VM image structure and its implications on Migration Strategies in cloud computing

Dr. Amit Kumar Chaturvedi[#1], Pankaj Nandan[*2]

*Engineering College, Ajmer[#1], Govt. Degree College, Basohli, J&K[*2]*

amit0581@gmail.com[#]

Abstract: *Virtualization is the technology that divides physical components on server in several isolated execution environments by deploying a layer VMM (Virtual machine manager) on top of hardware resources or OS. Each Execution environment i.e. VM independently runs with OS or application without any mutual interruption. In recent years, increasing number of users choose cloud data centre to hold their applications. Therefore, it is big issue to manage VMs on data center. For example, some server over utilized, while others underutilized. Some VM that fails to run it will affect other VMs as well. These all problems are solved by a critical technology called VM Migration. This technology inherits from Process Migration but Process Migration suffers from residential dependency problem and prevents to be used in cloud environment. In this paper, we have tried to explore the issues for live migration of VMs and parameters for performance measurement.*

Keywords: **Live migration, virtual machine, cloud server, virtualization, migration strategies, VM image.**

1. INTRODUCTION TO VIRTUALIZATION

The application of cloud computing for managing the web contents access and management is increasing day by day. The load on the cloud servers is also increasing proportionally. Virtual machines (VM) are akin to the computer system for users. The users are now more online than ever before and hence migrations of these VMs are required without stopping their work or processes. There are the various reasons for VMs migration online or offline. The online or live migration is the major requirement for the cloud servers, due to much more load of the cloud servers with live applications. Cloud providers offer services to users by using virtualization technologies and using large datacenters that have huge amounts of computing, storage and bandwidth resources. Service providers should have the ability to provision requested virtual resources across their distributed infrastructure (i.e., multiple datacenters) to achieve multiple goals including revenue maximization, operational costs reduction, energy efficiency, and green IT, or to simply satisfy geographic location constraints of users.

In the recent years the datacenters are scaled up more and for saving the energy involved in migration of VMs and due to their binding with the Service Level Agreements (SLAs), the researchers are more focused on developing innovative migration technologies, which take care of the migration issues and SLAs. The migration of VMs should also guarantee the Quality of Service (QoS) to ensure SLAs and minimize the cost involved like operations expenditure and capital expenditure.

Virtualization, which acts as the underlying technology for cloud computing, enables large amounts of third-party applications to be packed into virtual machines (VMs). VM migration enables servers to be reconsolidated or reshuffled to reduce the operational costs of data centers. The network traffic costs for VM migration currently attract limited attention. However, traffic and bandwidth demands among VMs in a data center account for considerable total traffic. VM migration also causes additional data transfer overhead, which would also increase the network cost of the data center.

2. MIGRATION STRATEGIES

VM Migration can be broadly categorized in two type: (1) Offline/Cold/Non Live Migration (2) Online/Hot/Live Migration. Offline Migration mechanism first stop the

currently running VMs and migrate VMs CPU and memory state to destination and finally restarts VMs at new destinations. While in Online Migration Mechanism keep the services running without any interruption during most of migration process.

There are a variety of reasons for migrating Virtual Machines (VMs) from one server to another. These VMs migration can help service providers to achieve the goals of saving energy, enhancing the resource efficiency and quality of service (QoS). There are two possible ways to migrate these VMs either offline or online (i.e. live). The online or live migration of virtual machine may be done either by using the post-copy migration strategy, or pre-copy migration strategy. In order to ensure the QoS and for performance consideration of the migration strategies, the parameters are the migration time, the downtime of VM, energy consumed during migration, and the bandwidth. These parameters should be considered while implementing the VM migration.

Live VM migration is required to full fill the running application resource demand. It facilitates the following features: Load Balancing, Proactive fault tolerance, Power management, Resource sharing, online system maintenance.

Live VM Migration Techniques:

It is important to know that what is to be migrated and how that migration process handles those contents.

