Research Journal of Applied Sciences, Engineering and Technology 8(6): 736-745, 2014
DOI:10.19026/rjaset.8.1029
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: April 08, 2014 Accepted: May 10, 2014 Published: August 15, 2014

**Research Article**

# Design and Analysis of Adaptive Load Balancing Approach in Cloud Infrastructure

[1]N.R. Ram Mohan and [2]E. Baburaj
[1]Department of Computer and Information Technology, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India
[2]Department of Computer Science and Engineering, Sun College of Engineering and Technology, Nagercoil, Tamilnadu, India

**Abstract:** In this study an Adaptive Load Balancing (ALB) approach is developed to effectively balance the load distributed across the cloud servers to minimize bandwidth and energy consumption on service provisioning. Cloud computing infrastructure has evolved as highly scalable services with massive computation power and storage capability with the resources being provided as service by the cloud environment and guarantees the Service Level Agreement (SLA). However, the needs of the subscribers have grown to an extent that there requires a big active platform for load balancing even if the resources are shared. Besides, the cloud computing paradigm also needs to optimally balance the load at the middle of the servers in order to avoid hotspot and improve resource utility. To perform energy conservation in cloud infrastructures, the use of chronological traffic data from data centers uses a service request prediction model. Collaborative provable data possession scheme adopt Homomorphic verifiable responses and hash index hierarchy but the drawback is that the match index structure are not matched properly with clustering model. Different level of power tariffs and requests made to the servers affect the decisions, where to serve the cluster needs. SLA Laws on privacy includes a factor that decides whether the loads can be moved in or out of a cluster, whereas they affect the overall energy consumption. ALB approach balances the load from every cluster group by minimizing the bandwidth and energy consumption. With repetitive query messaging, ALB collects the information about the current load of other group and then computes the average energy and bandwidth consumption of each group. The ALB Approach not only balances the energy consumption but also enhances the utilization of resources with minimal bandwidth usage. Extensive level of experimental studies is conducted to illustrate the efficiency and effectiveness of the proposed method. An experimental evaluation is accepted out to estimate the performance of the ALB approach with Virtual Machine (VM) energy-efficient cloud data centers. Performance metric for evaluation of ALB approach is measured in terms of energy consumption, bandwidth utilization rate, performance tradeoff and response time to service request, load balance factor and clustering efficiency.

**Keywords:** Adaptive load balancing approach, bandwidth utilization, cloud infrastructure, data centers, query messaging, service level agreement, virtual machine

## INTRODUCTION

Cloud computing is changing the lifestyles and considerably modify the way the people parse information. On the other hand, Cloud provides various platforms enabling the huge level of diversity of terminal devices owned by individuals to operate. The next generation of user devices offers not only steady readiness for operation, but also steady information consumption. In such an environment, surroundings computing, information storage and communication becomes effective. Cloud computing is an effective way to provide mechanisms that includes convenient and secure infrastructure with reduced cost of operations. Cloud computing relies on the data centers as their primary backend of customers includes computing infrastructure. Cooperative provable data possession scheme adopt the technique of homomorphic verifiable responses and hash index hierarchy. The homomorphic verifiable responses and hash index hierarchy is still a challenging problem in scheduling with the length irrelevant to the size of data blocks as shown in Shanbiao and Yan (2012).

Cost based scheduling algorithm as demonstrated in Selvarani and Sadhasivam (2010) made efficient mapping of tasks that measures the cost of resource and computing performance and at the same time also recovers the computation ratio by grouping the user tasks according to rigorous cloud resources. But, however the improvement of algorithm does not

**Corresponding Author:** N.R. Ram Mohan, Department of Computer and Information Technology, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India

concentrate on independent task scheduling in Cloud environment. Cloud scheduler who considers both user requirements and infrastructure properties fails to focus on extending trustworthy collection of the other certain user requirements.

The trust measurements were discussed in Anbang and Imad (2013) called as the DC-C but failed in identifying the building up resource's RCoT and its integrity measurements. Highly decentralized information accountability framework keep following the definite handling of the integrated users' data in the cloud. In particular, object-centered approach as shown in Lin *et al*. (2012) enables surround logging mechanism together with users' data and policies but fails to confirm the integrity of the JRE and the authentication of JARs.

The most promising one is a model in which public verifiability is enforced and does not allow TPA to audit the cloud data storage without demanding users' time, feasibility or resources. External auditor called as Third Party Auditor (TPA), on behalf of the cloud user confirm the integrity of the data stored in the cloud. Data storage in cloud as shown in Gagare *et al*. (2012) utilizes public key based homomorphic authenticator with accidental masking privacy preserving public auditing.

The task of allowing a Third Party Auditor (TPA) as illustrated in Cong *et al*. (2011) confirms the integrity of the dynamic data stored in the cloud. The foreword of TPA eliminates the connection of the client through the auditing data stored in the cloud. TPA intact in achieving maximum cloud computing scale, in which public verifiability is enforced and does not allow TPA to audit the cloud data storage without difficult users' time, probability of resources.

