What is Application Virtualization?

Over the last decade, Application Virtualization technology has emerged as one of the most prominent and useful virtualization techniques. In previous years, software application deployment was handled through traditional client-server management tactics.

Using these approaches, software applications were housed on file servers, and IT administrators used to install these applications manually on target machines. With the introduction of Group Policies and Active Directory (AD), software deployment on desired systems became a much more sophisticated and feasible process.

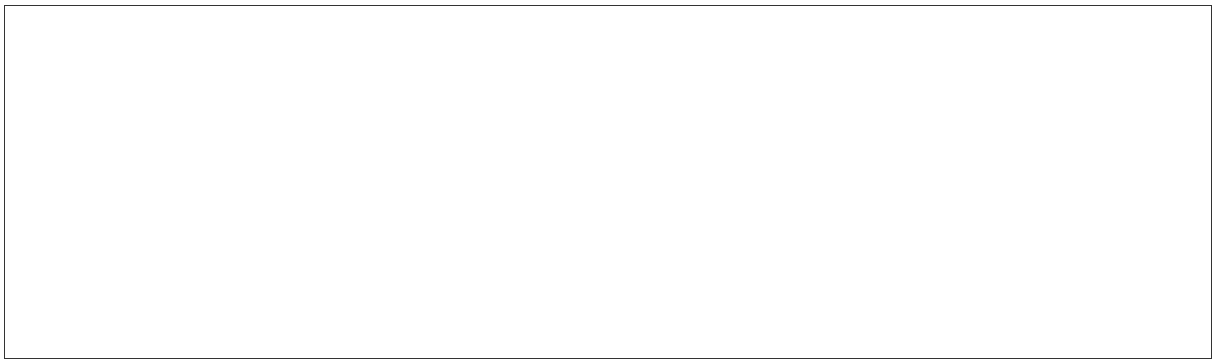
However, application manageability and compatibility issues remained major concerns to deal with. Thus, application virtualization was introduced as the ultimate solution to prevent all these problems. With this technology, a software application runs on the end user's system similar to a locally installed application. However, the app uses a sandbox environment in the background which removes the need to install the application to the local OS.

We will cover an in-depth overview of application virtualization and the benefits that it can bring to your organization. So, let's get started already!

How does it work?

Application virtualization refers to a software technology that lets its consumers use an application from a different device than the one on which it is actually installed. Using this technology, the application interacts and functions in the same manner as if it were natively installed on the user's machine. This ensures a seamless experience for the end-user. The user can perform any action on the application without the need to install it on the native device.

Usually, the IT administrator deploys remote applications on a central server in the organization's data rams to a user's desktop or other connected devices. The user may now access and use the programcenter. Application virtualization software is then used to distribute the prog as if it were installed locally on their system. The user's actions are then sent to the server for execution.



How Does Application Virtualization Differ From Server And Desktop Virtualization?

In desktop virtualization, desktops and operating systems are virtualized and deployed to local or distant clients. In this way, users can operate their desktop systems from any gadget. Whereas, in application virtualization, apps are encapsulated from the operating system.

App virtualization allows users to operate a virtualized application from literally anywhere. On the other hand, desktop virtualization provides the feasibility of utilizing virtualized operating systems and desktops from any machine.

With this technology, applications are completely isolated from the underlying operating system. With desktop virtualization, this abstraction layer is missing as apps are still bundled to the underlying OS.

Server virtualization is the process of subdividing a single physical server into several virtual servers in order to maximize the organization's resources. This technique boosts the server's utilization and facilitates an efficient disaster recovery mechanism. By virtually storing servers, the costs involved in setting up separate physical servers and their maintenance are ultimately reduced.

All three types of virtualizations described above operate with the goal of providing easier access to end-users via different means while reducing extra costs, thereby offering greater flexibility and minimized budgets for the organization.

Use Cases for Application Virtualization:

Here is a list of a few use cases where app virtualization can be employed:

1) Limitation of expenditure:

If you have a massive end-user base where the count of your end-users seems to be relentlessly growing, purchasing and managing separate expensive machines for every single user will undoubtedly drain your budget.

