Computer Systems Science & Engineering
DOI:10.32604/csse.2022.022122
Article

Tech Science Press

An Efficient On-Demand Virtual Machine Migration in Cloud Using Common Deployment Model

C. Saravanakumar[1,*], R. Priscilla[1], B. Prabha[2], A. Kavitha[3], M. Prakash[4] and C. Arun[5]

[1]Department of Information Technology, St. Joseph's Institute of Technology, Chennai, 600119, India
[2]Department of Information Technology, Loyola-ICAM College of Engineering and Technology, Chennai, 600034, India
[3]Department of Electronics and Communication Engineering, M. Kumarasamy College of Engineering, Karur, 639113, India
[4]Data Science and Analytics Center, Karpagam College of Engineering, Coimbatore, 641032, India
[5]Department of Electronics and Communication Engineering, R.M.K College of Engineering and Technology, Chennai, 601206, India
*Corresponding Author: C. Saravanakumar. Email: saravanakumarc@stjosephstechnology.ac.in

Received: 28 July 2021; Accepted: 30 August 2021

Abstract: Cloud Computing provides various services to the customer in a flexible and reliable manner. Virtual Machines (VM) are created from physical resources of the data center for handling huge number of requests as a task. These tasks are executed in the VM at the data center which needs excess hosts for satisfying the customer request. The VM migration solves this problem by migrating the VM from one host to another host and makes the resources available at any time. This process is carried out based on various algorithms which follow a predefined capacity of source VM leads to the capacity issue at the destination VM. The proposed VM migration technique performs the migration process based on the request of the requesting host machine. This technique can perform in three ways namely single VM migration, Multiple VM migration and Cluster VM migration. Common Deployment Manager (CDM) is used to support through negotiation that happens across the source host and destination host for providing the high quality service to their customer. The VM migration requests are handled with an exposure of the source host capabilities. The proposed analysis also uses the retired instructions with execution by the hypervisor to achieve high reliability. The objective of the proposed technique is to perform a VM migration process based on the prior knowledge of the resource availability in the target VM.

Keywords: Cloud computing; virtualization; hypervisor; VM migration; virtual machine

1 Introduction

Cloud computing is a service oriented architecture for providing all types of resources to the customer through virtualization. The physical resources are mapped into virtual resources by using the Virtual Machine Manager. It acts as an intermediate layer between bare metal resources and VM for handling the customer requesting tasks. VM migration is a process which moves the VM of one machine to another machine without any disconnection of actual resources such as storage, networks and memory. The VM migration is classified into two types namely live VM migration and Cold VM migration. Live migration process is used to move the running VM from one server to another server. VM states are not affected during the migration and also achieve zero downtime of VMs. The cold migration terminates all running VMs and then the VMs are migrated to another host with downtime. The states of suspended VMs are not retained during the migration process. VM migration solves the problem occurring in the faulty server due to unscheduled downtime. It is mainly considered during the disaster management process. Two kinds of VM migration are carried out in cloud computing; they are quick migration and live migration. Quick migration takes less time when compared to live migration. VM migration related to storage achieves the process of load balancing, upgrade of resources and its maintenance. Traditional method of migration uses maximum storage and workload. This problem is addressed by using SnapMig model and also achieves high performance with snapshot scheme [1]. The entire VM is migrated to other data centers for performing load balancing and maintenance through Host based virtualization method. The bandwidth which is consumed by the live migration of a VM is maximum, so it will be reduced by using an intelligent based network model [2]. The service quality is decreased due to the cloud requests being generated by the customer. The large number of VMs are managed for maintaining the high quality services through VM migration. Migration models are categorized into pre and post copy migration which are related to the performance parameters like usage of CPU, Network and memory etc. The performance is also improved by integrating various migration techniques with generic models [3]. Live migration is mainly used for performing the process like resource accessibility and maintenance, fault tolerance, resource flexibility and adjustment [4]. The VMs are moved from one physical machine to another without shut downing and isolating each other. The performance of VM migration is assessed based on the time taken by the migration, service availability and so on. The migration time is minimized by transferring the Least Recently Used (LRU) pages without resending the same page again to the data center [5]. The traditional VM migration techniques are restricted to predefined workload so, it is not suitable for heterogeneous workload. An agile based VM migration is used to maximize the performance without any consideration of the workload [6]. An overloaded problem are minimized by migrating the VM to another data center by considering the operations like load balancing of host and server level consolidation. The VMs are selected using the technique with minimized migration count and migration time without any SLA violations [7]. The Data center resource utilization is improved by optimizing the maintenance and operation cost of the cloud model. It also reduces the energy usage of the VM during migration with the proper scheduling methods and policies [8,9]. The VM migration process makes the VM in a running state until all data are transmitted to the corresponding data center. The data transmission occurs either online or offline. In an offline method the VM is in down state which leads to the dirty pages retransmission. This problem is overcome by using the online based deterministic algorithm in order to improve the performance [10]. A static VM migration fails to deploy the real time workload because of pre-allocated resources, so it will be addressed by implementing the dynamic migration with different workload [11,12]. A transient computing resumes the cloud server during the idle condition are not suitable for current requirements. This model uses preemptive property for reducing the VM pressure by reclaiming the resources based on the policies [13]. The integration of software base at the cloud platform is not suitable for functional mapping but to scale the customer base. The selection of proper platform integration supports high reliability on the cloud [14]. An orchestrator framework requires to allocate an adequate number of physical resources which are mapped to VM for handling the workload. It performs specific goals related to the cloud service with predetermined rules [15]. The fuzzy based model achieves highest trust on cloud resources because the resources are mapped to the correct VM for handling the customer requests without any latency [16]. The containers are deployed in the clouds which run on various VMs in order to manage workloads. An existing container based models suffer resource deadlines, which is tracked by setting user defined deadlines and fitness level using Genetic Algorithm [17].

