Energy Efficient Request Pool for a Green Private Cloud Architecture with Global Collaboration

Vidya D. Chachar[1], Nandkumar P. Kulkarni [2]

[1]Department of IT, STES SKNCOE Pune, India

Abstract: *In the new era of ICT, cloud computing is in demand. As cloud usage increases day by day, energy consumption required for its infrastructure also increases drastically. So it is critical to study energy consumption in cloud system infrastructure. Green Cloud Computing is the need in industries and research areas. GCC has become critical focus on economic and environmental policies of the countries. Cloud computing has some energy consumption defects due to its own features. To achieve the stability in cloud computing, unusable nodes would cause energy consumption and waste of resources. Client in cloud system exists with the number of redundant computing resources. The workload between servers and clients are distributed, so that entire computing resources cannot utilize effectively. To minimize above challenges, many researchers provided different solutions, but rarely provided specific framework or algorithmic solution to achieve GCC. In the perspective of cloud infrastructure, the existing systems consider the energy consumption only at server side. This paper modified existing RSDBASC (Resources Sharing Dynamically Balancing Algorithm with the cloud Servers and Clients) Algorithm and implemented new energy efficient request pool algorithm in order to achieve more energy efficiency in the cloud infrastructure. Proposed system saves the energy consumption at server as well as client side, So that entire computing resources can utilize effectively. Thus proposed system outperforms existing system in terms of average access time, resource utilization and energy consumption. There is no client starvation in the proposed system. Proposed system is energy efficient as compared to existing system by a factor of 29%.*

Keywords: Energy Efficient Frameworks, Green Cloud Computing (GCC), SCSN, Virtualization.

1. Introduction

GCC is efficient processing and utilization of computing infrastructure which reduce energy consumption. High-energy consumption translates to high operational cost, which reduces the profit margin of cloud providers. It also leads to high-carbon emissions which are not environment friendly. Hence, energy-efficient solutions are required to minimize the impact of cloud computing on the environment. Comparatively lower carbon emission is expected in cloud computing due to highly energy efficient infrastructure and reduction in the IT infrastructure [5].

There are studies as well where cloud computing leads to increase in energy consumption due to many reasons. So there is dire need to provide real-time solutions to save energy consumption in cloud infrastructure. Many Researchers provided different solutions, but rarely provide a specific framework or algorithmic solution to achieve GCC in respective of above problem. Junali Hu et al. proposed Green Private Cloud Architecture and algorithm based on virtualization to minimize the above challenges. This paper implemented request pool for existing system which receives all client requests and modified algorithm will forward the request of the client to server for better resource utilization.

Thus, with experiment it is achieved: reduction in average accessing time of computing resources and starvation of clients. This achieved the reduction in the energy consumption on the client as well as server side. The rest of this paper is organized as follows. Section 2 contains related work. Section 3 describes proposed system architecture. Section 4 elaborate mechanism and algorithms used. Section 5 shows the Experiment and its settings. Section 6 gives the results and its analysis. Section VII concludes with the future work.

2. Related Work

Cloud computing can consume more energy than traditional computing. Even with energy-saving techniques such as server virtualization and advanced cooling systems, cloud computing is not always the greenest computing technology [5]. Rongbo Zhu and authors presented the latest research advances in the area of green computing technologies, which mainly includes the development of truly sustainable computing technologies and related solutions. They categorized GCC as the green technology and mention GCC reflects diversity and the richness of green computing research activities [4]. Cloud computing can be seen as efficient processing and resource utilization of resources while minimizing the energy consumptions. Otherwise, growing demand of cloud computing may increase in huge energy consumption [1]. Using all the properties of cloud computing, GCC ensures that the future growth of cloud is environmentally sustainable and minimize the energy consumption of cloud and its components [2]. GCC can be achieved using different frameworks system resources, the resource utilization algorithm [6].

To avoid the problem of energy consumption in cloud computing Junali Hu and authors proposed and could achieve RSDBASC (Resource Sharing Dynamically Balancing Algorithm between cloud Server and Client) and Global Collaboration by which they could greatly improve the utilization of hardware resources and allocate the global resource more rationally[3]. In the existing system, to improve the PC terminal performance and to reduce a number of servers virtualization is used. By virtualization, heterogeneous systems can be deployed in a single physical machine. Thus implementing RSDBASC in green cloud architecture with global collaboration, they achieved great results for GCC. By proposed methodology [6] adding request pool for this system we could achieve more energy efficient results.