In such cases, virtualization will definitely come in handy since it possesses the capability to deliver all the essential applications to any target device.

2) Remote-safe approach:

Application Virtualization facilitates safe and secure remote access to critical applications from any end device. With remote work culture emerging as an increasingly effective work model globally, the majority of organizations have adapted themselves to transition into a remote work-from-home routine.

In such remote working scenarios, this state-of-the-art technology fits as the best approach since it provides both security and ease of access.

3) Implementation of in-house applications:

Another common scenario where virtualization can certainly prove to be useful is the deployment and implementation of in-house applications. Normally, such applications are often updated by the developers. Application virtualization allows for remote upgrades, installation, and distribution of important applications in a comprehensive manner. Therefore, this technology holds critical importance for organizations that develop and utilize in-house applications.

4) Rolling out cloud applications:

Application virtualization can assist in planning a sophisticated and managed approach for handling and ensuring a smooth cloud transition of an application that is currently being utilized as an on-premise version in parts of the same organization. In such scenarios, it’s necessary to ensure the proper functioning of the application during the ongoing rollout to cloud premises.

By utilizing a state-of-the-art virtualization platform, you can ensure maximum continuity and minimum disruption for your end-users. Such platforms will assist in guaranteeing a smooth delivery of both the on-premise and cloud versions of the application to different groups residing within the same workspace.

Benefits of Application Virtualization:

Some of the numerous advantages that app virtualization has to offer are described below concisely:

1) Easy access to updates:

For critical software applications, the latest software patches and upgrades are necessary to install on a regular basis. With application virtualization, the installation of these upgrades turns out to be a hassle-free procedure as the virtualized applications are updated on the centralized servers and not on end user's machines.

This necessarily means that once the updates are completed, the latest patches can be distributed straight to devices using virtualization software. For the end-user, these upgrades will take place in the background instantaneously without disrupting user productivity.

2) Feasible installation process:

Using virtualization, you just need to install the desired application on the server and afterward, it can be virtualized to as many devices as you wish. Virtualization simply removes the requirement of installing the application on every single device. As a result, the time and effort consumed during installation are minimized.

3) Simplified Deployment:

With application virtualization, deployment on clients' systems becomes a straightforward process. Clients or partners can easily install the application on their machines by using the sent executable file that contains the configured application code. Hence, deployment using app virtualization results in a hassle-free procedure.

4) Minimized Application Conflicts:

The installation of conflicting applications on the same system can sometimes create issues and cause one of the applications to crash.

Since virtualized applications necessarily run within a Sandbox or containerized environment, other applications installed on the same device cannot detect or collude with them. This leads to a smooth functionality for applications that usually conflict with each other while residing on the same physical machine.

A prominent example of such a scenario is related to the numerous versions of JAVA Runtime Environment (JRE). With the lack of app virtualization, two applications requiring different versions of JRE

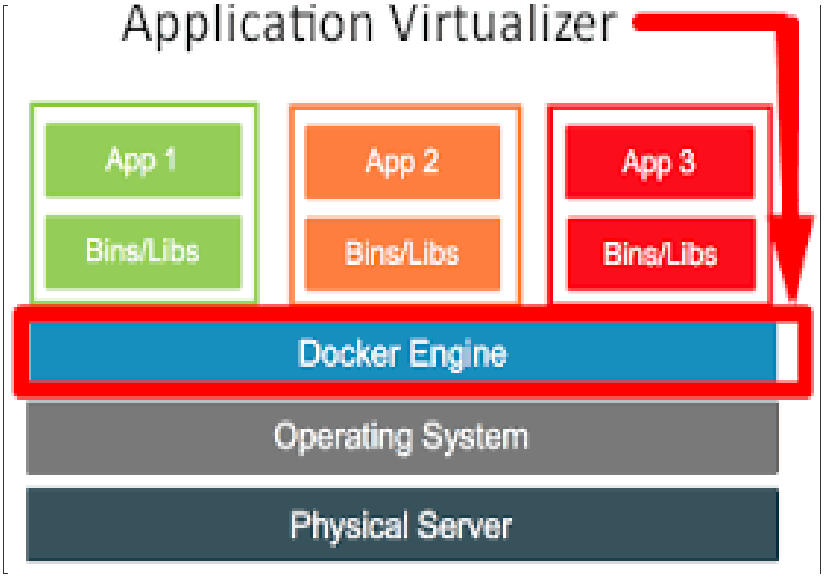
would require installations on two different systems. However, with virtualization in place, both these applications can easily reside within the same end machine.

5) Enhanced security:

Data security is one of the most crucial aspects for every organization's operations to run smoothly. Any sort of malware, virus, or potential threat that appears in any natively installed application or within the operating system of the end device does not affect the virtualized applications whatsoever as they are segregated from one another.

This technology provides a centralized control on the user's access right to specific virtualized applications. This results in a safe and systematic approach for protecting sensitive information that might be accessed via virtualized applications. For instance, if any company device is stolen or lost by an employee, the IT administrators can revoke access to the virtualized applications across the organization from that specific device to ensure data integrity and protection.

Why do you need an Application Virtualization solution?



The list of benefits to employing application virtualization knows no bounds. App virtualization allows IT administrators to install important apps only once on a dedicated server from where they can be later distributed to end devices using virtualization. It also assists in a smooth and uncomplicated software

update and patching mechanism. IT administrators can deploy virtualized applications on every connected device regardless of the operating system that's running on it.

Application virtualization software provides centralized access and authentication management to IT administrators. Using such solutions, IT admins can manage the permissions and access to virtualized applications efficiently.

Monitoring the usage of applications also becomes a simple task and access to sensitive critical information can be easily revoked via remote means in case of device theft or loss. Moreover, it's relatively easy to remove or delete the application completely from the server as compared to removing it from all individual devices.

The bottom line – Application virtualization solutions offer seamless app deployment features, boosted performance, and easy-to-maintain virtualization environments. Moreover, some solutions also provide a remote wipe feature to protect sensitive data in case of unwanted events. Licensing components and enhanced security are a few more innovative capabilities that such solutions tend to offer.

V2 Cloud – The Ultimate Virtualization Solution

With V2 Cloud, deploying application virtualization solutions becomes an incredibly easy and quick procedure. Our

scalable Desktop-as-a-Service (DaaS) solution provide a compelling digital workspace experience for your workforce, allowing users to access virtualized applications on the go.

VIRTUAL CLUSTERS AND RESOURCE MANAGEMENT

A physical cluster is a collection of servers (physical machines) interconnected by a physical network such as a LAN.

- This chapter – intro to – virtual clusters, its properties and potential applications.
- Three critical design issues of virtual clusters:
    - live migration of VMs
    - memory and file migrations

- Dynamic deployment of virtual clusters.

3.4.1 Physical versus Virtual Clusters

Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters.

- **The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks.**
- Figure 3.18 illustrates the concepts of virtual clusters and physical clusters.

Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters.

- The virtual cluster boundaries are shown as distinct boundaries.

The provisioning of VMs to a virtual cluster is done dynamically to have the following properties:

- The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OSes can be deployed on the same physical node.
- A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.
- The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance server utilization and application flexibility.
- VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.
- The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similar to the way an overlay network varies in size in a peer-to-peer (P2P) network.

- The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.

Since system virtualization has been widely used, it is necessary to

- effectively manage VMs running on a mass of physical computing nodes (also called virtual clusters) and
- build a high-performance virtualized computing environment.

This involves

- virtual cluster deployment,
- monitoring and management over large-scale clusters,
- resource scheduling
- load balancing
- server consolidation
- fault tolerance

- Figure 3.19 shows the concept of a virtual cluster based on application partitioning or customization.

The different colors in the figure represent the nodes in different virtual clusters.

- Issues to consider – how to efficiently store the large number of VM images in the system.

There are common installations for most users or applications, such as operating systems or user-level programming libraries.

- These software packages can be preinstalled as templates (called template VMs).
- With these templates, users can build their own software stacks.
- New OS instances can be copied from the template VM.
- User-specific components such as programming libraries and applications can be installed to those instances.

Three physical clusters are shown on the left side of Figure 3.18.

- Four virtual clusters are created on the right, over the physical clusters.
- The physical machines are also called **host systems**.
- In contrast, the VMs are **guest systems**.
- The host and guest systems may run with different operating systems.
- Each VM can be installed on a remote server or replicated on multiple servers belonging to the same or different physical clusters.
- The boundary of a virtual cluster can change as VM nodes are added, removed, or migrated dynamically over time.

3.4.1.1 Fast Deployment and Effective Scheduling

The system should have the capability of fast deployment.

- Here, deployment means two things:
    - to construct and distribute software stacks (OS, libraries, applications) to a physical node inside clusters as fast as possible,
    - to quickly switch runtime environments from one user's virtual cluster to another user's virtual cluster.

If one user finishes using his system, the corresponding virtual cluster should shut down or suspend quickly to save the resources to run other VMs for other users.

- The concept of "green computing" has attracted much attention recently.

- Approaches
    - Focus on saving the energy cost of components in individual workstations
    - Apply cluster-wide energy-efficient techniques on homogeneous workstations and specific applications.

- The live migration of VMs allows workloads of one node to transfer to another node.
- Problem –
    - it does not guarantee that VMs can randomly migrate among themselves.
    - potential overhead caused by live migrations of VMs
        - overhead also affects cluster utilization, throughput, and QoS issues.
    - challenge – determine how to design migration strategies to implement green computing without influencing the performance of clusters.
    - Another advantage of virtualization is **load balancing of applications** in a virtual cluster.
    - Load balancing can be achieved using
        - the load index and
        - frequency of user logins.
    - The automatic scale-up and scale-down mechanism of a virtual cluster can be implemented based on this model.
    - Consequently,
        - we can increase the resource utilization of nodes and
        - shorten the response time of systems.

- Mapping VMs onto the most appropriate physical node should promote performance.
- Dynamically adjusting loads among nodes by live migration of VMs is desired, when the loads on cluster nodes become quite unbalanced.

3.4.1.2 High-Performance Virtual Storage

- The template VM can be distributed to several physical hosts in the cluster to customize the VMs.

- Basically, there are four steps to deploy a group of VMs onto a target cluster:
  - preparing the disk image,
  - configuring the VMs,
  - choosing the destination nodes, and
  - executing the VM deployment command on every host.

- Every VM is configured with
  - a name,
  - disk image,
  - network setting, and
  - allocated CPU and memory.

- Each VM configuration is recorded into a file.
- Most configuration items use the same settings, while some of them, such as UUID, VM name, and IP address, are assigned with automatically calculated values.

- The deployment principle is to fulfill the VM requirement and to balance workloads among the whole host network.

3.4.2 Live VM Migration Steps and Performance Effects

- In a cluster built with mixed nodes of host and guest systems, the normal method of operation is to run everything on the physical machine.
- Virtual clusters can be applied in computational grids, cloud platforms, and high-performance computing (HPC) systems.
- Virtual clustering provides dynamic resources that can be quickly put together upon user demand or after a node failure.
- In particular, virtual clustering plays a key role in cloud computing.

- There are four ways to manage a virtual cluster.
    - cluster manager resides on a guest system
    - Cluster manager resides on the host systems. The host-based manager supervises the guest systems and can restart the guest system on another physical machine.
    - Use an independent cluster manager on both the host and guest systems – issue – makes infrastructure management more complex.
    - Use an integrated cluster on the guest and host systems. This means the manager must be designed to distinguish between virtualized resources and physical resources.

- A VM can be in one of the following four states.
    - An **inactive state** is defined by the virtualization platform, under which the VM is not enabled.
    - An **active state** refers to a VM that has been instantiated at the virtualization platform to perform a real task.
    - A **paused state** corresponds to a VM that has been instantiated but disabled to process a task or paused in a waiting state.
    - A VM enters the **suspended state** if its machine file and virtual resources are stored back to the disk.

