Huawei Cloud Certification Training

HCIA-Cloud Computing

Learning Guide

ISSUE: 5.0

HUAWEI TECHNOLOGIES CO., LTD

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base Bantian, Longgang Shenzhen 518129 People's Republic of China

Website: https://e.huawei.com

Huawei Certification System

The Huawei certification system is a platform for shared growth, part of a thriving partner ecosystem. There are two types of certification: one for ICT architectures and applications, and one for cloud services and platforms. There are three levels of certification available:

- Huawei Certified ICT Associate (HCIA)
- Huawei Certified ICT Professional (HCIP)
- Huawei Certified ICT Expert (HCIE)

Huawei certification courses cover the entire ICT domain, with a focus on how today's architecture generates cloud-pipe-device synergy. The courses present the latest developments of all essential ICT aspects to foster a thriving ICT talent ecosystem for the digital age.

HCIA-Cloud Computing V5.0 is intended for beginners. Focusing on the virtualization technologies and FusionAccess of Huawei, this course aims to help you learn the basics of cloud computing and resource pooling, and to cultivate cloud computing engineers that are capable of using FusionCompute for virtualization and managing FusionAccess.

The HCIA-Cloud Computing V5.0 certification includes the following courses: basics of cloud computing (server, storage, network, and OS), overview of FusionCompute, routine management and troubleshooting of virtualization resource pools, overview of FusionAccess and related components, as well as their installation, deployment, service management, and troubleshooting.

Passing the HCIA-Cloud Computing V5.0 certification means that you are capable of designing, managing, and maintaining FusionCompute and FusionAccess.


Huawei Certification
Cloud Service & Platform
ICT Infrastructure
HarmonyOS
IoT
GaussDB
MDC
Big Data
AI
openGauss
openEuler
Kunpeng Application Developer
Cloud
Storage
Intelligent Computing
Enterprise Communication
Intelligent Vision
Data Center
Datacom
WLAN
Security
SDN
Transmission
Access
LTE
5G
HCIE
Huawei Certified ICT Expert
HCIP
Huawei Certified ICT Professional
HCIA
Huawei Certified ICT Associate

Contents

1 Cloud Computing Basics

1.1 IT Basics

1.1.1 What Is IT?

1.1.1.1 IT Around Us

"IT" is the common term for an entire spectrum of technologies for information processing, including software, hardware, communications, and related services.

IT technologies around us are changing the way we live, for example, taxi hailing software that places and receives orders via apps, communications software that enables real-time voice calls over the Internet, and e-malls that provide online shopping experience via apps. These various IT software and hardware are disrupting and changing the way we live and work.

Figure 1-1 IT around us

Let's take the ride-hailing software as an example. Yidao Yongche is the first app that allows users to book professional car reservation services online. It was founded in 2010 in Beijing. Later, a series of ride-hailing software such as Uber, DiDi, and Gaode emerged. Companies that launch these ride-hailing software do not have a taxi. Instead, they build a service platform to bring ride-hailing resources together. Ride-hailing platforms connect vehicle drivers with users who need a ride via IT technology. Drivers who own a vehicle can register as ride-hailing service providers on the platform. As the number of the registered drivers increases, users on the platform who need a ride can get a quick response after they initiate a travel order.

Over the past decade, ride-hailing software has greatly changed the way we travel. In the past, we had to stop a taxi along the roadside. Now, we only need to initiate a travel order on our mobile phone and wait for the vehicle driver to pick us up at the designated place.

In summary, new technologies are changing our lives.

1.1.1.2 Core of IT

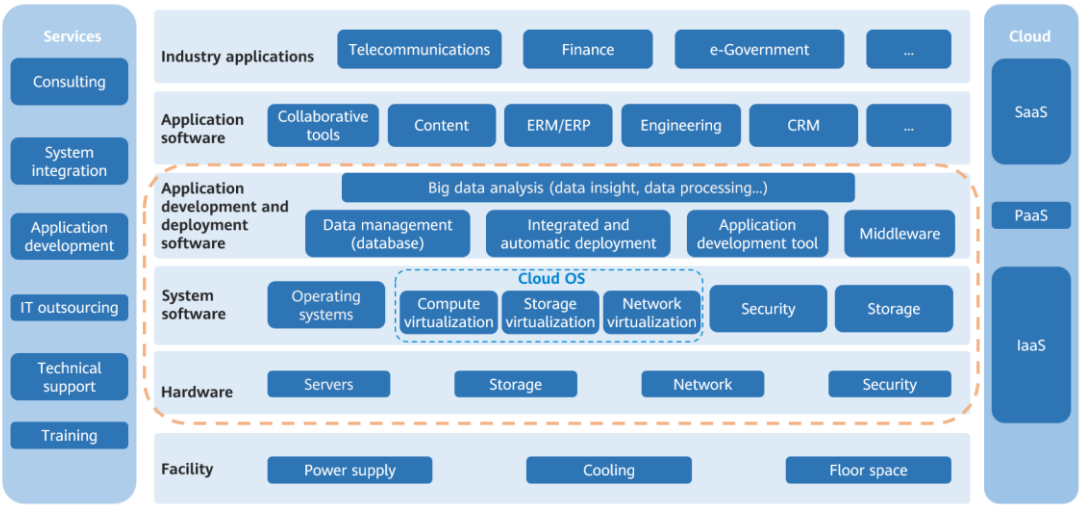


Figure 1-2 Data center–based IT architecture

Traditional IT infrastructure consists of common hardware and software components, including facilities, data centers, servers, network hardware, desktop computers, and enterprise application software solutions. In the new era, the IT architecture has changed. As shown in Figure 1-2, the cloud emerges based on IT infrastructure hardware, allowing you to develop application and deploy software on the cloud infrastructure. The cloud greatly changes the IT infrastructure of the Internet and addresses critical challenges to traditional IT.

1.1.2 Challenges to Traditional IT

1.1.2.1 Information Explosion Is Coming

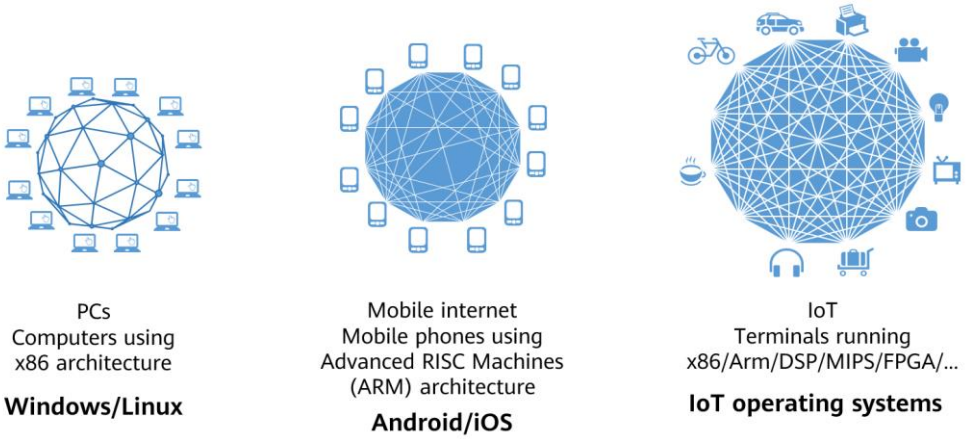


Figure 1-3 Development history of the Internet

With the proliferation of mobile Internet and the fully connected era, more terminal devices are being used every day, and data is exploding, posing unprecedented challenges to traditional ICT infrastructure. The Internet has gone through PCs, mobile Internet, and the Internet of Everything (IoE).

In the PC era, people are connected to each other through computers. In the mobile era, people are connected through mobile devices, such as phones and tablets. In the 5G era, all computers, mobile phones, and smart terminals are connected to each other, and we are ushering in the era of IoE.

In the IoE era, the entire industry will compete for ecosystem. From the PC era to the mobile era, and then to the IoE era, the ecosystem changes fast at the beginning, then tends to be relatively stable, and rarely changes when it is stable. In the PC era, a large number of applications run on Windows, Intel chips, and x86 architecture. In the mobile era, applications run on iOS and Android systems that use the ARM architecture.

The Internet has gone through two generations and is now ushering in the third generation, the Internet of Everything. Compared with the previous generation, the number of devices and the market scale of each generation increase greatly, presenting future opportunities. As the Intel and Microsoft in the PC era and the ARM and Google in the mobile era, each Internet generation has its leading enterprises who master the industry chain. In the future, those who have a good command of core chips and operating systems will dominate the industry.

1.1.2.2 Challenges to Traditional IT

As the Internet has grown, massive traffic, users, and data have been generated. The traditional IT architecture has been unable to meet the demands of fast developing enterprises. To keep up with the rapidly developing businesses, enterprises need to continuously purchase traditional IT devices. Therefore, the disadvantages of traditional IT devices gradually emerge:

- Long procurement period slows rollout of new business systems.
- The traditional centralized architecture of traditional IT has poor scalability. Scale-up expansion can only improve the processing performance of a single server.
- Traditional hardware devices are isolated from each other, and reliability mainly depends on software.
- Devices and vendors are heterogeneous and hard to manage.
- The performance of a single device is limited.
- Low device utilization leads to high total cost of ownership (TCO).

As we can see, traditional IT infrastructure cannot meet enterprises' requirements for rapid development. A new IT architecture is required to address the challenge. As a result, the IT infrastructure begins to change.

1.1.3 IT Development Trend

1.1.3.1 Enterprises Are Migrating To the Cloud Architecture

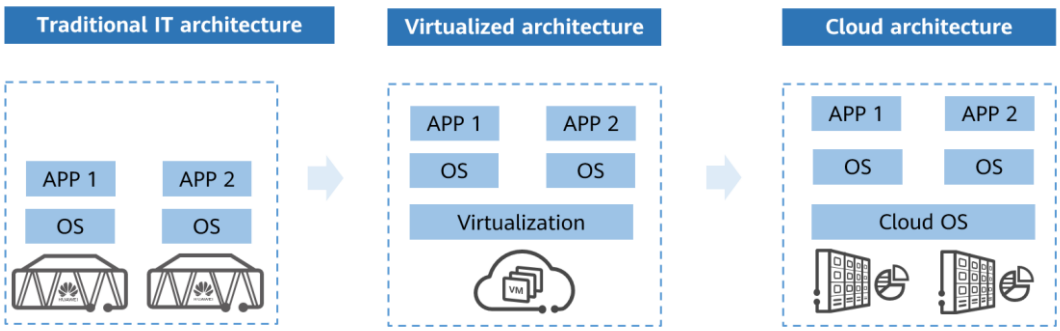


Figure 1-4 Evolution of the enterprise IT architecture

As shown in Figure 1-4, the enterprise IT architecture evolves from the traditional IT architecture to the virtualized architecture and then to the cloud architecture. Traditional IT infrastructure consists of common hardware and software components, including facilities, data centers, servers, network hardware, and enterprise application software solutions. This architecture requires more power, physical space, and money and is often installed locally for enterprise or private use only.

The virtualized architecture is based on virtualized underlying physical hardware. Enterprise service systems and other basic IT applications are deployed on the virtual environment. With the virtualization technology, computer components can run on the virtual environment rather than the physical environment. Virtualization enables maximum utilization of the physical hardware and simplifies software reconfiguration.

With a further developed architecture based on virtualization, the cloud architecture uses cloud technologies, including virtualization technology, distributed technology, and automatic O&M technology. It integrates an enterprise's IT resources, improves resource usage and scheduling efficiency, automates IT O&M, and provides self-service IT offerings.

Key features of cloud migration of enterprise data centers are as follows:

- From resource silos to resource pooling;
- From centralized to distributed architecture;
- From dedicated hardware to software-defined storage (SDS) mode;
- From manual handling to self-service and automatic service;
- And from distributed statistics to unified metering.

According to a report from an international authoritative statistics organization, from 2015, the computing industry accounts for one-third of the global IT revenue and 100% of IT growth. The traditional IT architecture development has almost stalled, and even declined in recent years.

In this case, it is easy to see that the IT architecture of an enterprise (an Internet enterprise or a traditional enterprise) will be gradually replaced by the cloud architecture. Moreover, an increasing number of enterprises are using cloud architecture worldwide.

Currently, about 70% of enterprises use cloud architecture in the US and 40% of enterprises use cloud architecture in China. Statistics show that this number is likewise rising with time.

1.2 Cloud Computing

As mentioned in the previous chapter, the industry built on cloud computing, computers, and virtualization has become the mainstream of the IT industry. Before we get into cloud computing, let's take a quick look at the evolution of computers and virtualization.

1.2.1 Computer Basics

1.2.1.1 What Is Computer?

A computer is a high-speed electronic device capable of performing numerical and logical calculations. It automatically stores and processes data according to a set of programming instructions given to it.

This is an official and formal definition. When it comes to computers, maybe the first thing comes to our mind is desktops, laptops, and servers. In fact, the devices such as storage, network, and security in the data center are all computer devices.

1.2.1.2 Timeline of Computer History

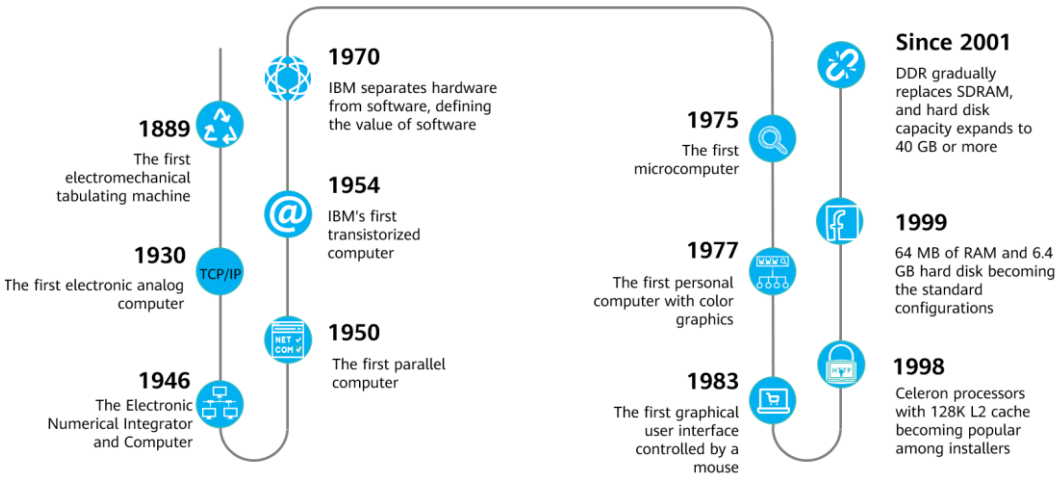


Figure 1-5 Timeline of computer history

Computing tools have progressed from simple to complex and from low to high level, such as knotting to abacus and calipers, and then mechanical computers. They have played historical roles in different periods and have also inspired the development of modern electronic computers.

As shown in Figure 1-5, the following events are the milestones of computer history:

- In 1889, American scientist Herman Hollerith developed an electromechanical tabulating machine for storing accounting data.
- In 1930, American scientist Vannevar Bush built the world's first analog computer with some digital components.

- In 1946, the U.S. military customized the world's first electronic computer, the Electronic Numerical Integrator and Computer.
- In 1950, the first parallel computer was invented, using von Neumann architecture: binary format and stored programs.
- In 1954, IBM made the first transistorized computer, using floating-point arithmetic for improved computing capabilities.
- In 1970, IBM System/370 was announced by IBM. It replaces magnetic core storage with large-scale integrated circuits, uses small-scale integrated circuits as logical components, and applies virtual memory technology to separate hardware from software, thereby defining the value of software.
- In 1975, MITS developed the world's first microcomputer.
- 1977, the first personal computer with color graphics was invented.
- In 1998, Celeron processors with 128K L2 cache became popular among installers, and 64 MB of memory and 15-inch displays became standard configurations.
- In 1999, Pentium III CPUs became a selling point for some computer manufacturers. The 64 MB of memory and 6.4 GB hard disk became standard configurations.
- Since 2001, Pentium 4 CPUs and Pentium 4 Celeron CPUs have been the standard configurations for computers. DDR has gradually replaced SDRAM as the common type of memory. In addition, 17-inch CRT or 15-inch LCD displays have been the preferred choice for customers. The capacity of hard disks has gradually expanded to 40 GB or more.

1.2.2 Virtualization Basics

1.2.2.1 What Is Virtualization?

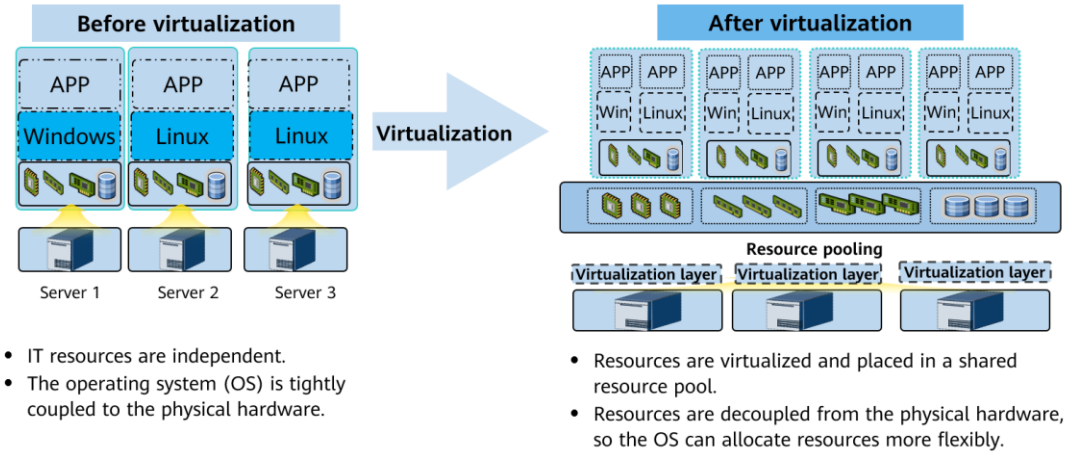


Figure 1-6 Virtualization structure

Virtualization has a wide range of meanings. Virtualization is the act of creating a virtual version of something, a logical representation of resources. Virtualization allows multiple virtual machines (VMs) to run on a physical server. The VMs share the CPU, memory, and I/O hardware resources on the physical server, but they are logically isolated from each other. Virtualization is the fundamental technology that powers cloud computing.

In Figure 1-6, the left picture shows the architecture before virtualization. After purchasing servers, enterprises install operating systems (OS), and deploy system applications and required basic environments on the servers.

The traditional enterprise IT architecture has two features: 1. The resources on each server are independent from each other. For example, the resources on the server 1 are standing idle while the resources on the server 2 are insufficient. As a result, the resources cannot be fully used. 2. The OS is tightly coupled to the physical hardware. The OS hardware drivers must adjust to the underlying physical servers because the OS is directly deployed on the hardware. This will make it difficult for enterprises to migrate their system applications to the physical servers of other vendors.

In terms of the virtualization architecture, after purchasing a physical server, enterprises deploy a virtualization layer on the server, turning hardware resources of the server into virtualized resources and putting them in a resource pool. Then, VMs are created based on the virtual resource pool to run enterprise service applications. Resources in this architecture are abstracted into a shared resource pool, greatly improving resource utilization and making resources no longer isolated. The virtualization layer decouples the physical hardware from the upper-layer OS, allowing you to flexibly migrate your applications as long as the virtual hardware structure of VMs is consistent.

1.2.2.2 Timeline of Virtualization History

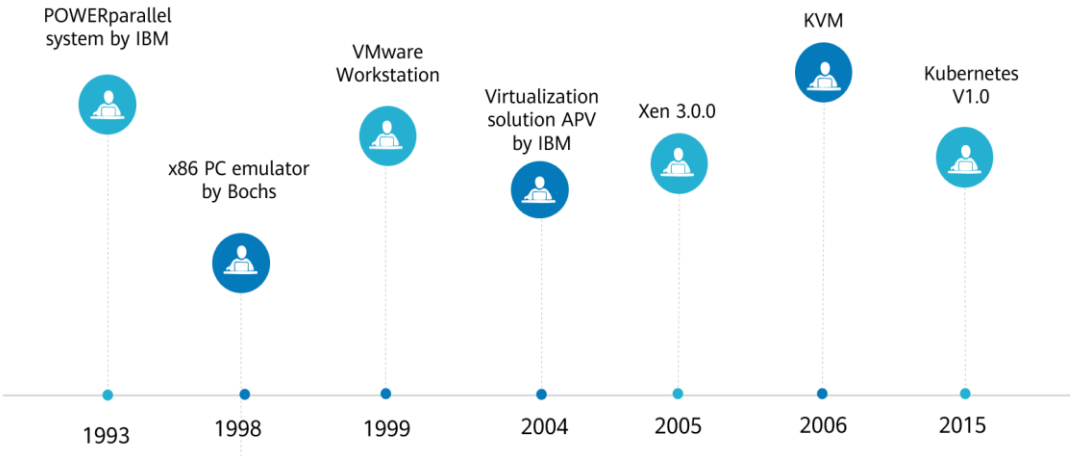


Figure 1-7 Timeline of virtualization history

Figure 1-7 shows the milestones in virtualization history:

- In 1993, IBM launched an upgradeable POWER parallel system, the first microprocessor-based supercomputer using RS/6000 technology.
- In 1998, Bochs released x86 PC emulator.
- In 1998, VMware was founded. In 1999, the company launched its first product, VMware Workstation, the commercial virtualization software that allows to run multiple operating systems on a single physical server. Since then, virtualization technology has been widely applied.
- In 1999, IBM first proposed the logical partitioning (LPAR) virtualization for the AS/400 system.
- In 2000, Citrix released XenDesktop, a desktop virtualization product.

- In 2004, IBM released the virtualization solution Advanced Power Virtualization (APV), which supports resource sharing. This solution was later renamed PowerVM in 2008.
- In 2005, Xen 3.0.0 was released as the first hypervisor with Intel® VT-x support. Xen 3.0.0 can run on 32-bit servers.
- In 2006, Qumranet, an Israeli startup, officially announced Kernel-based Virtual Machine (KVM).
- 2006–present defines cloud computing and big data era.
- In 2007, InnoTek, a German company, developed VirtualBox.
- In 2008, Linux Container (LXC) 0.1.0 was released to provide lightweight virtualization.
- In 2010, Red Hat released RHEL 6.0, removing Xen and leaving KVM as the only bundled virtualization option.
- In 2015, Kubernetes v1.0 was released, opening the cloud native era.

1.2.3 Cloud Computing Basics

In the first two chapters, we have learned about the development of computers and virtualization technology. Now, let's see what the cloud computing is.

1.2.3.1 What's Cloud Computing?

Since the emergence of cloud computing, it has been defined in a variety of ways. The National Institute of Standards and Technology (NIST) defines cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (such as networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This definition is widely accepted.

Key points:

1. Cloud computing is a model rather than a piece of technology.
2. With cloud computing, users can access IT resources such as networks, servers, storage, applications, and services easily.
3. Cloud computing enables ubiquitous access to resources connected to a network.
4. Resources can be quickly provisioned and released for elastic scaling. On-demand self-service enables minimal service provider interaction.

We can also take a look at cloud computing from another perspective. Let's split cloud computing into cloud and computing. Cloud is a metaphor for networks and the Internet. It is an abstract entity of the Internet and the underlying infrastructure required for establishing the Internet. Computing refers to a combined range of compute services (functions and resources). Cloud computing uses powerful computers to deliver resource services over the Internet.

Whatever you may know about cloud computing, it is now everywhere around us. Next, I will introduce cloud computing around us.

1.2.3.2 Cloud Services and Applications Around Us (Personal)

Now our daily life benefits from applications of cloud computing.

Baidu Wangpan is a cloud storage service provided by Baidu, which allows you to easily back up, synchronize, and share photos, videos, and documents. It is now widely used in China. Without cloud computing, we need to manually copy files to other hard disks to back up, synchronize, and share files. With cloud computing, you can easily back up, synchronize, and share files using a client connected to the Internet and installed either on a mobile phone or a PC. Resources are shared by using cloud computing technology. Therefore, the shared data can be downloaded easily by other people. In addition, data can be automatically synchronized if required.

There are various cloud applications, such as cloud albums (Baidu Cloud and iCloud Shared Album) and cloud music (NetEase Cloud Music, Kugou Music, Kuwo Music, and Xiami Music). From the applications we use in our life, we can see that cloud computing makes our life more convenient. Enterprises also use cloud computing to provide better products for better user experience.

1.2.3.3 Cloud Services and Applications Around Us (Enterprises)

Videoconferencing

Livestreaming

Figure 1-8 Cloud services and applications around us (enterprises)

Cloud Meeting provides an all-scenario, device-cloud synergy videoconferencing solution for intelligent communication and collaboration on different terminals, in different regions, and with collaborators in other companies. Livestreaming allows us to play games, learn technology, and make friends online with more fun.

Videoconferencing also changes rapidly as technology develops. The videoconferencing market in China is growing steadily, with an average annual growth rate of more than 20%, driven by the requirements of enterprises in sectors such as government, transportation, electric power, medical care, education, finance, and military. Currently, less than 5% of Chinese enterprises have video conference rooms, but more and more enterprises are realizing the value of efficient collaboration. The videoconferencing system has gradually become the common practice for higher work efficiency.

Cloud meeting is usually used for enterprise office, telemedicine, smart education, and enterprise organization building.

1.2.3.4 History of Cloud Computing

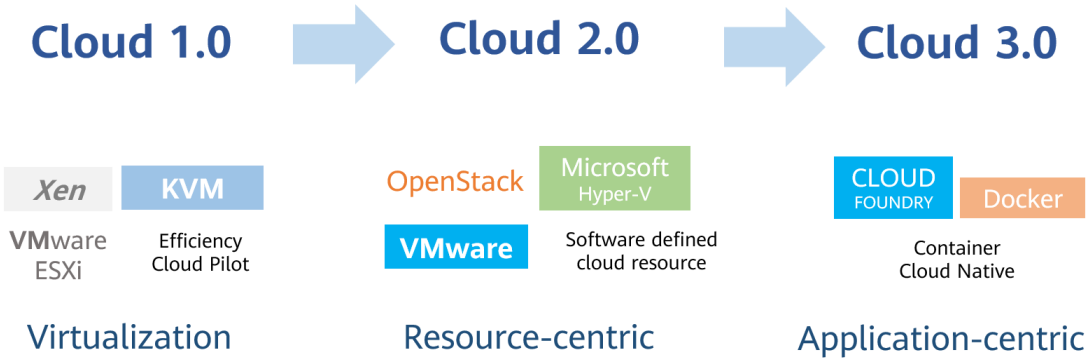


Figure 1-9 History of cloud computing

As shown in Figure 1-9, since the inception of the cloud computing, the IT architecture has evolved from the traditional non-cloud architecture to the cloud architecture. The IT architecture has experienced the following three milestones:

1. Cloud 1.0 features IT infrastructure resource virtualization for data center administrators. In this phase, virtualization technology is introduced to decouple IT applications from the underlying infrastructure, allowing multiple enterprise IT applications and operating systems to run or be deployed on the same physical server. More IT applications can run on fewer servers by using virtualization cluster scheduling software for higher resource utilization. HCIA Cloud Computing course focuses on this phase and describes the implementation and advantages of cloud computing in this phase.
2. Cloud computing 2.0 features resource servitization and management automation for infrastructure cloud tenants and cloud users. This phase features standard infrastructure services and resource scheduling automation software on the management plane, and software-defined storage and software-defined networking technologies on the data plane. By using these software and technologies, data center administrators do not need to manually handle resource application, release, and configuration. Resources will be distributed under necessary conditions (such as resource quota and permissions) with one-click automatic operation. This speeds up the distribution of infrastructure resources to enterprises for application deployment, reduces the time needed to prepare resources for IT application rollouts, and turns the static distribution mode of enterprise infrastructure into an elastic on-demand provisioning mode. This also supports enterprise IT department in shifting its core services towards agility and better responding to the ever-changing competitive development environment. Infrastructure resources in Cloud 2.0 phase for cloud tenants can be VMs, or containers (lightweight VMs), or physical machines. The cloud migration of enterprises in this phase does not involve changes in enterprise IT applications, middleware, and database software architecture above the infrastructure layer.
3. Cloud 3.0 features distributed microservice-based enterprise application architecture and Internet-based, reconstructed, and big data powered intelligent enterprise data architecture for enterprise IT application developers and Q&M personnel. Enterprise

IT architecture gradually expands from vertically layered architecture (siloed, highly complex, stateful, and large-scale architecture designed for each service application based on traditional commercial databases and middleware business suites) to the database (based on open source and highly shared resources across different service applications), middleware platform service, and distributed stateless architecture (more lightweight and decoupled, and the data and application logic completely separated). This helps enterprises take a step forward service agility, intelligence, and improved resource utilization for quick rollout of new services.

Looking back on the history of cloud computing, Cloud 1.0 is out of date, but it is still the cornerstone of cloud computing. Some enterprises adopt Cloud 2.0 for commercial use and are considering expanding the scale and evolving to Cloud 3.0. The other enterprises are evolving from Cloud 1.0 to 2.0, and are even evaluating and implementing the evolution from Cloud 2.0 to 3.0.

1.2.4 Benefits of Cloud Computing

1.2.4.1 On-Demand Self-Service

What's the first thing that comes to your mind when you hear on-demand self-service? You may think of shopping in a supermarket. In a supermarket, you can select products based on your requirements. You can compare product descriptions, prices, and brands, and determine which one to purchase based on price-performance ratio or other factors. This is called on-demand self-service. Similarly, you can download different Apps or purchase required services on Huawei Cloud on your own.

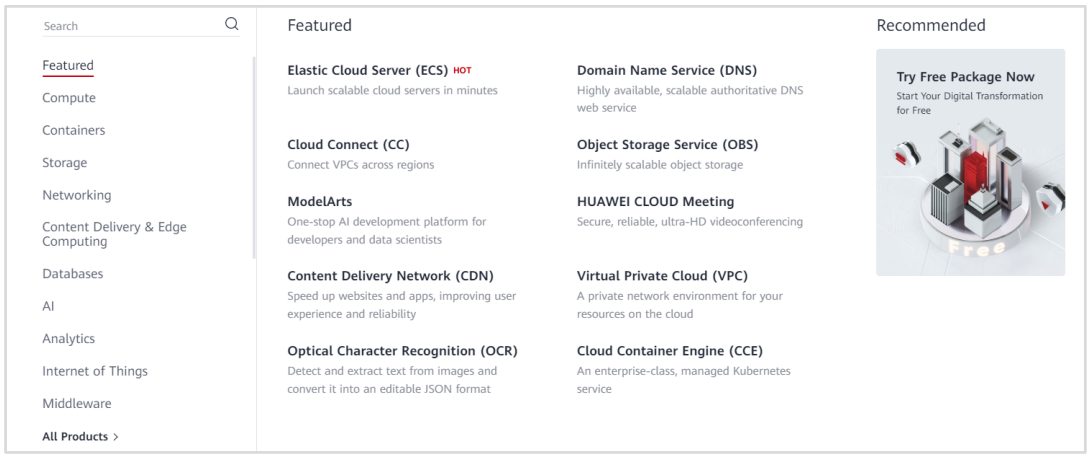


Figure 1-10 Featured cloud services

One of the prerequisites for on-demand self-service is to know your requirements and know which product can meet your requirements. A supermarket offers an enormous variety of products. Similarly, a cloud computing provider may provide many types of cloud products, as shown in Figure 1-10. You need to know which product can suit your needs before placing an order.

