An optimal VM Placement, Energy Efficient and SLA at Cloud Environment - A Comparative Analysis

Md.Rafeeq[1], Dr.C.Sunil Kumar[2], Dr.N.Subhash Chandra[3]

[1]Associate Professor, Dept of CSE, CMRTC, Hyderabad, Telangana, India: *rafeeq.mail@gmail.com*
[2]Professor in IT, SNIST, Yamnampet, Ghatkesar,Hyderabad, Telangana, India: *ccharupalli@gmail.com*
[3]Professor in CSE,CVRCE,Manglapalli, Ibrahimpatnam R.R(D), Telangana, India:*subhashchandra.n.cse@gmail.com*

Abstract- In the cloud computing framework, computing resources can be increased or decreased in response to the users' different application loads. The data is stored and the applications are running on the servers in the clouds. Users do not have to worry about lost or corrupt data. The clouds can distribute computing resources according to the users' needs or preferences to provide fl exible management. Users do not have to buy expensive computing devices. They only need to pay for the computing services provided by the clouds. Cloud computing provides a platform for computational experiments with abundant computing and storage resources. The system can be considered as a whole and the control and management decisions are sent as services to agents. The challenge in the present study is to reduce energy consumption thus guarantee Service Level Agreement (SLA) at its highest level.

Keywords — *load balancing, Service level agreement, Code Shortening, Energy efficient, Quality of Service (QoS), Service Level Agreements (SLA), Virtual Machine (VM), VM Allocation Performance Comparison, Evolution Application, Response Time Comparison.*

*****

1. INTRODUCTION

The load balancing techniques brings the advantage of lower response time [1]. However the cost of replication of resources is also to be taken care as an additional cost. The cloud data center based load balancing is distinguished from the domain name service based load balancing. The domain name service load balancers deploys the hardware and software components to balance load for the hardware resources, whereas the cloud based load balancing techniques deploys the software algorithms or protocols to distribute the load over multiple data center nodes. However the recent researches constraint to achieve the optimal SLA violation during VM Migration. Thus this work demonstrates A Service Level Agreement Effective Optimal Virtual Machine Migration Technique for Load Balancing on Cloud Data Centers using proposed three phase optimal virtual machine migration technique. To address this problem, the adoption of a technology called Virtualization is embraced. Through virtualization, a physical server can create multiple instances of virtual machines on it, where each virtual machine defines virtual hardware and software package on behalf of a physical server. In IaaS model, infrastructure requests are mainly served by allocating the VMs to cloud users [2]. Successful live migration of VMs among host to host without significant interruption of service results in dynamic consolidation of VMs. However, high variable workloads can cause performance degradation when an application requires increasing demand of resources. Besides power consumption we need to consider the performance as it puts Quality of Service (QoS) which is defined via Service Level Agreement (SLA). Storage systems come in all shapes and sizes, but one thing that they all have in common is that components fail, and when a component fails, the storage system is doing the one thing it is not supposed to do: losing data. Failures are varied, from disk sectors becoming silently corrupted, to entire disks or storage sites becoming unusable. The storage components themselves are protected from certain types of failures. To deal with these failures, storage systems rely on erasure codes. An erasure code adds redundancy to the system to tolerate failures. The simplest of these is replication, such as RAID-1, where each byte of data is stored on two disks. In that way any failure scenario may be tolerated, so long as every piece of data has one surviving copy. Replication is conceptually simple.

2. LITERATURE REVIEW

The Dynamic consolidation of virtual machines (VMs) is an effective way to improve the utilization of resources and energy efficiency in cloud data centers. Determining when it is best to reallocate VMs from an overloaded host is an aspect of dynamic VM consolidation that directly influences the resource utilization and quality of service (QoS) delivered by the system.comparing opti algorithm with few standard algorithms.

INTER-QUARTILE RANGE

It is method to allocation of virtual machine in cloud system. It is method of adaptive utilization Threshold which is work statics.

(IQR) interquartile range IQR= Q3- Q1, it is very similar of MAD(mean absolute deviation). MAD is another algorithm adaptive utilization Threshold.