1. **CPU State**: CPU state is little bit information and represents the lower bound service downtime.

2. **Memory Content**: This is larger part of information which represents the running process memory guest OS memory.VM is configured larger amount of memory but it is not fully utilized by VM. So, no need to transfer unused memory. Compression technique is used to speed up migration process. Following are Memory modules that need to be transferred. (1)VM Configured Memory (2) Hypervisor Allocated Memory (3)VM used Memory (4).Application Requested memory (5) Application Actively Dirtied memory.

3. **Storage Content**: It is optional part of Live VM Migration. Either new virtual disk storage is created on destination server and synchronise with source server or access the source disk storage through NAS. Different type storage content that has be transferred are followings: (1) Virtual Disk size (2). VM Used Blocks (3). Hypervisor Allocated Blocks.

Pre-copy technique: As shown below in figure 1 pre-copy technique have four important phase as Follows (1) Pre-Migration and Reservation stage (2) Iterative Dirty Page pre copying stage (3) Stop and Copy Stage (4) Post Migration Stage or Activation stage. In First phase this technique checks and reserve required resources at destination server and transfer all pages to destination while VM running at source server, In Second phase all dirty pages are send iteratively up to the predefined threshold. Memory that are continuously Modified/updated by end user are called Dirty page memory. These dirty page need to be resend in future iterations. In the third stage Migrating VM is suspended at source server and copy current CPU state and remaining dirty page to destination server. In the last phase when source server receives commitment from destination server, hypervisor at resume VM at destination server and process completed. Most of known hypervisor KVM, Xen, VMW are the uses this technique.

Post-copy technique: As shown in figure 2 Post-copy technique use following phase to migrate the VMs (1) Pre-Migration and Reservation stage (2) Stop and Copy Stage (3) Post Migration Stage or Activation stage (4) On Demand VM memory page transfer.

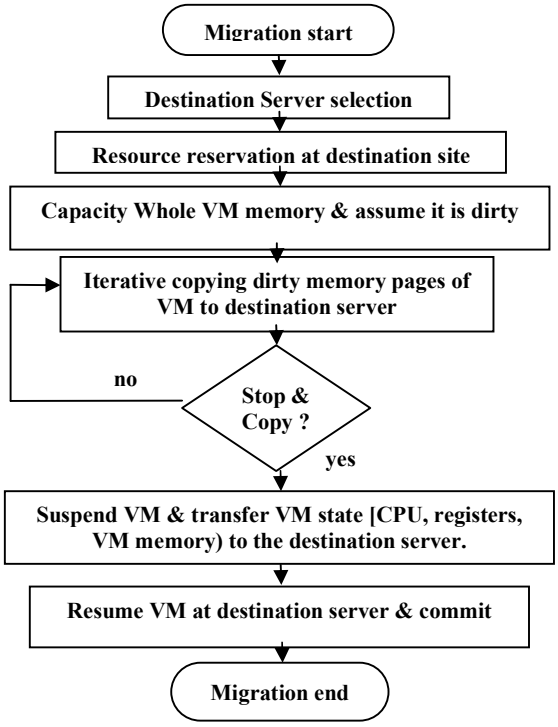


Figure 1: Pre-copy technique

First phase is same as pre-copy technique while in second phase source VM suspended and copy CPU and registers state to the destination server. In third phase, VM resumes at destination server. In last phase, remaining Memory page transferred on demand. There are various techniques to optimize on demand paging transfer like Demand paging, Active Push, Pre-Paging, and Dynamic self-Ballooning (DSB).

