ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

A Study on Hadoop Distributed File System used in Cloud Computing

Jaswinder Kaur, Parmeet Kaur

Assistant Professor, Dept. of Computer Science, Chandigarh University, Gharuan, Mohali. India

Assistant Professor, Dept. of Computer Science Chandigarh University, Gharuan, Mohali. India

ABSTRACT:- Hadoop is a open source framework for processing and storing large amount of data on clusters. It is created by Doug Cutting and Mike Cafarella. Hadoop have various modules for operating but in this paper we consider only one module of Hadoop which is Hadoop Distributed File System. HDFS deals only with the storage of the data on clusters in well organized manner. This file system operate with cluster and in this paper we also define cluster and architecture of HDFS with their components in detail. As we all know there are two operations which can be performed on files one is read and another is write. These operations are also described.

I. INTRODUCTION

Hadoop is a software platform in cloud computing for processing and storing large data sets on cluster. The cloud can execute parallel application. Hadoop makes it possible to manage thousands of terabytes of data. The system can continue operating uninterrupted in case of a node failure. Hadoop is actually implemented in Jan, 2008. Hadoop framework is used by major companies such as Microsoft, Yahoo, IBM, Amazon. Hadoop has number of components for operating in cloud computing such components are HDFS, MapReduce, Hive, Pig, HBase, ZooKeeper etc. and all components perform different functions for Hadoop in [9]. The subject of the this paper is Hadoop Distributed File System.

II. LITERATURE SURVEY

The PVFS is an open source parallel file system. A parallel file system is a type of distributed file system that distributes file data across multiple servers and provides for concurrent access by multiple tasks of a parallel application. PVFS was designed for use in large scale cluster computing. PVFS focuses on high performance access to large data sets. It consists of a server process and a client library. The client library provides for high performance access via the MPI in [2]. PVFS provides four important capabilities in one package. First, it provides a consistent file name space across the machine. Second, it supports transparent access for existing utilities. Third, it provides physical distribution of data across multiple disks in multiple cluster nodes. Fourth, high performance user space access for application. It must provide a name space that is the same across the cluster and it must be accessible via the utilities. PVFS file system may be mounted on all nodes in the same directory simultaneously, allowing all nodes to see and access all files. In order to provide high performance access to data stored on the file system by many clients. PVFS spreads data out across multiple cluster nodes, which are I/O nodes. By spreading data across multiple I/O nodes, applications have multiple paths to data through the network and multiple disks on which data is stored in [4].

Many such systems do not perform well with large no. of clients simultaneously accessing single server or simultaneously accessing the same file. PVFS management is not easier. Lots of development is taking place in PVFS, such as redundancy, more interesting data distributions and use of network protocols in [7]. A CEFT-PVFS has been designed and implemented to meet the critical demands on reliability while still being able to deliver high throughput. As the storage capacity increases exponentially, the storage cost decreases accordingly. The performance can be significantly reduced if the data servers, which also serve as computational nodes in a cluster, are heavily loaded by applications running in the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

cluster in [1]. In [6], the feasibility and scalability of building a considerably reliable storage system with multi-terabyte capacity without any additional hardware cost in a cluster. Four different protocols are designed to implement the fault tolerance in Parallel File System. Their experimental results and analytical analysis lead them to conclude that these protocols can improve the reliability while degrading the write performance.

In [3], Lustre is a scalable parallel file system for use on large-scale compute cluster. Luster needs to support many different networks. Luster is a clustered file system that combines features from many scalable previous distributed file system. The name Lustre is a portmanteau word derived from Linux and cluster. Lustre file System is available under the General Public License and provide high performance file systems for computer clusters. Client do not directly modify the objects in [5]. Lustre contains a unique architecture, with three major functional units. One is a single metadata server or MDS that contains a single metadata target or MDT for each Lustre file System. This stores namespace metadata, which includes filenames, directories and file layout. MDT controls file access and informs the client nodes which object make up a file. Second, are one or more object storage servers or OSS that stores file data on one or more object storage targets or OST. An OST is a dedicated object for read/write operations. The capacity of a Lustre file system is determined by the sum of the total capacities of the OSTs. Third, client that accesses and uses the file data in [8]. There is a possibility in Lustre file system in which OST object can be damaged or OST become read-only object.

