MODULE IV: FILE SYSTEMS AND I/O SYSTEMS

File system - Concept of file and directory - Various file operations - File organization concepts – sequential and indexed. Different directory structures – single level, two-level, and tree structured directories. - Different allocation methods – contiguous, linked and indexed allocations. Virtualization : Need of virtualization – cost , administration , fast deployment , reduce infrastructure cost – limitations.. Types of hardware virtualization: Full virtualization - partial virtualization - paravirtualization. Desktop virtualization: Software virtualization – Memory virtualization - Storage virtualization – Data virtualization – Network virtualization..Vmware features and infrastructure – Virtual Box - Thin client

File

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

File directories:

Collection of files is a file directory. The directory contains information about the files, including attributes, location and ownership. Much of this information, especially that is concerned with storage, is managed by the operating system.

Fundamental components of a file:

- **Name:** Name is the symbolic file name and is the only information kept in human readable form.
- **Identifier:** This unique tag is a number that identifies the file within the file system; it is in non-human-readable form of the file.
- **Type:** This information is needed for systems which support different types of files or its format.
- **Location:** This information is a pointer to a device which points to the location of the file on the device where it is stored.
- **Size:** The current size of the file (which is in bytes, words, etc.) which possibly the maximum allowed size gets included in this attribute.
- **Protection:** Access-control information establishes who can do the reading, writing, executing, etc.
- **Date, Time & user identification:** This information might be kept for the creation of the file, its last modification and last used.

File operations:

A file is an abstract data type. The operating system can provide system calls to create, write, read, reposition, delete, and truncate files. There are six basic file operations within an Operating system. These are:

- **Creating a file:** There are two steps necessary for creating a file. First, space in the file system must be found for the file. We discuss how to allocate space for the file. Second, an entry for the new file must be made in the directory.

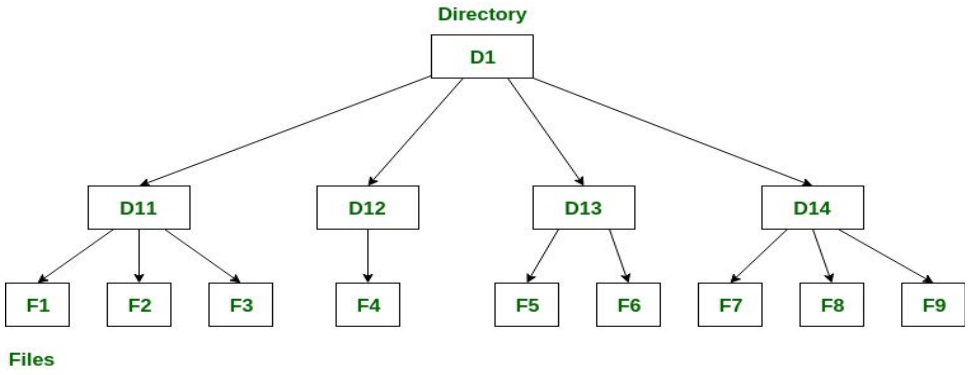
- **Writing a file:** To write to a file, you make a system call specify both the name of the file along with the information to be written to the file.
- **Reading a file**: To read from a file, you use a system call which specifies the name of the file and where within memory the next block of the file should be placed.
- **Repositioning inside a file**: First, the directory is searched for the file and the current position of the file is changed by the new position.
- **Deleting a file**: For deleting a file, you have to search the directory for the specific file. Deleting that file or directory releases all file space so that other files can re-use that space.
- **Truncating a file**: The user may wish for erasing the contents of a file but keep the attributes the same. Rather than deleting the file and then recreate it, this utility allows all attributes to remain unchanged except the file length and let the user add or edit the file content.

Advantages of maintaining directories are:

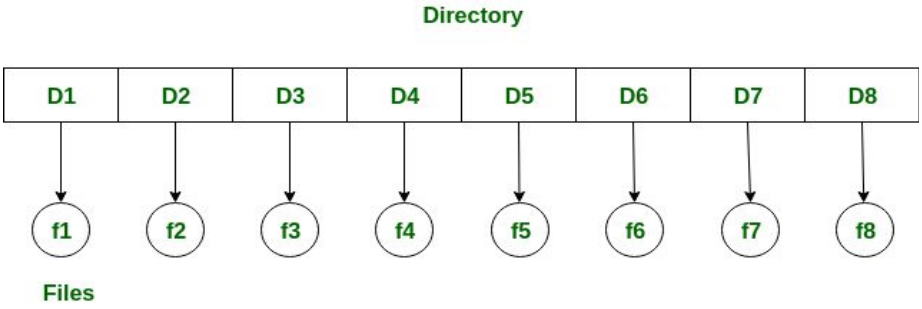
- **Efficiency:** A file can be located more quickly.
- **Naming:** It becomes convenient for users as two users can have the same name for different files or may have different names for the same file.
- **Grouping:** Logical grouping of files can be done by properties e.g. all java programs, all games etc.