3. Proposed Methodology

This paper proposes new request pool for existing system RSDBASC [3]. It contains -

1. Request pool algorithm for receiving the client request
2. Modified RSDBASC.

A. Architecture for proposed System

Figure 1 shows architecture diagram. Architecture divided into 2-tier client and server. It uses virtualization (single physical server can be divided into no of VMs) and the volunteer computing concept where clients can transform into similar cloud server node (SCSN), which can reduce the increase in the number of servers [4].

- **Attribute Data Collector:** It will collect CPU Utilization, RAM Utilization and Reply time 3 attributes from each VMs.

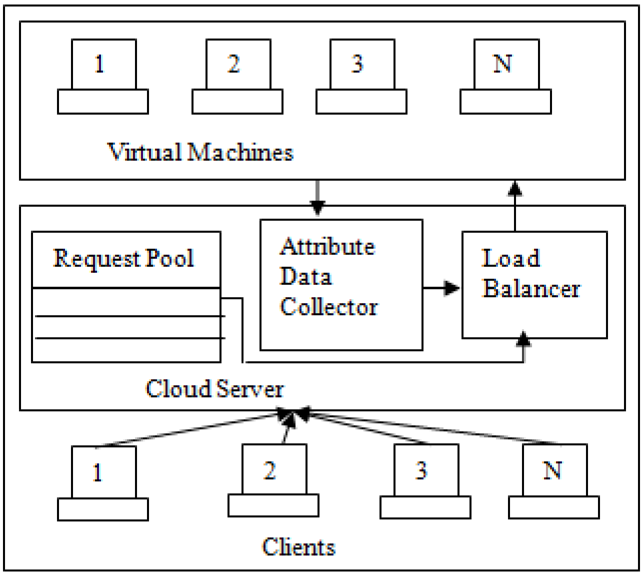


Figure 1: Architecture model of proposed methodology [6]

- **Load Balancer:** It is existing RSDBASC algorithm with the addition of new CPU utilization, RAM Utilization parameters. It mainly solves several key issues that include the equilibrium of sharing resources and the asymmetrically workload balance between the cloud servers and the clients in the Private Cloud system. Apart from RSDBASC, modified algorithm calculate each VM utilization (weight) and forward the client request to more efficient Cloud sever or VM.
- **Request Pool / Resource pool:** It will categorize all the client requests and dequeue the appropriate request to Load Balancer.

4. Mechanism

The core part of this system is RSDBASC Global Collaboration and request pool.

- **RSDBASC**: has multiple features - 1. Based on Virtualization, it is access mechanism of cloud servers and clients in the global collaboration. 2. It solves several key issues like equilibrium of sharing resources and asymmetrical workload balance between servers and clients in the private cloud.
- **GCMGPC**: can solve the problem from the competitive requirements of own computing resources, when the clients access the data and provide the data sharing service simultaneously [3].
- **Request pool**: All the client requests services by this pool, based on remaining packet size and average waiting time of corresponding packets.

A. Request pool Algorithm:

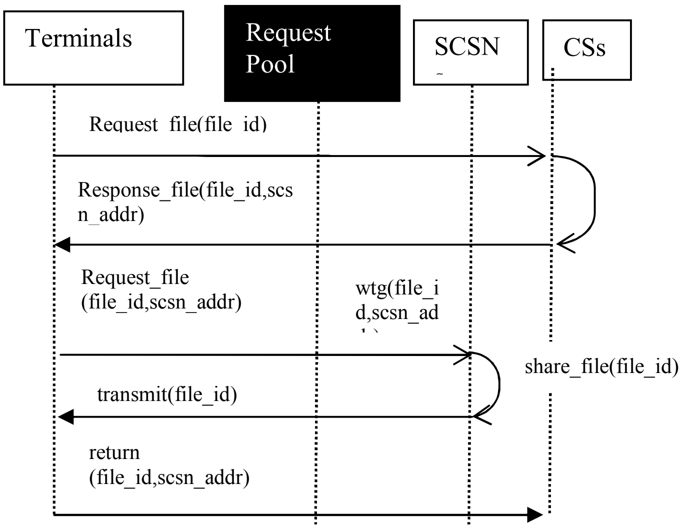
Step1: Clients send service requests to Cloud server.
Step2: These requests will be stored in the Request-Pool.
Step3: Select the requests based on no of the client requests and its waiting time in the queue.
Step4: Dqueue the request.
Step5: Forward this request to Load Balancer.