III. HDFS

The Hadoop Distributed File System (HDFS) is designed to store very large data sets. HDFS create several replications of the data block and distributes them accordingly in the cluster in way that will be reliable and can be retrieved faster. By default, HDFS maintains three copies of data blocks in a cluster. Two copies are on the same rack and third one is on the different rack. Hadoop will internally make sure that any node failure will never results in data loss because if one node gets failure in same rack then recover data from second data node in the same rack which contains two copies of data block and in another case if the rack gets failure then recover data from another rack in which the third copy of the data block is maintained. HDFS is different from other distributed file system because it is fault-tolerant and low cost H/W.

A. Hadoop Cluster

A hadoop cluster is specially designed for storing and analyzing large amount of data. A hadoop cluster distributes the workload across multiple nodes that work to process the data in parallel. Hadoop cluster is the proper arrangement of the nodes for better performance and because of this arrangement it is easy to find the nodes between the racks. Hadoop cluster contains the racks and rack contains the actual data nodes.

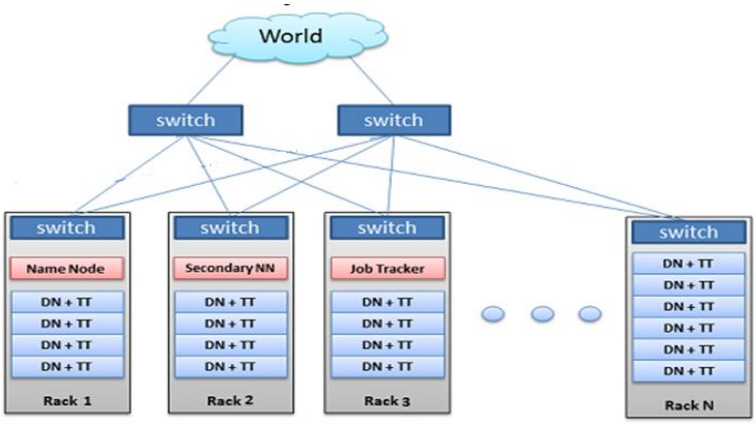

Fig.1. Hadoop Cluster

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

If anybody wants to know the no. of nodes in a particular cluster then it is compulsory to get information about the no. of racks in a cluster and no. of nodes per cluster. For example, if there are 40 racks in cluster and each rack have 100 nodes. Then this particular cluster contains 4000 nodes for storing and processing the data. The total no. of nodes is the product of no. of racks in a cluster and no. of nodes per rack.

In figure.1. the master nodes are the Name Node and Secondary Name Node for HDFS and Job Tracker is master node for MapReduce. HDFS is used for distributed data storage in cluster and MapReduce is used for distributed data processing. The Data Node is a slave for the Name Node and also for the Secondary Name Node and Task Tracker is slave for Job Tracker. Slave Nodes are the vast majority of machines and do all the work of storing the data and processing the data. Each slave runs both a Data Node and Task Tracker and these can communicate with each other. Each slave receive instructions from their master nodes and perform operations on data according to the instructions defined by the master node. The role of the Client machine is to load data into the cluster, describing how that data should be processed, and then retrieve or view the results when finish.

B. Architecture of HDFS

The architecture of HDFS is shown in figure.2. An HDFS cluster consists of a Name Node which manages the file system and regulates client access to files & Secondary Name Node for recovery process. Name Node connected to large no. of Data Nodes, store data as blocks. When client execute a query, it will reach out to the Name Node to get the file metadata information and then it will reach out to the Data Nodes to get the real data blocks. Data Nodes responsible

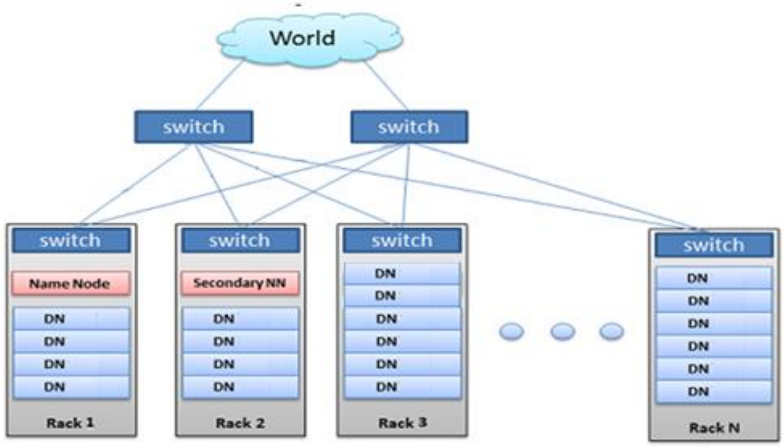

Figure.2.Architecture of HDFS

for serving read and write requests. Name Node determines mapping of blocks to Data Nodes. It performs operations like opening, closing and renaming files.

