International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

On Cloud Data Distribution Handling using Big Data Deployment

M. Paul N Vijaya Kumar[1], Syed Umar[2], Sara Abdu Aliyu[3]

Assoc. Professor, Dept. of Information Science, Wollega University, Nekemte, Ethiopia[1]

Professor, Dept. of Computer Science, Wollega University, Nekemte, Ethiopia[2]

Graduate Student, Nekemte, Ethiopia[3]

ABSTRACT: Cloud computing is another new model in computing. Cloud computing offers a comprehensive on-demand infrastructure for different workloads. In practice, clouds are mainly used to process large amounts of data. Processing large amounts of data is critical in research and business environments. The greatest challenges when dealing with large amounts of data are the enormous infrastructures required. Cloud computing offers a huge infrastructure for storing and processing big data. Vms can be deployed to the cloud as needed to process data by clustering Vms. The map reduce paradigm can be used to process data where the mapping maps a portion of the task to specific Vms in the cluster and reduces the individual combined outputs for each Vms to achieve an end result. We have proposed an algorithm to reduce the global distribution of data and processing time. We tested our solution in the Cloud Analyst simulation environment and found that our proposed algorithm significantly reduces overall cloud computing time.

KEYWORDS: Big Data, data deployment, Cloud computing

I. INTRODUCTION

The management and processing of large amounts of data is becoming increasingly important in research and business environments. Big data processing engines have grown significantly. The biggest challenge when processing large amounts of data is the required infrastructure, which may require large investments. For example, cloud computing can significantly reduce infrastructure costs by providing new business models in which providers offer virtualized infrastructures when needed. Cloud computing offers a needs-based infrastructure for adapting to different workloads. The main way to crack data, sending data to the computed nodes that were shared. Big data is a collection of big data that cannot be processed using conventional computer technology. Big data technologies are important for a more in-depth analysis that can lead to real new decisions, resulting in higher operational efficiency, lower business risk and lower costs. For the processing of big data, there are various technologies from various providers such as Amazon, IBM, Microsoft etc. For cloud computing and a distributed file system, Hadoop offers open source construction.

Hadoop uses the Map Reduce model. HDFS is a file system for performing the tasks used. Map Reduce is used. By reading the entry in large parts, he treats this entry and finally writes large editions. HDFS does not handle direct access correctly. The HDFS service is provided by two processes: the name button and the data node. The name node manages the file system administration and offers administration and control services. The data node provides block storage and retrieval services. A node name process is running in the HDFS file system and this is a single source of error. Hadoop Core offers automatic name copying and node name retrieval, but there are no failover services.

The main goal of our proposed system is to significantly reduce the time it takes to distribute data between virtual IT clusters in the cloud. In this way, you can process and manage large amounts of data faster and more securely. Section 2 discusses various data distribution techniques. Related activities are presented in Section 3. Chapter 4 presents our proposed system design. Chapter 5 provides implementation details. Section 6 presents the experimental results of performing a task in Cloud Sim. Chapter 7 concludes the document with future work.

II. DATA DISTRIBUTION TECHNOLOGY

Data distribution is a technique for distributing partitioned data on multiple delivered virtual machines. This technique requires the load balancing factor to be taken into account to distribute the same amount of data across all VMs provided. The different approaches for distributing data between the offered virtual machines are [1]:

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

2.1 Central approach

This approach uses a central archive to download the required dataset from the virtual machines. In the initialization script, all virtual machines connect to the central repository, allowing the virtual machines to restore the required data after start-up. Central server bandwidth becomes a bottleneck for all transmission. The limitation of the centralized approach, when requests are made in parallel, is that the central server shuts down and creates a "flash crowd effect", which can be caused by thousands of VMs requesting data blocks.

2.2 Semi-central approach

If multiple virtual machines are requested in parallel from the central server in the centralized approach, the connections are disconnected from the server. Semi-central approaches can reduce the stress of network infrastructure. It is then possible to share the data set on different computers in the data center. In this way, the VMs are not published at the same time. The limitation of this approach is that records change over time. Withdrawals may increase or increase over time, making bias predictable.