We define the upper utilization threshold shown in

(i) Tu = 1- S.IQR……..(i)

Maximum correlation The Maximum Correlation (MC) policy is based other idea proposed by Verma et al. [7]. The idea is that the higher the correlation between theresource usage by applications running on an oversubscribed server[3].In [6] explain memory, CPU utilization and power

consumption of server over the physical machine**.** We proposed updating of virtual machine policy. Interquartile range have method middle quarter of statics which is 50% of data/request of allocation .We update the size of statics request of allocation of virtual machine.

IQR = Q3 - Q1…….

(ii) Where Q1and Q3 quarter of request**.**

Updated IQR = $Q_B – Q_A$ ……..

(iii) Where $Q_B = Q_1 /2$, $Q_A = (Q_3 + Q_4)/2$

In IQR both side data/request left some data/request that maximum data loss and time extend allocate virtual machine. In new IQR allocation policy and update various metrics help to allocate of virtual machine.

Algorithm

1. **Input** : HOST history **Output**:new assign
2. Dataget ←UtilizationHistory();
3. IfMathUtil.countNonZeroBeginning(data)> safe value
4. return updated IQR allocation;
5. IQR call mathUtill;
6. Go to assign VM Selection ;

TABLE 1: COMPARATIVE METRICS OF ALGORITHM

Metrics	Proposed Algorithm	Existing Algorithm
No of VM migration	5241	5502
Overall SLA violation	0.30%	1.05%
No of host shutdown	1254	1549
Execution time Total StDev	0.01973 SEC	0.02571sec
Execution time Total mean	0.01538sec	0.01290sec

IQR algorithm improve metrics host shutdown, less no of VM migration,less percentage of SLA violation and also improve other combination of selection policy with that allocation policy .like IQRmmt, IQRmu, IQRmr etc. lot of work left to more improve in real machine in term of security with help of encryption method andalso import more statics method in cloud system.

MAD(Mean Absolute Deviation)

An energy efficient algorithm for reallocation of resources using an adaptive technique, Median Absolute Deviation (MAD), of setting threshold values dynamically based on the set of VMs instantiated and past historical data of resource usage by the VMs. algorithm for dynamic VM consolidation that can reduce power consumption significantly. The data center consists of N heterogeneous physical hosts and capabilities of each host are characterized by the following three attributes:

(i) Performance of CPU, that is, Million Instructions Per Second (MIPS) it can execute;

(ii) Amount of the RAM; and

(iii) Network bandwidth provisioned for the host.

The problem of VM placement can be viewed as binpacking problem where the physical hosts are considered as differently sized bins and the VMs to be placed can be considered as objects to be filled in the bins.

VM selection: The selection of VMs to migrate from an overloaded host so that host becomes non-overloaded. The following two policies are taken into consideration for the purpose of comparative study:

a. **Random Selection**: Random Selection (RS) policy select VMs for migration as per the uniformly distributed discrete random variable $X^d = U(0 \mid v_j|)$,whose values index set of VMs V_j allocated to a host j [9].

Algorithm: MADL (hostList) # P_m - power model

1. Initialize simulation parameters;
2. Repeat
3. foreach host in hostList do
4. if(hostUtilization> 1- s-MAD ·) then
5. vmsToMigrate.add (LVF (host))
6. migrationMap.add(getNewVmPlacement (vmsToMigrate))
7. vmsToMigrate.clear( )
8. Repeat
9. foreach host in hostList do
10. if(isHostMinUtilized(host)) then
11. vmsToMigrate.add(host.getVmList( ))
12. migrationMap.add(getNewVmPlacement (vmsToMigrate)) 13. return migrationMap

Procedure: LVF (host)

1. Begin
2. migratableVms ← getMigratableVms (host)
3. migratableVms ← sortByCpuUtilization(migratableVms) //Sorting VMs of current host in ascending order of CPU utilization
4. return migratableVms.get(0) //Returns smallest VM in CPU utilization

Least VM in CPU Utilization First: VMs selected for migration according to RS policy may create a larger void between T u and A u causing to resource underutilization. Therefore, we propose Least VM in CPU utilization First (LVF) policy for VM selection to migrate off an overloaded host for the purpose of minimizing the gap between T_u and A_u .

