Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.565 – 569

RESEARCH ARTICLE

Efficient Resource Utilization in Hadoop on Virtual Machine

Jinto Thomas[1], Pavan Kumar V[2], Manjunath Mulimani[3]

[1]Department of Computer Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, Karnataka, India

[2]Assistant Professor, Department of Computer Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, Karnataka, India

[3]Assistant Professor, Department of Computer Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, Karnataka, India

[1] jintovelly@gmail.com; [2] pavan.cs@sahyadri.edu.in; [3] manjunath.cs@sahyadri.edu.in

Abstract— Hadoop is one of open source software technology that is used for processing large amount of data across clusters of commodity servers in distributed manner. Mainly it is designed to provide high fault tolerance and scale up a single server to thousands numbers of machines. Hadoop uses Hadoop distributed file system (HDFS) which is open source implementation of Google File System (GFS) for data storage. Map/Reduce is the main functionality used for storing data in HDFS. We have environment where Hadoop is deployed in virtual machine in which we can use Kernel-based Virtual Machine (KVM) as virtualization infrastructure. Existing service just use the entire resources available for admitted job. In such situation resource utilization is not proper efficient. It exceeds the limit of minimum resources which is required to finish the job. This configuration is resulting in poor resource utilization with higher cost. So avoid this create a new cluster for each job is assigned. Instead of customer to decide the resources for the job, this model automatically select desired systems for finish the job with minimum resource utilization.

Keywords— Hadoop, Cluster, KVM, Virtualization, HDFS

I. INTRODUCTION

Nowadays, the growing members of the organization and their business purpose data analysing huge amount of data. The data mainly flows from social internet web services and other social media. Day by day the processing of the analytical data becomes difficult in terms of normal computer performance speed that includes accessing and processing time. Hadoop is an open source implementation where we can decompose huge amount of data and dividing many jobs into smaller partitions such that each partition can process them in parallel. The main functionalities of Hadoop are MAP and REDUCE. MAP is a function that processes the partitions with the key value prior to produce an intermediate result.

REDUCE is the task which has the intermediate result generated by the MAP as input and it will merge all the intermediate keys with the intermediate values.

Hadoop is just idea for storing and retrieval of large amount of data. It requires HDFS (Hadoop Distributer Fill System) as its storage unit .Even though it has some resemblances with the current distributed file system, but they are different. HDFS has high degree fault tolerance and is built to be deployed on lower cost hardware [6]. In the last few years cloud computing of MapReduce technology evolved separately in address of processing large amount of data. The key idea of MapReduce is dividing the data into fixed set of blocks and processing them in parallel .Initially Hadoop was designed to process data in physical clusters .When the cloud emerged Hadoop was also deployed as virtual clusters .Thus it reduces the power consumption and resource utilization.

A. ***Hadoop Distributed File System***

Hadoop Distributed File System has master-slave architecture in that one single Name Node act as the master node and other slaves are Data Nodes. Master or Name node will have all the responsibility to manage file system and regular access of files by the client systems. HDFS give a file system namespace area where user can store data in files. This file can be divided into number of blocks and stored in Data Nodes as shown in fig. 1. Name Node will decide the mapping of file blocks to all Data Nodes.

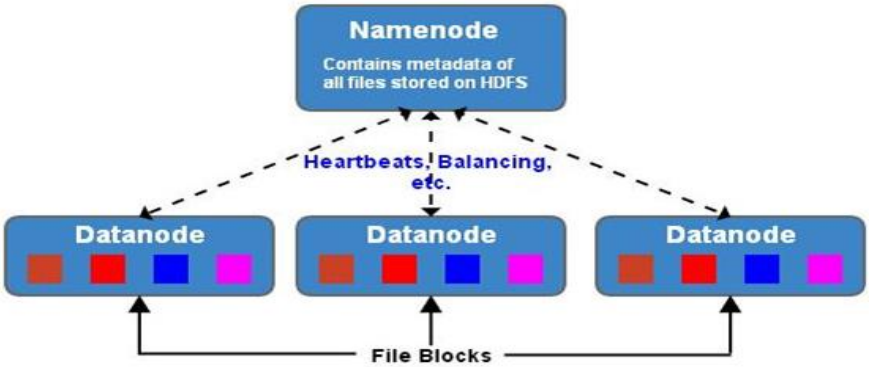

Fig. 1 HDFS Architecture

HDFS supports normal traditional file hierarchy structure. A user or an application can create and delete directories. Also store files inside these directories. Hadoop file system hierarchy is similar to most of the existing file systems; one can create and remove files, move a file from one directory to another, or rename a file. HDFS does not yet implement user quotas. But HDFS does not support hard links or soft links. The main objective of HDFS is to increase the reliability of data which is stored. It will be keeping minimum two copy of data block in different nodes. So failure of node will not affect the further processing. Here duplication of data will give high fault tolerance.

B. ***MapReduce***

MapReduce is framework proposed for analysing distributed data by Google in 2004. It mainly developed for easy way of processing and generating data with high fault tolerance and fast parallel processing. It is suitable for handling large amount of distributed job in environment like cloud [10].