In cloud computing, resource allocations range up based on the requirements elasticity and it is the key difference when compared to the existing multiprocessor task allocation. Scalable resources as shown in Jing *et al*. (2011) made economical allocation of resources an important problem but fails in considering tardiness of tasks, relaxed and considering bulk discount pricing. Automated calibration of resource allocation for parallel processing as described by Jianfeng and Wen-Syan (2009) does not assume availability of data statistics and application semantics but probe able tradeoff between parallelism benefits and overheads. Federated cloud a mechanism for sharing resources thereby increasing scalability. Allocating resources in cloud as demonstrated in Govindan *et al*. (2011) is a complex procedure to improve resource allocation and therefore agent based method fails in implementing the protocol model and testing the system using JADE which is a programming language.

Customers identify resource requirements such as memory, disk space, CPU, network bandwidth and seal them all mutually into Virtual Machines (VMs) by mapping it to substantial resources. Resource allocation in cloud is one of the major problems to be solved in cloud computing. Several researchers have conducted different methods for it. Ethernet over Wavelength-Division Multiplexing (WDM) as shown in Chunming *et al*. (2010) are one of the cost-effective means to support data transfers in this type of data-intensive applications. However, neither the traditional approaches begin light paths between given source destination pairs nor the existing application-level approaches. Traditional approach relies only on computing resources but take the fundamental power utilization for granted.

Dynamic Power Management (DPM) results in the majority of the savings as the average energy in cloud computing systems. The method DPM corresponds to the Dynamic Voltage and Frequency Scaling (DVFS) technology that ensures the hardware performance and energy consumption to balance the corresponding characteristics of the workload. A significant amount of energy is conserved as shown in Avinash *et al*. (2011) by migrating Virtual Machines (VM) running on underutilized equipment to additional machines and hibernate such underutilized machines. VM aims to design such a policy for energy-efficient cloud data centers. It makes use of chronological traffic data from data centers and use a service request prediction model.

Electric power capacity available in each area as shown in Kazuki and and Shin-Ichi (2011) assumes a cloud computing environment in which both processing ability and network bandwidth are allocated simultaneously. Power capacity for aggregating requests of multiple areas fails to switch between servers and bandwidth in sleep mode back to operating mode. The service is provided by the Cloud computing services provider. The resources are provided on-demand, to meet the requirement of the Service Level Agreement (SLA). Service Level Agreements (SLA) differentiates between workloads under bounded latency requirements for cost savings on geographical load balancing.

The workload to be executed as mentioned in Muhammad *et al*. (2012) did not succeeded to carry out probabilistic analysis for cost saving in demand response market. SLA in cloud computing is definite model as its end to-end Quality of Service (QoS) constraint because the process defines how abstract services interrelate to accomplish a definite goal. Furthermore, virtualization technologies transfer virtual machines to physical resources on the base of load change to accomplish the load balance of the complete system in a secure way.

Secure outsourcing mechanism for solving large-scale systems of Linear Equations (LE) as exposed in Cong *et al*. (2013) applies LU decomposition. The decomposition model to such large-scale LE would be

prohibitively expensive via a wholly different approach. Iterative method is much easier to execute in practice and only demands comparatively simpler matrix-vector operations. Statistical Process Control (SPC) charts as exemplified in Chang *et al*. (2013) identify performance anomalies and differential profiling to classify their root causes. By automating the tasks within the framework the model fails to expand the scope of automation, based on some detailed analysis of profiling data. Profiling data includes report generation of probable culprits and expect to find other areas of secure software development.

Security-Mediator (SEM) as depicted in Boyang *et al*. (2013) able to generate verification metadata (i.e., signatures) on outsourced data for data owners and decouples the anonymity protection mechanism from the PDP. All the works mentioned above provide mechanisms for security in cloud environment. Certain other problems related to security in the multi-cloud storage infrastructure have to be addressed. A definition of fairness in congested situation as presented in Tomita and Kuribayashi (2011) has multiple resource types, which are allocated simultaneously to each service request. On the basis of the above concepts, enhance the previous congestion control method so as to facilitate the fair resource allocation among users in congested situation. Next, work identifies a measure for evaluating fair resource allocation.

A decentralized fair resource allocation mechanism for such self-adaptation as shown in Rami and Vivek (2013), uses market-based heuristics does not enrich in CloudSim. Decentralized mechanism fails systematically while adding these modifications to scrutinize their consequence on the combined adaptation. A performance expressivity tradeoff as described in Guojun *et al*. (2010) professionally share confidential data on cloud servers using Hierarchical Identity Based Encryption (HIBE) system and the Cipher text-Policy Attribute-Based Encryption (CP-ABE) system. To achieve fine-grained and scalable data access control, encrypts each patient's PHR file as demonstrated in Kui *et al*. (2012). The multiple data owner scenario, divide the users in the PHR system into several security domains that very much reduce the key management complexity for owners and users (i.e.,) clients.

Based on the aforementioned techniques and methods, in this study, an Adaptive Load Balancing approach intends in consuming less energy and minimum bandwidth utilization by balancing load in data centers. In order to minimize the overall communication cost in cloud environment, ALB algorithm maintains certain level of balance with respect to clustering where a subset of cluster head is elected. Moreover, the main objective of ALB is to minimize the total response time of the tasks by distributing the workload. And also to extend the performance tradeoff results by distributing load equally to systems in cloud environment.

Additionally, ALB approach aims balancing the traffic flows in cloud computing system which in turn optimizes the energy consumption. ALB approach employs effective distribution of load system to enhance the Quality of Service (QoS) required for cloud application. On eliminating communication related delays and congestion related information losses in cloud environment, ALB improves the QoS.