The main objective of the proposed VM migration handles the requested VMs based on the current demand by achieving high availability. It uses a five step process for migrating the required VM to another host on the cloud based on the demand, namely VM request, selection of VM, Exposure of the VM to the host, Availability status and migration of VM.

The rest of the paper is organized as follows; Section 2 presents the related work of the proposed methodology. Section 3 represents the on demand VM migration process. Section 4 provides the significance of the dynamic VM migration. Section 5 formulates the mathematical analysis of the VM migration. The result and discussion is specified in Section 6 and finally Section 7 focuses on the conclusion and future work.

2 Related Work

Cloud computing uses the common deployment manager for interoperability among various host machines during VM migration. The migration processes perform various types' namely single VM migration, Multiple VM migration and clustered VM migration. The existing VM migration algorithms suffer dynamic workload management across the host which leads to performance degradation problems. Physical Machines are mapped to the VMs for satisfying the needs with well design structures related to capacity because the excess capacity leads worst resource utilization. The proposed PM Mapping process selects enough number of VM based demand which protect from resource underutilization. The proposed technique follows the migration based on the demand of the requesting host. A single VM migration process migrates to another host whereas multiple VMs migrate to the heterogeneous VM for better resource utilization. The cluster VMs are migrated to similar types of VM which present in other hosts for handling the complex workloads. The CDM [18] achieves dynamic on demand live migration with high availability of resources with scalability. The hypervisor uses the VM cluster layer for grouping the VMs based on the request from the host. Figs. 1 and 2 present the single and multi VM migration respectively. Fig. 3 shows the clustered VM migration process of the host machine. The proposed technique uses the hardware abstraction level virtualization because the VM and PM mapping process are carried out in the cloud infrastructure layer.