Directory Structures in Operating System

Directory is a container that is used to contain folders and file. It organizes files and folders into a hierarchical manner.



1. **Single-level directory –**
   Single level directory is the simplest directory structure.In it all files are contained in the same directory which make it easy to support and understand.
   A single level directory has a significant limitation, however, when the number of files increases or when the system has more than one user. Since all the files are in the same directory, they must have the unique name . if two users call their dataset test, then the unique name rule is violated.



Advantages:

- Since it is a single directory, so its implementation is very easy.
- If the files are smaller in size, searching will become faster.
- The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

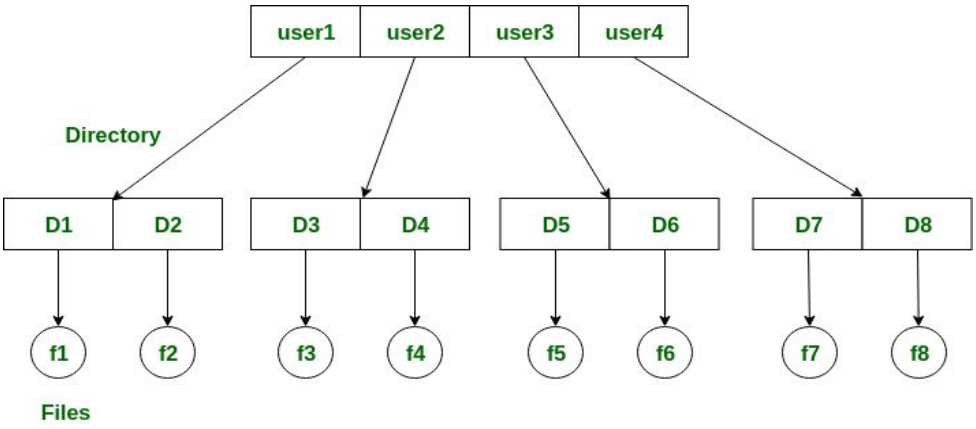
Disadvantages:

- There may be a chance of name collision because two files can not have the same name.
- Searching will become time taking if the directory is large.
- This can not group the same type of files together.

2. **Two-level directory –**

As we have seen, a single level directory often leads to confusion of files names among different users. The solution to this problem is to create a separate directory for each user.

In the two-level directory structure, each user has their own *user files directory (UFD)*. The UFDs have similar structures, but each lists only the files of a single user. The system's *master file directory (MFD)* is searched whenever a new user is logged in.



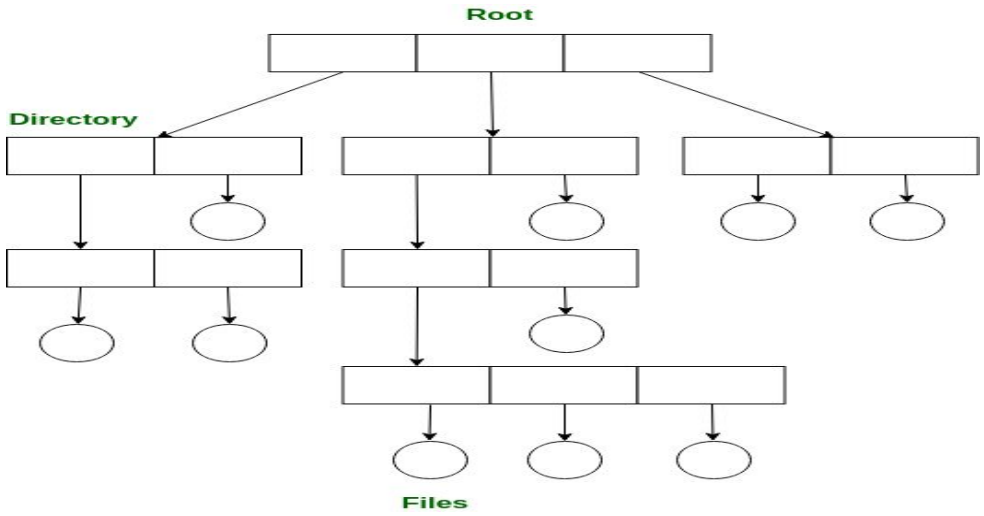
Advantages:

- We can give a full path like /User-name/directory-name/.
- Different users can have the same directory as well as file name.
- Searching for files becomes more easy due to path name and user-grouping.

