Institute of Advanced Engineering and Science

International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol. 3, No. 6, December 2014, pp. 338 – 345
ISSN: 2089-3337

A Review on Efficient Virtual Machine Live Migration: Challenges, requirements and technology of VM migration in cloud

Sangeeta Sharma[*] and Meenu Chawla[**]
[**]Department of Computer Science & Engg , MANIT, Bhopal

Article Info

Article history:
Received Jun 27, 2014
Revised Aug 19, 2014
Accepted Aug 29, 2014

Keyword:
Live migration
Resource management
Virtualization
Availability
Downtime

ABSTRACT

With the rise of cloud computing technology, various multimedia-driven applications in the domain of education, healthcare and surveillance are being developed and extensively used. Deployment of these applications leads to the significant multimedia-content traffic load, increased demand of resources, and failovers. Towards these issues, this paper presents a virtualized solution by means of virtual machine live migration approach to enhance availability, resource management, power management, and fault-tolerance. This paper focus on the challenges, requirements, and current work to identify the impact of virtual machine live migration in cloud environment to serve the modern datacenters with highly increasing needs.
This paper describes various virtual machine live migration techniques and group different methods based on their techniques. It shows the effect of various techniques towards the goal of efficient virtual machine live migration. Finally, paper is concluded with some open research issues and presents future work related to it.

Corresponding Author:
Sangeeta Sharma
Department of Computer Science and Engg , MANIT, Bhopal
sanjsharma29@gmail.com

1. INTRODUCTION

The general motivation for the adoption of cloud is to get information technology (IT) resources as a subscription base service. Cloud allows sharing of multitenant resources by several users through the Internet. Cloud gives the illusion of virtually infinite computing infrastructure to the users and pay as use model and provide every network-accessible computing resources as-a-service. The basic service model offered by cloud includes (1) Software as a service (SaaS), in which various softwares are available online for the users and pay as per use. (2) Platform as a service (PaaS), which facilitates users to develop, manage and hosting of software product from the third party infrastructure. (3) Infrastructure as a service (IaaS), which provides the storage computation and network capabilities as a services to the users in the form of virtual machine (VM). By moving the virtual machines, within a cloud environment provides fast and early scalability in order to fulfill dynamic resource needs.

Cloud offers a wide variety of software products, services and advanced capabilities: automated scalability, high availability, on-demand provisioning, and pay per use, being some of the most relevant. In order to support these capabilities, virtualization technology acts as a foundation. Virtualization gives an abstract view of hardware by means of virtual machine. Virtualization technology, runs several operating systems simultaneously on one physical machine in the form of virtual machine, has become a core concept in cloud based data centres shown in Figure 1. With virtualization technology, cloud based data centers are driven by benefit of hardware consolidation, load balancing, maintenance, and many more [1]. Because of these benefits, adoption of cloud model is prioritized over traditional computing models.

In the present scenario, majority of the applications dealt by the cloud computing are computation-intensive and data-intensive. These applications require large amount of resources and generate a large amount of dataset which can be further increased exponentially. Availability of resources and services, lower service delay, resource (capacity)

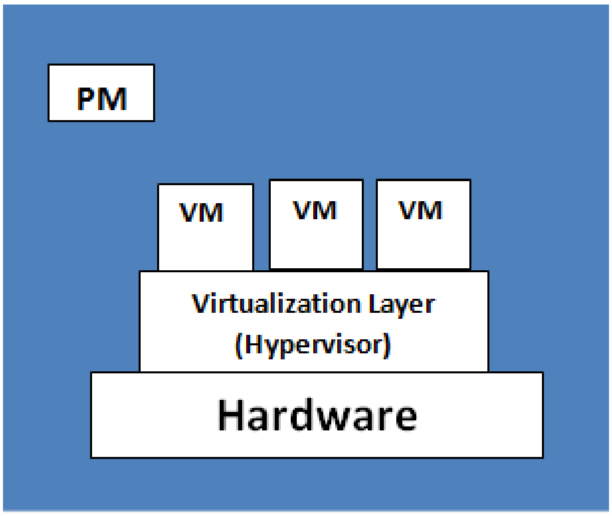

Figure 1. Virtualised view of physical machine

demand management, power management and fault tolerance for cloud data centers are the some challenges needs attention in order to handle these applications efficiently. In this paper virtual machine live migration presented as a virtualization tool for cloud in order to resolve these challenges up to a certain extent. Virtual machine live migration transparently moves a running virtual machine to a new physical host with minimal service interruption. Major forces behind the need of virtual machine live migration are towards fulfillment of the service level agreement (SLA) as committed by the cloud service provider especially on the point of:

- High service availability, i.e., the level of service given by the provider should be as high as possible.
- Resource management,(i.e.,) resources should be properly managed in order to fulfill the dynamic need of resources demanded by the user.
- Fault management, to act upon a potential failure in order to prevent from an imminent failure or to be able to act as recovery mechanism.
- Power management, by migrating the virtual machine from the under-utilized servers and switching off them.

Virtual machine live migration is supported by various widely available virtualization platforms like Xen [2], VMware [3], KVM [4], and HyperV [5]. It has become an attractive tool for the cloud managers. Following questions must be answered before the virtual machine live migration actually takes place: *Which is suitable type of virtual machine live migration method? What parameters affect the performance of virtual machine live migration? Which component of virtual machine has to be moved actually during the migration? What are the overheads imposed by the migration which may lead to performance loss?* In order to answer these questions, we need to understand how virtual machine migrates through the basic architecture of virtual machine live migration shown in Figure 2.

The goal of this paper is therefore to identify and discuss research challenges for adopting the virtual machine live migration as tool for better resource management which further leads to the high service performance of the cloud. The framework of this discussion is structured around the migration of four-layered virtual machine migration from source to destination host while the virtual machine is still running on source as shown in Figure 2. This paper also addresses the general principles and challenges that arise on each layer when virtual machine is migrating. The focus of the discussion in this paper will mostly be the first three components of virtual machine while some approaches for layer four of virtual machine are also discussed.

The major components of this paper as follows:

1. Categorization of the various types of virtual machine migration methods on the cloud environment.
2. Analysis of impacts of, performance parameters on virtual machine live migration and overhead imposed by the migration process.
3. Survey of various strategies used to increase the performance of virtual machine live migration.

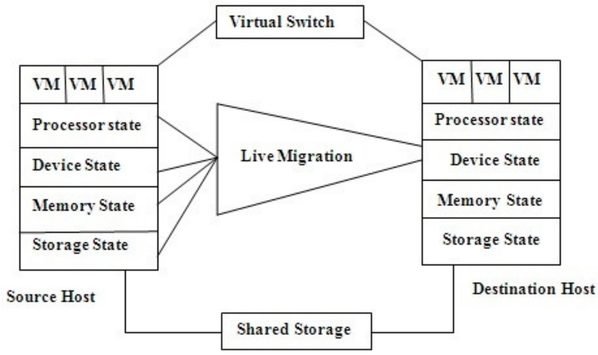

Figure 2. Overview of live migration architecture

4. A systematic identification of challenges.

The rest of the paper is organized as follows: Section 2. presents the major techniques for virtual machine live migration. Section 3. presents essential performance parameters which are used to define the effectiveness of migration mechanism. Section 4. shows the related work in order to improve the basic virtual machine migration approach. Finally, section 5. concludes the paper, summarizing the important points and discussing future work.

2. MIGRATION BACKGROUND AND TYPES

Live migration of virtual machine handles main four key aspects: processor state, connecting devices state, memory, and storage state.

- CPU and device state: The internal state of CPU and device information is small enough to transfer, which impose a minimum impact on the performance of live migration.
- Storage state: It is an optional part of virtual machine live migration because most of the cloud based data centers are equipped with network attached storage (NAS). So, if the on-disk virtual machine image is accessible to both the source and destination machines then there is no need to transfer the storage content during the virtual machine live migration. If not, full storage content has to be transfered which is very large in size, and will greatly affect the migration time.
- Memory state: Virtual machine memory state contains larger amount of information to transfer as compared to CPU state. It includes memory states of both the guest OS and all the running applications within the virtual machine. Transfer of larger memory content during the live migration of virtual machine affects the performance and migration time.

Majority of the research work focuses on the efficient live migration of virtual machine memory content. The following types of migration [6] are used for migrating memory content when a virtual machine is running a live service shown in Figure 3:

- Type 1: Stop and Copy is simple but time consuming approach in which virtual machine is stopped at source side and resumed on destination after the complete transfer of virtual machine.
- Type 2: Pre-Copy approach is a combination of iterative push phase followed by the stop and copy phase. In the iterative push phase, virtual machine memory pages are iteratively transferred to the destination till synchronized copy of image is available at the destination. Then virtual machine at source is suspended and resumed at destination with the last page memory. Its performance is highly dependent upon the memory page dirty rate. The majority of the literature work uses this type of virtual machine memory migration.
- Type 3: Post Copy approach gives low interruption delay by suspending the virtual machine at the source and resumes execution at the destination with its minimal processor state, and memory pages. The remaining pages are transferred iteratively as per the required demand till the complete memory image is transferred. In order to reduce the number of page faults during the post copy memory transferring phase, different variants are:
  - Post copy via active pushing: In this, memory pages are proactively transferred to the destination, by which several page faults are serviced simultaneously.

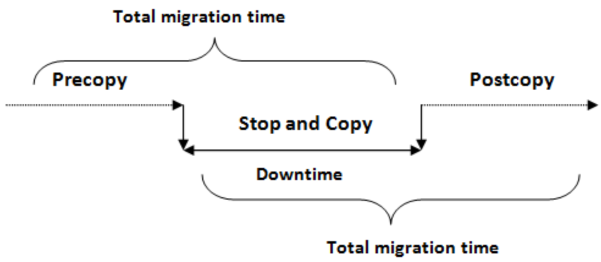

Figure 3. Types of live virtual machine migration

- Post copy via Pre-paging: With the requested page, additional pages are transferred based on the spatial locality.
- Hybrid approach: It is the proper mixing of pre-copy and post copy approach. At the first iteration of memory transfer it works as pre-copy approach and from the next iteration it works as post copy approach. It utilizes the features of both approaches.

3. PERFORMANCE PARAMETERS

In order to find the effective migration type, this section presents some performance parameters and factors which shows the effectiveness of virtual machine memory migration techniques and they are as follows:

1. Total migration time: It is the total time required to start execution at the destination node with the complete memory content of virtual machine. It should be as minimum as possible.
2. Downtime: It is the time difference when migrating virtual machine stops working at source, till it resumes again at destination. For higher availability, it is required to keep downtime short as possible.
3. Total pages transferred: It gives the total amount of pages transferred during the migration process including duplicated pages. It affects the available network bandwidth and slows down the migration process. It should also be as low as possible.
4. Overhead: It refers to the additional operations that are not part of the actual migration process, which are imposed by the migration technique used for virtual machine migration. It shows the impact of migration process on the performance of application. It simply leads to the application performance degradation.

Factors which affect the performance of virtual machine live migration are as follows:

1. Page dirty rate: It is the modification rate of virtual machine memory pages during the migration process which directly affects the number of pages transferred during the iterative copy phase.
2. Migration link bandwidth: Higher the link bandwidth, faster is the virtual machine migration takes place. It is the most influencing parameter which affects the performance of virtual machine live migration.
3. Total number of iterations: It shows the total number of iteration required to complete the migration process. It is directly proportional to the page dirty rate i.e., higher the dirty rate more the number of iterations required to complete the migration process, which further affects migration time.

Strunk [7] and Voorsluys [8] present a study of current approaches to determine the costs of virtual machine live migration and the parameters that may affect on migration costs.

4. LITERATURE SURVEY

This section presents analysis of several existing work for the improvement of virtual machine live migration processes, in order to contribute towards the development of optimized and efficient solution. To achieve the goal of lower service delay during migration, majority of work uses the pre-copy and post- copy approaches. Proper classification of existing works provides the clear vision to understand the open issues in the field of virtual machine live migration. These works are broadly categories into pre-copy and post-copy based virtual machine memory migration. Both pre-copy and post-copy approaches are counted as a key technique for the improvement of virtual machine live migration process, but the pre-copy approach is widely adopted by several literature works as well as by several virtualization platforms. For better understanding all the techniques are summerised in Table 1.

4.1. Pre- Copy based approaches

With the goal of higher performance Clark [6] and Nelson [9] uses pre-copy algorithm with VMotion and Xen as a virtualization platform respectively. Clark [6] propose the improved live migration approach with idea that initially all pages are considered as dirty and transferred up to certain threshold level or maximum number of iterations. After this point pages are considered as clean and only those pages are transferred which are dirtied during migration. In this way total number of pages transferred is reduced up to some extent. Nelson [9] also shows the downtime for migration of hundreds of virtual machine concurrently with variety of workloads.

Memory ballooning, Dynamic rate limiting and Rapid page dirtying are some of the popular techniques mentioned in pre-copy approach. Memory ballooning is mentioned by Waldspurger [10] and Bradford [11], which eliminates unused memory and transfer only used memory in order to save the pre-copy time. Clark [6] also mentions some approaches like Dynamic Rate Limiting i.e. which dynamically adopt the bandwidth limit during each pre-copy rounds, and Rapid Page Dirtying i.e. point out those pages which are updated extremely frequently, and transfer it at last round of pre-copy iteration. All the techniques improve the pre-copy approach up to certain extent and open a new way for the improvement of virtual machine live migration process. There are many more solutions are available but for better understanding they are categories as:

- Compression based
- Checkpointing based
- Other approaches

4.1.1. Compression based

The key idea behind the approaches in this category is to improve the performance by reducing the parameters like the total migration time, downtime, total transfer time and number of iterations, by compressing the memory pages with the minimum overhead.

Jin [12] presents an adaptive memory compression (MECOM) approach for virtual machine migration. Instead of simple compression, it presents an adaptive zero-aware memory compression algorithm based on the memory page characteristics. By using appropriate compression algorithm based on the memory page characteristics gives lossless compression with small overhead and reduces the performance parameters considerably as compared with Xen hypervisor.

The another approach is delta compression presents by the Svard [13], in which it transfer only the difference between two versions of memory pages instead of full page and then further compress it with binary run length encoding (RLE). It is suitable for the migration of large and heavy loaded virtual machines with a shorten migration and downtime. It is implemented with the modified KVM hypervisor.

The next work in this category is proposed by the Ma [14] which transfer only a non-empty memory pages in a compressed form (using RLE). It greatly increases the performance as compared to Xen with small overhead.

4.1.2. Checkpointing based

Checkpointing based approaches focuses on the fault management during the migration, while maintaining the performance. Checkpointing is a method of storing volatile state of virtual machine for the later restoration, at the time of recovery process from the failure. Traditionally checkpointing mechanism is done by stopping the virtual machine which leads to the higher service delay and also consumes lots of memory for storing the virtual machine state. These approaches work towards the efficient checkpointing in terms of service delay and storage with the migration process, some of them are discussed here.

Siripoonya [15] presents a live checkpointing of virtual machine by introducing the checkpointing thread. It parallerly perform checkpointing task with the running virtual machine by periodically interrupting the hypervisor with minimal service delay. It also assists in migration process, to incrementally copying the dirty pages. It is implemented in KVM hypervisor with the minimum checpointing overhead.

Park [16] presents an efficient and fast virtual machine chekpointing mechanism, by storing only a subset of non-redundant part of virtual machine memory pages. It is achieved by transparently intercepting the I/O request, and maintain updated mapping of memory pages. It saves only those memory pages which are not available on disk,

Table 1. Summarised view of various live migration techniques

S.No.	Migration Techniques	References	Techniques used	Platform used	Overhead
1.	Compression Based	[12]	CBC (Zero-aware characteristics based compression)	XEN	Due to Compression/Decompression process.
		[13]	Delta Compression with RLE	KVM	
		[14]	Memory exploration and encoding (RLE)	XEN	
2.	Checkpointing Based	[15]	Multiple thread	KVM	Due to saving of VM image
		[16]	Intercepting I/O request	XEN	
		[17]	CR/TR Motion	XEN	
3.	Other	[18]	CPU Scheduling	XEN	Application performance degradation
		[19]	TPM (Three Phase Migration)	XEN	Increased TMT and DT
		[20]	Reuse Distance	XEN	Due to calcultion of reuce distance
		[21]	Regression Method	XEN	Due to prediction process
4.	Post-Copy based	[22]	Adaptive pre-paging and dynamic self ballooning	XEN	High Downtime
		[23]	Reactive VM consolidation system	Own Prototype	Due to analysis of resource usage

which significantly reduce the storage space and time for checkpointing process. It is implemented in Xen and shows the considerable reduction in storage and time with respect to Xen.