## MATERIALS AND METHODS

We begin this section by considering an architecture view of Adaptive Load Balancing (ALB) which serves as the basis of our problem statement and also discuss how the ALB framework meets the design requirements discussed in the previous section. Two techniques for constructing our ALB are introduced: clustering server group, where the set of servers are grouped and repetitive query messaging is performed to satisfy the computational demand of scheduled load using the mathematical instruction followed in adaptive load balancing and repetitive query messaging with the help of clusters. The detailed process involved is discussed in the forthcoming section.

The process involved in Adaptive Load Balancing (ALB) is to find the most suitable group to share the load for avoiding imbalances with a minimum of engendered overhead. The energy resourceful scheduler for cloud computing applications with adaptive load balancing is designed to optimize energy consumption of data centers involved in cloud computing. The ALB approach treats communicational demands of the users equally important to that of the computing requirements. Moreover, the ALB approach equilibrium the communication flows produced by the users and consolidates users with a minimum amount of bandwidth consumption.

As communication flow by users in cloud infrastructure is extremely active and often difficult to predict, the ALB scheduler examines both the load on the links and the occupancy of outgoing queues. Further, the ALB allocates users required resources in such a manner that it offers the most of the obtainable bandwidth and penalizes resources whenever the load exceeds the available transmission capacity when the queues grows in size. The design of queuing analysis aids in ALB avoids tradeoff between the congestion and information losses. The overall architecture diagram of the Adaptive Load Balancing is shown in Fig. 1.

The ALB approach defined with the VM cloud data center follows few steps executed for every received cloud computing data center. A group of servers set connected to the VM data center networks with the highest available bandwidth, provided that at least one of the servers in the set accommodate the computational demands of the scheduled users. The available bandwidth is defined as an unused capacity of

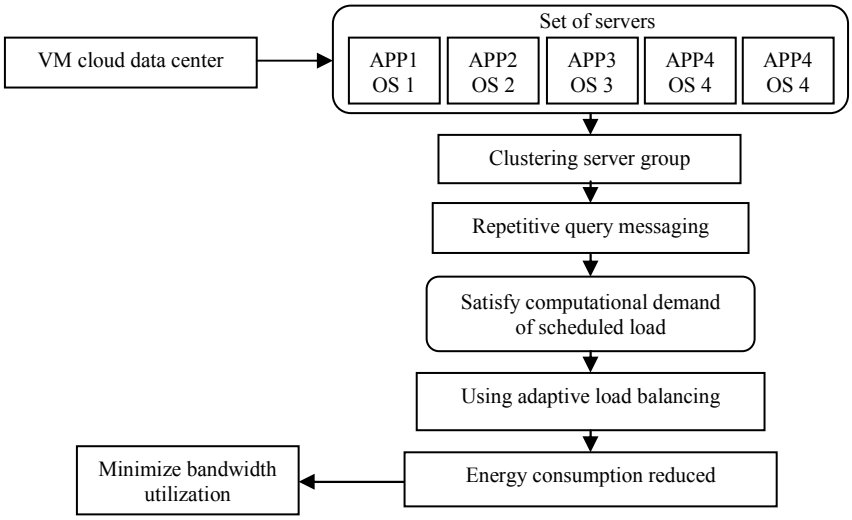


Fig. 1: Overall architecture diagram of ALB approach

the link connecting the group of servers (i.e., clustered server) to the rest of the VM data center network. Within the clustered group of servers, a computing server with smallest available computing capacity is chosen that satisfy the computational demands of the scheduled task.

Indeed, the ALB algorithm balances the loads with repetitive query messaging with little communications between the clouds, as the cloud are powered with servers. The ALB algorithm takes into account the cluster head each time when the imbalances occur with regard to a load threshold. The performance of Adaptive Load Balancing finally achieves minimal bandwidth utilization and reduced energy consumption through evaluation. The process of ALB is initiated using the mathematical instruction which is briefed below.

**Adaptive load balancing mathematical instruction:** In the VM cloud data center, the server is arranged in the form of frame 'F' and arranged in the form of component 'C'. Subsequently, frames form the set of components and select the group of servers with the largest available bandwidth. Moreover, ALB initially finds a component such that:

$$B(c_i) = max_{\forall c \epsilon C}(B(c)) \quad (1)$$

where, B is the available bandwidth of a component $c_i$ computed on a per-server basis. For a component $c_i \in$ C the available bandwidth computed is given as:

$$B(c_i) = \frac{Tc_i - \lambda c_i}{(Sum\ of\ c_i)} \quad (2)$$

where, T is the transmission capacity of a component $c_i$, calculated as a sum of maximum transmission speeds of all links connecting a component 'c' to the cloud infrastructure, λ is a currently effective transmission rate and $Sum\ of\ c_i$ is the total number of servers hosted in the component. Equation (2) provides an instantaneous measure of the available bandwidth. However, as most of the transmissions use full link capacity for a short response time, the available capacity is evaluated as an average over the time interval 'TI':

$$Bc_i(TI) = \frac{1}{TI}\int_0^{0+TI}\left(\frac{Tc_i - \lambda c_i(t)}{(Sum\ of\ c_i)}\right)dt = \frac{1}{Sum\ of\ c_i}\left(Tc_i - \frac{1}{TI}\int_0^{0+TI}\lambda c_i(t)dt\right) \quad (3)$$

