Affinity based VM Migration: Taxonomy and Challenges

Pooja Gupta,Sameer Dev Sharma, Abhishek Kumar Pathak, Amarjeet Rawat

Abstract— ***using virtualization many Virtual Machines can run parallel on the same Host. For dynamic resource management, virtual machines can be migrated from residing host to a different. But before starting the migration some questions need to be answered like when to start the virtual machine migration, which VM to migrated and where? The Virtual Machine migration methods on the Virtual cloud environment has already been researched at length, but very few studies have focused on affinity-relations among virtual machines during migration, hence the key objective of this research paper, is to explore the Affinity-aware VM migration in detail and propose Affinity-Aware VM migration algorithms for migration of a group of VMs with affinity to a destination Host with less capacity than required. This paper also provides a brief review of several virtual machine migration techniques.***

Keywords—Virtualization, Clusters, Affinity Grouping, Resource Allocation, VM Packing.

I. INTRODUCTION

A server into its idle state can consume up to 70% of its power [1]. For instance, the Data Centre in any part of the globe may be over-utilized in daytime (from 9 AM to 9 PM) but same Data Centre may be idle during night [2]. To cut this unnecessary power consumption virtualization is applied. Using virtualization, Cloud computing provides software, hardware and platform based services on subscription based model and hence reducing wastage of resources. In virtualization, computing resources are in the shape of virtual machine. Virtualization is the backbone of Cloud computing, it automates the resource management. Efficient resource management optimizes capacity and infrastructure management [3, 4]. Using Virtualization many VMs can run simultaneously on the same Host [5]. A virtual machine can experience a dynamic workload and hence, resources can be over-utilized (hot spot) or under-utilized (cold spot) [6]. The process of VM migration is important in cloud environment for dynamic resource management [7].

A. Purpose of VM Migration

Main reasons for VM migration can be categorized as follows [2]:

1 Server Consolidation
2 Load Balancing
3 Hot Spot Mitigation

1. VM Migration for Server Consolidation

Consolidation is done to cut server sprawl [2]. Server sprawl is a process that can occur in data centres and can lead to poor hardware resource utilization. New algorithms are required for server consolidation that can efficiently manage the data centres and can minimize server sprawl in data centres. The main VM packing algorithms are bin-packing or vector- packing that make server as a bucket and pack whatever number VMs as could be expected on a similar Host in order to optimize resource utilization [8]. When many VMs are migrated to a Host other under-utilized Hosts can be powered off. Server consolidation helps to cut power-consumption and operational costs of the system.

2. VM Migration for balancing load

Load balancing in data centres is usually done to divide load evenly across all Hosts in the system [9]. Main purpose of load balancing is to make sure that none of the Host is over-utilized or under-utilized. Load balancing can be done in another case when a VM is experiencing a resource crunch and migrated to another machine which can host its resource requirements [2].

3. VM Migration for hot spot mitigation

VMs can be migrated for removing hot spot [9]. Upper threshold is a limit value of high resource utilization similarly lower threshold is a limit value of low resource utilization [9]. Any Host that is using resources more than upper threshold is said to have created hot spot, and any Host who is using resources less than lower threshold is said to have created cold spot.

B. Points to consider during VM migration

Before migrating a VM following points need to be considered:

1: At what time to start migration of a VM
2: VM Selection for migration.
3: Destination host machine selection for migration.

1. When to start migration of a VM

To trigger VM migration there can be many reasons:

a) Periodic migration

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author

Ms. Gupta*, Assistant Professor, Computer Application department, DIT University, Uttarakhand,India.

Mr. Sharma, Assistant Professor, Department of Computer Application, Uttaranchal University,India.

Mr. Abhishek Kumar Pathak, Assistant Professor, Assistant Professor, Department of Computer Application and holds the position of Faculty of Management and Business Studies.