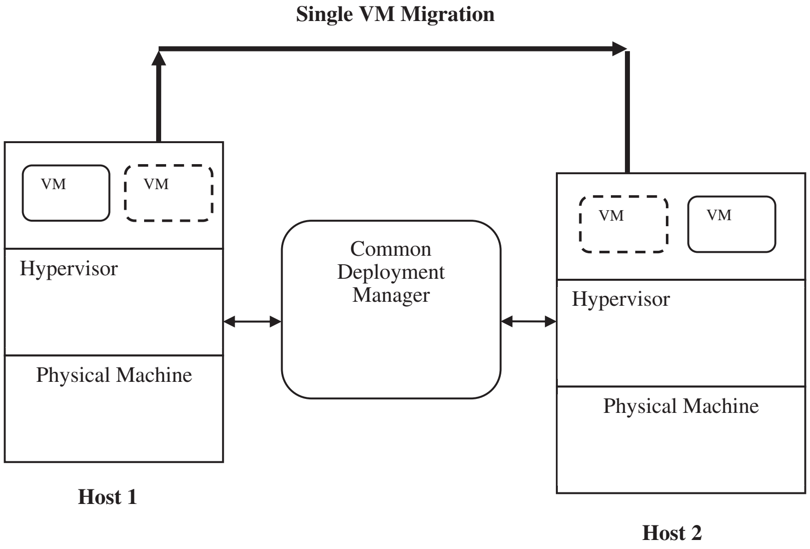

Figure 1: Single VM migration

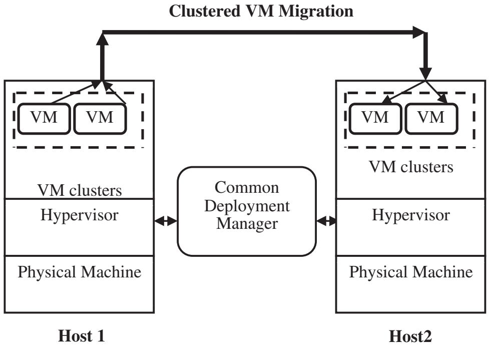

Figure 2: Multi VM migration

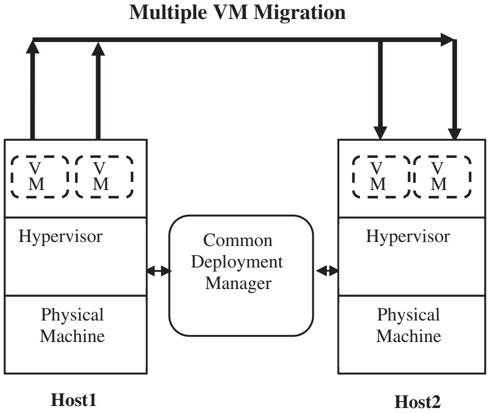

Figure 3: Clustered VM migration

3 Proposed on Demand VM Migration

VM migration is the process of moving a VM from one host machine to another host machine for handling the workload of the cloud. The proposed migration technique uses the dynamic way of managing VM migration among the host. Various steps are used for performing migration processes namely VM migration request, VM selection, VM exploration, VM binding and VM migration. VMs are selected based on states such as active, inactive, idle, terminate, migrate and initializing states and so on. The selection process is mainly based on the idle state of the VM and it checks the capacity of the available VM. If the capacity of the VM is matched with a request, then that VM is selected for migration. The requesting host requests the VM to the common deployment manager by generating the VM request with required parameters like current workload, required workload and so on. CDM selects the suitable host and VM from the data center. The VM is exposed to the CDM by the selected host then the availability of the VM and its properties are intimated to the requested host. The VM is migrated to the host for further workload execution. The VMs are migrated in various levels such as single VM level migration, Multiple VM level migration and Clustered VM level migration. A service level agreement is established among the hosts in the data center for sharing the resources with reliability. The objective of the proposed technique is used to migrate the VM, based on the current requirement of the host with dynamic workload in order to achieve high availability. The proposed VM selection and migration is shown in Fig. 4.