Similar to the case of components, a frame is identified with the most of the available bandwidth, ALB finds a frame $f_i \epsilon$F such that:

$$Bf(f_i) = \max_{\forall f \epsilon F}(Bf(f)) \quad (4)$$

$Bf$ is the available bandwidth of a frame $f_i$ computed on a per server basis cloud infrastructure. For a component $f_i \in$ F the available bandwidth computed as:

$$Bf_i(t) = \frac{1}{TI}\int_0^{0+TI}\frac{Tf_i - \lambda f_i(t)dt}{Sum\ of\ f_i} = \frac{1}{Sum\ of\ f_i}\left(Tf_i - \frac{1}{TI}\int_0^{0+TI}\lambda f_i(t)dt\right) \quad (5)$$

where, $Tf_i$ is the transmission capacity of a frame 'i', calculated as a sum of maximum transmission speeds of all links connecting a frame 'i' of cloud group, $\lambda f_i$ is a currently effective transmission rate of the user and $Sum\ of\ f_i$ is the number of servers hosted in the frame in a cloud infrastructure.

**Repetitive query messaging through clusters:** Once the components and frame is identified with the most of the available bandwidth in ALB approach, a cluster

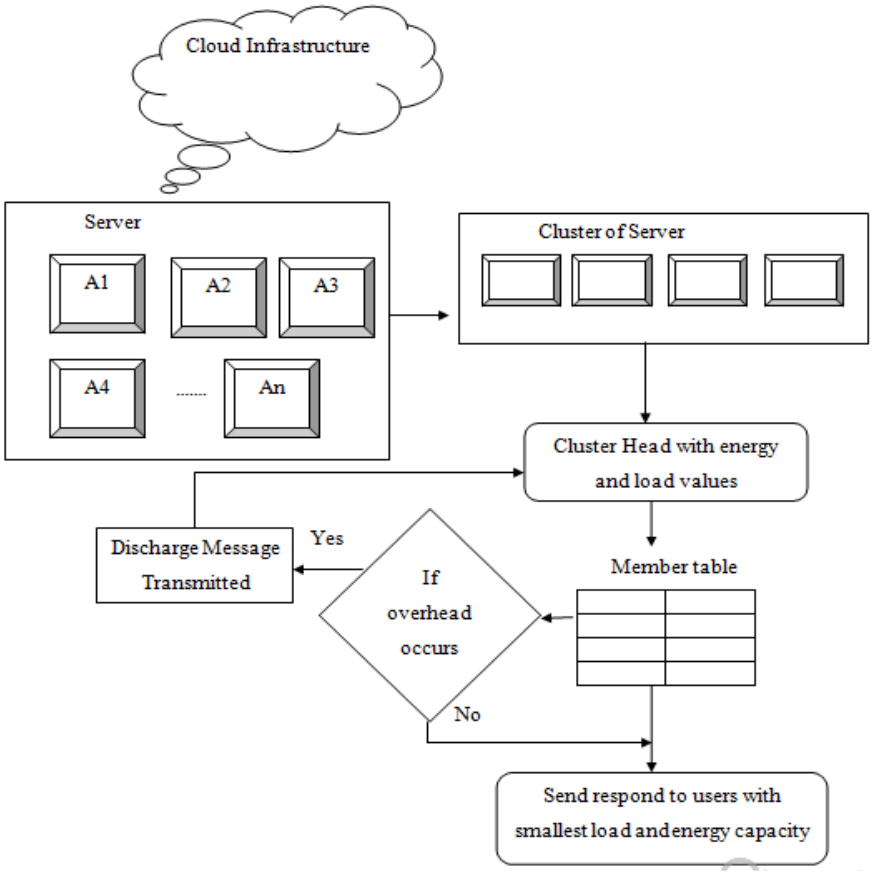


Fig. 2: Diagrammatic form of load balance using clustering

head is elected for its relatively high energy capacity. Energy is the critical resource in ad hoc networks. In ALB approach, the cluster head consumes less energy than a normal server as it has other functionalities to perform including coordination between its members, cluster maintenance and load balancing. Indeed, to avoid frequent changes in selecting cluster heads in VM cloud data center, the significant updated information exchanges are selected followed by cluster head evolution.

As mentioned through shape in Fig. 2, the pivotal role of a cluster head is to maintain the load balancing in each cluster. The cluster head periodically collects the information about each data center, such as energy and load values and the respective values are stored in the members table. Whenever a server attains the overload from users, a discharge message is transmitted to its cluster head. The secondary server consults its member table and chooses the one which has the minimum load and the energy capacity followed by a response sent to the concerned users. Whenever a new server joins a cluster group, the member table is updated accordingly. The load balancing algorithm followed for ADP is explained as below:

Begin
Initialize

**Input:** Time Interval 'TI', $Tf_i$ is the transmission capacity of a frame 'i', $Sum\ of\ f_i$ is the number of servers hosted in VM data center.