The proposed VM selection policy sorts all the VMs in increasing order of CPU utilization that are executing on an overloaded host and selects a VM for migration present at 0th index of the sorted list, or smallest in terms of CPU utilization, in order to eliminate minimum loadfrom the host in each iterative step. The process repeats until the host does not become non-overloaded, that is, utilization of the host reaches below the threshold utilization (T_u) for the host

Table 2. Simulation Results of MADRS and MADLV heuristics

Energy efficient resource management techniques such as dynamic VM consolidation can cut-down CO2 emission and increase RoI for Cloud providers by switching-off the idle servers in order to eliminate idle power consumption.

the study will refer to the proposed MAD-MU algorithm in [8] and the proposed Shaw and Singh Algorithm implemented MM which is currently in Clouds framework and it employs MAD technique to determine the upper threshold, and MU method to select the VMs for migration. SSA, which depends on MM to select VMs for migration as well as determining the upper threshold, used DES for its best CPU utilization prediction in future.

Since the problem of dynamic consolidation of VMs in cloud computing data centres is wide extent, it is broken down into the four following phases [4]:

- Phase 1: Identification of overloaded hosts.
- Phase 2: Identification of underloaded hosts.
- Phase 3: Selection of VMs to migrate from overloaded hosts.
- Phase 4: Determining appropriate destinations for migration.

The study presented an optimized algorithm for identification of underloaded hosts and proposed an equation for calculation of the dynamic lower threshold. Using this threshold, VMs were migrated from underloaded hosts more accurately, allowing them to be switched off. This way, the researchers eliminated unnecessary migrations and decreased SLA violation, and on the other hand, optimized switch offs resulted in decreased energy consumption in the entire data centre.

LR(Local Regression) LRR(Robust local regression):

The mean value of the sample means of the time before a host is switched to the sleep mode for the LR-MMT-algorithm combination is 1933 seconds with the 95% CI: (1740, 2127). This means that on average a host is switched to the sleep mode after approximately 32 minutes of activity. This value is effective for real-world systems, as modern servers allow low-latency transitions to the sleep mode consuming low power. Meisner et al. [42] have shown that a typical blade server consuming 450 W in the fully utilized state consumes approximately 10.4 W in the sleep mode, while the transition delay is 300 ms. The mean number of host transitions to the sleep mode for our experiment setup (the total number of hosts is 800) per day is 1272 with 95% CI: (1211, 1333). The mean value of the sample means of the time before a VM is migrated from a host for the same algorithm combination is 15.3 seconds with the 95% CI: (15.2, 15.4). The mean value of the sample

Policy	Energy (kWh)	SLAV	VM migrations
MADLVF	69.46	0.58	12,465
MADRS	56.76	0.88	9,757
Difference	12.70	- 0.30	2,708

means of the execution time of the LR-MMT-1.2 algorithm on a server with an Intel Xeon 3060 (2.40 GHz) processor and 2 GB of RAM is 0.20 ms with the 95% CI: (0.15, 0.25).

The VM placement problem could be modeled as bin packing problem with variable bin sizes and prices. The physical nodes can be represented as the bin, VMs that have to be allocated could be viewed as the items, bin size can be seen as available CPU capacities and price can be seen as the power consumption by the nodes. The modified BFD was named PABFD (power aware best fit decreasing) algorithm which first sorts the VMs according to their CPU utilization in decreasing order and then for each VM it checks all the hosts and find the suitable host where the increase of power consumption is minimum. At final steps, it allocates the VM to that host. The algorithm is given as Algorithm.

Algorithm : Power aware best fit decreasing(PABFD)

```
1.  Input: hostList, VMList  Output: allocation of VMs
2.  VMList.sortDecreasingUtilization()
3.  for each VM in VMList do
4.          minPower ←MAX
5.          allocatedHost ←null
6.          foreach host in hostList do
7.              if host has enough resources for VM
8.                  power←estimatePower(host,VM)
9.                      If power < minPower
10.                     allocatedHost ← host
11.                     minPower ← power
12.         If allocatedHost ≠ null then
13.             allocation.add(VM,allocatedHost)
14. return allocation
```

The quality of the IaaS layer in cloud computing can be evaluated by keeping consideration of both power consumption and quality of service (QoS). In this work we put our focus on minimizing power consumption without making drastic alterations over the other areas, i.e., to meet the quality of IaaS. We follow some heuristics for dynamic consolidation of VMs based on the past resource usage data. We followed and did the same to detect both underloaded and overloaded hosts and also for VM selections as discussed earlier and in [3]. Now for VM placement, instead of using Best Fit Decreasing algorithm, we propose some additional algorithms based on the solutions of

bin packing problem that are likely to decrease the power consumption as well as maintaining the quality of service.

