Unit 2: Cloud Computing

Virtualization and Cloud Infrastructure

Course Material

Contents

1 Introduction to Virtualization **4**
- 1.1 Need for Virtualization . . . 4
  - 1.1.1 Comparison: Physical vs. Virtual Infrastructure . . . 4
- 1.2 What is a Virtual Machine? . . . 5

2 Virtualization Architecture **5**
- 2.1 Virtualization Layer Components . . . 5
- 2.2 Traditional vs. Virtualization Architecture . . . 5
  - 2.2.1 Key Differences . . . 5

3 Hypervisor Technology **6**
- 3.1 Types of Hypervisors . . . 6
  - 3.1.1 Type 1: Bare Metal Hypervisor . . . 6
  - 3.1.2 Type 2: Hosted Hypervisor . . . 7
  - 3.1.3 Type 3: Embedded Hypervisor . . . 7
- 3.2 Hypervisor Comparison . . . 8

4 Virtualization Techniques **8**
- 4.1 Full Virtualization . . . 8
- 4.2 Para-virtualization . . . 9
- 4.3 Full Virtualization vs. Para-virtualization . . . 10

5 Application-Level Virtualization **10**
- 5.1 Types of Application Virtualization . . . 10
  - 5.1.1 1. Application Streaming . . . 10
  - 5.1.2 2. Thin Client Technology . . . 11
  - 5.1.3 3. Terminal Services / Remote Desktop Virtualization . . . 11
  - 5.1.4 4. Desktop Virtualization (VDI - Virtual Desktop Infrastructure) 11
  - 5.1.5 5. Application Sandboxing . . . 12

6 Storage Virtualization **12**
- 6.1 Need for Storage Virtualization . . . 12
- 6.2 File Storage (NAS) vs. Block Storage (SAN) . . . 13
- 6.3 Storage Virtualization Architecture . . . 13
- 6.4 Storage Virtualization Concepts . . . 13
  - 6.4.1 Data Tiers . . . 13
  - 6.4.2 Storage Pooling . . . 13

7 Network Virtualization **14**
7.1 Traditional Network Virtualization 14
7.1.1 Traditional Network Virtualization (TNV) Characteristics 14
7.2 Modern Network Virtualization (MNV) 15
7.2.1 Key Features 15
7.3 Network Virtualization Scenarios 15
7.3.1 Case I: Same VLAN, Same Hypervisor 15
7.3.2 Case II: Same VLAN, Different Hypervisor 16
7.3.3 Case III: Different VLAN, Same Hypervisor 17
7.4 VLAN in Network Virtualization 17

8 Service Virtualization **17**
8.1 Components and Architecture 18
8.2 Key Concepts 18
8.2.1 VM Square (Service Catalog) 18
8.2.2 Data Commingling 18
8.2.3 Multi-tenancy 18
8.2.4 Service Level Agreement (SLA) 19
8.2.5 Service Monitoring 19
8.2.6 Subscription Model 19
8.3 Service Virtualization Benefits 20

9 Integration of Physical and Virtual Machines **20**
9.1 Hybrid Infrastructure 20
9.2 Integration Strategies 20
9.3 Use Cases for Physical Machines 21

10 Benefits of Virtualization **21**
10.1 Server Consolidation 21
10.2 Scalability 21
10.3 Good Degree of Customization 22
10.4 Additional Benefits 22

11 Virtualization Challenges and Considerations **22**
11.1 Performance Overhead 22
11.2 Security Considerations 23
11.3 License Complexity 23
11.4 Skills and Training 23

12 Future Trends in Virtualization **24**
12.1 Containers and Microservices 24
12.2 Edge Computing Virtualization 24
12.3 AI-Driven Virtualization Management 24
12.4 Serverless Computing 24

13 Summary **25**

14 Review Questions **25**

15 Glossary **26**

16 References **26**

1 Introduction to Virtualization

Virtualization is a fundamental technology that enables cloud computing by creating virtual versions of physical computing resources. This abstraction layer allows multiple virtual systems to run on a single physical machine, maximizing resource utilization and flexibility.

1.1 Need for Virtualization

The transition from physical to virtual infrastructure addresses several critical challenges in modern computing environments. Understanding these motivations is essential for appreciating the value virtualization brings to cloud computing.