B. Process of Proposed System:

Figure 2 shows the process of proposed system. It shows how file is shared between CS, Terminal, SCSN and Request pool.

C. Key Indicator of Proposed System:

ti : The capture time of the file *i*
si : The size of the file *i*
l: The remaining computing resources
pi : The probability that file *i* is accessed in the future.
nw: The network bandwidth when file is sharing.
r: Current RAM utilization
c: Current CPU Utilization
uri : The update frequency of the file *i*
vri : The access frequency of the file *i*
r: The consumption of computing resources transmitted by each bit of data



Wtg= remaining packets * packetsize

Figure 2: The process of proposed system

TI- average time to capture files i [3]

$$TIi = \frac{1}{i}\sum_{i=1}^{i} t_i \qquad (1)$$

CI = identify how much resources consumption in future.

Here l = (100-c) + (100-r) + Remaining computing resources [3]

$$\text{CIi} = \frac{r \times si}{l} pi \qquad (2)$$

PI = performance improvement [3]

$$\text{PIi} = \text{Pi}\,\frac{vri}{uri} \times \frac{si - nw \times ti}{si} \qquad (3)$$

To reflect the priorities between server own resources and SCSN is present. Server own resources is Pwm and SCSN's resources as Pws.

$Pwm > Pws$ and $Pwm + Pws = 1$

CoIj : as the delay which the client j provide the sharing data, when it satisfy the own resource service. Share(i) = 1 when SCSNs are sharing the resources, otherwise it is 0 [3].

CoIi=

$$\frac{\sum\_(i\in SAL(j)) \; [\![Share(i)\times si]\!]}{Pws \times [\sum k\in Sact, k_{\neq j}\big((i\in SAL(j))\big)([\![Share(i)\times si)] + Pwm \times \sum k\in Sactask} \quad (4)$$

Based on CI and PI, the utility to enhance overall system performance of the sharing data service in the SCSNs can be reflected by the definition of the SI.

$$\text{SIi} = wp \times \text{PIi} - wc \times \text{CIi} \; [3]$$

The global collaboration RSDBASC can be expressed as the following optimization problem: Using (3) and (4)

$$\sum_{i\in Share} Share(i) \times PIi - \sum_{k\in Sack} CoIk \;\; [3]$$

5. Experiment

To create VMs software VMware is installed. To prove the heterogeneity of the system we used Windows XP/7 server and Red Hat Linux for deployment of the system. After analysis and experiment, we set Pwm = 0.6 and Pws = 0.4 for better results.

6. Result and Analysis

For experiment, we set different numbers of VLANs (3, 6, 9, 12, and 15) in different sets. Each set of experiment

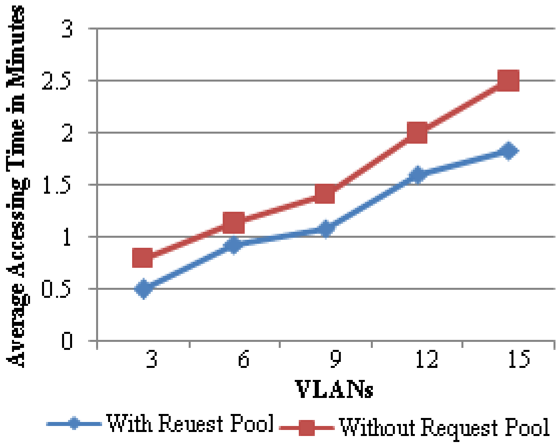


Figure 3: The comparison of the access time with the base system in the global collaboration

For experiment, we set different numbers of VLANs (3, 6, 9, 12, and 15) in different sets. Each set of experiment client is between 3 to 15. Number of files deployed are 15, from100KB to 400KB. It calculates sharing access time and the sharing degree of system resources. From the analysis of experiments, the proposed system shortening the average accessing time of data sharing as shown in Figure 3 and improves working efficiency of whole system as shown in figure .4. The energy saved is more as compare to base system as shown in figure.5.