The main idea of the proposed adaptive-threshold algorithms is to adjust the value of the upper utilization threshold depending on the strength of the deviation of the CPU utilization. In case higher the deviation, more likely that the CPU utilization will reach 100% and cause an SLA violation. To calculate the upper CPU utilization threshold few statistical methods are used. These statistical methods to determine over-utilized and under-utilized hosts, and policies to select a VM to be migrated, can be combined to form various strategies. The destination hosts is chosen in order to minimize power consumption. Few of adaptive-threshold algorithms are based on statistical methods: Median Absolute Deviation (MAD), Local Regression (LR) and Interquartile Range (IQR).

Minimum migration time policy: Once assessing the host's CPU utilization levels and identifying any probable or overloaded host, VMs selection algorithm performs offloading of that host node to avoid any probability of SLA violation. The developed MMT selection policy performs migration of only those VMs (v) which requires minimal migration time than the other. In this paper, the migration time has been estimated in terms of the resource, RAM being used by VM divided by the supplementary network bandwidth available for host j. Let V_j be a set of VMs connected with the host j.

The Maximum Correlation(MC) policy is based on the idea proposed by Verma et al. [17]. The idea is that the higher the correlation between the resource usage by applications running on an oversubscribed server, the higher the probability of the server overloading. According to this idea, we select those VMs to be migrated that have the highest correlation of the CPU utilization with other VMs. To estimate the correlation between CPU utilizations by VMs, we apply the multiple correlation coefficient [38]. It is used in multiple regression analysis to assess the quality of the prediction of the dependent variable. The multiple correlation coefficient corresponds to the squared correlation between the predicted and the actual values of the dependent variable. It can also be interpreted as the proportion of the variance of the dependent variable explained by the independent variables.

THR(Threshold) POLICY: CPU utilization of the host node has been used for identifying overloaded hosts. Unlike conventional schemes, based on static threshold, in this paper we have developed a dynamic threshold based adaptive CPU utilization and overload detection scheme. It enables the proposed system to behave in real time scenario where there is highly fluctuating resource utilization. It adjusts resource utilization threshold based on the variation in CPU utilization and utility map. It can be observed that higher deviation might even result into 100% CPU utilization that signifies higher overloading probability. To enable dynamic threshold detection scheme, we have used IQR and LRR algorithm. In our proposed model, the resource utilization has been examined at the interval of 5 min and on each odd iteration, IQR algorithm has been used, while LRR has been scheduled for even iterations.

The performance has been compared with different approaches like conventional IQR, LR, MAD, THR algorithm based CPU utilization estimation scheme and MMT, MC and RS based VM selection. Here, it should be noted that the other existing approaches (IQR, LR, MAD, THR and LRR) have been employed with BFD based Considering better efficiency of IQR and LRR.

CPU utilization threshold	VM selection			VM placement
IQR	MMT	MC	RS	BFD
LR	MMT	MC	RS	BFD
MAD	MMT	MC	RS	BFD
THR	MMT	MC	RS	BFD
LRR	MMT	MC	RS	BFD

Table 2. Implementation and simulation scenarios

To perform better analysis, we have examined different algorithms with three different VM selection and placement policies. The proposed Optimal VMM Techniques based consolidation exhibits minimal migration thus enabling minimal downtime probability. SLAV, where the proposed evolutionary computing based proposed system has exhibited minimal SLA violation with MMT selection policy. the SLA performance degradation, where it can be observed that the proposed A-GA based VM placement strategy with MMT VM selection policy and LRR based overload detection and resource prediction can enable minimum performance degradation.

