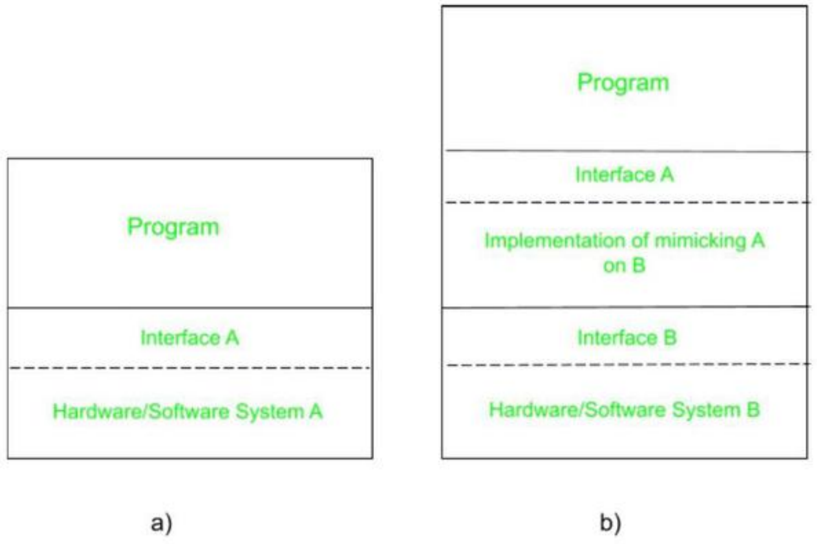
Virtualization in Distributed System

A virtual resource, such as a server, desktop, operating system, file, storage, or network, is created through the process of virtualization. Virtualization's main objective is to manage workloads by fundamentally altering conventional computing to make it more scalable. It is possible to think of threads and processes as ways to do multiple tasks simultaneously. We can construct (parts of) programmes that appear to run simultaneously thanks to them. Naturally, this simultaneous execution is fiction on a single-processor computer. A single thread or process will only have one instruction at a time because there is only one CPU. The illusion of parallelism is achieved by moving back and forth quickly between threads and processes.

Resource virtualization is a term used to describe the difference between having a single CPU and being able to pretend there are multiple CPUs.

The Role of Virtualization in Distributed Systems:

Every (distributed) computer system actually provides a programming interface to higher-level software, as can be seen in the below Fig (a). There are many various kinds of interfaces, from the fundamental instruction set that a CPU provides to the enormous selection of application programming interfaces that are included with many modern middleware systems. To emulate the behaviour of another system, virtualization essentially involves extending or replacing an existing interface, as seen in Fig (b).


Program
Interface A
Hardware/Software System A
a)
Program
Interface A
Implementation of mimicking A on B
Interface B
Hardware/Software System B
b)

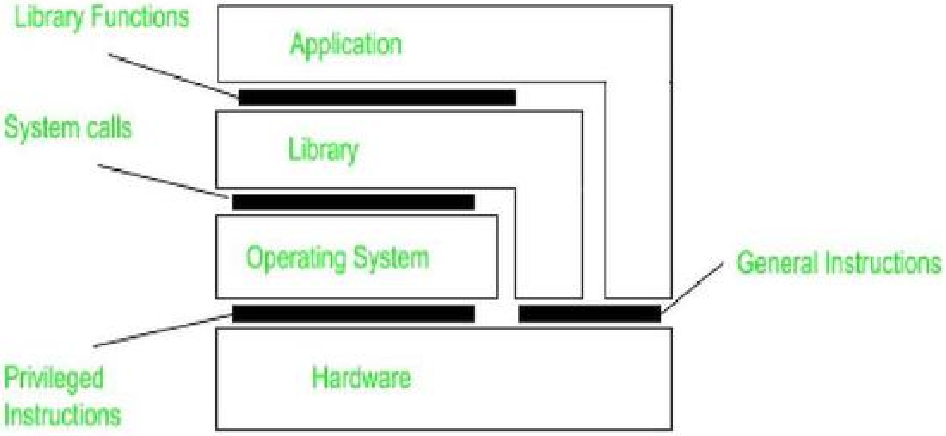
By effectively having each application run on its own virtual computer, possibly with the associated libraries and operating system, which in turn run on a common platform, virtualization can help minimise the diversity of platforms and machines. A high degree of portability and flexibility is offered by virtualization.