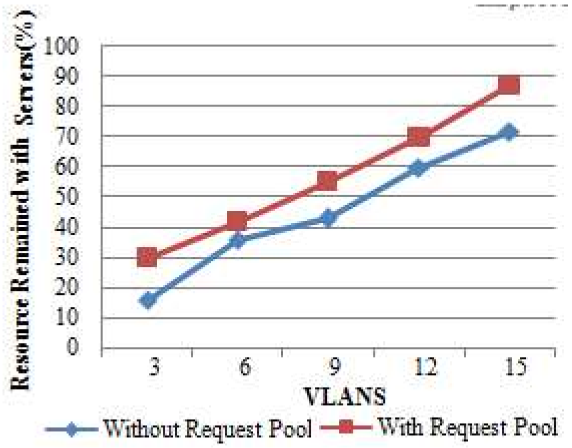


Figure 4: The comparison of the server utilization with the base system in the global collaboration.

7. Conclusion and Future Scope

Though the industry has adopted cloud computing, GCC is still in early stage. This paper, provided implementation of request pool on existing system to achieve more energy efficient cloud computing based on virtualization, with global collaboration. By experiment, proposed system achieved reduction in average accessing time of computing resources, efficient resource utilization and energy saving approximately 29%.

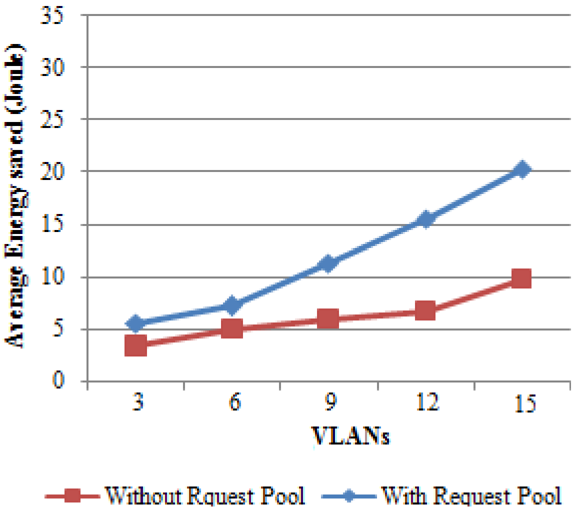


Figure 5: The comparison of average energy saved with the base system in the global collaboration

In the future work, the data service quality can be increase, threshold settings can be adaptive, and it can be good foundation for public cloud.

References

[1] Gregory Katsaros , Josep Subirats, J. Oriol Fito, Jordi Guitart, Pierre Gilet and Daniel Espling, December 2012 "A service framework for energy-aware monitoring and VM management in Clouds," Future Generation Computer SystemsElsevier,Vol. 29, Issue 8,pp 2077–2091.

[2] Jayant Baliga, Robert W. A. Ayre, Kerry Hinton, and Rodney S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," Proceedings of IEEE, Vol. 99, No. 1, January 2011

[3] Juanli Hu, Jiabin Deng and JueboWu, February 2013 "A green private cloud architecture with global collaboration," Telecommunication Systems Springer, Vol. 52, Issue 2 , pp 1269-1279.

[4] Rongbo Zhu, Zhili Sun and Jiankun Hu, February 2012 "Special section: Green computing, Future," Future Generation Computer Systems ElsvierVol. 28, Issue 2, pp 368-370.

[5] Saurabh Kumar Garg and Rajkumar Buyya, 2012. "Green Cloud computing and Environmental Sustainability," Harnessing Green IT: Principles and Practices edited by San Murugesan and G. R. Gangadharan, John Wiley & Sons, Ltd.

[6] Vidya Chachar and Prof. Nandakumar Kulkarni, October 2013- March 2014 "Resource Utilization Algorithm for Green Cloud Computing with Global Collaboration", Cyber Times International Journal of Technology & Management, vol. 7, Issue 1,pp 560-564.

[7] Cfmlprojects.org, June 2013, "Javasysmon.jar", Available at: <http://cfmlprojects.org/artifacts/org/getrailo/lib/javasysmon/0.3.4/>, [Accessed April 2014].

[8] Java.net, "appframework.jar", Available at: <https://java.net/projects/appframework/sources/svn/show/trunk/AppFramework>, [Accessed April 2014].

[9] Javaworld, May 2005, "SCSN Creation", Available at: http://www.javaworld.com/article/2071877/enterprise-java/peer-to-peer-applications-made-easy.html> [Accessed April 2014].

[10] Vmware 2011, "Virtualization and Cloud Computing" Available at: http://www.vmware.com, [Accessed July 2014].