1.2.4.2 Widespread Network Access

We can think of cloud computing as a combination of the Internet and computing. Therefore, network access is a built-in attribute of cloud computing.

Now, almost everyone has access to the Internet. We can access the Internet via such electronic devices as PCs, tablets, and cell phones. This means you can use cloud computing with much ease and convenience.

In a word, as long as you can access the resource pools offered by cloud service providers through the network, you can use cloud services anytime and anywhere.

1.2.4.3 Resource Pooling

Resource pooling is also one of the prerequisites for on-demand self-service. Through resource pooling, we cannot only put commodities of the same type together, but also provide commodities in units of finer granularity. In a supermarket, different types of commodities are placed in different areas. In this way, customers can quickly find the commodities they need. However, this can only be considered as resource classification, not resource pooling. What is resource pooling?

Resource pooling not only places resources of the same type into a resource pool, but also breaks up all resources into the smallest possible unit.

We can take instant noodles as an example. You may encounter the situation that one pack is not enough but two packs are too many for you. But the smallest unit for instant noodles is the pack. Resource pooling can solve the problem. It puts all the noodles in one pool and you can buy as many as you need. The cafeteria is a good example of resource pooling. In a cafeteria, juices may be separated by flavors, and you can take as much as you need.

Another function of resource pooling is to shield the differences between different resources. If pooled cola is provided in a restaurant, customers cannot see whether Pepsi, Coca-Cola, or both are in the pool. In cloud computing, resources that can be pooled include compute, storage, and network resources. Compute resources include CPUs and memory. If CPUs are pooled, their smallest unit is core, and CPU vendors such as AMD and Intel are not displayed.

1.2.4.4 Rapid Deployment and Elastic Scaling

Enterprise business may fluctuate. To ensure stable service running during peak traffic hours, enterprises purchase more servers to scale out. When the access traffic decreases, enterprises release the servers to scale in. This is called rapid elastic scaling.

In cloud computing, you can choose to manually or automatically (using preset policies) scale resources. You can scale in or out servers, or scale up or down a server.

This feature enables you to ensure stable running of services and applications while reducing costs. When an enterprise is in its infancy, it can purchase a small amount of resources. It can purchase more as its business grows. In special periods, all resources can be used for key services as needed. In non-special periods, idle resources can be used for other purposes. If the resources cannot meet your requirements in special periods, you can purchase more resources. After entering non-special periods, you can release the additional resources. In a word, you can enjoy the great convenience with cloud computing.

1.2.4.5 Metered Services

Services in cloud computing are measured by duration, resource quota, or traffic. Measured services enable auto scaling based on business scale and much better resource allocation.

Users can clearly see the usage of their purchased services, and can purchase the required number of services.

It should be noted that measuring is not billing. A measured service facilitates billing.

Most cloud computing services are chargeable while some services are free of charge. For example, you can use Auto Scaling (AS) for free, except the resources that are scaled out.

1.2.5 Cloud Computing Services and Deployment

1.2.5.1 Service Models for Cloud Computing

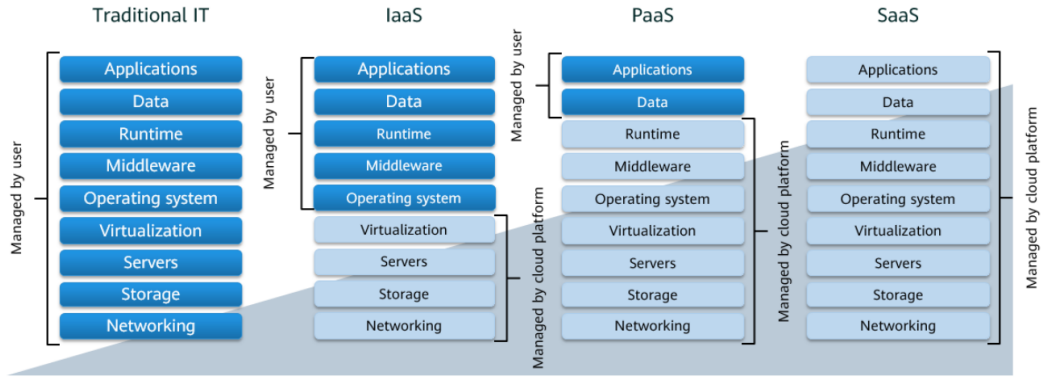


Figure 1-11 Service models for cloud computing

As shown in Figure 1-11, all models have the same hierarchical architecture. Users only need to focus on their applications. Data is generated during your use of the applications. An application program can run only after the lowest-layer hardware resource, the OS running on the hardware resource, middleware running on the OS, and running environment of the application are all ready. The architecture of cloud computing can be divided into three layers. Applications and data belong to the software layer. Hardware resources (servers, storage, and networking resources) and virtualization belong to the infrastructure layer. OSs, middleware, and runtime belong to the platform layer.

In IaaS model, cloud service providers are responsible for infrastructure layer, and users are responsible for other layers. In PaaS model, cloud service providers are responsible for infrastructure and platform layers, and users are responsible for software layer. In SaaS model, cloud service providers are responsible for all three layers.

We can use an example to illustrate these models. Figure 1-11 shows the configuration requirements of a standalone game *Sekiro: Shadows Die Twice*. The content was obtained from a Chinese game portal GamerSky.

System Requirements			
Minimum Requirements		**Recommended Requirements**	
OS	Windows 7 64-bit | Windows 8 64-bit | Windows 10 64-bit	OS	Windows 7 64-bit | Windows 8 64-bit | Windows 10 64-bit
CPU	Intel Core i3-2100 | AMD FX-6300	CPU	Intel Core i5-2500K | AMD Ryzen 5 1400
RAM	4 GB	RAM	8 GB
Storage	25 GB available space	Storage	25 GB available space
Graphics	NVIDIA GeForce GTX 760 | AMD Radeon HD 7950	Graphics	NVIDIA GeForce GTX 970 | AMD Radeon RX 570

Figure 1-12 Configuration requirements of *Sekiro: Shadows Die Twice*

In this figure, we can see the hardware requirements of the game. You purchase a computer, install an OS, and install the game software, this is a traditional IT architecture model. You purchase a cloud server from a cloud service provider, install an OS using the image, and download and install the game software, this is IaaS model. When installing such a large-scale game, the following error may occur:

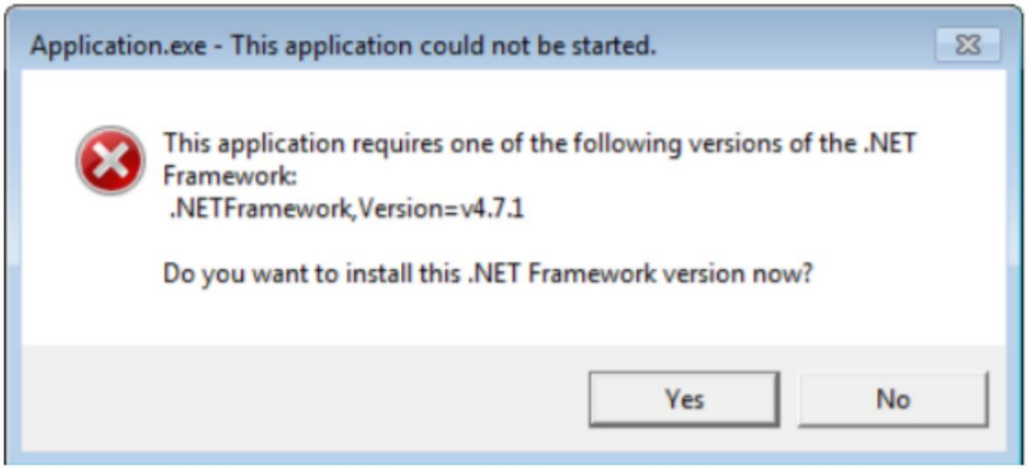


Figure 1-13 .NET Framework initialization error

The error occurs because the running environment .NET Framework is not installed. You purchase a cloud server with OS and .NET Framework installed, this is PaaS model.

You purchase a cloud server with OS, .NET Framework, and game software installed, and all you need to do is to enter your username and password to play the game, this is SaaS model.

1.2.5.2 Deployment Models for Cloud Computing

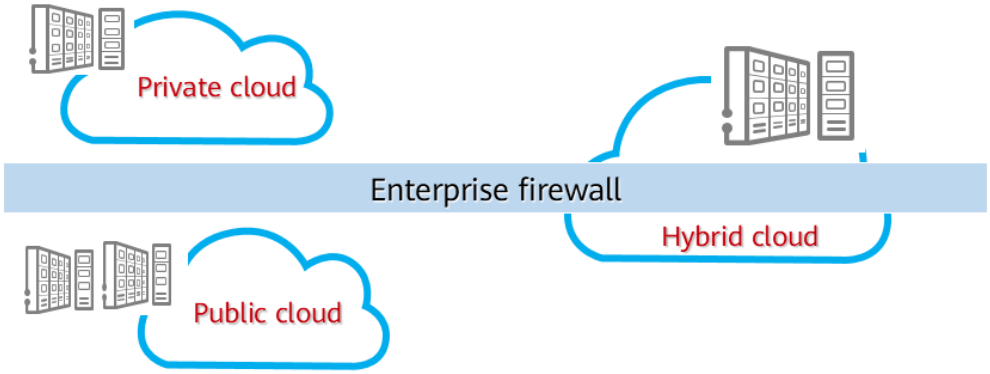


Figure 1-14 Deployment models for cloud computing

- Public cloud

  Public cloud is the first deployment model and is well known to the public. Currently, public cloud can provide users with many services. Users can access IT services through the Internet easily.

  Public clouds are usually built by cloud service providers. Service providers are responsible for the hardware and management of cloud computing. End users only need to purchase cloud computing resources or services. Public cloud resources are open to the public, and you need to connect to the Internet to use public cloud resources.

- Private cloud

  Private cloud is cloud infrastructure operated solely for a single organization. All data of the private cloud is kept within the organization's data center. Attempts to access such data will be controlled by ingress firewalls deployed for the data center, offering maximum data protection. The private cloud can be deployed based on the existing architecture of an organization. Therefore, the hardware devices that had been purchased at a high price can be used in the private cloud, avoiding waste of money. Everything has two sides. If an enterprise adopts the private cloud, data security can be ensured and existing devices can be used. However, as time goes by, the devices become increasingly old, and replacing these devices costs a lot. In addition, data cannot be shared among users or enterprises.

  A new form of private cloud has emerged in recent years. You can purchase dedicated cloud services on the public cloud and migrate your key services to the public cloud. In this way, you can enjoy dedicated, isolated compute and storage resources with high reliability, high performance, and high security.

- Hybrid cloud

  Hybrid cloud is a flexible cloud computing model. It consists of at least two of the public cloud, private cloud, and industry cloud. User services can be switched between these clouds as required. For security and ease of control, not all the enterprise information is placed on the public cloud. In this case, most enterprise users of cloud computing will adopt the hybrid cloud model. Many enterprises choose to use a combination of public and private clouds. Public clouds only charge

users for the resources they use, greatly reducing the cost of enterprises with demand spikes. To give just one example, some retailers face demand spikes during holidays, or seasonal fluctuation in their business. The hybrid cloud can also meet your disaster recovery requirements. In this case, if a disaster occurs on the services deployed in the private cloud, the services can be transferred to the public cloud. This is a highly cost-effective approach. Another approach is to deploy part of your services on one public cloud, and use another public cloud for disaster recovery.

A hybrid cloud allows you to take advantage of both public and private clouds. It enables flexible transfer of applications among multiple clouds. In addition, it is cost-effective.

Of course, the hybrid cloud model has its disadvantages. You will face maintenance and security challenges due to complex settings in this model. In addition, because a hybrid cloud is a combination of different cloud platforms, data, and applications, integration can be a challenge. When developing a hybrid cloud, you may face issues of compatibility between infrastructures.

1.3 Mainstream Cloud Computing Vendors and Technologies

1.3.1 AWS

Amazon Web Services (AWS) is a cloud computing platform provided by Amazon. AWS provides users with a set of cloud computing services, including scalable computing, storage services, databases, and applications, helping enterprises reduce IT investment and maintenance costs. AWS provides a complete set of infrastructure and application services, enabling users to run almost all applications on the cloud, including enterprise applications, big data projects, social games, and mobile applications.

In addition, AWS has established an extensive partner ecosystem. AWS partners can gain support in the AWS-based businesses through a set of programs, including VMware Cloud on AWS, Distribution Program for Resellers, Managed Service Provider (MSP) Program, SaaS Factory Program, Competency Program, Public Sector Program, Marketplace Channel Programs.

AWS brings consulting partners and technology partners together to its platform. Consulting partners include system integrators, strategic and consulting vendors, agencies, managed service providers, and value-added distributors. AWS Partner Network (APN) is a community of independent software vendors (ISVs), and vendors distributing SaaS and PaaS, developer tools, as well as management and security solutions. AWS Cloud Control API gives developers a set of standardized APIs to build and run open source software in the cloud.

Main customers include enterprises that run e-commerce and media platforms, websites, and social applications.

AWS service value:

- Low price
- More usage

- Infrastructure expansion
- Economies of scale
- Technological innovation and ecosystem construction

1.3.2 VMware

1.3.2.1 VMware Overview

In 1998, VMware was founded. One year later, the company launched the commercial virtualization software VMware Workstation that can run smoothly on the x86 platform, marking its first step forward towards virtualization. In 2009, VMware launched VMware vSphere, the industry's first cloud operating system, and then launched the vCloud plan to build new cloud services.

VMware delivers private, public, and hybrid cloud solutions designed for specified service requirements.

VMware offers hybrid cloud products and services built based on the software-defined data center that brings together virtualized compute, storage, and networking.

VMware Cloud Foundation provides integrated cloud native infrastructure, making it easy to run enterprise applications in private environment.

VMware is a leading provider of multi-cloud services for all apps, enabling digital innovation through enterprise control. VMware vSphere helps you run, manage, connect and secure your applications in a common operating environment across its hybrid clouds and cloud native public clouds.

1.3.2.2 VMware Services

Figure 1-15 VMware services

Since its inception in 1998, VMware has been dedicated to providing customers with the flexibility and diversity required for building the future through disruptive technologies such as edge computing, artificial intelligence, blockchain, machine learning, and

Kubernetes. Currently, VMware provides services in four areas: digital workspace, cloud environment, application modernization, and Telco cloud.

- Digital workspace: Enable any employees to work from anywhere, anytime with seamless employee experiences.
- Cloud environment: Build, run, manage, connect, and secure all applications on any cloud.
- Application modernization: Modernize applications to accelerate digital innovation.
- Telco cloud: Build, run, manage, connect, and secure all applications on any cloud.

1.3.3 Huawei Cloud Overview

1.3.3.1 Huawei Cloud

Huawei Cloud is a public cloud service brand that leverages Huawei's more than 30 years of expertise in the ICT field to provide innovative, secure, and cost-effective cloud services.

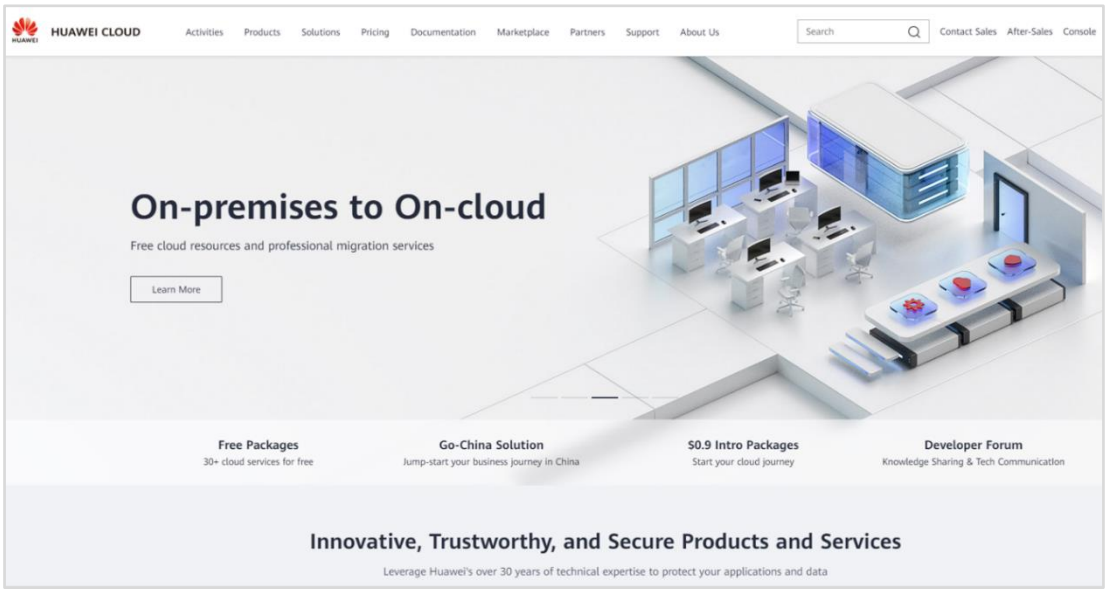


Figure 1-16 Huawei Cloud

In the previous chapter, we mentioned that public clouds are available for all users. You purchase any service you want on a public cloud portal simply with an official website account. All cloud service providers aim to minimize access latency when choosing their regions. In China, a place with mild winters and cool summers is ideal for a data center site to save on electricity. Guizhou, a province in southwestern China, is one such place. However, when a data center is deployed in Guizhou, the use of public cloud services in more distant regions would be prone to high latency due to the distance. As a result, public cloud vendors build data centers in regions that can ensure fast access from major cities. In addition to fast and stable access regardless of geographic location, diverse cloud services are essential for vendors to stay relevant. As one of the leading cloud service providers in the world, Huawei Cloud boasts a wide range of cloud services.

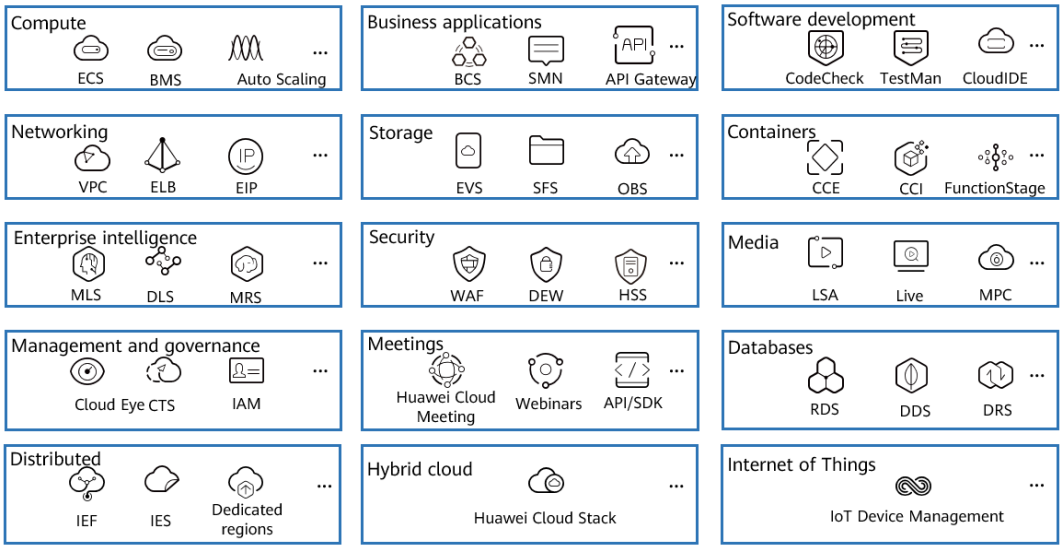


Figure 1-17 Huawei Cloud Services

Huawei Cloud has continuously upgraded its full-stack cloud native technologies. So far, they have launched 200+ cloud services and 200+ solutions.

1.3.3.2 Huawei Cloud Stack

Huawei Cloud Stack is cloud infrastructure deployed at the on-premises data centers of government and enterprise customers. It combines the advantages of private cloud and public cloud, allowing you to quickly launch innovative services like you always do on the public cloud and to manage your resources like you always do on the private cloud. Huawei Cloud Stack can adjust to your organizational structure and business processes, serving you as a single cloud. Huawei Cloud Stack can be used for medium and large enterprises that require local data storage or that require physical isolation of devices.

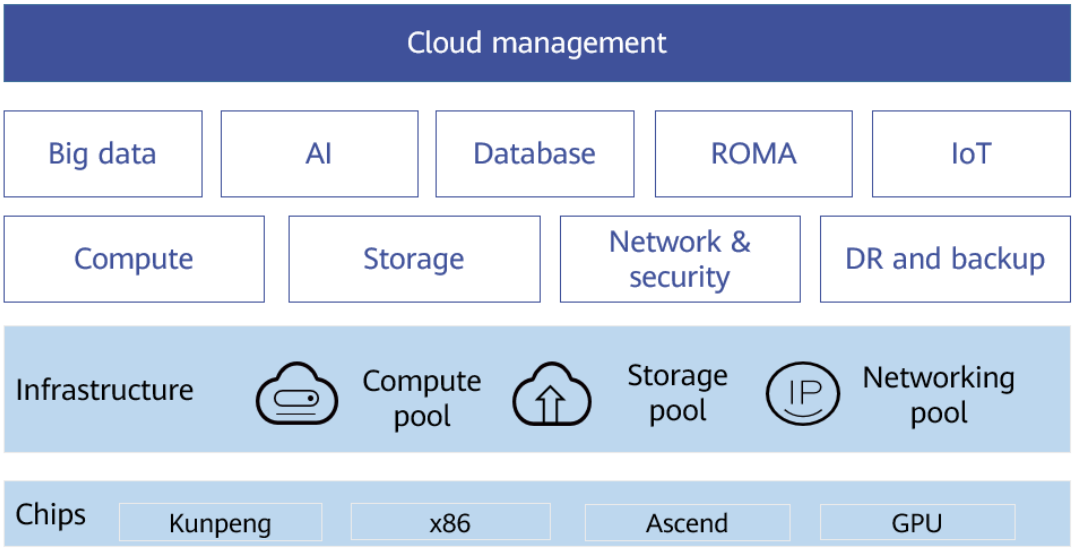


Figure 1-18 Huawei Cloud Stack

Huawei Cloud Stack can be used for cloud migration, cloud native transformation, big data analysis, AI applications, industry clouds, and city clouds.

Huawei Cloud Stack has the following advantages:

- AI enablement, data enablement, and application enablement: on-premises deployment of public cloud services;
- Multi-level cloud management: matching the enterprise governance architecture, featuring cloud federation, multi-level architecture, and intelligent O&M;
- Cloud-edge collaboration: extending intelligence to the edge, featuring unified framework, out-of-the-box edges, and video AI/IoT access;
- Secure and reliable: leading functions and performance, featuring full-stack security, one cloud with two pools, and strong ecosystem.

1.3.3.3 Huawei Cloud Data Centers: Innovative Chips

Figure 1-19 Overview of Huawei chip series

Chips are the core and most difficult part of R&D in the IT industry, which requires long-term investment.

Huawei has over 20 years of experience in chip R&D and is constantly innovating chips for the Cloud 2.0 era. We have launched a full series of chips for next-generation cloud data centers.

The full series of chips are:

- Compute chips: full series of AI processors;
- Network chips: Huawei's next-generation network chips Hi1822 use the NP-like programmable architecture and support offloading of multiple protocols;
- Storage chips: The fourth generation of storage chips improves the performance by over 75% and bandwidth by over 60%. Thanks to the intelligent multi-stream technology, the latency was decreased by about 15%;
- Security chips: Huawei has built security and trustworthiness into chips. They provide comprehensive protection for firmware, identities, software systems, and data management.

AWS, VMware, and Huawei Cloud supply cloud solutions featuring resource pooling, unified management, and on-demand self-service. They leverage virtualized computing, storage, and networking technologies to provide users with ultimate experience.

In the subsequent courses, let's take a closer look at these technologies and dig deeper into the principles of cloud computing.

1.4 Quiz

An engineer has purchased several cloud servers and other cloud resources on Huawei Cloud and wants to set up a simple forum website. He went on a business trip with the website unfinished. But he can also access the cloud servers to continue with his website at the hotel far away from his home. What value of cloud computing does this case reflect?

2 Server Basics

Servers are the foundation of all service platforms, including cloud computing platforms. But what is a server? What are the key technologies for servers? Let's find the answers in this chapter, and start our learning journey into cloud computing.

2.1 Introduction to Servers

2.1.1 What Is a Server?

2.1.1.1 Server Definition and Features

A server is a type of computer. It runs faster, carries more loads, and costs more than ordinary computers. A server provides services to users. There are file servers, database servers, and application servers.

A server is a mainstream computing product developed in 1990s. It can provide network users with centralized computing, information release, and data management services. In addition, a server can share drives, printers, and modems to which the server is connected, and dedicated communication devices with network users. So far, general-purpose servers are still the most widely used basic IT devices in enterprises, accounting for 90% or even higher of enterprise IT computing devices.

Servers are similar to PCs, but with features PCs do not have.

A server has the following features:

- R: Reliability – the duration that the server operates consecutively
- A: Availability – percentage of normal system uptime and use time
- S: Scalability – including hardware expansion and operating system (OS) support capabilities
- U: Usability – easy to maintain and restore server hardware and software
- M: Manageability – monitoring and alarm reporting of server running status, and automatically intelligent fault processing

The preceding features are necessary for a qualified server.

2.1.1.2 Server Application Scenarios

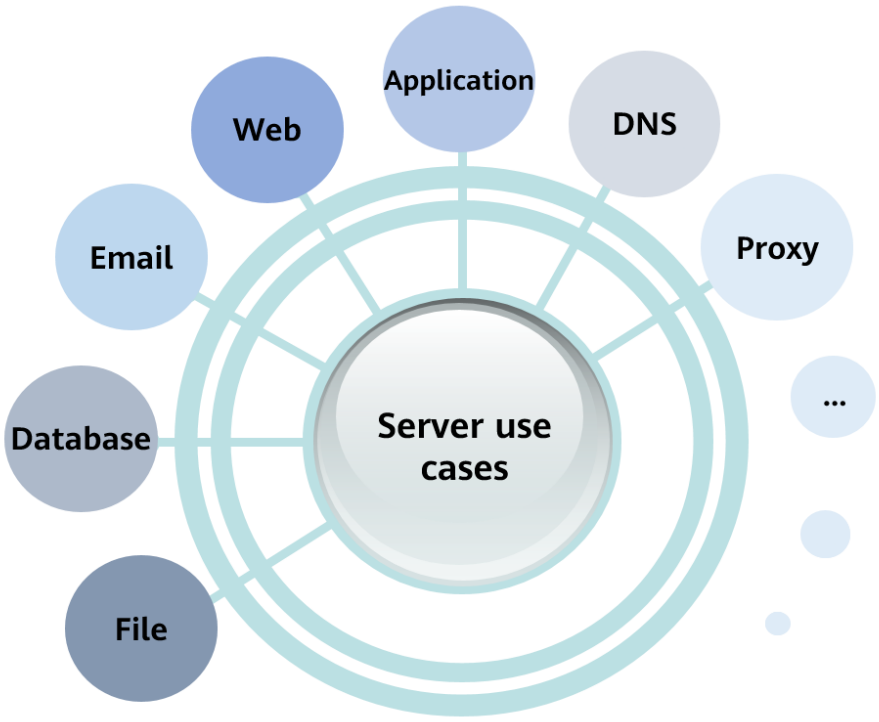


Figure 2-1 Server use cases

Servers have been widely used in various fields, such as the telecom, government, finance, education, enterprise, and e-commerce. Servers can provide users with the file, database, email, and web services.

As shown in Figure 2-1, there are two types of server application deployment architectures:

- C/S: short for Client/Server. In this architecture, the server program runs on the server, and the client software is installed on the client. The server and client perform different tasks. The client carries the front-end GUI and interaction operations of users, and the server processes the background service logic and request data. This greatly improves the communication speed and efficiency between the two ends. For example, you can install the vsftpd program on a file server and start the service. After you install the FileZilla or WinSCP client on your PC, you can upload and download files using the client.
- B/S: short for Browser/Server. In this architecture, users only need to install a browser. The application logic is centralized on the server and middleware, which improves the data processing performance. For example, when accessing a website, we only need to enter the domain name of the website in the browser, for example, www.huawei.com. Then we can see the web services provided by the background servers of the website. We do not need to care the background servers that provide services, such as the database service, proxy service, and cache service.

2.1.2 Server Development History

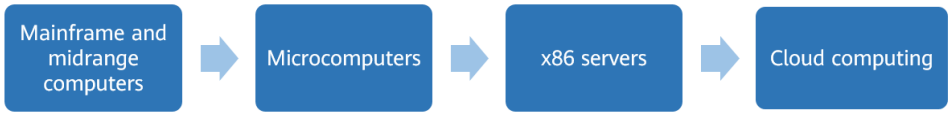


Figure 2-2 Server development history

Figure 2-2 shows the four phases of server development.

- Mainframe and midrange computers

In the 1940s and 1950s, the first generation of vacuum tube computers emerged. The computer technology develops rapidly from vacuum tube computers, transistor computers, integrated circuit computers, to large-scale integrated circuit computers.

In the 1960s and 1970s, mainframes were scaled down for the first time to meet the information processing requirements of small- and medium-sized enterprises and institutions. The cost was acceptable.

- Microcomputers

In the 1970s and 1980s, mainframes were scaled down for the second time. Apple Inc. was founded in 1976, and launched Apple II in 1977. In 1981, IBM launched IBM-PC. After several generations of evolution, it occupied the personal computer market and made personal computers popular.

- x86 servers

In 1978, Intel launched the first-generation x86 architecture processor, 8086 microprocessor.

In 1993, Intel officially launched the Pentium series, which brought the x86 architecture processor to a new level of performance.

In 1995, Intel launched Pentium Pro, the x86 processor for servers, ushering in the x86 era. The standardization and openness of Pentium Pro also promoted the market development and laid a solid foundation for the cloud computing era.