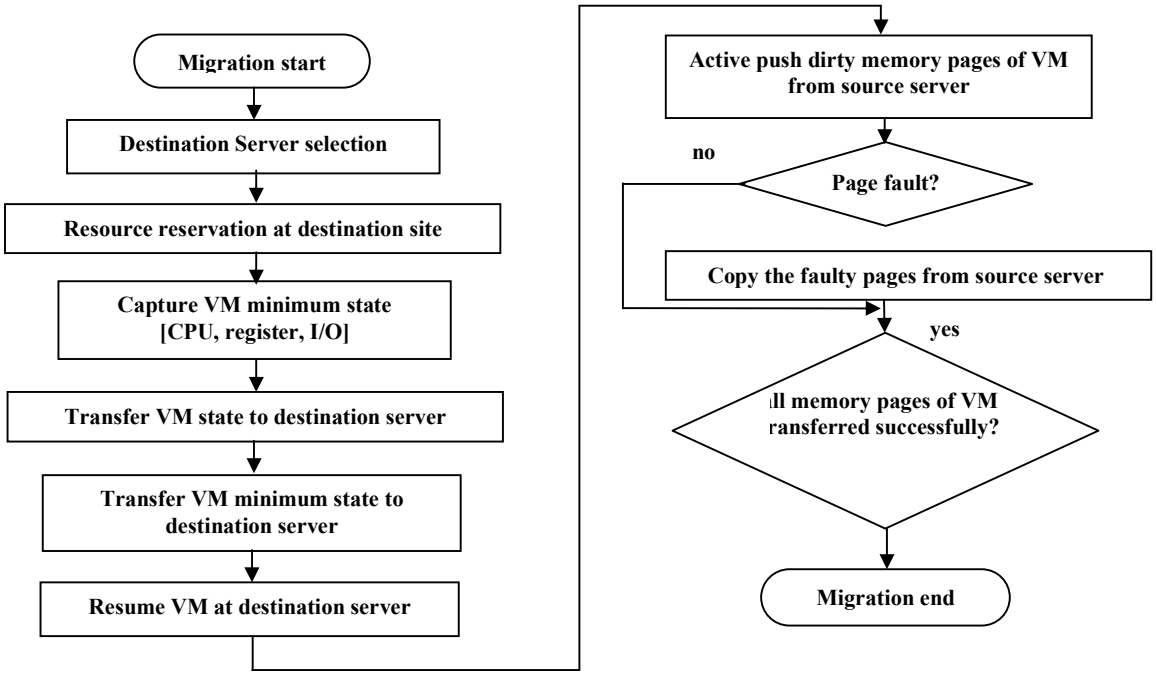


Figure 2: Post-Copy technique

3. RELATED WORK ON VM MIGRATION

Application of cloud computing are increasing continuously because of its solutions are available easily and cost effective. Even there are multiple cloud applications for variety of user requirements on pay per use basis. This also poses challenges before the cloud service providers. As the number of cloud service providers increased but proportionally clients are also increased drastically. New dimensions of applications increased the demands of more cloud resources like servers, data stores, security tools, etc. some of the research papers are included to prepare this papers. These papers are summarized below with their declared objectives, contributions, and conclusions.

Zhang, Han, and his team proposed a network-aware VM migration (NetVMM) in an overcommitted cloud and formulate it into a non-deterministic polynomial time-complete problem. This study aims to minimize network traffic costs by considering the inherent dependencies among VMs that comprise a multi-tier application and the underlying topology of physical machines and to ensure a good trade-off between network communication and VM migration costs. The mechanism that the swarm intelligence algorithm aims to find is an approximate optimal solution through repeated iterations to make it a good solution for the VM migration problem. In this study, genetic algorithm (GA) and artificial bee colony (ABC) are adopted and changed to suit the VM migration problem to minimize the network cost. Experimental results show that GA has low network costs when VM instances are small. However, when the problem size increases, ABC is advantageous to GA. The running time of ABC is also nearly half than that of GA. To the best of our knowledge, we are the first to use ABC to solve the NetVMM problem [1].