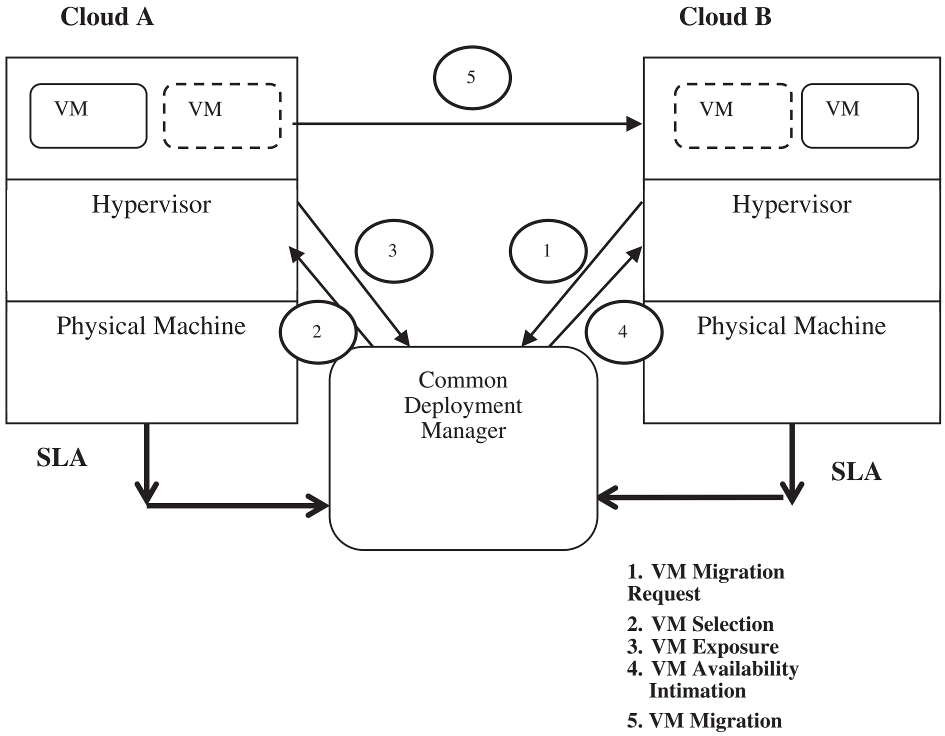

Figure 4: Proposed VM migration

4 Significance of Dynamic VM Migration

Data centers have many physical machines which host various VMs, based on the user request with the corresponding resources. The VMs suffer a workload variation due to heterogeneous customer requests. It leads to the problem of changing the required resources for task execution. This problem is overcome by increasing the workload with the selection of idle VM in the same data center. If the physical machines have a capacity to execute the workload, then the current demand of the customer's task is not handled it leads to suffer the migration problem. VM migration is the only solution to address these issues and also to minimize the SLA violations which occurs during workload management. Normally the VM migration is done in one data center which gives effective performance. If the VM migration is in a different location, then it will be managed by using an efficient monitoring and management method. Migration will happen in various types depending upon the requirement namely memory level, network level, file system level and device level migration. Memory level migration is done per-copy based as well as demand based. File migration never copy the file; until it changes the location from local to remote location. Network migration achieves the migration to the new MAC and port address. Device migration process is performed in two levels such as virtualization level and emulation level. The significance of VM migration is related to the factors like data center infrastructure expansion, disaster management, load balancing and so on. The proposed dynamic VM migration improves the migration performance by considering on demand interaction among the hosts in the data center. Tab. 1 represents the capability of the VM during migration.

Table 1: Mapping value of VM migration

Mapping value	Migration capability
0	Original precopy
1	CPU throttling
2	Delta compression
3	Data compression
4	post copy

Algorithm1 gives the steps involved in VM migration request generated by requested host machine.

Algorithm 1: VM Migration Request Algorithm

```
Input: Number of VM, VM Size, Task Size, Requester Cloud ID
Output: VM Status Intimation
Let Number of VM as |VM|
Let Number of Task as | Task|
For allVM$_i$ Ɛ VM and Task$_i$ Ɛ Task do
    If |Task| <= |VM| then
    Task$_i$→ VM$_i$
    MapTask++;
    Else If |Task| > |VM| then
    Requested VM ← |Task$_i$| > |MapTask$_i$|
    End If
End For
Calculate the VM size | Requested VM |
Return Requester VM, Cloud ID
```

VM explorations with the required steps are specified in algorithm 2.

Algorithm 2: VM Exploration Algorithm

```
Input: Number of VM, VM Size, Task Size
Output: VM Availability with IS
For allVM$_i$ Ɛ VM and Task$_i$ Ɛ Task do
L1: If( |Task$_i$| == |VM$_i$|) then
Task$_i$→ VM$_i$
Maptask++;
Else If( |Task$_i$| > |VM$_i$|) then
```

Algorithm 2 (continued).

```
Task$_i$→ VM$_i$
MapTask++;
End if
If (|MapTask$_i$|<=|Current VM$_i$| )then
Goto L1;
Else
    Available VM ← |VM| - |Current VM|
End if
End for
Explore the VM avaiablity to the CMD with Cloud ID;
Return Available VM with Status
```

The on-demand VM migration process steps are represented in the algorithm 3 with required parameters related to the performance.