- Cloud computing

Since 2008, the concept of cloud computing has gradually become popular. Cloud computing is regarded as a revolutionary computing model because it enables the free flow of supercomputing capabilities through the Internet. Enterprises and individual users do not need to purchase expensive hardware. Instead, they can rent computing power through the Internet and pay only for the functions they need. Cloud computing allows users to obtain applications without the complexity of technologies and deployment. Cloud computing covers development, architecture, load balancing, and business models, and is the future model of the software industry.

The computing industry has developed for nearly half a century and continuously changed other industries. The computing industry itself is evolving.

In the early mainframe and midrange computer era, dedicated computing is used, which is called computing 1.0. In the x86 era, under the leadership of Intel and driven by Moore's Law, computing has shifted from dedicated to general-purpose. A large number of data centers have emerged, which is called computing 2.0. With the rapid development of digitalization, the world is developing towards intelligent. Computing is

not limited to data centers, but also enters the full-stack all-scenario (computing 3.0) era. This era is featured by intelligence, so it is also called intelligent computing.

2.1.3 Server Types

2.1.3.1 Server Classification - Hardware Form

Server Category					
	Mainframe	Midrange computer	Tower server	Blade server	Rack server
Hardware form					

Figure 2-3 Server classification - Hardware form

As shown in Figure 2-3, servers can be classified into the following types based on hardware forms:

- Tower server

Some tower servers use a chassis roughly the same size as an ordinary vertical computer, while others use a large-capacity chassis, like a large cabinet.

- Rack server

A rack server looks different from a computer, but looks similar to a switch. The specifications of a rack server include 1 U (1 U = 1.75 inches), 2 U, and 4 U. A rack server is installed in a standard 19-inch cabinet. Most of the servers in this structure are functional servers. A rack server is usually small in size. Multiple servers can be placed in a cabinet at the same time to obtain a higher processing capability.

- Blade server

Each blade server is a plugboard equipped with processors, memory modules, hard drives, and related components. Due to the special architecture, blade servers require dedicated chassis. Generally, a chassis can hold several to dozens of blade servers, suitable for scenarios such as high-performance computing, front-end servers running multiple applications, and backend central databases.

For details about mainframes and midrange computers, see the preceding description.

2.1.3.2 Server Classification - Service Scale

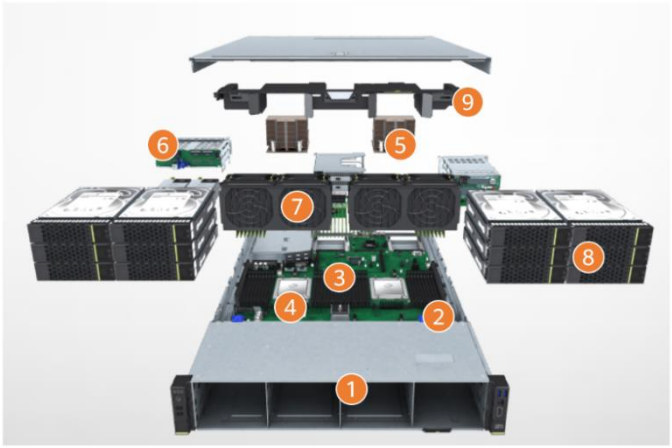
Server Category				
Service scale	Entry-level server	Work group server	Department-level server	Enterprise-level server
	Similar to a PC server	Low-end server that provides small-scale services (about 50 clients)	Mid-range server that serves about 100 clients	High-end server that is accessed by hundreds of clients

Figure 2-4 Server classification - Service scale

As shown in Figure 2-4, servers can be classified into entry-level servers, work group servers, department-level servers, and enterprise-level servers based on the service scale.

2.1.4 Server Hardware

2.1.4.1 Hardware Structure



1 Chassis
2 Motherboard
3 Memory
4 CPU
5 CPU heat sink
6 Power supply unit (PSU)
7 Fan
8 Drive
9 Air duct

Figure 2-5 Hardware structure

Figure 2-5 shows the hardware structure of a Huawei TaiShan 200 server. You can learn about the internal components of the server and their locations. Important hardware in the server, such as the CPU, memory, and drive, will be described in detail later.

2.1.4.2 CPU

2.1.4.2.1 CPU Definition and Components

The Central Processing Unit (CPU) is the computing and control core of a computer.

The CPU is the core processing unit on a server, and a server is an important device on the network and needs to process a large number of access requests. Therefore, servers must have high throughput and robust stability, and support long-term running. The CPU is the brain of a computer and is the primary indicator for measuring server performance.

The CPU, internal storage, and input/output devices are key components of a computer. The CPU interprets computer instructions and processes computer software data.

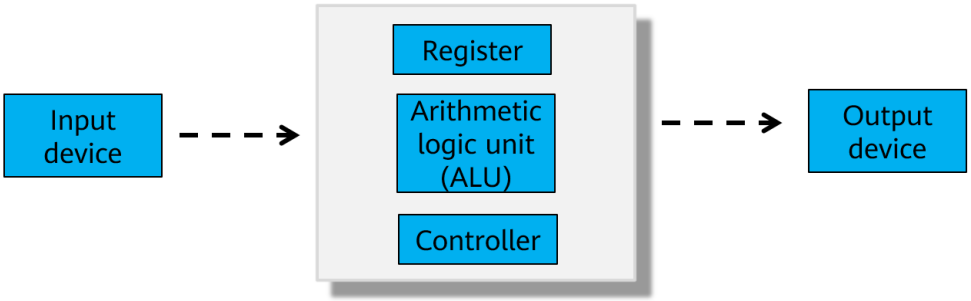


Figure 2-6 CPU structure

As shown in Figure 2-6, the middle part is the CPU structure. The CPU consists of a logic operation unit, a control unit, and a storage unit.

- The computer controls the entire computer according to a pre-stored program, and the program refers to an instruction sequence that can implement a function. The controller is an organization that issues commands to various logic circuits according to the instructions. The controller is a command center of the computer, controls work of an entire CPU, and determines automation of a running process of the computer.
- The ALU is a part of a computer that performs a variety of arithmetic and logical operations. Basic operations of an ALU include arithmetic operations such as addition, subtraction, multiplication, and division, logical operations such as AND, OR, NOT, and XOR, and other operations such as shift, comparison, and transfer. The ALU is also called the arithmetic logic component.
- The register is used to temporarily store the data involved in operations and the operation results. It can receive, store, and output data.

2.1.4.2.2 CPU Frequency

Generally, the following CPU frequency parameters are used to measure CPU performance:

- The dominant frequency is also called clock speed. It indicates, in MHz or GHz, the frequency at which a CPU computes and processes data.
- The external frequency is the reference frequency of a CPU, measured in MHz. The CPU external frequency determines the speed of the motherboard.
- The bus frequency directly affects the speed of data exchange between a CPU and a dual in-line memory module (DIMM).
- The multiplication factor is the ratio of the dominant frequency to the external frequency.

2.1.4.3 Memory

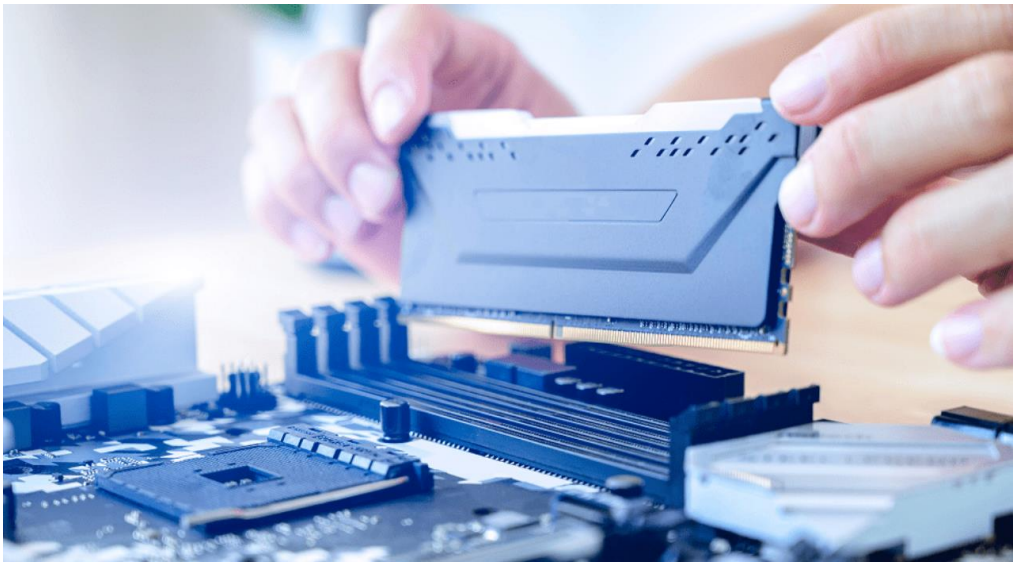


Figure 2-7 Memory

The storage, an important computer component, is used to store programs and data. For computers, the memory function can be supported and normal working can be ensured only when the storage is available. Storage is classified, by purpose, into main memory and external storage. Main memory, referred to as internal storage, is the storage space that the CPU can address.

As a main computer component, the memory is in opposition to the external storage. Programs, such as the Windows OS, typing software, and game software, are usually installed on external storage devices such as drives. To use these programs, you must load them into the memory. Actually, the memory is used when we input a piece of text or play a game. Bookshelves and bookcases for putting books are just like the external storage, while the desk is like the memory. Generally, we store large volumes of data permanently in the external storage and store small volumes of data and a few programs temporarily in the memory.

The memory, one of important computer components, communicates with the CPU. The memory consists of the memory chip, circuit card, and edge connector.

Comply with the following principles when installing DIMMs on servers:

- DIMMs on the same server must be of the same model.
- At least one DIMM must be configured in slots supported by CPU 1.
- Optimal memory performance can be achieved if the processors in a server are configured with the same number of DIMMs and the DIMMs are evenly distributed among the memory channels. Unbalanced configuration impacts memory performance and is not recommended.

2.1.4.4 Drive

The drive is the most important storage device of a computer.

The drive interface, connecting a drive to a host, is used to transmit data between the drive cache and the host memory. The drive interface type determines the connection

speed between the drive and the computer, how quickly programs run, and overall system performance.

	SATA	SAS	NL-SAS	SSD
Rotational speed (RPM)	7,200	15,000/10,000	7,200	N/A
Serial/Parallel	Serial	Serial	Serial	Serial
Capacity (TB)	1 TB/2 TB/3 TB	0.6 TB/0.9 TB	2 TB/3 TB/4 TB	0.6 TB/0.8 TB/1.2 TB/1.6 TB
MTBF (h)	1,200,000	1,600,000	1,200,000	2,000,000
Remarks	Developed from ATA drives, SATA 3.0 supports data transfer up to 600 MB/s. The annual failure rate of SATA drives is about 2%.	SAS drives are designed to meet high-performance enterprise requirements and are compatible with SATA drives. The transfer rate ranges from 3.0 Gbit/s to 6.0 Gbit/s, and can increase to 12.0 Gbit/s. The annual failure rate of SAS drives is less than 2%.	An NL-SAS drive is an enterprise-level SATA drive with a SAS interface. It is used to implement tiered storage in a drive array, simplifying drive array design. The annual failure rate of NL-SAS drives is about 2%.	A solid-state drive (SSD) is a hard drive housing a solid-state electronic storage chip array. An SSD consists of a control unit and a storage unit (flash or DRAM chip). An SSD is the same as a common hard drive in terms of interface specifications and definition, function, usage, and product shape and size.

Figure 2-8 Drive types

Common drive types include SATA, SAS, NL-SAS, and SSD. The preceding table lists their characteristics. MTBF is short for Mean Time Between Failures. A larger value indicates a lower failure rate of the drive. SATA and NL-SAS drives are cheaper, SAS drives are more expensive, and SSDs are the most expensive.

2.1.4.5 RAID Controller Card

2.1.4.5.1 RAID Overview

Generally, a server has a RAID controller card. The RAID controller card is also called the disk array card. To understand a RAID controller card, you need to know what RAID is.

Redundant Array of Independent Disks (RAID) is a data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both. For details about the working principles of RAID, see the chapter of storage basics.

The RAID controller card provides the following functions:

- Combines several drives into a system that is managed by the array controller according to certain requirements.
- Improves the performance and reliability of the disk subsystem.

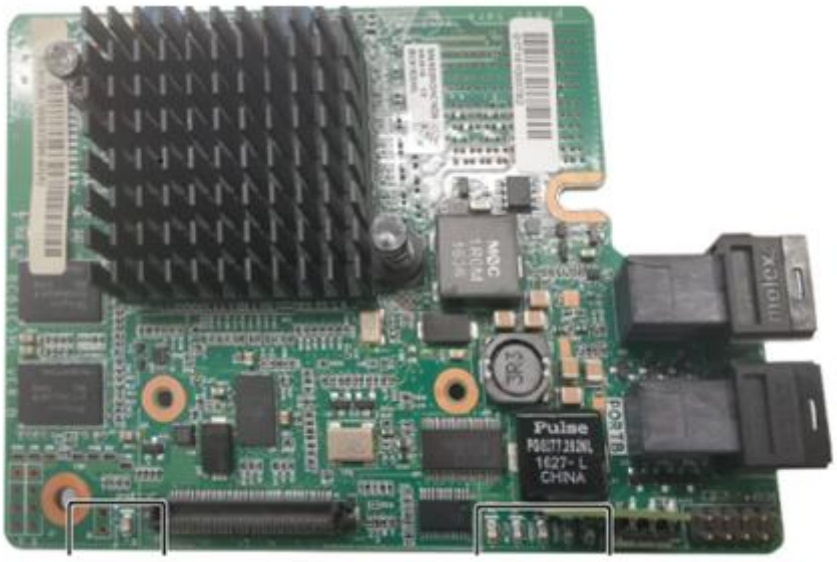


Figure 2-9 LSI SAS3108 RAID controller card

2.1.4.5.2 RAID Hot Spare and Reconstruction

Hot spare definition: If a drive in a RAID array fails, a hot spare is used to automatically replace the failed drive to maintain the RAID array's redundancy and data continuity.

Hot spare is divided into the following two types:

- Global: The spare drive is shared by all RAID arrays in the system.
- Dedicated: The spare drive is used only by a specific RAID array.

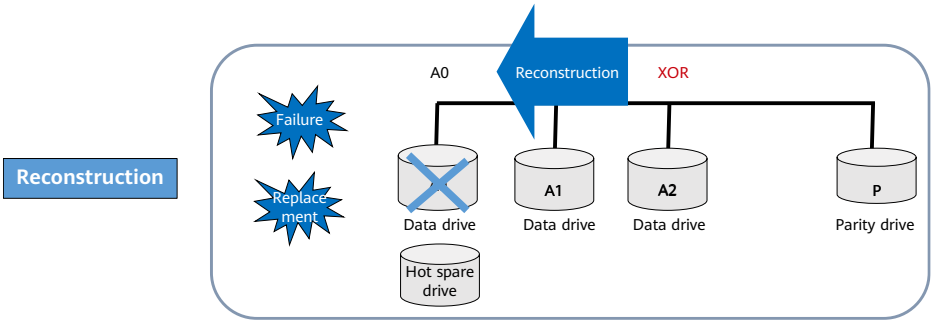


Figure 2-10 Hot spare and reconstruction

The process of restoring data from a faulty data drive to a hot spare drive is called data reconstruction. Generally, the data parity mechanism in RAID is used to reconstruct data.

Data parity: Redundant data is used to detect and rectify data errors. The redundant data is usually calculated through Hamming check or XOR operations. Data parity can greatly improve the reliability, performance, and error tolerance of the drive arrays. However, the system needs to read data from multiple locations, calculate, and compare data during the parity process, which affects system performance.

Generally, RAID cannot be used as an alternative to data backup. It cannot prevent data loss caused by non-drive faults, such as viruses, man-made damages, and accidental deletion. Data loss here refers to the loss of OS, file system, volume manager, or application system data, not the RAID data loss. Therefore, data protection measures,

such as data backup and disaster recovery, are necessary. They are complementary to RAID, and can ensure data security and prevent data loss at different layers.

2.1.4.5.3 RAID Implementation - Hardware

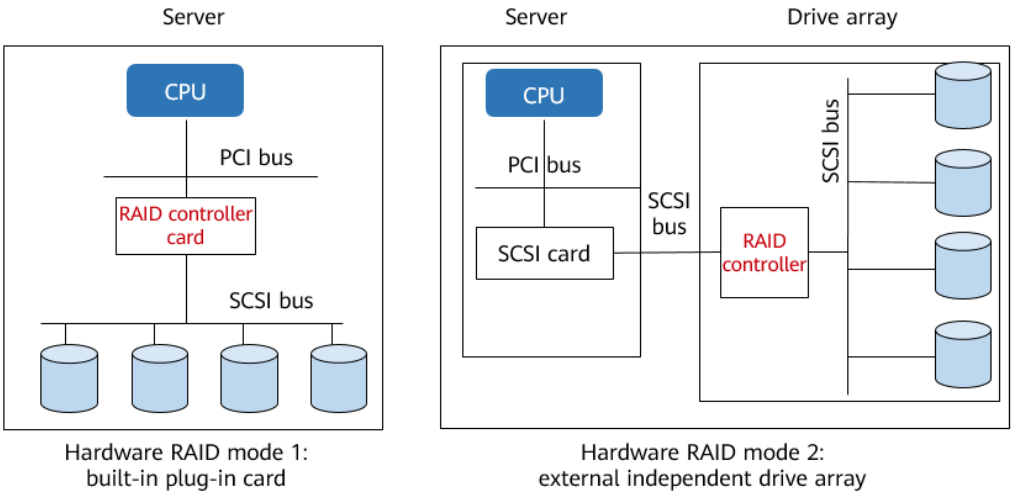


Figure 2-11 RAID implementation - Hardware

As shown in Figure 2-11, hardware RAID is implemented using a hardware RAID adapter card.

The hardware RAID can be a built-in or external RAID.

A RAID controller card has a processor inside and can control the RAID storage subsystem independently from the host. The RAID controller card has its own independent processor and memory. It can calculate parity information and locate files, reducing the CPU computing time and improving the parallel data transmission speed.

2.1.4.5.4 RAID Implementation - Software

Software RAID implements RAID functions by installing software on the OS.

Software RAID has the following characteristics:

- Software RAID does not require expensive RAID controller cards, reducing the cost.
- RAID functions are performed by CPUs, requiring significant CPU resources, such as for large numbers of RAID 5 XOR operations.

Compared with hardware RAID, software RAID has other defects. For example, software RAID does not support the following functions:

- Hot swap of drives
- Drive hot spare
- Remote array management
- Support for bootable arrays
- Array configuration on drives
- S.M.A.R.T. for drives

2.1.4.5.5 RAID Implementation - Mode Comparison

Figure 2-12 compares software RAID and hardware RAID.

Mode	Software RAID	Built-in RAID	External RAID
Characteristics	All RAID functions are implemented by CPUs, resulting in high CPU usage and reduced system performance.	Built-in RAID improves performance by reducing host CPU usage caused by intensive RAID operations.	External RAID, connecting to a server through a standard controller, is independent of the operating system. All RAID functions are implemented by the microprocessor on the external RAID storage subsystem.
Advantages	▫ Low implementation cost ▫ Flexible configurations	▫ Data protection and high speed ▫ Better fault tolerance and performance than software RAID ▫ More cost-effective than external RAID ▫ Support for bootable arrays	▫ Provides ultra-large-capacity storage systems for high-end servers. ▫ Configures dual controllers to improve data throughput or provide shared storage for the two-node cluster. ▫ Supports hot swapping. ▫ Delivers better scalability.

Figure 2-12 RAID implementation - Mode comparison

2.1.4.6 NIC

A network interface card (NIC or network adapter) is an indispensable part of a computer network system. An NIC enables a computer to access networks.

For a server, the NIC provides the following functions:

- Fixed network address
- Data sending and receiving
- Data encapsulation and decapsulation
- Link management
- Encoding and decoding

Huawei servers have the following four types of NICs:

- LOM card

It is embedded directly into the PCH chip on the server motherboard and cannot be replaced.

It provides two external GE electrical ports + two 10 Gbit/s optical/electrical ports. LOM cards do not occupy PCIe slots.

- PCIe card

Huawei has both self-developed and purchased PCIe cards. They can be installed in standard PCIe slots.

- FlexIO card

Huawei-developed, non-standard PCIe card, which can only be used with Huawei rack servers.

- Mezzanine card

Mezzanine cards are only used on the compute nodes of Huawei E9000 blade servers.

2.1.4.7 PSU and Fan Module

Figure 2-13 PSU and fan module

As shown in Figure 2-13, the power supply unit (PSU) and fan module support the power load and heat dissipation of the server. Redundant PSUs and fan modules are deployed on servers to ensure continuous running of the system.

The following power supply redundancy modes are available:

- 1+1: In this mode, each module provides 50% of the output power. When one module is removed, the other provides 100% of the output power.
- 2+1: In this mode, each module provides 1/3 of the output power. When one module is removed, each of the other two modules provides 50% of the output power.

2.1.5 Key Server Technologies

2.1.5.1 BMC

2.1.5.1.1 What Is IPMI?

The Intelligent Platform Management Interface (IPMI) is a set of open and standard hardware management interface specifications that defines specific methods for communication between embedded management subsystems.

The IPMI is an industrial specification used for peripherals in Intel-based enterprise systems. This interface specification was laid down by Intel, HP, NEC, Dell, and SuperMicro. Users can use the IPMI to monitor the physical health status of servers, such as the temperature, voltage, fan status, and power status. Moreover, the IPMI is a free specification. Users do not need to pay for this specification.

IPMI development:

- In 1998, Intel, DELL, HP, and NEC put forward the IPMI specification. The temperature and voltage can be remotely controlled through the network.
- In 2001, the IPMI was upgraded from version 1.0 to version 1.5. The PCI Management Bus function was added.
- In 2004, Intel released the IPMI 2.0 specification, which is compatible with the IPMI 1.0 and 1.5 specifications. Console Redirection is added. Servers can be remotely

managed through ports, modems, and LANs. In addition, security, VLANs, and blade servers are supported.

IPMI information is exchanged using the baseboard management controller (BMC). Entry-level intelligent hardware, not the OS, handles management.

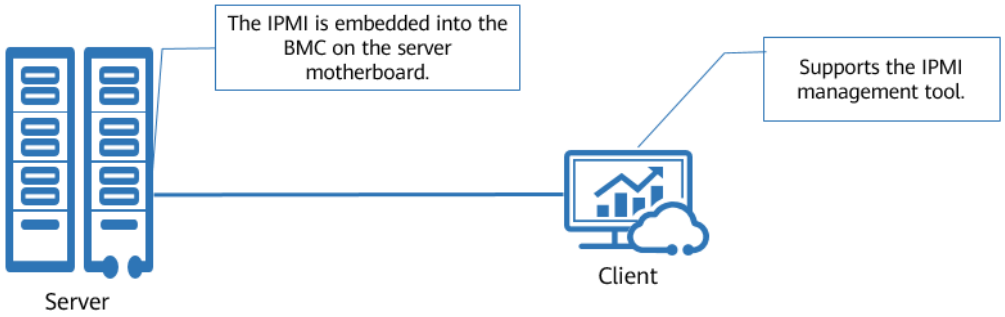


Figure 2-14 IPMI

The IPMI management tool on the client remotely manages the server through the IPMI port on the server.

2.1.5.1.2 BMC

The BMC complies with the IPMI specification. It collects, processes, and stores sensor signals, and monitors component operating status. It supplies the chassis management module with managed objects' hardware status and alarm information. The management module uses this information to manage the devices.

The BMC provides the following functions:

- Remote control
- Alarm management
- Status check
- Device information management
- Heat dissipation control
- Support for IPMItool
- Web-based management
- Centralized account management

2.1.5.1.3 iBMC

The Huawei Intelligent Baseboard Management Controller (iBMC) is a Huawei proprietary embedded server management system designed for the whole server lifecycle.

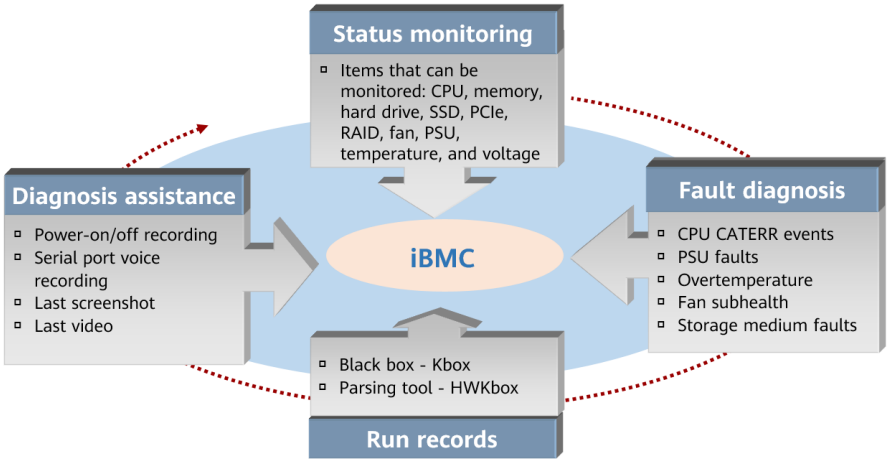


Figure 2-15 iBMC

The iBMC provides a series of management tools for hardware status monitoring, deployment, energy saving, and security, and standard interfaces to build a comprehensive server management ecosystem. The iBMC uses Huawei-developed management chip Hi1710 and multiple innovative technologies to implement refined server management.

The iBMC provides a variety of user interfaces, such as the CLI, web-based user interface, IPMI integration interface, SNMP integration interface, and Redfish integration interface. All user interfaces adopt the authentication mechanism and high-security encryption algorithm to enhance access and transmission security.

2.1.5.2 BIOS

The Basic Input/Output System (BIOS) is a system's foundation: a group of programs providing the most direct control of system hardware.

The BIOS is a bridge between the system kernel and the hardware layer. The BIOS is a small system at the motherboard level. It initializes the hardware of the system (mainly the motherboard), such as the CPU, memory, drive, keyboard, graphics card, and NIC. Initialization is the main task of the BIOS. The BIOS on a traditional PC has an int19 software interrupt function. After the initialization is complete, the BIOS enters the int19 interrupt mode to search for the boot medium, such as a floppy disk, CD-ROM, hard drive, flash memory, or network, reads the content of the first sector to 0000:7C00 in the memory, and goes to this address. int19 is an OS bootloader. Therefore, the BIOS can boot the OS. Of course, many BIOS functions have been added, for example, ACPI interface for power management, USB driver, PXE network boot function, drive encryption, TPM interface, BIOS configuration interface, and BIOS automatic recovery.

The BIOS provides the following functions:

- Hardware detection and initialization
- OS boot
- Advanced power management

2.2 Quiz

Which types of hardware are critical in the hardware structure of the server? What are their functions?

3 Storage Technology Basics

Data is the most important asset for every user. This chapter describes how and where data is stored, and provide the key data storage technologies in cloud computing.

3.1 Storage Basics

3.1.1 What Is Storage

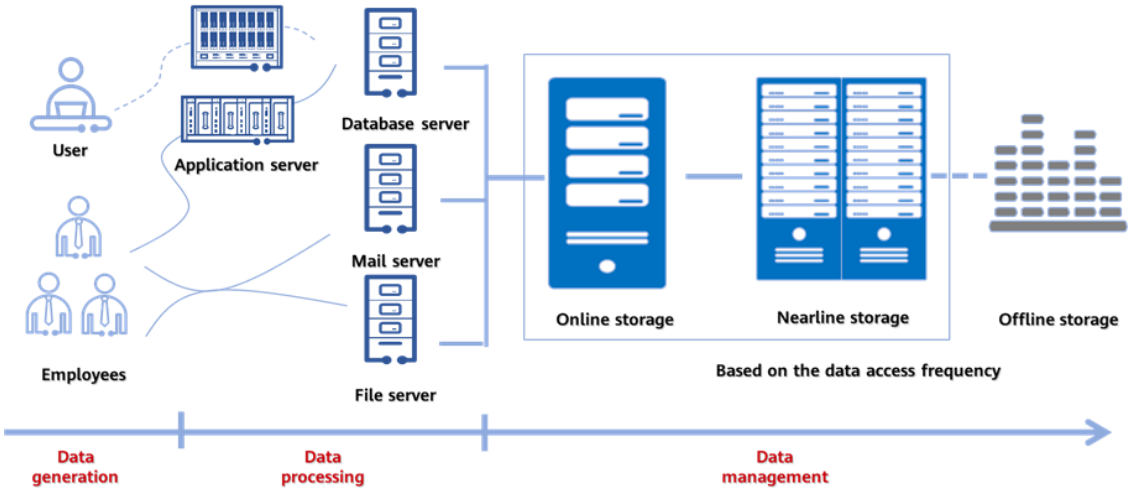


Figure 3-1 Data storage procedure

Figure 3-1 shows the generation, processing, and management of data. Storage is used to manage data.

Storage is defined in a narrow sense and broad sense.

The narrow-sense storage refers to specific storage devices, such as CDs, DVDs, Zip drives, tapes, and disks.

The broad-sense storage consists of the following four parts:

- Storage hardware (disk arrays, controllers, disk enclosures, and tape libraries)
- Storage software (backup software, management software, and value-added software such as snapshot and replication)
- Storage networks (HBAs, Fibre Channel switches, as well as Fibre Channel and SAS cables)
- Storage solutions (centralized storage, archiving, backup, and disaster recovery)

3.1.2 History of Storage

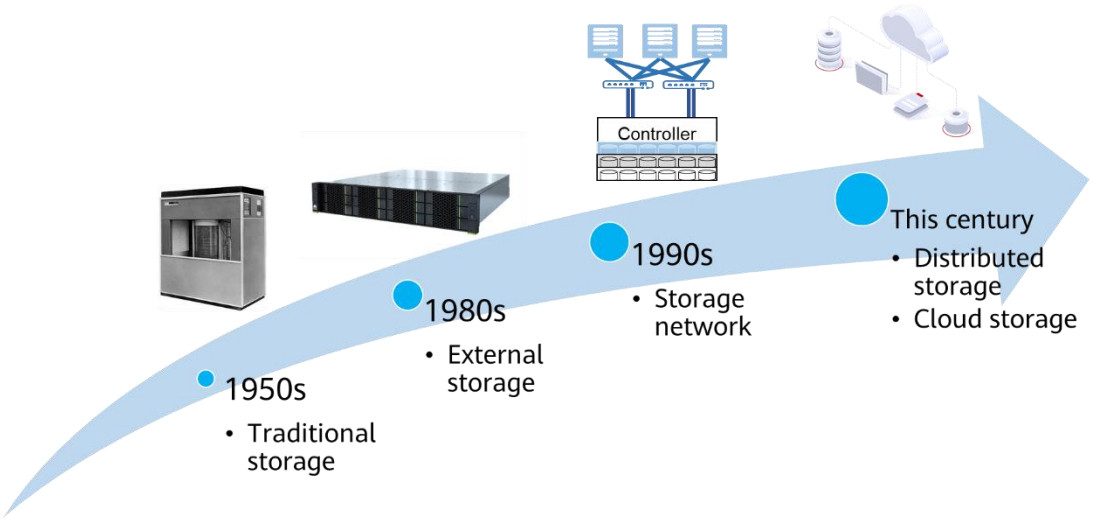


Figure 3-2 History of storage

As shown in Figure 3-2, the storage architecture has gone through the following development phases: traditional storage, external storage, storage network, distributed storage, and cloud storage.

- Traditional storage: refers to individual disks. In 1956, IBM invented the world's first mechanical hard drive that has fifty 24-inch platters and the total storage capacity of just 5 MB. It is about the size of two refrigerators and weighs more than a ton. It was used in the industrial field at that time and was independent of the mainframe.
- External storage refers to direct-attached storage. The earliest form of external storage is JBOD, which stands for Just a Bunch Of Disks. JBOD is identified by the host as a stack of independent disks. It provides large capacity but low security.
- Storage network: A storage area network (SAN) is a typical storage network that transmits data mainly over a Fibre Channel network. Then, IP SANs emerge.
- Distributed storage and cloud storage: Distributed storage uses general-purpose servers to build storage pools and is more suitable for cloud computing.

3.1.2.1 Storage Development - from Server Attached Storage to Independent Storage Systems

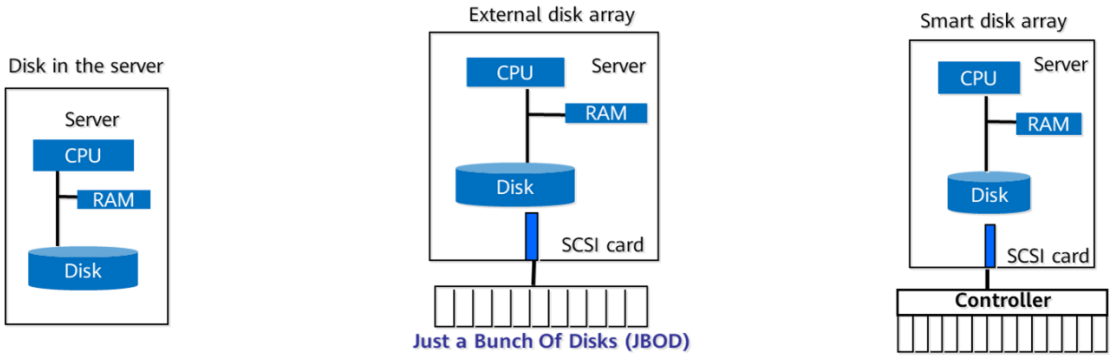


Figure 3-3 Development from server attached storage to independent storage systems

In the early phase of enterprise storage, disks are built in servers. As storage technologies develop, the limitations of this architecture gradually emerge.

- Disks in the server are prone to become system performance bottleneck.
- The number of disk slots is limited, thereby limiting capacity.
- Data is stored on individual disks, resulting in poor reliability.
- Storage space utilization is low.
- Data is scattered in local storage systems.

To meet new storage requirements, external disk arrays are introduced. Just a Bunch Of Disks, or JBOD combines multiple disks to provide storage resources externally. It just refers to a collection of disks without control software to coordinate and control resources and does not support redundant array of independent disks, or RAID. This architecture resolves the problem of limited disk slot quantities of the server, thereby improving system capacity.

As the RAID technology emerges, disk arrays with the RAID technology used become smarter. RAID resolves the problems of the limited disk interface performance and the poor reliability of individual-disk storage.

3.1.2.2 Storage Development: from Independent Storage Systems to Network Shared Storage

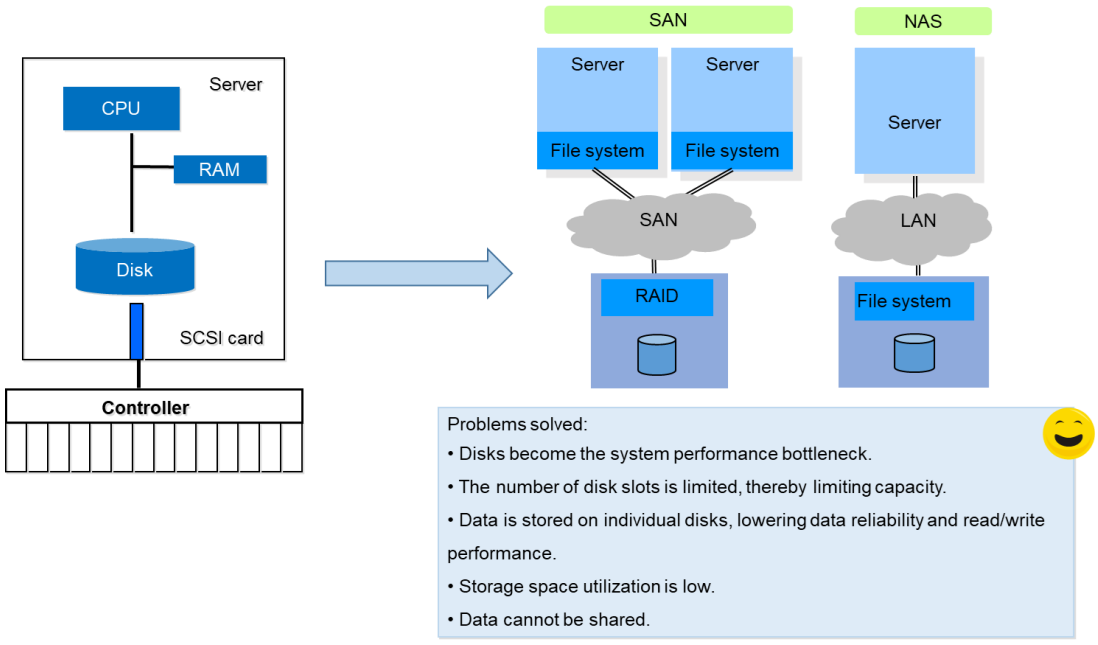


Figure 3-4 Development from independent storage systems to network shared storage

As mentioned in the previous section, the direct connection between the storage and server through the controller resolves the problems caused by the limited disk slot quantities, individual-disk storage, and limited disk interface performance.

However, other problems remain, such as low storage space utilization, decentralized data management, and inconvenient data sharing. We will learn how the network shared storage such as SAN and NAS solves these pain points.

3.1.3 Mainstream Disk Types

The concept of disks has been described in **2 Server Basics**, and details are not described herein again.

	SATA	SAS	NL-SAS	SSD
Rotational speed (rpm)	7,200	15,000/10,000	7,200	N/A
Serial/Parallel	Serial	Serial	Serial	Serial
Capacity (TB)	1 TB/2 TB/3 TB	0.6 TB/0.9 TB	2 TB/3 TB/4 TB	0.6 TB/0.8 TB/1.2 TB/1.6 TB
MTBF (h)	1,200,000	1,600,000	1,200,000	2,000,000
Remarks	Being developed from ATA disks, SATA 3.0 supports up to 600 MB/s data transmission rate. The annual failure rate of SATA disks is about 2%.	SAS disks are designed to meet enterprises' high performance requirements, and are compatible with SATA disks. The transfer rate ranges from 3.0 Gbit/s to 6.0 Gbit/s, and will be increased to 12.0 Gbit/s. The annual failure rate of SAS disks is less than 2%.	NL-SAS disks are enterprise-class SATA drives with SAS interfaces. They can be applied in a disk array to implement storage tiering, which simplifies the design of the disk array. The annual failure rate of NL-SAS disks is about 2%.	Solid state disks (SSDs) are made up of solid-state electronic storage chip arrays. Each SSD consists of a control unit and a storage unit (DRAM or flash chip). SSDs are the same as the common disks in the regulations and definition of interfaces, functions, usage, as well as the exterior and size.

Figure 3-5 Mainstream disk types

To understand disks, we need to know some disk metrics, such as disk capacity, rotational speed, average access time, date transfer rate, and input/output operations per second (IOPS). Rotational speed is specific to HDDs.

- Disk capacity is measured in MB or GB. The factors that affect the disk capacity include the single platter capacity and the number of platters.
- Rotational speed is the number of rotations made by disk platters per minute. The unit is rotation per minute (rpm). In most cases, the rotational speed of a disk reaches 5400 rpm or 7200 rpm. The disk that uses the SCSI interface reaches 10,000 rpm to 15,000 rpm.
- Average access time is the average seek time plus the average wait time.
- Data transfer rate of a disk is the speed at which data is read from or written to the disk. It is measured in MB/s. The data transfer rate consists of the internal data transfer rate and the external data transfer rate.
- IOPS indicates the number of input/output operations or read/write operations per second. It is a key metric to measure disk performance. For applications with frequent random read/write operations, such as online transaction processing (OLTP), IOPS is a key metric. Another key metric is the data throughput, which indicates the amount of data that can be successfully transferred per unit time. For applications that require a large number of sequential read/write operations, such as video editing and video on demand (VoD) at TV stations, the throughput is more of a focus.

When measuring the performance of a disk or storage system, we usually consider the following metrics: average access time, data transfer rate, and IOPS. To be specific, shorter average access time, higher data transfer rate, and higher IOPS indicate better disk performance.

3.1.4 Storage Networking Types

3.1.4.1 Introduction to DAS

As shown in Figure 3-6, direct attached storage (DAS) is a type of storage that is attached directly to a computer through the SCSI or Fibre Channel interface. DAS does not go through a network so that only the host to which the storage device is attached can access it. That is, if a server is faulty, the data in the DAS device that connects to the server is unavailable. Common interfaces include small computer systems interface (SCSI), serial attached SCSI (SAS), external SATA (eSATA), serial ATA (SATA), Fibre Channel, USB 3.0 and Thunderbolt.

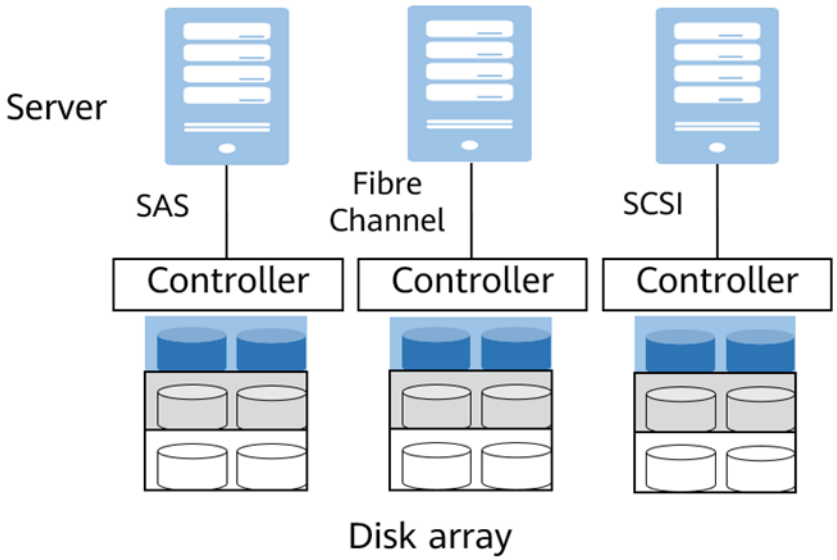


Figure 3-6 Architecture of DAS

The DAS device communicates with the server or host through SAS channels (3 Gbit/s, 6 Gbit/s, and 12 Gbit/s). However, with the strengthening of CPU processing capability, expanding of storage disk space, and increasing of disk quantities in a disk array, the SAS channel will become the I/O bottleneck. Limited SAS interface resources of a server host limit the channel bandwidth.

3.1.4.2 Introduction to NAS

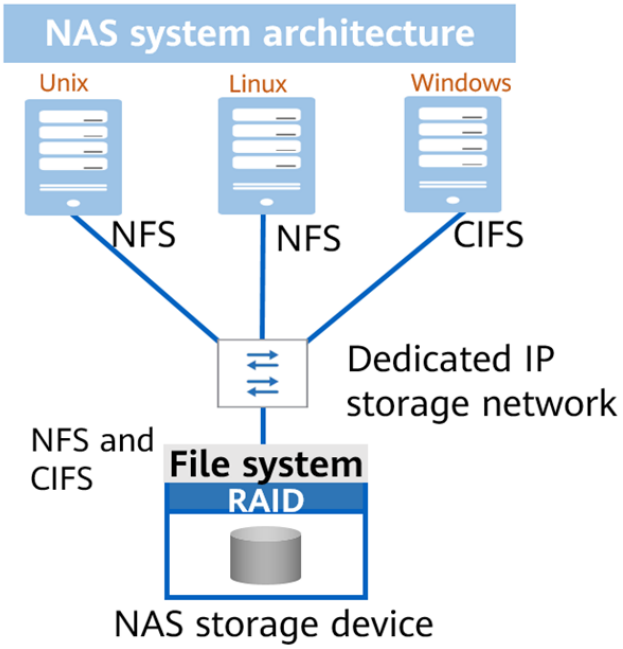


Figure 3-7 Architecture of NAS

As shown in 3.1.4.2 Figure 3-7, network attached storage (NAS) is a type of storage that connects to a group of computers through a standard network (for example, an Ethernet network). A NAS device has a file system and an assigned IP address, and may be regarded as a shared disk in Network Neighborhood.

Developing networks drove the need for large-scale data sharing and exchange, leading to dedicated NAS storage devices.

Access mode: Multiple front-end servers share space on back-end NAS storage devices using CIFS or NFS. Concurrent read and write operations can be performed on the same directory or file.

In the NAS system, Linux clients mainly use Network File System (NFS) protocol, and Windows clients mainly use Common Internet File System (CIFS) protocol. The NAS file system is on the back-end storage device.

NFS is an Internet standard protocol created by Sun Microsystems in 1984 for file sharing between systems on a local area network (LAN).

It uses the Remote Procedure Call (RPC) protocol.

- RPC provides a set of operations to achieve remote file access that are not restricted by machines, operating systems (OSs), and lower-layer transmission protocols. It allows remote clients to access storage over a network like accessing a local file system.
- The NFS client sends an RPC request to the NFS server. The server transfers the request to the local file access process, reads the local disk files on the server, and returns the files to the client.

CIFS is a network file system protocol used for sharing files and printers between machines on a network. It is mainly used to share network files between hosts running Windows.

NAS is a file-level storage architecture that meets the requirements of work teams and departments on quick storage capacity expansion. Currently, NAS is widely used to share documents, images, and movies. NAS supports multiple protocols (such as NFS and CIFS) and supports various OSs. Users can conveniently manage NAS devices by using Internet Explorer or Netscape on any work station.

3.1.4.3 Introduction to SAN

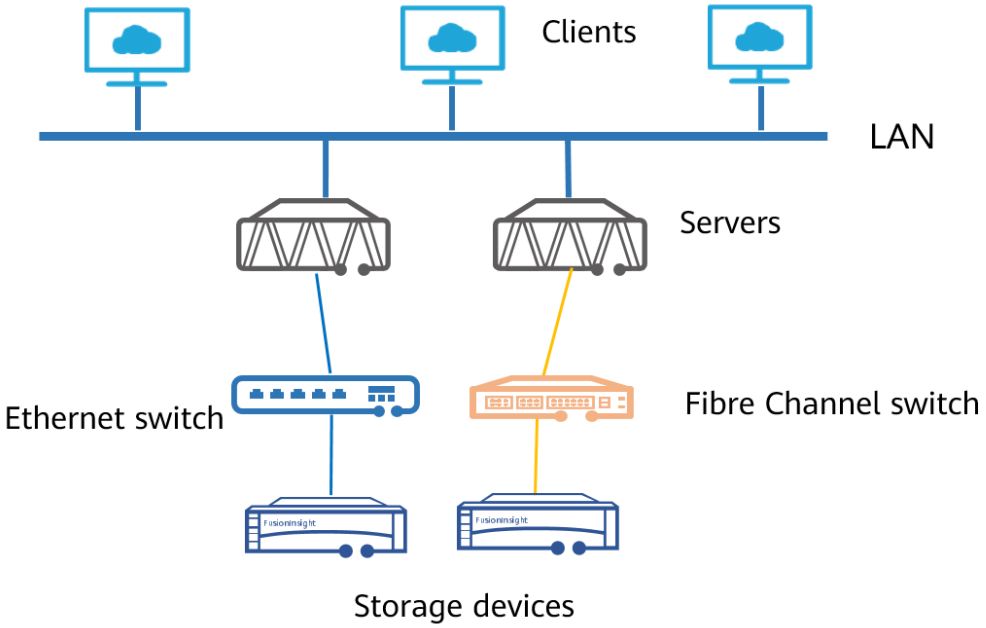


Figure 3-8 Architecture of SAN

The storage area network (SAN) is a dedicated storage network that connects one or more network storage devices to servers. It is a high-performance and dedicated storage network used between servers and storage resources. In addition, it is a back-end storage network independent from a LAN. The SAN adopts a scalable network topology for connecting servers and storage devices. The storage devices do not belong to any of the servers but can be shared by all the servers on the network.

SAN features fast data transmission, high flexibility, and reduced network complexity. It eliminates performance bottlenecks of the traditional architecture and massively improves the backup and disaster recovery efficiency of remote systems.

A SAN is a network architecture that consists of storage devices and system components, including servers that need to use storage resources, host bus adapters (HBAs) that connect storage devices, and Fibre Channel switches.

On a SAN, all communication related to data storage is implemented on a network independent of the application network. Therefore, SAN improves I/O capabilities of the entire network without affecting the existing application network, offers a backup connection for the storage system, and supports high availability (HA) cluster systems.

With the development of SAN technologies, three SAN types are made available: FC SAN, IP SAN, and SAS SAN. The following describes FC SAN and IP SAN.

3.1.4.3.1 Introduction to FC SAN

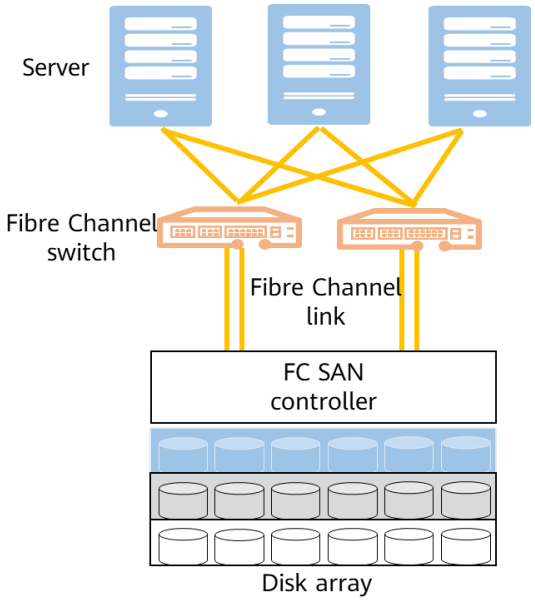


Figure 3-9 Architecture of FC SAN

As shown in Figure 3-9, on an FC SAN, each storage server is configured with two network interface adapters. One is a common network interface card (NIC) that connects to the service IP network. The server interacts with the client through this NIC. The other is an HBA that connects to the FC SAN. The server communicates with the storage devices on the FC SAN through this adapter.

3.1.4.3.2 Introduction to IP SAN

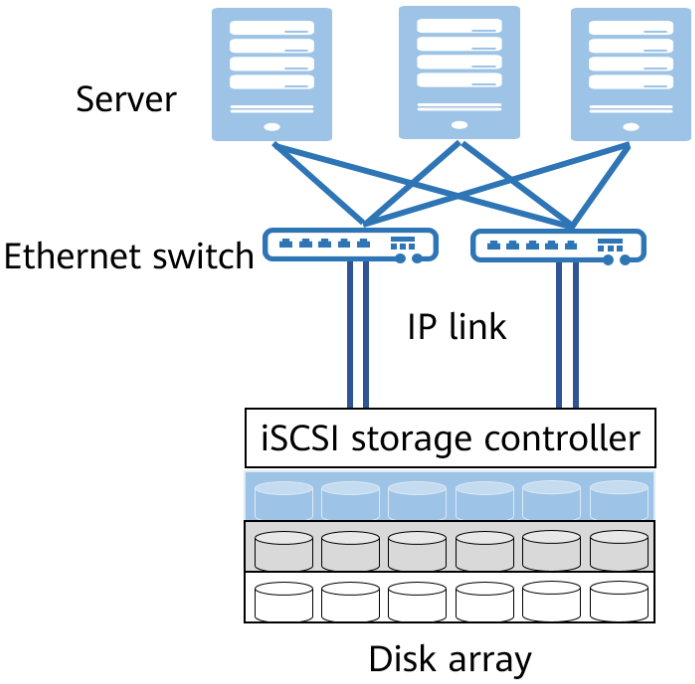


Figure 3-10 Architecture of IP SAN

IP SAN has become a popular network storage technology in recent years. The early SANs are all FC SANs, where data is transferred in the fibre channel in blocks. Due to the incompatibility between FC protocol and the IP protocol, customers who want to use FC SAN have to purchase its devices and components. As a result, a large number of small and medium-sized users may flinch at its high cost and complicated configuration. Therefore, FC SAN is mainly used for middle- and high-end storage that requires high performance, redundancy, and availability. To popularize SANs and leverage the advantages of SAN architecture, technicians consider to combine SANs with prevailing and affordable IP networks. Therefore, the IP SAN that uses the existing IP network architecture is introduced. IP SAN is a combination of the standard TCP/IP protocol with the SCSI instruction set and implements block-level data storage based on the IP network.

The difference between IP SAN and FC SAN lies in the transfer protocol and medium. Common IP SAN protocols include Internet SCSI (iSCSI), Fibre Channel over IP (FCIP), and Internet Fibre Channel Protocol (iFCP). iSCSI is the fastest growing protocol standard. In most cases, IP SAN refers to iSCSI-based SAN.

The iSCSI-based SAN uses an iSCSI initiator (server) and an iSCSI target (storage device) on the IP network to form a SAN.

3.1.4.3.3 Comparison Among Storage Networking Types

	DAS	NAS	SAN	
			FC SAN	IP SAN
Transmission mode	SCSI, Fibre Channel, and SAS	IP	Fibre Channel	IP
Data type	Block-level	File-level	Block-level	Block-level
Application scenario	Any	File servers	Database applications	Video security
Advantage	Easy to understand; robust compatibility	Easy to install; low cost	High scalability and performance; high availability	Strong scalability; low cost
Disadvantage	Difficult management; limited scalability; low storage space utilization	Low performance; inapplicable to some applications	Expensive and complex configuration; poor networking compatibility	Low performance

Figure 3-11 Comparison among storage networking types

Figure 3-11 describes the three storage networking types. SAN and NAS complement each other to provide access to different types of data.

3.1.5 Storage Types

3.1.5.1 Centralized Storage

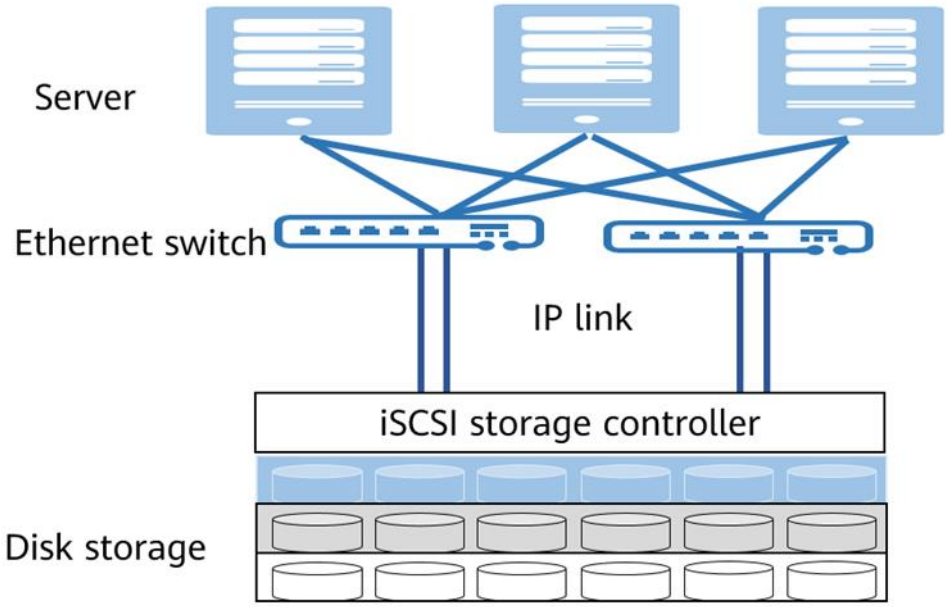


Figure 3-12 Architecture of centralized storage

By "centralized", it is meant that all resources are centrally deployed and are used to provide services over a unified interface. Centralized storage means that all physical disks are centrally deployed in the disk enclosure and are used to provide storage services externally through the controller. Centralized storage typically refers to disk arrays.

In terms of technical architectures, centralized storage is classified into SAN and NAS. SANs can be classified into Fibre Channel SAN (FC SAN), Internet Protocol SAN (IP SAN),

and Fibre Channel over Ethernet SAN (FCoE SAN). Currently, FC SAN and IP SAN technologies are mature, and FCoE SAN is still in the early stage of its development.

A disk array combines multiple physical disks into a single logical unit. Each disk array consists of one controller enclosure and multiple disk enclosures. This architecture delivers an intelligent storage space featuring high availability, high performance, and large capacity.

3.1.5.2 Distributed Storage

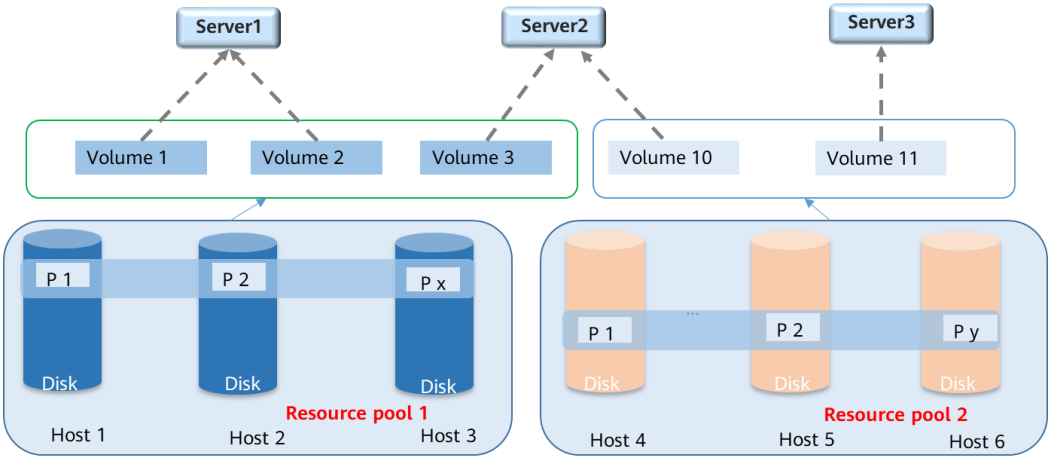


Figure 3-13 Architecture of distributed storage

Unlike centralized storage, distributed storage does not store data on one or more specific nodes. It virtualizes all available space distributed on each host of an enterprise to a virtual storage device. In this way, the data stored in this virtual storage device is also distributed over the storage network.

As shown in Figure 3-13, distributed storage uses general-purpose servers rather than storage devices. A distributed storage system does not have any controller enclosure or disk enclosure. All disk storage resources are delivered by general-purpose x86 servers. Clients are delivered by the distributed storage system to identify and manage disks, as well as establish data routing and process read/write I/Os.

The distributed storage client mode has advantages and disadvantages. In terms of capacity expansion, an x86 server with a client installed can be a part of the distributed storage system. This mode delivers great scalability. However, in addition to the applications running on the server, the client software installed on the server also consumes compute resources. When you plan a distributed storage system, you must reserve certain amounts of compute resources on servers you intend to add to this system. Therefore, this mode has certain requirements on the hardware resources of the server. In a traditional centralized storage system, data is read and written by controllers. However, the number of controllers is limited. In a distributed storage system, servers with clients can read and write data, breaking the limit on the number of controllers and improving the read and write speed to some extent. However, the paths for reading and writing data need to be calculated each time data is read and written. If there are too many clients, the path calculating is complicated. When the optimum performance is reached, adding more clients cannot further improve the performance.

For high data availability and security, the centralized storage system uses the RAID technology. RAID can be implemented by hardware and software. All disks must be deployed on the same server (hardware RAID requires a unified RAID card, and software RAID requires a unified OS). Disks of a distributed storage system are distributed in different servers, resulting in that the RAID mechanism is unavailable.

Therefore, in a distributed storage system, a copying mechanism is introduced to ensure high data reliability. The copying mechanism copies and stores data on different servers. If a server is faulty, data will not be lost.

3.1.5.3 Storage Service Types

3.1.5.3.1 Block Storage

Block storage commonly uses an architecture that connects storage devices and application servers over a network. This network is used only for data access between servers and storage devices. When there is an access request, data can be transmitted quickly between servers and backend storage devices as needed. From a client's perspective, block storage functions the same way as disks. One can format a disk with any file system and then mount it. A major difference between block storage and file storage is that block storage provides storage spaces only, leaving the rest of the work, such as file system formatting and management, to the client.

Block storage uses evenly sized blocks to store structured data. In block storage, data is stored without any metadata. This makes block storage useful when applications need to strictly control the data structure. A most common usage is for database. Databases can read and write structured data faster with raw block devices.

Currently, block storage is usually deployed in FC SAN and IP SAN based on the protocols and connectors used. FC SAN uses the Fibre Channel protocol to transmit data between servers (hosts) and storage devices, whereas, IP SAN uses the IP protocol for communication. The FC technology can meet the growing needs for high-speed data transfer between servers and large-capacity storage systems. With the FC protocol, data can be transferred faster with low protocol overheads, while maintaining certain network scalability.

File Storage has the following advantages:

- Offers long-distance data transfer with a high bandwidth and a low transmission bit error rate.
- Based on the SAN architecture and massive addressable devices, multiple servers can access a storage system over the storage network at the same time, eliminating the need for purchasing storage devices for every server. This reduces the heterogeneity of storage devices and improves storage resource utilization.
- Protocol-based data transmission can be handled by the HBA, occupying less CPU resources.

In a traditional block storage environment, data is transmitted over the fibre channel via block I/Os. To leverage the advantages of FC SAN, enterprises need to purchase additional FC components, such as HBAs and switches. Enterprises usually have an IP network-based architecture. As technologies evolve, block I/Os now can be transmitted over the IP network, which is called IP SAN. With IP SAN, legacy infrastructure can be reused, which is far more economical than investing in a brand new SAN environment. In

addition, many remote and disaster recovery solutions are also developed based on the IP network, allowing users to expand the physical scope of their storage infrastructure.