**Output:** Minimal bandwidth utilization and Energy Consumption.

Allocate VM cloud Data Center
  While TI>$Tf_i$ do
    Condition = False
  For 'n' servers do generate cluster head from server group
    Compute member table with energy and load values
    If (overhead occurs)
      Discharge Message transmitted to cluster head
    Else
      Member table send respond to users with least load and energy capacity
    End If
  End For
End While
End

The above pseudo code explains the process involved in ALB. The cluster head server maintains their member tables in order to control their member

loads. Periodically, each node, member of a cluster, sends a message and communicates its energy and load values to the cluster head which updates its table. The thresholds for ALB approach are defined for each server, in such a way that a server is able to carry out the process with minimum energy. Each server checks its load and energy and compares them with the two thresholds in a periodic manner. If one of the two thresholds is reached, the server sends a message (i.e.,) discharge message to its cluster head. The ALB consults its members table and further it chooses the one which has the smallest load and the energy capacity. If one such node is found, the cluster head sends a positive response to server indicating the address of the new server that will receive the extra load.

**Preventing congestion in server using ALB:** Once the process of clustering and query message is obtained, the next goal of the ALB approach is to maintain load balance and prevent server congestion. A supportive measure considered in ALB is to determine the available bandwidth within the data center. However, such a measure does not detain the system dynamics, such as unexpected increase in the transmission rate of the cloud applications. To have a more precise measure of the server congestion, ALB scales the measures of the available bandwidth with respect to the component and frame with the component related to the size of the row:

$$R(t) = 1 - \frac{1}{TI}\int_0^{0+TI}(e^{-\frac{(r(t)-1}{Rmax}})dt \quad (6)$$

where, $r$ is an instantaneous occupancy of the row measured at the time 't', $R_{max}$ is the maximum allowed size of the row. ALB aims to favor the empty row with minimum occupancy and penalize highly loaded rows. Moreover, the ALB also establishes the speed of load balancing with growing congestion control.

## RESULTS AND DISCUSSION

The Adaptive Load Balancing approach is measured against the Virtual Machine (VM) for energy-efficient cloud data centers. For experimental discussions, set of parameters are taken for the evaluation and implemented using JAVA Cloud Sim simulator. The specified toolkit has been selected as a simulation platform as it is a present simulation structure in Cloud computing environments. Compared to the simulation toolkits (e.g., SimGrid, CloudSim), it provides copy of on-demand virtualization enabled bandwidth and submission management. Virtual machine simulated data center comprises of 8 GB of RAM and 1 TB of storage. Energy consumption by the hosts is defined according to the ALB as described in Section above.

The user present needs for provisioning of 290 assorted VMs pack the power of the virtual data center. Each VM runs a web-application or any kind of application with variable workload, which is modeled to generate the utilization of bandwidth according to uniformly distributed random variable. The ALB approach is compared against the Energy conservation in cloud infrastructures with Service Request Prediction (SRP) model and Homomorphic Verifiable responses Hash Index Hierarchy (HVHIH) in terms of energy consumption, bandwidth utilization, performance tradeoff, response time, load balance factor and clustering efficiency.

The average amount of system energy used for the query processing in cloud structure is termed as energy consumption. The energy consumption of ALB against existing system is measured in terms of Joules (J) with bandwidth utilization performed to measure the usage of processing resources for balancing the load of each factor, measured in terms of Kilo bits per second (Kbps) whereas the performance tradeoff of ALB is the effective result obtained on the overall system in cloud infrastructure, measured in terms of percentage (%).

The response time for ALP is evaluated that measures the average amount of time consumed to response to the request send from the clients (i.e.,) users which is measured in terms of seconds (sec). The load balance factor for ALB across multiple servers evaluates the maximal throughput used to avoid the overload and increases the reliability which is measured in terms of Mega Bytes (MB). Finally, the clustering efficiency for ALB is defined with effective result on grouping of similar servers to improve the query processing result which is measured in terms of percentage (%).

**Performance of adaptive load balancing:** Adaptive Load Balancing (ALB) approach is compared against the existing Energy conservation in cloud infrastructures with Service Request Prediction (SRP) model and Homomorphic Verifiable responses Hash Index Hierarchy (HVHIH). The evaluation table given below and graph describes the ALB approach improvements when compared with existing system.

Table 1 and Fig. 3 describe the energy consumption based on the Virtual Machine counts (VM). The energy consumption is increased gradually as the virtual machine counts increases. As the count of machines increase in cloud infrastructure, the energy consumption of ALB is reduced to approximately 30-40% when compared with the SRP model in Avinash *et al*. (2011). This is because the ALB checks the load and its energy and compares them with two thresholds in the member table. The usage of member Table 2 information for query processing in ALB minimizes the

Table 1: Tabulation of energy consumption

| | Energy consumption (J) | | |
|---|---|---|---|
| VM counts | SRP model | HVHIH mechanism | ALB approach |
| 2 | 9.26 | 7.58 | 5.15 |
| 4 | 11.15 | 9.18 | 7.07 |
| 6 | 12.18 | 10.22 | 8.25 |
| 8 | 14.89 | 13.26 | 10.89 |
| 10 | 16.25 | 14.23 | 10.55 |
| 12 | 20.46 | 18.95 | 15.45 |
| 14 | 23.91 | 21.14 | 17.25 |

Table 2: Tabulation for bandwidth utilization

| | Bandwidth utilization (Kbps) | | |
|---|---|---|---|
| No. of tasks | SRP model | HVHIH mechanism | ALB approach |
| 5 | 2500 | 2205 | 2005 |
| 10 | 2600 | 2310 | 2150 |
| 15 | 2800 | 2355 | 2235 |
| 20 | 2920 | 2430 | 2460 |
| 25 | 2990 | 2645 | 2515 |
| 30 | 3265 | 2890 | 2750 |
| 35 | 3620 | 3245 | 3015 |