- VMs can be live-migrated from one physical machine to another;
- When a VM fails, one VM can be replaced by another VM on a different node, as long as they both run with the same guest OS.
- The migration copies the VM state file from the storage area to the host machine

- Figure 3.20 shows the process of life migration of a VM from host A to host B. (six steps)

- Steps 0 and 1: Start migration.
    - Makes preparations for the migration, including determining the migrating VM and the destination host.
    - Migration –
        - manual done by user or
        - automatically started by strategies such as load balancing and server consolidation.

- Steps 2: Transfer memory.
    - Since the whole execution state of the VM is stored in memory, sending the VM's memory to the destination node ensures continuity of the service provided by the VM.
    - All of the memory data is transferred in the first round, and
    - Then the **migration controller** recopies the memory data which is changed in the last round.

- These steps keep iterating until all modified data is copied to destination node.

- Step 3: Suspend the VM and copy the last portion of the data.
  - The migrating VM's execution is suspended when the last round's memory data is transferred.
  - Other non-memory data such as CPU and network states are sent.
  - During this step, the VM is stopped and its applications will no longer run.
  - This "service unavailable" time is called the "downtime" of migration, which should be as short as possible so that it can be negligible to users.

- Steps 4 and 5: Commit and activate the new host.
  - After all the needed data is copied to the destination host, the VM reloads the states and recovers the execution of programs in the destination host, and the service provided by this VM continues.
  - Redirect the network connection to the new VM and clear the dependency to the source host.
  - Finally remove the original VM from the source host.

- When a VM runs a live service, it is necessary to ensure that the migration has
  - negligible downtime,
  - the lowest network bandwidth consumption possible, and
  - a reasonable total migration time.
  - Migration should not disrupt other active services residing in the same host through resource contention (e.g., CPU, network bandwidth).

3.4.3 Migration of Memory, Files, and Network Resources

- Since clusters have a high initial cost of ownership,
  - (including space, power conditioning, and cooling equipment,)
  - leasing or sharing access to a common cluster is an attractive solution when demands vary over time.
- Shared clusters offer economies of scale and more effective utilization of resources by multiplexing.
- Early configuration and management systems focus on expressive and scalable mechanisms for defining clusters for specific types of service, and physically partition cluster nodes among those types.
- When one system migrates to another physical node, we should consider the following issues.
  - Memory Migration
  - File System Migration
  - Network Migration
  - Live Migration of VM Using Xen

-

3.4.3.1 Memory Migration

- This is one of the most important aspects of VM migration.
- Moving the memory instance of a VM from one physical host to another
    - depend upon the characteristics of application/workloads supported by the guest OS.

- Memory migration can be in a range of hundreds of megabytes to a few gigabytes in a typical system today, and
- it needs to be done in an efficient manner.

- The **Internet Suspend-Resume (ISR)** technique exploits temporal locality as memory states are likely to have considerable overlap in the suspended and the resumed instances of a VM.

- **Temporal locality** refers to the fact that the memory states differ only by the amount of work done since a VM was last suspended before being initiated for migration.

- To exploit temporal locality, each file in the file system is represented as a **tree of small subfiles**.
- A copy of this tree exists in both the suspended and resumed VM instances.
- The advantage of using a tree-based representation of files is that the caching ensures the transmission of **only those files which have been changed**.

- The ISR technique deals with situations where the migration of live machines is not a necessity.
- Predictably, the downtime (the period during which the service is unavailable due to there being no currently executing instance of a VM) is high, compared to some of the other techniques discussed later.

3.4.3.2 File System Migration

- To support VM migration, a system must provide each VM with a consistent, location-independent view of the file system that is available on all hosts.

Solution 1

- A simple way to achieve this is to provide each VM with its own virtual disk which the file system is mapped to and transport the contents of this virtual disk along with the other states of the VM.
- However, due to the current trend of high capacity disks, migration of the contents of an entire disk over a network is not a viable solution.

Solution 2

- Another way is to have a global file system across all machines where a VM could be located. This way removes the need to copy files from one machine to another because all files are network accessible.

Solution 3

- A distributed file system is used in ISR serving as a transport mechanism for propagating a suspended VM state.

