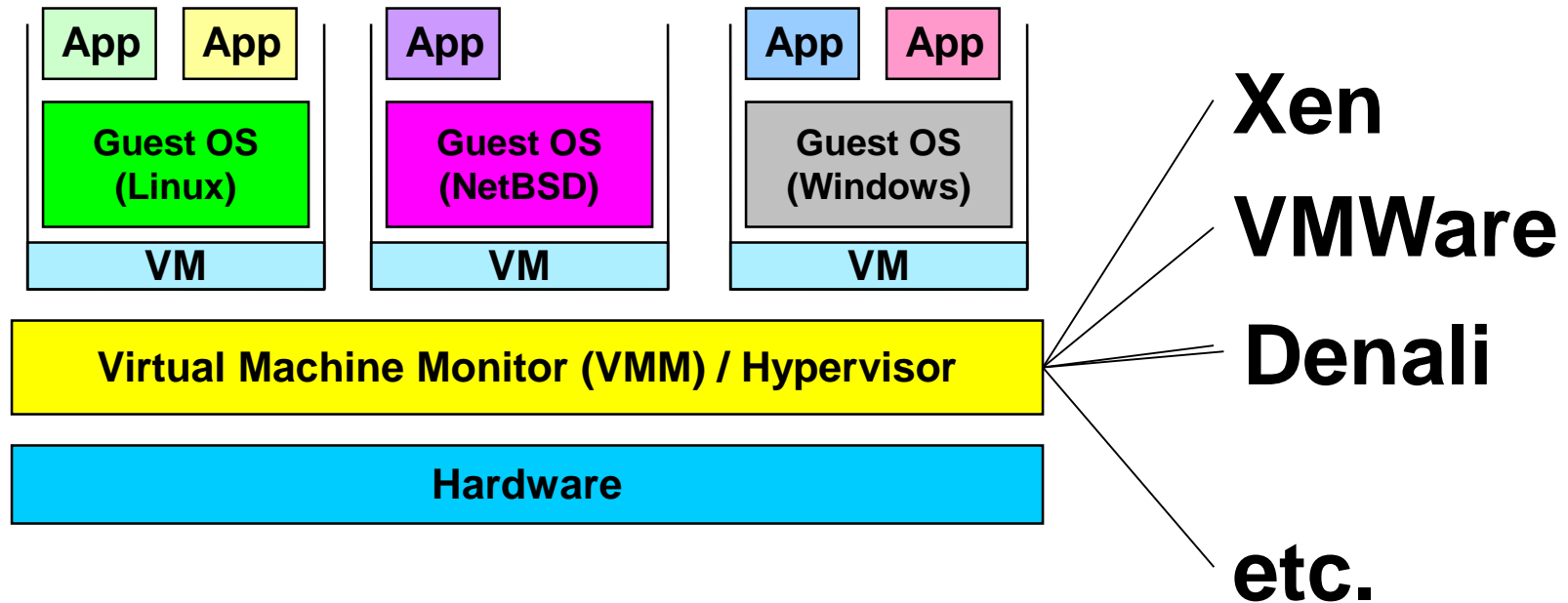


Virtualization

Virtual Machines

- VM technology allows multiple virtual machines to run on a single physical machine.



Virtualization in General

Advantages of virtual machines

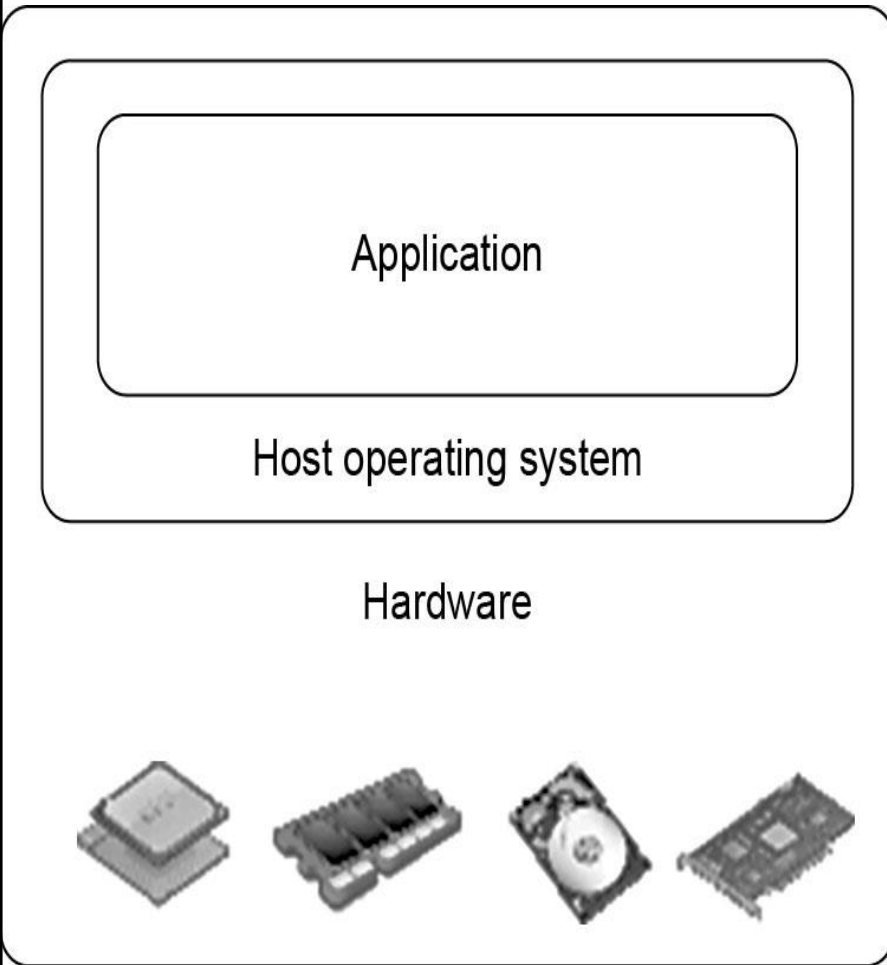
- Run **operating systems** where the physical hardware is unavailable,
- Easier to create new machines, backup machines, etc.,
- Software testing using “clean” installs of operating systems and software,
- **Emulate more** machines **than** are **physically available**,
- Timeshare lightly loaded systems on one host,
- Debug problems (suspend and resume the problem machine),
- Easy migration of virtual machines (shutdown needed or not).
- Run legacy systems.