i. Name Node

It provides the file metadata information such as name of Data Node, name of block, name of file. The Name Node is the central controller of HDFS. It does not hold any cluster data itself. The Name Node only knows what blocks make up a file and where those blocks are located in the cluster shown in figure.3.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

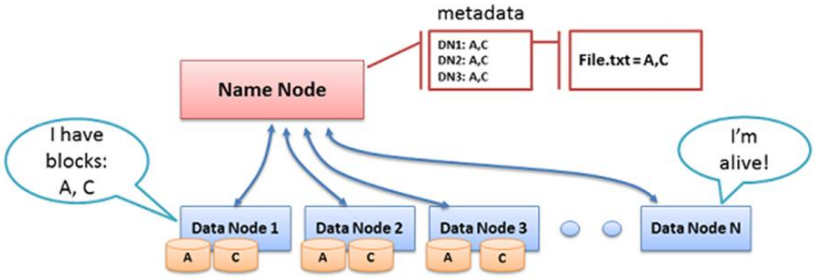

Fig.3.Name Node

During normal operation Data Nodes send heartbeats to the Name Node to confirm that the Data Node is operating. The default heartbeat interval is 3 seconds and each Data Node sends heartbeat after every 3 seconds. If Name Node does not receive heartbeat from Data Node in 10 min then Name Node consider that Data Node is unavailable & make decision for managing the data blocks of that Data Node. Every tenth heartbeat is a Block Report in which the Data Node tells the Name Node about all the blocks it has. The block reports allow the Name Node build its metadata. Metadata maintained by the Name Node which contains the name of files, name of blocks of each file and name of Data Nodes which contains the blocks of the file. The metadata is used for referring the Data Nodes.

ii. Secondary Name Node

The Secondary Name Node occasionally connects to the Name Node (by default, one hour) and gets a copy of the Name Node's metadata. The Secondary Name Node combines this information in a recent set of files. Secondary Name Node is shown in figure.4.

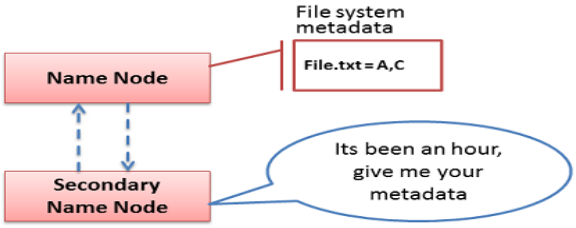

Fig.4.Secondary Name Node

If Name Node getting failure then the files retained by the Secondary Name Node can be used to recover the Name Node. The administrator may configure the Secondary Name Node setting for connecting with Name Node. During the failing condition of the Name Node, the Secondary Name Node provides the name of all files, name of blocks of each file and name of the Data Nodes for each block of a file. The time duration for communication is provided at the configuration time.

C. Operations in HDFS

HDFS is designed for distributed data storage and for storage there are two possible operations one is read operation and another is write operation. In read operation the client want some data for their use from the cluster which is already in a cluster and after analyzing the metadata the Name Node provides the needed data to the client. In write operation, the client wants to store data on cluster. The Name Node must update the metadata information for future references and provides the Data Nodes under the control of the client.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

i. Read Operation

When a Client wants to retrieve a file from HDFS it again consults the Name Node and asks for the block locations of the file. The Name Node reads metadata information and control of that Data Node is transferred to client for their purpose. The Client reads one block at a time.

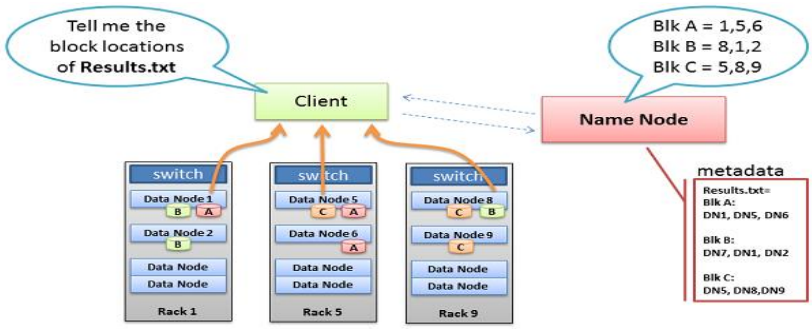

Fig.5.Read operation

In figure.5. the client wants to read the blocks of the results file and send query to the Name Node. Then Name Node performs the analysis of the metadata information. After analyzing the metadata the Name Node provides the location of the Data Node which has the needed data block. Then client can read block from the Data Node through Name Node. The Name Node is actually used for creating the connection between client and the accurate Data Node.

