

Classification of Virtualization Environment for Cloud Computing

A. Souvik Pal^{1*}, B. Prasant Kumar Pattnaik²

^{1,2}*School of Computer Engineering, KIIT University, Bhubaneswar, 751024, India*

¹*Souvikpal22@gmail.com, ²patnaikprasantfcs@kiit.ac.in

Abstract

Cloud Computing is a relatively new field gaining more popularity day by day for ramified applications among the Internet users. Virtualization plays a significant role for managing and coordinating the access from the resource pool to multiple virtual machines on which multiple heterogeneous applications are running. Various virtualization methodologies are of significant importance because it helps to overcome the complex workloads, frequent application patching and updating, and multiple software architecture. Although a lot of research and study has been conducted on virtualization, a range of issues involved have mostly been presented in isolation of each other. Therefore, we have made an attempt to present a comprehensive survey study of different aspects of virtualization. We present our classification of virtualization methodologies and their brief explanation, based on their working principle and underlying features.

Keywords: Cloud Computing, Virtualization, Virtual Machine, Hypervisor.

1. Introduction

Cloud computing or Internet computing is used for enabling convenient, on-demand network access to a networks, servers, mass storage and application specific services with minimal effort to both service provider and end user [14]. For simplicity, A Cloud itself an infrastructure or framework that comprises a pool of physical computing resources i.e. a set of hardware, processors, memory, storage, networks and bandwidth, which can be organized on Demand into services that can grow or shrink in real-time scenario [15]. In surge of demand of internet and its immense usage all over the globe, Computing has moved from the traditional computing to distributed high performance computing say distributing computing, subsequently Grid Computing and then computing through clouds.

1.1 Need for Virtualization

A virtualization environment that enables the configuration of systems (i.e. compute power, bandwidth and storage) as well as helps the creation of individual virtual machines, are the key features of cloud computing. Virtualization provides a platform with complex IT resources in a scalable manner (efficiently growing), which is ideal for delivering services. At a fundamental level, virtualization technology enables the abstraction or decoupling of the application payload from the underlying physical resources [16]; the Physical resources can be changed or transformed into virtual or logical resources on-demand which is sometimes known as Provisioning. However, In traditional approach, there are mixed hardware environment, multiple management tools, frequent application patching and updating, complex workloads and multiple software architecture. But comparatively in cloud data center far better approach like homogeneous environment, standardize management tools, minimal application patching and

updating, simple workloads and single standard software architecture [17].

1.2 Characteristics of Virtualization

The cloud computing virtualization environment has the following characteristics discussed below:

1.2.1 Consolidation

Virtualization eliminates the need of a dedicated single system to one application and multiple OS can run in the same server. Both old and advanced version of OS may capable of deploying in the same platform without purchasing additional hardware and new required applications may be run simultaneously on their respective OS.

1.2.2 Easier development flexibility

Application developers may able to run and test their applications and programs in heterogeneous OS environments on the same virtualized machine. It facilitates the virtual machines to host heterogeneous OS. Isolation of different applications in their respective virtual partition also helps the developers.

1.2.3 Migration and cloning

virtual machine can be moved from one site to another to balance the workload. As the result of migration, users can access updated hardware as well as make recovery from hardware failure. Cloned virtual machines are easy to deploy in the local sites as well as remote sites.

1.2.4 Stability and security

In a virtualized atmosphere, host operating systems are hosting different types of multiple guest operating systems containing multiple applications. Each virtual machine is isolated from each other and they are not at all interfering into the other's work which in turn helps the security and stability aspect.

1.2.5 Paravirtualization

Paravirtualization is one of the important aspects of virtualization. In a virtual machine, guest OS can run on the host OS with or without modifications. If any changes or modifications are made to the operating system to be familiar with the Virtual Machine Manager (VMM), this process is said to be “paravirtualized”.

2. Classification

In this paper, we have classified virtualization environment into six categories based on Scheduling-based, Load distribution-based, Energy-aware-based, Operational-based, Distribution pattern-based and Transactional-based as shown in the figure [1].

We assume the host physical machines as a set of physical resources.

$S = \{\text{CPU cores, Memory, Storage, I/O, Networking}\}$ and the pool of physical resources, denoted by P , is the sum of all set of physical resources.

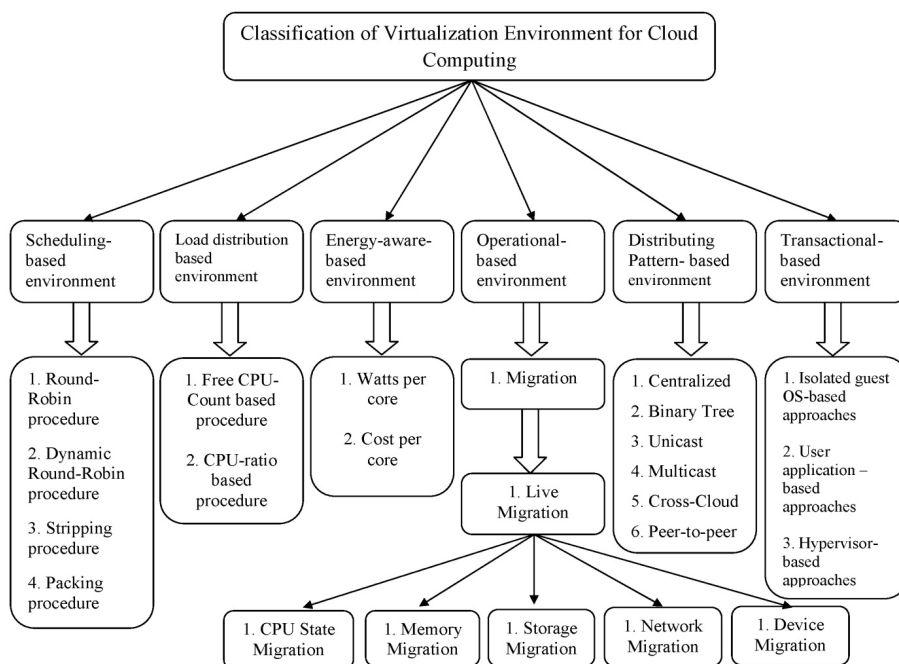
$$P = S_1 + S_2 + \dots + S_n = \sum S_i$$

Let us consider the resource pool set P into two subsets:

$$P = S_j^1 + S_k^2 \quad (j \neq k) \quad (j, k \text{ indicates natural number}).$$

= Set of physical machines having available resources to host VMs + Set of the remaining physical machines not having available resources to host VMs.

Fig.1. Schematic Diagram of Classification of Virtualization Environment



2.1 Scheduling-based Environment

Here we have classified scheduling environment into four sub-categories. These are as follows:

2.1.1 Round-Robin Procedure

The Round Robin scheduling process broadens the VMs across the pool of host resources as evenly as possible [1]. Eucalyptus cloud platform currently uses this as default scheduling policy [2]. Here we describe the procedure as below:

Step1: for each new VMs, it iterates sequentially until it is able to find an available resources (physical machine that is capable of hosting VMs) from the resource pool P .

Step1 (a): If found, then matching is done between physical machine and VM.

Step2: For next VM, the policy iterates sequentially through resource pool P from the previous point where it left off the last iteration and choose the nearest host that can serve the VM.

Step 3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.1.2 Dynamic Round-Robin Procedure

This method is an extension of Round-Robin which includes two considerations [3]:

Consideration1: One physical machine can host multiple virtual machines. If anyone these VMs has finished its work and the remaining others are still working on the same physical machine. Then no more VMs are getting hosted on this physical machine being in “retiring” state that means the physical machine can be shut down when the remaining VMs complete their execution.

Consideration2: when a physical machine does not wait for the VMs to finish and goes to the “retired” state for a long time, then it will be forcefully migrated to the other active physical machines and will shut down after the completion of the migration process.

So, this scheduling technique will consume less power than the Round-Robin procedure.

2.1.3 Stripping Procedure

The Stripping Scheduling policy broadens the VMs across the pool of host resources as many as possible [1]. For example, OpenNabula cloud platform currently uses this scheduling policy [4]. The procedure of this sort of stripping scheduling is given below:

Step1: For each new VM, it first discards the set S_k^2 .

Step2: From the set S_j^1 , it finds the physical machine hosting least number of VMs.

Step2 (a): if found, then matching is done between physical machine and VM.

Step3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.1.4 Packing Procedure

The Packing policy spread the VMs across the pool of host resources as few as possible [1]. OpenNabula cloud platform currently uses this scheduling policy and implemented as Greedy policy option in Eucalyptus [2, 4]. Here the technique as follows:

Step1: For each new VM, it first discards the set S_k^2 .

Step2: From the set S_j^1 , it finds the physical machine hosting maximum number of VMs.

Step2 (a): if found, then matching is done between physical machine and VM.

Step3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.2 Energy Aware-based Environment

In this environment, energy-efficiency issues are considered and the procedures used in each category mentioned below:

2.2.1 Watts per core

Watt per core policy wants to seek the host taking the minimum additional wattage per core, sinking overall power consumption. It is assumed that no additional power will be consumed during shut down or hibernating mode. The procedure as follows:

Step1: For each new VM, it first discards the set S_k^2 .

Step2: From the set S_j^1 , it finds the physical machine taking the minimum additional wattage per CPU core based on each physical machine's power supply.

Step2 (a): if found, then matching is done between physical machine and VM.

Step3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.2.2 Cost per core

The Cost per core policy is energy-aware policy and also minimizes the cost estimation by seeking the host that would capture least additional cost per core. Assumption is same as in Watt per core policy. Here we describe the method:

Step1: For each new VM, it first discards the set S_k^2 .

Step2: From the set S_j^1 , it finds the physical machine taking the minimum additional cost per CPU core based on each physical machine's power supply and electricity cost.

Step2 (a): if found, then matching is done between physical machine and VM.

Step3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.3 Load-balancing-based Environment

In the virtualization environment, the load-balancer has the responsibility to reduce the load overhead and that's why we made a classification of load balancing approaches.

Here we consider CPU core set as Q which is subdivided into allocated CPU core subset denoted as Q_1 and free CPU core subset denoted as Q_2 .

2.3.1 Free-CPU-Count based Procedure

This Load balancing policy is the Free-CPU-Count policy & wants to minimize the CPU load on the hosts [1]. OpenNabula cloud platform currently uses this policy as the Load Aware policy [4]. Here we describe the procedure:

Step1: For each new VM, it first discards the set S_k^2 .

Step2: From the set S_j^1 , it finds the physical machine having maximum number of free CPU cores from set Q .

Step2 (a): if found, then matching is done between physical machine and VM.

Step3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.3.2 Ratio-based Load Balancing Procedure

It is an enhanced version of Count-based Load Balancing technique which is also wants to minimize the CPU load on the hosts [1]. The procedure is as follows:

Step1: For each new VM, it first discards the set S_k^2 .

Step2: From the set S_j^1 , it finds the physical machine having maximum ratio of Q_2 and Q_1 ($Q_2/Q_1 > 1$) from set Q .

Step2 (a): if found, then matching is done between physical machine and VM.

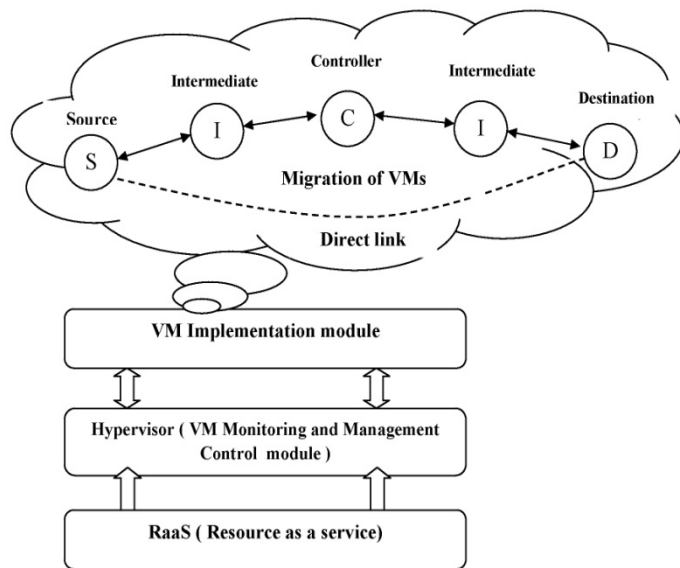
Step3: Now go to Step1 and iterates the whole process until all VMs are allocated.

2.4 Operational-based Environment

In operational-based procedure, we explain the general movement of the virtual machines from the local sites to the remote sites in the same cloud or between the cross clouds.

2.4.1 Migration

Virtual Machine migration is the process of transferring a VM from one host physical machine to another host machine which is either currently running or will be running or may be booted up after placing the new VMs. Migration is initiated due to 1) Lack of resources in the source node or local site or 2) When remote node is running VMs on behalf of local node due to dynamic unavailability of resources or 3) For minimizing the number of host machines that are running remote sites (less energy consumption) or 4) For maximizing allocation of VMs among the local machines rather than remote machines. We will discuss the migration process as depicted in the figure [2].

Fig.2. Schematic Diagram of Migration Process.

In this context, Resource as a Service (RaaS) [5] is a physical layer comprising of a pool of physical Resources i.e. Servers, networks, storage, and data center space, which provides all the resources to the VMs. Hypervisor layer, is the management layer which provides overall management, such as decision making regarding where to deploy the VMs, admission control, resource control, usage of accounting etc. The implementation layer provides the hosting environment for VMs. Migration of a VM needs the coordination in the transfer of source and Destination (D) host machines and their states [6]. Main migration algorithm, Migration request handler, Initiate transfer handler and Forced migration handler are there to handle the migration process [7]. The overall migration steps are as follows:

Step1: The Controller and the intermediate nodes (I) send the migration request to the possible future destinations. D-nodes can be in remote sites and the request can be forwarded to another remote sites depending upon resource availability.

Step2: The receiver may accept or reject depending on the account capabilities and usage of accounts.

Step2 (a): If all the business rules and accounting policy permits the request, then “initiation transfer” reply message will be sent to the requesting nodes through I-nodes (Intermediate-node).

Step3: Now getting the reply message, the transfer operation from the S-node to the D-node via I-nodes and VMs on the source site will be migrated to the destination sites. Here transfer initiation is done by the C-node to D-node and the transfer is verified through token [7] by the controller with the help of S-node.

In migration, monitoring is essentially needed to ensure that VMs obtaining the capacity stipulated in the Service-Level-Agreement and to get the data for accounting of the resources, which has been used by the service providers [7]. The monitoring

issues may be the amount of RAM or the network bandwidth or the number of currently logged-in users. *Grid Monitoring Architecture (GMA)* has been defined by the Global grid Forum [8].

2.4.2 Live Migration

In the surge of rapid usage of virtualization, migration procedure has been enhanced Due to the advantages of live migration, say server consolidation, and resource isolation [9]. Live migration of virtual machines [10, 11] is a technique in which the virtual machine seems to be active and gives responses to end-users all the time during migration process. Live migration facilitates energy-efficiency, online maintenance, and load balancing [12]. Live migration is sometimes known as called real-time or hot migration in cloud computing environment. While the virtual machine is running on the source node or one host server and during live migration, the virtual machine is moved to the target node or another host server without interrupting any active network connections or without any visible effect from the user’s point of view. Live migration helps to optimize the efficient utilization of available CPU resources. Different live migration algorithms are described in [18, 19, 20, and 21]. Here we will discuss few processes of live migration.

CPU state Migration: Migration of CPU state is concerned in the context with process migration. While migrating the CPU state, the process control block (PCB), the number of cores, processor speed, and required process-specific memory will all transfer from the source host to the destination host.

Memory Migration: While concerned with the memory migration, all the pages are transferred from the source host A to the destination host B. In this memory migration, pre-copy [22] migration is concerned, where all pages are iteratively copied from source host to destination host during the first round. Assume, there are n numbers of rounds; and subsequent rounds up to n copy only the pages those are getting dirtied (dirty pages) during the previous transfer round (indicated by dirty bitmap). Here we have to consider that every virtual machine has some set of pages which update very frequently and this phenomenon leads to the performance degradation of the pre-copy. For every iteration, a “dirtying rate” is calculated depending upon the length of the pages and the number of pages being dirtied. Christopher Clark et al. [10] have bounded the number of iterations of pre-copying, based on the writable working set (WWS) according to the behavior of typical server workloads.

Storage Migration: To maintain VM migration, the system has to provide each VM with a location-independent, consistent vision of the file system and that is accessible on all hosts. Each VM are using its own virtual disk, to which the corresponding file system is mapped and transfers the contents of the virtual disk to the source machine. We depend on the storage area networks (SAN) or NAS to permit us to migrate connections to the storage devices. This phenomenon allows us to migrate a disk by

reconnecting to the disk on the target machine.

Network Migration: To locate the remote systems and to communicate with a virtual machine, a virtual IP address (known to other units) has been assigned to each virtual machine. This IP address and the IP address of the currently hosting machine of the VM are distinct. Each virtual machine can have its individual unique virtual MAC address. The hypervisor maps between the virtual IP & the MAC addresses to their corresponding virtual machines. It is to be considered that all the virtual machines should be in the same IP subnet. While migrating to the target node, an ARP broadcast has to be sent to the network declaring that the IP address has moved to a new MAC address (physical location) and TCP connections survive the migration.

Device Migration: Hypervisor makes the physical hardware virtualized and represents each VM with a standard set of virtual devices. Well-known physical hardware are completely emulated by these virtual devices in an effective way and these devices translate the VM requests to the system hardware. Device migration requires the dependency on host-specific devices as it may be difficult to migrate due to an awkward provisional state (for example, CD recording device while recording) or unavailability for migration.

2.5 Distributed Pattern-based Environment

Previously we have explained different types of scheduling policy, energy efficient technique, load balancing and migration procedure. But there must be a virtual machine deployment pattern which helps the distribution of the virtual machines making it more efficient in faster response time, minimizing communication latency, avoiding congestion and dynamic updation. The classification is as follows:

2.5.1 Centralized distribution

Centralized distribution is the traditional approach. It can be implemented in a simple way so that the users and the administrators can use easily. The VM images are stored in the central NFS (Network File Server) server and the client nodes retrieve copies of VMs from the central node on demand. This type of multiple point-to-point transfer creates an inconsistent situation when a large number of clients want to access multi-terabyte file. So, client-transfer should be synchronized.

2.5.2 Balanced Binary tree Distribution

Balanced Binary tree based distribution is used for reducing the overhead of network congestion and allowing parallel transfers. Here all the computing nodes are set in balanced binary tree pattern making the source node as the root node. The root node always carries the VM images and distributes the VM images from the parent node to child node. Newly arrived child node can easily get the data from its parent node. Thus data and VM images are flowing through the entire binary tree. But when a node is crashed, then all the child node of this very node will get stopped. So time-out and re-transmission strategy make it resolved.

2.5.3 Unicast distribution

Unicast distribution distributes the VM images in a sequential order to the destination nodes even in remote sites. But it is so time consuming and faces the network congestion.

2.5.4 Multicast

Multicast is efficient way for distribution of VMs. In multicast, the information or data is transmitted to a required group of destination nodes in a single transmission. Packets are sent in a group so that it can minimize the CPU load but increase the packet loss probability. It results best in Local Area Network (LAN).

2.5.5 Distribution between cross clouds

Multicast does well in LAN. But sometimes transfer is required beyond the LAN. Considering an example, more than one private, physically distinguishable desktop cloud is sharing the same data and information and they are using multicast distribution method. But transferring the data is forbidden by their network policy. To overcome this type of constraint, peer-to-peer or balanced binary tree distribution mechanisms is used over the common network linking those clouds.

2.5.6 Peer-to-Peer distribution

Peer-to-Peer is a decentralized approach. There is no centralized server, every node in the system works as a server or client. Every VM node may acts as sender or receiver or sender & receiver both. It is possible to multiple transfers of different files to the same node. Bit Torrent Protocol is the example of peer-to-peer distribution [13].

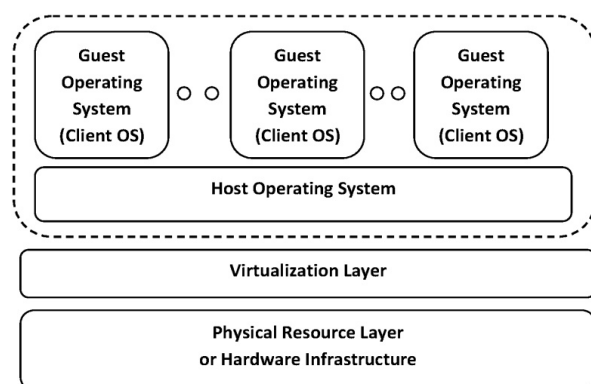
2.6 Transactional-based Environment

In this category, we have explained different architecture-based virtualizations and deployment of different operating systems & new applications.

2.6.1 Isolated Guest Operating System-based virtualization approaches

In this approach, host OS is running on the hardware infrastructure. It supports multiple guest virtualized OS on the same physical server and it is capable of maintaining isolation of different guest OS shown in the figure [3]. All the operating systems are using the same kernel and hardware infrastructure. The host OS controls the guest OS.

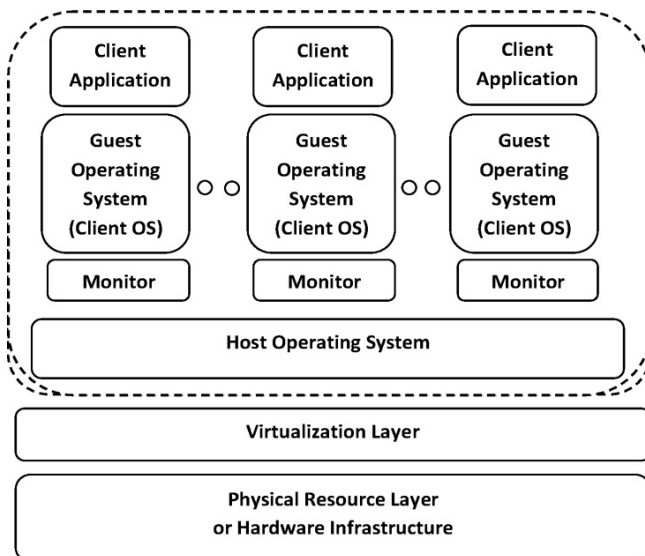
Fig.3. A Scenario of Operating System-based Virtualization.



2.6.2 User application-based virtualization approaches

In this approach, virtualization is done according to the users' on demand requirement and it is hosted on the upper of the host OS as shown in the figure [4]. On getting request, emulation of VM containing its own guest operating system & related applications is carried out by this virtualization method so that users can get their specific on-demand service from the emulated VMs.