Disadvantages:

- A user is not allowed to share files with other users.
- Still it is not very scalable, two files of the same type cannot be grouped together in the same user.

3. **Tree-structured directory –**
   A tree structure is the most common directory structure. The tree has a root directory, and every file in the system has a unique path.



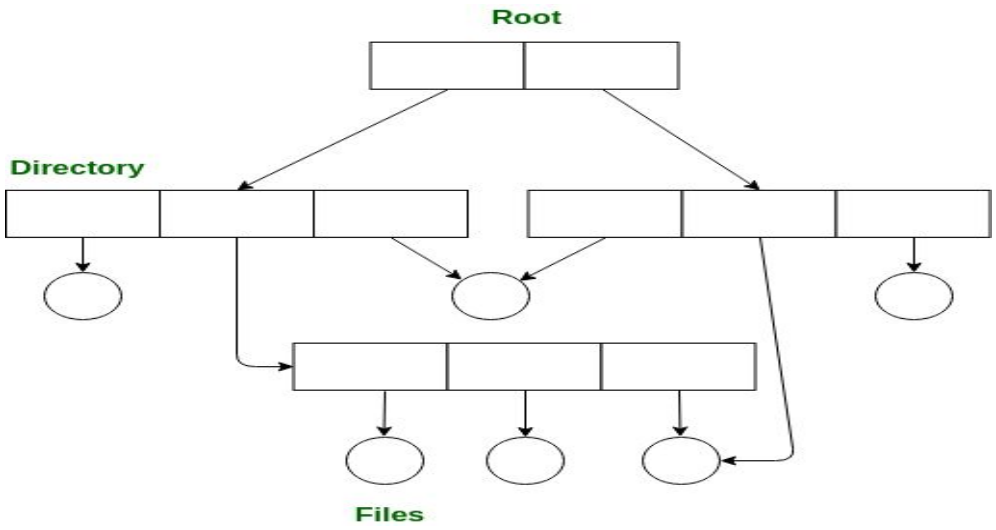
Advantages:

- Very generalize, since full path name can be given.
- Very scalable, the probability of name collision is less.
- Searching becomes very easy, we can use both absolute path as well as relative.

Disadvantages:

- Every file does not fit into the hierarchical model, files may be saved into multiple directories.
- We can not share files.
- It is inefficient, because accessing a file may go under multiple directories.

4. **Acyclic graph directory –**
   An acyclic graph is a graph with cycle and allows to share subdirectories and files. The same file or subdirectories may be in two different directories. It is a natural generalization of the tree-structured directory.
   It is the point to note that a shared file is not the same as a copy file . If any programmer makes some changes in the subdirectory it will reflect in both subdirectories.



Advantages:

- We can share files.
- Searching is easy due to different-different paths.

Disadvantages:

- We share the files via linking, in case of deleting it may create the problem,
- If the link is softlink then after deleting the file we left with a dangling pointer.
- In the case of hardlink, to delete a file we have to delete all the references associated with it.

File organization concepts

File organization refers to the way data is stored in a file. File organization is very important because it determines the methods of access, efficiency, flexibility and storage devices to use. There are four methods of organizing files on a storage media. This include:

- Sequential
- Random
- Indexed-sequential

1. **Sequential Access Method**
   - A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other. This access method is the most primitive one.
   - The idea of Sequential access is based on the tape model which is a sequential access device. We consider a Sequential access method is best because most of the records in a file are to be processed. For example, transaction files.

   Example: Compilers usually access files in this fashion.

Advantages

- The sorting makes it easy to access records.
- The binary chop technique can be used to reduce record search time by as much as half the time taken.

Disadvantages

- The sorting does not remove the need to access other records as the search looks for particular records.
- Sequential records cannot support modern technologies that require fast access to stored records.
- The requirement that all records be of the same size is sometimes difficult to enforce.

2. **Random or direct file organization**

- Random access file organization provides, accessing the records directly.
- Each record has its own address on the file with the help of which it can be directly accessed for reading or writing.
- The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.
- To access a file stored randomly, a record key is used to determine where a record is stored on the storage media.
- **Magnetic** and **optical** disks allow data to be stored and accessed randomly.
- Sometimes it is not necessary to process every record in a file. It is not necessary to process all the records in the order in which they are present in the memory. In all such cases, direct access is used.
- The disk is a direct access device which gives us the reliability to random access of any file block. In the file, there is a collection of physical blocks and the records of those blocks.

