ISSN No. 0976-5697

DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.5098

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A NOVEL CLOUD SYSTEM STORAGE AND COMPUTING VIRTUALIZATION MANAGEMENT FRAMEWORK

L. Srinivasa Rao
Computer science and Engineering
Mother Teresa Institute of Science and Technology
Sathupally, India

Dr.I.Raviprakash Reddy
Dept of Information Technology
G Narayanamma Institute of Technology and Science,
Hyderabad, India

Abstract: With the tremendous growth in migration to cloud computing for traditional applications, the demand for virtualization is the new need for further research and enhancements. The high demand for virtualization in cloud computing is been addressed with the technology enhancement and in terms of virtual machines. Majorly the hardware component management of virtual machine is considerably reaching the pick of research with the recent advancements by multiple companies and open source researches like AlphaVM, Hyper-V, Integrity Virtual Machines, JPC (Virtual Machine), PowerVM, Sun xVM, VMware Workstation and z LPARs. Another most important component of virtualization and virtual machine image storage is been the bottleneck for the further advancement. The limitations are been addressed by clustering and storage area networking, however these existing solutions are no match for the industrial grade demand for the VM image storage requirements.Hence in this work we propose a cloud based virtual machine storage framework designed for large scale deployment satisfying the other needs for storage features like replication and measurements of performance improvements. This work also demonstrates the advancements in performance for virtualization management

Keywords: Virtual machine,Migration,Virtualization Bottleneck,Storage Architecture, Virtualization Management Framework

1. INTRODUCTION

The core of cloud computing for providing the scalable infrastructure is virtualization. Virtualization is used to manipulate the underlying computing and communication infrastructure in order to create multiple instances of the same resources to be dedicatedly allocates to various customers or customer applications [1]. The virtualization as a technique allows the service providers to create different types of computing environments to match the customer requirements on a same server. Virtualization allows the service providers to meet the need for creating the scalable and application depended environments. At the same time, the customer can also reduce the cost for maintaining computing and storage related infrastructure on side. The majority of the service providers deploy the use of virtual machines in order to separate actual hardware from the computing environments.

To understand the applicability of virtualization through virtual machines, we consider the fact that in case of private and public cloud the requirements for multiple systems, maximization of resources and reducing the cost for implementation is the prime important factor. In case of a private cloud the customer tend to rent the infrastructure from a vendor or in some cases choose to host the infrastructure from own premises. In case of the on premises hosting the same virtual machine can be used for variety of purposes and in case of off premises hosting the provider can use the same configuration for multiple cu0stomers. Hence In both the cases, the implementation of virtualization through virtual machines reduces the cost [2].

In the other side for public cloud environment, the user can choose multiple configurations for virtual machines to suite the requirements. Thus the cost implementation for managing separate and custom configuration of hardware reduces for the providers.

In order to understand the improvements required for virtualization, we understand the hardware components influences the performance of virtualization. The major components for virtualization are similar to the hardware resources required for generic computing. The infrastructure deals with the processor management, memory management, peripheral management, network management and storage management. The storage management for virtual machines plays the major role for performance enhancements. The detail study is been conducted in the later part of this work.

The storage of virtual machine image is the core of this research. Prior to the study of virtual machine image formats, we understand the detail configurations of virtual hard disk. The deployment of virtual hard disk allows the service providers, developers and users to replicate multiple environments on single hard disk.Moreover the recent advancement is research shows the benefits of applicability of virtual hard disk as robust to transport on multiple host server, provide snapshot backups and in time recovery, provide security and finally image based managements. The details of the virtual machine formats are also been considered in this work.

We also consider the virtual machine monitor software and understand the scope of enhancement in the same. Virtual machine management using software components are popular and demonstrated using many researches. The virtual machine monitor or VMM are the software components residing on the host server to manage and monitor the virtual machines. The software provides a platform for the server based host operating systems to host other computing environments virtually. The detail study of the VMMs is also conducted in this work.

The recent studies have demonstrated that the storage of virtual machine images are the biggest bottleneck of the advancement and reducing cost factors. We have understood the implications of the solutions in terms of deploying storage area networks. However the deployment of SAN in order to manage the storage distributions is not cost effective. Hence in this work, we try to propose a novel approach for virtual machine distributed image storage formats with efficient management and monitoring. The major outcomes of this work is a management platform for managing virtual machine replication and storage using peer to peer transfer protocol and study the improvements in performances [3].

The rest of the work is presented as in Section II we consider the enhancements produced with VM migration, in Section III we understand the bottlenecks of virtualization, in Section IV we consider the components to be taken under consideration for further research, in Section V we consider the VM Image formats, in Section VI we consider the VM Image Storage architecture, in Section VII we consider the parameters to be included for performance comparison, in Section VIII we propose the novel compute and storage virtualization management framework, we demonstrates the results in Section IX and in Section X we produce the conclusion and further scope of this work.

2. VM MIGRATION BENEFITS

The recent researches have demonstrated that the migration and adaptation of virtual Machine can enhance the performance of the client application and infrastructure cost can be reduced. Here in this work we consider the factors influencing the performance and productivity with the migration [4]:

A. Detailed Control:

The Virtual Machinescome with a reduced abstraction is the system level and allows the provider, customer and researchers to access more properties of the system. The access to computing environment data, system level codes, hardware utilization statistics, traces of the active application, failing and down timing component configurations and the guest operating system configuration parameters and the ability to control them independently helps to understand the performance perimeters [Table -1].

TABLE I
PARAMETERS FOR OPEN CONTROL

Type	**Name**	**Access Permission**	
		Traditional	**Virtual Machine**
Processing	CPU Type	Not Allowed	Allowed
	Allocation	Allowed	Allowed
	Priority	Allowed	Allowed
Memory	Size	Allowed	Allowed
	Buffer	Not Allowed	Allowed
Storage	Access IDE Bus	Not Allowed, Physical	Allowed, Logical
	Capture Mode	Not Allowed	Allowed
	Library Group	Allowed, Physical	Allowed, Logical
Network	IP Address	Allowed	Allowed
	MAC Address	Not Allowed	Allowed
	Internal Network	Partially Allowed	Allowed

B. Reduced Hardware Constraints:

The Virtual Machinescomes with a flexibility to change or alter the operating system and hardware components seamlessly. After the initial cost for setting up a virtual environment, the provider, customer and researchers are free to modify the computing system including the operating system, libraries, tools and other supporting patches without investing the full time needed for computing system change or upgrade [Table -2].

TABLE II
UP-GRADATION CONSTRAINTS

Type	**Name**	**Accessibility**	
		Traditional	**Virtual Machine**
Operating System	Version	Available	Available
Operating System	Version	Available	Available
	Interoperability	No Continuous Availablity	Available
	Patch	Available	Available
IDE	Patch	Available	Available
	Device Driver	No Continuous Availablity	Available
	Version	Available	Available
Configuration Time	-	Very High	Low

A. Replication Control:

The replication of the Virtual Machines using the snapshot feature allows the provider, customer and researchers to take timely and on demand backups of the virtual machine images. Thus the backups help to quickly reproduce the same computing environment without investing the complete setup time [Table -3]. Table-III

Type	**Accessibility**	
	Traditional	**Virtual Machine**
Windows Server	50 to 90 Mins	Just in Time
MAC Servers	40 to 60 Mins	Just in Time
Linux Servers	30 to 40 Mins	Just in Time

B. Availability:

The Virtual Machines are hosted by all service providers with similar configurations but with added advantages. Hence adopting to Virtual Machine computing is the best choice to avoid the lack of support and facility availability [Table – 4].

TABLE III
VM SUPPORT FOR CLOUD SERVICE PROVIDERS

	1	**2**	**3**	**4**	**5**
Windows Server	YES	YES	YES	YES	YES
MAC Servers	YES	YES	YES	YES	YES
Linux Servers	YES	YES	YES	YES	YES

1. Amazon, 2.Microsoft, 3.Google, 4. IBM,5.Private Cloud

C. Regular Updates

The application on Virtual Machines hosted on cloud is always liable for automatic and regular updates from the service provider without any extra cost. However in the other side, hosting the traditional system demands the cost and time implications for updates.

D. Cost Control

Due to the tremendous competition in the cloud service provider space, the drop of price for each virtualization component used in the virtual machine configuration is dropping with an increasing speed. Hence rather than up-gradation cost for traditional systems, the cloud based virtual machines are very much cost effective [Table 5].

TABLE V
COST DROP HISTORY FOR VIRTUAL MACHINES HOSTING
(APPROXIMATED VALUES)

	1	**2**	**3**	**4**	**5**
2012	$0.64	$0.70	$0.63	$0.61	$0.66
2013	$0.48	$0.45	$0.49	$0.47	$0.50
2014	$0.35	$0.39	$0.31	$0.30	$0.35
2015	$0.28	$0.26	$0.29	$0.26	$0.27

1. Amazon, 2.Microsoft, 3.Google, 4. IBM,5.Private Cloud

E. Collaborative Approach

The need for sharing the same testing and development environment is always a challenge in modern industrial requirements. Hence sharing the virtual machine image will certainly help in order to replicate the same development and testing environment timely and easily.