Algorithm 3: On Demand VM Migration

```
Input: Available VM, Requested VM$_{size}$, Requested Cloud ID
Output: VM Migration Response
For allVM$_i$ Ɛ VM do
    Collect the availability VM from CDM;
    Select the idle VM from VM List;
    Response VM$_i$ ←Idle VM$_i$;
End for
    Intimate the Response VM Size to the Requested Cloud;
    Migrate the VM to the Requested Cloud with ID;
Return VM Migration Response;
```

5 Mathematical Formulation of VM Migration

The initial configuration of the proposed dynamic model is represented in the Eqs. (1)–(6). It gives the migration rate and time of cloud A and cloud B with CDM level. The initial configuration of the proposed dynamic model is represented in the Eqs. (1)–(6). It gives the migration rate and time of cloud A and cloud B with CDM level.

$$Migration\ Rate\ of\ Cloud\ (CloudA) = \mu_{VM_A} \tag{1}$$

$$Migration\ Time\ of\ Cloud\ A\frac{1}{\mu_{VM_A}} \tag{2}$$

$$Migration\ Rate\ of\ Cloud\ B = \mu_{VM_B} \tag{3}$$

$$Migration\ Time\ of\ Cloud\ B = \frac{1}{\mu_{VM_B}} \tag{4}$$

$$Migration\ Ratein\ the\ CDM = \mu_{VM_{CDM}} \tag{5}$$

$$Migration\ Time\ of\ CDM = \frac{1}{\mu_{VM_{CDM}}} \tag{6}$$

The interaction of the cloud B to the CDM for requesting the VM is shown in Eq. (7).

$$Interaction_{(CloudB,CDM)} = \left[\frac{\delta_{VM_B}}{\mu_{VM_B}}\right]^{CloudB} \times \left[1-\frac{\delta_{VM_B}}{\mu_{VM_B}}\right] \times \left[\frac{\delta_{VM_{CDM}}}{\mu_{VM_{CDM}}}\right]^{CDM} \times \left[1-\frac{\delta_{VM_{CDM}}}{\mu_{VM_{CDM}}}\right] \tag{7}$$

Cloud A and CDM interaction for selecting the suitable VM is represented in Eq. (8)

$$Interaction_{(CloudA,CDM)} = \left[\frac{\delta_{VM_A}}{\mu_{VM_A}}\right]^{Cloud_A} \times \left[1-\frac{\delta_{VM_A}}{\mu_{VM_A}}\right] \times \left[\frac{\delta_{VM_{CDM}}}{\mu_{VM_{CDM}}}\right]^{CDM} l \times \left[1-\frac{\delta_{VM_{CDM}}}{\mu_{VM_{CDM}}}\right] \tag{8}$$

Cloud A and Cloud B interaction during the VM migration process is calculated in Eq. (9).

$$Interaction_{(CloudA,CloudB)} = \left[\frac{\delta_{VM_A}}{\mu_{VM_A}}\right]^{Cloud_A} \times \left[1-\frac{\delta_{VM_A}}{\mu_{VM_A}}\right] \times \left[\frac{\delta_{VM_B}}{\mu_{VM_B}}\right]^{CloudB} \times \left[1-\frac{\delta_{VM_B}}{\mu_{VM_B}}\right] \tag{9}$$

The VM rate of the various levels of the VM migration process is presented in the Eqs. (10)–(12).

$$VM\ Ratein\ Cloud\ A = \left[\frac{\delta_{Total}}{\mu_{Total}-\delta_{Total}}\right] - \left[\frac{\delta_{VM_A}}{\mu_{VM_A}-\delta_{VM_A}}\right] \tag{10}$$

$$VM\ Ratein\ Cloud\ B = \left[\frac{\delta_{Total}}{\mu_{Total}-\delta_{Total}}\right] - \left[\frac{\delta_{VM_B}}{\mu_{VM_B}-\delta_{VM_B}}\right] \tag{11}$$

$$VM\ Rateduring\ Migration = \left[\frac{\delta_{VM_A}}{\mu_{VM_A}-\delta_{VM_A}}\right] - \left[\frac{\delta_{VM_B}}{\mu_{VM_B}-\delta_{VM_B}}\right] \tag{12}$$