Zhang, Fu, and Yahyapour proposed a LayerMover that is a Fast virtual machine migration over WAN with three layer image structure. In this paper, they propose a new space-efficient VM image structure—three-layer structure. According to different functions and features, the data of a VM are separated into an Operating System (OS) layer, a Working Environment (WE) layer, and a User Data (UD) layer. It mainly focuses on improving migration performance through optimizing the data deduplication technique. Their experimental results show that three-layer image structure can improve data sharing between VMs, and the similarity ratio between WE images can reach 70%. The tests for LayerMover indicate that it can be significantly beneficial to VM migration across data centers, especially when multiple VMs which share base images are migrated [2]. Atif and Strazdins presented ARRIVE-F, a novel open source framework which addresses the issue of heterogeneity in virtualized compute farms, such as those hosted by a cloud infrastructure provider. Unlike the previous attempts, our framework is not based on linear frequency models and does not require source code modifications or off-line profiling. The heterogeneous compute farm is first divided into a number of homogeneous sub-clusters. The framework then carries out a lightweight ‘online’ profiling of the CPU, communication and memory subsystems of all the active jobs in the compute farm. From this, it constructs a performance model to predict the execution times of each job on all the distinct sub-clusters in the compute farm. Based upon the predicted execution times, the framework is able to relocate the compute jobs to the currently best-suited hardware platforms such that the overall throughput of the compute farm is increased. We utilize the live migration feature of virtual machine monitors to migrate the job from one sub-cluster to another. The prediction accuracy of our performance estimation model is over 80%. The implementation of ARRIVE-F is lightweight, with an overhead of 3%. Experiments on a synthetic workload of scientific benchmarks show that we are able to improve the throughput of a moderately heterogeneous compute farm by up to 25%, with a time saving of up to 33% [3].

Sun, Liao and his team proposes a new improved serial migration strategy and introduce the post-copy migration scheme into it. We then propose the m mixed migration strategy that is based on the improved serial migration strategy and the parallel migration strategy. Furthermore, we develop queuing models (i.e., the M/M/C/C and the M/M/C queuing models) to quantify performance metrics, such as the blocking ratio and average waiting time of each migration request. We evaluate the performance of the proposed migration strategy by conducting mathematical analysis, the numerical results show that our proposed strategy outperforms the existing approach [4]. Umesh and Keahey address the problem of network contention between the migration traffic and the Virtual Machine (VM) application traffic for the live migration of co-located Virtual Machines. Here, they propose a traffic-sensitive live VM migration technique to reduce the contention of migration traffic with the VM application traffic. It uses a combination of pre-copy and post-copy techniques for the migration of the co-located VMs (those located on the same source host), instead of relying on any single pre-determined technique for the migration of all the VMs. They show that the approach minimizes the network contention for migration, thus reducing the total migration time and the application degradation [5]. In cloud-based IOT application, sudden changes in the sensing environment cause spikes of data flowing into the cloud. This causes resource starvation in the virtual machine and initiates migration of virtual machine from one physical server to other. This paper proposes a resource-aware virtual machine migration technique. Any sudden change in the sensing environment is observed by clustering the servers. The suitable target server is selected based on the resource utilization and job arrival rate of the destination server. The proposed technique is implemented in cloud platform running analytics on smart agriculture application. The evaluation results show that the proposed method outperforms the state of art techniques in terms of the number of migrations, energy utilization and migration time [6].

Sotiriadis, Bessis, Buyya presented a Cloud VM scheduling algorithm that takes into account already running VM re- source usage over time by analyzing past VM utilization levels in order to schedule VMs by optimizing performance. We observe that Cloud management processes, like VM placement, affect already deployed systems (for example this could involve throughput drop in a database cluster), so we aim to minimize such performance degradation. The results show that our solution refines traditional instant-based physical machine selection as it learns the system behavior as well as it adapts over time. The analysis is prosperous as for the selected setting we approximately minimize performance degradation by 19% and we maximize CPU real time by 2% when using real world workloads [7]. Satpathy, Addya, and his team proposed a two-tier virtual machine placement algorithm. Firstly, we propose a queueing structure to manage and schedule a large set of VMs. Secondly, a multi-objective VM placement algorithm called crow search based VM placement (CSAVMP) is proposed to reduce the resources wastage and power consumption at the data centers. VM migration is an indispensable part of any cloud platform for activities like maintenance, load balancing, fault tolerance etc. Three different migration strategies namely serial, parallel, improved serial have been tested and a comparative result has been produced [8].