Next approach in this category presents by Liu [17], which provides fast and transparent virtual migration for both LAN and WAN environment by using checkpointing/replay and trace/replay mechanism. With minimal overhead, a trace daemon continuously logs non-deterministic events of virtual machines and stored at the source host. Stored trace file is used to synchronize the migrated virtual machines execution state, by iteratively transferred to the target host. It greatly reduces downtime and network bandwidth consumption as compared to Xen.

4.1.3. Other Approaches

Several different approaches contribute in this category towards the goal of improvement of virtual machine live migration. Some effective solutions are introduced here.

Jin [18] presents an optimized virtual machine live migration approach by proper scheduling of virtual CPU. It controls the memory dirty rate through adjusting the virtual CPU working frequency. With the controlled dirty rate, it shows the improvement on the performance of migration process as compared to basic pre-copy approach. It works well when network bandwidth is low and improves the migration liveness at the cost of application performance.

Luo [19] presents the mechanism for a whole system migration including processor state, device state, memory and storage. It is three-phase incremental migration approach, in which it gradually transfer the memory and disk content through the following three phases i.e. pre-copy, freeze and copy, and post copy. With maintaining the disk data integrity and consistency, it also reduces the migration time and downtime as compared to Xen migration process.

Another different approach based on one of the cache replacement policy is present by Alamadari [20]. It analyzes the reuse pattern of memory pages and calculates the distance between the same page reuse again. The logic behind this is pages with lower distance are the frequently updated pages and not a correct choice for transfer. In this way at every iteration pages are transferred according to reuse distance and it considerably reduce the total page transferred which leads to the improvement in migration time and downtime.

Virtual machine migration process itself consumes resources which effects on the performance of applications and the resource availability for the migrating virtual machine as well as for other concurrent virtual machines. Wu et al [21] builds a performance model for virtual machine live migration on Xen, to show the impact of resources allocation on the migration time. Based on the regression method, performance model predict a virtual machine migration time on the basis of its applications behaviour and the resources available for the migration.

4.2. Post Copy Approaches

Based on post copy approach small number of solutions is available, some of them are discussed here.

Hines [22] presents an approach which uses Adaptive pre-paging and Dynamic Self Ballooning (DSB) mechanism, to reduce the migration time, total page transferred and network overhead. Adaptive pre-paging eliminates all duplicate pages before the transmission and Dynamic Self Ballooning (DSB) eliminates the free memory page for transfer. In this way it significantly reduced the total page transferred and speeds up migration with negligible performance degradation.

Hirofuchi [23] presents an advanced virtual machine dynamic consolidation system by using virtual machine migration process. With response to every changing resources usage (like overload condition), virtual machine locations are reactively optimized. This approach shows the efficient resource utilization with the use of post copy live migration approach.

All these solutions address the need and applicability of virtual machine live migration, and open the path to work towards the performance and implementation challenges.

5. DISCUSSION AND CONCLUSION

Migrating the virtual machines with zero service interruption is known as virtual machine live migration mechanism and it is counted as an important tool for cloud resource managers with the benefits of load balancing, resource utilization, higher availability and power management. On the other side with the growth of data-intensive (like, multimedia content driven) applications, size of virtual machines increased drastically. Migration of these large data-intensive virtual machines imposed a large overhead, which leads to the application service degradation. There are various approaches are available to increase the virtual machine migration performance with the minimum overhead.

As a conclusion, after reviewing some of the most relevant solutions for virtual machine live migration, found that low bandwidth, impact on application performance are still a challenge need a proper attention . To handle these issues are considered as a future task in the field of virtual machine live migration.