ii. Write operation

Multiple copies of blocks are placed in cluster to avoid data loss and for increasing the reliability. The default no. of copies is three, two on same rack & one on different rack. During write operation the client sends query to the Name Node and then Name Node transfer the control of Data Nodes to the client through Name Node. The Name Node automatically provides the Data Nodes in a way that will be reliable and can be retrieved faster. This operation is shown in figure.6.

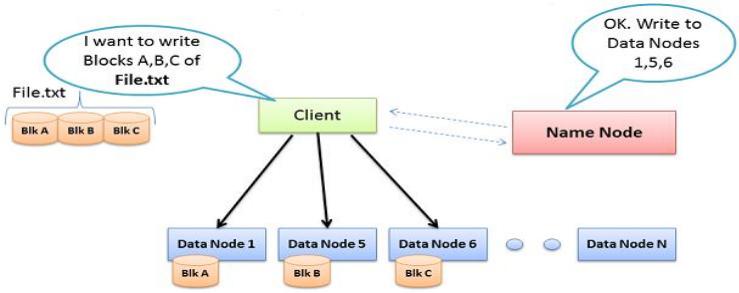

Fig.6.Write operation

In the above figure, the client wants to write the data blocks A, B & C of File. It sends the request to the Name Node then Name Node provides the Data Nodes for storing the blocks of the file such as 1, 5, 6. But the client does not know about the Data Nodes because it is the internal working of the HDFS. It provides only the storage space for data blocks.

D. Rack awareness

Rack awareness is maintained by the Name Node like metadata but it contains name of racks and the corresponding Data Nodes of particular rack. There are two reasons for Rack awareness: Data loss prevention, and network performance. If any rack fails in a cluster then the recovery process uses the rack aware and metadata information. In figure.7 if rack 9 gets failure then rack aware provides the names of the Data Nodes such as Data Node 8 and Data Node 9. Then metadata provides the names of the blocks which were retained by these Data Nodes and then Name Node takes perfect decision for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

recovery. Each block of data will be replicated to multiple machines to prevent the failure of one machine from losing all copies of data. Rack aware information increases the performance of the network because it provides the quick search results.

IV. CONCLUSION

The Hadoop distributed file system is widely used in cloud computing. This technology is provides the fast access to the data. The user can easily work with cloud Hadoop cluster. This technology has advanced features such as reliability, efficiency and security. But security issue exists in HDFS and others can refer this topic for future. HDFS architecture makes the read and write operations efficient with the use of hadoop cluster. After all simulation we conclude that it is the best technology in cloud computing.

REFERENCES

[1] Y. Zhu, H. Jiang, X. Qin, "Scheduling for Improved Write Performance in a Cost-Effective, Fault-Tolerant, Parallel Virtual File System", In the Proceedings of Cluster World Conference, 2003.
[2] http://en.wikipedia.org/wiki/ParallelVirtualFileSystem
[3] P. Braam, P. Schwan, R. Brightwell, "Portals and Networking for the Lustre File System", Sandia Corporation.
[4] http://www.parl.clemson.edu/pvfs/desc.html
[5] http://en.wikipedia.org/wiki/Lustre(fileSystem)
[6] Y. Zhu, H. Jiang, X. Qin, "Design, Implementation and Performance Evaluation of a Cost-Effective, Fault-Tolerant Parallel Virtual File System", Research Computing Facility at University of Nebraska, Lincoln.
[7] W. Ligon, R. Ross, "PVFS: Parallel Virtual File System".
[8] http://www.linuxjournal.com/content/lustre-distributed-file-system
[9] K. Shvachko, H. Kuang, S. Radia, R. Chansler, "The Hadoop Distributed File System", Proceedings in IEEE, 2010.
[10] http://wiki.lustre.org/manual/LustreManual20.html
[11] http://en.wikipedia.org/wiki/Apache_Hadoop
[12] http://www-01.ibm.com/software/data/infosphere/hadoop/