The waiting of the cloud A and cloud B is calculated in the Eqs. (13) and (14)

$$VM\ Waiting\ Time\ in\ Cloud\ A = \left[\frac{1}{\mu_{Total}-\delta_{Total}}\right] - \left[\frac{1}{\mu_{VM_A}-\delta_{VM_A}}\right] \tag{13}$$

$$VM\ Waiting\ Timein\ Cloud\ A = \left[\frac{1}{\mu_{Total}-\delta_{Total}}\right] - \left[\frac{1}{\mu_{VM_B}-\delta_{VM_B}}\right] \tag{14}$$

Total migration time of the cloud A and cloud B is represented in the Eqs. (15) and (16) respectively.

$$Total\ Migration\ Waiting\ Time = |VM\ Waiting\ Timein\ Cloud\ A| + |VM\ Waiting\ Timein\ Cloud\ B| \tag{15}$$

$$Total\ Number\ of\ VM\ in\ Migration = |VM_{Cloud_A}| + |VM_{Cloud_B}| + |VM_{CDM}| \tag{16}$$

The VM interaction status and migration status are specified in Eqs.(17) and (18)

$$VM\ Interaction\ Status = \left\{ \begin{array}{ll} Interaction_{CloudB,CDM}, & Migration\ Request == True \\ Interaction_{CloudA,CDM}, & VMExposure == True \\ Interaction_{CDM,CloudB}, & VMIntimation == True \\ NoInteraction, & Otherwise \end{array} \right\} \tag{17}$$

$$VM\ Migration\ Status = \left\{ \begin{array}{ll} |VM_{Size_A}| \geq |VM_{Size_B}|, & Migration\ Success \\ |VM_{Size_A}| < |VM_{Size_B}|, & Migration\ Redirection \\ NoInteraction, & Otherwise \end{array} \right\} \tag{18}$$

6 Result and Discussions

The live migration process keeps the VM in running states and the data transmission in an offline manner which leads to high transmission time. An offline transmission suffers a large amount of dirty pages in the destination VM, so it can be handled by using the RO (Random Online) algorithm. This algorithm performs the migration by introducing an integrated version of migration time and downtime minimization. VM migration takes larger time because of the network bandwidth factors such as transmission of stored data, configuration file of the system and traffic of the network etc. it is reduced by eliminating the duplicate data in both source VM and target VM. This problem is overcome by using HM (Hash Map) algorithm with proper maintenance of SLA (Service Level Agreement) [19]. VM migration with energy is optimized in the data center and improves the performance in the network using the greedy approach. It targets the parameters such as link usage level, RTT (Round-Trip Time) and data loss [20]. The data center efficiency is measured by considering the parameters like VM provisioning technique and location of target VM. Energy consumption is minimized by migrating a limited number VMs at the particular time. An energy Aware (EA) method has been implemented for reducing the VM migration time [21]. A federated based VM migration methods support the other provider to meet the service quality to the customer by sharing the resources. It performs the VM migration process by achieving the reduced migration cost with load balancing capability [22,23]. The existing VM migration methods are based on the specific workload and VM selection for migration is done in advance with a fixed number of VMs [24]. This problem is solved by considering on-demand dynamic heterogeneous workloads with exact number of VMs that are migrated to the destination host. The proposed analysis compares the performance of the VM migration in three different levels namely single, multiple and clustered VMs with different workload conditions. The parameter considered for the analysis based on the retired instructions which is used for the migration process. Normally the VMs are de-scheduled during the migration which leads the guest instructions to be used by the hypervisor. Physical machine retires the native instructions in order to protect the sensitive instructions which are executed by the un-trusted VMs with high performance. The retired instructions leave the VM after producing the result with high visibility in the VCPU architectural conditions. The proposed method is considered as the retired instruction for performing the dynamic VM migration process.

The proposed algorithm is compared with different parameters for analyzing the reliability of VM migration is represented in Tab. 2. A single level VM migration process performance parameters are shown in Fig. 5. The Multi level and clustered level VM performance analysis is shown in Figs. 6 and 7 respectively.

