Because of these short comings, the use of SAP is limited for certain applications and thus HADOOP can be used in its place to overcome its drawbacks. (V.Naresh kumar *et. al*, 2012)

Advantages of Hadoop

It is suitable for sharing distributed file system and process easily. Command interface is used to interact with HDFS. The built-in servers of namenode and datanode are useful to check current status of clusters. HDFS provides authentication and file permissions.

Built-In-Redundancy: Out-of-box redundancy is supplied by HDFS and failover capabilities that require no manual intervention.

Big data capable: HDFS can be used to tackle big data use cases. Rate of HDFS can supply data to programming layers, equates to faster processing time and quicker answer to complex questions.

Portability: Most effective advantage of HDFS is, it provides portability between various distribution of files in Hadoop, which helps minimizing vendor lock-in.

Cost-Effective: HDFS is open source software, that translates into real cost saving for the users.

Hadoop using MapReduce is used for parallelizing tasks. It is a key to store large data process. It's not only provide the need of fast response, but rather it provides processing of large amount of data.

Table 2 The daemons has specific roles as given in the table below. (Chuck Lam, 2011)

Namenode	This acts as master for distributed storage and computation. Namenode directs the slave and keeps the track of your files and overall health of distributed file system. If the whole namenode is failed, the whole cluster will stop working which is a main disadvantage of namenode.
Datanode	It performs the grunt work of distributed file system. If you want to read or write a HDFS file, this file will broken into blocks so that the namenode will guide you which datanode each block resides in. For processing the local files, clients directly communicates with datanode daemons. These datanode can also communicate with other datanode for replication.
Secondary namenode	Each cluster has one SNN which acts as a assistant daemon for monitoring the state of cluster.SNN does not receive or record real time changes to HDFS instead it communicates with namenode and take the snapshots of HDFS metadata. This snapshots will help to minimize loss of data as mentioned earlier failure of namenode is main disadvantage of Hadoop.
Job tracker	It act as a liaison between your application and Hadoop. If a user has given the output or the written code to clusters, The jobtracker executes plan and determines which files should process. These files are assigned with the nodes to different tasks and monitors all the task which are running. If one of them fails it again re-launch the task and assign new node to the particular tasks. There is only one Jobtracker present in one clusters of Hadoop.
Tasktracker	It manages the execution of individual tasks on each slave node. The responsibility of tasktracker is to communicate continuously with jobtracker. If the jobtracker fails to contact with tasktracker, it will assume that tasktracker is failed and new node will b assigned.

Disadvantage of Sap

Expensive: As system exists in multiple functionalities the price paid for that software is very expensive. Therefore, the software, hardware, training, implementation are truly expensive in SAP.

Example: HDFS are not "high performance", they are efficient, practical and redundant. Hadoop is free to use. You can buy "premium" versions, download it, install it and make use of it by using any programming language. The hardware part of Hadoop can be combined together with the programming

language and performed greatly, as it will not care much about the hardware. This is used to solve huge problems. (Datastax, 2013)

CONCLUSION

It can be concluded that for development of Business Requirement Documentation (BRD) Hadoop exists more functionalities than SAP. Both the technologies are used in BRD, however there are certain architectural difference including processing of data, flexibility, cost effective. Hadoop is a simple download process and block based rather than files. So, many of the entrepreneurs prefer easy applications with multiple functionalities, one of which is Hadoop than using SAP. Therefore, this paper provides the brief introduction of Hadoop and disadvantages of using SAP on large scale industries.

References

Tom white, Hadoop: The Definitive Guide, 3rd Edition, O'REILLY YAHOO PRESS ISBN: 978-1-449-31152-0, 2012.

V.Naresh kumar, Pawan kumar Illa,2012, Using SAP R/3 for Implementing ERP System, International Journal Of Computer Application, Issue2 Volume3 pp-1-14.

VMware, 2012, Vitalizing apache Hadoop, 1-8.

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, 2010, The Hadoop Distributed File System, IEEE, Sunnyvale, California USA, pp 1-10.

DATASTAX CORPORATION, 2013, Comparing the hadoop distributed file system(HDFS) with the cassandra file system(CFS), pp 1-13.

Tutorials Point (I) Pvt. Ltd, 2014, HAdoop big data analysis framework, pp 1-62.

Matei Zaharia, UC Berkeley RAD Lab, 2009, Introduction to MapReduce and Hadoop, 1-61.

Chuck Lam, Manning Publications Co, 2011, Hadoop in Action, 1-336.

****

How to cite this article:

Veena V Deolankar., Nupoor Deshpande and Mandar Lokhande.2016, Analysis of Hadoop Over Sap Software Solutions. *Int J Recent Sci Res.* 7(3), pp. 9212-9215.

TSSN 0976-3031