Virtualization for Datacenter Automation

To serve millions of clients, simultaneously

- Server Consolidation in Virtualized Datacenter
- Virtual Storage Provisioning and De-provisioning
- Cloud Operating Systems for Virtual Datacenters
- Trust Management in virtualized Datacenters

Difference between Traditional Computer and Virtual machines



(a) Traditional computer

Virtual Machine, Guest Operating System, and VMM (Virtual Machine Monitor)

Virtual Machine

A representation of a real machine using software that provides an operating environment which can run or host a guest operating system.

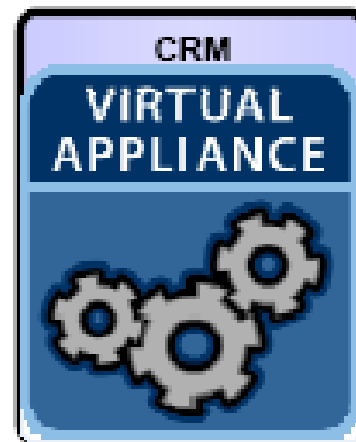
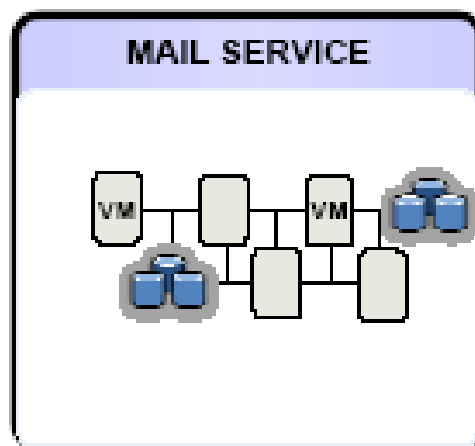
Guest operating system

An OS running in a virtual machine environment that would otherwise run directly on a separate physical system.

The Virtualization layer is the middleware between the underlying hardware and virtual machines represented in the system, also known as **Virtual machine monitor (VMM) or Hypervisor.**

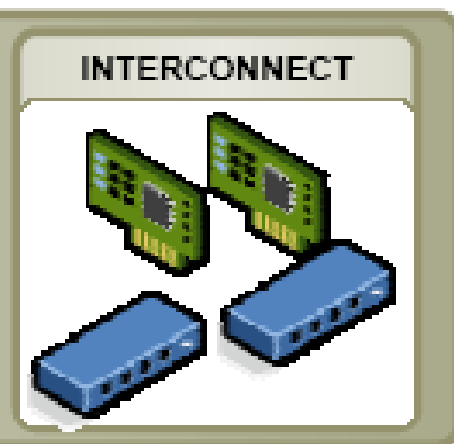
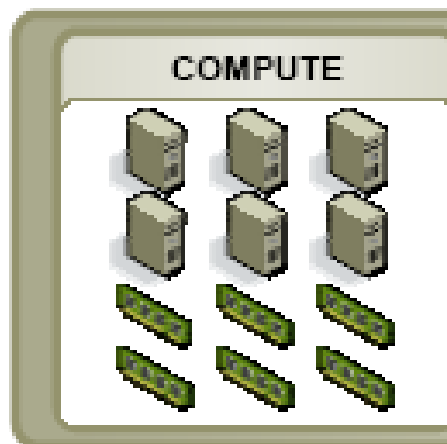
User's view of virtualization

LOGICAL VIEW



Virtualization Layer - Optimize HW utilization, power, etc.

PHYSICAL VIEW



Virtualization Ranging from Hardware to Applications in Five Abstraction Levels

Application level

JVM / .NET CLR / Panot

Library (user-level API) level

WINE/ WABI/ LxRun / Visual MainWin / vCUDA

Operating system level

Jail / Virtual Environment / Ensim's VPS / FVM

Hardware abstraction layer (HAL) level

VMware / Virtual PC / Denali / Xen / L4 /
Plex 86 / User mode Linux / Cooperative Linux

Instruction set architecture (ISA) level

Bochs / Crusoe / QEMU / BIRD / Dynamo

Virtualization at ISA (Instruction Set Architecture) level

Emulating a given ISA by the ISA of the host machine.

- e.g, MIPS (Microprocessor without Interlocked Pipeline Stages) binary code can run on an x-86-based host machine with the help of ISA emulation.
 - Typical systems: Bochs, Crusoe, Qemu, BIRD, Dynamo

Advantage:

- It can run a large amount of legacy binary codes written for various processors on any given new hardware host machines
- best application flexibility

Shortcoming & limitation:

- One source instruction may require tens or hundreds of native target instructions to perform its function, which is relatively slow.
- V-ISA requires adding a processor-specific software translation layer in the compiler.

Virtualization at Hardware Abstraction level

Virtualization is performed right on top of the hardware.