REFERENCES

[1] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *Communications Magazine, IEEE*, vol. 50, no. 9, pp. 34–40, 2012.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[3] "VMware Incorporation," http://www.vmware.com.

[4] "KVM," http://www.linux-kvm.org.

[5] "Microsoft Corporation," http://www.microsoft.com/en-us/server-cloud/hyper-v-server/.

[6] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.

[7] A. Strunk, "Costs of virtual machine live migration: A survey," in *Services (SERVICES), 2012 IEEE Eighth World Congress on*. IEEE, 2012, pp. 323–329.

[8] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Cloud Computing*. Springer, 2009, pp. 254–265.

[9] M. Nelson, B.-H. Lim, G. Hutchins *et al.*, "Fast transparent migration for virtual machines." in *USENIX Annual Technical Conference, General Track*, 2005, pp. 391–394.

[10] C. A. Waldspurger, "Memory resource management in vmware esx server," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 181–194, 2002.

[11] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schiöberg, "Live wide-area migration of virtual machines including local persistent state," in *Proceedings of the 3rd international conference on Virtual execution environments*. ACM, 2007, pp. 169–179.

[12] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live virtual machine migration with adaptive, memory compression," in *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*. IEEE, 2009, pp. 1–10.

[13] P. Svard, J. Tordsson, B. Hudzia, and E. Elmroth, "High performance live migration through dynamic page transfer reordering and compression," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 542–548.

[14] Y. Ma, H. Wang, J. Dong, Y. Li, and S. Cheng, "Me2: Efficient live migration of virtual machine with memory exploration and encoding," in *Cluster Computing (CLUSTER), 2012 IEEE International Conference on*. IEEE, 2012, pp. 610–613.

[15] V. Siripoonya and K. Chanchio, "Thread-based live checkpointing of virtual machines," in *Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*. IEEE, 2011, pp. 155–162.

[16] E. Park, B. Egger, and J. Lee, "Fast and space-efficient virtual machine checkpointing," in *ACM SIGPLAN Notices*, vol. 46, no. 7. ACM, 2011, pp. 75–86.

[17] H. Liu, H. Jin, X. Liao, C. Yu, and C.-Z. Xu, "Live virtual machine migration via asynchronous replication and state synchronization," *parallel and distributed Systems, IEEE Transactions on*, vol. 22, no. 12, pp. 1986–1999, 2011.

[18] H. Jin, W. Gao, S. Wu, X. Shi, X. Wu, and F. Zhou, "Optimizing the live migration of virtual machine by cpu scheduling," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1088–1096, 2011.

[19] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machines using block-bitmap," in *Cluster Computing, 2008 IEEE International Conference on*. IEEE, 2008, pp. 99–106.

[20] J. Alamdari and K. Zamanifar, "A reuse distance based precopy approach to improve live migration of virtual machine," in *Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on*. IEEE, 2012, pp. 551–556.

[21] Y. Wu and M. Zhao, "Performance modeling of virtual machine live migration," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 492–499.

[22] M. R. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*. ACM, 2009, pp. 51–60.

[23] T. Hirofuchi, H. Nakada, S. Itoh, and S. Sekiguchi, "Reactive consolidation of virtual machines enabled by postcopy live migration," in *Proceedings of the 5th international workshop on Virtualization technologies in distributed computing*. ACM, 2011, pp. 11–18.

BIOGRAPHY OF AUTHORS

Sangeeta Sharma received her Bachelor of Engineering (BE) degree under the department of Information Technology (IT) from RGPV Technical University, Bhopal, India in 2006, and Master of Technology (M.Tech.) degree under the department of Computer Science and Engineering from Maulana Azad National Institute of Technology, Bhopal, India in 2011. She is currently working towards her PhD degree under same department at Maulana Azad National Institute of Technology,Bhopal, India. Her main research interest includes virtualization technology for cloud computing.

Meenu Chawla is Professor in the Department of Computer Science and Engineering, at Maulana Azad National Institute of Technology, Bhopal, India. Her current research interests include Adhoc network, Wireless network, Cloud Computing, etc. She has authored and published over 50 research Papers.