F. Manageable DataLoads

The recent researches have demonstrated that the closeness of the data and computational units can increase the speed of computation. However when the data is large and cannot be fetched in smaller amount of memory spaces, then the making the computational unit closer to the data is efficient solution. Hence hosting the virtual machines on the cloud with big data management features are quite efficient and directing towards the improvement in performance.

3. VIRTUALIZATION BOTTLENECKS

With the in detail understanding of benefits to be achieved by deploying virtual machines rather than the on-side traditional infrastructures, we also try to realise the bottlenecks for adopting cloud based virtual machines in larger scale.

Here we understand the bottlenecks for virtual machines adaptations:

A. *Access* Delay:

The industry of application development is diversified based on the need for availability, latency and accuracy. Few of the applications and customer demands for high availability with data accuracy and on the other side, few applications and customer will demand for more timely response.

In case of the locally hosted applications if the infrastructure is hosted on-premises, then the network delay can be reduced, which is caused by network or internet connection response time delay in case of cloud based virtual machine networks. Hence these set of customers will not prioritize the cloud based virtual machine infrastructure and will not migrate their applications.

B. Acceptability:

The next bottleneck we encounter in our studies is the general acceptability for customers and researchers. Few customers and researchers will prefer to have their own applications running on some on-side infrastructure rather than cloud hosted virtual machines for better control. This nature of the customers and researchers are reducing the migration speed.

C. Provisioning Deficiency:

In spite of getting hosted on cloud based virtual machines, some customers may demand to open the visibility of the underlying hardware infrastructure for better understanding of the application development where the properties depends on the physical infrastructure.

The cloud service providers are most not-likely to open the access of logs and parameters for the customers. Hence this problem is causing the bottleneck for the migration to the cloud based virtual machine infrastructure.

D. Interactivity Limitations:

This is also one of the limitations comes alongside with the migration of existing virtual machines to the cloud. Many of the virtual machine management software suites do not provide much control when migrated to the cloud. Thus this brings the barrier for the migration.

4. VIRTUALIZATION COMPONENTS AND VMMS

After the detail understanding of the benefits and causes of the virtualization, here we understand the hardware and software components of virtualization using virtual machines. These two components will help us to understand and propose the novel framework.

A. Hardware Components for Virtualization:

Virtualization using Virtual Machines enables to isolate the physical hardware components from the software stack and increases the productivity discussed in early part on this work. In the process of virtualization, we understand the standard framework resides on the physical hardware and the virtual machine replicates the hardware logically. In the virtual machine the operating system gets install and runs the applications [Figure – 1].

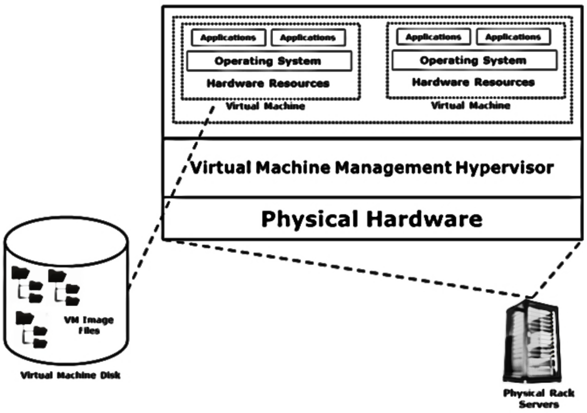

Figure.1 Hardware Components for Virtualization

Here we understand the solutions provided by multiple companies and by the independent searches, a very minimum scope is available to go for the enhancements. Thus we focus for the improvements possible in case Hypervisor architecture.

B. Software Components or Hypervisor for Virtualization:

In order to provide the enhancements for Hypervisor technology, here we understand the architecture.The Hypervisor is mainly responsible for controlling the virtual machines along with managing the networking components, file systems, virtual machine images, software stack, the replication and other software controller. The Hypervisor is divided into three major layers as control layer, network layer and interfacing layer between physical and logical networking layers [Figure – 2].

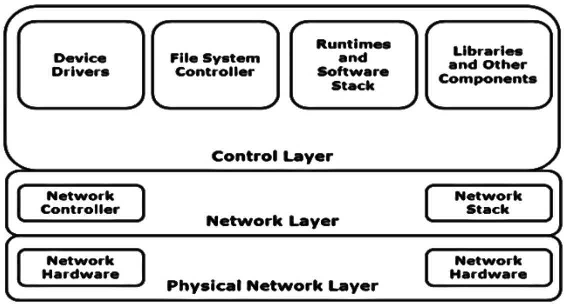

Figure.2Generic Architecture for Hypervisor

This understanding helps us to realize that the aggregation of multiple physical servers is still not reached the highest pick of performance and research. The aggregation of the physical server running virtual machines are still not been realized due to the complex and diversified nature of the server vendors and configurations [5] [6].

Hence we propose the novel approach for storage and computing virtualization management. The detail of the proposed framework is discussed in the later part of this work [7].

5. VM IMAGE FORMATS

The Virtual Machine Image is similar to the standard hard disk file. The partitioned properties are also similar to the physical hard disk containers [8] [9].

The virtual machine image or the virtual hard disk is very popular for the properties like moving the image file seamlessly over multiple host server systems, easy backups and recovery powered by snapshot mechanism, applicability of security patch and antivirus software stacks, replication control powered by guest to host and vice versa and finally the life cycle management controls.

Here also we understand the types or the formats of virtual machine images to be stored and compared [10] [11]:

A. Fixed Length Image File Format:

The Fixed Length Image File Format needs to be configured at the beginning of the virtual machine creation. Once created the complete file size allocation needs to assigned and once assigned the size cannot be reduced or increased.

This file format is limited in use as the growth of data and applications generating more data cannot be predefined.

B. Dynamic Length Image File:

On the other hand, Dynamic Length Image File is a variable length image format. The Dynamic Length Image File can be extended by size automatically when the data or the size of the software stacks need to be increased.

However the increment happens by the multiple of data block size, which is generally defined at 512MB in most of the systems.

The problem with Dynamic Length Image File is if the small amount changes need to be applied on parallel replicated systems, then the complete file needs to be replaced.

Hence due to the lack of delta change management properties, the portability is restricted in these files.

C. Dependent Image File:

Due to the problems of delta change management in previous file formats, the Dependent Image File manages the complete change management with a property called Undo Change. This property connects the modified file with the previous file as child file only with the change information.

Hence this file format is majorly accepted due to its portability for movements over multiple host servers.

However in above mentioned file formats, the complete isolation is employed from host server file systems for security reasons. Sometimes the need to share the files or locally generated server logs can be useful for managing applications better, thus this is a noted limitations in all the file formats.

D. Linked Image File:

The Linked Image File formats are majorly similar to the other file formats, except the property of connectivity between the physical hard drive and the logical hard drives.

This feature enables the developers and researchers to closely analyses and incorporates the server based file systems into the application and increases the efficiencies for some cases of application development like emulated storage or physically reading the network parameters through the applications.

6. VM IMAGE STORAGE ARCHITECTURE

Here we understand the internet storage architectures used for storing the virtual machine image formats. We encounter the following types of storage formats [12]:

A. VDI

The Virtual Disk Image or VDI are generally created and stored using the four components as header description block, image mapper block, image alignment block and finally the data block for 1 MB each.

B. VMDK

The Virtual Machine Disk is called VMDK and stared using the open file storage standards and majorly used in multiple virtual machines for storing the disk image.

Initially the file format was developed by the company VMware. After a while to enhance the research and acceptability, VMware made the file storage format open for development and research.

C. VHD

The Virtual Hard Disk image file is called VHD. The VHD is similar to the standard hard disk file. The partitioned properties are also similar to the physical hard disk containers.

The first of its kind of VHD is introduced by a company called Connectix, later which was acquired by Microsoft in the year of 2003 for their complete range of Virtual PC product range.

D. RAW

Another format called the ISO format or the RAW format for storing the virtual machine image files are also very popular. The RAW file formats allows the host virtual machine image to be replicated on other physical host systems.

As the size of the ISO or the RAW file format storage is less space consuming comparted to the other file storage formats, hence RAW format is gaining its popularity and getting widely accepted.

Here we also understand the internal conversion of the formats [Table – 6]

TABLE VI
INTEROPERABILITY OF STORAGE ARCHITECTURES

	VDI	**VMDK**	**VHD**	**RAW**
VDI	-	NO	NO	YES
VMDK	NO	-	NO	YES
VHD	NO	NO	-	YES
RAW	YES	YES	YES	-

During the study it is been observed that the requirement for the automation for the conversion of different virtual machine image formats are also limited.Hence we take the opportunity to automate the conversion process using the following script:

```
$qemu-imgconvert -f raw -O qcow2 vmimage1.img
vmimage1.qcow2
$qemu-imgconvert -f vmdk -O raw vmimage1.vmdk
vmimage1.img
$qemu-imgconvert -f vmdk -O qcow2
vmimage1.vmdk vmimage1.qcow2
$glance image-update --property hw_disk_bus='ide'
image_id
$VBoxManageclonehd ~/VirtualBox\
VMs/vmimage.vdivmimage.img --format raw
```

The above script will convert the .img RAW image into an open stack format. Furthermore the same VMDK file into a RAW file. Henceforth the same file is also converted into Open Stack format. The complete script is completely automatic and can convert any given file formatto RAW format and vice versa.

7. PERFORMANCE COMPARISON MATRIX

The performance of a Virtual Machine needs to be evaluated based on multiple factors to understand the quality of service.During the service level agreement signing process both the customer and the service provider agree to the listed performance parameters. Hence monitoring the performance is one of the key parameter to monitor the same. Hence we understand and propose a set of parameters to monitor the same. The parameters are defined considering overall requirements from the customers and reading multiple service level agreements [Table – 7].

TABLE VII
VIRTUAL MACHINE PERFORMANCE EVALUATION MATRIX

Type	**Name**	**Description**
Dash Board Parameters (Over All Monitoring)	Name of VM	Unique Name of the VM
	Type	Host Architecture Type
	Overall Health Indicator	Running, Stopped, Critical
	Last Backup	Last Backup Date and Time
	Total Availability	Time of Total Availability

	Memory Utilization	Total Memory Utilization time
	Disk Utilization	Total Disk Utilization in GB
	Network Utilization	Total Network Utilization in time
Memory Parameters	Active Memory	Amount of Active Memory in GB
	Over heading Memory	Amount of Over heading Memory in GB
	Swappable Memory	Amount of Swappable Memory in GB / MB
	Total Shared Memory	Amount of Total Shared Memory
	Memory Temperature	Temperature of the Memory Units
Storage Parameters	Container Name	Unique name of the Storage Container
	Container Size	Container Size in GB
	Container Utilization	Container Utilization in GB
Networking Parameters	Network Card ID	Unique id for the Network Interface Card
	Up Time	Total Up Time
	Down Time	Total Down Time
	IP Address	Unique assigned IP Address
	MAC Address	Unique assigned MAC Address
	Data Transfer Rate	Data Transfer Rate in Megabytes per second
Peripheral Parameters (I/O Device)	Device ID	Unique Device ID
	Type	Read or Write
	Read Count	Number of Read Operations
	Write Count	Number of Write Operations

After defining the novel matrix for virtual machine performance evaluation, we test the performance of the proposed system, which is discussed in later part of this work.

8. PROPOSED STORAGE AND COMPUTING VIRTUALIZATION MANAGEMENT FRAMEWORK

After the detail understanding of the virtual machine framework and implementations, we realize the need for improvement in Hypervisor framework. Majorly we understand the following problems are generally encountered in most of the popular Hypervisor software.

The situation where multiple physical hardware servers are utilized for same or multiple clients, then the monitoring and managing individual hardware is nearly impossible from a single firmware or Hypervisors as the Hypervisors are restricted to specific hardware vendors.

Hence we propose the Storage and Computing Virtualization Management Framework [Figure – 3].

The short comings are listed here:

- Lack of overall monitoring
- Lack of backup and restorationcontrol
- Cross hardware and image format replication control
- Simplifiedmonitoring and management of computational capacities
- Simplified monitoring and management of storage capabilities

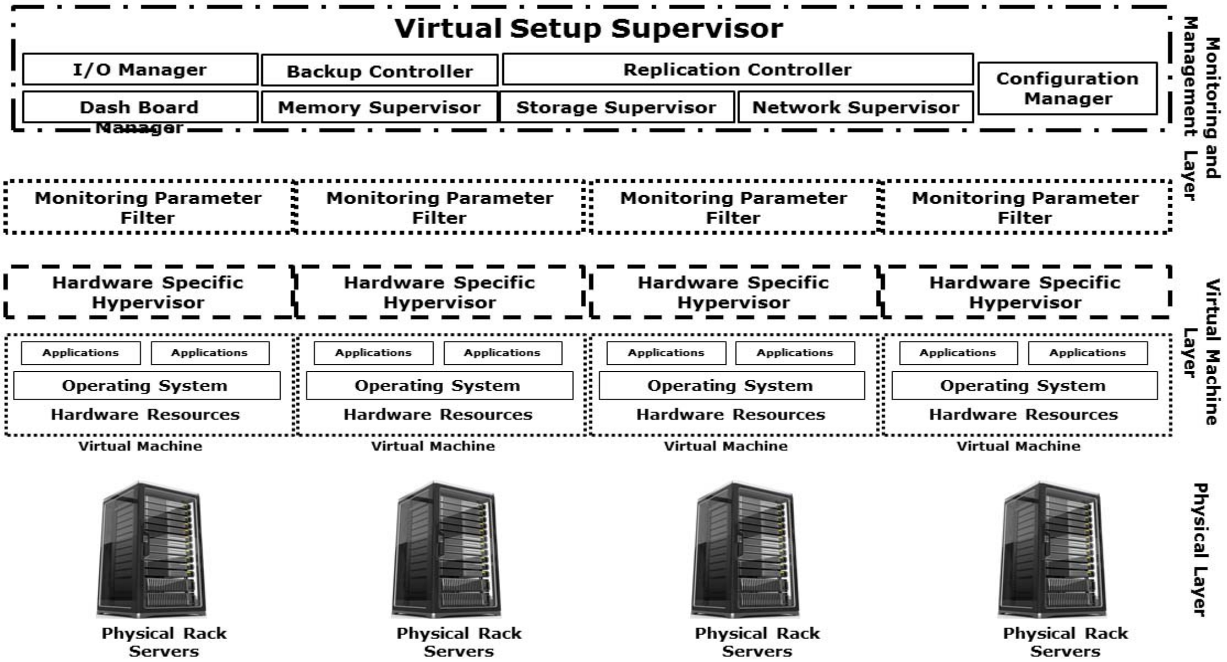

Figure.3 Storage and Computing Virtualization Monitoring and Management Framework

Hence we propose the novel framework to overcome all the listed shortcomings.

The proposed framework is a collection of software and monitoring applications stacks developed based on the open source Hypervisor implementation called Kernel Based Virtual Machine or KVM.

Here we describe the components of the proposed framework:

A. Physical Layer

The physical layer is consisting of multiple physical hardware servers from different vendors. The servers are generic configuration and can be from any vendor specific configurations.

The servers are equipped with a small software package to read the server configuration and performance parameters mentioned in the previous part of this workto be supplied to the next layer.

B. Virtual Machine Layer

The virtual machine layer is generic and standard virtual machine implementation on top of the physical hardware layers.

The virtual machine layers are equipped with another small software agent to collect the virtual machine performance parameters from the specific Hypervisor tool and collect the parameters from the physical layer of the proposed implementation. The software agent will forward the same to the next layer.

C. Monitoring and Management Layer

The top layer of implementation is the Monitoring and Management Layer and works as the name suggests.

The management and monitoring layer is consisting of multiple software agents and described in detail here:

- **Dash Board**: The Dash Board is the overall system report for the complete framework. The portal will be giving the information of the system based on the parameters like Unique Name of the VM, Host Architecture Type, Running or Stopped or Critical status of the system, Last Backup Date and Time, Time of Total Availability, Total Memory Utilization time, Total Disk Utilization in Gigabytes and Total Network Utilization in time.

- **Configuration Manager**: The configuration manager is deployed to detect the change in physical or virtual level configuration and inform the respective software manager available in the framework in monitoring and management layer.

- **Memory Supervisor**: The memory supervisor is responsible for keeping the memory monitoring system updated. The memory supervisor will monitor the system based on the parameters like Amount of Active memory in Gigabytes, Amount of Over heading Memory in Gigabytes, Amount of Swappable Memory in Gigabytes or Megabytes, Amount of Total Shared Memory and Temperature of the Memory Units.

- **Storage Supervisor**: The storage supervisor is responsible for keeping the storage monitoring system updated. The storage supervisor will monitor the system based on the parameters like unique name of the Storage Container, Container Size in Gigabytes and Container Utilization in Gigabytes.

- **Network Supervisor**: The network supervisor is responsible for keeping the network monitoring system updated. The storage supervisor will monitor the system based on the parameters like unique id for the Network Interface Card, Total up Time, Total down Time, Unique assigned IP Address, unique assigned MAC Address and Data Transfer Rate in Megabytes per second.

- **I/O Manager**: The input/output or the peripheral supervisor is responsible for keeping the input/output or the peripheral monitoring system updated. The input/output or the peripheral supervisor will monitor the system based on the parameters likeUnique Device ID, Read or Write type, Number of Read Operations and Number of Write Operations.

- **Backup Controller**: The backup controller is responsible for taking the backups and restoring the same image into different formats on the underlying physical servers.

- **Replication Controller**: The replication controller is responsible for replicating the delta change or the complete replications over multiple underlying physical servers in the physical layer.

The majority of the problems identified in the parallel research outcomes are been solved in the implementation. Mostly the proposed cross image conversion automation, with the use of backupand replication controller this work demonstrates satisfactory outcomes.

Also we present the formulationof the process of property information collection software agent here:

$$p_1 = \sum_{i=1}^{n} p_i(t_1)$$

$$p_n = \sum_{i=1}^{n} p_i(t_n)$$

… Eq 1

The first set of properties, denoted by p, containing the information regarding the dash board are collected time to time to be processed in the framework.

$$q_1 = \sum_{i=1}^{n} q_i(t_1)$$

$$q_n = \sum_{i=1}^{n} q_i(t_n)$$

… Eq 2

The set of properties, denoted by q, containing the information regarding the memory are collected time to time to be processed in the framework.

$$r_1 = \sum_{i=1}^{n} r_i(t_1)$$

$$r_n = \sum_{i=1}^{n} r_i(t_n)$$

… Eq 3

The set of properties, denoted by r, containing the information regarding the storage containers are collected time to time to be processed in the framework.

$$s_1 = \sum_{i=1}^{n} s_i(t_1)$$

$$s_n = \sum_{i=1}^{n} s_i(t_n)$$

…. Eq 4

The set of properties, denoted by s, containing the information regarding the networking components are collected time to time to be processed in the framework.

$$t_1 = \sum_{i=1}^{n} t_i(t_1)$$

$$t_n = \sum_{i=1}^{n} t_i(t_n)$$

… Eq 5

The set of properties, denoted by t, containing the information regarding the peripheral devices are collected time to time to be processed in the framework.

$$\begin{bmatrix} p_1 & q_1 & r_1 & s_1 & t_1 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ p_n & q_n & r_n & s_n & t_n \end{bmatrix}$$

… Eq 6

The final set of parameters are pre-processed and normalized to be kept in the final matrix for overall parameter access of the system.

9. RESULTS

The novel monitoring and management framework for the compute and storage virtualization is been applied and the results observed are satisfactory. The framework is been analysed for performance based on the parameters defined on the previous part of this work.

The performance of the application is been tested on the parameters like Unique Name of the VM, Host Architecture Type, Running or Stopped or Critical status of the system, Last Backup Date and Time, Time of Total Availability, Total Memory Utilization time, Total Disk Utilization in Gigabytes, Total Network Utilization in time, Amount of Active memory in Gigabytes, Amount of Over heading Memory in Gigabytes, Amount of Swappable Memory in Gigabytes or Megabytes, Amount of Total Shared Memory, Temperature of the Memory Units, unique name of the Storage Container, Container Size in Gigabytes, Container Utilization in Gigabytes, unique id for the Network Interface Card, Total up Time, Total down Time, Unique assigned IP Address, unique assigned MAC Address, Data Transfer Rate in Megabytes per second, Unique Device ID, Read or Write type, Number of Read Operations and Number of Write Operations.

The result of the finding are observed here [Table – 8]:

TABLE VIII
PROPERTY AVAILABILITY MATRIX

Parameter Observation	**Availability of the Properties**		
	Test Process – 1 (Duration – 60 Mins)	**Test Process – 2 (Duration – 90 Mins)**	**Test Process – 3 (Duration – 200 Mins)**
Name of VM	Available	Available	Available
Type	Available	Available	Available
Overall Health Indicator	Available	Available	Available
Last Backup	Available	No Continuous Availability	Available
Total Availability	Available	Available	No Continuous Availability
Memory Utilization	Available	Available	Available
Disk Utilization	Available	Available	Available
Network Utilization	Available	No Continuous Availability	No Continuous Availability
Active Memory	Available	No Continuous Availability	No Continuous Availability
Over heading Memory	Available	Available	No Continuous Availability
Swappable Memory	Available	Available	Available
Total Shared Memory	Available	No Continuous Availability	No Continuous Availability
Memory Temperature	Available	Available	Available
Container Name	Available	Available	Available
Container Size	Available	Available	Available
Container Utilization	Available	Available	Available
Network Card ID	Available	Available	Available
Up Time	Available	Available	No Continuous Availability
Down Time	Available	Available	No Continuous Availability
IP Address	Available	Available	Available
MAC Address	Available	Available	Available
Data Transfer Rate	Available	Available	Available
Device ID	Available	Available	Available
Type	Available	Available	Available
Read Count	Available	Available	No Continuous Availability
Write Count	Available	Available	No Continuous Availability

However the some of the cases are also been observed, where No Continuous Availability is recorded.This is also depends on the physical network connectivity between the physical server and the framework system [Table – 9].

TABLE IX
PROPERTY AVAILABILITY MATRIX

Parameter Observation	**Test Process (Duration – 200 Mins)**
Name of VM	40
Type	Active Hosts, Active Hosts, Active Racks
Overall Health Indicator	Active Racks
Last Backup	24.0hrs
Total Availability	24.0hrs
Memory Utilization	0.049
Disk Utilization	0.049
Network Utilization	-
Active Memory	0.049
Over heading Memory	0.049
Swappable Memory	0.049
Total Shared Memory	0.049
Memory Temperature	32
Container Name	NODE
Container Size	160
Container Utilization	159.889
Network Card ID	NAT0
Up Time	-
Down Time	-
IP Address	Hidden
MAC Address	Hidden
Data Transfer Rate	87.839
Device ID	DIV0
Type	All Types
Read Count	-
Write Count	-

10. CONCLUSION AND FUTURE SCOPE

In this work we have considered the benefits of virtual machine migrations from the on-side implementation of the systems and the benefits like Detailed Control, Reduced Hardware Constraints, Replication Control, Availability, Regular Updates, Cost Control, Collaborative Approach and Manageable Data Loads can be achieved. Also we considered the bottlenecks of virtualization and in virtualization adaptation as Access Delay, Acceptability by the researchers, Provisioning Deficiency and Interactivity Limitations. We further studied the components playing major role in virtualization to be taken under consideration for further research as Hardware Components for Virtualization and Software Components or Hypervisor for Virtualization and identified that the dependencies and deficiencies in hypervisors. This work also consider the virtual machine image storage format in order to achieve better understanding of nature of the virtual disk files as Fixed Length Image File Format, Dynamic Length Image File, Dependent Image File and Linked Image Fileand realized the most suitable image formats. In this work we also understood virtual machine storage architectures as Virtual Disk Image, Virtual Machine Disk, Virtual Hard Disk and ISO format or the RAW format and implemented the automated type conversion script.We also defined the performance evaluation matrix consisting of the parameters like Unique Name of the VM, Host Architecture Type, Running or Stopped or Critical status of the system, Last Backup Date and Time, Time of Total Availability, Total Memory Utilization time, Total Disk Utilization in Gigabytes, Total Network Utilization in time, Amount of Active memory in Gigabytes, Amount of Over heading Memory in Gigabytes, Amount of Swappable Memory in Gigabytes or Megabytes, Amount of Total Shared Memory, Temperature of the Memory Units, unique name of the Storage Container, Container Size in Gigabytes, Container Utilization in Gigabytes, unique id for the Network Interface Card, Total up Time, Total down Time, Unique assigned IP Address, unique assigned MAC Address, Data Transfer Rate in Megabytes per second, Unique Device ID, Read or Write type, Number of Read Operations and Number of Write Operations. Finally we proposed the architecture for novel compute and storage virtualization management framework consisting of Physical Layer, Virtual Machine Layer, Monitoring and Management Layer with Dash Board, Configuration Manager, Memory Supervisor, Storage Supervisor, Network Supervisor, I/O Manager, Backup Controller and Replication Controller. Also in the performance matrix we have observed high availability of the performance matrix parameters and better control on the system.

The future work in this direction is to understand and work on the virtualization hardware components and study the energy efficiency of the system.

REFERENCES

[1]. J. Rao, Y. Wei, J. Gong and C.-Z. Xu "Qos guarantees and service differentiation for dynamic cloud applications", IEEE Trans. Netw. Serv. Manag., vol. 10, no. 1, pp.43 - 55 2013

[2]. Li, A. Raghunathan and N.K. Jha, &ldquo,Secure Virtual Machine Execution under an Untrusted Management OS,&rdquo, Proc. Int',l Conf. Cloud Computing, pp. 172-180, July 2010.

[3]. Intel VT-D, http://www.intel.com/technology/virtualization/technology.htm, 2012

[4]. BYTEmark, http://www.tux.org/mayer/linux/byte/bdoc.pdf, 2012

[5]. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, &ldquo,Above the Clouds: A Berkeley View of Cloud Computing,&rdquo, Technical Report UCB/EECS-2009-28, http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html, 2010

[6]. T. Deshane, Z. Shepherd, J.N. Matthews, M. Ben-Yehuda, A. Shah and B. Rao, &ldquo,Quantitative Comparison of Xen and KVM,&rdquo, Xen Summit Boston 2008, http://xen.org/xensummit/xensummit_ summer_2008.html, 2010

[7]. Y. Dong, X. Yang, X. Li, J. Li, K. Tian and H. Guan, &ldquo,High Performance Network Virtualization with SR-IOV,&rdquo, Proc. IEEE 16th Int',l Symp. High Performance Computer Architecture (HPCA), pp. 1-10, 2010

[8]. P. Padala Automated management of virtualized data centers, 2010 :Univ. of Michigan

[9]. T. Patikirikorala, A. Colman, J. Han and L. Wang "A systematic survey on the design of self-adaptive software systems using control engineering approaches", Proc. Symp. Softw. Eng. Adaptive Self-Manag. Syst., pp.33 -42 2012

[10]. Wang and Y. Wang "Coordinating power control and performance management for virtualized server clusters", IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 2, pp.245 -59 2011

[11]. Wang, M. Chen and X. Fu "MIMO power control for high-density servers in an enclosure", IEEE Trans. Parallel Distrib. Syst., vol. 21, no. 10, pp.1412 -1426 2010

[12] T. Patikirikorala, A. Colman, J. Han and L. Wang "A multi-model framework to implement self-managing control systems for QoS management", Proc. Int. Symp. Softw. Eng. Adaptive Self-Manag. Syst., pp.218 -227 2011.