- It generates virtual hardware environments for VMs, and manages the underlying hardware through virtualization.
- Typical systems: **VMware, Virtual PC, Denali, Xen**

Advantage

- Has higher performance and good application isolation

Shortcoming & limitation

- Very expensive to implement (complexity)

Virtualization at Operating System (OS) level

It is an abstraction layer between traditional OS and user applications.

- This virtualization creates isolated containers on a single physical server and the OS-instance to utilize the hardware and software in datacenters.
- Typical systems: Jail / Virtual Environment / Ensim's VPS / FVM

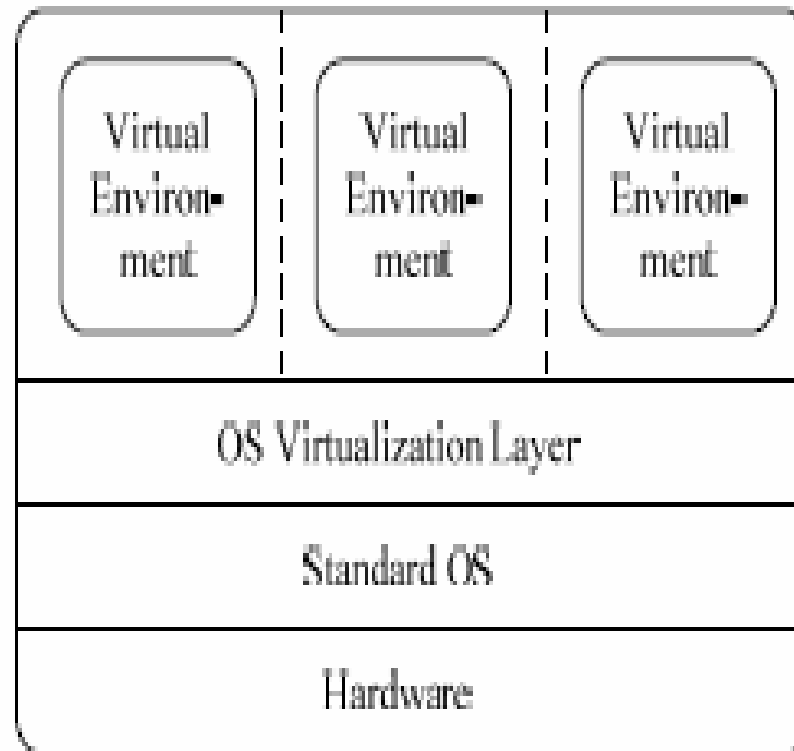
Advantage

- Has minimal startup/shutdown cost, low resource requirement, and high scalability, etc.

Shortcoming & limitation

- All VM's at the operating system level must have the same kind of guest OS
- Poor application flexibility and isolation.

Virtualization at OS Level



The virtualization layer is inserted inside an OS to partition the hardware resources for multiple VMs to run their applications in virtual environments

Most reported **OS-level virtualization systems** are **Linux-based**.

Virtualization Support for Linux and Windows NT Platforms

Virtualization Support and Source of Information

Linux vServer for Linux platforms (<http://linux-vserver.org/>)

OpenVZ for Linux platforms [65]; <http://ftp.openvz.org/doc/OpenVZ-Users-Guide.pdf>)

FVM (Feather-Weight Virtual Machines) for virtualizing the Windows NT platforms)

Brief Introduction on Functionality and Application Platforms

Extends Linux kernels to implement a security mechanism to help build VMs by setting resource limits and file attributes and changing the root environment for VM isolation

Supports virtualization by creating *virtual private servers (VPSes)*; the VPS has its own files, users, process tree, and virtual devices, which can be isolated from other VPSes, and checkpointing and live migration are supported

Uses system call interfaces to create VMs at the NY kernel space; multiple VMs are supported by virtualized namespace and copy-on-write

Library Support level

It creates execution environments for running alien programs on a platform rather than creating VM to run the entire operating system.

- It is done by API call interception and remapping.
- Typical systems: **Wine, WAB, LxRun , VisualMainWin, vCUDA**

Advantage

- It has very low implementation effort

Shortcoming & limitation

- poor application flexibility and isolation

Virtualization with Middleware/Library Support

Middleware and Library Support for Virtualization

Middleware or Runtime Library and References or Web Link

WABI (<http://docs.sun.com/app/docs/doc/802-6306>)

Lxrun (Linux Run) (<http://www.ugcs.caltech.edu/~steven/lxrun/>)

WINE (<http://www.winehq.org/>)

Visual MainWin (<http://www.mainsoft.com/>)

vCUDA (Example 3.2) (IEEE *IPDPS* 2009 [57])

Brief Introduction and Application Platforms

Middleware that converts Windows system calls running on x86 PCs to Solaris system calls running on SPARC workstations

A system call emulator that enables Linux applications written for x86 hosts to run on UNIX systems such as the SCO OpenServer

A library support system for virtualizing x86 processors to run Windows applications under Linux, FreeBSD, and Solaris

A compiler support system to develop Windows applications using Visual Studio to run on Solaris, Linux, and AIX hosts

Virtualization support for using general-purpose GPUs to run data-intensive applications under a special guest OS

User-Application level

It **virtualizes an application as a virtual machine.**

- Typical systems: **JVM , NET CLI , Panot**
- This layer sits as an application program on top of an operating system and exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.

Advantage

- Has the best application isolation

Shortcoming & limitation

- Low performance, low application flexibility and high implementation complexity.

Relative Merits of Virtualization at Various Levels

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

More Xs mean higher merit

Hypervisor

A hypervisor is a hardware virtualization technique allowing multiple operating systems, called guests to run on a host machine. This is also called the **Virtual Machine Monitor (VMM)**.