1.1.1 Comparison: Physical vs. Virtual Infrastructure

Aspect	Physical Infrastructure	Virtual Infrastructure
Hardware Requirements	Requires dedicated workstations, servers, networking devices, storage devices, and other physical devices	No physical hardware required; resources are abstracted and pooled
Provisioning Time	Time-consuming and costly to provision new resources	Highly scalable; can be easily scaled up or down on demand
Scalability	Limited scalability due to physical constraints such as space, power, and cooling	Highly flexible; resources can be modified, reallocated dynamically
Physical Access	Requires physical access to hardware for maintenance and updates	No physical space required; can be accessed from anywhere remotely
Security Model	Relies on physical security measures such as access control and surveillance	Enhanced digital security measures such as encryption, firewalls, and access controls
Disaster Recovery	Data recovery can be lost and more difficult; requires manual backup procedures	Automated and cloud-based backup; easier disaster recovery
Resource Utilization	Under-utilization of resources	Efficient utilization through resource pooling and sharing
Cost Structure	High initial capital expenditure	Reduced capital expenditure; pay-as-you-go model

Table 1: Physical vs. Virtual Infrastructure Comparison

1.2 What is a Virtual Machine?

A **Virtual Machine (VM)** is a digitalized replication or emulation of a physical machine. Just as a physical machine has an operating system, applications, and hardware resources, a virtual machine provides the same functionality but in a software-defined environment.

Key Definition

A Virtual Machine creates an isolated environment that behaves like a complete computer system, with its own virtual CPU, memory, storage, and network interfaces, all running on top of a physical host system.

2 Virtualization Architecture

2.1 Virtualization Layer Components

The virtualization layer is responsible for three critical functions:

1. **Creation of Virtual Machines**: Instantiating new VM instances with specified resource allocations
2. **Allocation of Resources**: Distributing physical hardware resources (CPU, memory, storage, network) among virtual machines
3. **Coordination Among Virtual Machines**: Managing interactions, preventing conflicts, and ensuring isolation between VMs

2.2 Traditional vs. Virtualization Architecture

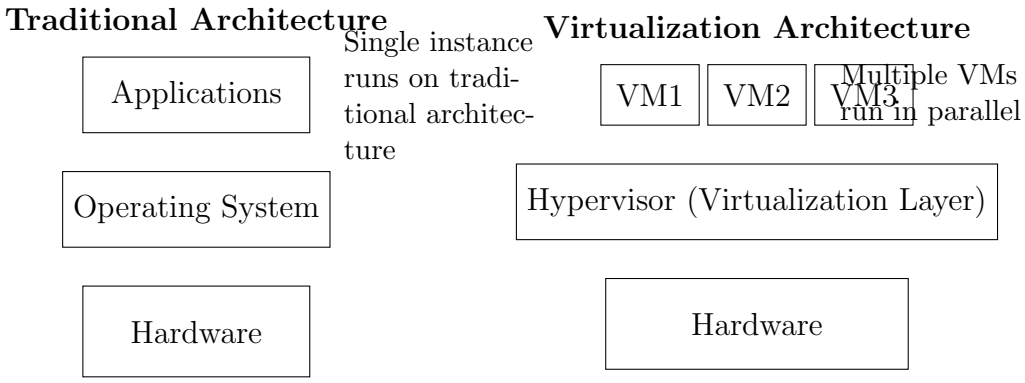


Figure 1: Traditional vs. Virtualization Architecture

2.2.1 Key Differences

- **Traditional Architecture**:
  - Only a single instance can run on the hardware

  - Hardware and software are tightly coupled
  - Under-utilization of resources

- **Virtualization Architecture**:
  - Multiple VMs run in parallel on the same hardware
  - Hardware and software are loosely coupled
  - Efficient utilization of resources
  - Different applications with different configurations can run on the same physical infrastructure

3 Hypervisor Technology

A **Hypervisor** (also known as a **Virtual Machine Monitor - VMM**) is a software program that manages and enables the execution of multiple operating systems on a single piece of physical hardware. The hypervisor is the core component that makes virtualization possible.

3.1 Types of Hypervisors

There are three main types of hypervisors, each with distinct characteristics and use cases:

3.1.1 Type 1: Bare Metal Hypervisor

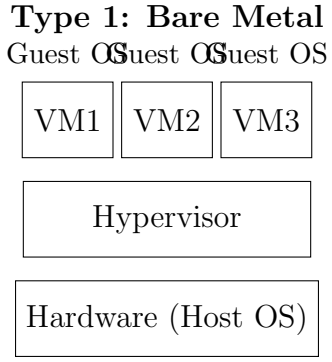


Figure 2: Type 1 Hypervisor Architecture

Characteristics:

- Runs directly on physical hardware
- VMs have direct access to hardware resources
- Higher performance due to reduced overhead
- Commonly used in enterprise data centers and cloud environments
- **Examples**: VMware ESXi, VMware vSphere, Microsoft Hyper-V, Xen

3.1.2 Type 2: Hosted Hypervisor

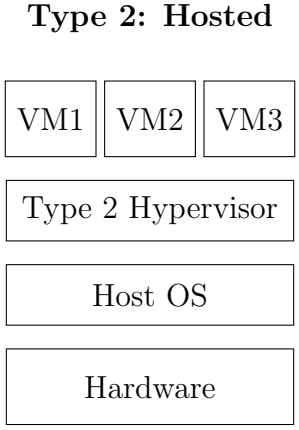


Figure 3: Type 2 Hypervisor Architecture

Characteristics:

- Runs on top of a host operating system
- Additional layer introduces some performance overhead
- Easier to set up and use for desktop virtualization
- Suitable for development, testing, and personal use
- **Examples**: VMware Workstation, Oracle VirtualBox, Parallels Desktop

3.1.3 Type 3: Embedded Hypervisor

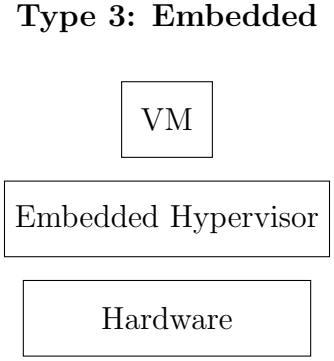


Figure 4: Embedded Hypervisor Architecture

Characteristics:

- Embedded directly into the firmware
- Provides direct conversion between hardware and VMs
- Part of the hardware itself

- Designed for specific embedded systems
- Offers high performance and isolation
- Especially used in automotive, manufacturing, and automated control systems
- **Use Cases**: Industrial automation, real-time systems, medical devices

3.2 Hypervisor Comparison

Feature	**Type 1 (Bare Metal)**	**Type 2 (Hosted)**	**Type 3 (Embedded)**
Performance	High	Moderate	Very High
Overhead	Low	Higher due to host OS	Minimal
Use Case	Data centers, enterprise	Desktop, development	Embedded systems, automotive
Setup Complexity	Complex	Easy	Specialized
Hardware Access	Direct	Through host OS	Direct, firmware-level

Table 2: Hypervisor Types Comparison

4 Virtualization Techniques

4.1 Full Virtualization

Full virtualization provides complete abstraction of the underlying hardware, allowing unmodified guest operating systems to run on virtual machines.

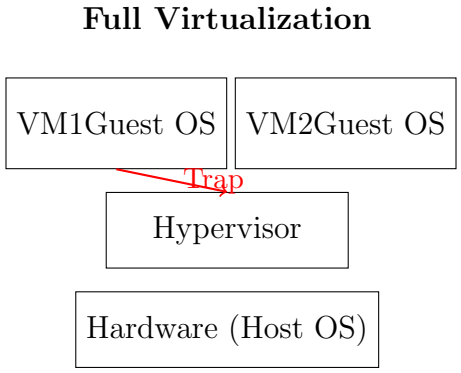


Figure 5: Full Virtualization Architecture

Characteristics:

- Virtual machines execute privileged instructions without direct hardware access

- Underlying hardware is not in the control of VMs
- Hypervisor decides which machine can access hardware resources
- Known as **"Trap and Emulate"** method
- Guest OS remains unmodified

Process Flow:

1. VM assumes that instruction is executed directly on underlying hardware
2. When a privileged instruction is attempted, it triggers a trap
3. Hypervisor intercepts the instruction
4. Hypervisor decides if the instruction affects underlying hardware
5. If safe, hypervisor emulates the instruction; otherwise, it's blocked
6. Result is returned to the VM

4.2 Para-virtualization

Para-virtualization involves modifying the guest operating system to be aware of the virtualization environment, enabling direct communication with the hypervisor.

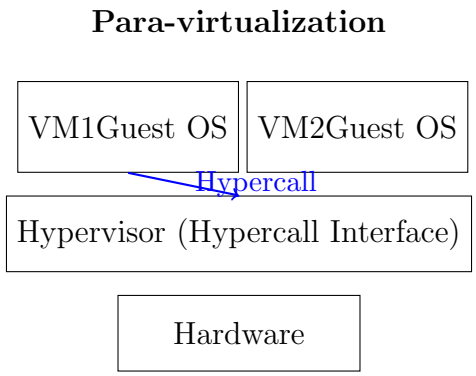


Figure 6: Para-virtualization Architecture

Characteristics:

- Virtual machines execute privileged instructions directly with underlying hardware
- For execution of any privileged instruction, VM communicates directly with hypervisor
- Uses **hypercalls** - a special API for guest-hypervisor communication
- Also called **"Hypercalls"** method
- Guest OS is modified to be virtualization-aware
- Better performance than full virtualization

Process Flow:

1. Guest OS sends hypercall to hypervisor
2. Hypervisor processes the request
3. Direct hardware interaction when appropriate
4. Results returned to guest OS

4.3 Full Virtualization vs. Para-virtualization

Aspect	Full Virtualization	Para-virtualization
Guest OS Modification	Not required	Required
Performance	Lower (due to trap overhead)	Higher (direct hypercalls)
Hardware Access	Indirect (through emulation)	More direct (through hypercalls)
Privileged Instructions	Trapped and emulated	Direct hypercalls
Complexity	Lower for guest OS	Higher (needs modification)
Compatibility	High (any unmodified OS)	Limited (only modified OS)

Table 3: Full Virtualization vs. Para-virtualization

5 Application-Level Virtualization

Application virtualization separates applications from the underlying operating system, enabling them to run in isolated environments without full OS virtualization.

5.1 Types of Application Virtualization

5.1.1 1. Application Streaming

Application streaming delivers applications to users on-demand without full installation on local machines.

Key Features:

- Applications run from a central server
- Only required components are streamed to client
- Reduces local storage requirements
- Centralized management and updates
- Examples: Microsoft App-V, Citrix Virtual Apps

5.1.2 2. Thin Client Technology

Thin Client Definition

An application virtualization approach where the application is independent of the operating system on a local device, and the UI only displays the results of the application execution.

Characteristics:

- Client requests are forwarded to a server location
- Server processes the request
- Only display information is sent back to client
- Installation, execution, and storage all happen on server
- Minimal client-side requirements

5.1.3 3. Terminal Services / Remote Desktop Virtualization

Terminal services allow multiple users to access applications running on a central server through remote desktop sessions.

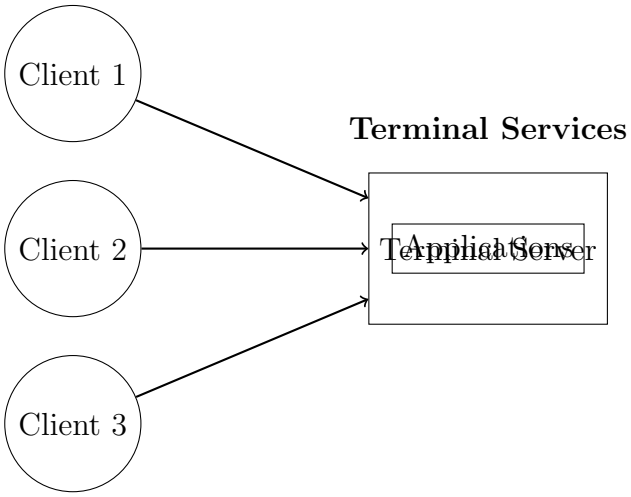


Figure 7: Terminal Services Architecture

5.1.4 4. Desktop Virtualization (VDI - Virtual Desktop Infrastructure)

Desktop virtualization provides full desktop environments that run in virtual machines on centralized servers.

Features:

- Complete desktop OS virtualized
- User accesses virtual desktop remotely
- Centralized management and security

- Consistent experience across devices
- Examples: VMware Horizon, Citrix Virtual Desktops

5.1.5 5. Application Sandboxing

Sandboxing isolates applications in controlled environments for testing and security purposes.

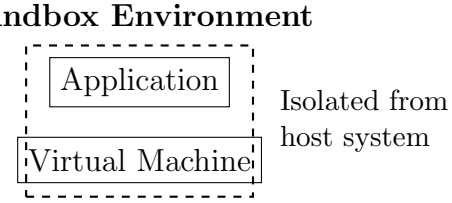


Figure 8: Application Sandboxing

Use Cases:

- Testing untrusted applications
- Malware analysis
- Development and debugging
- Security research
- Example: VMware ThinApp, Docker containers

6 Storage Virtualization

Storage virtualization abstracts physical storage resources to create a unified, logical storage pool that can be managed centrally and allocated dynamically.

6.1 Need for Storage Virtualization

- **Remote Access**: Access storage resources from anywhere
- **Extending Lifetime**: Maximize utilization of older storage devices
- **Centralized Management**: Easy storage management, backup, and archival
- **Scalability**: Add or remove storage without disruption
- **Resource Optimization**: Better utilization of available storage

Aspect	NAS (Network Attached Storage)	SAN (Storage Area Network)
Definition	File-level storage accessed over network	Block-level storage accessed over dedicated network
Data Organization	Stores, retrieves, and presents data in files and folders by setting up a hierarchy	Data is divided into fixed-size chunks (blocks) stored in different locations. Each chunk has a unique identifier (address)
Storage Type	File-based storage	Block-based storage
Protocol	Uses file-sharing protocols (NFS, SMB/CIFS)	Uses block protocols (iSCSI, Fibre Channel)
Performance	Moderate	High performance
Cost	More affordable	Highly expensive
Use Case	File sharing, collaboration, general storage	Databases, virtualization, high-performance applications
Scalability	Easy to scale	Scalable but complex

Table 4: NAS vs. SAN Comparison

6.2 File Storage (NAS) vs. Block Storage (SAN)

6.3 Storage Virtualization Architecture

6.4 Storage Virtualization Concepts

6.4.1 Data Tiers

- **Hotline Data**: Frequently accessed data requiring fast storage (SSD, high-performance storage)
- **Coldline Data**: Infrequently accessed data suitable for archival storage (tape, low-cost storage)

6.4.2 Storage Pooling

Storage pooling aggregates multiple physical storage devices into a single virtual storage resource.

Process:

1. Physical storage devices are divided into multiple chunks
2. Chunks are stored in different physical storage locations
3. User accesses data through the virtualization layer
4. System uses linked lists or mapping tables to track chunk locations

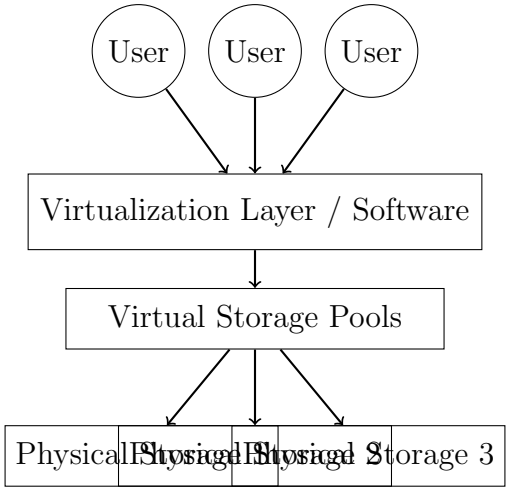


Figure 9: Storage Virtualization Architecture

5. Data is transparently retrieved from multiple locations

Benefits:

- Simplified management
- Easy to store, recover, backup, and archive
- Extends lifetime of old devices using advanced concepts
- Improved fault tolerance through redundancy
- Dynamic allocation of storage resources

7 Network Virtualization

Network virtualization creates virtual networks that operate independently of the underlying physical network infrastructure, enabling greater flexibility and efficient resource utilization.

7.1 Traditional Network Virtualization

7.1.1 Traditional Network Virtualization (TNV) Characteristics

- **Virtual Switch**: Software-based switch within hypervisor
- **Logical Switch**: Enables communication between VMs on same host
- **VLAN Support**: Virtual LANs for network segmentation
- **Limitation**: VMs on one hypervisor cannot communicate with VMs on another hypervisor using only virtual switches
- **Physical Dependency**: Communication between different hosts requires physical network infrastructure

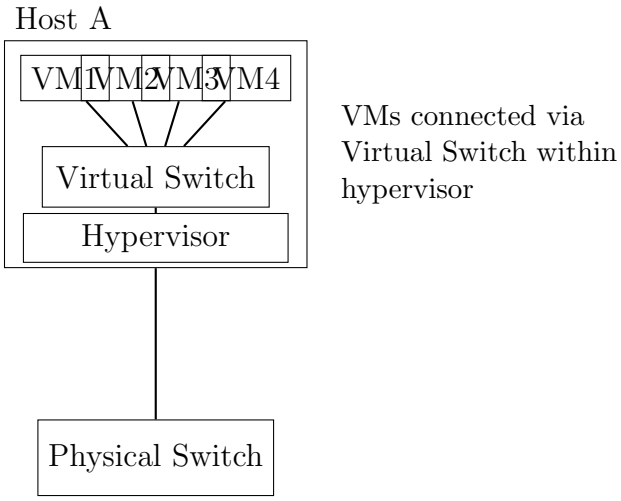


Figure 10: Traditional Network Virtualization

7.2 Modern Network Virtualization (MNV)

Modern Network Virtualization overcomes the limitations of traditional approaches by providing software-defined networking capabilities that span multiple hosts.

7.2.1 Key Features

- **Software-Defined**: Network topology defined in software
- **Cross-Host Communication**: VMs on different hypervisors can communicate through virtual networks
- **Encryption**: Secure communication between VMs across physical infrastructure
- **Dynamic Routing**: Intelligent path selection and load balancing
- **Examples**: VXLAN (Virtual Extensible LAN), VMware NSX

7.3 Network Virtualization Scenarios

7.3.1 Case I: Same VLAN, Same Hypervisor

Scenario: VM1 wants to communicate with VM3 (both on Host A, same VLAN)

Traditional Network Virtualization (TNV) Process:

1. Message from VM1 sent to Logical Virtual Switch (LVS)
2. Logical switch checks whether VM1 and VM3 belong to the same VLAN
3. If yes, message is forwarded directly to VM3
4. Communication happens entirely within the hypervisor

Modern Network Virtualization (MNV) Process:

1. Message from VM1 sent to Logical switch of VLAN

2. Switch checks VM3's VLAN membership
3. Forwards to hypervisor for delivery
4. Similar to TNV for same-host communication

7.3.2 Case II: Same VLAN, Different Hypervisor

Scenario: VM1 (Host A) wants to communicate with VM5 (Host B), both in same VLAN

Traditional Network Virtualization (TNV) Process:

1. Message from VM1 to LVS of Host A
2. Check VM5's VLAN; VM5 belongs to Host B, so inform Hypervisor A
3. If VM1 and VM5 are authenticated, send to the physical switch
4. Since VM1 and VM5 belong to different hypervisors, forward to router
5. Router finds the responding switch to communicate with the proper VM
6. Forward to Host B, then to its LVS, then to VM5

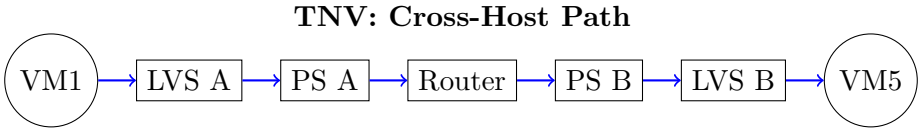


Figure 11: Traditional Network Virtualization - Cross-Host Communication

Modern Network Virtualization (MNV) Process:

1. Message encrypted at source
2. VM1 to Logical Switch to Hypervisor A
3. Encrypted message sent to Logical Router
4. Router denies or encrypts the message based on authentication
5. Sends to Hypervisor B
6. Decrypted and delivered to VM5
7. MNV provides enhanced security through encryption

7.3.3 Case III: Different VLAN, Same Hypervisor

Scenario: VM1 (VLAN 100) wants to communicate with VM4 (VLAN 200), both on Host A

Traditional Network Virtualization (TNV) Process:

1. Message goes through the physical switch (PS) layers
2. Routed through logical router to cross VLAN boundaries
3. More complex than same-VLAN communication

Modern Network Virtualization (MNV) Process:

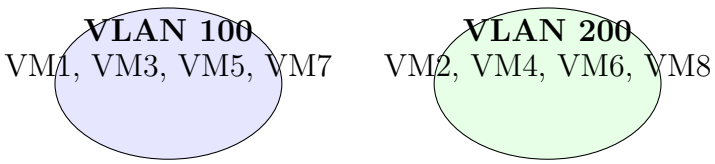
1. More efficient routing within the virtualization layer
2. Can communicate through logical components without always requiring physical infrastructure

7.4 VLAN in Network Virtualization

VLAN (Virtual Local Area Network) enables logical segmentation of networks regardless of physical topology.

Benefits:

- Network segmentation for security
- Broadcast domain isolation
- Improved network performance
- Flexible network management
- Resource allocation per department/function



Logical Isolation Despite Physical Proximity

Figure 12: VLAN Segmentation

8 Service Virtualization

Service virtualization creates virtual services that simulate the behavior of dependent components in software applications, enabling testing and development without access to actual services.

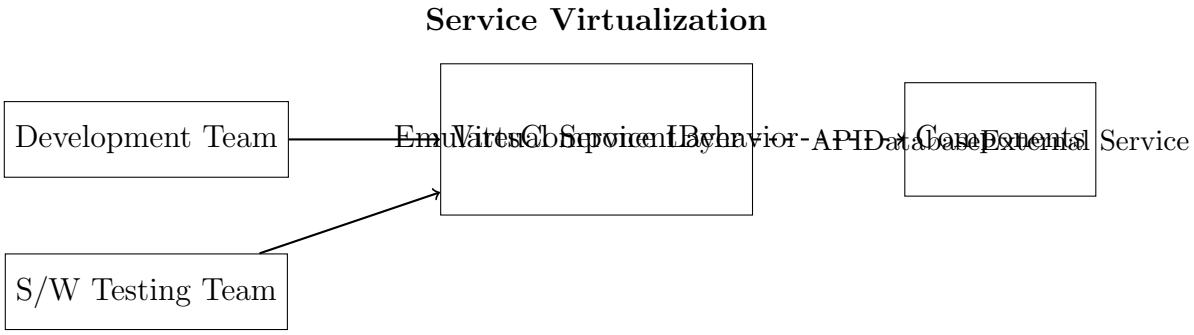


Figure 13: Service Virtualization Architecture

8.1 Components and Architecture

8.2 Key Concepts

8.2.1 VM Square (Service Catalog)

A comprehensive catalog of available virtualized services that teams can access for development and testing.

Features:

- Centralized repository of virtual services
- Easy discovery and provisioning
- Version control for services
- Access management

8.2.2 Data Commingling

Critical Concern

Data commingling occurs when data that is supposed to be isolated gets mixed together, potentially causing security breaches, compliance issues, or data corruption.

Prevention Strategies:

- Proper tenant isolation in multi-tenant environments
- Access control and authentication
- Data encryption
- Regular auditing

8.2.3 Multi-tenancy

Multi-tenancy allows multiple customers (tenants) to share the same infrastructure while maintaining data isolation.

Characteristics:

- Resource sharing with logical separation
- Cost efficiency
- Centralized management
- Scalability
- Security isolation between tenants

8.2.4 Service Level Agreement (SLA)

An SLA defines the expected service level between provider and consumer.

Key Components:

- **Availability**: Uptime guarantees (e.g., 99.9%)
- **Performance**: Response time, throughput
- **Support**: Response times for issues
- **Security**: Data protection measures
- **Penalties**: Compensation for SLA violations

8.2.5 Service Monitoring

Continuous monitoring ensures services meet defined SLA requirements.