Internet SCSI (iSCSI), Fibre Channel over IP (FCIP), and Fibre Channel over Ethernet (FCoE) are the major IP SAN protocols.

- iSCSI encapsulates SCSI I/Os into IP packets and transmits them over TCP/IP. iSCSI is widely used to connect servers and storage devices because it is cost-effective and easy to implement, especially in environments without FC SAN.
- FCIP allows FCIP entities, such as FCIP gateways, to implement FC switching over IP networks. FCIP combines the advantages of FC SAN and the mature, widely-used IP infrastructure. This gives enterprises a better way to use existing investments and technologies for data protection, storage, and migration.
- FCoE achieves I/O consolidation. Usually, one server in a data center is equipped with two to four NICs and HBAs for redundancy. If there are hundreds of servers in a data center, numerous adapters, cables, and switches required make the environment complex and difficult to manage and expand. FCoE achieves I/O consolidation via FCoE switches and Converged Network Adapters (CNA). CNAs replace the NICs and HBAs on the servers and consolidate IP traffic and FC traffic. In this way, servers no longer need various network adapters and many independent networks, thus the requirement of NICs, cables, and switches is reduced. This massively lowers the costs and management overheads.

Block storage is a high-performance network storage, but data cannot be shared between hosts in block storage. Some enterprise workloads may require data or file sharing between different types of clients, and block storage cannot do this.

3.1.5.3.2 File Storage

File storage provides file-based, client-side access over the TCP/IP protocol. In file storage, data is transferred via file I/Os in the local area network (LAN). A file I/O is a high-level request for accessing a specific file. For example, a client can access a file by specifying the file name, location, or other attributes. The NAS system records the locations of files on disks and converts the client's file I/Os to block I/Os to obtain data.

File storage is a commonly used type of storage for desktop users. When you open and close a document on your computer, you used the file system. Clients can access file systems on the file storage for file upload and download. Protocols used for file sharing between clients and storage include CIFS (SMB) and NFS. In addition to file sharing, file storage also provides file management functions, such as reliability maintenance and file access control. Although there are differences in managing file storage and local files, file storage is basically a directory to users. One can use file storage almost the same as using local files.

Because NAS access requires the conversion of file system format, it is not suitable for applications using blocks, especially database applications that require raw devices.

File Storage has the following advantages:

- Comprehensive information access: Local directories and files can be accessed by users on other computers over LAN. Multiple end users can collaborate with each other based on same files, such as project documents and source code.
- Good flexibility: NAS is compatible with both Linux and Windows clients.

- Low cost: NAS uses common and low-cost Ethernet components.

3.1.5.3.3 Object Storage

Users who frequently access the Internet and use mobile devices often need object storage techniques. The core of object storage is to separate the data path from the control path. Object storage does not provide access to original blocks or files, but to the entire object data via system-specific APIs. You can access objects using HTTP/REST-based uniform resource locators (URLs), like you access websites using browsers. Object storage abstracts storage locations as URLs so that storage capacity can be expanded in a way that is independent of the underlying storage mechanism. This makes object storage an ideal way to build a large-scale system with high concurrency.

As the system grows, object storage can still provide a single namespace. This way, applications or users do not need to worry about which storage system they are using. By using object storage, you do not need to manage multiple storage volumes like using a file system. This greatly reduces O&M workloads.

Object storage has many advantages in processing unstructured data over traditional storage and delivers the advantages of both SAN and NAS. It is independent of platforms or locations, offering scalability, security, and data sharing:

It can distribute object requests to large-scale storage cluster servers. This enables an inexpensive, reliable, and scalable storage system for massive amounts of data. Other advantages of object storage are as follows:

- Security: data consistency and content authenticity. Object storage uses special algorithms to generate objects with strong encryption. Requests in object storage are verified in storage devices instead of using external verification mechanisms.
- Platform-independent design: Objects are abstract containers for data (including metadata and attributes). This allows objects to be shared between heterogeneous platforms, either locally or remotely, making object storage the best choice in cloud computing.
- Scalability: The flat address space used enables object storage to store a large amount of data without compromising performance. Both storage and OSD nodes can scale independently in terms of performance and capacity.

OSD intelligently manages and protects objects. Its protection and replication capabilities can be self-healed, enabling data redundancy at a low cost. If one or more nodes in a distributed object storage system fail, data can still be accessed. In such cases, three data nodes concurrently transfer data, making the transfer fast. As the number of data node servers increase, read and write speed up accordingly. In this way, performance is improved.

3.1.5.3.4 Summary of Block Storage, File Storage, and Object Storage

In block storage, file systems reside on top of application servers, and applications directly access blocks. The FC protocol is usually used for data transfer, and it has a higher transmission efficiency than the TCP/IP protocol used in file storage. The header of each protocol data unit (PDU) in TCP/IP is twice larger than the header of a data frame in FC. In addition, the maximum length of an FC data frame is larger than that in Ethernet. But data cannot be shared between hosts in block storage. Some enterprise workloads may require data or file sharing between different types of clients, and block

storage cannot do this. In addition, block storage is complex and costly because additional components, such FC components and HBAs, need to be purchased.

File systems are deployed on file storage devices, and users access specific files, for example, opening, reading from, writing to, or closing a file. File storage maps file operations to disk operations, and users do not need to know the exact disk block where the file resides. Data is exchanged between users and file storage over the Ethernet in a LAN. File storage is easy to manage and supports comprehensive information access. One can share files by simply connecting the file storage devices to a LAN. This makes file sharing and collaboration more efficient. But file storage is not suitable for applications that demand block devices, especially databases systems. This is because file storage requires the conversion of file system format and users access specific files instead of data.

Object storage uses a content addressing system to simplify storage management, ensuring that the stored content is unique. It offers terabyte to petabyte scalability for static data. When a data object is stored, the system converts the binary content of the stored data to a unique identifier. The content address is not a simple mapping of the directory, file name, or data type of the stored data. OBS ensures content reliability with globally unique, location-independent identifiers and high scalability. It is good at storing non-transactional data, especially static data and is applicable to archives, backups, massive file sharing, scientific and research data, and digital media.

3.2 Key Storage Technologies

3.2.1 RAID Technology

3.2.1.1 What Is RAID

Redundant Array of Independent Disks (RAID) combines multiple physical disks into one logical disk in different ways, improving read/write performance and data security. With the development of RAID technology, RAID can be divided as seven basic levels (RAID 0 to RAID 6). In addition, there are some combinations of basic RAID levels, such as RAID 10 (combination of RAID 1 with RAID 0) and RAID 50 (combination of RAID 5 with RAID 0). Different RAID levels represent different storage performance, data security, and storage costs.

3.2.1.2 RAID Data Organization Forms

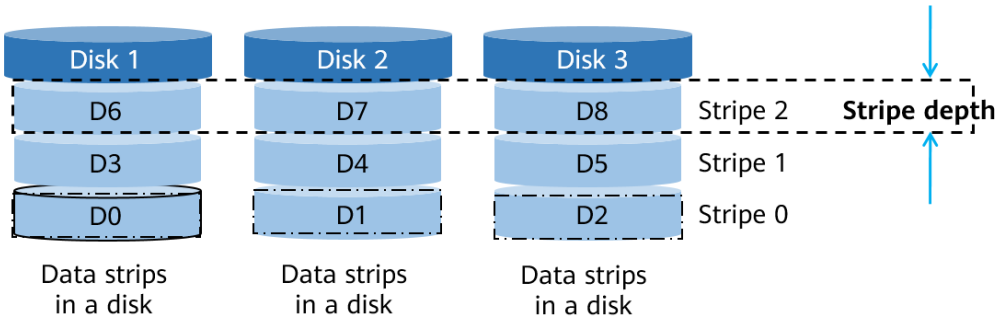


Figure 3-14 Data organization forms of RAID

RAID divides space in each disk into multiple strips of a specific size. Written data is also divided into blocks based on the strip size. The following concepts are involved:

Strip: A strip consists of one or more consecutive sectors in a disk, and multiple strips form a stripe.

Stripe: A stripe consists of strips of the same location or ID on multiple disks in the same array.

Stripe width indicates the number of disks used in an array for striping. For example, if a disk array consists of three member disks, the stripe width is 3.

Stripe depth indicates the capacity of a strip.

3.2.1.3 RAID Data Protection Techniques

RAID generally protects data by the following methods:

- Mirroring: Data copies are stored on another redundant disk, improving reliability and read performance.
- Parity check algorithm (XOR): Parity data is additional information calculated using user data. For a RAID array that uses parity, an additional parity disk is required. The XOR (symbol: ⊕) algorithm is used for parity.

XOR is widely used in digital electronics and computer science. XOR is a logical operation that outputs true only when inputs differ (one is true, the other is false).

- 0 ⊕ 0 = 0, 0 ⊕ 1 = 1, 1 ⊕ 0= 1, 1 ⊕ 1 = 0

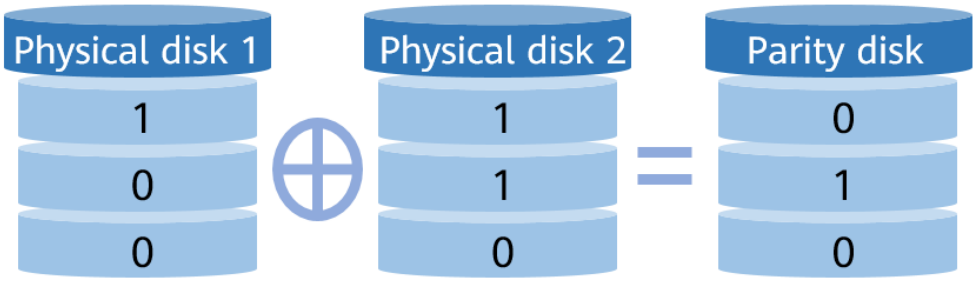


Figure 3-15 XOR check

3.2.1.4 RAID Hot Spare and Reconstruction

If a disk in a RAID array fails, a hot spare is used to automatically replace the failed disk to maintain the RAID array's redundancy and data continuity.

Hot spare is classified into the following types:

- Global: The spare disk is shared by all RAID groups in the system.
- Dedicated: The spare disk is used only by a specific RAID group in the system.

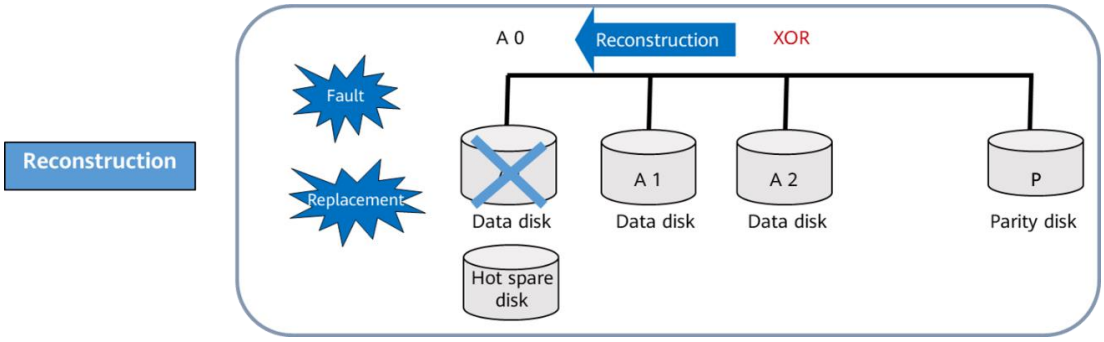


Figure 3-16 Hot spare and reconstruction

Data reconstruction: indicates a process of reconstructing data from a failed data disk to the hot spare disk. Generally, the data parity mechanism in RAID is used to reconstruct data.

Data parity: Redundant data is used to detect and rectify data errors. The redundant data is usually calculated through Hamming check or XOR operations. Data parity can greatly improve the reliability, performance, and error tolerance of the drive arrays. However, the system needs to read data from multiple locations, calculate, and compare data during the parity process, which affects system performance.

Generally, RAID cannot be used as an alternative to data backup. It cannot prevent data loss caused by non-drive faults, such as viruses, man-made damages, and accidental deletion. Data loss here refers to the loss of operating system, file system, volume manager, or application system data, not the RAID data loss. Therefore, data protection measures, such as data backup and disaster recovery, are necessary. They are complementary to RAID, and can ensure data security and prevent data loss at different layers.

3.2.1.5 Common RAID Levels

3.2.1.5.1 RAID 0

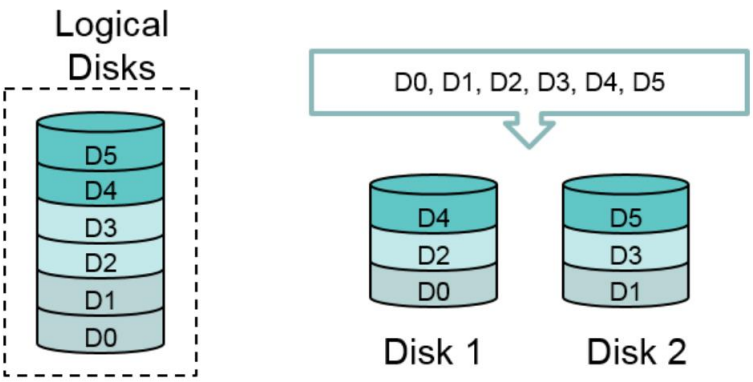


Figure 3-17 RAID 0 diagram

RAID 0 is a simple data striping technology without parity. In essence, RAID 0 is not a real RAID, because it offers no redundancy. In RAID 0, disks are striped to form a large-capacity storage space (as shown in Figure 3-17), data is distributed across all disks, and reading data from multiple disks can be processed concurrently. RAID 0 allows I/O operations to be performed concurrently, improving utilization of the bus bandwidth. In addition, RAID 0 requires no data parity, thereby providing the highest performance. If a

RAID 0 group consists of *n* disks, theoretically, the read and write performance of the group is *n* times that of a single disk. Due to the bus bandwidth restriction and other factors, the actual performance is lower than the theoretical one.

RAID 0 features low cost, high read/write performance, and 100% disk usage. However, it offers no redundancy. In the event of a disk failure, data is lost. Therefore, RAID 0 is applicable to applications that have high requirements on performance but low requirements on data security and reliability, such as video/audio storage and temporary storage space.

3.2.1.5.2 RAID 1

RAID 1, also known as mirror or mirroring, is designed to maximize the availability and repairability of user data. RAID 1 automatically copies all data written to one disk to the other disk in a RAID group.

RAID 1 writes the same data to the mirror disk while storing the data on the source disk. If the source disk fails, the mirror disk takes over services from the source disk. RAID 1 delivers the best data security among all RAID levels because the mirror disk is used for data backup. However, no matter how many disks are used, the available storage space is only the capacity of a single disk. Therefore, RAID 1 delivers the lowest disk usage among all RAID levels.

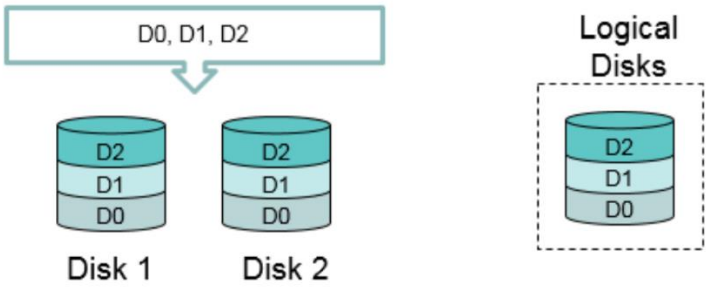


Figure 3-18 RAID 1 diagram

Figure 3-18 shows the diagram of RAID 1. There are two disks, Disk 1 and Disk 2. RAID 1 stores the data (D1, D2...) in the primary disk (Disk 1), and then stores the data again in Disk 2 for data backup.

RAID 1 is the highest in unit storage cost among all RAID levels. However, it delivers the highest data security and availability. RAID 1 is applicable to online transaction processing (OLTP) applications with intensive read operations and other applications that require high read/write performance and reliability, for example, email, operating system, application file, and random access environment.

3.2.1.5.3 RAID 3

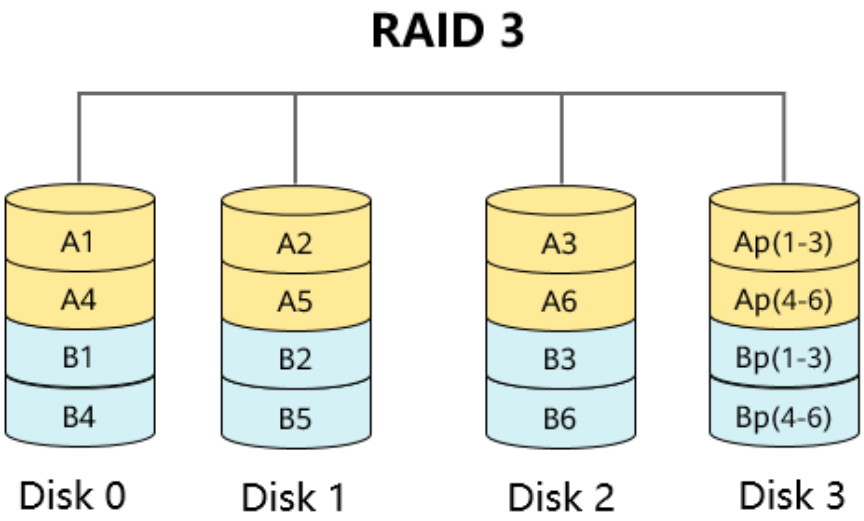


Figure 3-19 RAID 3 diagram

RAID 3 is a parallel access array that uses one disk as the parity disk and other disks as data disks. Data is stored to each data disk by bit or byte. RAID 3 requires at least three disks. XOR check is performed for data in the same stripe on different disks, and the parity data is written into the parity disk. The read performance of a complete RAID 3 group is the same as that of a RAID 0 group. Data is concurrently read from multiple disk strips, providing high performance and data fault tolerance. In RAID 3 level, when data is written, the system must read all data blocks in the same stripe to calculate a check value and write the new value to the parity disk. The write operation involves four operations: writing a data block, reading data blocks in the same stripe, calculating a check value, and writing the check value. As a result, the system overhead is high and the performance decreases.

If a disk in RAID 3 is faulty, data reading is not affected. The system reconstructs the data based on the parity data and other intact data. If the data block to be read is located on the faulty disk, the system reads all data blocks in the same strip and reconstructs the lost data based on the parity value. As a result, the system performance decreases. After the faulty disk is replaced, the system reconstructs the data on the faulty disk to the new disk in the same way.

RAID 3 requires only one parity disk. The disk usage is high. In addition, concurrent access delivers high performance for a large number of read and write operations with high bandwidth. RAID 3 applies to applications that require sequential access to large amounts of data, such as image processing and streaming media services. Currently, the RAID 5 algorithm is continuously improved to simulate RAID 3 when a large amount of data is read. In addition, the performance of RAID 3 deteriorates greatly when a disk is faulty. Therefore, RAID 5 is often used to replace RAID 3 to run applications that feature continuous, high bandwidth, and a large number of read and write operations.

3.2.1.5.4 RAID 5

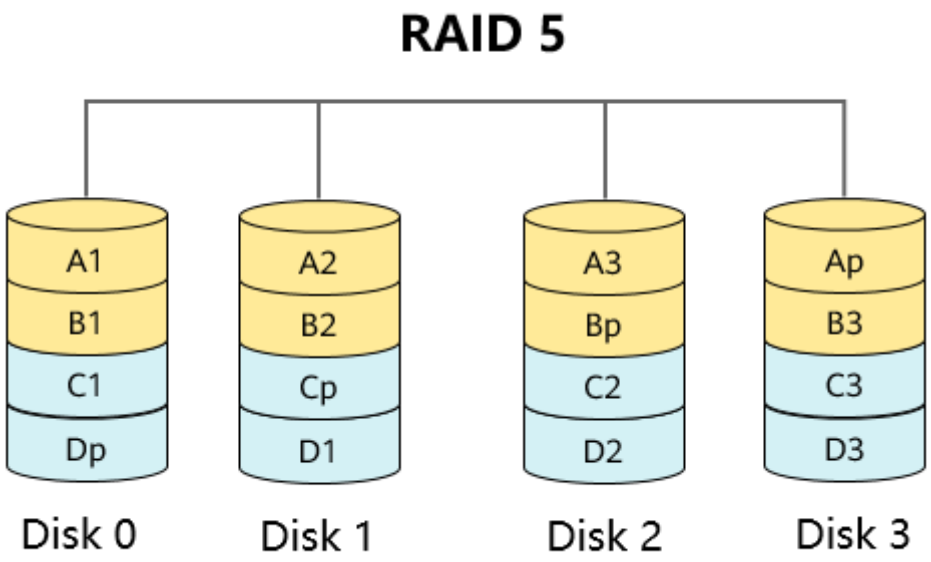


Figure 3-20 RAID 5 diagram

RAID 5 is a compromise between RAID 0 and RAID 1. RAID 5 offers slower write speeds due to the parity check information but could offer the same read performance as RAID 0. In addition, RAID 5 offers higher disk usage and lower storage costs than RAID 1 because multiple data records of RAID 5 share the same parity check information. It is widely used at present.

In a RAID 5 group, data and associated parity check information are stored on the member disks. To be specific, the capacity of N - 1 disks is used to store the data, and the capacity of one disk is used to store the parity check information (N indicates the number of disks). Therefore, if a disk in RAID 5 is damaged, data integrity is not affected, ensuring data security. After a damaged disk is replaced, RAID 5 automatically reconstructs data on the faulty disk based on the parity check information, ensuring high reliability.

The available capacities of all the disks in a RAID 5 group must be the same. If not, the available capacity depends on the smallest one. It is recommended that the rotational speeds of the disks be the same. Otherwise, the performance is affected. In addition, the available space is equal to the space of N – 1 disks. RAID 5 has no independent parity disk, so the parity information is distributed across all disks, occupying the capacity of one disk.

In RAID 5, disks stores data and parity data. Data blocks and associated check information are stored on different disks. If one data disk is faulty, the system reconstructs data on the faulty disk based on data blocks and associated check information on other disks in the same strip. Like other RAID levels, the performance of RAID 5 is greatly affected during data reconstruction.

RAID 5 is a storage protection solution that balances storage performance, data security, and storage cost. It can be considered as a compromise between RAID 0 and RAID 1. RAID 5 can meet most storage application requirements. Most data centers adopt RAID 5 as the protection solution for application data.

3.2.1.5.5 RAID 6

RAID 6 breaks through the limitation of disk redundancy.

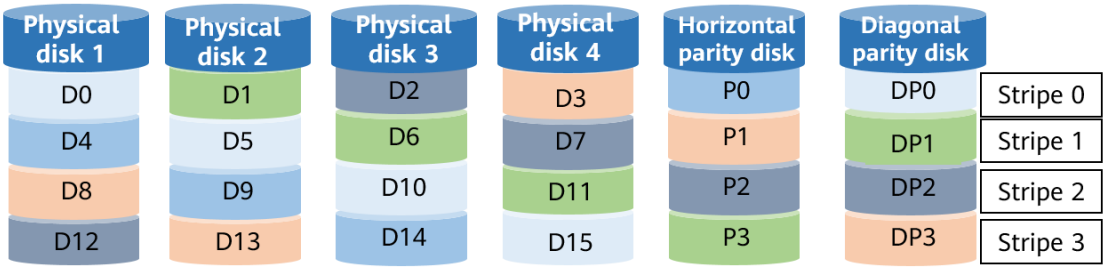


Figure 3-21 RAID 6 DP diagram

In the past, there was a low probability that two disks were faulty at the same time. However, due to increase in capacity and density of FC and SATA disks, RAID 5 reconstruction needs longer time, the risk that two disks are faulty at the same time also increases greatly. Enterprise-level storage must attach great importance to this risk. Therefore, RAID 6 is introduced.

The RAID levels described in the previous sections only protect data loss caused by the failure of a single disk. If two disks are faulty at the same time, data cannot be restored. As shown in Figure 3-21, RAID 6 adopts double parity to prevent data loss in the event of simultaneous failure of two disks, ensuring service continuity. RAID 6 is designed based on RAID 5 to further enhance the data security. It is actually an extended RAID 5 level.

RAID 6 must support the recovery of both actual data and parity data and the RAID controller design is more complicated. As a result, RAID 6 is more expensive than other RAID levels. In most cases, RAID 6 can be implemented by using two independent parity columns. Parity data can be stored on two different parity disks or distributed across all member disks. If two disks fail at the same time, the data on the two disks can be reconstructed by solving the equation with two unknowns.

Alternatively, RAID 6 can be implemented by using double parity (DP).

- RAID 6 DP also has two independent parity data blocks. Parity values in the horizontal parity disk are also called parity check values, which are obtained by performing the XOR operation on user data in the same stripe. As shown in Figure 3-21, P0 is obtained by performing an XOR operation on D0, D1, D2, and D3 in stripe 0, and P1 is obtained by performing an XOR operation on D4, D5, D6, and D7 in stripe 1. Therefore, P0 = D0 ⊕ D1 ⊕ D2 ⊕ D3, P1 = D4 ⊕ D5 ⊕ D6 ⊕ D7, and so on.
- The diagonal parity uses the diagonal XOR operation to obtain the row-diagonal parity data block. A process of selecting a data block is relatively complex. DP0 is obtained by performing an exclusive OR operation on D0 on a stripe 0 of a hard disk 1, D5 on a stripe 1 of a hard disk 2, D10 on a stripe 2 of a hard disk 3, and D15 on a stripe 3 of a hard disk 4. DP1 is obtained by performing an exclusive OR operation on D1 on a stripe 0 of a hard disk 2, D6 on a stripe 1 of a hard disk 3, D11 on a stripe 2 of a hard disk 4, and P3 on a stripe 3 of a first parity hard disk. DP2 is obtained by performing an exclusive OR operation on D2 on a stripe 0 of a hard disk 3, D7 on a stripe 1 of a hard disk 4, P2 on a stripe 2 of an odd even hard disk, and D12 on a stripe 3 of a hard disk 1. Therefore, DP0 = D0 ⊕ D5 ⊕ D10 ⊕ D15, DP1 = D1 ⊕ D6 ⊕ D11 ⊕ P3, and so on.

RAID 6 features fast read performance and high fault tolerance. However, the cost of RAID 6 is much higher than that of RAID 5, the write performance is poor, and the design and implementation are complicated. Therefore, RAID 6 is seldom used and is mainly applicable to scenarios that require high data security. It can be used as an economical alternative to RAID 10.

3.2.1.6 Introduction to RAID 2.0

- RAID 2.0

RAID 2.0 is an enhanced RAID technology that effectively resolves the following problems: prolonged reconstruction of an HDD, and data loss if a disk is faulty during the long reconstruction of a traditional RAID group.

- RAID 2.0+

RAID 2.0+ provides smaller resource granularities (tens of KB) than RAID 2.0 to serve as the units of standard allocation and reclamation of storage resources, similar to VMs in computing virtualization. This technology is called virtual block technology.

- **Huawei RAID 2.0+**

Huawei RAID 2.0+ is a brand-new RAID technology developed by Huawei to overcome the disadvantages of traditional RAID and keep in line with the storage architecture virtualization trend. RAID 2.0+ implements two-layer virtualized management instead of the traditional fixed management. Based on the underlying disk management that employs block virtualization (Virtual for Disk), RAID 2.0+ uses Smart-series efficiency improvement software to implement efficient resource management that features upper-layer virtualization (Virtual for Pool). Block virtualization is to divide disks into multiple contiguous storage spaces of a fixed size called a chunk (CK).

3.2.1.7 RAID 2.0+ Block Virtualization

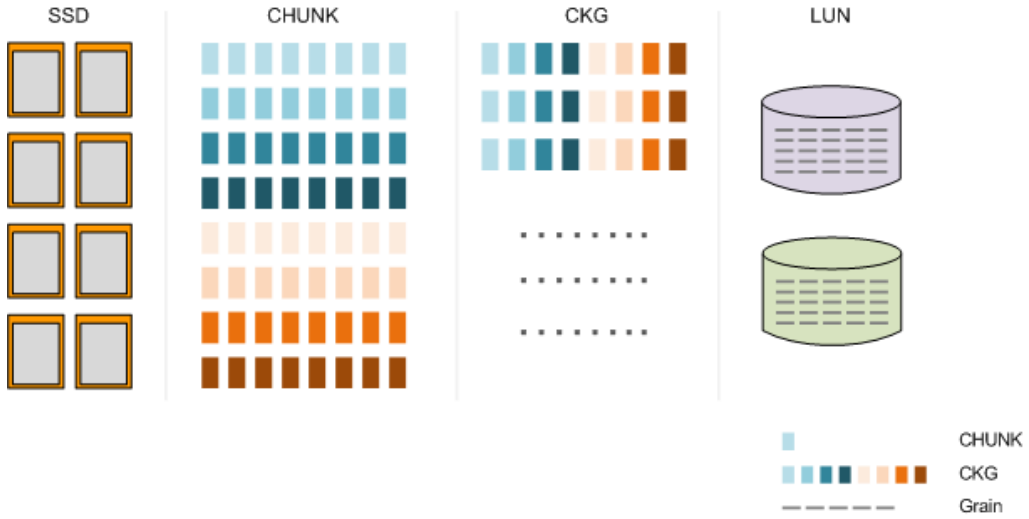


Figure 3-22 Working principles of RAID 2.0+ block virtualization

- The working principles of RAID 2.0+ block virtualization are as follows:

1. Multiple SSDs form a storage pool.

2. Each SSD is then divided into CKs of a fixed size (typically 4 MB) for logical space management.
3. CKs from different SSDs form chunk groups (CKGs) based on the RAID policy specified on DeviceManager.
4. CKGs are further divided into grains (typically 8 KB). Grains are mapped to LUNs for refined management of storage resources.

- RAID 2.0+ outperforms traditional RAID in the following aspects:
  - Service load balancing to avoid hot spots: Data is evenly distributed to all disks in the resource pool, protecting disks from early end of service lives due to excessive writes.
  - Fast reconstruction to reduce risk window: When a disk fails, the valid data in the faulty disk is reconstructed to all other functioning disks in the resource pool (fast many-to-many reconstruction), efficiently resuming redundancy protection.
  - Reconstruction load balancing among all disks: All member disks in a storage resource pool participate in reconstruction, and each disk only needs to reconstruct a small amount of data. Therefore, the reconstruction process does not affect upper-layer applications.