Type 1: bare metal hypervisor

- sits on the bare metal computer hardware like the CPU, memory, etc.
- All guest operating systems are a layer above the hypervisor.
- The original CP/CMS hypervisor developed by IBM was of this kind.

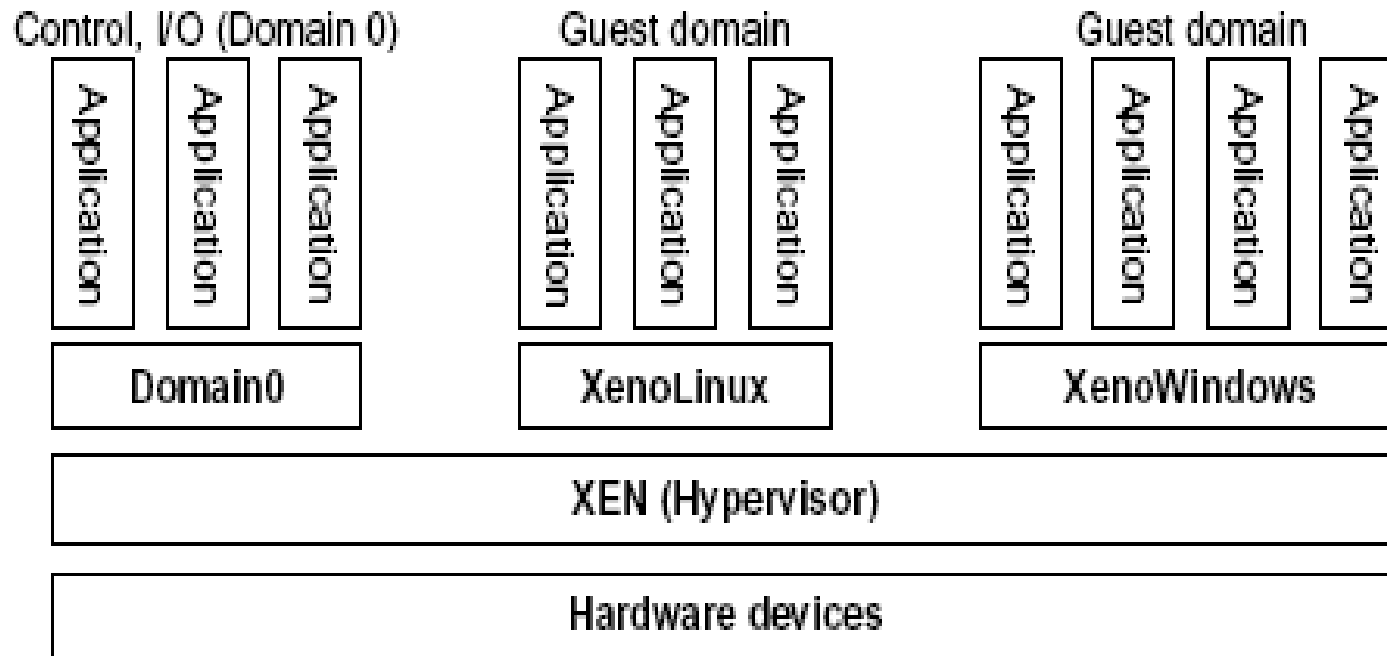
Type 2: hosted hypervisor

- Run over a host operating system.
- Hypervisor is the second layer over the hardware.
- Guest operating systems run a layer over the hypervisor.
- The OS is usually unaware of the virtualization

Major VMM and Hypervisor Providers

VMM Provider	Host CPU	Guest CPU	Host OS	Guest OS	VM Architecture
VMware Work-station	X86, x86-64	X86, x86-64	Windows, Linux	Windows, Linux, Solaris, FreeBSD, Netware, OS/2, SCO, BeOS, Darwin	Full Virtualization
VMware ESX Server	X86, x86-64	X86, x86-64	No host OS	The same as VMware workstation	Para-Virtualization
XEN	X86, x86-64, IA-64	X86, x86-64, IA-64	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, windows XP and 2003 Server	Hypervisor
KVM	X86, x86-64, IA64, S390, PowerPC	X86, x86-64, IA64, S390, PowerPC	Linux	Linux, Windows, FreeBSD, Solaris	Para-Virtualization

The XEN Architecture



The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.

The XEN Architecture

- Xen is an open source hypervisor developed by Cambridge University.
- Based on Linux

Core components:

- Hypervisor,
- Kernel,
- applications

Full Virtualization vs. Para-Virtualization

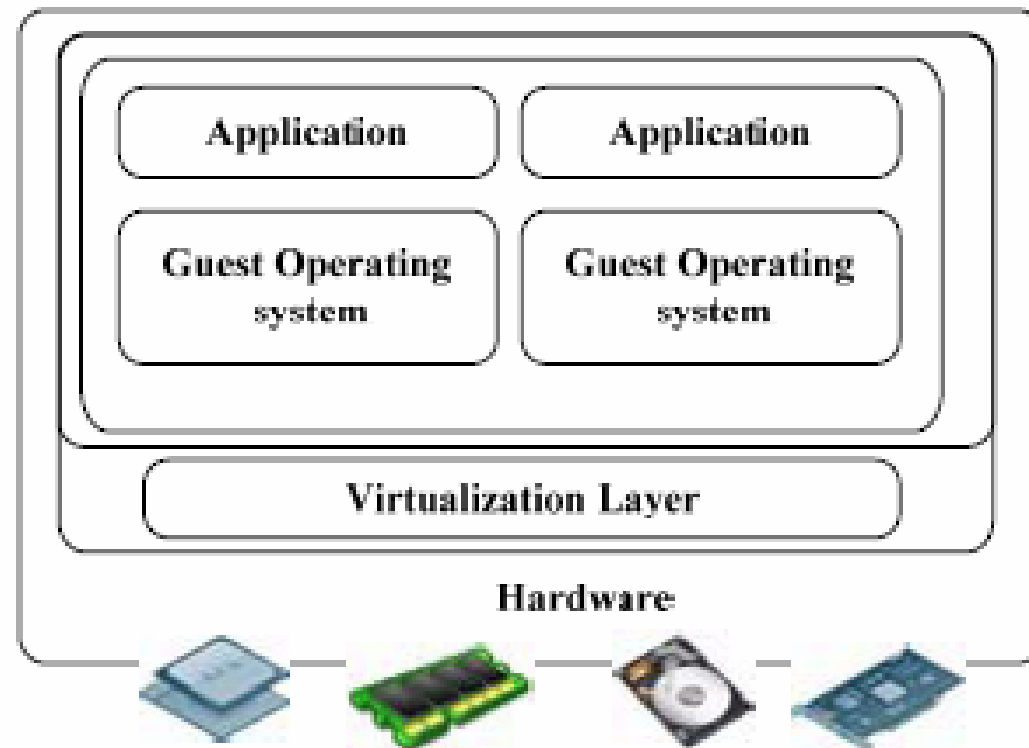
Full virtualization

- **Does not need to modify guest OS**, and **critical instructions are emulated by software through the use of binary translation.**
- **VMware Workstation** applies full virtualization, which uses binary translation to automatically modify x86 software on-the-fly to replace critical instructions.
- Advantage: no need to modify OS.
- **Disadvantage: binary translation slows down the performance.**

Para virtualization

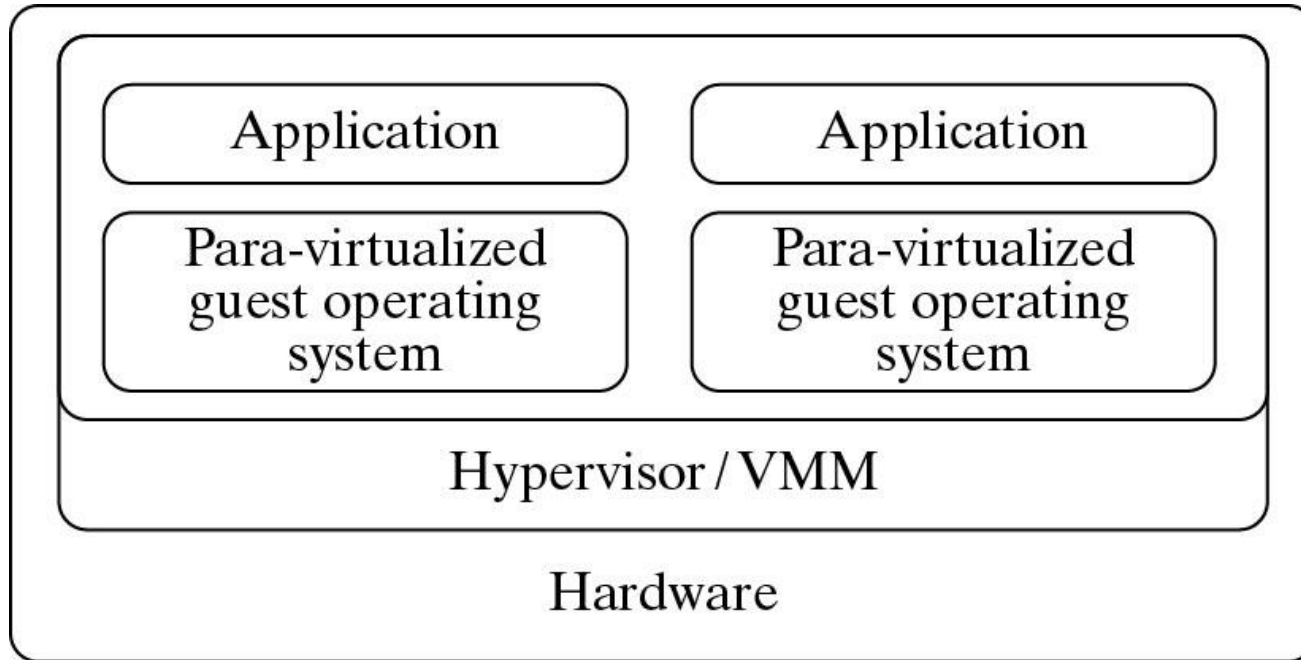
- **Para virtualization must modify guest OS**, non-virtualizable instructions are replaced by hypercalls that communicate directly with the hypervisor or VMM.
- Reduces the overhead, but cost of maintaining a paravirtualized OS is high.
- *Para virtualization is supported by **Xen, Denali and VMware ESX.***

