Virtualization simulates the interface to a physical object by any one of four means:

1. Multiplexing. Create multiple virtual objects from one instance of a physical object. For example, a

processor is multiplexed among a number of processes or threads.

2. Aggregation. Create one virtual object from multiple physical objects. For example, a number of

physical disks are aggregated into a RAID disk.

3. Emulation. Construct a virtual object from a different type of physical object. For example, a physical

disk emulates a random access memory.

4. Multiplexing and emulation. Examples: Virtual memory with paging multiplexes real memory and

disk, and a Virtual address emulates a real address; TCP emulates a reliable bit pipe and multiplexes

a physical communication channel and a processor.

Virtualization abstracts the underlying resources and simplifies their use, isolates users from one

another, and supports replication, which, in turn, increases the elasticity of the system. Virtualization is

a critical aspect of cloud computing, equally important to the providers and consumers of cloud services,

and plays an important role in:

• System security because it allows isolation of services running on the same hardware.

• Performance and reliability because it allows applications to migrate from one platform to another.

• The development and management of services offered by a provider.

• Performance isolation.

Virtualization has been used successfully since the late 1950s. A virtual memory based on paging

was first implemented on the Atlas computer at the University of Manchester in the United Kingdom in

1959. In a cloud computing environment a VMM runs on the physical hardware and exports hardware level

abstractions to one or more guest operating systems.A guest OS interacts with the virtual hardware

in the same way it would interact with the physical hardware, but under the watchful eye of the VMM

which traps all privileged operations and mediates the interactions of the guest OS with the hardware.

For example, a VMM can control I/O operations to two virtual disks implemented as two different sets

of tracks on a physical disk. New services can be added without the need to modify an operating system.

User convenience is a necessary condition for the success of the utility computing paradigms. One

of the multiple facets of user convenience is the ability to run remotely using the system software and

libraries required by the application. User convenience is a major advantage of a VM architecture over

a traditional operating system. For example, a user of the AmazonWeb Services (AWS) could submit an

Amazon Machine Image (AMI) containing the applications, libraries, data, and associated configuration

settings. The user could choose the operating system for the application, then start, terminate, and

monitor as many instances of the AMI as needed, using the Web Service APIs and the performance

monitoring and management tools provided by the AWS.

There are side effects of virtualization, notably the performance penalty and the hardware costs. As

we shall see shortly, all privileged operations of a VM must be trapped and validated by theVMM, which

ultimately controls system behavior; the increased overhead has a negative impact on performance. The

cost of the hardware for a VM is higher than the cost for a system running a traditional operating system

because the physical hardware is shared among a set of guest operating systems and it is typically

configured with faster and/or multicore processors, more memory, larger disks, and additional network

interfaces compared with a system running a traditional operating system.