2.3 Hierarchical approach

When new data is constantly being added, the semi-centralized approach is very difficult to maintain. The hierarchical approach creates a relay structure where the data is not retrieved from the VMs in the original repository, but the data from the parent node of the hierarchy. In this way, all virtual machines can access the central server to retrieve data, and this recovery data is made available to other virtual machines, etc. The limitation of this approach is that fault tolerance cannot be provided during transmission. If one of the virtual machines crashes, the installation of the virtual machine fails after start-up.

2.4 P2P access

The hierarchical approach requires more synchronization, and certain P2P streaming overlays such as PPLive or Sop cast, which are based on multiple hierarchical trees (one node belongs to multiple trees), are used to implement this approach. With this approach, each system acts as both a server and a client. To access virtual machines, the data center environment has low latency, no firewall or NAT problem and no ISP traffic shaping to provide a P2P delivery method for big data in the center

2.5 Literature survey

Different techniques for processing data distribution are proposed. The most effective technique for spreading data is peer-to-peer. In [2], S. Lough ran et al. Presents a framework that enables the dynamic implementation of a Map Reduce service in the virtual infrastructures of public or private cloud providers. The strategy is followed by Map Reduce to move the calculation to the location where the data is stored instead of the data in the compute node [3]. The deployment process architecture creates a set of virtual machines (VMs) based on user-supplied specifications. The framework automatically configures a full Map Reduce architecture using a catalog of services and then processes the data. Data partitioning is performed to distribute data to provisioned virtual machines. The data distribution of the data entry is divided. This item is divided into several parts. Partitioning involves partitioning at a central location. By processing the data, this data is distributed among the virtual machines. [1]. The service capacity of the p2p system has changed in two regimes.

III. PROPOSED ARCHITECTURE

One is the transition phase and the second is the stable phase. . In the analytical model of the transition phase, the high demands try to catch up with the system and track measurements that show an exponential growth in service capacity. In the second regime, the stationary phase, the service capacity of the p2p system evolves with and follows the load and speed with which the other party leaves the system.

1. In [6], Gang Chen presents the BestPeer ++ system that integrates cloud computing, databases and peer-to-peer technologies to provide elastic data sharing services. Rather than improving the usability of P2P networks, the database community has proposed a number of peer-to-peer database management systems. In [7], Mohammed Radi proposed a Round Robin service broker policy to select the data center to process the application. In [9], Tanveer Ahmed

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

Yogendra Singhr presents a comparison between the different policies used for load balancing using a tool called cloud analyst. Several compared strategies include Round Robin, Equalized Current Execution Load, Throttled Load Balancing. As a result, the overall response time of the Round Robin policy and the ESCEL policy is almost the same, and that of a limited policy is very low compared to the Round Robin policy and the ESCEL policy. In [10] Soumya Ray and Ajanta De Sarkar present a concept of Cloud Computing, as well as research challenges in the field of load balancing. It also focuses on the pros and cons of cloud computing. It also focuses on the study of the load balancing algorithm and a comparative overview of the algorithms in cloud computing with regard to resource utilization, stability, static or dynamic and process migration. In [11], Pooja Samal, Pranati Mishra presents the problem of load distribution on various nodes of a distributed system solved by the proposed works. This improves resource utilization and task response times by analyzing variants of the Round Robin algorithm.

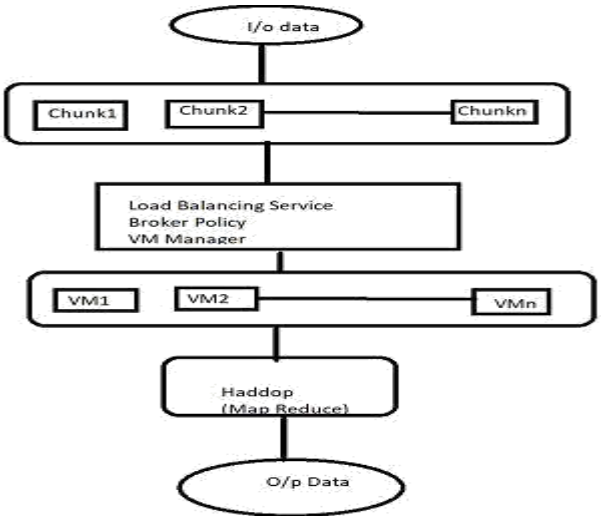

Fig1: Architecture of system

3.1 User input processing

The user can provide certain parameters to the virtual infrastructure, such as the number and size of the slave nodes. In addition to this user, also specify the input files and the path to the output folder. This is all that is done thanks to the interaction between the end user and the user with the system, the proposed framework automatically performs the rest of the process.