Table 3: Tabulation of performance tradeoff

| | Performance tradeoff (%) | | |
|---|---|---|---|
| No. of users | SRP model | HVHIH mechanism | ALB approach |
| 20 | 85 | 74 | 99 |
| 40 | 83 | 70 | 98 |
| 60 | 82 | 72 | 97 |
| 80 | 81 | 78 | 97 |
| 100 | 80 | 71 | 96 |
| 120 | 80 | 73 | 94 |
| 140 | 79 | 68 | 93 |

energy consumption which is dropped to 15-25% when compared with the HVHIH mechanism in Shanbiao and Yan (2012).

Table 2 describes the bandwidth utilization of SRP model, HVHIH Mechanism and ALB approach with respect to number of tasks performed and a graph is depicted in Fig. 4.

Figure 4 shows the bandwidth utilization of the ALB approach compared against the SRP model in Avinash *et al*. (2011) and HVHIH mechanism. As the task gets increased, bandwidth utilization in the cloud environment is reduced to 14-20% in ALB approach when compared with the SRP model. The bandwidth utilized of ALB approach is also compared with the HVHIH mechanism, where the result obtained is examined. The bandwidth of a frame in ALB approach is computed in terms of per server basis which reduces the bandwidth utilization to 2-10% when compared with the cloud infrastructure.

Table 3 describes the performance tradeoff of the SRP model, HVHIH Mechanism and ALB approach. The users taken for the evaluation are from 20, 40 and 60 up to 140 counts, respectively. As the user count increases, the performance tradeoff is also improved using ALB approach. The experimental result are plotted in graph and depicted in Fig. 5.

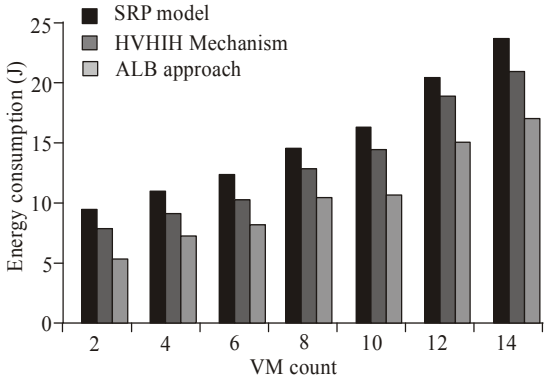


Fig. 3: Energy consumption measure

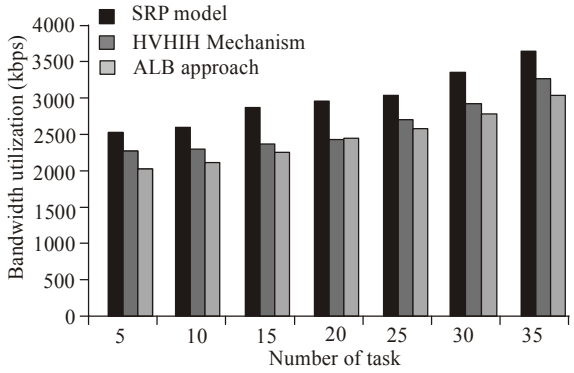


Fig. 4: Bandwidth utilization measure

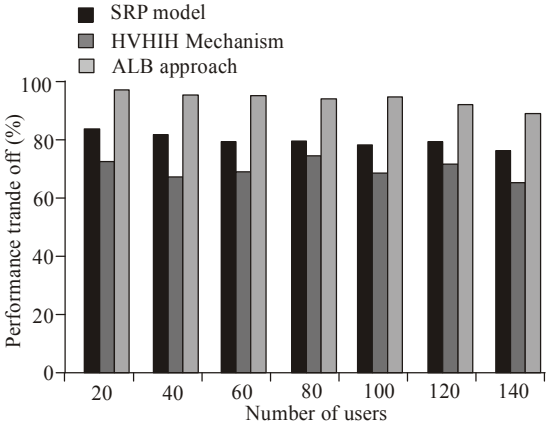


Fig. 5: Performance tradeoff measure

Figure 5 describes the performance tradeoff of the ALB and existing system. From the figure it is evident that performance tradeoff is improved using ALB, as the ALB scales the measures of the available bandwidth $B\ c_i(t)$ and $Bf_i(t)$ within the component with reference to the size of the row, which improves the performance tradeoff. The performance of ALB approach is approximately 14-16% improved when compared with SRP model in Avinash *et al*. (2011) and 20-30% improved when compared with the HVHIH mechanism.