Paulraj, Francis, and his team proposed a combined forecasting technique to predict the resource requirement of any virtual machine. Based on the current and predicted resource utilization, live migration is performed by Combined Forecast Load-Aware technique. The outcomes indicate that the proposed approach has minimized the number of migrations, energy usage, and the message overhead when compared with the existing state-of-art technique [9]. Yazidi, Ung, and his team considered the problem of scheduling VM migration in those scenarios. Comprehensive simulations and real-life experimental results show that our approach is able to decrease the volume of separated traffic by a factor larger than 30% [10]. Filho, Monteiro, and his team presented a cloud computing background, a review of several proposals, a discussion of problem formulations, advantages and shortcomings of reviewed works. Furthermore, it highlights the

challenges for new solutions and provides several open issues, showing the relevancy of the topic in an increasing and demanding market [11]. Jin, Gao, and his team address the problem by designing an optimization scheme for live migration, under which according to pre- copy speed, the VCPU working frequency may be reduced so that at a certain phase of the pre-copy the remaining dirty memory can reach a desired small amount. The method improves migration liveness at the cost of application performance, and works for those applications for which interruption causes much more serious problems than quality deterioration. Compared to the original live migration, our experiments show that the optimized scheme can reduce up to 88% of application downtime with an acceptable overhead [12]. Wang, chen, Yuen and his team considered both delay and migration cost. They we formulate a joint optimization problem with both delay and migration cost considered, which is proved to be NP-hard. Simulation results demonstrate the effectiveness of the schemes, and show that the proposed algorithm is able to leverage the spatial variation in the VMs migration for delay and cost optimization [13].

This paper provides a better understanding of live migration of virtual machines and its main approaches. Specifically, it focuses on reviewing state-of-the-art optimization techniques devoted to developing live VM migration according to memory migration. This work also highlights the open research issues that necessitate further investigation to optimize the process of live migration for virtual machines [14]. They present different schemes to classify commonalities and discrepancies between the perspectives of researchers, based on metrics derived from the literature. Finally, open issues, challenges, and future directions are discussed for improving existing schemes and approaches [15]. To mitigate the service paralysis caused by DC failures, in this paper, we propose an effective parallel inter-DC VM migration framework with the consideration of VM grading, load balancing and elastic optical bandwidth allocation. Compared with previous end-to-end VM migration approach, our method shrinks migration duration exponentially with the higher migration profit and the lower blocking rate. Finally, the simulation results testify these superiorities, and the improvement ratio arrives up to 43% [16].

Saraswathi, Kalaashri, Padmavathi presented dynamic resource allocation scheme in cloud computing. Main principle of this work is that low priority jobs should not delay the execution of high priority jobs and to dynamically allocate VM resources for a user job within deadline [17]. Raghunath and Annappa presented an intelligent decision maker to trigger the migration by predicting the future workload and combining it with predicted performance parameters of migration process. Experimental results shows that migration is triggered at an appropriate point such that there are sufficient amount of resources available (15–20% more resources than high valued threshold method) and no application performance degradation exists as compared to properly chosen threshold method for triggering the migration [18].

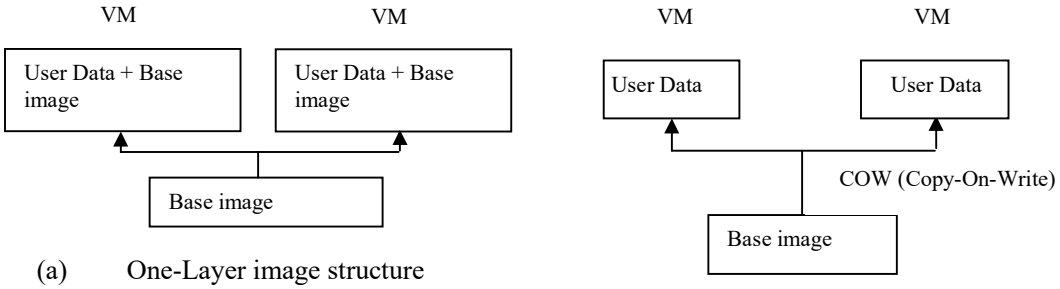
This paper proposes a demand-based preferential resource allocation technique that designs a market-driven auction mechanism to identify users for resource allocation based on their payment capacities and implements a payment strategy based on a buyer's service preferences. A comparison is drawn between the proposed allocation technique and the famous off-line VCG auction mechanism and results show a performance benefit in revenues to service provider, payments of cloud users besides ensuring an optimum resources use [19]. Shahapure, and Jayarekha proposed a new paradigm for virtual machine migration, based on the demands of the clients is considered. To achieve this an algorithm called Virtual Machine Migration Approach Based on Distance and Traffic is developed. The methodology is based on the traffic and the geographical locations of the data centers. The algorithm is executed at regular intervals to check for the traffic in the network. It also checks for the distance of the data centers where the client request has to be placed. The request is placed to the nearest data center with less traffic. This improves the performance by reducing the round trip time [20]. In this paper, we propose to virtualized the resources embedded on the vehicular nodes to allow multiple tenants to coexist and deploy their services on the same underlying mobile substrate. They study the

problem of virtual machine migrations through V2V communications between mobile nodes. To evaluate the impact of such migrations on the resource allocation process, we use the real traces of a bus transit system to simulate a vehicular network where virtual machines migrate via V2V communications. Our results show that virtual machines of several hundreds of Megabytes can migrate between moving buses [21]. This paper proposes several Vehicular Virtual Machine Migration (VVMM) schemes: VVMM-U (Uniform), VVMM-LW (Least Workload), VVMM-MA (Mobility Aware) and MDWLAM (Mobility and Destination Workload Aware Migration). Their performance is evaluated with respect to a set of metrics through simulations with varying levels of vehicular traffic congestion, Virtual Machine sizes and levels of load restriction. The most advanced scheme (MDWLAM), takes into account, the workload and mobility of the original host as well as those of the potential destinations. By doing so a valid destination will both have time to receive the workload and migrate the new load when necessary. The behavior of various algorithms is compared and the MDWLAM has been shown to demonstrate best performance, exhibiting migration drop rates that are negligibly small [22].

4. Image Structure of Virtual Machine

Cloud providers (such as Amazon EC2, Microsoft Azure, etc) use templates, called base images or gold images, to fast deploy VMs. A base image normally is installed with an OS or a combination of an OS and a software stack. This structure also will result in repeated OS data store, since many software stacks in data center are using same OS, for example, 56.4% VMs in Amazon EC2 are running with Ubantu.

A software stack denotes a couple of applications, which work together to provide a specific service. For example, LAMP (short for Linux, Apache, MySQL (or Maria DB) and PHP) is a typical software stack for web service.



(a) One-Layer image structure

(b) Two-layer image structure

Figure 3: Structure of One-Layer and Two-Layer image of VM

A base image only installs an OS, applications will be installed in User Data (UD). Many VMs in a data center are working with the same type of application stack. Cloud providers deploy some common software stacks as base images. VM images are stored in a coarse granularity which leads to limited data sharing between VMs.

At the beginning of cloud computing, cloud providers create VMs by cloning base images. The users of a VM directly operate with a replica of a template image. All data of the VM are stored in this image copy. It is named as one-layer image, shown above in figure 1 (a), and this cloning operation is time-consuming. The deployment time span is unbearable when a large number of VMs are requested by a user. Furthermore, each VM will generally a copy of a base image, which is space-consuming as well.

To tackle this issue of one-layer image structure, a novel VM deployment approach-COW, was proposed. By COW, a new empty image is created on the top of a base image. The base image remains read-only to the upper image, and all new data and the modifications to the base image are redirected to the new image file. The disk data of a VM are split into two image files: (1) Base image, and (2) VM image. This structure dramatically improves VM deployment speed because only a link between a new empty image and a base image is needed for each VM creation. Furthermore, a base image is shared by multiple VMs, which lowers the total space consumption of a data canter.

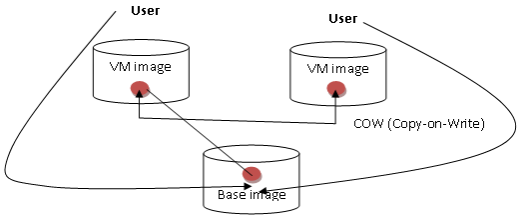


Figure 4: Application of Copy-on-Write

Data deduplication is a technology to reduce the total size of a dataset. It eliminates redundancy on block level and is more coarse grained than comparison. It chucks a file into blocks, and calculates a fingerprint for each block. Figerprints are utilized to identify duplicate blocks. MD5 and SHA-1 are two main algorithms for fingerprint calculation.
Files can be chunked into fixed-size or variable-size blocks. Fixed-size chucking is simple, but it suffers from the bit-shifting problem. However, variable-size chucking can recognize this change, and shifts the block boundaries. Rabin fingerprint is a typical variable-size chunking algorithm. Nevertheless, variable-size chunking is compute-inverse, which makes it hard to gain big performance improvement for live VM migration. Therefore, fixed-sized chunking is more popular for Live VM migration.

5. CONCLUSION

There are many issues with the Live VM migration and among other the major is how to fast migrate the so big storage data of these VMs through the Internet. Following this issue, we come to conclusion that VM structure plays the important role in such a migration. We have discussed various structures of VMs on the server. The VM images in a data center are stored in a finer granularity, which provides more migration optimization possibilities for VM migration. One-layer image structure is time-consuming, whereas, the two-layer image structure lowers storage space consumption and facilitates VM deployment speed. The software stacks may be place at different positions like (1) in the base image layer, or (2) in the User Data (UD) layer.

Acknowledgments

The authors are highly thankful for the people who are directly or indirectly supports for the preparation of this paper.

REFERENCES

[1] *W. Zhang, S. Han, Hui He, H. Chen, "Network-aware virtual machine migration in an overcommitted cloud", Future Generation Computer Systems, vol. 76 (2017), pp. 428-442*

[2] *Fei Zhang, Xiaoming Fu, R. Yahyapour, "LayerMover: Fast virtual machine migration over WAN with three-layer image structure", Future Generation Computer Systems, vol. 83 (2018), pp. 37-49*

[3] *Muhammad Atif, Peter Strazdins, "Adaptive parallel application resource remapping through the live migration of virtual machines", Future Generation Computer Systems, vol. 37 (2014), pp. 148-161*

[4] *Gang Sun, Dan Liao, Vishal Anand, Dongcheng Zhao, Hongfang Yu, "A new technique for efficient live migration of multiple virtual machines", Future Generation Computer Systems, vol. 55 (2016), pp. 74-86*

[5] *Umesh Deshpande, Kate Keahey, "Traffic-sensitive Live Migration of Virtual Machines", Future Generation Computer Systems, vol. 72 (2017), pp. 118-128*

[6] *G. J. Leelipushpam Paulraj, S. A. John Francis, J. Dinesh Peter, I. J. Jebadurai, "Resource-aware virtual machine migration in IoT cloud", Future Generation Computer Systems, vol. 85 (2018), pp. 173-183*

[7] *Stelios Sotiriadis, Nik Bessis, Rajkumar Buyya, "Self managed virtual machine scheduling in Cloud systems", Information Sciences vol. 433-434 (2018), pp. 381-400*