Advantages

- Quick retrieval of records.
- The records can be of different sizes.

3. Indexed sequential access

- The index sequential access method is a modification of the direct access method. Basically, it is a combination of both sequential access as well as direct access.
- The main idea of this method is to first access the file directly and then it accesses sequentially. In this access method, it is necessary for maintaining an index. The index is nothing but a pointer to a block. The direct access of the index is made to access a record in a file.

- The information which is obtained from this access is used to access the file. Sometimes the indexes are very big. So to maintain all these hierarchies of indexes are built in which one direct access of an index leads to information of another index access.

FILE ALLOCATION METHODS

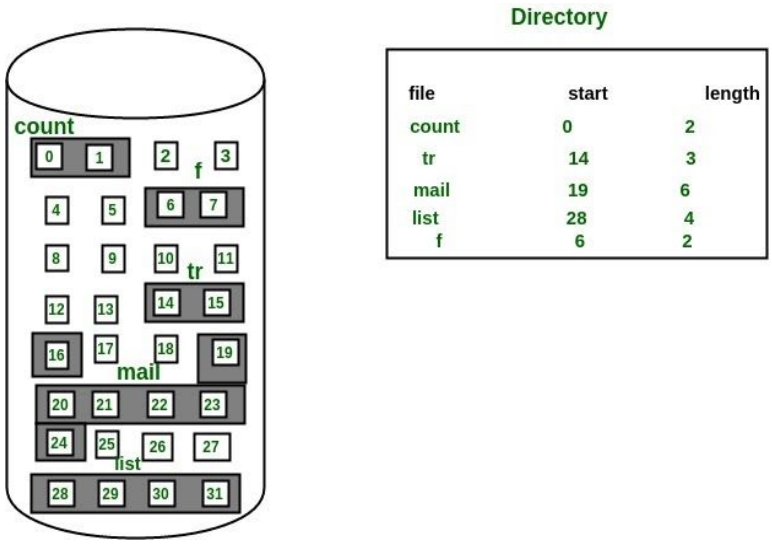
1. Contiguous Allocation

- In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: *b, b+1, b+2,……b+n-1.*
- This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file.

The directory entry for a file with contiguous allocation contains

- Address of starting block
- Length of the allocated portion.

The *file 'mail'* in the following figure starts from the block 19 with length = 6 blocks. Therefore, it occupies *19, 20, 21, 22, 23, 24* blocks.



Advantages:

- Both the Sequential and Direct Accesses are supported by this. For direct access, the address of the kth block of the file which starts at block b can easily be obtained as (b+k).
- This is extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.

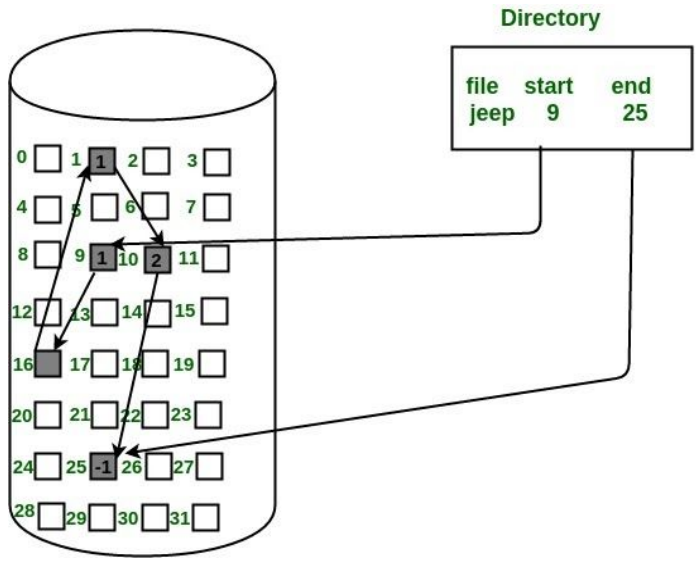
Disadvantages:

- This method suffers from both internal and external fragmentation. This makes it inefficient in terms of memory utilization.
- Increasing file size is difficult because it depends on the availability of contiguous memory at a particular instance.

2. Linked Allocation (Non-contiguous allocation)

- In this scheme, each file is a linked list of disk blocks which need not be contiguous. The disk blocks can be scattered anywhere on the disk.
- The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file.

The file 'jeep' in the following image shows how the blocks are randomly distributed. The last block (25) contains -1 indicating a null pointer and does not point to any other block.



Advantages:

- This is very flexible in terms of file size. File size can be increased easily since the system does not have to look for a contiguous chunk of memory.
- This method does not suffer from external fragmentation. This makes it relatively better in terms of memory utilization.

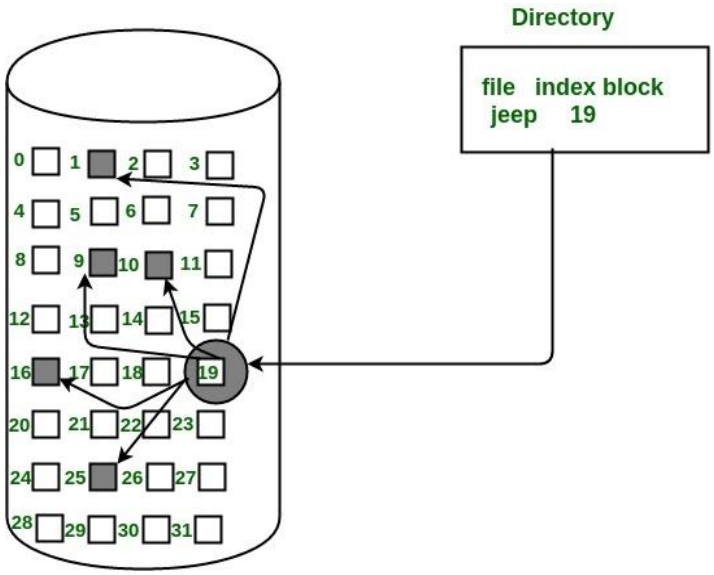
Disadvantages:

- Because the file blocks are distributed randomly on the disk, a large number of seeks are needed to access every block individually. This makes linked allocation slower.

- It does not support random or direct access. We can not directly access the blocks of a file. A block k of a file can be accessed by traversing k blocks sequentially (sequential access) from the starting block of the file via block pointers.
- Pointers required in the linked allocation incur some extra overhead.

3. Indexed Allocation

- In this scheme, a special block known as the **Index block** contains the pointers to all the blocks occupied by a file. Each file has its own index block.
- The ith entry in the index block contains the disk address of the ith file block. The directory entry contains the address of the index block as shown in the image:



Advantages:

- This supports direct access to the blocks occupied by the file and therefore provides fast access to the file blocks.
- It overcomes the problem of external fragmentation.

Disadvantages:

- The pointer overhead for indexed allocation is greater than linked allocation.
- For very small files, say files that expand only 2-3 blocks, the indexed allocation would keep one entire block (index block) for the pointers which is inefficient in terms of memory utilization. However, in linked allocation we lose the space of only 1 pointer per block.

Virtualization

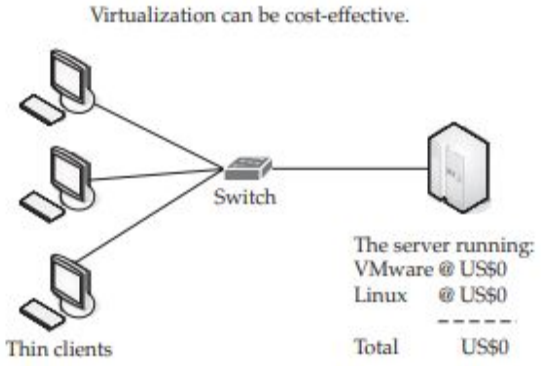
Operating system virtualizations include different users operating different applications on a single computer at a time.

- **Server virtualization** This is a method of partitioning a physical server computer into multiple servers so that each has the appearance and capabilities of running on its own dedicated machine. An example of this is VMware or Hyper-V
- **Application virtualization** This is a method that describes software technologies that separate them from the underlying operating system on which they are executed. A fully virtualized application is not installed in the traditional sense, although it still executes as though it were. The application is tricked at run time to believe that it is directly interfacing with the original OS and the resources it manages.
- **Presentation virtualization** This method isolates processing from the graphics and I/O, which makes it possible to run an application in one location (the server) but be controlled in another (the thin client). In this method, a virtual session is created and the applications project their interfaces onto the thin clients. It can either run a single application or present an entire desktop.

Need for Virtualization

1. **Cost**

   Depending on your solution, you can have a cost-free datacenter. You do have to shell out the money for the physical server itself, but there are options for free virtualization software and free operating systems. Microsoft's Virtual Server and VMware Server are free to download and install. If you use a licensed operating system, of course that will cost money. For instance, if you wanted five instances of Windows Server on that physical server, then you're going to have to pay for the licenses. That said, if you were to use a free version of Linux for the host and operating system, then all you've had to pay for is the physical server. Naturally, there is an element of "you get what you pay for." There's a reason most organizations have paid to install an OS on their systems. When you install a free OS, there is often a higher total cost of operation, because it can be more labor intensive to manage the OS and apply patches.