3. PROPOSED WORK

This work proposes a novel framework for capturing and storage of traffic data. During the multi node traffic data analysis, controlling the replication in order to reduce the cost is also been a challenge. This work also addresses this problem using Erasure encoding for low cost replication. The recent research outcomes demonstrate the use of agent based sensor networks to accumulate the road traffic data. However a multipurpose framework for accumulating and managing the traffic data is still a demand.

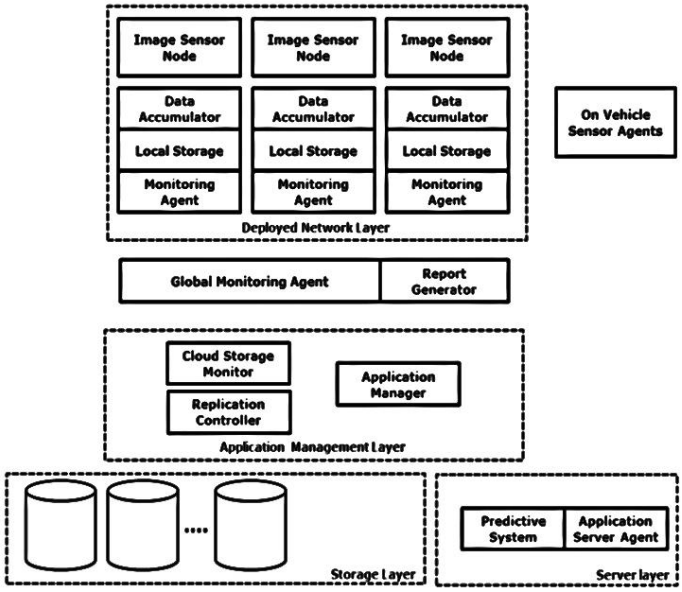

Figure – 1: Proposed Framework for Road Traffic Data Management with replication control

Thought, those approaches does not propose any technique to reduce the cost and improve the service level agreements to match with the current industry and research demands. Thus, this work proposes a cloud based automatized framework for virtual machine migration to increase the SLA without compromising the cost for storage and energy. The major achievement of this work is to minimize the SLA violation compared to existing virtual machine migration techniques for load balancing. The extensive practical demonstrations of virtualization and migration benefits are also carried out in this work. With the extensive experimental setup the work furnishes the comparative analysis of simulations for popular existing techniques and the proposed framework.

we propose the novel framework for road traffic data management control with replication control on cloud storage. In the proposed framework we have considered the layer based approach for better controlling and management of the agent based components. The agents in the wireless network are single function oriented but the collective network is multi-purpose.

In this study we propose the framework consisting of deployed network layer, monitoring layer, application management layer, storage layer and finally the server based server layer.

4. PROPOSED NOVEL VIRTUAL MACHINE BASED LOAD MIGRATION TECHNIQUE

Load Balancing Techniques on cloud computing is the generic framework based process where the generated workloads are distributed over multiple data center resources. The load balancing techniques brings the advantage of lower response time [1]. However the cost of replication of resources is also to be taken care as an additional cost. The cloud data center based load balancing is distinguished from the domain name service based load balancing.. However the recent researches constraint to achieve the optimal SLA violation during VM Migration. Thus this work demonstrates A Service Level Agreement Effective Optimal Virtual Machine Migration Technique for Load Balancing on Cloud Data Centers using proposed three phase optimal virtual machine migration technique.

The Virtual Machines are hosted by all service providers with similar configurations but with added advantages. Hence adopting to Virtual Machine computing is the best choice to avoid the lack of support and facility availability.

Optimal Migration Cost Control

Due to the tremendous competition in the cloud service provider space, the drop of price for each virtualization component used in the virtual machine configuration is dropping with an increasing speed. Hence rather than up-gradation cost for traditional systems, the cloud based virtual machines are very much cost effective [Table 7].
This work deploys a cost evaluation function to determine the most suitable virtual machine to be migrated considering the least SLA violation.