3.2.2 Storage Protocol

3.2.2.1 SCSI Protocol

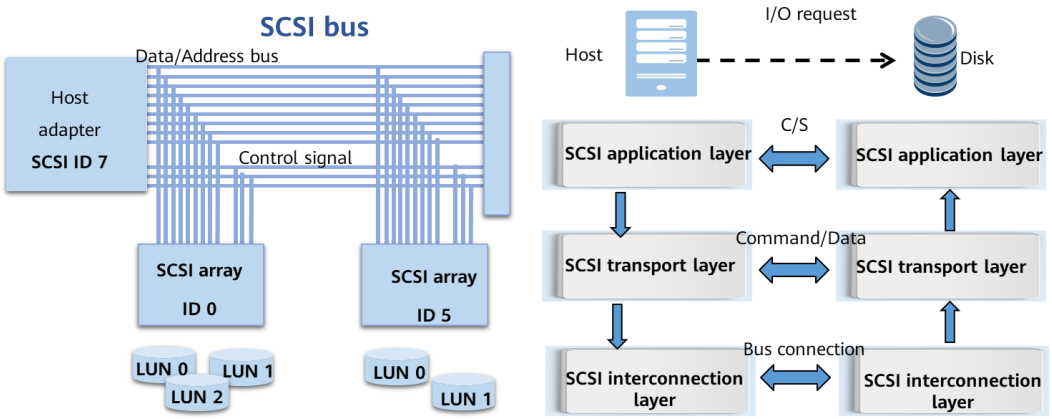


Figure 3-23 SCSI protocol

Computers communicate with storage systems through buses. The bus is a path through which data is transferred from the source device to the target device. To put it simple, the high-speed cache of the controller functions as the source device and transfers data to target disks, which serve as the target devices. The controller sends a signal to the bus processor requesting to use the bus. After the request is accepted, the controller's high-speed cache sends data. During this process, the bus is occupied by the controller and other devices connected to the same bus cannot use it. However, the bus processor can interrupt the data transfer at any time and allow other devices to use the bus for operations of a higher priority.

A computer has numerous buses, which are like high-speed channels used for transferring information and power from one place to another. For example, the

universal serial bus (USB) port is used to connect an MP3 player or digital camera to a computer. The USB port is competent to the data transfer and charging of portable electronic devices that store pictures and music. However, the USB bus is incapable of supporting computers, servers, and many other devices.

In this case, SCSI buses are applicable. SCSI, short for Small Computer System Interface, is an interface used to connect between hosts and peripheral devices including disk drives, tape drives, CD-ROM drives, and scanners. Data operations are implemented by SCSI controllers. Like a small CPU, the SCSI controller has its own command set and cache. The special SCSI bus architecture can dynamically allocate resources to tasks run by multiple devices in a computer. In this way, multiple tasks can be processed at the same time.

SCSI is a vast protocol system evolved from SCSI-1 to SCSI-2 and then to SCSI-3. It defines a model and a necessary command set for different devices (such as disks, processors, and network devices) to exchange information using the framework.

3.2.2.2 iSCSI Protocol

iSCSI encapsulates SCSI commands and block data into TCP packets and transmits the packets over an IP network. iSCSI uses mature IP network technologies to implement and extend SANs.

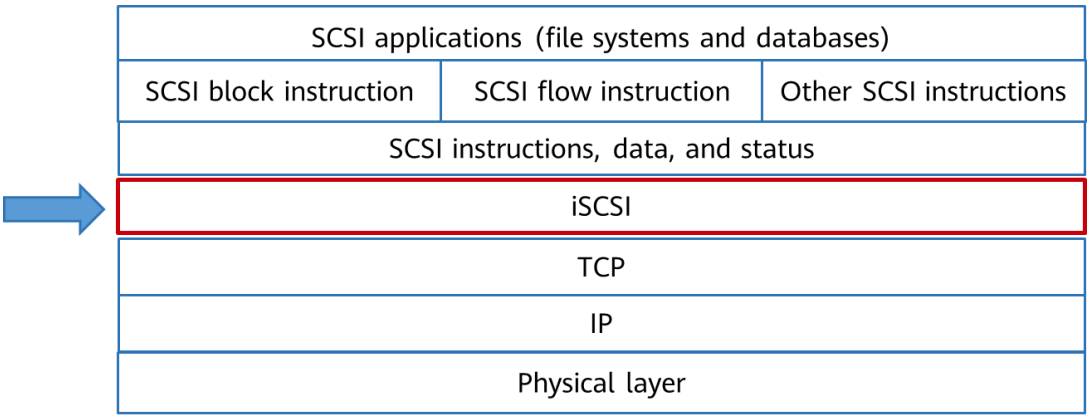


Figure 3-24 iSCSI protocol

The SCSI controller card is used to connect to multiple devices to form a network, but the devices can communicate with each other on the network and cannot be shared on the Ethernet. If devices form a network through SCSI and the network can be mounted to an Ethernet, the devices can interconnect and share with other devices as network nodes. As a result, the iSCSI protocol evolved from SCSI. The IP SAN using iSCSI converts user requests into SCSI codes and encapsulates data into IP packets for transmission over the Ethernet.

The iSCSI scheme was initiated by Cisco and IBM and then advocated by Adaptec, Cisco, HP, IBM, Quantum, and other companies. iSCSI offers a way of transferring data through TCP and saving data on SCSI devices. The iSCSI standard was drafted in 2001 and submitted to IETF in 2002 after numerous arguments and modifications. In Feb. 2003, the iSCSI standard was officially released. The iSCSI technology inherits advantages of traditional technologies and develops based on them. On one hand, SCSI technology is a storage standard widely applied by storage devices including disks and tapes. It has been keeping a fast development pace since 1986. On the other hand, TCP/IP is the most

universal network protocol and IP network infrastructure is mature. The two points provide a solid foundation for iSCSI development.

Prevalent IP networks allow data to be transferred over LANs, WANs, or the Internet using new IP storage protocols. The iSCSI protocol is developed by this philosophy. iSCSI adopts IP technical standards and converges SCSI and TCP/IP protocols. Ethernet users can conveniently transfer and manage data with a small investment.

3.2.2.2.1 iSCSI Initiator and Target

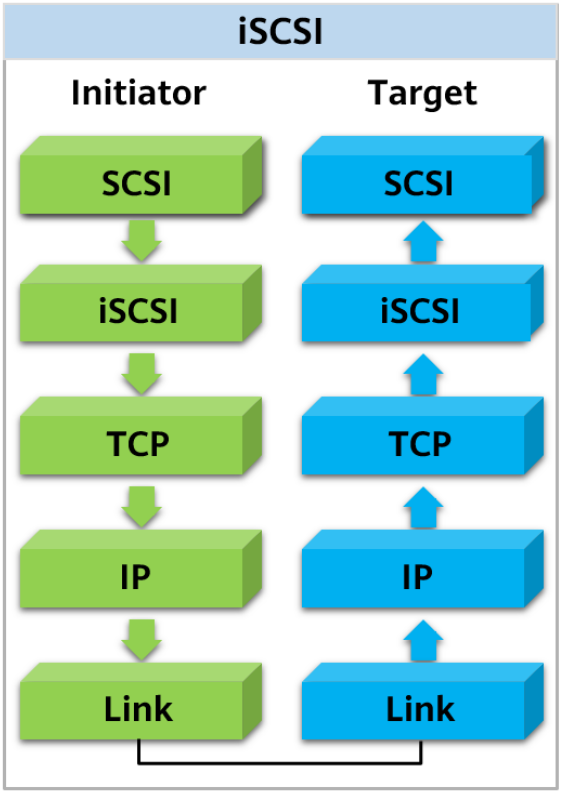


Figure 3-25 iSCSI initiator and target

The iSCSI communication system inherits some of SCSI's features. The iSCSI communication involves an initiator that sends I/O requests and a target that responds to the I/O requests and executes I/O operations. After a connection is set up between the initiator and target, the target controls the entire process as the primary device.

- There are three types of iSCSI initiators: software-based initiator driver, hardware-based TCP offload engine (TOE) NIC, and iSCSI HBA. Their performance increases in that order.
- iSCSI targets include iSCSI disk arrays and iSCSI tape libraries.

The iSCSI protocol defines a set of naming and addressing methods for iSCSI initiators and targets. All iSCSI nodes are identified by their iSCSI names. This method distinguishes iSCSI names from host names.

iSCSI uses iSCSI names to identify initiators and targets. Addresses change with the relocation of initiator or target devices, but their names remain unchanged. When setting up a connection, an initiator sends a request. After the target receives the request, it checks whether the iSCSI name contained in the request is consistent with that bound

with the target. If the iSCSI names are consistent, the connection is set up. Each iSCSI node has a unique iSCSI name. One iSCSI name can be used in the connections from one initiator to multiple targets. Multiple iSCSI names can be used in the connections from one target to multiple initiators.

The functions of the iSCSI initiator and target are as follows:

- Initiator

The SCSI layer generates command descriptor blocks (CDBs) and transfers them to the iSCSI layer.

The iSCSI layer generates iSCSI protocol data units (PDUs) and sends them to the target over an IP network.

- Target

The iSCSI layer receives PDUs and sends CDBs to the SCSI layer.

The SCSI layer interprets CDBs and gives responses when necessary.

3.2.2.3 Convergence of Fibre Channel and TCP

Ethernet technologies and Fibre Channel technologies are both developing fast. Therefore, it is inevitable that IP SAN and FC SAN that are complementary coexist for a long time.

The following protocols use a TCP/IP network to carry FC channels:

- Internet Fibre Channel Protocol (iFCP) is a gateway-to-gateway protocol that provides Fibre Channel communication services for optical devices on TCP/IP networks. iFCP delivers congestion control, error detection, and recovery functions through TCP. The purpose of iFCP is to enable current Fibre Channel devices to interconnect and network at the line rate over an IP network. The frame address conversion method defined in this protocol allows Fibre Channel storage devices to be added to the IP-based network through transparent gateways.
- Fibre Channel over Ethernet (FCoE) transmits Fibre Channel signals over an Ethernet, so that Fibre Channel data can be transmitted at the backbone layer of a 10 Gbit/s Ethernet using the Fibre Channel protocol.

3.2.2.3.1 iFCP Protocol

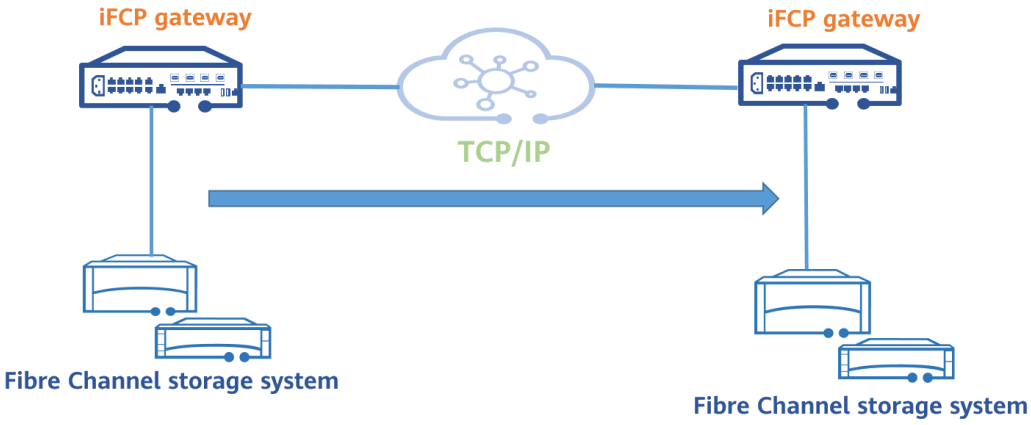


Figure 3-26 iFCP protocol

iFCP is a gateway-to-gateway protocol that provides Fiber Channel communication services for Fibre Channel devices on an TCP/IP network to implement end-to-end IP connection. Fibre Channel storage devices, HBAs, and switches can directly connect to iFCP gateways. iFCP provides traffic control, error detection, and error recovery through TCP. It enables Fibre Channel devices to interconnect and network at the line rate over an IP network.

The frame address conversion method defined in the iFCP protocol allows Fibre Channel storage devices to be added to the TCP/IP-based network through transparent gateways. iFCP can replace Fibre Channel to connect to and group Fibre Channel devices using iFCP devices. However, iFCP does not support the merge of independent SANs, and therefore a logical SAN cannot be formed. iFCP outstands in supporting SAN interconnection as well as gateway zoning, allowing fault isolation and breaking the limitations of point-to-point tunnels. In addition, it enables end-to-end connection between Fibre Channel devices. As a result, the interruption of TCP connection affects only a communication pair. SANs that adopt iFCP support fault isolation and security management, and deliver higher reliability than SANs that adopt FCIP.

3.2.2.3.2 iFCP Protocol Stack

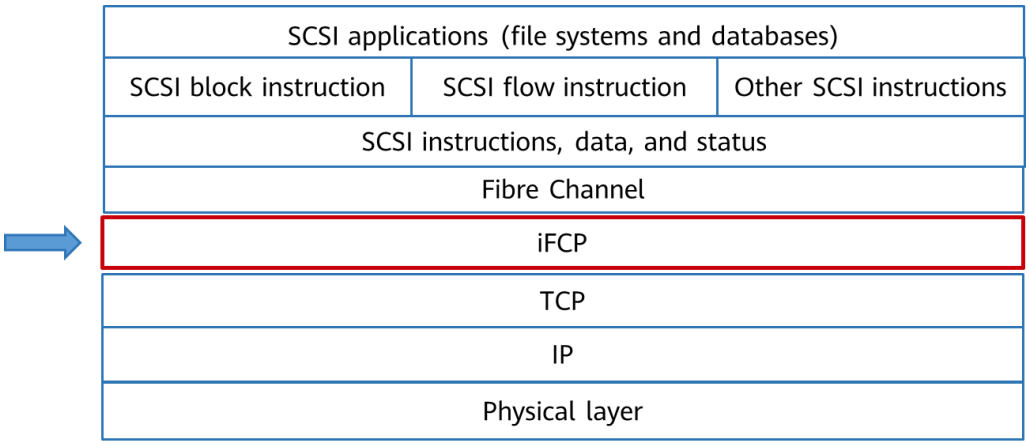


Figure 3-27 IFCP protocol stack

Fibre Channel only allows data to be transmitted locally, while iFCP enables data to be transmitted over an IP network and remotely transmitted across WANs through routers by encapsulating IP headers. In this way, enterprise users can use the existing storage devices and network architecture to share storage resources with more applications, breaking the geographical limitations of the traditional DAS and SAN architectures without changing the existing storage protocols.

3.2.2.3.3 FCoE Protocol

Fibre Channel over Ethernet (FCoE) allows the transmission of LAN and FC SAN data on the same Ethernet link. This reduces the number of devices, cables, and network nodes in a data center, as well as power consumption and cooling loads, simplifying management.

FCoE encapsulates FC data frames in Ethernet frames and allows service traffic on a LAN and SAN to be transmitted over the same Ethernet.

From the perspective of Fibre Channel, FCoE enables Fibre Channel to be carried by the Ethernet Layer 2 link. From the perspective of the Ethernet, FCoE is an upper-layer protocol that the Ethernet carries, like IP or IPX.

3.2.2.3.4 FCoE Protocol Encapsulation

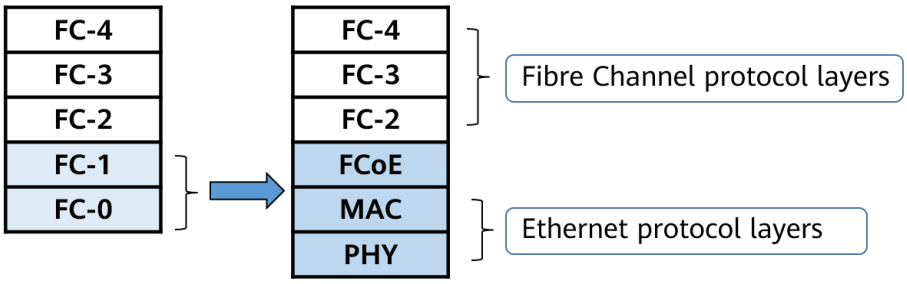


Figure 3-28 FCoE protocol encapsulation

The Fibre Channel protocol stack has five layers. FC-0 defines the medium type, FC-1 defines the frame coding and decoding mode, FC-2 defines the frame division protocol and flow control mechanism, FC-3 defines general services, and FC-4 defines the mapping from upper-layer protocols to Fibre Channel. FCoE encapsulates contents in the FC-2 and above layers into Ethernet packets for transmission.

3.3 Quiz

What are the relationships between DAS, NAS, SAN, block storage, file storage, and object storage?

4 Network Technology Basics

Network technologies are the basis for the interconnection of all platforms and services. What exactly is a network? What are the basic principles of network communication? And what are the common network technologies? This course will answer these questions and more.

4.1 IP Address Basics

4.1.1 What Is an IP Address?

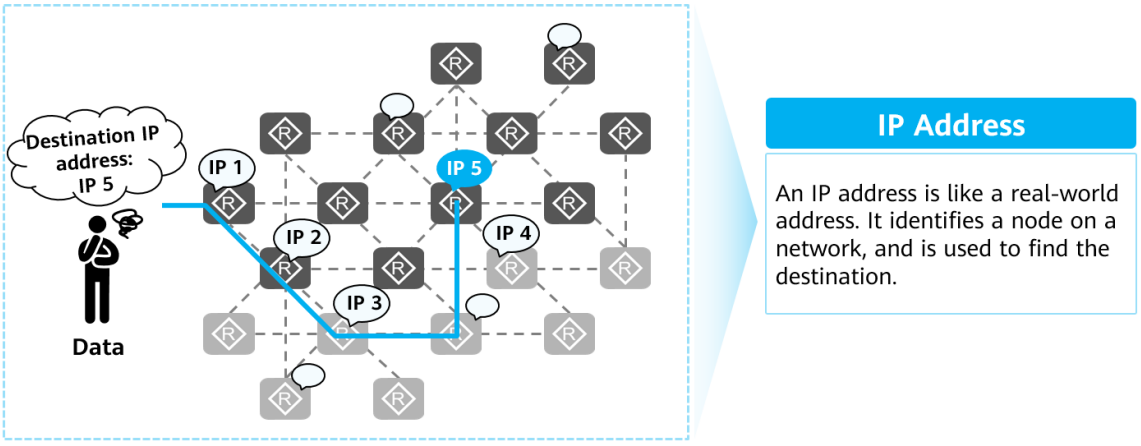


Figure 4-1 IP address

An IP address is a unique logical address used to identify a device that sends or receives data packets on a network.

On an IP network, to connect a PC to the Internet, you need to apply an IP address for the PC. An IP address is like a real-world address. It identifies a node on a network, and is used to find the destination. Global network communication is based on IP addresses.

An IP address is an attribute of an interface on a network device, not an attribute of the network device itself. To assign an IP address to a device is to assign an IP address to an interface of the device actually. If a device has multiple interfaces, each interface requires at least one IP address. (An interface that requires an IP address is usually the interface on a router or a computer.)

4.1.2 IP Address Format

- IP address format:

An IP address has 32 bits and consists of four bytes. For the convenience of reading and writing, an IP address is usually in the format of dotted decimal notation.

- Dotted decimal notation:

This type of IP address format is commonly used because it is easy to understand. However, a communication device uses binary digits to calculate the IP address. Therefore, it is necessary to master the conversion between decimal and binary digits.

- IPv4 address range:

0.0.0.0–255.255.255.255

4.1.3 IP Address Structure

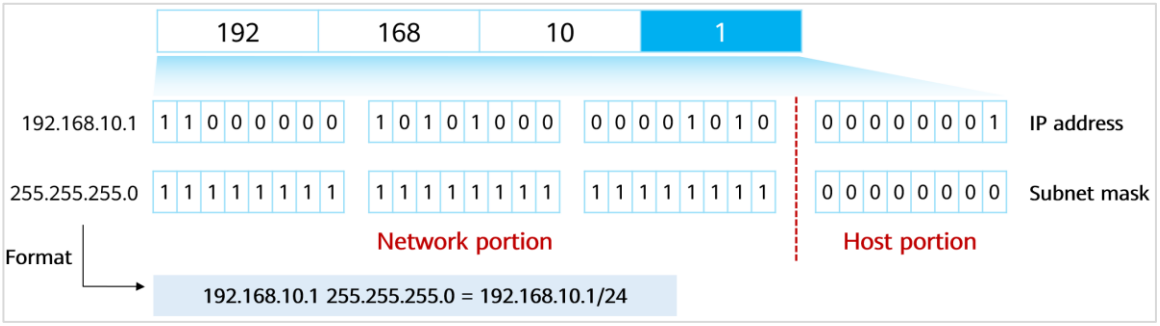


Figure 4-2 IP address structure

As shown in Figure 4-2, an IPv4 address consists of two parts:

1. Network portion: identifies a network segment.

- IP addresses do not show any geographical information. The network bits indicate the segment to which an IP address belongs.
- Network devices with same network bits are located on the same network, regardless of their physical locations.

2. Host portion: uniquely identifies a host on a network segment.

An IP address should be used together with a subnet mask. A subnet mask is also called a netmask.

- Same as an IP address, a subnet mask consists of 32 bits, and is also displayed in dotted decimal notation generally.
- A subnet mask is not an IP address. A subnet mask written in the binary format consists of consecutive 1s and 0s.
- Generally, the number of 1s in a subnet mask is the length of the subnet mask. For example, the length of the subnet mask 0.0.0.0 is 0, and that of 252.0.0.0 is 6.
- How to identify the network and host bits in an IP address: In a subnet mask, bits with the value of 1 correspond to the network bits in an IP address, while bits with the value of 0 correspond to the host bits. In other words, the number of 1s in a subnet mask equals to the number of network bits in an IP address, while the number of 0s equals to the number of host bits.

4.1.4 IP Address Classes (Classified Addressing)

Figure 4-3 IP address classes

IP addresses are classified into five classes to facilitate IP address management and networking:

- The easiest way to determine the class of an IP address is to check the first bits in its network bits. The class fields of class A, class B, class C, class D, and class E are binary numbers 0, 10, 110, 1110, and 1111, respectively.
- Class A, B, and C addresses are unicast IP addresses (except some special addresses). Only these three types of addresses can be assigned to hosts.
- Class D addresses are multicast IP addresses.
- Class E addresses are used for special experimental purposes.

Note: This section focuses only on class A, B, and C addresses.

Comparison between class A, B, and C addresses:

- Networks using class A addresses are called class A networks. Networks using class B addresses are called class B networks. Networks using class C addresses are called class C networks.
- The number of network bits of a class A network is 8. The number of network bits is small, so the number of addresses that can be assigned to the hosts is large. The first bit in the network bits of a class A network is always 0. The address range is 0.0.0.0–127.255.255.255.
- The number of network bits of a class B network is 16, and the first two bits are always 10. The address range is 128.0.0.0–191.255.255.255.
- The number of network bits of a class C network is 24. The number of network bits is large, so the number of addresses that can be assigned to the hosts is small. The first three bits in the network bits of a class C network are always 110. The address range is 192.0.0.0-223.255.255.255.

Note:

- A host refers to a router or a computer, and the IP address of an interface on a host refers to the host IP address.
- Multicast address: Multicast refers to one-to-many message transmission.

4.1.5 Public/Private IP Address

- Public IP address

Public IP addresses are assigned by the Internet Corporation for Assigned Names and Numbers (ICANN) to ensure that each IP address is unique on the Internet. Public IP addresses can be used for accessing the Internet.

- Private IP address

Some networks do not need to connect to the Internet, for example, a network in a closed lab of a university. However, the IP addresses of network devices on the lab network still need to be unique to avoid conflicts. Some IP addresses of classes A, B, and C are reserved for this kind of situation. These IP addresses are called private IP addresses.

Class A: 10.0.0.0–10.255.255.255

Class B: 172.16.0.0–172.31.255.255

Class C: 192.168.0.0–192.168.255.255

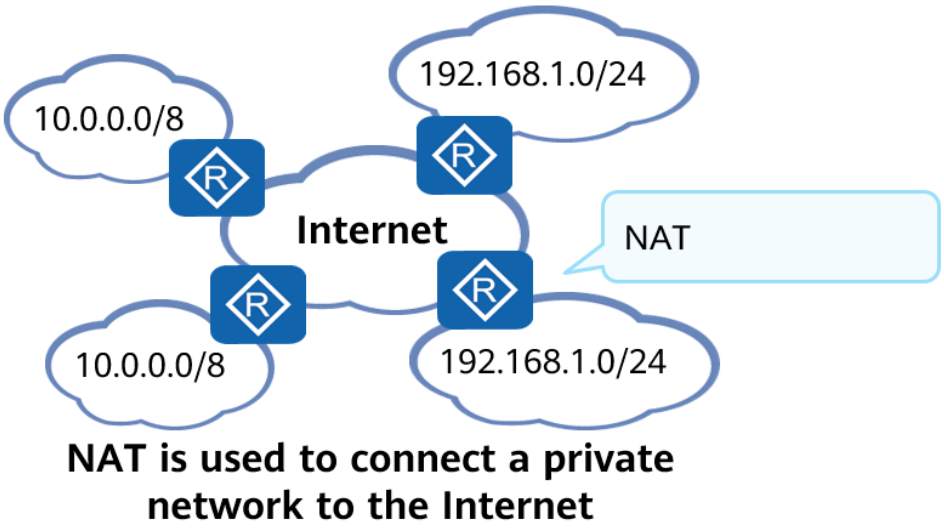


Figure 4-4 Connecting a private network to the Internet

Private IP addresses are used to resolve IP address shortage. At first, a private network is not allowed to directly connect to the Internet because it uses a private IP address. Due to actual requirements, many private networks also want to be connected to the Internet to communicate with the Internet or other private networks through the Internet. The interconnection between a private network and the Internet is implemented through the network address translation (NAT) technology.

Note:

- NAT is used to translate private IP addresses into public IP addresses.
- ICANN is a standards organization that oversees global IP address allocation.

4.1.6 Special IP Addresses

There are some special IP addresses that have special meanings and functions.

Special IP Address	IP Address Range	Function
Limited broadcast address	255.255.255.255	Packets that use this address as the destination address will be sent to all hosts on the same network segment. (The destination range is limited by the gateway.)
Any address	0.0.0.0	This address is the network address of any network, or the IP address of an interface on a network.
Loopback address	127.0.0.0/8	This address is used to test the software system of a device.
Link-local address	169.254.0.0/24	When a host fails to obtain an IP address automatically, the host can use a link-local address for temporary communication.

Figure 4-5 Special IP addresses

- 255.255.255.255

This address is called a limited broadcast address and can be used as the destination IP address of an IP packet.

After receiving an IP packet whose destination IP address is a limited broadcast address, a router stops forwarding the IP packet.

- 0.0.0.0

If this address is used as a network address, it refers to the network address of any network. If this address is used as a host address, it refers to an interface IP address of a host on the network.

For example, when a host does not obtain an IP address during startup, it can send a DHCP Request packet with the source IP address being 0.0.0.0 and the destination IP address being a limited broadcast address to the network. The DHCP server will assign an available IP address to the host after receiving the DHCP Request packet.

- 127.0.0.0/8

This address is a loopback address that can be used as the destination IP address of an IP packet. It is used to test the software system of the device.

An IP packets whose destination IP address is a loopback address cannot leave the device which sends the packet.

- 169.254.0.0/16

If a network device is configured to automatically obtain an IP address but does not find an available DHCP server on the network, the device uses an IP address on the 169.254.0.0/16 network segment for temporary communication.

Note: DHCP is used to dynamically allocate network configuration parameters, such as IP addresses.

4.1.7 Subnet Mask and Available Host Address

Generally, the network range defined by a network ID is called a network segment. The subnet mask is used to calculate the network ID and host ID in an IP address.

Generally, in a network segment, the first address is the network address of the network segment, and the last address is the broadcast address of the network segment. Network addresses and broadcast addresses cannot be used as the addresses of nodes or network devices. Therefore, the number of available IP addresses on a network segment is the number of IP addresses on the entire network segment minus 2. If the number of host

bits of a network segment is *n*, the number of IP addresses on the network segment is 2^n, and the number of available host addresses is 2^n - 2 (subtracting the network address and broadcast address).

4.1.8 IP Address Calculation

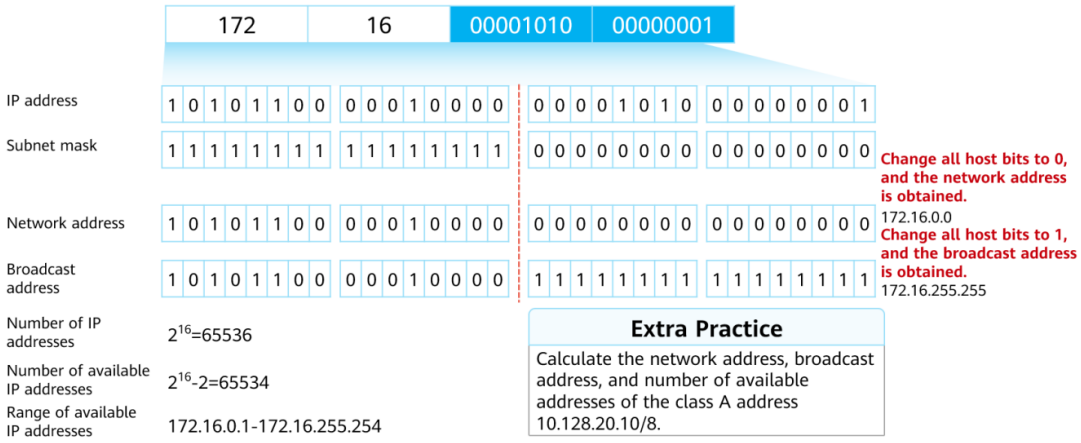


Figure 4-6 IP address calculation

As shown in Figure 4-6, the address calculation formula is as follows:

- Network address: Change all host bits of an IP address to 0, and the result is the network address of the network to which the IP address belongs.
- Broadcast address: Change all host bits of an IP address to 1, and the result is the broadcast address of the network to which the IP address belongs.
- Number of IP addresses: 2^n, where *n* indicates the number of host bits.
- Number of available IP addresses: 2^n - 2, where *n* indicates the number of host bits.

Based on these rules, you can easily calculate the required IP addresses.

The answer to the extra practice in Figure 4-6 is as follows:

- Network address: 10.0.0.0
- Broadcast address: 10.255.255.255
- Number of IP addresses: 224
- Number of available IP addresses: 222 (224 - 2)
- Range of available IP addresses: 10.0.0.1-10.255.255.254

4.1.9 Subnet Division

In practice, if a class A network is assigned to an organization but the number of hosts in the organization is less than 16777214, a large number of IP addresses will be idle and wasted. Therefore, a more flexible method is required to divide the network based on the network scale. The idea is to divide a network into multiple subnets for different organizations to use through the variable length subnet mask (VLSM) technology. VLSM can be used on both public networks and enterprise networks. VLSM allows an organization to divide a network into multiple subnets based on the network scale for different departments to use.

For example, a company is assigned a class C IP address 201.222.5.0. Assume that 20 subnets are required and each subnet contains five hosts. How should we divide the subnets?

In the preceding example, 201.222.5.0 is a class C address, whose default subnet mask is 24. Assume that 20 subnets are required and each subnet contains five hosts. The last byte (8 bits) of 201.222.5.0 should be divided into subnet bits and host bits.

The number of subnet bits determines the number of subnets. As this address is a class C address, the total number for subnet bits and host bits is 8. Because the value 20 is in the range of 2^4 (16) to 2^5 (32), 5 bits should be reserved for subnet bits. The 5-bit subnet part allows a maximum of 32 subnets. The 3 bits left are host bits, which means that there are a maximum of 2^3 (8) IP addresses. Except for one network address and one broadcast address, six addresses can be used by hosts.

The network segments are:

201.222.5.0–201.222.5.7

201.222.5.8–201.222.5.15

201.222.5.16–201.222.5.23

...

201.222.5.232–201.222.5.239

201.222.5.240–201.222.5.247

201.222.5.248–201.222.5.255

4.2 Introduction to Network Technologies

4.2.1 Network Basics

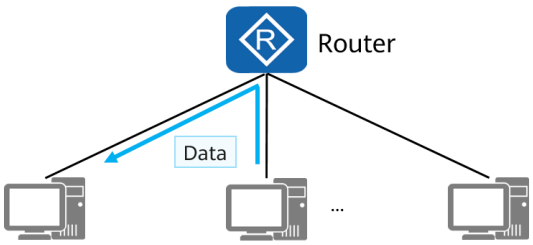
4.2.1.1 Concept of Network Communication

Communication refers to the information transfer and exchange between people, between people and things, and between things through a certain medium and action.

Network communication refers to communication between terminal devices through a computer network.

A. Files are transferred between two computers (terminals) through a network cable.

C. A computer (terminal) downloads files through the Internet.



B. Files are transferred among multiple computers (terminals) through a router.

Figure 4-7 Network communication

Examples of network communication:

- A: Two computers are connected through a network cable to form a simple network.
- B: A router (or switch) and multiple computers form a small-scale network. In such a network, files can be freely transferred between every two computers through a router.
- C. If a computer wants to download files from a website, it must access the Internet first.

The Internet is the largest computer network in the world. Its predecessor, Advanced Research Projects Agency Network (ARPANET), was born in 1969. The wide popularization and application of Internet is one of the signs of entering the information age.

4.2.1.2 Information Transfer Process

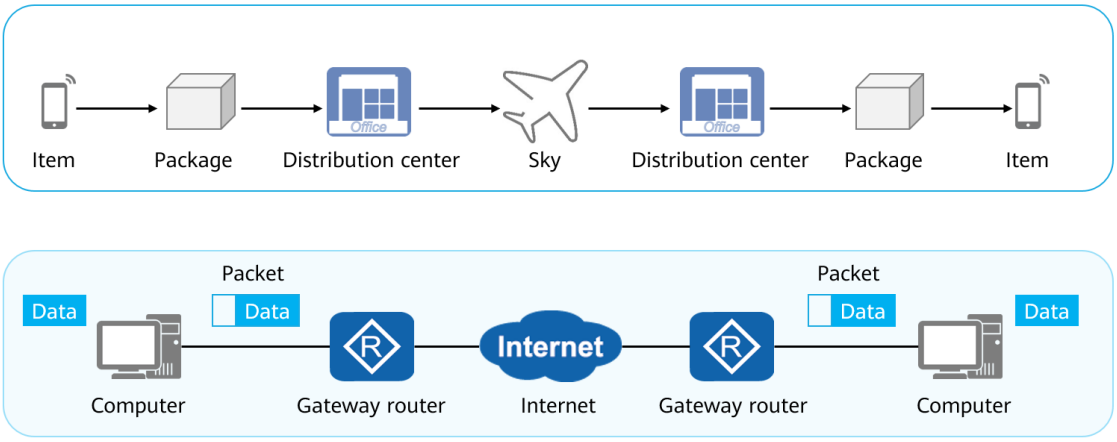


Figure 4-8 Information transfer process

There are many similarities between virtual information transfer and real item transfer. We can compare the express delivery process with the network communication process.

1. Items to be delivered:

- The information (or data) generated by the application.

2. The item is packed into a package and pasted with a package label containing the receiver's name and address:

- The application packs the data into an original data payload and adds a header and a tail to form a packet. The important information in the packet is the address of the receiver, that is, the destination address.
- Encapsulation is a process in which new information segments are added to an information unit, forming a new information unit.

3. The package is delivered to a distribution center in which packages are sorted based on the destination addresses. The packages destined for the same city are placed in the same plane for airlift:

- The packet reaches the gateway through a network cable. After receiving the packet, the gateway decapsulates the packet, obtains the destination address, re-

encapsulates the packet, and sends the packet to different routers based on the destination address. The packet is transmitted through the gateway and router, leaves the local network, and is transmitted through the Internet.

- The network cable is the medium for information transmission, and plays the same role as the highway for item transmission.

4. After the plane arrives at the destination airport, the packages are taken out for sorting, and the packages destined for the same area are sent to the same distribution center:

- The packet is transmitted through the Internet and reaches the local network where the destination address resides. The gateway or router of the local network decapsulates and encapsulates the packet, and then determines the next-hop router according to the destination address. Finally, the packet reaches the gateway of the network where the destination computer resides.

5. The distribution center sorts the packages according to the destination addresses on the packages. The courier delivers the packages to the receiver. The receiver unpacks the package, confirms that the items are intact, and signs for the package. The entire express delivery process is complete:

- After the packet reaches the gateway of the network where the destination computer resides, the gateway decapsulates and encapsulates the packet, and then sends the packet to the corresponding computer according to the destination address. After receiving the packet, the computer verifies the packet. If the packet passes verification, the computer accepts the packet and sends the data payload to the corresponding application program for processing. A complete network communication process is complete.

4.2.1.3 What Is a Gateway

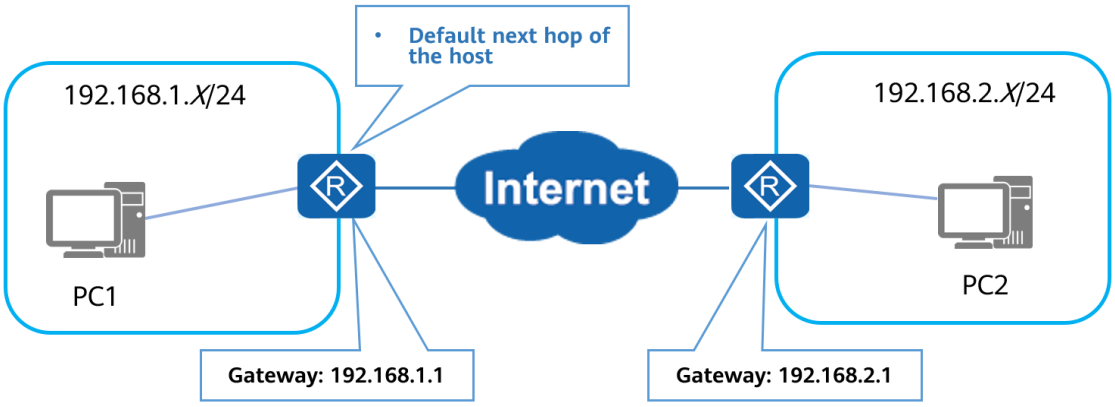


Figure 4-9 What is a gateway

A gateway is also called an inter-network connector or a protocol converter. A default gateway implements network interconnection above the network layer.

Just like you must walk through a door when entering a room, information sent from one network or network segment to another must pass through a gateway. We can say the gateway is the door to another network.

A gateway plays significant roles in not only its role but also its configuration:

- When a host (such as a PC, server, router, or firewall) wants to access another network segment, the gateway is responsible for sending ARP packets, and receiving and forwarding subsequent data packets.
- After the gateway is configured, the default route is generated on the host, with the next hop being the gateway.

4.2.1.4 Basic Architecture of a Communication Network

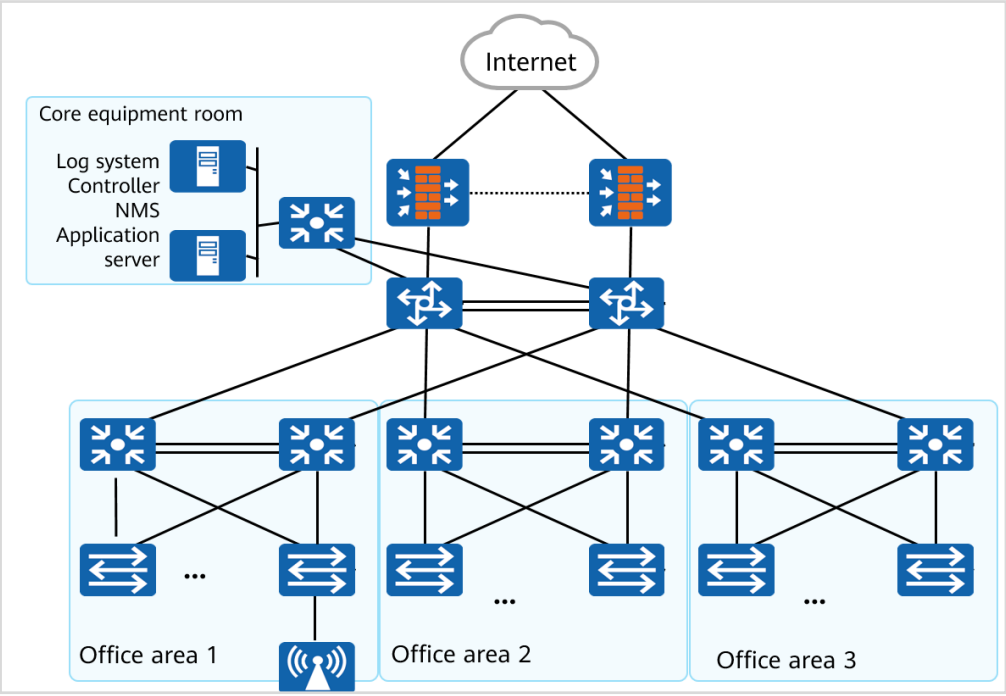


Figure 4-10 Basic architecture of a communication network

Figure 4-10 shows an enterprise data center network (DCN). The major requirements of an enterprise for the DCN include service operation and computing, data storage, and service access.

The DCN thereby needs to enable device-device and device-user interconnection and provide external access capabilities for services. Devices on such a network collaborate with each other to implement communication:

- Access switches connect to user hosts in office areas. Aggregation switches aggregate traffic from access switches.
- Routers forward traffic between different office areas and between internal and external networks.
- Firewalls implement access control for areas of different security levels and between internal and external networks to ensure secure access.

In conclusion, a communication network consists of routers, switches, firewalls, PCs, network printers, servers, and more, and its basic function is to implement data communication.

4.2.1.5 Network Device - Switch

Generally, on a campus network, switches are closest to end users, and Layer 2 switches (also known as Ethernet switches) are deployed at the access layer. Layer 2 refers to the

data link layer of the TCP/IP model. A switch connects end users to a network and forwards data frames.

A switch can:

- Connect terminals (such as PCs and servers) to the network.
- Isolate collision domains.
- Broadcast unknown packets.
- Learn MAC addresses and maintain the MAC address table.
- Forward packets based on the MAC address table.

Note:

Broadcast domain: a group of nodes, among which a broadcast packet from one node can reach all the other nodes.

Collision domain: an area where a collision occurs when two devices on the same network send packets at the same time.

Media Access Control (MAC) address: uniquely identifies a network interface card (NIC) on a network. Each NIC requires and has a unique MAC address.

MAC address table: exists on each switch and stores the mapping between MAC addresses and switch interfaces.

4.2.1.6 Network Device - Router

Working at the network layer, a router forwards data packets on the Internet. Based on the destination address in a received packet, a router selects a path to send the packet to the next router or destination. The last router on the path is responsible for sending the packet to the destination host.

A router can:

- Implement communication between networks of the same type or different types.
- Isolate broadcast domains.
- Maintain the routing table and run routing protocols.
- Select routes and forward IP packets.
- Implement WAN access and network address translation (NAT).
- Connect Layer 2 networks built through switches.

4.2.1.7 Network Device - Firewall

As a network security device, a firewall is used to ensure secure communication between two networks. Located between two networks of different trust levels (for example, an enterprise intranet and the Internet), a firewall controls the communication between the two networks and forcibly implements unified security policies to prevent unauthorized access to key information resources, ensuring system security.

A firewall can:

- Isolate networks of different security levels.
- Implement access control (using security policies) between networks of different security levels.
- Perform user identity authentication.

- Implement remote access.
- Encrypt data and provide virtual private network (VPN) services.
- Implement NAT.
- Provide other security functions.

4.2.2 Network Reference Model and Data Encapsulation

4.2.2.1 OSI Reference Model

To achieve compatibility between networks and help vendors produce compatible network devices, the International Organization for Standardization (ISO) launched the Open Systems Interconnection (OSI) reference model in 1984. It was quickly adopted as the basic model for computer network communication.

7. Application layer	Provides interfaces for applications.
6. Presentation layer	Converts data formats to ensure the application layer of one system can identify and understand the data generated by the application layer of another system.
5. Session layer	Establishes, manages, and terminates sessions between two parties.
4. Transport layer	Establishes, maintains, and cancels one-time end-to-end data transmission processes, controls transmission speeds, and adjusts data sequencing.
3. Network layer	Defines logical addresses and transfers data from sources to destinations.
2. Data link layer	Encapsulates packets into frames, transmits frames in P2P or P2MP mode, and implements error checking.
1. Physical layer	Transmits bit streams over transmission media and defines electrical and physical specifications.

Figure 4-11 OSI reference model

The OSI reference model is also called the seven-layer model. The seven layers from bottom to top are as follows:

- Physical layer: transmits bit streams between devices and defines physical specifications such as electrical levels, speeds, and cable pins.
- Data link layer: encapsulates bits into octets and octets into frames, uses link layer addresses (MAC addresses in Ethernet) to access media, and implements error checking.
- Network layer: defines logical addresses for routers to determine paths and transmits data from source networks to destination networks.
- Transport layer: implements connection-oriented and non-connection-oriented data transmission, as well as error checking before retransmission.
- Session layer: establishes, manages, and terminates sessions between entities at the presentation layer. Communication at this layer is implemented through service requests and responses transmitted between applications on different devices.
- Presentation layer: provides data encoding and conversion functions so that data sent by the application layer of one system can be identified by the application layer of another system.
- Application layer: provides network services for applications and is closest to users.

4.2.2.2 TCP/IP Reference Model

The TCP/IP reference model has become the mainstream reference model of the Internet because the TCP and IP protocols are widely used and the OSI model is too complex.

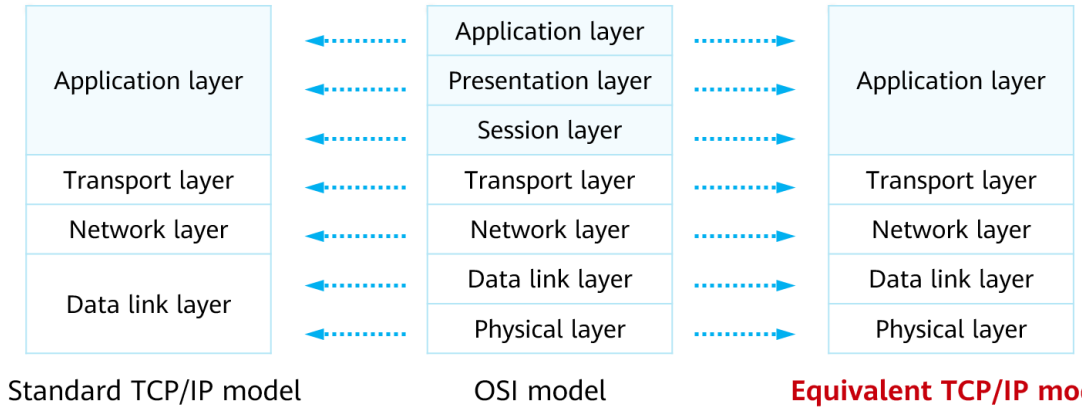


Figure 4-12 TCP/IP model

Similar to the OSI model, the Transmission Control Protocol/Internet Protocol (TCP/IP) model adopts a hierarchical architecture, and adjacent layers are closely related.

The standard TCP/IP model combines the data link layer and physical layer in the OSI model into the network access layer. This division mode is contrary to the actual protocol formulation. Therefore, the equivalent TCP/IP model that integrates the standard TCP/IP model and the OSI model is proposed. Contents in the following sections are based on the equivalent TCP/IP model.

TCP/IP was originated from a packet switched network research project funded by the US government in the late 1960s. Since the 1990s, the TCP/IP model has become the most commonly used networking model for computer networks. It is a truly open system, because the definition of the protocol suite and its multiple implementations can be easily obtained at little or even no cost. It thereby became the basis of the Internet.

Like the OSI reference model, the TCP/IP model is developed in different layers, each of which is responsible for different communication functions. The difference is, the TCP/IP model has a simplified hierarchical structure that consists of only five layers: application layer, transport layer, network layer, data link layer, and physical layer. As shown in Figure 4-12, the TCP/IP protocol stack corresponds to the OSI reference model and covers all layers in the OSI reference model. The application layer contains all upper-layer protocols in the OSI reference model.

The TCP/IP protocol stack supports all standard physical-layer and data-link-layer protocols. The protocols and standards at the two layers will be further discussed in following sections.

Comparison between the OSI reference model and TCP/IP protocol stack:

- Similarities

1. They are both hierarchical and both require close collaboration between layers.
2. They both have the application layer, transport layer, network layer, data link layer, and physical layer. (Note: The TCP/IP protocol stack is divided into five layers here to facilitate comparison. In many documents, the data link layer and physical layer of TCP/IP are combined into the data link layer, which is also called network access layer.)
3. They both use the packet switching technology.
4. Network engineers must understand both models.

- Differences

1. TCP/IP includes the presentation layer and session layer into the application layer.
2. TCP/IP has a simpler structure with fewer layers.
3. TCP/IP standards are established based on practices during the Internet development and are thereby highly trusted. In comparison, the OSI reference model is based on theory and serves as a guide.

4.2.2.3 Data Encapsulation on the Sender

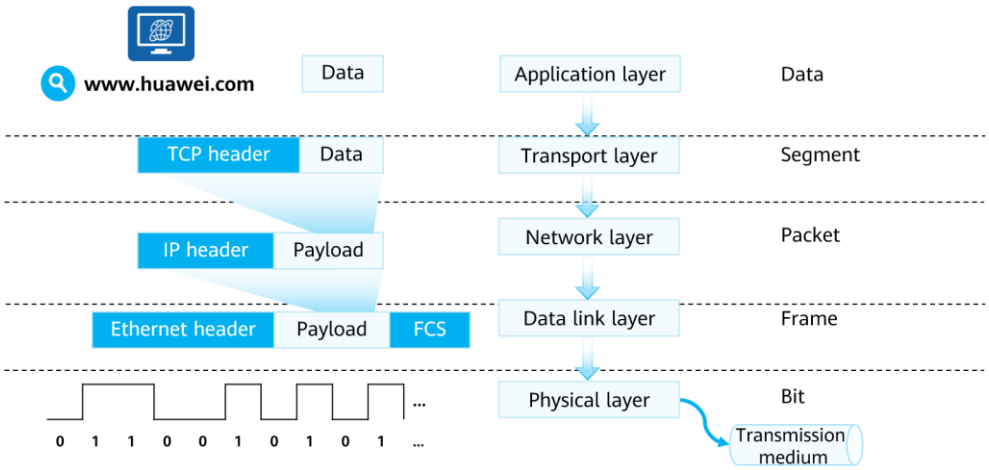


Figure 4-13 Data encapsulation on the sender

Assume that you are using a web browser to access Huawei's official website. After you enter the website address and press **Enter**, the following events occur on your computer:

- Internet Explorer (application) invokes HTTP (application-layer protocol) to encapsulate the application-layer data. (**Data** in the figure should also include the HTTP header, which is not shown here.)
- HTTP uses TCP to ensure reliable data transmission and thereby transmits the encapsulated data to the TCP module.
- The TCP module adds the corresponding TCP header information (such as the source and destination port numbers) to the data transmitted from the application layer. The protocol data unit (PDU) is called a segment.
- On an IPv4 network, the TCP module sends the encapsulated segment to the IPv4 module at the network layer. (On an IPv6 network, the segment is sent to the IPv6 module for processing.)
- After receiving the segment from the TCP module, the IPv4 module encapsulates the IPv4 header. Here, the PDU is called a packet.
- Ethernet is used as the data link layer protocol. Therefore, after the IPv4 module completes encapsulation, it sends the packet to the Ethernet module (such as the Ethernet adapter) at the data link layer for processing.
- After receiving the packet from the IPv4 module, the Ethernet module adds the corresponding Ethernet header and FCS frame trailer to the packet. Now, the PDU is called a frame.

- After the Ethernet module completes encapsulation, it sends the data to the physical layer.
- Based on the physical media, the physical layer converts digital signals into electrical signals, optical signals, or electromagnetic (wireless) signals.
- The converted signals are then transmitted on the network.

4.2.2.4 Data Transmission on the Intermediate Network

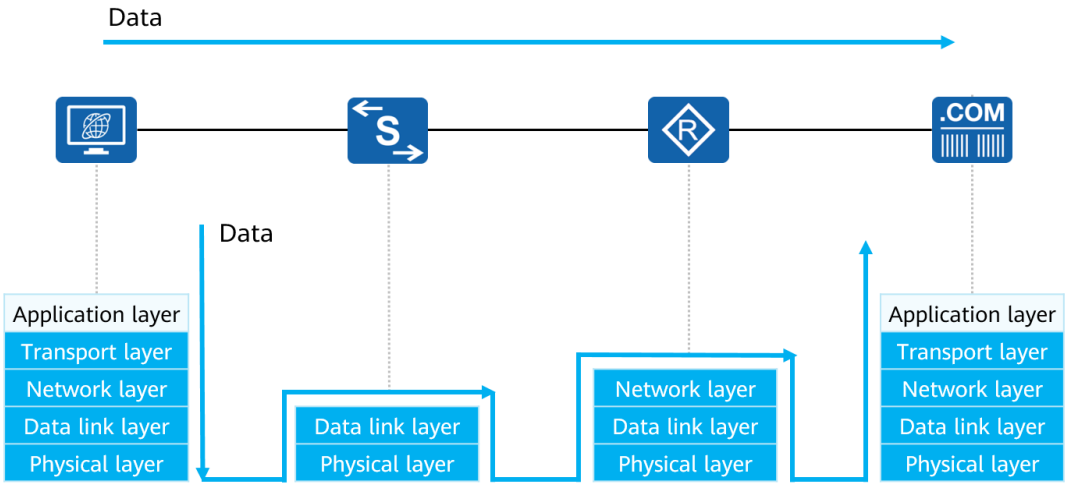


Figure 4-14 Data transmission on the intermediate network

Encapsulated data is transmitted on the network.

In most cases:

- A Layer 2 device (such as an Ethernet switch) only decapsulates the Layer 2 header of the data and performs the corresponding switching operation based on the Layer 2 header information.
- A Layer 3 device (such as a router) only decapsulates the Layer 3 header and performs the corresponding routing operation based on the Layer 3 header information.

4.2.2.5 Data Decapsulation on the Receiver

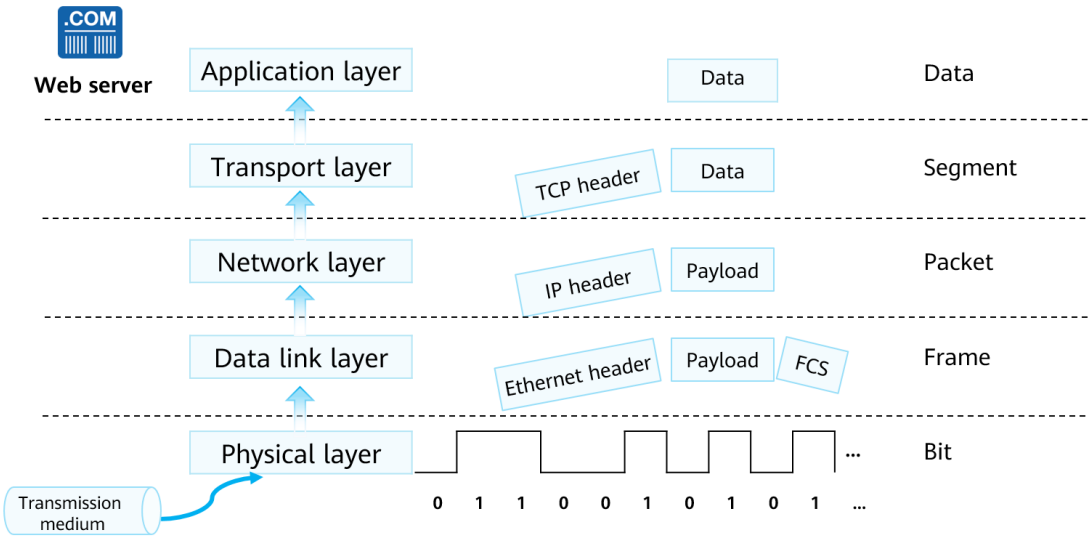


Figure 4-15 Data decapsulation on the receiver

As shown in Figure 4-15, after being transmitted over the intermediate network, the data finally reaches the destination server. Based on the information in different protocol headers, the data is decapsulated layer by layer, processed, transmitted, and finally sent to the application on the web server for processing.

4.2.3 Introduction to Common Protocols

4.2.3.1 Common TCP/IP Protocols

Application layer	Telnet	FTP	TFTP	SNMP
	HTTP	SMTP	DNS	DHCP
Transport layer	TCP		UDP	
Network layer	ICMP		IGMP	
	IP			
Data link layer	PPPoE			
	Ethernet		PPP	
Physical layer	...			

Figure 4-16 Common TCP/IP protocols

Figure 4-16 shows some common TCP/IP protocols.

- Hypertext Transfer Protocol (HTTP): used to access various pages on web servers.
- File Transfer Protocol (FTP): used to transfer data from one host to another.
- Domain Name Service (DNS): translates domain names of hosts into IP addresses.
- Transmission Control Protocol (TCP): provides reliable and connection-oriented communication services for applications. Currently, TCP is used by many popular applications.
- User Datagram Protocol (UDP): provides connectionless communication services, without guaranteeing the reliability of packet transmission.

- Internet Protocol (IP): encapsulates transport-layer data into data packets and forwards packets from source sites to destination sites. IP provides a connectionless and unreliable service.
- Internet Group Management Protocol (IGMP): manages multicast group memberships. Specifically, IGMP sets up and maintains memberships between IP hosts and their directly connected multicast routers.
- Internet Control Message Protocol (ICMP): sends control messages based on the IP protocol and provides information about various problems that may exist in the communication environment. Such information helps administrators diagnose problems and take proper measures to resolve the problems.
- Address Resolution Protocol (ARP): a TCP/IP protocol that discovers the data link layer address associated with a given IP address. It maps IP addresses to MAC addresses, maintains the ARP table that caches the mapping between IP addresses and MAC addresses, and detects IP address conflicts on a network segment.

The following sections describe several of these protocols in detail.

4.2.3.2 TCP

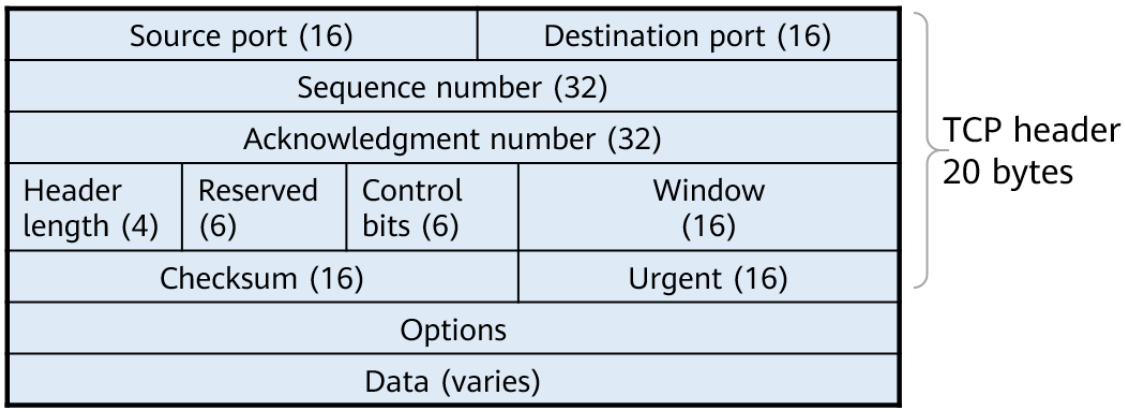


Figure 4-17 TCP packet format

Working at the transport layer, TCP provides reliable and connection-oriented services for applications.

TCP provides reliability in the following aspects:

- Connection-oriented transmission: A connection must be established before either side sends data.
- Maximum segment size (MSS): limits the maximum length of a TCP packet sent to the receiver. When a connection is established, both parties of the connection advertise their MSSs to make full use of bandwidth resources.
- Transmission acknowledgment mechanism: After the sender sends a data segment, it starts a timer and waits for an acknowledgment from the receiver. If no acknowledgment is received when the timer expires, the sender resends the data segment.
- Checksum of the header and data: TCP maintains the checksum of the header and data, implementing end-to-end check to verify whether the data changes during transmission. If the checksum of a received segment is incorrect, TCP discards the

segment and does not acknowledge the receipt of the segment. In this case, TCP starts the retransmission mechanism.

- Flow control: Each party of a TCP connection has a buffer with a fixed size. The receiver allows the sender to send only the data that can be stored in the receive buffer, which prevents buffer overflow caused by the high transmission rate of the sender.

4.2.3.3 UDP

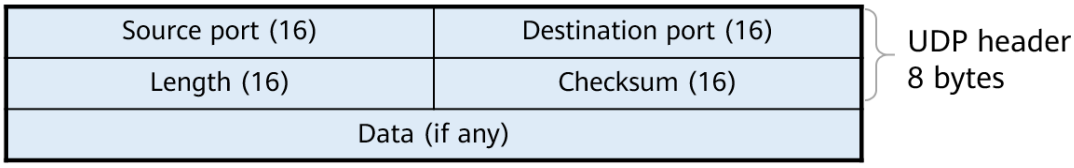


Figure 4-18 UDP packet format

Also working at the transport layer, UDP provides connectionless services for applications. That is, no connection needs to be established between the source and destination ends before data transmission. UDP does not maintain connection states or sending and receiving states. Therefore, a server can transmit the same message to multiple clients at the same time.