Amarjeet Rawat' Assistant Professor, department of Computer Applications, UIM, Uttaranchal University,India.

b) Hot spot mitigation
c) Excess spare capacity
d) Load imbalance
e) Addition/removal of VM or Hosts

2. VM selection for migration

A VM may be selected for migration for different reasons:

a) Migrate a VM that is resource constrained

In the first case only a single VM will be migrated, i.e. a VM that is experiencing resource crunch. But this is not the case always. Migration needs high time and effort so a VM with less migration effort should be selected for migration. For example: VM1 is facing resource crunch in accessing a large amount of memory, so migrating VM1 will need high time and effort. Another VM on the same Host which is utilizing less memory is VM2 and it is better to migrate VM2 because total migration time will be less (assuming that memory released by VM2 is enough to moderate hot spot) and memory released by VM2 can be allocated to VM1. Figure 1.1 presents the migration case when a VM is resource constrained.

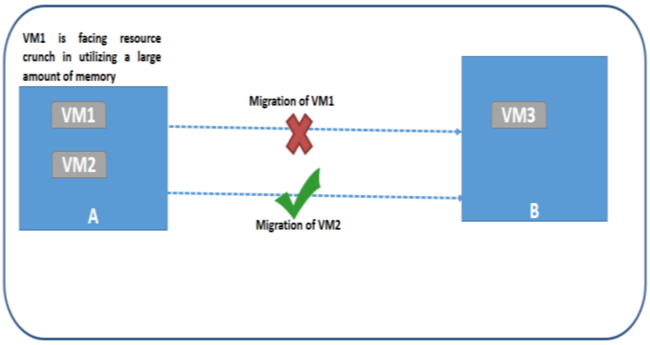
VM1 is facing resource crunch in utilizing a large amount of memory
VM1
VM2
A
Migration of VM1
Migration of VM2
VM3
B

Figure 1.1: Migration when VM is resource constrained

b) Migrate VMs for laod balancing or consolidation

In this case, all VMs and Hosts in the system must be checked for their availability and requirements, and then sort all the VMs according to their resource utilization level, appropriate VM can be selected and migrated to the Host that has enough hosting power.

c) Migrate affinity based VMs on the same host

Affinity-aware VMs can be transferred to the one Host to reduce communication cost [11], such as, if two VMs are communicating with each other and their inter-communication cost is more than their intra- communication cost. Then it is a better option to place them on same Host because this will reduce the network traffic and hence communication cost among VMs will also be reduced. Memory sharing VMs can also effect migration. Migrating all memory sharing VMs onto same Host would result in efficient memory utilization.

3. Destination host machine selection for migration

Most crucial phase of VM migration is to find the best destination Host to accommodate the coming VMs after migration. Destination Host choice to minimize waste of resources is a domain of explore itself.

a. Criteria for Destination Host Choice

1. Depending upon available resource capacity

Only resource capacity of a Host should not be considered, there are other factors as well which affect the choice of destination Host, like whether destination Host is a best fit solution? Or the performances of local VMs that are already running on it will be affected or not?

2. Depending upon affinity of VMs

This mechanism takes affinity among VMs into considering while grouping them for migration. Study of these mechanisms indicates that allocating all communicating VMs on the same Host can decrease the communication overhead. VMs sharing memory can also be allocated on the same Host in order to save the memory.

II. LITERATURE REVIEW

VM migration in an important process that can decrease the amount of active Hosts in a data centre [12]. VM migration also helps in workload balancing across all the virtual machines so that the over usage or under usage of the machines can be avoided. Workload balancing fairly distributes the load among all the machines and supports to avoid any circumstances of uneven load balancing; some machines are working at full load and other machines stay unused [6]. VM migration mechanism can be classified as follows:

i. Offline migration and live migration,

ii. Single migration and multiple migration,

iii. Non-affinity migration and Affinity –aware migration.

A. Offline VM migration

The first step in offline VM migration technique is to stop the execution of running VMs before migrating their memory and status to the destination [7]. Once all memory footprint and status are migrated only then VMs are restarted at the new destination. Two factors are considered during VM migration.

1. **Downtime:**
   Time during which VM is suspended and its facilities a re not accessible [12].
2. **Migration time:**
   total time needed to transfer a VM to fresh location at source without influencing its accessibility [13].