Table 2: Reliability analysis with various

Algorithm	Parameter					
	Selected VMs count	Migrated VM count	Avg SLA violation	Over loaded destination host count	VM migration success ratio (%)	Reliability (%)
Random online algorithm	20	14	10.17	5	70	68
Hash map algorithm	18	15	8.90	3	83.33	74
Energy aware algorithm	16	11	10.25	4	68.75	64
Federated based VM migration algorithm	25	20	9.60	2	80	72
Proposed migration algorithm	12	12	7.26	0	98	100

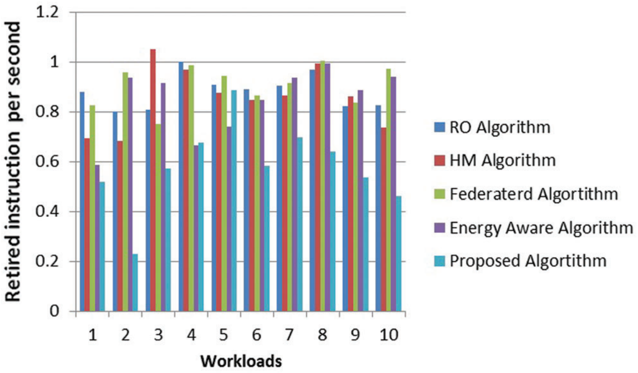

Figure 5: Single VM performance analysis

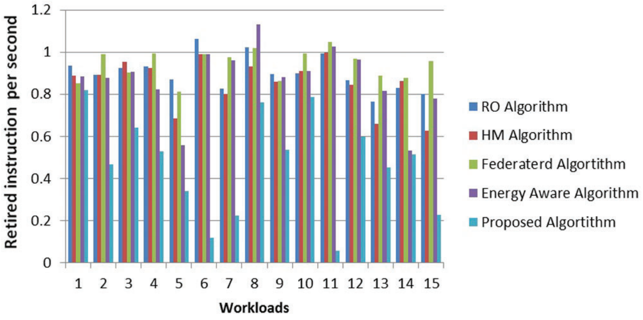

Figure 6: Multi VM performance analysis

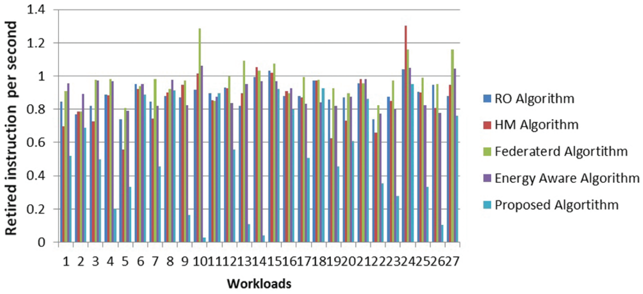

Figure 7: Cluster VM performance analysis

7 Conclusion

The distributed computing domain uses the cloud computing concepts for providing the high quality services to the customer. These services are classified into software based service, platform based service and infrastructure based services and so on. The VM migration process plays a vital role during the handling of request process. The existing VM migration process has a problem of identifying the capabilities of the source host with their capacity. The proposed VM migration provides support to the requested hosts with the information about the source host and current VM availability. The process has been used in three categories based on the size of the request such as single, multiple and clustered VM. The requested host uses the common deployment manager for establishing proper communication between source and destination host. Various algorithms are compared with the retired instructions that are used by the hypervisor. The proposed VM migration process minimizes the retired instructions in order to achieve high reliability. In future, this VM migration process can be extended to the federated cloud with VM classification process with high availability.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Yaodong, B. Mao, J. Hong, Y. Yuekun, L. Hao *et al.,* "Accelerating VM live storage migration by leveraging the existing VM snapshots in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1416–1427, 2018.

[2] Y. Yaodong, B. Mao, J. Hong, Y. Yuekun, L. Hao *et al.,* "Snapmig: Accelerating VM live storage migration by leveraging the existing VM snapshots in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1416 –1416 1427, 2018.

[3] K. Vincent, M. Éric and H. Fabien, "Scheduling live migration of virtual machines," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 282–296, 2020.

[4] W. Huandong, L. Yong, Z. Ying and J. Depeng, "Virtual machine migration planning in software-defined networks," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 1168–1182, 2019.

[5] B. Debabrota, W. Xiayang, H. Yang, C. Haibo and B. Stéphane, "Learn-as-you-go with megh: Efficient live migration of virtual machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 8, pp. 1786 –1786 1801, 2019.

