Geo-Distributed Data Center Allocation and Transfer of Large Data in Cloud VM

L akshmi Prasanna Ande[1], Devi Prasanna Kodi[2]
[1]M.Tech Student of Swarnandhra College of Engineering and Technology, Seetharampuram, AP-India
[2]Asst.Prof, CSE Dept Swarnandhra College of Engineering and Technology, Seetharampuram, AP-India

Abstract- Cloud computing provides useful IT services to consumers and to host multiple datacenters around the world. The dynamic flexibility of resource allocation in cloud computing is represented by virtualization technologies. Virtualization technology dynamically allocates server resources based on application queries, but there is considerable demand for large data exchange from data centers, such as large data migration. The challenge of planning different levels of urgent data transfer for processing large amounts of data is a difficult task to ensure successful data processing and analysis of large computer infrastructures. The skewness algorithm measures the uneven use of multiple resources in each virtual machine (VM) and an asymmetry based on the VM width load balancing is used. By minimizing the indirect value of each VM, you can easily combine different resources and improve the use of server resources. The proposed algorithm avoids overloading with effective load balancing and future load forecasting and minimizes power consumption by optimizing server resource utilization performance. As a result, the cloud computing model is profitable and has great potential, improved energy efficiency in dynamic load scenarios.

Keywords- Cloud Computing, Data Centers, Skewness Algorithm, Virtual Machine, Resource Management.

I. INTRODUCTION

Cloud computing systems that cover multiple geographic locations are now common in order to bring services closer to users, reduce energy costs, and ensure network / power failure reliability. Amazon, Google, Microsoft and Facebook have invested heavily in building large data centers around the world to deliver their services The basic requirement of this geo-distributed system of data centers is the transfer of large amounts of data from one data center to another, for example, Migration of virtual machines replication of content such as video and accumulation of large data, such as data genomically by many data centers to one for processing with a structure similar to MapReduce . Although dedicated broadband connections are commonly used in data centers of the same cloud service provider, most of the data volumes associated with transmissions between sites are often large enough to grow backbone networks, leading to competition for bandwidth between the various transmission tasks . The situation is worsening in far-reaching long-range marine fiber connections. A key challenge is an efficient planning of dynamic transfer new requirements between servers so that transmission jobs with different levels of difficulty reflected by the different data transmission deadlines can be dynamically and optimally configured to fully utilize the available bandwidth at any given time.

Here, a new optimization model is proposed for dynamic, highly efficient planning of bulk data transfers in geo-distributed data center systems in three ways. Data transfer models are modeled as data transfer tasks that delay tolerance with different end times due to the flexibility of the transmission plan.

First, it does to prevent the problem of sending large information to new, better transport routes. This optimization model increases the overall utility by completing a copy of this time. This optimization model allows you to dynamically update and monitor system, monitor schedules with dynamic data requests. Dynamic algorithm adjustments are easier in the future: the second dynamic algorithm is used to solve the best transmission bandwidth problem. These solutions refer to the various levels of efficiency optimization and complexity complexity. Third, the OpenFlow API-based SDN makes fundamentals plus and prepares the system algorithmically by replacing it. The actual experiment of the real world, along with three algorithms related to official, delay and overloading efficiency compared to existing literature algorithms.

II. RELATED WORK

A. Creation of Virtual Machine

Computational virtualization is the creation of virtual (non-physical) versions, such as hardware platforms, operating systems, and storage devices or network resources. VM Live Migration is a widely used technology for dynamic resource allocation in virtualized environments. A process that implements more than one logical computer system, resulting in a physical hardware set

.

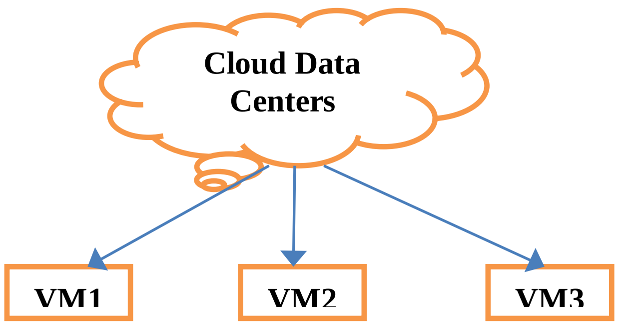


Fig.1: Virtual Machines

B. Resource management

Dynamic resource management has become an active research area in the cloud computing paradigm. Resource costs vary considerably depending on configuration. Therefore, efficient resource management is most important to both cloud providers and cloud users. The success of cloud management software depends heavily on flexibility. Ensure the productivity you need while providing the scale and efficiency to use critical hardware resources. Successful resource management solutions for the environment must begin to load, balance, and provide a rich set of resource controls for better isolation to efficiently use the underlying resources.