In the offline migration process, there will be some downtime during stop and restart phases of virtual machine migration technique. This downtime can affect the performance of the application.

B. Live VM migration

Live migration technique shifts the complete operating systems instances without any significant downtime from source to destination [14]. Live migration helps in load imbalance, fault management, network usage optimization and cloud bursting. Cloud managers are mostly interested in live migration[15].

Two techniques (Push and Pull) were proposed to support the workload distribution in a setup with multiple VMs using automated live migration technique [6].Workloads of VMs can trigger the migration process [2]. Researchers propose an intelligent decision maker. This decision maker can forecast the expected workload of the machine and can trigger the live migration [12].

C. Single and Multiiple VM Migration

A lot of research is being done in single VM migration direction but the number of research that considers the dilemma of live migration for multiple VMs is very less [7]. Some authors work in this area and pproposed a live migration test and capture on a Linux-based virtualization platform at the production level that dem ands the requirement of a stronger multi-VM migration approach. To cut down the total migration time for the live-migration of multiple VMs, a programming model is proposed by authors that defines the cost-sharing role between the perceived downtime of the user and the use of resources. An enhanced serial migration approach with post-copy migration technique was proposed by some authors. On the basis of enhanced serial migration technique and the parallel migration technique, authors also suggest m-mixed migration technique but their work did not consider affinity of VMs for migration [14]. A scheduling method was suggested to optimize the multi-virtual machine migration, its prime consideration was on the modeling and formalizing the migration crisis of various virtual machines [15].

D. Eneragy savign VM Migration Research

The process of virtual machine migration is good to maximize resource utilization but network energy is consumed during VM migration hence smallest number of machine should be migrated; otherwise it would contribute to the total CO2 emissions.

When deploying applications in distributed cloud network s within decentralized service delivery architectures, energy consumption is the primary problem [16]. some virtual machine migration techniques also focus on energy-saving. Server consolidation in virtualization provide energy efficiency [17]. These techniques improve the resource usefulness and performance because of reduces energy consumption and carbon footprint [18, 19, 20].

The frequency The frequency of VM migrations can be minimized by using ideal internet deterministic algorithms, VM Placement Optimization Algorithm and MBFD Algorithm, it works fine with a diver infrastructure and VMs. This technique is independent of workload type [21]. Its drawback is that; it lacks on a real-world cloud platform. A decision whether to perform migration or not should be taken before actual VM migration. A load prediction algorithm is proposed by the authors. If algorithm comes with up the decision to migrate then a suitable destination host is searched for migration [22].

Another work is recommended that can monitor the interr elationship between energy consumption, frequency of VM migrations and SLA violation[23]. Now the focus of research has been shifted to live VM migration techniques for energy –aware computing[24].

E. Affinities-AwareVM Migration

Affinity –aware VM migration is still not fully explored. Few researches have been done in this area. All the existing VM selection policies for VM migration e.g. minimum migration time (MMT), maximum correlation (MC) and random choice (RC) don't focus on the affinity of VM's for migration.

Pacer [25] and COMMA [26] reported a performance downfall in a special situation where two or more VMs are working on a single job and these VMs are spread across the different geographically clouds because inter-cloud link bandwidth drives significant degradation in the performance. Usually in a cloud data centre different racks are there and physical machines (PM) resides on a number of racks. It is observed that network bandwidth inside the rack is better than between the racks. PMs are distributed on these racks based on some logical groups for specific applications. [27] Identifies four relations between VMs.

1) VMs on same rack
2) VMs on different racks
3) VMs of same group
4) VMs from different group

It is better to place VMs with affinity on the same PM. These affinity relations can be memory share or traffic dependency. For security reasons and to distribute the load evenly, VMs who compete for resources or belong to dissimilar cloud customers must not be allotted on one PM [27]. Above researches confirm the importance of affinity-aware VM placement. Affinity relations among VMs for instance memory share, traffic dependency and resource competition are important to consider in order to save energy and increase performance. [11] Identifies the affinity-related VM grouping- scheduling and offers an alliance system which uses an algorithm named heuristic bin packing to arrange VM groups to PMs.