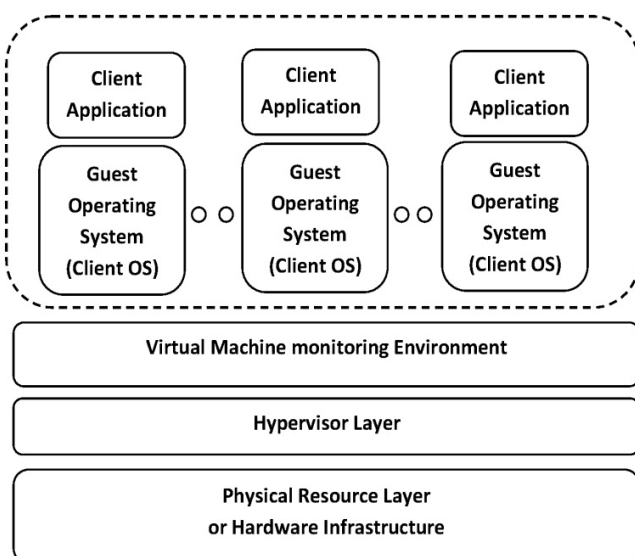
Fig.4. A Scenario of Application-based Virtualization



2.6.3 Hypervisor-based virtualization approaches

Hypervisor is a mainframe operating system, which allows other operating systems to run on the same system concurrently. And its monitoring system monitors the accesses of Virtual Machines. Hypervisor is accessible in the booting time of the system to regulate the allocation of hardware infrastructure to the multiple VMs from the resource layer. The architecture model is shown in the figure[5].

Fig.5. A Scenario of hypervisor-based Virtualization



3. Summary and Future Directions

Due to the large number of virtualization technique reviewed in this paper, it is difficult to compare all their quantitative performance. Here we present a comprehensive review study on different aspects virtualization procedure and the inter relationship among them. We have shown here what should be considerations while doing virtualization.

In this paper, different types of scheduling technique and load balancing technique are explained and these procedures are implemented in most of the cases in management node and load balancer. Load balancer is responsible for reducing the complex load overhead. Here, we have described two types of energy-efficient policies which can be quickly, easily and trivially implemented in existing cloud platforms, without any necessity for external systems and also provides the cloud administrators with a convenient, suitable and simple way of improving power efficiency and minimizing energy costs across their data centers. Combining these aspect, we can suppose that better load balancing algorithm and scheduling technique are to be introduced which should be more powerful & efficient and also less energy-consuming. While considering different virtual machine distribution technique, no matter if it is a public or private or community cloud connected via LAN or various remote cloud computing sites connected through internet. Among these distributing techniques, multicast seems to be most efficient because it minimizes CPU load, reduces power-consumption and effectively cost-efficient. Virtual machine image distribution techniques should be in an efficient way so that while migration comes into the picture, source & destination nodes or local or remote sites should be commented in a particular pattern. Live migration is dynamic abstraction layer which provides energy-efficiency, online maintenance, and load balancing. That's why, live migration is hot topic in the field of cloud computing virtualization. Different virtualized approaches enlighten that what should be architecture based on application or OS or hypervisor on-demand basis. In hypervisor-based security constraints are there. As hypervisor controls all the accesses of VMs and monitors the environment, so failure of the hypervisor or crashing of hypervisor or attack on hypervisor by the intruders may lead to performance degradation. Hence, security aspects are to be considered while making virtualization.

4. Conclusion

Cloud computing is one of the most emerging technology in the Internet. In the context of cloud computing, virtualization is used computer resources to imitate other computer resources or whole computers. This survey study has represented a wide range of overview of the research work in the field of Cloud Computing with respect of the virtualization methods. We have discussed many procedures, schemes and mentioned their silent features. Particularly, we have looked forward for the issues in scheduling, load distribution, energy efficiency, distribution pattern and also

transactional approaches. We have discussed the basic principles of the various virtualization techniques, through which a general classified virtualization concept has been represented. This survey is done on the basis of virtualization environment in cloud computing, which may help the researchers to expand the concepts of virtualization and this may lead to practical implementation of virtualization in industry level in a private cloud.

5. References

1. Ryan Jansen and Paul R. Brenner, "Energy Efficient Virtual Machine Allocation in the Cloud", in Proceedings of 2011 International Green Computing Conference and Workshops (IGCC), 2011, pp.1~8.
2. Souvik pal, Suneeta Mohanty, Dr. P.K. Pattnaik, and Dr. G.B. Mund, "A Virtualization Model for Cloud Computing", in the Proceedings of International Conference on Advances in Computer Science, 2012, pp. 10~16.
3. Ching-Chi Lin, Pangfeng Liu, and Jan-Jan Wu, "Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems" in the Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing, 2011, pp. 81~88.
4. I. Raicu, Y. Zhao, I. Foster, and A. Szalay, "Accelerating large-scale data exploration through data diffusion", in the Proceedings of the 2008 International Workshop on Data-Aware Distributed Computing, ACM, 2008, pp. 9-18.
5. Souvik Pal and P. K. Pattnaik, "Efficient architectural Framework of Cloud Computing", in "International Journal of Cloud Computing and Services Science (IJ-CLOSER)", Vol.1, No.2, June 2012, pp. 66~73
6. "CIM System Virtualization White Paper," in the Proceedings of Distributed Management Task Force-Informational, Nov. 2007, pp.1~33.
7. Eric Elmroth and Lars Larson, "Interfaces for placement, Migration, and Monitoring of Virtual machines in Federated clouds", in Proceedings of 2009 Eighth International Conference on Grid and Cooperative Computing", 2009, pp. 253-260.
8. B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski, and M. Swamy, "A Grid Monitoring Architecture," in Proceeding of Grid Working Draft-Informational, Aug. 2002. pp. 1-7.
9. K. Ye, X. Jiang, D. Ye, and D. Huang, "Two Optimization Mechanisms to Improve the Isolation Property of Server Consolidation in Virtualized Multi-core Server," in Proceedings of 12th IEEE International Conference on High Performance Computing and Communications, 2010, pp. 281~288.
10. C. Clark, K. Fraser, S. Hand, J. 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, 2005, pp. 286.
11. M. Nelson, B. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in Proceedings of the annual conference on USENIX Annual Technical Conference, 2005, pp. 25.
12. Kejiang Ye, Xiaohong Jiang, Dawei Huang, Jianhai Chen, and Bei Wang, "Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments", in Proceedings of 2011 IEEE 4th International Conference on Cloud Computing", 2011, pp. 267 - 274
13. Matthias Schmidt, Niels Fallenbeck, Matthew Smith, and Bernd Freisleben, "Efficient Distribution of Virtual Machines for Cloud", in the proceedings of Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference, 2010, pp. 567~574.
14. P. Mell et al, "NIST definition of cloud computing", vol. 15, October 2009.
15. V. Sarathy et al, "Next generation cloud computing architecture-enabling real-time dynamism for shared distributed physical infrastructure", "19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10)", Larissa, Greece, 28-30 June 2010, pp. 48-53.
16. Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopala, James Broberga, and Ivona Brandicc, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Volume 25, Issue 6, June 2009, pp. 599-616.
17. Huaglority Tianfield, "Cloud Computing Architectures", in the Proceedings of Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference, 2011, pp. 1394 - 1399.
18. 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 Proceedings of the IEEE International Conference on Cluster Computing, 2008, pp. 99~106.
19. H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in the Proceedings of the 18th ACM international symposium on High performance distributed computing, 2009, pp. 101~110.
20. H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live virtual machine migration with adaptive memory compression," in the Proceeding of IEEE International Conference on Cluster Computing, 2009, pp. 1~10.
21. M. 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, 2009, pp. 51~60.
22. Marvin M. Theimer, Keith A. Lantz, and David R. Cheriton, "Preemptable remote execution facilities for the V-system", in the Proceedings of the tenth ACM Symposium on Operating System Principles, 1985, pp. 2~12.