Architectures of Virtual Machines :

Understanding the variations in virtualization requires an understanding of the common four types of interfaces that computer systems provide, at four different levels.

1. A point of contact between hardware and software, made up of commands that can be executed by any application.
2. A point of contact between hardware and software that is made up of machine instructions that are only accessible to privileged programmes, such as an operating system.
3. An operating system interface that is made up of system calls.
4. An interface made up of library calls, which often constitutes an API (application programming interface) (API).

The following figure illustrates these various categories. The goal of virtualization is to replicate the functionality of these interfaces.



Virtualization can happen in one of two ways. First, we can create a run-time system, which effectively offers an abstract instruction set for use in running programmes. Instructions can be mimicked, as is done when running Windows applications on UNIX platforms, or they can be interpreted, as is the case for the Java runtime environment. Keep in mind that in the latter scenario, the emulator will also need to replicate how system calls behave, which has historically proven to be anything but simple. This kind of virtualization emphasises that virtualization is primarily only done for a single process by referring to a process virtual machine.

An alternative method of virtualization is to offer a system that is effectively constructed as a layer that entirely shields the original hardware while exposing the entire instruction set of that hardware (or other hardware) as an interface.

Article Tags : Computer Networks Technical Scripter Distributed System Technical Scripter 2022 virtualization

Recommended Articles

1. Virtualization | VMware: Full Virtualization
2. Distributed System - Thrashing in Distributed Shared Memory
3. Distributed System - Types of Distributed Deadlock
4. Operating system based Virtualization

5. How to Find Out the Virtualization Type in Linux System?
6. Distributed Consensus in Distributed Systems
7. What is Scalable System in Distributed System?
8. Linux Virtualization - Chroot Jail
9. Linux Virtualization : Resource throttling using cgroups
10. Linux Virtualization : Linux Containers (lxc)
11. Types of Server Virtualization in Computer Network
12. Virtualization in Cloud Computing and Types
13. Characteristics of Virtualization
14. Virtualization | A Machine Reference Model
15. On-Premises Cost Estimates of Virtualization
16. Need of Virtualization and its Reference Model
17. Hardware Based Virtualization
18. Virtualization | Xen: Paravirtualization
19. Difference between Cloud Computing and Virtualization
20. Difference between Cloud Computing and Virtualization
21. Server Virtualization
22. Network Virtualization in Cloud Computing
23. Pros and cons of Virtualization in Cloud Computing
24. Difference Between Docker and Virtualization
25. Virtualization Fundamentals in CCNA

Read Full Article

A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Open In App

Company

About Us

Legal

Careers

In Media

Contact Us

Advertise with us

Campus Training Program

Explore

Job-A-Thon Hiring Challenge

Hack-A-Thon

GfG Weekly Contest

Offline Classes (Delhi/NCR)

DSA in JAVA/C++

Master System Design

Master CP

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

DSA Concepts

Data Structures

Arrays

Strings

Linked List

Algorithms

Searching

Sorting

Mathematical

Dynamic Programming

DSA Roadmaps

DSA for Beginners

Basic DSA Coding Problems

DSA Roadmap by Sandeep Jain

DSA with JavaScript

Top 100 DSA Interview Problems

All Cheat Sheets

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

AngularJS

NodeJS

Express.js

Lodash

Computer Science

Python

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Deep Learning Tutorial

Competitive Programming

Top DSA for CP

Top 50 Tree Problems

Top 50 Graph Problems

Top 50 Array Problems

Top 50 String Problems

Top 50 DP Problems

Top 15 Websites for CP

Interview Corner

Company Wise Preparation

Preparation for SDE

Experienced Interviews

Internship Interviews

Competitive Programming

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

Python Interview Question

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

System Design

What is System Design

Monolithic and Distributed SD

Scalability in SD

Databases in SD

High Level Design or HLD

Low Level Design or LLD

Top SD Interview Questions

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

Aptitude Preparation

English Grammar

Commerce

Accountancy

Business Studies

Economics

Human Resource Management (HRM)

Management

Income Tax

Finance

Statistics for Economics

SSC/ BANKING

SSC CGL Syllabus

SBI PO Syllabus

SBI Clerk Syllabus

IBPS PO Syllabus

IBPS Clerk Syllabus

Aptitude Questions

SSC CGL Practice Papers

UPSC

Polity Notes

Geography Notes

History Notes

Science and Technology Notes

Economics Notes

Important Topics in Ethics

UPSC Previous Year Papers

Write & Earn

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships