International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-3 Issue-4, September 2014

Review on KVM Hypervisor

Pankaj R Kadam, Nilesh V Alone

Abstract*: *In today's world the boom of cloud computing is made possible due to the virtualization technology. This technology was invented before couple of decades but cloud computing leveraged this technology to build scalable reliable platform which can deploy many applications in the cloud. Virtualized resources come under the infrastructure as service. There are many virtualization technologies available right now in the market which includes VMware, Hyper V, LXC, Xen and KVM. KVM is also known as kernel virtualized machine as it is completely integrated into the Linux kernel itself so it is able to use many native functionalities which are available in the Linux kernel itself. KVM provides the full visualization of guests which allows guests to run unmodified. KVM is completely open source and have a strong community behind this which keeps updating the KVM. In this paper we have reviewed the KVM and studied its detailed architecture. We have also done some tests and evaluation which proves the significant upper hand of the KVM over other hypervisors available in the market.

***Index Terms*— Bare Metal Performance, Benchmarking Cloud computing, Full Virtualization, open source virtualization.**

I. INTRODUCTION

Cloud computing is all about providing and consuming the computing power resources. These resources can be in any form like virtual machines, online software services and even virtualized hardware. Cloud or public clouds are the huge pools of these resources which geographically distributed over the globe in the various datacenters. Due to their huge capacity and virtualization technology clouds are scalable and flexible in the nature. One can easily increase or expand his computing resource requirement in the cloud without requiring deploying the actual hardware. Instant resources are allocated to the users on demand from the already deployed hardware. Cloud computing is very successful invention in the computer science in recent years as many studies have proven its tremendous growth rate. Gartner in its recent study predicted the cloud market to reach to more than $180 billion by the end of the 2015. [1] In nskinc study 88 percent companies reported that they have saved the significant amount of the money moving to the cloud. [2]

Virtualization is the base of the cloud as it allows cloud service providers to divide actual physical resources to virtual resources. These virtual resources are then allocated to the individual clients or the consumers. Although many virtualized resources are on the single hardware node but it provide privacy and security to each tenant of the cloud or the user of the cloud. Virtualization is available in the market from the 1960 and it was first used in the mainframe computers to divide the resources logically so they can be utilized by different applications running on the mainframe computers. Time sharing mechanism was used in back then so that it can execute multiple applications. This was not a robust solution for the virtualizing the resources as error caused by one application was affecting all other applications running on that hardware. There was no isolation in the hardware and at application level which was causing many issues with the system. In 1970, IBM proposed the mainframe computer which can actually divide the hardware and also provided the isolation part which was missing from the previous solution. Administrators were able to partition the hardware on the basis of their application requirements. In 1980 due to invention of x86 architecture computer hardware prices were dropped and consumers were able to purchase the hardware at low cast. Dream of personal computer was realized and users were able to execute their applications on the personal computer. These computers are also able to do the multitasking so people felt that there is no need of virtualization now as we are getting the enough resources in the budget cost. Around 2005, Intel extended its IA 32 instruction set to support virtualization technology. This move by Intel made significant changes as virtualization was made available to everyone as mediocre CPU were also supporting the virtualization natively. This move really helped to attract people to virtualized environment.

Due to availability of the hardware which supports the virtualization company like VMware [3] started developing the hypervisors which allow users to create virtual machines on their computers itself. Virtualization enabled them to extract the features from the real hardware to isolated environment called as virtual machines. This virtual machine can have different operating system than host operating system. At the start isolation of the various applications was the primary use of the virtual machines but today virtual machines are primarily used for the consolidation purpose. In many data centers the 20 to 30 servers are virtualized and consolidated on the single physical hardware node. Consolidation process reduces the hardware costing and end up saving in energy and capital spending. Consolidated environments are easy to manage as they can be migrated to and from various hosts available.

II. VIRTUAL MACHINE MONITOR

These virtual machines are managed by virtual machine monitor (VMM) or also called as the hypervisor.

Revised Manuscript Received on 30 September 2014.
* Correspondence Author
Pankaj R Kadam*, R. H. Sapat College of Engineering, Management Studies & Research, Nashik, M.S. India.
Prof. Nilesh V Alone*, R. H. Sapat College of Engineering, Management Studies & Research, Nashik, M.S. India.

VVM generally represents software layer in between the actual host hardware and virtual machines running on it. The hardware which runs the VMM is also called as the host or the node. On this host VMM is installed directly or on the operating system which manages the virtual machines. Management operation includes creation, deletion and migration of virtual machines. VMM also controls the life cycles like start, stop and restart operations of the virtual machine. VMM also have the various set of tools to allocate the resources to virtual machines.

VMM's can be broadly classified to two types namely called as bare metal and hosted virtual machine monitors. In type 1, bare metal virtual machine monitors are directly installed on the hardware so they are also called as native hypervisors. There is no need of operating system for the host in this bare metal installation. The Guest operating system lies above the hypervisor directly.

In type 2 which is called as hosted hypervisor needs a base operating system on which it runs as an application. This type involve three layers first one is the base operating system, second one the hypervisor itself and on third it hosts the guest operating system. Type 2 hypervisor are used for doing the testing and sandboxing purpose while most of the datacenters use Type 1 hypervisors in the actual hardware deployments.

The main goal which emerged over time regarding the virtualization is to achieve the native performance of the hardware in the virtualized environment. So the hypervisors should consider about providing the native and close to the real hardware performance in the virtual machines. A virtual machine generally has at least one virtual CPU resource while full-fledged virtual machines have vCPU, virtual memory and SAN disk storage. So each host need to use some kind of sharing algorithm. Virtual machine monitor or the hypervisor does this for host machine. Time sharing is generally used by the VMM to allocate the virtual CPU's to the virtual machines. Memory management is also handled by the VMM, VMM divides the host physical memory into various sections which can be easily accessible from the virtual machines. Each machine gets its memory addresses allocated to itself. Besides the CPU and memory there are various devices which makes virtual machine a complete machine. Virtual machine doesn't have the access to the physical hardware devices so these devices are provided by the virtual devices. These virtual devices are needed to install its device drivers at first installation process.

Hardware assisted virtualization made memory management more efficient. Hardware itself started supporting the VMM which allowed VMM to execute the privileged instructions on the host machine by guest operating system. New operating system runs applications in the less privileged mode while operating system runs in the most privileged mode so that it can access various IO and hardware resources required. CPU contains various privilege rings the one which is close to the CPU have high privileges and operating system run in that ring while applications run in the outer rings. Virtual machine requires some privileges from the ring 0 as they need to execute the code of the operating system. Some software techniques like binary transition and traps emulation allowed VMM to do this but there were high overhead involved. Hardware assisted virtualization solved this problem and allowed the virtual machine to execute their code in more privileged rings without any additional costs. Hardware support also includes Intel's extended paging table and AMD's Nested paging table which allowed translation between the virtual memory. This eliminated the need of holding the shadow page in the memory which in turn increased the performance of the system.

VMM can realize the virtualization by two means called as para virtualization and full virtualization. In para virtualization guest do not run in the most privileged ring i.e. ring 0. So to execute this instruction one need to modify the operating system so it can support the guest operating system. This involves patching of the operating system so that it can run smoothly in virtualized platform. Other technique is called as full virtualization which uses the hardware assisted virtualization. Hardware support allows one to run the operating without any modification in the code of the operating system. This provides the secure environment for the users.

III. KERNEL BASED VIRTUAL MACHINE

Qumranet developed the first version of KVM in February 2007. Intel and AMD exposed the hardware virtualization support to all of its processor variants which boost the research in this segment. In September 2008 Qumranet was acquired by the red hat and developed the KVM into a robust product. [4] The source code of KVM was already open sourced by the Qumranet which allowed many companies to explore this and developed their own variants. The open source community allowed many improvements in upcoming KVM versions. KVM is natively present in all the red hat and cent OS variants. KVM code was originally 100% orthogonal to kernel of the Linux so the merging of the Linux and kvm was a sophisticated merger. The further development of the KVM was also significant as KVM was able to port to many available architecture like power PC and IA64. Redhat and open community kept the wheel running and many new virtualized features are released by the KVM.

Currently KVM open source community is known as open virtualization alliance. [5] This alliance is managed by its members who include many giant companies like HP, IBM, Intel, Net App and Redhat. KVM use full virtualization technique to virtualize the virtual machines. KVM support all major operating systems which includes Linux and windows on all x86 hardware. Since version 2.6 KVM became the part of the Linux kernel itself so KVM is now available in the kernel of Linux itself. Which is actually a biggest benefit for KVM as it is developed alongside the actual Linux development. KVM gains performance benefits from the algorithms which are implemented in the Linux kernel. So any improvement in the Linux kernel actually benefits the KVM performance as KVM uses much native functionality implemented in the Linux kernel.

As KVM is part of the Linux it can leverage Security-Enhanced Linux which was developed by the US Security Agency. [6] It allows one to set various security policies and access controls at various levels.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Tight integration with Linux kernel yields many such advantages to the users which include performance and security benefits.

IV. KVM ARCHITECTURE

As we know that Linux kernel have all the tools necessary to run the virtual machines as kernel includes tools like memory management, device drivers, process scheduling and IO provisioning. KVM uses the Linux kernel as the hypervisor. There is no need to implement the separate code or mechanism to schedule the processes of running virtual machines. KVM is small code base of 42 thousand lines which is included in the kernel itself so you can consider KVM as module of the kernel. [7] The architecture of the KVM can be diagrammatically represented as below

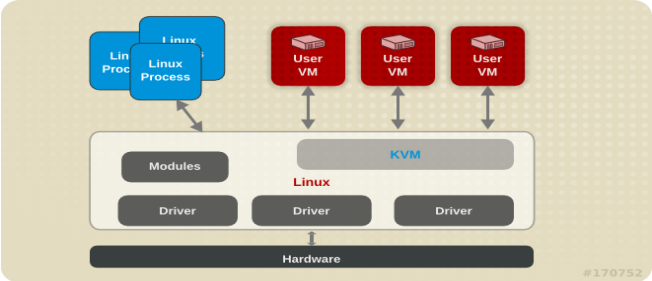

Figure 1 Architecture of KVM

As KVM is developed after the invention hardware virtualization support, KVM leverage this technology to introduce new mode called as the guest mode. In normal Linux operating system process can run in the two modes popularly known as kernel mode and user mode. KVM introduces the third mode called as guest mode. Guest mode is used to execute the guest code only and guest IO operations are handled by the user mode. Kernel mode can switch to guest mode to handle the IO exits and IO operations. Virtual machine runs as the process in the host on which KVM hypervisor is used. Concept of the thread is used to implement the virtual CPU; these threads can be managed by CPU scheduler. Devices like Disk, Network cards and other IO devices are directly accessed by virtual machines which improve the IOPS and overall performance of the guest machines running on the KVM host. Network traffic is sent via a software bridge so there is no overhead in the network also.

KVM implements virtual machines as the processes which are assigned the processing power by the CPU scheduler itself. This solves the vCPU emulation problem and there is no need to do the binary transition which is way complex and involve overhead. As multiple virtual machines are expected to run on a single host they can share the computing power by using native time slicing technique. Hardware assisted virtualization allows one to run the guest OS in the kernel mode and can send trap to the kernel. In Intel VT X technology a control structure is maintained called as VMCS which holds the information about the guest state and host states. There states include registers, memory and interrupts details. VT x allow direct emulation of the virtual CPU however interrupts are still handled by the kernel mode. Various schedulers like NUMA (Non-Uniform Memory Access), CFS (Completely Fair Scheduler) and RCU are used to provide the computing power to the processes which are running as virtual machines. KVM can support 16 vCPU.

Due to massive use of Linux kernel code KVM uses the same memory management mechanism which is used by the Linux kernel to allocate memory to the processes. Each VM appears to have the physical memory which is actually the virtual memory. Memory addresses inside the VM maps them to the process memory with which the KVM host is running. This process memory is then mapped to physical memory of the host. KVM connect these two layers of the translation with the help of shadow page table or assisted hardware assisted virtualization. KVM keeps a second set of page table which is called as shadow page table. When any guest write a page fault occurs then this write is emulated by the host and an entry is added to the shadow page table. Another solution to the memory mapping is Extended Page Table. This is a hardware solution which is provided by all the leading processor manufacturers. Guest OS maintains guest process table and host maintains the VM table. When any TLB miss occur CPU does the two lookups.

After processor and memory the next essential thing for a virtual machine to function is various hardware devices. To emulate these devices KVM uses QEMU[8]. QEMU is implemented as process which is started for each VM and certain devices are attached to virtual machine via this process. These devices are also called as Para virtualized devices. QEMU emulates the entire PC platform which includes graphics, disk drives and networking. When any IO operation is performed by VM, these are intercepted by the KVM and these requests are redirected to QEMU which handles these requests. As there is very less emulation in between the actual hardware and virtual machines, this really improves the performance of these devices. Any type of dongle of USB device can be recognized by the virtual machine as its native device. Virtual machines can also access the PCI slots via PCI pass through which allows access to any graphic card installed in the PCI slot.

Various operations in KVM are implemented using the ioctl calls. These calls are handled by the interface known as /dev/kvm. This interface is created when you install the KVM module in the Linux kernel. These system calls are used to create new virtual machine, assigning the disk, assigning the memory and various other operations. Ioctl's are divided into three group called system level requests, VM level requests and vCPU level requests. /dev/kvm maintains its own file descriptor when a new vCPU is created it creates new file descriptor for it. Following are the common ioctl's involved in the creation of the virtual machines.

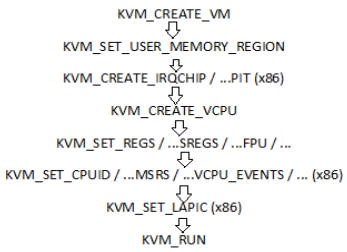

Figure 2 Flow of ioctls

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The other normal operating of the virtual machine involves lots of switches in between the user mode, kernel mode and guest mode. User mode is generally used while using ioctl or any IO operation required. Guest keep executing its code until any IO operation is required or any external event occurs. Kernel mode maintains the execution of the guest workflow if guest exists due to any memory requirement or the IO requests kernel handles that request and resumes the work flow back to the normal. In guest mode system uses the extended instruction set which can run on the virtual CPU normally. This will continue to work until there is any external interrupt.

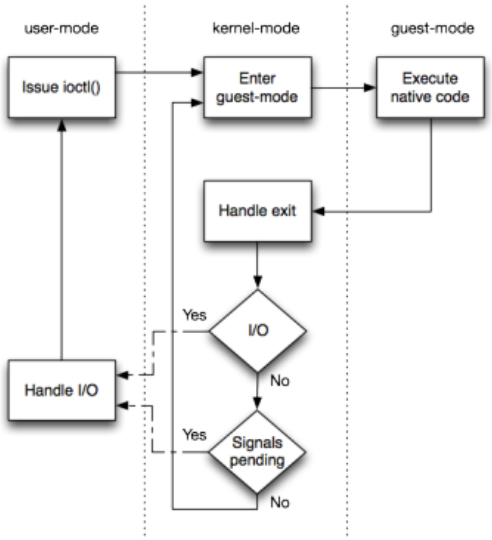

Figure 3 Switch of modes

These switches between the kernel mode and user mode are pretty fast but still consume some overhead CPU cycles. IO operations are also requires a switch to user mode, as disk is considered as a files and data is read from the disk when required. This user mode emulated IO can slow down the VM operations hence they have implemented the backend called Virtio. [8] Virtio is used to write the VMM independent device drivers which improves the IO speed and ensures the bare metal speed for the virtual machines.

V. EXPERIMENT SETUP

We have studied the architecture of the KVM and also seen the facts which give the various advantages to the KVM over other hypervisors available. To study and compare the relative performance we have done the following setup:

We have setup three host servers with following specifications:

1. Server: i7 3770K processor, 8 GB RAM, 40 GB SSD space Hypervisor: KVM
2. Server: i7 3770K processor, 8 GB RAM, 40 GB SSD space Hypervisor: Xen 6.2
3. Server: i7 3770K processor, 8 GB RAM, 40 GB SSD space Hypervisor: VMware Exsi 5.5

We have created one virtual machine on the each hypervisor to do our benchmarking. Virtual machine has the specification 1 vCPU, 1 GB RAM and 20 GB SSD space. These Virtual machines have Cent OS 6.4 installed on them.

VI. EVALUATION AND RESULT

To evaluate the performance of these virtual machines we have installed the phoenix test suite on each of the virtual machines and ran some of the famous performance benchmarking test on these virtual machines. The test we have done includes the C-Ray 1.1 test, Apache Benchmark 2.4.7 test, RAMspeed copy test for both the integer and floating values, RAMspeed scale test for both the floating point and integer values, PostMark 1.51 test, 7-Zip Compression 9.20 test and OpenSSL test. Below are the description of these various tests and results obtained on virtual machines.

A. C Ray Test:

C Ray test is designed to calculate the CPU's floating point performance here we have compared the performance of the virtual machines where vCPU's are the main compute power resource. This trace rays in a multithreaded manner (16 threads per core) which generated a 1600 x 1200 image [9]. This test benchmarks the processor of the virtual machine.

Table 1. C ray Benchmarking Test

Test Number	KVM	Std. Dev	Xen	Std. Dev	VMware	Std. Dev
1	406	-	344	-	512	-
2	359	-	375	-	286	-
3	372	6%	369	0.8 %	344	30%
4	396	5%	336	-	283	30%
5	370	5%	373	-	226	33%
6	366	4%	381	-	219	34%
Average	378	-	363	-	312	-

The values are in seconds. Seconds taken by the processor to generate an image. Lower values represent better performance. In this benchmark KVM lags behind the performance on the Xen and VMware.

B. Apache Benchmarking Test:

This test is ab type test. [10] This test benchmarks the whole system performance. This test builds a profile and measures how many requests a given system can sustain in the controlled environment. In this 100 apache requests are carried out concurrently.

Table 2. Apache Benchmarking Test

Test Number	KVM	Std. Dev	Xen	Std. Dev	VMware	Std. Dev
1	2283	-	2226	-	1285	-
2	1939	-	2200	-	1219	-
3	1603	17	1794	11	1489	10
4	2004	14	1780	12	1139	11
5	1561	15	1739	12	1279	10

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

6	1656	15	1627	13	1163	9
Average	1841	-	1894	-	1262	-

The values represented here are in http requests served per second. Higher values are better. Standard deviation is presented in the percentage. In overall system performance KVM beats VMware by a significant difference while Xen stands as a winner by very little margins

C. *RAMspeed SMP 3.50 Test:*

This benchmark test calculates the memory performance under both the integer and floating point values. [11] This measures both the memory and cache bandwidth values. [12] We have perform two test called as copy test and scale test on selected hypervisors. In copy operation we are simply copying the data from one location of memory to another while in scaling we are multiplying the value with certain number and then we are moving it to the specified value.

a. Copy

Table 3 RAMspeed copy Benchmarking Test:

Test	KVM	Type	Xen	Type	VMware	Type
1	5931	I	6074	I	5801	I
2	5500	F	5218	F	4982	F

I Integer
F Floating
Values are in MB/s and higher values represent the better performance. In Copy performance KVM beats both Xen and VMware.

b. Scale

Table 4. RAMspeed Scale Test

Test	KVM	Type	Xen	Type	VMware	Type
1	5647	I	6057	I	6029	I
2	5188	F	5337	F	5064	F

In the scaling operation Xen performed better than both the KVM and VMware.

D. *PostMark Test:*

NetApp have developed this PostMark 1.51 Test which emulates and generates various files like a mail server or apache server. This test generates 25,000 transactions by using 500 files of various sizes ranging from 5 to 512 kilobytes. This benchmark judges the performance of the disk attached in the unit of transaction per seconds

Table 5. PostMark Benchmarking Test Results

Test	KVM	Std. Dev	Xen	Std. Dev	VMware	Std. Dev
1	700	-	534	-	905	-
2	710	-	492	-	909	-
3	704	0.7%	516	4%	880	1%
4	-	-	570	6%	-	-
5	-	-	593	7%	-	-
6	-	-	515	7%	-	-
Average	705	-	537	-	898	-

Values presented in the table are in transaction per second unit. Higher values represent the better performance. In this benchmark KVM beats the Xen TPS performance while VMware stand out to be the leader.

E. *7 Zip Compression Test:*

7-Zip program comes with benchmarking software itself. [13] We have run this benchmarking for measuring the performance of the virtual CPU.

Table 6. 7 Zip Compression Benchmarking Test Results

Test Number	KVM	Std. Dev	Xen	Std. Dev	VMware	Std. Dev
1	2090	-	1652	-	1570	-
2	2468	-	1969	-	2688	-
3	2217	8 %	1739	9 %	2676	27%
4	2265	6 %	1652	8. %	3426	29%
5	1977	8 %	1696	7 %	3942	31%
6	2488	9 %	1582	7 %	2414	29%
Average	2251	-	1715	-	2786	-

The values presented in the above table are in million instructions per second (MIPS). Higher values mean high performance. In this benchmark KVM beats the Xen performance while VMware stands as the leader of the test.

F. *OpenSSL 1.0.1g Test*

This test measures RSA 4096-bit performance of OpenSSL. OpenSSL is free toolkit which implements the SSL and TLS protocol. This is also CPU intensive test and judges capabilities of the processor.

Test	KVM	Std. Dev	Xen	Std. Dev	VMware	Std. Dev
1	70	-	63	-	53	-
2	77	-	61	-	47	-
3	74	5%	65	3 %	47	7%

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

4	59	11 %	-	-	52	6%
5	84	12 %	-	-	53	6%
6	63	12 %	-	-	47	6%
Average	71	-	63	-	50	-

The values presented in the above table are represented in the signs per second unit. In this test KVM outperforms than both Xen and KVM. We can clearly see the increase in the performance as KVM is able to 72 sings per second while Xen does 64 signs and VMware falls short to only 50 signs per second.

We have conducted total of 6 benchmarking tests to judge the performance of the virtual machine running on the KVM, Xen and Exsi 5.5 hypervisor. In Graphical performance which we have benchmarked using the Cray test KVM fall short while VMware was the leader. In the Apache benchmarking test Xen and KVM performance was equal but they beat the VMware hypervisor by significant margins. RAMspeed which benchmarked the memory performance KVM beats both Xen and VMware. In IO performance KVM beats the Xen with significant margins. In the openssl and 7 Zip compression test KVM beats to manage the Xen and VMware.

With the above summary of the result, KVM easily beats the Xen in our various benchmark tests performed. VMware found as the leader in many tests but the licensing cost and vendor lock in makes it unfavorable choice for the most of the public cloud providers. So in open source space one can now look up for the KVM as an alternative to Xen

VII. CONCLUSION

In this paper we have studied the detail structure of the KVM hypervisor. KVM uses most of the functionalities from the Linux kernel hence provide better performance in terms of IO operations and overall performance. We have also done some benchmarks on the KVM virtual machines and compared them with the other hypervisor performance. The benchmark results showed that KVM performance is better than the Xen hypervisors in many tests. Although VMware perform better in benchmark results but the licensing cost and vendor lock in makes it an unfavorable choice for the public cloud providers. KVM has a great future ahead as it is now developed as an enterprise class product which can be deployed in the large public clouds

ACKNOWLEDGEMENT

We sincerely thanks to all the people who helped help us to write this research paper. We also like to thank all the open source software developer communities because of which we are able to use the software's used in our experimental setup and conducted tests

REFERENCES

1. Ian Marriott, "Will Cloud Services Make or Break Your Offshore Provider" Gartner documents
2. Nskinc study on cloud computing, Nskinc white paper 2012
3. VMware www.vmware.com
4. The Evolution of Virtualization: Qumranet joins Red Hat http://www.redhat.com/promo/qumranet/
5. Open Virtualization Alliance https://openvirtualizationalliance.org/
6. SE Linux Project http://selinuxproject.org/
7. KVM Source Code http://git.kernel.org/pub/scm/virt/kvm/kvm.git/
8. Virtio Drivers http://www.linux-kvm.org/page/Virtio
9. C Ray Ray tracing benchmark http://www.futuretech.blinkenlights.nl/c-ray.html
10. Apache HTTP server benchmarking tool http://httpd.apache.org/docs/current/programs/ab.html
11. RAMspeed Benchmark http://www.alasir.com/software/ramspeed/
12. Jinho Hwang, Sai Zeng and Frederick y Wu, Timothy Wood "A Component-Based Performance Comparison of Four Hypervisors" IEEE Publications.
13. 7 Zip Test http://openbenchmarking.org/test/pts/compress-7zip

AUTHOR PROFILE

Pankaj R Kadam received his BE degree from Pune University. He is currently pursuing his master degree in computer science. He also has 2.5 Years of working experience in the cloud and worked on the various cloud technologies. He is currently working with a private firm and managing the cloud servers for the clients

Nilesh V Alone is currently working as Assistant Professor in Department of Computer Engineering, R. H. Sapat College of Engineering, Management Studies & Research, Nashik. He is having 13 years of experience in teaching. His area of interest is in Cloud Computing and Machine Learning.