Full Virtualization



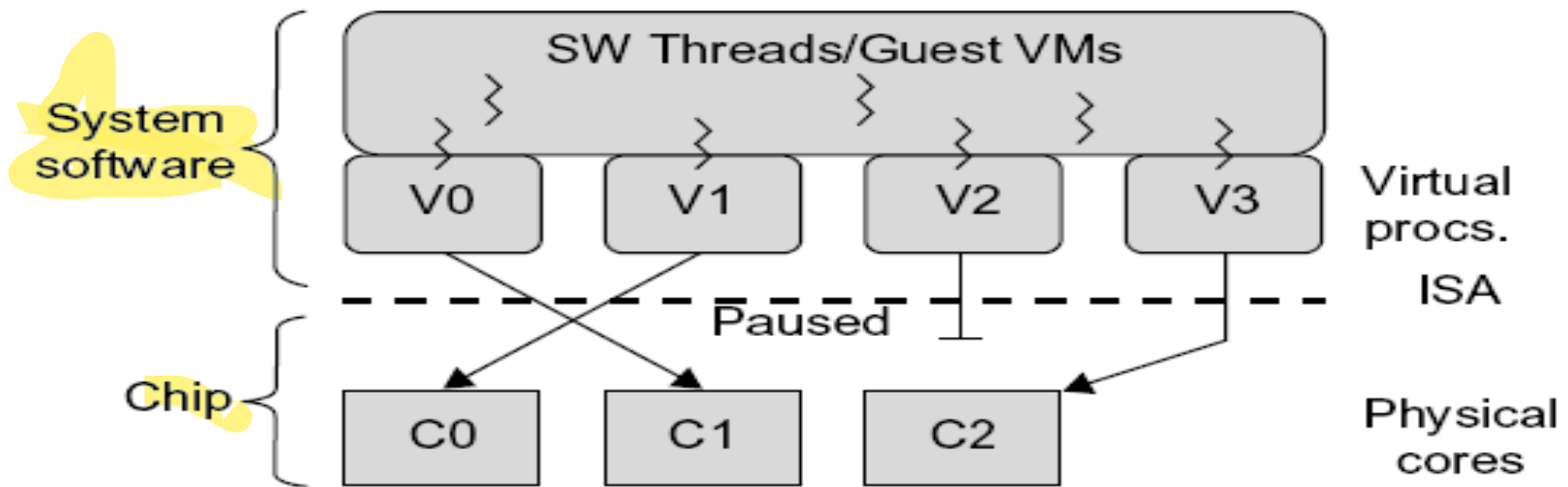
The concept of full virtualization using a hypervisor or a VMM directly sitting on top of the bare hardware devices.

Para- Virtualization with Compiler Support.



The KVM builds offers kernel-based VM on the Linux platform, based on para-virtualization

Multi-Core Virtualization: VCPU vs. traditional CPU



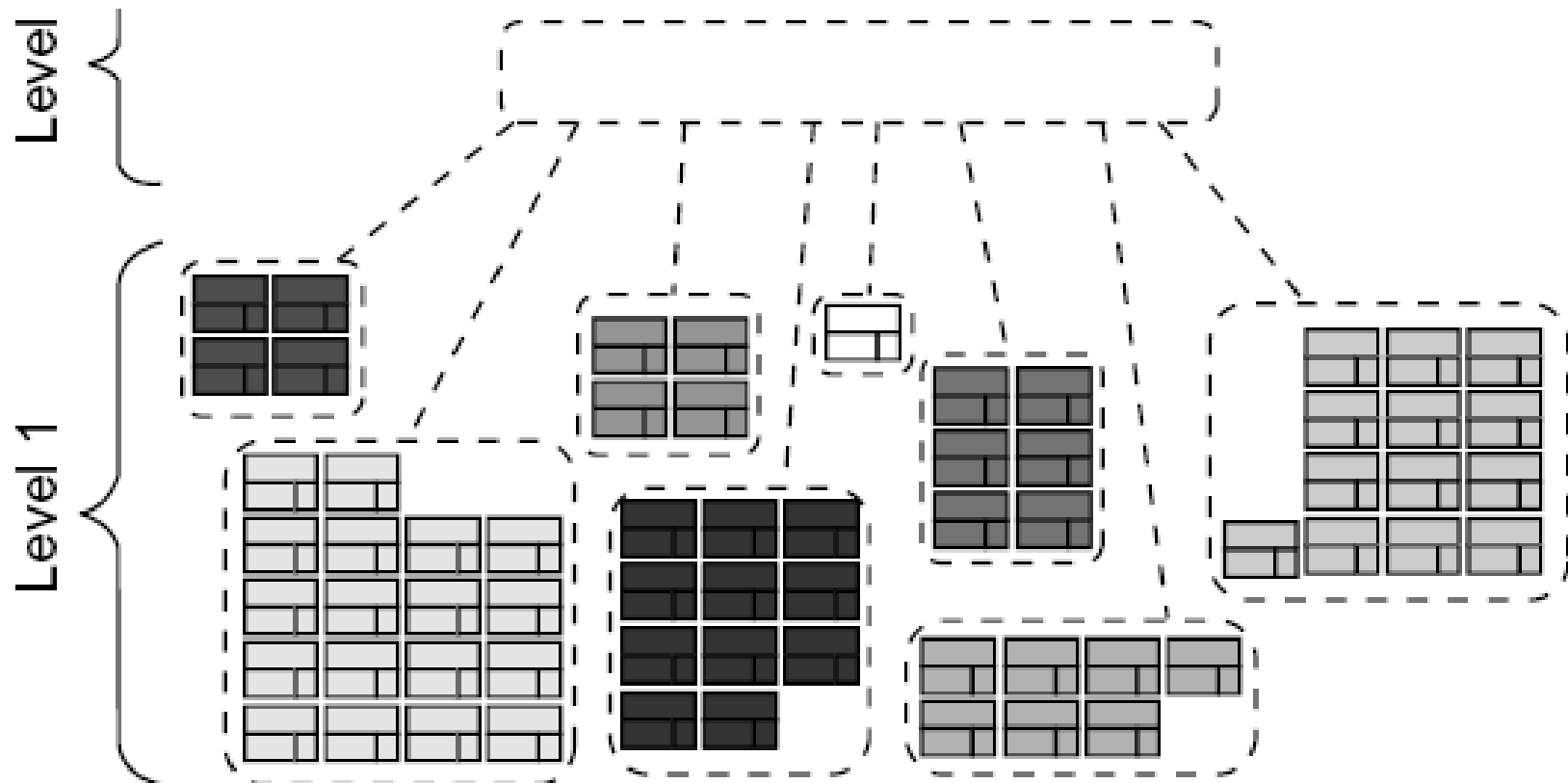
Four VCPUs are exposed to the software, only three cores are actually present. VCPUs V0, V1, and V3 have been transparently migrated, while VCPU V2 has been transparently suspended.

Virtual Cores vs. Physical Processor Cores

Physical cores	Virtual cores
The actual physical cores present in the processor.	There can be more virtual cores visible to a single OS than there are physical cores.
More burden on the software to write applications which can run directly on the cores.	Design of software becomes easier as the hardware assists the software in dynamic resource utilization.
Hardware provides no assistance to the software and is hence simpler.	Hardware provides assistance to the software and is hence more complex.
Poor resource management.	Better resource management.
The lowest level of system software has to be modified.	The lowest level of system software need not be modified.

Virtual Clusters in Many Cores

Space Sharing of VMs -- Virtual Hierarchy

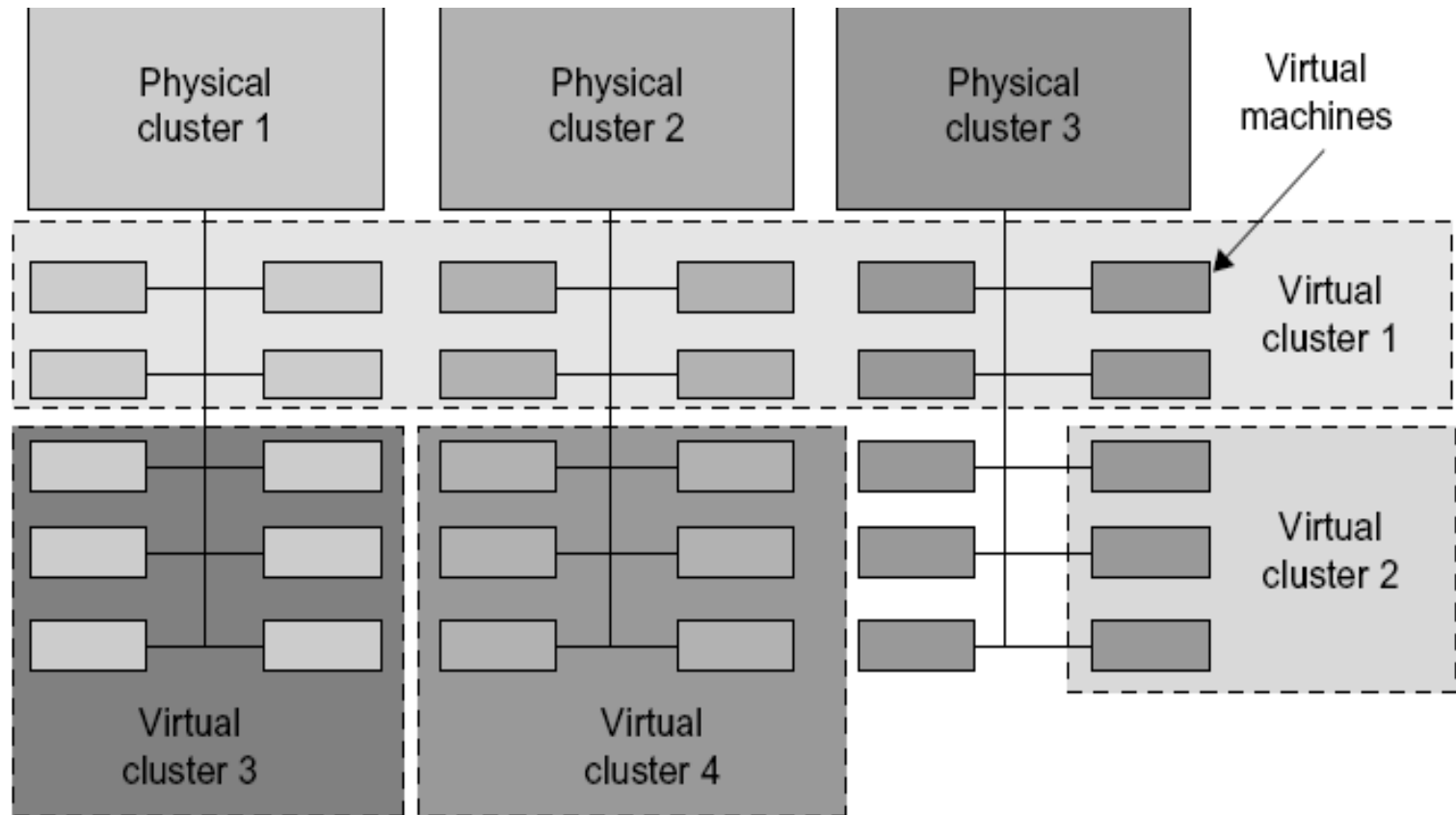



(b) Multiple virtual clusters assigned to various workloads

Virtual Cluster Characteristics

- The virtual cluster nodes can be either physical or virtual machines. Multiple VM's running with different OS's 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.
- VMs can be 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, similarly to the way an overlay network varies in size in a P2P network.

Virtual Clusters vs. Physical Clusters



A cloud platform with 4 virtual clusters over 3 physical clusters shaded differently. 