Monitoring Aspects:

- Performance metrics
- Availability tracking
- Resource utilization
- Error rates
- User experience
- Compliance adherence

8.2.6 Subscription Model

Service virtualization often uses subscription-based pricing:

- Pay-per-use
- Monthly/Annual subscriptions
- Tiered pricing based on usage
- Resource-based billing

8.3 Service Virtualization Benefits

1. **Intermediary Support**: Service virtualization acts as an intermediary, providing mock services
2. **Resource Management**: Efficient allocation and management of virtualized services
3. **Testing Efficiency**: Enable parallel development and testing without dependency on actual services
4. **Cost Reduction**: Reduce need for expensive test environments
5. **Faster Development**: Remove bottlenecks caused by unavailable dependencies
6. **Risk Mitigation**: Test against various scenarios without affecting production

9 Integration of Physical and Virtual Machines

Modern IT infrastructures often combine both physical and virtual machines to optimize performance, cost, and flexibility.

9.1 Hybrid Infrastructure

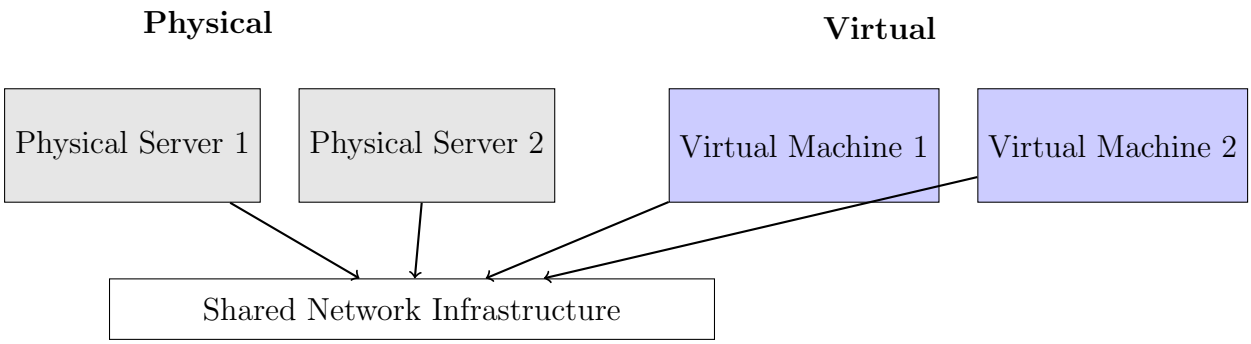


Figure 14: Hybrid Physical-Virtual Infrastructure

9.2 Integration Strategies

- **Workload Placement**: Determine which workloads run on physical vs. virtual infrastructure
- **Network Integration**: Seamless communication between physical and virtual systems
- **Storage Integration**: Shared storage accessible by both physical and virtual machines
- **Management Integration**: Unified management tools for both environments
- **Security Integration**: Consistent security policies across physical and virtual resources

9.3 Use Cases for Physical Machines

Despite virtualization benefits, some scenarios still require physical machines:

- High-performance computing (HPC) applications
- Latency-sensitive applications
- Licensing constraints
- Hardware-specific requirements
- Maximum resource isolation needs
- Regulatory compliance requirements

10 Benefits of Virtualization

10.1 Server Consolidation

Definition: Combining multiple physical servers into fewer machines using virtualization.

Benefits:

- Reduced hardware costs
- Lower power consumption
- Reduced cooling requirements
- Smaller data center footprint
- Simplified management

Example: 10 physical servers with 20% utilization can be consolidated to 2-3 physical servers running multiple VMs at 60-70% utilization.

10.2 Scalability

Virtualization provides both vertical and horizontal scalability:

- **Vertical Scaling**: Add more resources (CPU, RAM) to existing VMs
- **Horizontal Scaling**: Add more VM instances to handle increased load
- **Elastic Scaling**: Automatically scale based on demand
- **Rapid Provisioning**: Deploy new VMs in minutes vs. days for physical servers

10.3 Good Degree of Customization

- **Resource Allocation**: Precisely allocate CPU, memory, storage per VM
- **OS Selection**: Run different operating systems on same hardware
- **Network Configuration**: Custom network topologies per application
- **Security Policies**: Tailored security settings per VM
- **Software Stack**: Independent software configurations

10.4 Additional Benefits

Benefit	Description
Disaster Recovery	Easy backup, replication, and restoration of entire VM environments
Testing & Development	Quickly create isolated test environments; snapshots for rollback
Legacy Application Support	Run older applications on legacy OS without maintaining old hardware
Energy Efficiency	Reduce power consumption through server consolidation
Hardware Independence	VMs portable across different physical hardware
Reduced Downtime	Live migration of VMs for maintenance without service interruption
Cost Savings	Lower capital and operational expenses
Improved Resource Utilization	Maximize use of available hardware resources
Simplified Management	Centralized management and automation tools
Isolation	Applications isolated from each other; failures contained