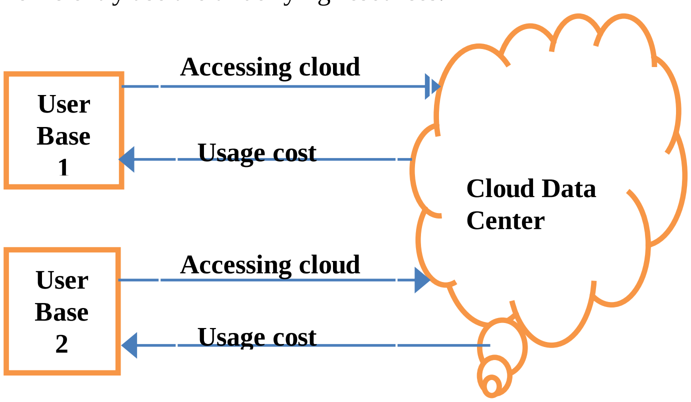


Fig.2: Resource Allocation to User Bases

C. Load prediction

Load prediction can capture the exact application resources used without looking inside the VM. This algorithm can identify trends in resource usage patterns and greatly reduce deployment progress. When load prediction is deactivated, the algorithm simply uses the last observed load when determining. The number of migrations in the load forecasting system is less than expected.

D. Load Balancing

cloud load balancing is the process of allocating loads and computing resources to one or more servers. This type of deployment provides maximum efficiency while minimizing reaction time. The workload can be divided into two or more servers, hard drives, network interfaces, or other computer resources to increase resource usage and system response time.Therefore, the effective use of cloud load balancing to ensure business continuity for high-traffic sites. The general goal of using a balanced load is:

- Maintain rigidity of the system.
- Improved system performance.
- Prevent system damage.

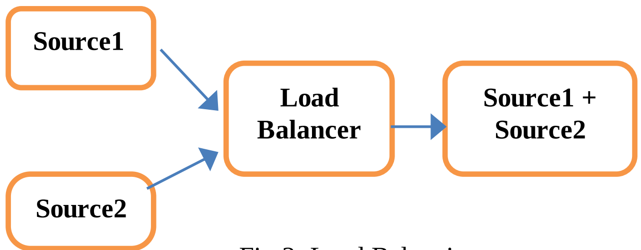


Fig.3: Load Balancing

E. Skewness Algorithm

Skewness measures the uneven use of the server. By reducing asymmetry, improve overall server utilization under multilateral resource constraints. If there is a connection, select VM, which can minimize the distortion of the server by removing it. Among all of the servers, accept one server as VM, the server that best reduce the distortion. If everything is the same, accept this VM and choose the target server that can minimize skewness. The skewness algorithm improves the overall use of server capacity by mixing loads with different resource requirements.

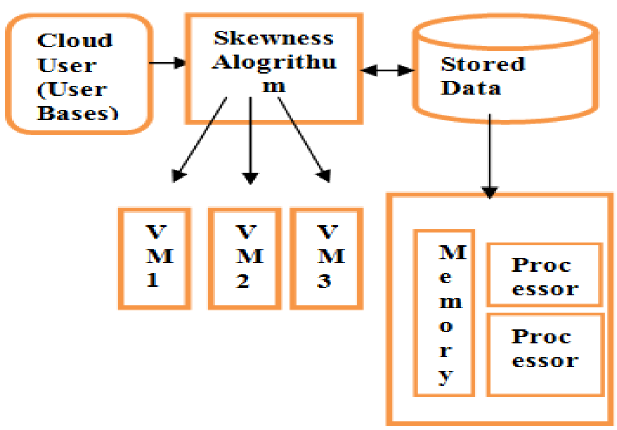


Fig.4: System Design for a Skewness Algorithm

The skewness Algorithm introduce the concept of skewness to measure inequality in the multidimensional use of server

resources. Scripts can be minimized to combine different types of workloads and improve overall use of server resources.
Skewness is a metric used to measure the symmetry of a dataset's distribution. The inaccuracy of the analysis domain is calculated as follows:

Skewness = $K_3 / E_{SD}{}^3$

Where:

If $n \geq 3$, $K_3 = [n\text{-}S_i\,(E_i\text{-}E_n)^3] / [(n\text{-}1) - (n\text{-}2)]$;

If $n < 3$, $K_3 = 0$;

The summation of all samples in the region
n is the number of samples in the total region
Ei is the linear value of the sample i and is set to 0 for each sample with $E_i < mS_v$ or $E_i > MS_v$,
E_n is the observed average energy of the area of the linear unit (m^2 / m^3)
mS_v is the minimum integral threshold during processing (dB re 1 m -1).
MS_v is the maximum integration threshold during processing (dB re 1 m -1).
E_{SD} standard deviation (standard energy deviation)

$$E_{SD} = \sqrt{\sum_i \frac{(E_i - E_n)^2}{n-1}}$$

This introduces the concept of strain in order to quantify the variation of the usage of multiple resources on the server. And that it is the number of resources we consider, n try to ri and is the use of resources of the ith. It define the resources and skewness of server P. Here, r is the average utilization of all resources of the server p. In fact, all types of resources since it is important to performance, it need to consider the bottleneck resources in the above calculations. By minimizing the distortion, we can use a combination of different types successfully loaded, to improve the overall utilization of the server resources.

Results:

<u>Response Time For Different Userbases</u>

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	212.51	40.86	361.62
Data Center processing time:	0.33	0.01	0.86

Response Time by Region

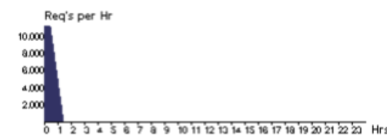
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	300.43	216.12	361.62
UB2	50.21	40.86	60.86
UB3	199.93	167.11	242.12

User Base Hourly Response Times

<u>Performance Of A Datacenter</u>

Data Center Hourly Loading

DC1



Cost

Total Virtual Machine Cost ($):	0.30
Total Data Transfer Cost ($):	0.12
Grand Total: ($)	0.42

Data Center	VM Cost $	Data Transfer Cost $	Total $
DC1	0.30	0.12	0.42

III. FUNCTIONALITIES

- Server storage virtualization: Data center load consolidation and balancing
- WAN Live Data Center Migration: A Strong Approach to Collaboration Environments
- Transfer data between mass centers using NetStitcher
- Resource and energy management of a hosting center
- Achieve high utilization with software managed WAN

IV. CONCLUSION

Design of a resource management system for cloud computing, deployment, and presentation and evaluation. Based on the changing requirements of adaptive multiplexing of physical resources, a system of virtual us. Since it corresponds to the capacity of the server being fully utilized, it is used as an asymmetric metric that combines VM resources and different characteristics. The algorithm has achieved both green calculations for a system with multiple resource constraints and to avoid overloading.

V. REFERENCES

[1]. http://www.datacentermap.com – data centers
[2]. https://pidatacenters.com/colocation/#data-center-infrastructure - infrastructure
[3]. https://edisciplinas.usp.br/pluginfile.php/98907/mod_resource/content/1/Chapter7-Cloud-Architecture-May2-2010.pdf
[4]. Z. Liu, M. Lin, A. Wireman, S. H. Low, and L. L. Andrew, ``Greening geographical load balancing,'' in *Proc. Int. Conf. Meas. Model. Comput. Syst.*, 2011, pp. 233_244. – load balancing.
[5]. ."Amazon elastic compute cloud (Amazon EC2)," http://aws.amazon.com/ec2/, 2012
[6]. N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter Bulk Transfers with Netstitcher," in Proceedings of the ACM SIGCOMM 2011 conference, New York, NY, USA, 2011, pp. 74–85.

Author Details:

Lakshmi Prasanna Ande
M.Tech(CSE),
Swarnandhra College of Engineering &Technology
Seetharampuram - 534281

Devi Prasann Kodi,
Assistant Professor,
CSE Department,
Swarnandhra College of Engineering &Technology
Seetharampuram - 534281