Virtual Cluster Projects

Experimental Results on Four Research Virtual Clusters

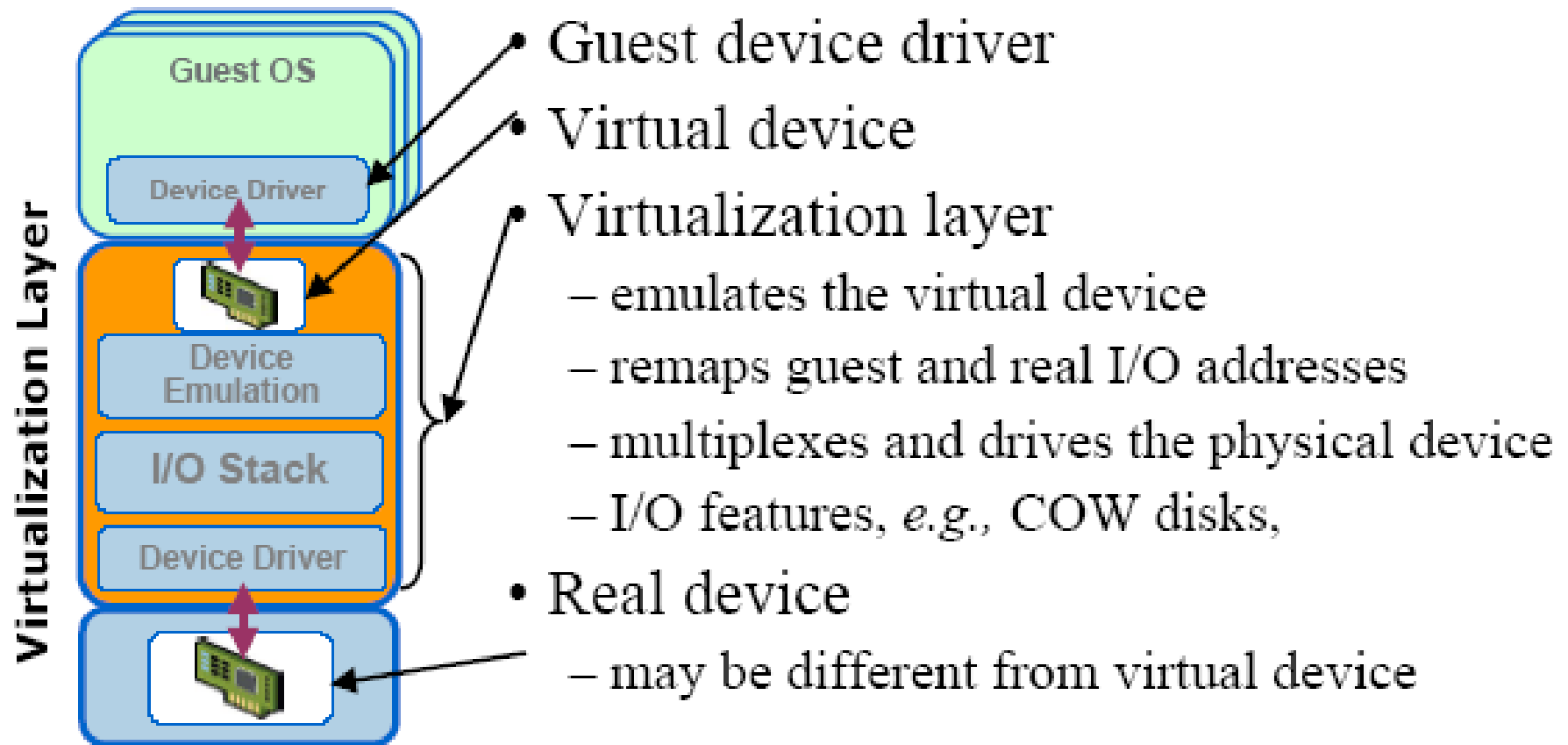
Project Name	Design Objectives	Reported Results and References
Cluster-on-Demand at Duke Univ.	Dynamic resource allocation with a virtual cluster management system	Sharing of VMs by multiple virtual clusters using Sun GridEngine [12]
Cellular Disco at Stanford Univ.	To deploy a virtual cluster on a shared-memory multiprocessor	VMs deployed on multiple processors under a VMM called Cellular Disco [8]
VIOLIN at Purdue Univ.	Multiple VM clustering to prove the advantage of dynamic adaptation	Reduce execution time of applications running VIOLIN with adaptation [25,55]
GRAAL Project at INRIA in France	Performance of parallel algorithms in Xen-enabled virtual clusters	75% of max. performance achieved with 30% resource slacks over VM clusters

Cloud OS for Building Private Clouds

VI Managers and Operating Systems for Virtualizing Data Centers

Manager/ OS, Platforms, License	Resources Being Virtualized, Web Link	Client API, Language	Hypervisors Used	Public Cloud Interface	Special Features
Nimbus Linux, Apache v2	VM creation, virtual cluster, www .nimbusproject.org/	EC2 WS, WSRF, CLI	Xen, KVM	EC2	Virtual networks
Eucalyptus Linux, BSD	Virtual networking (Example 3.12 and [41]), www .eucalyptus.com/	EC2 WS, CLI	Xen, KVM	EC2	Virtual networks
OpenNebula Linux, Apache v2	Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/	XML-RPC, CLI, Java	Xen, KVM	EC2, Elastic Host	Virtual networks, dynamic provisioning
vSphere 4 Linux, Windows, proprietary	Virtualizing OS for data centers (Example 3.13), www .vmware.com/ products/vsphere/ [66]	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vCloud partners	Data protection, vStorage, VMFS, DRM, HA

Current virtual I/O devices



CPU, Memory and I/O Virtualization

- CPU virtualization demands hardware-assisted traps of sensitive instructions by the VMM
- Memory virtualization demands special hardware support to help translate virtual address into physical address and machine memory in two stages.
- I/O virtualization is the most difficult one to realize due to the complexity of I/O service routines and the emulation needed between the guest OS and host OS.