- The actual file systems themselves are not mapped onto the distributed file system.
- Instead, the VMM only accesses its local file system.
- The relevant VM files are explicitly copied into the local file system for a resume operation and taken out of the local file system for a suspend operation.
- This approach relieves developers from the complexities of implementing several different file system calls for different distributed file systems.
- It also essentially disassociates the VMM from any particular distributed file system semantics.
- However, this decoupling means that the VMM has to store the contents of each VM's virtual disks in its local files, which have to be moved around with the other state information of that VM.

Solution 4

- In smart copying, the VMM exploits **spatial locality**.
- to transmit only the **difference between the two file systems at suspending and resuming locations.**
- This technique significantly reduces the amount of actual physical data that has to be moved.

Solution 5 – proactive state transfer solution – predict new location

- In situations where there is no locality to exploit, a different approach is to **synthesize much of the state at the resuming site**.
- On many systems, user files only form a small fraction of the actual data on disk.
- Operating system and application software account for the majority of storage space.
- The proactive state transfer solution works in those cases where the resuming site can be predicted with reasonable confidence.

3.4.3.3 Network Migration

- A migrating VM should maintain all open network connections without relying on forwarding mechanisms on the original host or on support from mobility or redirection mechanisms.

- To enable remote systems to locate and communicate with a VM, each VM must be assigned a **virtual IP address** known to other entities.
- This address can be distinct from the IP address of the host machine where the VM is currently located.
- Each VM can also have its own distinct **virtual MAC address**.
- The VMM maintains a mapping of the virtual IP and MAC addresses to their corresponding VMs. (*ARP Table*)

Solution 1 – Virtual IP and MAC address

- In general, a migrating VM includes all the protocol states and carries its IP address with it.
- If the source and destination machines of a VM migration are typically connected to a single switched LAN, an unsolicited ARP reply from the migrating host is provided advertising that the IP has moved to a new location.
- This solves the open network connection problem by reconfiguring all the peers to send future packets to a new location.
- Although a few packets that have already been transmitted might be lost, there are no other problems with this mechanism.

Solution 2 – Switched Networks

- Alternatively, on a switched network, the migrating OS can keep its original Ethernet MAC address and rely on the network switch to detect its move to a new port.

Solution 3 – Live migration

- Live migration means moving a VM from one physical node to another while keeping its OS environment and applications unbroken.
- This capability is used to provide efficient
    - online system maintenance,
    - reconfiguration,
    - load balancing, and
    - proactive fault tolerance.

- It also provides desirable features to satisfy requirements for **computing resources** in modern computing systems, including **server consolidation, performance isolation, and ease of management.**
- Note:
    - Traditional migration suspends VMs before the transportation and then resumes them at the end of the process.
    - By importing the precopy mechanism, a VM could be live migrated without stopping the VM and keep the applications running during the migration.

Solution 3 – Cluster Environment with network-accessible storage system, like storage area network (SAN) or network attached storage (NAS)

- Only memory and CPU status needs to be transferred from the source node to the target node.

Solution 4 – precopy approach

- Precopy approach – first transfers all memory pages, and then only copies modified pages during the last round iteratively.

Solution 5 – Postcopy approach

- Here, all memory pages are transferred only once during the whole migration process and the baseline total migration time is reduced.
- But the downtime is much higher than that of precopy due to the latency of fetching pages from the source node before the VM can be resumed on the target.

- With the advent of multicore or many-core machines, abundant CPU resources are available which can be used to compress page frames and the amount of transferred data can be significantly reduced.
- Memory compression algorithms typically have little memory overhead.
- Decompression is also simple and very fast

3.4.4 Dynamic Deployment of Virtual Clusters

- Table 3.5 summarizes four virtual cluster research projects –We briefly introduce them here just to identify their design objectives and reported results.
    - Cellular Disco at Stanford is a virtual cluster built in a shared-memory multiprocessor system.

- INRIA virtual cluster was built to test parallel algorithm performance.
- COD was developed at Duke University to support dynamic resource allocation with a virtual cluster management system.
- VIOLIN clusters was built at Purdue University using multiple VM clustering to prove the advantage of dynamic adapt