[6] Y. Simin, L. Yuqing, W. Chaoying and H. Tao, "An agile framework adaptive to complicated memory workloads for VM migration," in *Int. Conf. on Computer Science and Network Technology (ICCSNT)*, Dalian, China, IEEE, pp. 42–46, 2017.

[7] W. Quanwang, I. Fuyuki, Z. Qingsheng and X. Yunni, “Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters,” *IEEE Transactions on Services Computing*, vol. 12, no. 4, pp. 550 –550 563, 2019.

[8] B. Salah-Eddine, Q. Alejandro and B. Martine, “Live placement of interdependent virtual machines to optimize cloud service profits and penalties on slas,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 237–249, 2019.

[9] M. Prakash and C. Saravanakumar, “An authentication technique for accessing de-duplicated data from private cloud using one time password,” *International Journal of Information Security and Privacy*, vol. 11, no. 2, pp. 1–10, 2017.

[10] K. Balasaravanan and C. Saravanakumar, “An enhanced security measure for multimedia images using hadoop cluster,” *International Journal of Operations Research and Information Systems*, vol. 12, no. 3, pp. 1–7, 2021.

[11] A. O. Fathy Atya, Q. Zhiyun, V. K. Srikanth and P. Thomas La, “Catch me if you can: A closer look at malicious co-residency on the cloud,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 560 –560 576, 2019.

[12] Y. Chao, H. Hongchao, L. Wenyan and W. Yawen, “An effective and scalable VM migration strategy to mitigate cross-VM side-channel attacks in cloud,” *China Communications IEEE*, vol. 16, no. 4, pp. 151 – 151 171, 2019.

[13] X. Chao, L. Chuang and H. Jie, “Scalability analysis of request scheduling in cloud computing,” *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 249–261, 2019.

[14] Z. Fan, F. Majd, H. Kai and U. K. Samee, “Empirical discovery of power-law distribution in mapreduce scalability,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 744–755, 2019.

[15] L. Keqin, “Quantitative modeling and analytical calculation of elasticity in cloud computing,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1135–1148, 2020.

[16] M. Reza Rahimi, V. Nalini, M. Sharad and V. Athanasios, “On optimal and fair service allocation in mobile cloud computing,” *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 815 –815 828, 2018.

[17] A. Thamer and S. I. Leandro, “Management of container-based genetic algorithm workloads over cloud infrastructure,” in *Proc. of the 17th ACM Int. Conf. on Computing Frontiers*, Sicily, Italy, pp. 229–232, 2020.

[18] C. Saravanakumar and C. Arun, “Efficient idle virtual machine management for heterogeneous cloud using common deployment model,” *KSII Transactions on Internet and Information Systems*, vol. 10, no. 4, pp. 1501–1518, 2016.

[19] A. Rohit and S. K. Mohanty, “A scalable attribute-based access control scheme with flexible delegation cum sharing of access privileges for cloud storage,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 32–44, 2020.

[20] S. Stelios, B. Nik, A. Cristiana and B. Rajkumar, “Elastic load balancing for dynamic virtual machine reconfiguration based on vertical and horizontal scaling,” *IEEE Transactions on Services Computing*, vol. 12, no. 2, pp. 319–334, 2019.

[21] T. Cheikhou and T. Fatoumata, “An energy-efficient VM migrations optimization in cloud data centers,” in *Proc. of IEEE AFRICON*, Accra, Ghana, pp. 1–8, 2019.

[22] H. Inkwon and P. Massoud, “Hierarchical, portfolio theory-based virtual machine consolidation in a compute cloud,” *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 63–77, 2018.

[23] B. Azzedine, G. Shichao and R. E. De Grande, “A task-centric mobile cloud-based system to enable energy-aware efficient offloading,” *IEEE Transactions on Sustainable Computing*, vol. 3, no. 4, pp. 248–261, 2018.

[24] C. Saravanakumar, M. Geetha, S. Manoj Kumar, S. Manikandan, C. Arun *et al.,* “An efficient technique for virtual machine clustering and communications using task-based scheduling in cloud computing,” *Scientific Programming*, vol. 2021, no. 5586521, pp. 1–15, 2021.