MapReduce works by dividing process into two phases: the map phase and reduce phase. The map function takes input and generates key value prior to produce an intermediate result. Fig. 2 shows MapReduce function structure. The key/value pair generated by the map function will be stored in the disk as intermediate file. The intermediate file will shuffle/sorted and input to the reduce function that produce result.

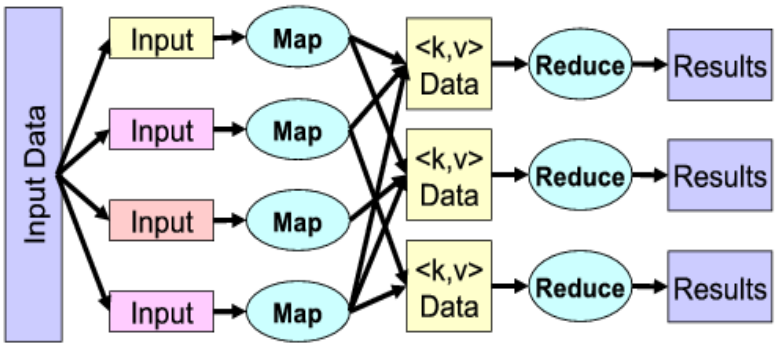

Fig. 2 MapReduce Structure

In this paper first we discussed about the work done by Hadoop in virtual machine to reduce the resource initialization .Secondly we'll discuss about the solutions and the algorithm which gives an efficient resource initialization in the model discussed above.

II. RELATED WORK

As we discussed in the previous section initially Hadoop system was running on physical machine .In HDFS usage of physical machine it leads to more power consumption and resource initialization like memory. To remedy this come up with a solution of displaying the Hadoop on virtual machine .The main advantages of these are one can completely utilize the resources, make Hadoop more reliable and save power .Based on these there are different models in this scenario. The existing cloud models are based on per job or per customer approach .For example, Amazon Elastic Map [10] customers will buy the clusters based on demand for each job .Once the job is submitted, cloud services will provide create cluster with VMS. Once the job is done destroy the cluster which was created.

Here resource utilization is restricted per job level. Another way is that a customer can buy or lease dedicated cluster from cluster services like Amazon Elastic Compute Cloud [8]. This approach optimizes resource initialization to the customer level. In this first operational model it is completely managed by the customers. Once the job is submitted, the resources are specified by the customers itself on the job basis. Here cloud provided ensures that only the requested resources are provided. Second one was partially customer based model. In this model cluster configuration is already done by the service provided. From the available clusters the customer will select one of the configuration to process the job. In both the cases risk is only for the services provided because when the job is submitted with a specific deadline, it is not the sure that all the jobs will be finished within the specified deadline. To solve this we have come up with a model which is completely based on the cloud services provided. In the existing system let us consider that the selected cluster configuration consists of 6 VMS and it finishes the job within 190ns. But the customer's deadline was 200ns. Same job if experimented with 5 VMS it will finish the job within 190ns. As an end user there is no much difference between 180ns and 190ns. Here we could have reduced the resource usage by deducting the number of systems by one. The solution what we suggest is that once the job is submitted, initially identify what is the minimum number of machines are required to finish the job. Then assign the jobs only to the required systems.

III.MODEL AND ARCHITECTURE

A. *Design*

The proposed model consists of 3 modules:

1. Application servers
2. Admin
3. Master node

Once the job is submitted by the customer with the deadline, the modelled system will calculate the minimum number of nodes which are required to finish the job within the given time. Once it is calculated, assign the job to the master node as shown in Fig. 3 Master will process the job with the selected slave nodes.

Fig.3 shows the architecture design of the proposed model. Here the customers will interact with the system through the application servers which will assign the job with the deadline .Once the job is assigned the admin will request the master slave for the information about the slave virtual machine. Information that includes CPU time, free RAM size and the

hard disk free space. Then calculate the minimum requirement for executing the job. Admin will be maintaining all the virtual machine details like ip address, login name and the password in his database. Whenever a new machine is added the details should be added by the admin. The main functionality is calculating the minimum nodes done by the admin. The proposed cluster configuration algorithm based on predictive analysis will ensure that the nodes are selected for the job in optimal manner.

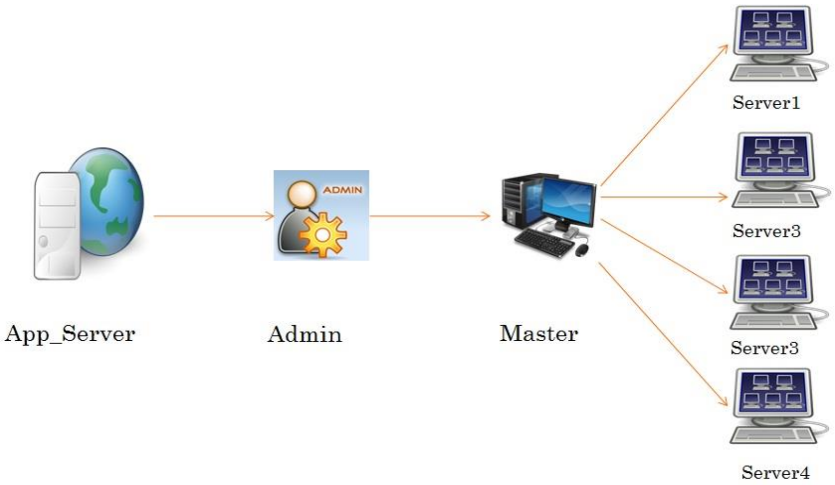

Fig. 3 Architecture of proposed model

B. ***Predictive Analysis***

Here during the production and testing of the product life cycle need to store all the processing information into database. It includes job name, size of the data processed, and time taken to produce the result. When a job is assigned, should see that same type of job already done before, to predict the minimum requirement this details will help in future.

C. ***Proposed algorithm***

Once a job is assigned, you need to check whether that type of job is assigned before. If yes then you need to go to the predictive phase of the algorithm. In predictive phase check whether any job is present which has equal or greater size than that of the submitted and whether the job can be finished within the given deadline. Select the job details from the retrieved data which has used minimum number of nodes. Assign job to the same system.

If there is no such job present in the database ten divide the job into different categories based on their file size. For instance, till 500MB category-1, 500-1024MB category-2 and above 1024MB category-3. Find the category of the assigned job. Retrieve all jobs from database having same category. Among those based on the previous performance, predict and select system for next processing.

Pseudocode 1: find cluster configuration with minimum number of nodes:

Require: jobsize=tsize; time=deadline;
Available_node=n; free_space[0...n]=hd_size;
total_size=sum_of_free_space[0..n];

Retrieve all jobs where **fzize>=jobsize** and **deadline == time**
if job is present **then**
Select the **job** with **min_no_of_nodes**

else
find the **category** of job;
num=jobsize/category;
limit=deadline;
num_sys=infinity;
Retrieve all jobs under selected **category**;
for all retrieved job **do**
if (num*time)<=limit **then**
if (num_sys>num_sys_of_job) **then**


```
                                    select=id_of_job;
                                    limit=num*limit;
                                    num_sys=num_sys_of_jon;
                            end if
                    end if
            end for
    end if
```

This algorithm will give best cluster configuration with minimum number of systems. This approach will help to reduce the source initialization.

IV. CONCLUSIONS

This paper presents a new MapReduce cloud service model, for data analytics in the cloud. Existing cloud services for MapReduce are insufficient and ineffective for production workloads. In contrast to current services, it select the best cluster configuration for the jobs using proposed algorithm, by deferring execution of definite jobs, allows the cloud provider to enhance its total resource distribution efficiently and reduce its costs. Also uses a sole secure instant VM provision technique that assurances fast response time guarantees for short interactive jobs, a significant proportion of new MapReduce workloads.

REFERENCES

[1] F. Jun, et al., "Evaluating I/O Scheduler in Virtual Machines for Mapreduce Application," Proc. Grid and Cooperative Computing (GCC), 2010 9th International Conference on, 2010, pp. 64-69.
[2] S. Ibrahim, et al., "Evaluating MapReduce on Virtual Machines: The Hadoop Case," Book Evaluating MapReduce on Virtual Machines: The Hadoop Case, Series Evaluating MapReduce on Virtual Machines: The Hadoop Case 5931,ed., Editor ed. Springer Berlin Heidelberg, 2009, pp. 519-528.
[3] D. Borthakur et al. Apache Hadoop goes realtime at Facebook In SIGMOD, 2011.
[4] B. Sotomayor, K. Keahey, I. Foster Combining Batch Execution and Leasing Using Virtual Machines in HPDC, 2010.
[5] A. Verma, L. Cherkasova, and R. H. Campbell Resource Provisioning Framework for MapReduce Jobs with Performance Goals In Middleware, 2011.
[6] K. Karun A and Chithranjan K, A review on Hadoop-HDFS infrastructure extensions, Proceeding of 2013 IEEE conference on Information and Communication Technologies,pp 132-137.
[7] M Ishii, J Han and H Makino, Design and Performance Evaluation for Hadoop Cluster on Virtualized Environment. IEEE conference on ICOIN, 2013, pp 244-249.
[8] B Palanisamy, A Singh, L Liu and B Langston Cost-effective Provisioning for MapReduce in a Cloud, 2013.
[9] Guanghui Xu, Feng Xu and Hongxu Ma, Deploying and Researching Hadoop in Virtual Machines, Proceeding of the IEEE International Conference on Automation and Logistics Zhengzhou, China, August 2012, pp 395-399.
[10] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simpli_ed Data Processing on Large Clusters, in OSDI, 2004
[11] Amazon Elastic MapReduce. [Online]. Available: http://aws.amazon.com/elasticmapreduce/, 2014.