Table 4: Tabulation of response time

| | Response time to service requests (sec) | | |
|---|---|---|---|
| No. of requests | SRP model | HVHIH mechanism | ALB approach |
| 4 | 1123 | 1051 | 999 |
| 8 | 987 | 904 | 852 |
| 12 | 1521 | 1410 | 1385 |
| 16 | 221 | 202 | 150 |
| 20 | 1830 | 1695 | 1423 |
| 24 | 2216 | 1999 | 1847 |
| 28 | 2565 | 2340 | 2245 |

Table 5: Tabulation of load balance factor

| | Load balance factor (MB) | | |
|---|---|---|---|
| Server number | SRP model | HVHIH mechanism | ALB approach |
| Server 1 | 45 | 58 | 70 |
| Server 2 | 77 | 82 | 90 |
| Server 3 | 48 | 55 | 67 |
| Server 4 | 56 | 68 | 78 |
| Server 5 | 39 | 45 | 56 |
| Server 6 | 55 | 65 | 77 |
| Server 7 | 60 | 70 | 80 |

Table 6: Tabulation of clustering efficiency

| | Clustering efficiency (%) | | |
|---|---|---|---|
| No. of frames | SRP model | HVHIH mechanism | ALB approach |
| 10 | 85 | 86 | 95 |
| 20 | 84 | 90 | 96 |
| 30 | 83 | 89 | 95 |
| 40 | 81 | 87 | 95 |
| 50 | 80 | 90 | 95 |
| 60 | 79 | 86 | 96 |
| 70 | 75 | 90 | 96 |

Table 4 and Fig. 6 describes the response time of the service request using the SRP model, HVHIH mechanism and ALB approach. The request taken for the evaluation is 4, 8, 12 up to 28 requests, respectively. The response time to service the request is better than the SRP model in Avinash *et al*. (2011) and HVHIH mechanism as the response time using ALB of a frame 'i', is calculated through $Tf_i$, where the maximum response speeds of all links connecting a frame 'i' in a cloud group is also analyzed. Equation (3) in ALB approach reduces the response time drastically when compared with the SRP model in Avinash *et al*. (2011). ALB response time is approximately 10-15% decreased in SRP model and 2-8% decreased using HVHIH mechanism.

The Table 5 describes the load balance factor of the ALB approach against the SRP model and HVHIH Mechanism.

Figure 7 illustrates the load balance factor using the three works SRP model, HVHIH Mechanism and ALB approach. The load balance factor of the ALB approach is improved as it obtains the cluster head each time when the imbalances occur with respect to load threshold. The load balance factor is 15-25% improved in ALB approach when compared with the SRP model

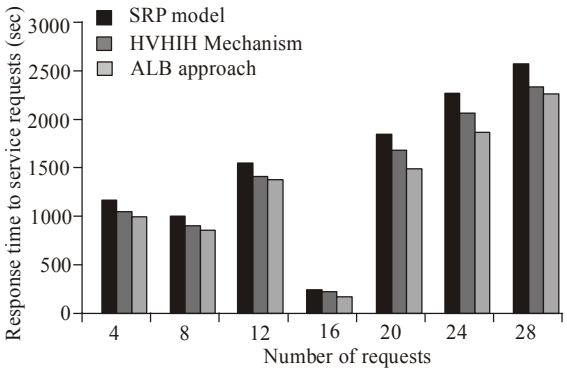


Fig. 6: Measure of response time

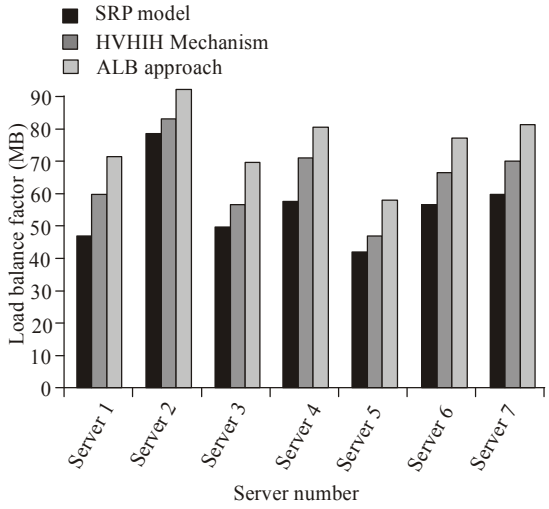


Fig. 7: Measure of load balance factor

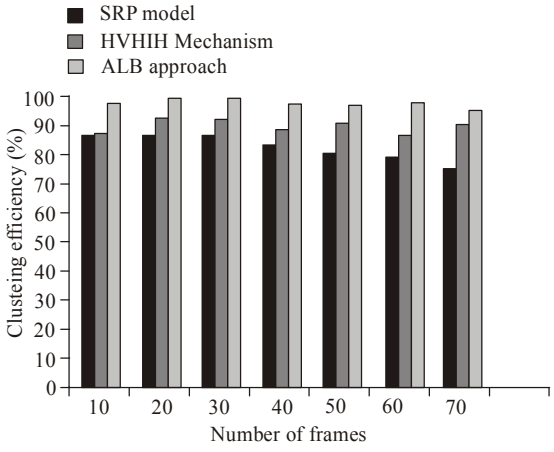


Fig. 8: Clustering efficiency measure

in Avinash *et al*. (2011) and 8-12% improved in the HVHIH mechanism in Shanbiao and Yan (2012).

Table 6 and Fig. 8 describe the clustering efficiency based on the frame count in the cloud infrastructure. Homomorphic verifiable responses and

hash index hierarchy in Shanbiao and Yan (2012) are not integrated with clustering model, so it attains the lesser efficiency when compared with the ALB approach. The clustered group of servers in ALB approach chooses a computing server to satisfy the computational demands. As a result, the ALB approach improves efficiency in clustering by approximately 8-18% when compared with the SRP model in Avinash *et al*. (2011) and HVHIH Mechanism in Shanbiao and Yan (2012).