2. **Administration**

   Having all your servers in one place reduces your administrative burden. According to VMware, you can reduce your administrative burden from 1:10 to 1:30. What this means is that you can save time in your daily server administration or add more servers by having a virtualized environment. The following factors ease your administrative burdens:

   • A centralized console allows quicker access to servers.

   • CDs and DVDs can be quickly mounted using ISO files.

   • New servers can be quickly deployed.

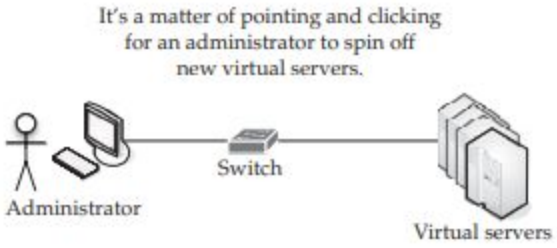
   • New virtual servers can be deployed more inexpensively than physical servers.

   • RAM can be quickly allocated for disk drives.

   • Virtual servers can be moved from one server to another.

3. **Fast Deployment**

   Because every virtual guest server is just a file on a disk, it's easy to copy (or clone) a system to create a new one. To copy an existing server, just copy the entire directory of the current virtual server.



   This can be used in the event the physical server fails, or if you want to test out a new application to ensure that it will work and play well with the other tools on your network. Virtualization software allows you to make clones of your work environment for these endeavors. Also, not everyone in your organization is going to be doing the same tasks. As such, you may want different work environments for different users. Virtualization allows you to do this.

4. **Reduced Infrastructure Costs**

We already talked about how you can cut costs by using free servers and clients, like Linux,as well as free distributions of Windows Virtual Server, Hyper-V, or VMware. But there are also reduced costs across your organization. If you reduce the number of physical servers you use, then you save money on hardware, cooling, and electricity. You also reduce the number of network ports, console video ports, mouse ports, and rack space.

Some of the savings you realize include

- Increased hardware utilization by as much as 70 percent
- Decreased hardware and software capital costs by as much as 40 percent
- Decreased operating costs by as much as 70 percent

Types of hardware virtualization

Hardware virtualization is of three kinds.

1. Full Virtualization: Here the hardware architecture is completely simulated. Guest software doesn't need any modification to run any applications.
2. partial virtualization: Here the virtual machine simulates the hardware & is independent. Furthermore, the guest OS doesn't require any modification.
3. Para-Virtualization: Here, the hardware is not simulated; instead the guest software runs its isolated system.

Paravirtualization is virtualization in which the guest operating system (the one being virtualized) is aware that it is a guest and accordingly has drivers that, instead of issuing hardware commands, simply issue commands directly to the host operating system. This also includes memory and thread management as well, which usually require unavailable privileged instructions in the processor.

Full Virtualization is virtualization in which the guest operating system is unaware that it is in a virtualized environment, and therefore hardware is virtualized by the host operating system so that the guest can issue commands to what it thinks is actual hardware, but really are just simulated hardware devices created by the host.

Partial virtualization, including address space virtualization, the virtual machine simulates multiple instances of much of an underlying hardware environment, particularly address spaces. Usually, this means that entire operating systems cannot run in the virtual machine—which would be the sign of full virtualization— but that many applications can run. A key form of partial virtualization is address space virtualization, in which each virtual machine consists of an independent address space. This capability requires address relocation hardware, and has been present in most practical examples of partial virtualization.

Desktop virtualization

Desktop virtualization, often called client virtualization. It is a virtualization technology used to separate a computer desktop environment from the physical computer. Desktop virtualization is considered a type of client-server computing model because the "virtualized" desktop is stored on a centralized, or remote, server and not the physical machine being virtualized.

Software virtualization

It is also called application virtualization. It is the practice of running software from a remote server. Software virtualization is similar to that of virtualization except that it is capable of abstracting the software installation procedure and creating virtual software installation. Many applications & their distributions became typical tasks for IT firms and departments. The mechanism for installing an application differs. So virtualized software is introduced which is an application that will be installed into its self-contained unit and provide software virtualization. Some of the examples are Virtual Box, VMware, etc.

Advantages of Software Virtualization

1) Client Deployments Become Easier: Copying a file to a workstation or linking a file in a network then we can easily install virtual software.

2) Easy to manage: To manage updates becomes a simpler task. You need to update at one place and deploy the updated virtual application to all clients.

3) Software Migration: Without software virtualization, moving from one software platform to another takes much time for deploying and impact on end user systems. With the help of a virtualized software environment the migration becomes easier.

Memory virtualization

memory virtualization decouples volatile random access memory (RAM) resources from individual systems in the data centre, and then aggregates those resources into a virtualized memory pool available to any computer in the cluster.The memory pool is accessed by the operating system or applications running on top of the operating system. The distributed memory pool can then be utilized as a high-speed cache, a messaging layer, or a large, shared memory resource for a CPU or a GPU application.

Memory virtualization allows networked, and therefore distributed, servers to share a pool of memory to overcome physical memory limitations.With this capability applications can take advantage of a very large amount of memory to improve overall performance, system utilization, increase memory usage efficiency, and enable new use cases. Software on the memory pool servers allows nodes to connect to the memory pool to contribute memory, and store and retrieve data. Management software and the technologies of memory overcommitment manage shared memory, data insertion, eviction and provisioning policies, data assignment to contributing

nodes, and handles requests from client nodes. The memory pool may be accessed at the application level or operating system level. At the application level, the pool is accessed through an API or as a networked file system to create a high-speed shared memory cache. At the operating system level, a page cache can utilize the pool as a very large memory resource that is much faster than local or networked storage.

Storage virtualization

Storage virtualization is also known as cloud storage. It is the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device.The process involves abstracting and covering the internal functions of a storage device from the host application, host servers or a general network in order to facilitate the application and network-independent management of storage.

The management of storage and data is becoming difficult and time consuming. Storage virtualization helps to address this problem by facilitating easy backup, archiving and recovery tasks by consuming less time. Storage virtualization aggregates the functions and hides the actual complexity of the storage area network (SAN).

Storage virtualization can be implemented by using software applications or appliances. There are three important reasons to implement storage virtualization:

1. Improved storage management in a heterogeneous IT environment
2. Better availability and estimation of down time with automated management
3. Better storage utilization

Data virtualization

Data virtualization is the process of aggregating data from different sources of information to develop a single, logical and virtual view of information so that it can be accessed by front-end solutions such as applications, dashboards and portals without having to know the data's exact storage location.

Many organizations run multiple types of database management systems, such as Oracle and SQL servers, which do not work well with one another. Therefore, enterprises face new challenges in data integration and storage of huge amounts of data. With data virtualization, business users are able to get real-time and reliable information quickly, which helps them to make major business decisions.

The process of data virtualization involves abstracting, transforming, federating and delivering data from disparate sources. The main goal of data virtualization technology is to provide a single point of access to the data by aggregating it from a wide range of data sources. This allows users to access the applications without having to know their exact location.

The most recent implementation of the data virtualization concept is in cloud computing technology.

Data virtualization software is often used in tasks such as:

- Data integration
- Business integration
- Service-oriented architecture data services
- Enterprise search

Network virtualization

Network virtualization refers to the management and monitoring of an entire computer network as a single administrative entity from a single software-based administrator's console. Network virtualization also may include storage virtualization, which involves managing all storage as a single resource. Network virtualization is designed to allow network optimization of data transfer rates, flexibility, scalability, reliability and security. It automates many network administrative tasks, which actually disguise a network's true complexity. All network servers and services are considered one pool of resources, which may be used without regard to the physical components. Network virtualization is especially useful for networks experiencing a rapid, large and unpredictable increase in usage.

The intended result of network virtualization is improved network productivity and efficiency, as well as job satisfaction for the network administrator.

Network virtualization involves dividing available bandwidth into independent channels, which are assigned, or reassigned, in real time to separate servers or network devices.

Network virtualization is accomplished by using a variety of hardware and software and combining network components. Software and hardware vendors combine components to offer external or internal network virtualization. The former combines local networks, or subdivides them into virtual networks, while the latter configures single systems with containers, creating a network in a box. Still other software vendors combine both types of network virtualization.

Vmware features and infrastructure

VMware ESX Server is a robust, production-proven virtualization layer that abstracts processor, memory, storage and networking resources into multiple virtual machines. **ESX Server** delivers the highest levels of performance, scalability and robustness required for enterprise IT environments. VMware offers its VMware Server, a free entry-level hosted virtualization product for Linux and Windows servers.

"Virtualization and VMware have become mainstream in the past year, and many customers have deployed thousands of VMware server environments across their enterprises. With VMware Server, we are ensuring that every company interested in, considering or evaluating server virtualization for the first time has access to the industry leading virtualization technology," said Diane Greene, VMware president. "VMware Server makes it easy and compelling for companies new to virtualization to take the first step toward enterprise-wide virtual infrastructure." Features VMware Server, the successor to VMware GSX Server, enables

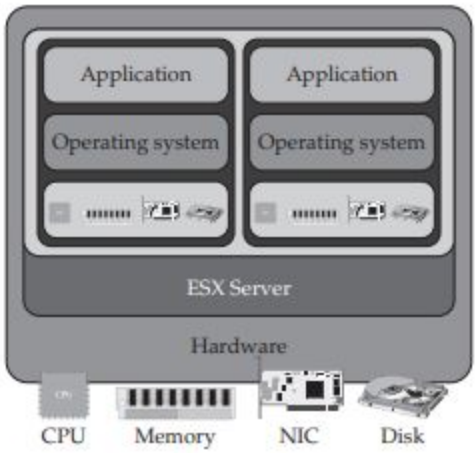
users to quickly provision new server capacity by partitioning a physical server into multiple virtual machines, bringing the powerful benefits of virtualization to every server.

VMware Server is feature-packed with the following market-leading capabilities:

- Support for any standard x86 hardware
- Support for a wide variety of Linux and Windows host operating systems, including 64-bit operating systems
- Support for a wide variety of Linux, NetWare, Solaris x86, and Windows guest
- operating systems, including 64-bit operating systems
- Support for Virtual SMP, enabling a single virtual machine to span multiple physical processors
- Quick and easy, wizard-driven installation similar to any desktop software
- Quick and easy virtual machine creation with a virtual machine wizard
- Virtual machine monitoring and management with an intuitive, user-friendly remote console

VMware Infrastructure

VMware is the biggest name in virtualization, and they offer VMware Infrastructure, which includes the latest version of VMware ESX Server 3.5 and VirtualCenter 2.5. VMware Infrastructure will allow VMware customers to streamline the management of IT environments.



Virtual Box

VirtualBox is open-source software for virtualizing the x86 computing architecture. It acts as a hypervisor, creating a VM (virtual machine) in which the user can run another OS (operating system).

Hypervisor is computer hardware, firmware, or software that generates and manages virtual machines. Its primary function is to allocate system resources properly to each virtual machine it manages, ensuring they all operate properly and efficiently.

The operating system in which VirtualBox runs is called the **"host" OS.** The operating system running in the VM is called the "guest" OS. VirtualBox supports Windows, Linux, or macOS as its host OS.

When configuring a virtual machine, the user can specify how many CPU cores, and how much RAM and disk space should be devoted to the VM. When the VM is running, it can be "paused." System execution is frozen at that moment in time, and the user can resume using it later.

Thin client

A thin client is a lightweight computer that has been optimized for establishing a remote connection with a server-based computing environment. The server does most of the work, which can include launching software programs, performing calculations, and storing data. This contrasts with a fat client or a conventional personal computer; the former is also intended for working in a client–server model but has significant local processing power, while the latter aims to perform its function mostly locally.

A **thin client** is a computer that runs from resources stored on a central server instead of a localized hard drive. Thin clients work by connecting remotely to a server-based computing environment where most applications, sensitive data, and memory, are stored.

What are the benefits of a thin client?

Thin clients have a number of benefits, including:

- Reduced cost
- Increased security
- More efficient manageability
- Scalability

Thin client deployment is more cost effective than deploying regular PCs. Because so much is centralized at the server-side, thin client computing can reduce IT support and licensing costs.

Security can be improved through employing thin clients because the thin client itself is restricted by the server. Thin clients cannot run unauthorized software, and data can’t be copied or saved anywhere except for the server. System monitoring and management is easier based on the centralized server location.

Thin clients can also be simpler to manage, since upgrades, security policies, and more can be managed in the data center instead of on the endpoint machines. This leads to less downtime, increasing productivity among IT staff as well as endpoint machine users.

Different uses of thin client

There are three ways a thin client can be used: shared services, desktop virtualization, or browser based.

With **shared terminal services**, all users at thin client stations share a server-based operating system and applications. Users of a shared services thin client are limited to simple tasks on their machine like creating folders, as well as running IT-approved applications.

Desktop virtualization, or UI processing, means that each desktop lives in a virtual machine, which is partitioned off from other virtual machines in the server. The operating system and applications are not shared resources, but they still physically live on a remote server. These virtualized resources can be accessed from any device that is able to connect to the server.

A **browser-based approach** to using thin clients means that an ordinary device connected to the internet carries out its application functions within a web browser instead of on a remote server. Data processing is done on the thin client machine, but software and data are retrieved from the network.