All these results inspire us to go ahead with the further search on:

1. How to group a number of VMs with affinity association across them,
2. How to assign these groups of VMs to a single PM with maximum capacity across all available PMs But still less than the total VMs requirement.

III. PROPOSED WORK

Communication dependency among the VMs can be derived from network traffic, if a number of the VMs with communication dependency square measure packed and migrated to a PM, then the networking cost across physical network may be decreased while the application overall performance may be increased. [11] Considered any nature of dependencies across VM pair as affinity aware VMs.

Selection of destination PM for VM migration is still an area of research itself. To address the problem of destination PM choice for migration of a group of VMs with affinity, we have proposed two cases. Case I deals with the problem when destination PM has sufficient capacity to serve the demand of affinity-based group of migrating VMs. Case II deals with the problem when destination PM has maximum available resources among all PMs but not sufficient to cater the demand of group of migrating VMs.

Cases during affinity-based VM Migration

Let's assume all PMs are homogeneous and their load capacity is same. Each PM can host at most 4 VMs. for the simplicity we've divided the affinity-based VM migration into three cases.

Case I: It represents the scenario when there exist a Host in the arrangement that has adequate resources to cater the requirement of migrating group of VMs.

Case II: It represents the scenario when there exists a Host in the system that has maximum available resources among all Hosts but not sufficient to cater the demand of migrating group of VMs. case 2 is further divided into two cases as II (a) and II (b).

- ***Case II (a):*** It represents a scenario when there is possibility of swapping of VMs among Hosts.
- ***Case II (b):*** It represents a scenario when swapping of VMs is not possible.

Case III: This case is an extension of case II, It represents a scenario when there exists a Host in the system that has maximum available resources among all Hosts but not sufficient to cater the demand of migrating group of VMs and the same Host has some affinity-based local VMs already, so migrating any VM that is affinity related to some other VMs to make space for another group of affinity –aware VMs is useless.

1. Case I

When there exists a PM that has sufficient resources to cater the demand of group of migrating VMs. For the simplicity assume there are three Hosts and three VMs (VM_1, VM_2 and VM_3) in a system. VM_1, VM_2 and VM_3 are organized on host 'A', 'B' and 'C' respectively. These VMs (VM_1, VM_2 and VM_3) are affinity based. To reduce network traffic they should be migrated to the same host. Case I is simple, as it says there exists a host that has enough resources to run VM_1, VM_2 and VM_3. Figure 3.1 presents the case I when there exists a Host that has sufficient resources to cater the demand of group of migrating VMs.

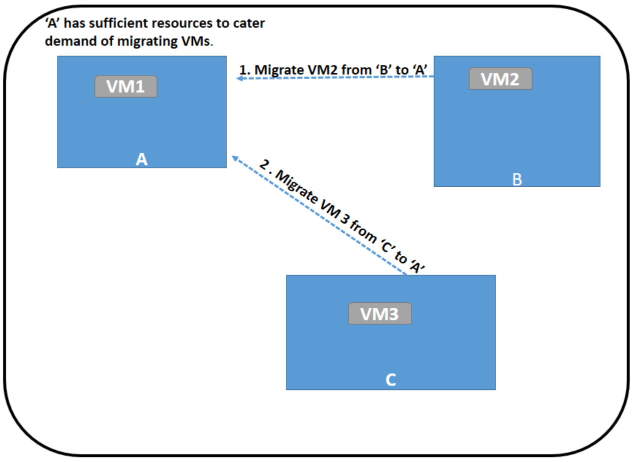
'A' has sufficient resources to cater demand of migrating VMs.
VM1
A
1. Migrate VM2 from 'B' to 'A'
VM2
B
2 . Migrate VM 3 from 'C' to 'A'
VM3
C

Figure 3.1: Case I: when there exists a Host that has sufficient resources to cater the demand of group of migrating VMs

*2.*Case II

When there exists a Host that has maximum available resources among all Hosts but still not sufficient to cater the demand of migrating group of VMs.