3.2 Centralized partitioning

The user can upload a huge file to the cloud. Centralizing data partitioning divides data entered in bulk into multiple parts. The user can download a large data file during partitioning, which is partitioned and stored on cloud servers, they divide large files into smaller parts and this also reduces the load on the storage server. The partition is automatically taken when the file is downloaded.

3.3 Data Distribution

The data chunks from the data repository are distributed to the VMs. These VMs will process the data. Distributing the data on each VM is an NP-hard problem. Our proposed system uses Peer-to-Peer data distribution technique in which there is point to point connection between the VMs. Service capacity of the p2p system is modified in two regimes. One is the transient phase and second is the steady phase. In transient phase analytical models busty demands are tried to catch up by system and trace measurement that exhibit the exponential growth of service capacity. In the second regime, the steady phase the service capacity of the p2p system scales with and tracks the offered load and rate at which the peer exits the system.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

IV. IMPLEMENTATION DATA

Installation of Ubuntu 18.04, Hadoop 2.7.1, Eclipse. The add-ons required by Hadoop are installed with the necessary apt-get-install Linux packages such as java JDK 1.7.

4.1 Cloud analyst:

Cloud Analyst is a tool based on a graphical interface developed on Cloud Sim architecture. Cloud Sim is a toolbox that can perform models, simulations and other experiments. The biggest problem with Cloud Sim is that every task has to be done programmatically. This allows the user to perform repeated simulations with small parameter changes in a very simple and fast way. With the cloud analyst, you define the location of the users who create the application as well as the location of the data centers. Cloud Analyst allows you to define various configuration parameters such as the number of users, the number of virtual machines, the number of processors, network bandwidth, the amount of memory and other parameters. The parameter-based tool calculates the simulation result and displays it graphically. The result is response time, processing time and costs. [13]

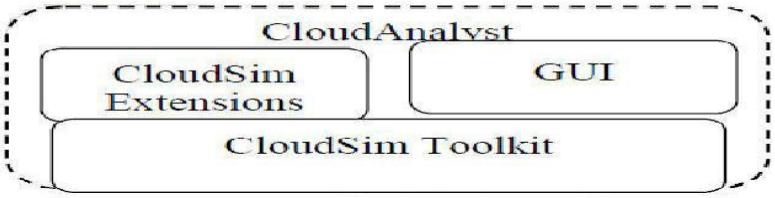

Fig2: Cloud sim Toolkit

4.2 Methodologies for problem solving and efficiency problems

The two main methodologies used to distribute the load between the virtual machines are the load balancing strategies of the strangled virtual machines and Round Robin.

1. Limited: in the limited algorithm to perform the requested operation, the client requests the load balancer to find an appropriate virtual machine. First of all, the process started with the preservation of the complete list of virtual machines, each line is indexed individually to speed up the search process. If a machine match is found based on size and availability, the load balancer accepts the client request and assigns the virtual machine request to the client. If no virtual machine meeting the criteria is available, load balancing returns -1 and the request is queued.

2. Round Robin: Round Robin is the simplest planning technique using the principle of time slots. In Round Robin, time is divided into several sections and a specific part of time is given to each node, i.e. it uses the principle of time planning. Each node receives a quantum and the node will perform its operations in this quantum. Based on this delay, resources are provided by the service provider to the requesting provider. Therefore, the Round Robin algorithm is very simple, but in this algorithm on the scheduler, there is an additional charge that determines the quantum size. [8]

V. PERFORMANCE ANALYSIS

The round robin load balancing algorithm and the limited load balancing algorithm are implemented for the simulation. The Java language was used to implement the VM load balancing algorithm. The configuration of the various components must be defined in the cloud analysis tool in order to analyze the load balancing policy. We need to define the data center configuration, user base configuration, and configuration parameters for application delivery. We took two data centers. The simulation time is 60 hours.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

|| Volume 8, Issue 4, April 2020 ||

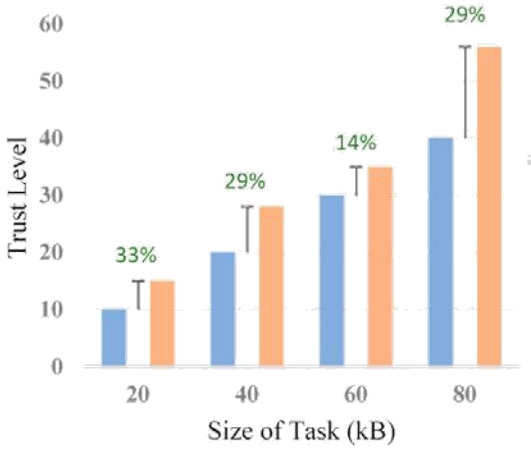

Fig3: Comparison of average response time of VM Load

In Cloud Analyst, the results are calculated after the simulation, as shown in the following images Figure 3 above clearly show that the overall response in the round robin is much greater, while the overall response time in the choked algorithm is improved. Therefore, we can easily identify between all algorithms. The choked algorithm is st. In this case, the transfer of the load takes place in a virtual machine

VI. CONCLUSION

This article presents a concept of cloud computing and focuses on the challenges of load balancing. It also focuses on the time it takes to process big data. We also conducted the comparative study of Big Data, Hadoop and Cloud Computing. We have also seen cloud connectivity with Adoop. A lot of time is spent studying various load balancing algorithms, followed by a reduction of the Hadoop map for data partitioning. This document is intended to perform a qualitative analysis of the VM load balancing algorithm prior to hm and then implement it with Hadoop in Cloud Sim and java. Load balancing in the cloud significantly improves the performance of the cloud service. This prevents server overload, affecting performance and improving response time. We simulated two different scheduling algorithms to execute user requests in a cloud environment. Each algorithm is followed and the scheduling criterion likes the average response time and data center derivation. We effectively used Hadoop to analyse data developed in the cloud.

REFERENCES

1.Luis M. Vaquero, Member, Antonio Celorio, Felix Cuadrado, Member, IEEE, and Ruben Cuevas, (2015) “Deploying Large -Scale Datasets on-Demand in the Cloud: Treats and Tr icks on Data Distribution” IEEE Trans. vol. 3, no. 2.
2.S.Loughran, J.Alcaraz, Caler oand J.Guijarro, (2012) “Dynamic cloud deployment of a map reduce architecture,” IEEEInternetComput.,vol.16,no.6,pp.40–50.
3.Jeffrey Dean and Sanjay G hemawat,(2008) “Map Reduce: Simplified Data Processing on Large Clusters”.
4.L. Garcés-EriceAffiliated with Institute EURECOM, E. W. Biersack, P. A. Felbe, K. W. Ross, G. Urvoy-Keller,(2003) “Hierarchical Peer-to-Peer Systems” Volume 2790 of the series Computer Science pp 1230-1239.
5. Xiangying Yang and Gustavo de Veciana, (2004) “Service Capacity of Peer to Peer Networks”IEEE
6. Gang Chen, Tianlei Hu, Dawei Jiang, Peng Lu, Kian-Lee Tan, Hoang Tam Vo, and Sai Wu, (2014) ”Best Peer++:A Peer-to-Peer Based Large-Scale Data Processing Platform” IEEE Trans. vol. 26, no.
7. Mohammed Radi, (2014)“Efficient Service Broker Policy For Large-Scale Cloud Environments” IJCCSA, Vol.2, No.
8. Tejinder Sharma, Vijay Kumar Banga, (2013)“Efficient and Enhanced Algorithm in Cloud Computing” IJSCE ISSN: 2231-2307, Volume-3, Issue-1
9. Tanveer Ahmed , Yogendra Singh, (2012)”Analytic Study Of Load Balancing Techniques Using Tool Cloud Analyst.” IJERA, Vol. 2, Issue 2, pp.1027-1030
10. Soumya Ray and Ajanta De Sarkar (2012)“Execution Analysis of Load Balancing Algorithm in Cloud Computing Environment” IJCCSA, Vol.2, No.5
11. Pooja Samal, Pranati Mishra, (2013)“Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing” IJCSIT, Vol. 4 (3) , 416-419
12. Samip Raut, Kamlesh Jaiswal, Vaibhav Kale, Akshay Mote, Soudamini Pawar, Hema Kolla
(2016)“Survey on Data Distribution Handling Techniques on Cloud” IJRAET, Volume-4, Issue -7
13. Bhathiya Wickremasinghe “Cloud Analyst: A Cloud Sim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environment”.