As the final point of experimental evaluation, load balancing is an important factor that is to be analyzed to improve the response time for the service requests and minimal bandwidth utilization based on tasks. This is achieved by improved management of the energy by reducing load imbalance in cloud infrastructure based on clustering. The simulation results show a significant improvement of response time, bandwidth utilization and a good energy management for a great number of users with higher scalability in the size of work.

## CONCLUSION

The role of load balancing in modern VM cloud computing data centers presented a load balancing approach to optimize system energy consumption and bandwidth utilization. ALB provides adaptive load balancing in VM data center. In addition, the distribution of information between different servers helps to avoid congestion hotspots and information losses due to the overflow in cloud infrastructure. Moreover, the ALB collects the information about the current load of other group by a repetitive query messaging. As a result, ALB improves quality of service by reducing the delays related to communication and information loses due to congestion. The validation results, obtained from the VM cloud data center confirm the improvements in load balancing factor. The experimental result of ALB approach is compared with existing SRP and HVHIH system to attain minimal energy consumption and bandwidth utilization. Performance tradeoff is also improved in ALB approach with minimal time taken for responding to the requests. Clustering efficiency is also improved to approximately 10.71% using adaptive load balancing factor.

## REFERENCES

Anbang, R. and M.A. Imad, 2013. Towards trustworthy resource scheduling in clouds. IEEE T. Inf. Foren. Sec., 8(6): 973-984.

Avinash, M., M. Mukesh, D. Sanket and R. Shrisha, 2011. Energy conservation in cloud infrastructures. Proceeding of the IEEE International System Conference, pp: 456-460.

Boyang, W., S.S.M. Chow, L. Ming and L. Hui, 2013. Storing shared data on the cloud via security-mediator. Proceeding of the IEEE 33rd International Conference on Distributed Computing Systems (ICDCS, 2013), pp: 124-133.

Chang, L., C. Jinjun, N. Surya, P. Suraj and Z. Xuyun, 2013. A privacy leakage upper-bound constraint based approach for cost-effective privacy preserving of intermediate datasets in cloud. IEEE T. Parall. Distr., 24(6): 1192-1202.

Chunming, Q., Y. Dantong, J. Tao and L. Xin, 2010. Application-specific resource provisioning for wide-area distributed computing. IEEE Network, 24(4): 25-34.

Cong, W., R. Kui, W. Jia and W. Qian, 2013. Harnessing the cloud for securely outsourcing large-scale systems of linear equations. IEEE T. Parall. Distr., 24(6): 1172-1181.

Cong, W., L. Jin, R. Kui, W. Qian and L. Wenjing, 2011. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE T. Parall. Distr., 22(5): 847-859.

Gagare, G.J., P.P. Ghorpade, K.B. Jachak and S.K. Korde, 2012. Homomorphic authentication with random masking technique ensuring privacy and security in cloud computing. BIOINFO Security Inform., 2(2): 49-52.

Govindan, V.K., M.V. Haresh and S. Kaladyy, 2011. Agent based dynamic resource allocation on federated clouds. Proceeding of the IEEE Recent Advances in Intelligent Computational Systems (RAICS, 2011). Trivandrum, pp: 111-114.

Guojun, W., W. Jie and L. Qin, 2010. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. Proceeding of the 17th ACM Conference on Computer and Communications Security, pp: 735-737.

Jianfeng, Y. and L. Wen-Syan, 2009. Calibrating resource allocation for parallel processing of analytic tasks. Proceeding of the IEEE International Conference on e-Business Engineering, pp: 327-332.

Jing, F., K. Karthik, N. Yamini and L. Yung-Hsiang, 2011. Resource allocation for real-time tasks using cloud computing. Proceeding of IEEE 20th International Conference on Computer Communications and Networks (ICCCN, 2011), pp: 1-7.

Kazuki, M. and K. Shin-Ichi, 2011. Evaluation of optimal resource allocation method for cloud computing environments with limited electric power capacity. Proceeding of the IEEE 14th International Conference on Network-Based Information Systems (NBIS), pp: 1-5.

Kui, R., L. Ming, Y. Shucheng, L. Wenjing and Z. Yao, 2012. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE T. Parall. Distr., 20(20): 20.

Lin, D., S. Sundareswaran and A.C. Squicciarini, 2012. Ensuring distributed accountability for data sharing in the cloud. IEEE T. Depend. Secure, 9(4): 556-568.

Muhammad, A.A., G. Rajesh and S. Ryo, 2012. Energy efficient geographical load balancing via dynamic deferral of workload. Proceeding of the IEEE 5th International Conference on Cloud Computing, pp: 188-195.

Rami, B. and N. Vivek, 2013. A decentralized self-adaptation mechanism for service-based applications in the cloud. IEEE T. Software Eng., 39(5): 591-612.

Selvarani, S. and G.S. Sadhasivam, 2010. Improved cost-based algorithm for task scheduling in cloud computing. Proceeding of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp: 1-5.

Shanbiao, W. and Z. Yan, 2012. Secure collaborative integrity verification for hybrid cloud environments. Int. J. Coop. Inf. Syst., 21(3): 165-197.

Tomita, T. and S. Kuribayashi, 2011. Congestion control method with fair resource allocation for cloud computing environments. Proceeding of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp: 1-6.