The proposed framework is classified into three major algorithm components as VM identification, VM migration and CAlgorithms for all three phases are been discussed here.
The proposed framework is classified into three major algorithm components as VM identification, VM migration and Cost Function. Algorithms for all three phases are been discussed here:

- Virtual Machine Identification
- Virtual Machine Allocation
- Cost Function

Table 3.Reduction of Cost for Virtual Machine Migration

Server Type	Amazon Cloud	Microsoft Azure Cloud	Google App Engine Cloud	IBM Bluemix Cloud
2013	$0.64	$0.70	$0.63	$0.61
2014	$0.48	$0.45	$0.49	$0.47
2015	$0.35	$0.39	$0.31	$0.30
2016	$0.28	$0.26	$0.29	$0.26
2017	$0.15	$0.16	$0.20	$0.16

The first phase of the algorithm analyses the highest loaded node and migrates the virtual machine to the available less

loaded node. After identifying the source and destination, the algorithm identifies the virtual machine to be migrated.

Calculate the load on each node in the data center

i)Virtual Machine Identification

$$Phy_{CPUCapacity} = \sum_{i=1}^{n} VM(i)_{CPUCapacity} \quad (1)$$

$$Phy_{MemoryCapacity} = \sum_{i=1}^{n} VM(i)_{MemoryCapacity} \quad (2)$$

$$Phy_{IOCapacity} = \sum_{i=1}^{n} VM(i)_{IOCapacity} \quad (3)$$

$$Phy_{NetworkCapacity} = \sum_{i=1}^{n} VM(i)_{NetworkCapacity} \quad (4)$$

$$\Pi = (Phy_{CPUCapacity} + Phy_{MemoryCapacity} + Phy_{IOCapacity} + Phy_{NetworkCapacity}) = \quad (5)$$

In the second step, the algorithm identifies the highest and lowest loaded node in the data center .

$$\Pi_{MAX} = \begin{cases} If\ \Pi_i > \Pi_j, then\ \Pi_{MAX} = \Pi_i \\ Else\ \Pi_j > \Pi_i, then\ \Pi_{MAX} = \Pi_j \end{cases} \quad (6)$$

$$\Pi_{MIN} = \begin{cases} If\ \Pi_i < \Pi_j, then\ \Pi_{MIN} = \Pi_i \\ Else\ \Pi_j < \Pi_i, then\ \Pi_{MIN} = \Pi_j \end{cases} \quad (7)$$

Once the source and destination is identified as MAX and MIN respectively.

$$VM(i) = VM(i)_{CPUCapacity} + VM(i)_{MemoryCapacity} + VM(i)_{IOCapacity} + VM(i)_{NetworkCapacity} \quad (8)$$

$$\Pi_{MAX} - VM(i) = \Delta_{Source} \quad (9)$$

$$\Pi_{MIN} + VM(i) = \Delta_{Destination} \quad (10)$$

After the calculation of the new load, the source and destination nodes must obtain the optimal load condition, where the loads are nearly equally balanced.

$$\begin{cases} If\ \Delta_{Source} \approx \Delta_{Destination}, Then\ Migrate\ VM(i) \\ Else\ i = \in (n) \end{cases} \quad (11)$$

Where n is total number of virtual machines in Source node.

ii)Virtual Machine Allocation

During the second phase of the algorithm, this work analyses the time requited for VM allocation for the selected virtual machine with other parameters like Energy consumption, Number of host shutdowns, Execution time - VM selection time, Execution time - host selection time and Execution time - VM reallocation time. These parameters will help in generating the cost function the identification of virtual machine to be migrated is carried out. During the identification, the optimal load balanced condition is identified.

Calculate the Energy consumption at the source before migration:

$$E_{Source} = \sum_{i=1}^{t} (E_{CPU} + E_{NETWORK} + E_{IO} + E_{MEMORY})_i \quad (12)$$

Calculate the Energy consumption at the destination after migration:

$$E_{Destination} = \sum_{i=1}^{t} (E_{CPU} + E_{NETWORK} + E_{IO} + E_{MEMORY})_i \quad (13)$$

Calculate the difference in Energy consumption during migration:

$$E_{Diff} = |E_{Source} - E_{Destination}| \quad (14)$$

Calculate the Number of host shutdowns, Execution time - VM selection time, Execution time - host selection time and Execution time - VM reallocation time during migration:

$$\begin{pmatrix} Host_{Down} & VM_{SelectionTime} \\ Host_{SelectionTime} & VM_{ReallocationTime} \end{pmatrix} \quad (15)$$