Table 5: Comprehensive Virtualization Benefits

11 Virtualization Challenges and Considerations

11.1 Performance Overhead

While virtualization is highly efficient, there is some performance overhead:

- Hypervisor layer introduces small latency
- I/O operations may be slower than bare metal

- Memory management overhead
- Network virtualization adds minimal latency

Mitigation:

- Hardware-assisted virtualization (Intel VT-x, AMD-V)
- SR-IOV for direct hardware access
- NUMA-aware VM placement
- Proper resource allocation and tuning

11.2 Security Considerations

- **Hypervisor Vulnerabilities**: Hypervisor is a critical attack surface
- **VM Escape**: Potential for attackers to break out of VM isolation
- **Inter-VM Attacks**: VMs on same host could potentially attack each other
- **Sprawl**: Uncontrolled VM proliferation increases attack surface

Best Practices:

- Keep hypervisor patched and updated
- Implement network segmentation
- Use encryption for data at rest and in transit
- Regular security audits
- VM lifecycle management
- Access control and authentication

11.3 License Complexity

- Software licensing in virtual environments can be complex
- Some vendors charge per physical CPU, others per virtual CPU
- Compliance tracking across dynamic VM environments
- Need for proper license management tools

11.4 Skills and Training

- Requires specialized knowledge and skills
- Different skillset than traditional infrastructure management
- Ongoing training needed as technologies evolve
- Need for automation and scripting capabilities

12 Future Trends in Virtualization

12.1 Containers and Microservices

Containers provide lightweight virtualization at the application level:

- Faster startup than VMs
- Lower resource overhead
- Better portability
- Ideal for microservices architectures
- Technologies: Docker, Kubernetes, containerd

12.2 Edge Computing Virtualization

- Virtualization extending to edge locations
- Low-latency applications
- IoT device management
- 5G network functions virtualization

12.3 AI-Driven Virtualization Management

- Predictive resource allocation
- Automated performance optimization
- Intelligent workload placement
- Anomaly detection and self-healing

12.4 Serverless Computing

- Next evolution beyond containers
- Function-as-a-Service (FaaS)
- Event-driven execution
- Complete abstraction of infrastructure

13 Summary

Virtualization is a cornerstone technology enabling modern cloud computing. Key takeaways include:

- **Core Concept**: Virtualization abstracts physical resources into logical, software-defined resources
- **Hypervisors**: Three types (Bare Metal, Hosted, Embedded) enable VM management
- **Virtualization Types**: Includes compute (VMs), storage, network, application, and service virtualization
- **Techniques**: Full virtualization and para-virtualization offer different trade-offs
- **Benefits**: Server consolidation, scalability, flexibility, and cost savings
- **Challenges**: Performance overhead, security, licensing, and skills requirements
- **Integration**: Modern infrastructures combine physical and virtual resources
- **Future**: Containers, edge computing, AI-driven management, and serverless computing

14 Review Questions

1. Explain the key differences between physical and virtual infrastructure.
2. What are the three types of hypervisors? Provide examples and use cases for each.
3. Compare and contrast full virtualization and para-virtualization.
4. Describe the architecture of storage virtualization and explain the difference between NAS and SAN.
5. What is the purpose of VLANs in network virtualization?
6. Explain how service virtualization benefits software development and testing teams.
7. Discuss three major benefits and three challenges of virtualization.
8. How does Modern Network Virtualization (MNV) improve upon Traditional Network Virtualization (TNV)?
9. What is server consolidation and what benefits does it provide?
10. Describe scenarios where physical machines might be preferred over virtual machines.

15 Glossary

Virtualization Creating virtual versions of physical computing resources

Virtual Machine (VM) A software-based emulation of a physical computer

Hypervisor Software that manages and runs virtual machines

Full Virtualization VMs run unmodified guest OSes using trap-and-emulate

Para-virtualization VMs use modified guest OSes with hypercalls for better performance

NAS Network Attached Storage - file-level storage over network

SAN Storage Area Network - block-level storage over dedicated network

VLAN Virtual Local Area Network - logical network segmentation

Service Virtualization Simulating dependent services for testing

Multi-tenancy Multiple customers sharing infrastructure with isolation

SLA Service Level Agreement - defines expected service levels

Server Consolidation Combining multiple servers using virtualization

16 References

1. Portnoy, M. (2012). *Virtualization Essentials*. Sybex.
2. VMware. (2024). *vSphere Documentation*. VMware Inc.
3. Microsoft. (2024). *Hyper-V Technical Reference*. Microsoft Corporation.
4. Kusnetzky, D. (2011). *Virtualization: A Manager's Guide*. O'Reilly Media.
5. Golden, B. (2010). *Virtualization for Dummies*. Wiley Publishing.