Case II (a): VM migration with swapping

For example, let's assume VM_1, VM_3, VM_6 and VM_7 are affinity based VMs and they should be migrated to the same the same Host. Host 'A' has maximum available resources among all hosts but still not sufficient to run VM_1, VM_3, VM_6 and VM_7. If 'A' does not have sufficient capacity to host migrating VM(s), 'A' will provide a list of possible swap candidates, a swap candidate is a subset of its hosted VMs appropriate for potential swapping with VM_3. If 'B' finds a suitable swap candidate (say, VM_2), then the swapping will be performed with VM_1 migrating to 'A' and the suit able swap candidate (VM_2) migrating to 'B'. This swapping operation is useful only when it provides better VMs placement. Two types of VMs that could be potential swap candidates during this situation are:

Type 1: A conceivable applicant could be a VM that will benefit by shifting from 'B' to 'A'. Applicant could be any VM whose system traffic to machine 'A' is more than its inner machine traffic and may have taken a tried at moving to machine 'A' ineffectively previously.

Type 2: Another conceivable competitor could be a remote VM, any VM which isn't speaking with other VMs, execution of such VM would not be influenced independent of where it is set.

In this case we can first migrate the VM_3 to 'A', then migrate VM_2 to 'B' and finally move VM_6 and VM_7 to 'A'. Swapping of VM_2 with VM_3 is done only if VM_3 will benefit from moving 'A' to 'B'. VM_3 traffic to machine 'B' could be higher than its intra-traffic within machine 'A'. Fig. 3.2 presents the case of VM migration with swapping when there exists a Host that has maximum available resources among all Hosts but not sufficient to cater the demand of group of migrating VMs.

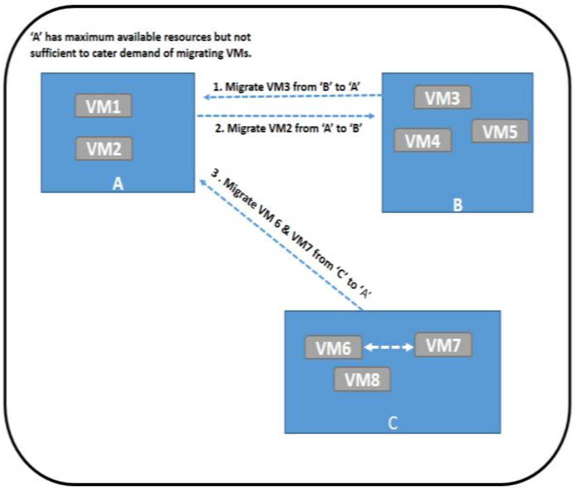
'A' has maximum available resources but not sufficient to cater demand of migrating VMs.
VM1
VM2
A
1. Migrate VM3 from 'B' to 'A'
2. Migrate VM2 from 'A' to 'B'
VM3
VM4
VM5
B
3. Migrate VM 6 & VM7 from 'C' to 'A'
VM6
VM7
VM8
C

Fig. 3.2: case II (a): VM migration with swapping when there exists a Host that has maximum available resources among all Hosts but not sufficient to cater the demand of group of migrating VMs.

Case II (b): Without swapping

Swapping of VMs is not always possible, there may be case when there are no proper swap candidates, and then 'A' provides a list of neighbours: nodes which are adjacent to it by means of network bandwidth/latency. In this case when 'B' will call individually every neighbours in turn by recursively initiating the relocation procedure. To limit the number of attempts, every node is communicated at most once. Server will quit the migration attempts if no neighbour can host the VM. Fig.3.3 presents the case of VM migration without swapping when there exists a Host that has maximum available resources among all Hosts but not sufficient to cater the demand of group of migrating VMs.

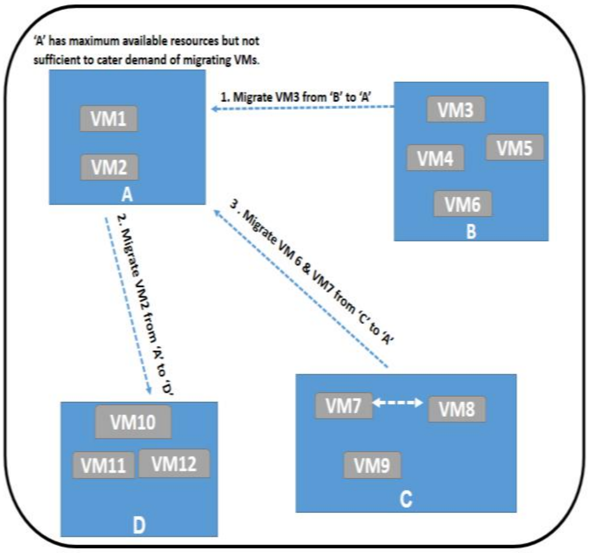
'A' has maximum available resources but not sufficient to cater demand of migrating VMs.
VM1
VM2
A
1. Migrate VM3 from 'B' to 'A'
VM3
VM4
VM5
VM6
B
2. Migrate VM2 from 'A' to 'D'
3. Migrate VM 6 & VM7 from 'C' to 'A'
VM10
VM11
VM12
D
VM7
VM8
VM9
C

Fig. 3.3: case II (b): VM migration without swapping when there exists a Host that has maximum available resources among all Hosts but not sufficient to cater the demand of group of migrating VMs.

Proposed Algorithm for Case II (a) and Case II ((b)
m=total no of Hosts available
VMG=a group of affinity-aware VMs to migrate
HG=A group of Hosts where VMG are residing

Algorithm (1)

```
Sort Hosts in decreasing order of resources available.
Max_Host= Host with maximum available resources
v € VMG
p € HG
If Max_Host_Capacity< total_VMG_requirement then
    Select suitable swap candidate in Max_Host
    If suitable swap candidate available then
        Perform swapping
        P ← Local_VMs_of_Max_Host
        Max_Host ← v
        flag= 1
    else
        for i=2 to i ≤ m do
            If Host_i_Capacity ≥ Local_VMs_of_Max_Host_Req then
                migrate local VMs of Max_Host to Host_i
                break
            else
                i++
            end if
        end for
    end if
    if i==m then
        return failed
    else if flag ==1
        Max_Host ← VMG-v
    else
        Max_Host ← VMG
    end if
    Update the resource availability of all Hosts
    Return success
end if
end if
```

3.1.1 Case 3

3. Case III

When Host has some local affinity-aware VMs, so migrating any VM that is affinity related to some other VMs to make space for another group of affinity –aware VMs is useless. We have to think of some different solution. In this case second best Host let's say Host_2(in terms of remaining resource available) can be considered as the destination and if local VMs in PM_2 are not affinity-aware then they can be migrated to some other PM to make space for gang migration of affinity-aware VMs.

Let's assume 'A' is the machine with maximum resource available but still not sufficient and all local VMs on 'A' are affinity based, so VM from 'A' cannot be migrated to make space for gang migration. In this particular case, second best host let's say 'B' (in terms of remaining resource available) can be considered as the destination host and if local VMs in 'B' are not affinity-aware then they can be migrated to some other host to make space for gang migration of affinity-aware VMs.

Algorithm for case III

m=total no of Hosts available

VMG=a group of affinity-aware VMs to migrate

HG=A group of all Hosts where VMG are residing

Max_Host= Host with maximum available resources

V € VMG

P € HG

Algorithm (2)

```
Begin
Sort all Hosts in decreasing order of resources available.
If Max_Host_Capacity < total_VMG_requirement, then
  If local VMs in Max_Host are affinity based then
    for i=2 to $i \leq m$ do
      if local VMs in $H_i$ are not affinity based then
        for j=3 to $j \leq m$ do
          If $H_j$_Capacity >= Local_VMs_of_$H_i$_Req then
            migrate local VMs of $H_i$ to $H_j$
            break
          else
            j++
          end if
        end for
      else
        i++
      end if
    end for
  end if
  if i==m
    return failed
  else
    $H_i \leftarrow$ VMG
    Update the resource availability of all Hosts
    Return success
  end if
end if
```

IV. RESULT DISCUSSION AND CONCLSION

We have proposed two algorithms for affinity based VM migration. Both algorithms consider the situation when destination PM does not enough resources to cater the demand of migrating VMs. In first algorithm we assume that local VMs in destination PM are not affinity based to any local VM can be migrated to free the space for affinity based migrating VMs.

Second algorithm focus on the case when all local VMs in destination PM are affinity based, so migrating those affinities based local VMs is pointless, hence we have to look for second best PM in terms of availability of resources. We check all the PMs as a candidate for destination PM. If no PM is available, then we drop the idea of migration.

For further study, authors will try investigate some techniques to implement proposed affinity-aware VM migration algorithms and perform experiments in order to compare their efficiencies.

REFERENCES

1. Beloglazov, Anton, Rajkumar Buyya, Young Choon Lee, and Albert Zomaya. "A taxonomy and survey of energy-efficient data centers and cloud computing systems." *Advances in computers* 82, no. 2 (2011): 47-111.
2. Mishra, Mayank, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo. "Dynamic resource management using management using virtual machine migrations." *IEEE Communications Magazine* 50, no. 9 (2012).
3. Amazon, https://aws.amazon.com/ec2/.
4. Vmware,https://www.vmware.com/solutions/consolidation.html.
5. Pachorkar, Nilesh, and Rajesh Ingle. "Affinity Aware VM Colocation Mechanism for Cloud." *International Journal of Advanced Computer Research* 4, no. 4 (2014): 956.
6. Forsman, Mattias, Andreas Glad, Lars Lundberg, and Dragos Ilie. "Algorithms for automated live migration of virtual machines." *Journal of Systems and Software* 101 (2015): 110-126.
7. Sun, Gang, Dan Liao, Vishal Anand, Dongcheng Zhao, and Hongfang Yu. "A new technique for efficient live migration of multiple virtual machines." *Future Generation Computer Systems* 55 (2016): 74-86.

8. Kamali, Shahin. "Efficient Bin Packing Algorithms for Resource Provisioning in the Cloud." In *Algorithmic Aspects of Cloud Computing*, pp. 84-98. Springer International Publishing, 2016.
9. Choudhary, Anita, M. C. Govil, Girdhari Singh, Lalit K. Awasthi, E. S. Pilli, and Nitin Kumar. "Improved Virtual Machine Migration Approaches in Cloud Environment." In *Cloud Computing in Emerging Markets (CCEM), 2016 IEEE International Conference on*, pp. 17-24. IEEE, 2016.
10. Md.Anit Khan, Andrew Paplinski, Abdul Malik Khan, Manzur Murshed, and Rajkumar Buyya, Dynamic virtual Machine Consolidation Algorithms for Energy-Efficient Cloud Resource Management: A Review, Sustainable Cloud and Energy Services: Principles and Practice, W.Rivera (editor), Springer International Publishing AG, 2017(in press, accepted in April 2017).
11. Chen, Jianhai, Kevin Chiew, Deshi Ye, Liangwei Zhu, and Wenzhi Chen. "AAGA: Affinity-aware grouping for allocation of virtual machines." In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pp. 235-242. IEEE, 2013.
12. Raghunath, Bane Raman, and B. Annappa. "Virtual Machine Migration Triggering using Application Workload Prediction." *Procedia Computer Science* 54 (2015): 167-176.
13. Wang, Chonghua, Zhiyu Hao, Lei Cui, Xiangyu Zhang, and Xiaochun Yun. "Introspection-Based Memory Pruning for Live VM Migration." *International Journal of Parallel Programming* (2016): 1-12.
14. Clark, Christopher, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. "Live migration of virtual machines." In *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pp. 273-286. USENIX Association, 2005.
15. Medina, Violeta, and Juan Manuel García. "A survey of migration mechanisms of virtual machines." *ACM Computing Surveys (CSUR)* 46, no. 3 (2014): 30.
16. Sabry, Noah, and Paul Krause. "Optimal Green Virtual Machine Migration Model." *International Journal of Business Data Communications and Networking (IJBDCN)* 9, no. 3 (2013): 35-52.
17. Kaur, Tarandeep, and Inderveer Chana. "Energy efficiency techniques in cloud computing: A survey and taxonomy." *ACM Computing Surveys (CSUR)* 48, no. 2 (2015): 22.
18. Kansal, Nidhi Jain, and Inderveer Chana. "Existing load balancing techniques in cloud computing: a systematic review." *Journal of Information Systems and Communication* 3, no. 1 (2012): 87.
19. Wang, Rui, Wei Le, and Xuejie Zhang. "Design and implementation of an efficient load-balancing method for virtual machine cluster based on cloud service." (2011): 321-324.
20. Brunzel, Tino, and Débora Di Giacomo. "Cloud computing evaluation: how it differs to traditional IT outsourcing." (2010).
21. Beloglazov, Anton, Jemal Abawajy, and Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." *Future generation computer systems* 28, no. 5 (2012): 755-768.
22. Shaw, Subhadra Bose, and Anil Kumar Singh. "Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center." *Computers & Electrical Engineering* 47 (2015): 241-254.
23. Choudhary, Anita, M. C. Govil, Girdhari Singh, and Lalit K. Awasthi. "Energy-efficient resource allocation approaches with optimum virtual machine migrations in cloud environment." In *Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference on*, pp. 182-187. IEEE, 2016.
24. Zhao, Jia, Liang Hu, Yan Ding, Gaochao Xu, and Ming Hu. "A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment." *PloS one* 9, no. 9 (2014): e108275.
25. Zheng, Jie, T. E. Ng, Kunwadee Sripanidkulchai, and Zhaolei Liu. *Pacer: Taking the guesswork out of live migrations in hybrid cloud computing*. Rice University Technical Report, Tech. Rep. TR13-01, 2013.
26. Zheng, Jie, Tze Sing Eugene Ng, Kunwadee Sripanidkulchai, and Zhaolei Liu. "Comma: Coordinating the migration of multi-tier applications." In *ACM SIGPLAN Notices*, vol. 49, no. 7, pp. 153-164. ACM, 2014.
27. Su, Kui, Lei Xu, Cong Chen, Wenzhi Chen, and Zonghui Wang. "Affinity and conflict-aware placement of virtual machines in heterogeneous data centers." In *Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on*, pp. 289-294. IEEE, 2015.

AUTHORS PROFILE

Ms. Gupta, is associated with DIT University as an Assistant Professor in Computer Application department. Ms. Gupta has done M. Tech (CSE) with Honors from Uttarakhand Technical University. She is currently pursuing PhD from DIT University in Deep Learning. She has published 10 papers in International and National journals of repute. She has 12 years of teaching experience. Her subject interests include Design and Analysis of Algorithm, cloud Computing, Modeling and Simulation and MATLAB. Her research interest include Machine learning and deep learning.

Mr. Sharma, is presently working with department of Computer Application as an Assistant Professor Uttaranchal University. He is MCA, M.Phil in Computer Science and presently pursuing his PhD in Deep Learning. He has vast experience a mixture of teaching and industry. He has 3 years of Industry and over 12 years of teaching experience. He has published 6 papers in International and National journals of repute. His subject specialization is Java, Advance Java, C++, LINUX, and research interest is in Machine learning, Deep Learning, and Big Data.

Mr. Abhishek Kumar Pathak is serving as Assistant Professor in Department of Computer Application and holds the position of Faculty of Management and Business Studies. He has more than 6years experience of teaching in postgraduate and under graduate level. His Area of interest are Web Development and currently PhD pursuing from UTU. He is also the member of Alumni Association and Cultural Committee of University as well as UIM.

Presently working with UIM, Uttaranchal University as an Assistant Professor in the department of Computer Applications. Earlier worked as a Head of Department, Faculty of Computer Application in Alpine Institute of Management & Technology, Dehradun. Mr. Rawat is pursuing Ph.D. (Deep Leaning) in CSE. Mr. Rawat has 11 years of Teaching Experience. He has received M. Tech (CSE) from Graphic Era University and Masters in Computer Application from MDU, Rohtak.