Henceforth the comparative analysis is been demonstrated in the results and discussion section.

iii)Cost Analysis of Migration

The optimality of the algorithm focuses on the SLA. During the final phase of the algorithm, the migrations is been

validated with the help of the cost function to measure the optimality of the cost. The final cost function is described here:

$$Cost(VM) = \mathrm{E}_{Diff} + \begin{pmatrix} Host_{Down} & VM_{SelectionTime} \\ Host_{SelectionTime} & VM_{\mathrm{Re}allocationTime} \end{pmatrix} + SLA_{Violation} \quad (16)$$

5. PERFORMANCE AND RESULTS

The results demonstrates the most effective and sustainable nature of the framework. However the due to the network congestion during the data transmission, it is been observed that the availability of the parameter values are seems not to be available for longer runs.

Table 4.SLA Violation Improvement

Policies	SLA Violation (In %)	Change (Decreased) Existing – Proposed	Change in %
IQR MC	1.13%	0.0015	13
IQRMMT	1.05%	0.0007	7
LR MC	3.17%	0.0219	69
LRMMT	3.16%	0.0218	69
LR MU	3.39%	0.0241	71
LR RS	3.17%	0.0219	69
LRR MC	3.16%	0.0218	69
LRRMMT	3.39%	0.0241	71
LRR MU	3.74%	0.0276	74
LRR RS	3.57%	0.0259	73
MAD MC	1.53%	0.0055	36
MAD MMT	1.31%	0.0033	25
MAD MU	1.53%	0.0055	36
MAD RS	1.56%	0.0058	37
THR MC	3.09%	0.0211	68
THRMMT	3.25%	0.0227	70
THR MU	2.73%	0.0175	64
THR RS	3.13%	0.0215	69
OPT ALGO	0.98%	-	-

this work analyses the percentage of SLA violation during the proposed method and compare with the existing policies.

Table 5. Comparison of Energy Consumption

Policies	**Energy (kWH)**	**Change (Increased) Proposed – Existing**	**Change in %**
IQR MC	46.86	2.46	5
IQRMMT	47.85	1.47	3
LR MC	44.35	4.97	11
LRMMT	45.37	3.95	9
LR MU	40.38	8.94	22
LR RS	40.35	8.97	22
LRR MC	40.37	8.95	22
LRRMMT	40.38	8.94	22
LRR MU	40.14	9.18	23
LRR RS	40.54	8.78	22
MAD MC	44.99	4.33	10
MAD MMT	45.61	3.71	8
MAD MU	47.36	1.96	4
MAD RS	44.71	4.61	10
THR MC	40.85	8.47	21
THRMMT	41.81	7.51	18
THR MU	44.08	5.24	12
THR RS	41.34	7.98	19
OPT ALGO	49.32	-	-

The proposed framework, demonstrates nearly 10% increase compared to the existing policies due to improvement in SLA.

the proposed technique is been tested for the load balancing with the below furnished simulation setup Finally, the proposed technique is been tested for the load balancing with the below furnished simulation setup

Table 6.Load Balancing Simulation Setup

Simulation Duration (In Secs)	**Requests per User**	**Data Size (Bytes)**	**Avg. Users**	**Virtual Machines**	**Memory**	**CPU**
216000	120	2000	2000	5	512	2.4 GhZ

The CPU utilization achieved during the simulation is furnished below [Table – 17] and 100% of the CPU utilization is been achieved during load balancing.

Table7.Load Balancing Simulation

Cloudlet ID	STATUS	Datacenter ID	VM ID	Start Time	Finish Time	Time	Utilization
1	SUCCESS	1	0	0	800	800	100%
2	SUCCESS	2	0	0	800	800	100%
3	SUCCESS	3	0	0	800	800	100%
9	SUCCESS	1	0	800	1601	801	100%
10	SUCCESS	2	0	800	1601	801	100%
11	SUCCESS	3	0	800	1601	801	100%
25	SUCCESS	1	0	1601	2402	801	100%
28	SUCCESS	2	0	1601	2402	801	100%
31	SUCCESS	3	0	1601	2402	801	100%
37	SUCCESS	1	0	2402	3203	801	100%
40	SUCCESS	2	0	2402	3203	801	100%
43	SUCCESS	3	0	2402	3203	801	100%
26	SUCCESS	1	3	2405	3208	803	100%