[8] *A. Satpathy, S. K. Addya, A. K. Turuk, B. Majhi, G. Sahoo, "Crow search based virtual machine placement strategy in cloud data centers with live migration", Computers and Electrical Engineering, vol. 69 (2018), pp. 334-350*

[9] *G. J. Leelipushpam Paulraj, S. A. ohn Francis, J. D. Peter, I. J. Jebadurai, "A combined forecast-based virtual machine migration in cloud data centers", Computers and Electrical Engineering, vol. 69 (2018), pp. 287-300*

[10] *A. Yazidi, Frederik Ung, H. Haugerud , K. Begnum, "Effective live migration of virtual machines using partitioning and affinity aware-scheduling", Computers and Electrical Engineering, vol. 69 (2018), pp. 240-255*

[11] *Manoel C. Silva Filho, Claudio C. Monteiro, Pedro R.M. Inácio, Mário M. Freire, "Approaches for optimizing virtual machine placement and migration in cloud environments: A survey", Journal of Parallel and Distributed Computing, vol. 111 (2018), pp. 222-250*

[12] *Hai Jin, Wei Gao, SongWua, Xuanhua Shi, Xiaoxin Wu, Fan Zhou, "Optimizing the live migration of virtual machine by CPU scheduling", Journal of Network and Computer Applications, vol. 34 (2011), pp. 1088-1096*

[13] *Xiumin Wanga, Xiaoming Chen, Chau Yuen, Weiwei Wu, Meng Zhang, Cheng Zhan, "Delay-cost tradeoff for virtual machine migration in cloud data centers", Journal of Network and Computer Applications, vol. 78 (2017), pp. 62-72*

[14] *Mostafa Noshy, Abdelhameed Ibrahim, Hesham Arafat Ali, "Optimization of live virtual machine migration in cloud computing: A survey and future directions", Journal of Network and Computer Applications, vol. 110 (2018), pp. 1-10*

[15] *Mirsaeid Hosseini Shirvani, Amir Masoud Rahmani, Amir Sahafi, "A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: Taxonomy and challenges", Journal of King Saud University - Computer and Information Sciences, 2018, pp. 1-20*

[16] *Cunqian Yu, Weigang Hou, Lei Guo, Yue Zong, "Parallel virtual machine migration in WDM optical data center networks", Optical Switching and Networking vol. 20 (2016), pp. 46-54*

[17] *Saraswathi AT, Kalaashri.Y. RA, Dr.S.Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing", Procedia Computer Science, vol. 47 (2015), pp. 30 - 36*

[18] *Bane Raman Raghunath and B. Annappa, "Virtual Machine Migration Triggering using Application Workload Prediction", Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), Procedia Computer Science, vol. 54 (2015), pp. 167 - 176*

[19] *Narander Kumar, Swati Saxena, "A Preference-based Resource Allocation In Cloud Computing Systems", 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), Procedia Computer Science, vol. 57 (2015), pp. 104 - 111*

[20] *Nagamani H Shahapure, P Jayarekha, "Distance and Traffic Based Virtual Machine Migration for Scalability in Cloud Computing", International Conference on Computational Intelligence and Data Science (ICCIDS 2018), Procedia Computer Science, vol. 132 (2018), pp. 728-737*

[21] *B. Barona, M. Campista, P. Spathisa, L. Henrique M. K. Costa, M. Diasde Amorim , Otto Carlos M.B. Duarte, Guy Pujolle, Yannis Viniotis, "Virtualizing vehicular node resources: Feasibility study of virtual machine migration", Vehicular Communications, vol. 4 (2016), pp. 39-46*

[22] *Tarek K.Refaat, Burak Kantarci, Hussein T. Mouftah, "Virtual machine migration and management for vehicular clouds", Vehicular Communications, vol. 4 (2016), pp. 47-56.*