29	SUCCESS	2	3	2405	3208	803	100%
32	SUCCESS	3	3	2405	3208	803	100%
35	SUCCESS	1	3	2405	3208	803	100%
49	SUCCESS	2	0	3203	4004	801	100%
52	SUCCESS	3	0	3203	4004	801	100%
55	SUCCESS	1	0	3203	4004	801	100%
293	SUCCESS	2	3	20071	20874	803	100%
296	SUCCESS	3	3	20071	20874	803	100%

6. CONCLUSION

Load Balancing can be achieved through virtual machine migration. However the existing migration techniques constraints to improve the SLA and often compromise to a higher scale on the other performance evaluation factors. This work, demonstrates the optimal three phase virtual machine migration technique with up to 70% improvement to retain SLA compared to the other virtual machine migration technique. The work also elaborates on the virtual machine image operability most suitable for migration and determines the best format. The comparative analysis is been done with the proposed technique with the existing techniques like IQR MC, IQRMMT, LR MC, LRMMT, LR MU, LRR MC, LRRMMT, LRR MU, LRR RS, LR RS, MAD MC, MAD MMT, MAD MU, MAD RS, THR MC, THRMMT, THR MU and THR RS. The work also furnishes the practical evaluation results from the simulation to retain the improvement of the other parameters at least to the mean of other techniques during SLA improvement.

REFERENCES

[1] Anurudh Kumar Upadhyay et al, "Updating of Inter-quartile range Virtual Machine Allocation policy in cloud computing", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 8 (2) , 2017, 295-297

[2] Zhou Zhou,1 Zhigang Hu,1 and Keqin Li2 "Virtual Machine Placement Algorithm for Both Energy-Awareness and SLA Violation Reduction in Cloud Data Centers," Hindawi Publishing Corporation Scientific Programming Volume 2016, Article ID 5612039, 11 pages.

[3] J.K. Verma, C.P. Katti, P.C. Saxena, "MADLVF: An Energy Efficient Resource Utilization Approach for Cloud Computing "I.J. Information Technology and Computer Science, 2014, 07, 56-64.

[4] Mohammed Rashid Chowdhury, Mohammad Raihan Mahmud and Rashedur M. Rahman*, "Implementation and performance analysis of various VM placement strategies in CloudSim Chowdhury et al. Journal of Cloud Computing: Advances, Systems and Applications (2015) 4:20

[5] Guruh fajar shidik,azari,khabib mutofa "Evaluation of selection policy with various virtual machine instances in dynamic VM consolidation for energy efficient at cloud data center." Journal of networks vol 10 no 07 july 2015.

[6] Alireza Najari1, Ahvaz, Iran "Optimization of Dynamic Virtual Machine Consolidation in Cloud Computing Data Centers," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 9, 2016.

[7] Anton Beloglazov∗ and Rajkumar Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE, Concurrency Computat.: Pract. Exper. 2011; 00:1–24

[8] Guangjie Han 1,*, Wenhui Que 1 , Gangyong Jia 2 and Lei Shu 3, " An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing " , Sensors 2016, 16, 246; doi:10.3390/s16020246

[9] Shalini Soni, Vimal Tiwari " Energy Efficient Live Virtual Machine Provisioning at Cloud Data Centers - A Comparative Study " , International Journal of Computer Applications (0975 – 8887) Volume 125 – No.13, September 2015

[10] Perla Ravi Theja1 , S. K. Khadar Babu2, " Evolutionary Computing Based on QoS Oriented Energy Efficient VM Consolidation Scheme for Large Scale Cloud Data Centers " , CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume 16, No 2 Sofia • 2016

[11] H. Khazaei, J. Mi˘sic, V. B. Mi ´ ˘sic, and S. Rashwand, "Analysis of ´ a pool managementscheme for cloud computing centers," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5, pp. 849–861, 2013

[12] Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency Computat. Pract Exper 24:1397–1420.doi:10.1002/cpe.1867.