UDP applies to applications that require high transmission efficiency or have the reliability guaranteed at the application layer. For example, the Remote Authentication Dial-In User Service (RADIUS) protocol used for authentication and accounting and Routing Information Protocol (RIP) are based on UDP.

4.2.3.4 TCP vs. UDP

TCP and UDP are often compared because they both work at the transport layer and provide transmission services for the application layer. Figure 4-19 compares TCP and UDP.

TCP	UDP
• Connection-oriented • Reliable transmission with flow and congestion control • Header length: 20–60 bytes • Applies to applications that require reliable transmission, such as file transfer	• Connectionless • Unreliable transmission, with packet reliability guaranteed by upper-layer applications • Short header length of 8 bytes • Applies to real-time applications, such as video conferencing

Figure 4-19 TCP vs. UDP

TCP is reliable, but its reliability mechanism leads to low packet transmission efficiency and high encapsulation overhead.

UDP is connectionless and unreliable, but its transmission efficiency is higher.

They both have advantages and disadvantages and apply to different scenarios.

4.2.3.5 Telnet

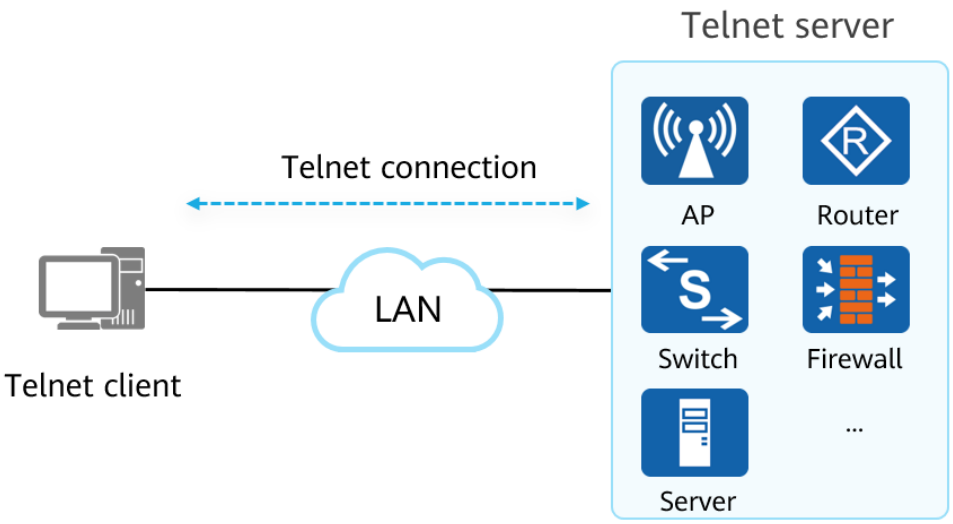


Figure 4-20 Telnet connection

Telnet provides remote login services on data networks. It allows users to remotely log in to a device from a local PC. Telnet data is transmitted in plaintext. Telnet enables network administrators to remotely log in to network devices for configuration and management.

As shown in Figure 4-20, a user connects to a Telnet server through a Telnet client program. The commands entered on the Telnet client are executed on the server, as if the commands were entered on the console of the server.

However, Telnet has the following disadvantages:

- Data is transmitted in plaintext, which does not ensure confidentiality.
- The authentication mechanism is weak. Users' authentication information is transmitted in plaintext and may be eavesdropped. Telnet supports only the traditional password authentication mode and is vulnerable to attacks.
- A client cannot truly identify the server. As a result, attackers can use a bogus server to launch attacks.

SSH was designed to resolve the preceding issues.

4.2.3.6 SSH

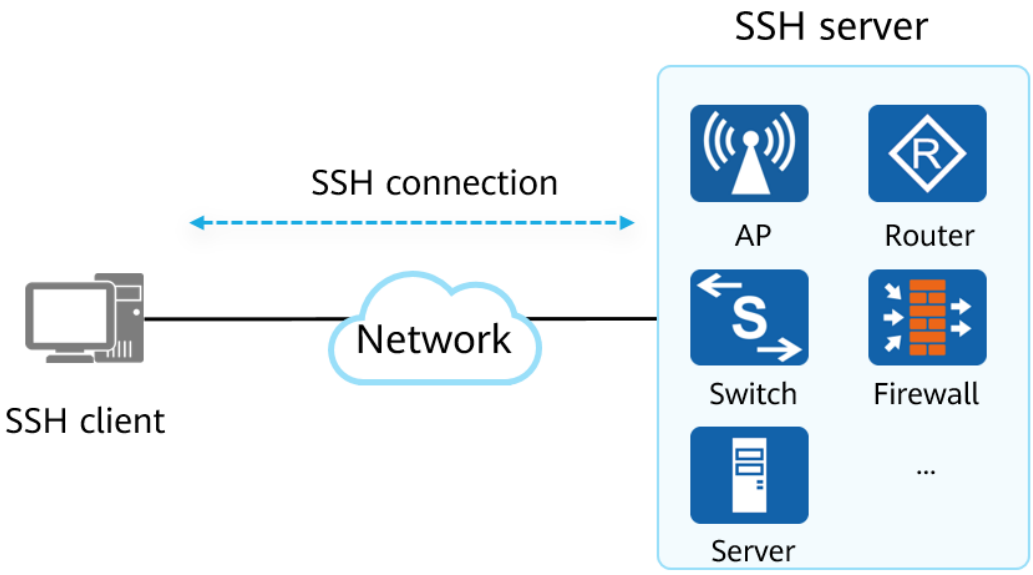


Figure 4-21 SSH connection

SSH provides similar functions as Telnet. SSH is a network security protocol that employs encryption and authentication mechanisms to implement services such as secure remote access and file transfer.

SSH was developed to resolve security issues that Telnet may bring, ensuring secure remote access to network devices.

SSH uses the client/server architecture and involves three layers: transport layer, authentication layer, and connection layer.

SSH protocol layers:

- Transport layer: establishes a secure encryption channel between a client and a server to provide sufficient confidentiality protection for phases that require high data transmission security, such as user authentication and data exchange.
- Authentication layer: runs over transport-layer protocols and helps a server authenticate login users.
- Connection layer: divides an encryption channel into several logical channels to run different applications. It runs over authentication-layer protocols and provides services such as session interaction and remote command execution.

4.2.3.7 Telnet vs. SSH

Telnet	SSH
• Data is transmitted in plaintext. • Weak authentication mechanism: User authentication information is transmitted in plaintext. • Only traditional password authentication is available. • A client cannot truly identify a server.	• Data is transmitted in ciphertext. • User authentication information is transmitted in ciphertext. • In addition to password authentication, SSH servers support multiple user authentication modes, such as public key authentication that has higher security. • Encryption and decryption keys are dynamically generated for communication between the client and server. • Provides the server authentication function for clients.

Figure 4-22 Telnet vs. SSH

Figure 4-22 compares Telnet and SSH.

In general, SSH encrypts data before sending it, ensuring data transmission security. It applies to scenarios where encrypted authentication is required. Telnet is still used in tests or scenarios where encryption is not required (such as on a LAN).

4.3 Switching Basics

4.3.1 Ethernet Switching Basics

4.3.1.1 Ethernet Protocol

Ethernet is the most common communication protocol standard used by existing local area networks (LANs). It defines the cable types and signal processing methods that are used on a LAN.

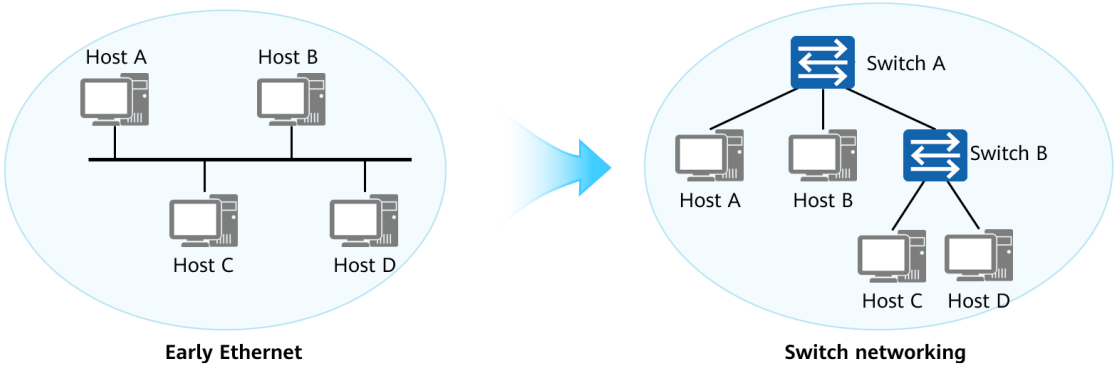


Figure 4-23 Evolution of Ethernet networking

As shown in Figure 4-23, the Ethernet has evolved from the hub networking to the switch networking.

- Early Ethernet: Ethernet networks are broadcast networks established based on the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) mechanism.

Collisions restrict Ethernet performance. Early Ethernet devices such as hubs work at the physical layer, and cannot confine collisions to a particular scope. This restricts network performance improvement.

- Switch networking: Working at the data link layer, switches are able to confine collisions to a particular scope, thereby helping improve Ethernet performance. Switches have replaced hubs as mainstream Ethernet devices. However, switches do not restrict broadcast traffic on the Ethernet. This affects Ethernet performance.

4.3.1.2 Layer 2 Ethernet Switch

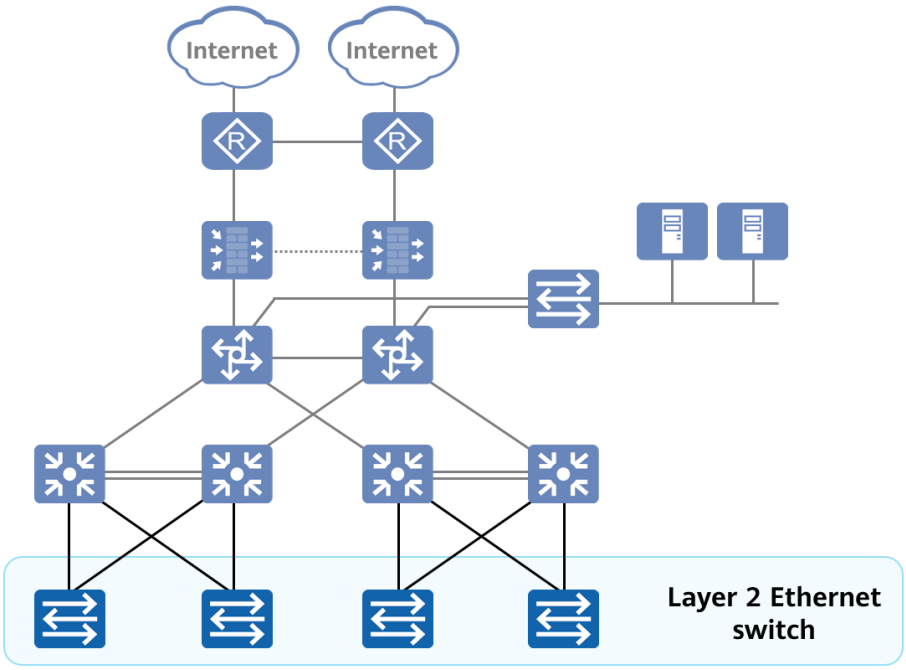


Figure 4-24 Architecture of a communication network

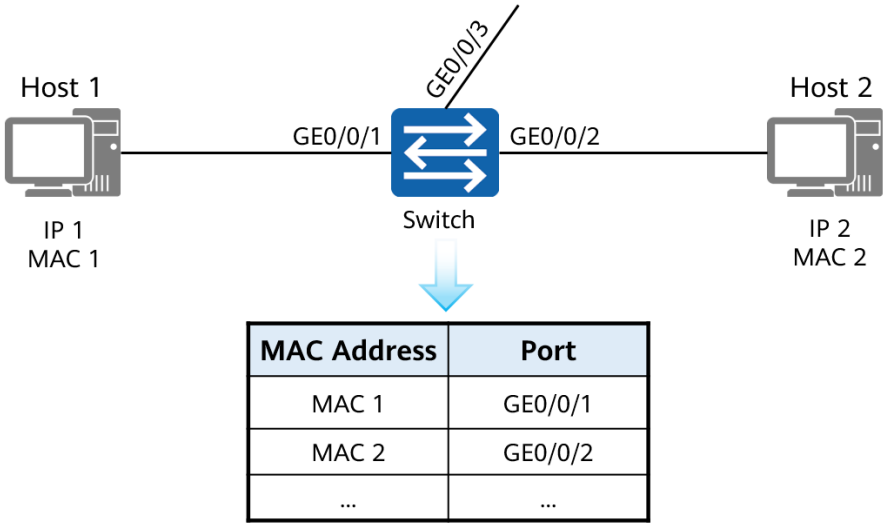
As shown in Figure 4-24, Layer 2 Ethernet switches are located at the edge of a communication network and function as access devices for user and terminal access. Layer 2 Ethernet switches forward data through Ethernet interfaces. Specifically, a switch performs addressing and forwards data only based on the MAC address in the Layer 2 header of an Ethernet data frame.

On a campus network, a switch is the device closest to end users and is used to connect terminals to the campus network. Switches at the access layer are typically Layer 2 switches. A Layer 2 switch works at the second layer (data link layer) of the TCP/IP model and forwards data packets based on MAC addresses. In Figure 4-24, Layer 3 switches are above the Layer 2 switches. Generally, routers are required to implement network communication between different LANs. As data communication networks expand and more services emerge on the networks, increasing traffic needs to be transmitted between networks. Routers cannot adapt to this development trend because of their high costs, low forwarding performance, and small interface quantities. New devices capable of high-speed Layer 3 forwarding are required. Layer 3 switches are such devices.

Note: The switches involved in this course refer to Layer 2 Ethernet switches.

4.3.1.3 MAC Address Table

Each switch has a MAC address table that stores the mapping between MAC addresses and switch interfaces.



MAC Address	Port
MAC 1	GE0/0/1
MAC 2	GE0/0/2
...	...

Figure 4-25 MAC address table

A MAC address table records the mapping between MAC addresses learned by a switch and switch interfaces. When forwarding a data frame, the switch looks up the MAC address table based on the destination MAC address of the frame. If the MAC address table contains an entry mapping the destination MAC address of the frame, the frame is directly forwarded through the outbound interface in the entry. If there is no match of the destination MAC address of the frame in the MAC address table, the switch floods the frame to all interfaces except the interface that receives the frame.

4.3.1.4 Three Frame Processing Behaviors of a Switch

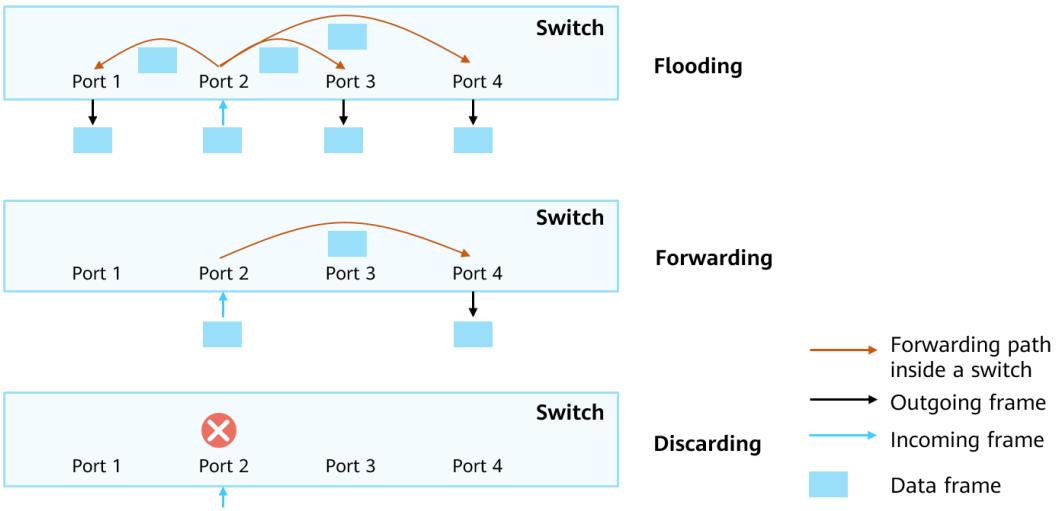


Figure 4-26 Frame processing behaviors of a switch

A switch forwards each frame that enters an interface over a transmission medium, which is also the basic function of a switch.

Figure 4-26 shows that a switch processes frames in three ways: flooding, forwarding, and discarding.

- Flooding: The switch forwards the frames received from an interface to all other interfaces.
- Forwarding: The switch forwards the frames received from an interface to another interface.
- Discarding: The switch discards the frames received from an interface.

A switch process a received frame based on the destination MAC address of the frame and the MAC address table.

- Flooding: If the destination MAC address of the frame received by the switch is a broadcast MAC address or does not match any entry in the MAC address table, the switch floods the frame.
- Forwarding: If the destination MAC address of the frame received by the switch is a unicast MAC address and matches an entry in the MAC address table, and the interface that receives the frame is different from that of the matched entry, the switch forwards the frame.
- Discarding: If the destination MAC address of the frame received by the switch is a unicast MAC address and matches an entry in the MAC address table, and the interface that receives the frame is the same as that of the matched entry, the switch discards the frame.

4.3.1.5 Working Principles of Switches

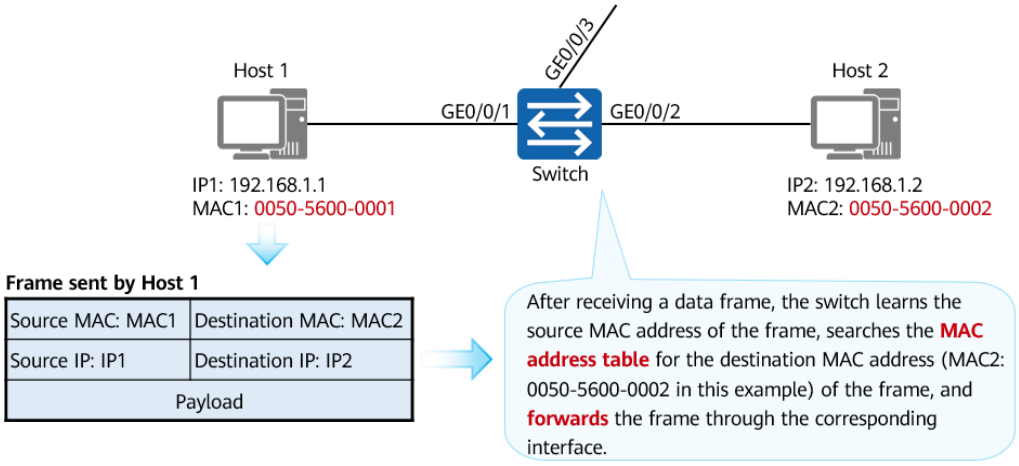


Figure 4-27 Working principles of switches

Layer 2 switches work at the data link layer and forward frames based on MAC addresses. Different interfaces on a switch send and receive data independently, and each interface belongs to a different collision domain. This effectively isolates collision domains on the network.

Layer 2 switches maintain the mappings between MAC addresses and interfaces by learning the source MAC addresses of Ethernet frames in a table called a MAC address table. Layer 2 switches look up the MAC address table to determine the interface to which a frame is forwarded based on the destination MAC address of the frame.

Figure 4-27 shows the working principles of a switch.

- Host 1 sends a unicast frame with the destination MAC address being the MAC address of Host 2.
- When GE0/0/1 of the switch receives the frame, the switch learns the mapping between GE0/0/1 and MAC1 and stores the mapping in the MAC address table.
- The switch then searches its MAC address table based on the destination MAC address of the frame to determine whether to forward, flood, or discard the frame.

4.3.2 VLAN Basics

4.3.2.1 Why Do We Need VLANs

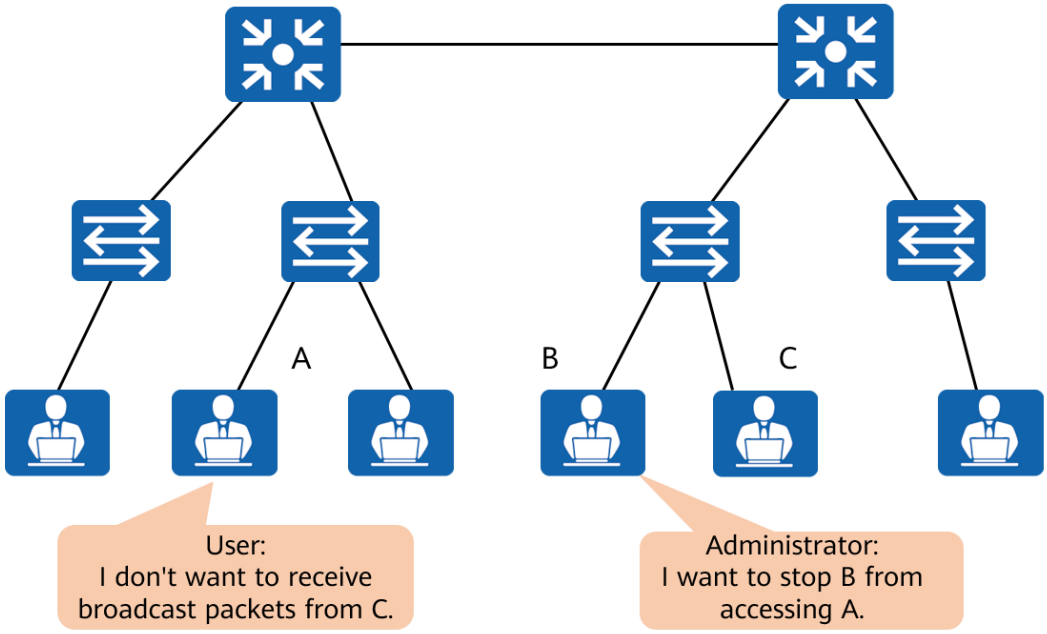


Figure 4-28 Why do we need VLANs

As shown in Figure 4-28, traditional Ethernet switches learn source MAC addresses (MAC addresses of hosts connected to the switch interfaces) of received frames to generate a forwarding table, based on which the switch then forwards frames. All the interfaces can communicate with each other, meaning that maintenance personnel cannot control forwarding between interfaces. Such a network has the following disadvantages:

- Low network security: The network is prone to attacks because all interfaces can communicate with each other.
- Low forwarding efficiency: Users may receive a large number of unnecessary packets such as broadcast packets, which consume a lot of bandwidth and host CPU resources.
- Low service scalability: Network devices process packets on an equal basis and cannot provide differentiated services. For example, Ethernet frames used for network management cannot be preferentially forwarded.

In a word, broadcast packets have a wide-ranging impact on a network, and Ethernet has no method for forwarding control. The Virtual Local Area Network (VLAN) technology solves this problem.

4.3.2.2 Objectives of the VLAN Technology

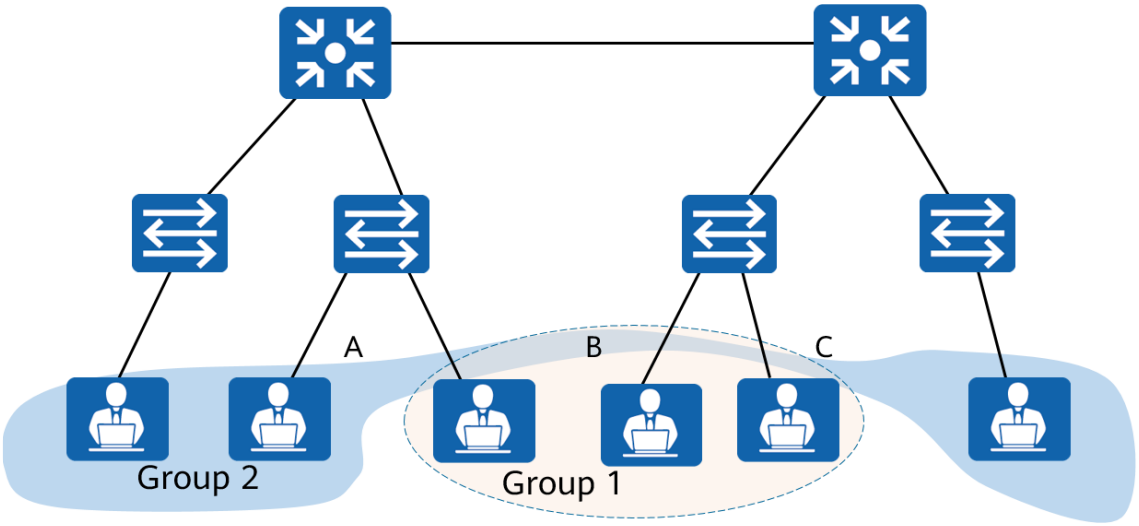


Figure 4-29 Objectives of the VLAN technology

As shown in Figure 4-29, the VLAN technology divides users into multiple logical groups (networks). Intra-group communication is allowed, whereas inter-group communication is prohibited. Layer 2 unicast, multicast, and broadcast packets can be forwarded only within a group. In addition, group members can be easily added or deleted.

In short, the VLAN technology provides a management method for controlling the communication between terminals. As shown in the figure above, PCs in Group 1 and PCs in Group 2 cannot communicate with each other.

4.3.2.3 What Is VLAN

The VLAN technology logically divides a physical LAN into multiple VLANs (broadcast domains).

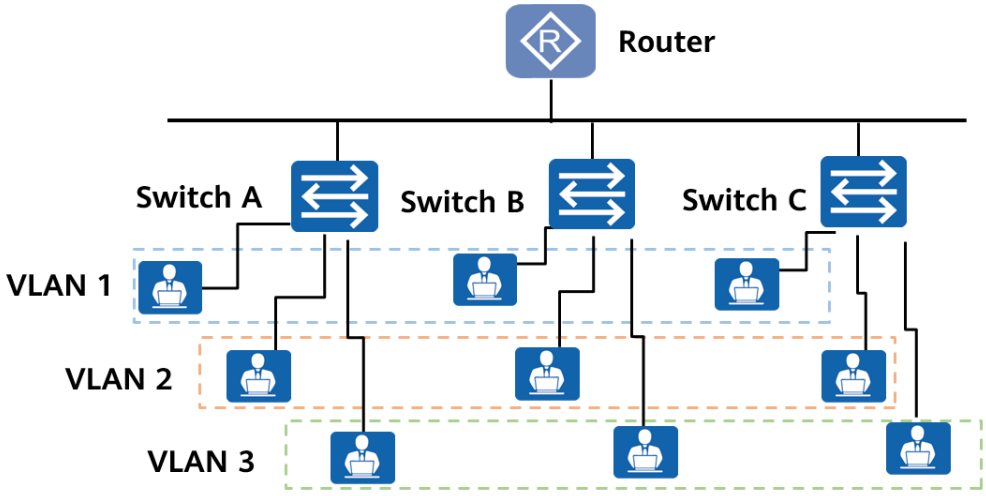


Figure 4-30 VLAN

Hosts within a VLAN can communicate with each other but cannot communicate directly with hosts in other VLANs. This confines broadcast packets within a single VLAN. Inter-VLAN communication is not allowed, which improves network security. For example, if enterprises in the same building establish their own LANs, the cost is high. If enterprises share the same LAN in the building, there may be security risks. In this case, the VLAN technology can be adopted to enable enterprises to share the same LAN while ensuring information security.

Figure 4-30 shows a typical VLAN networking. Three switches are deployed at different locations, for example, on different floors of a building. Each switch is connected to three PCs that belong to different VLANs (for example, VLANs for different enterprises).

4.3.2.4 VLAN Frame Format

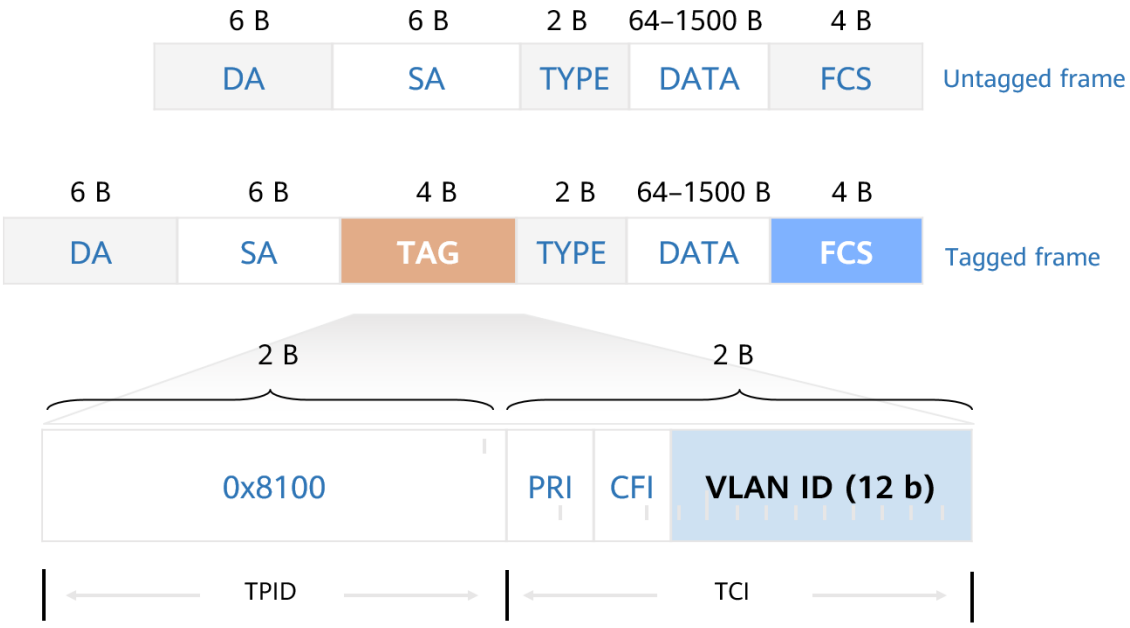


Figure 4-31 VLAN frame format

As shown in Figure 4-31, IEEE 802.1Q adds a 4-byte VLAN tag to an Ethernet frame header.

Tag Protocol Identifier (TPID): identifies a frame as an 802.1Q-tagged frame. This field is of 2 bytes and has a fixed value of 0x8100.

Tag Control Information (TCI): indicates the control information of an Ethernet frame. This field is of 2 bytes.

- Priority: identifies the priority of an Ethernet frame. This field is of 3 bits. The value of this field ranges from 0 to 7, providing differentiated forwarding services.
- Canonical Format Indicator (CFI): indicates the bit order of address information in an Ethernet frame. This field is used in token ring or FDDI source-routed MAC methods and is of 1 bit.

- VLAN Identifier (VLAN ID): controls the forwarding of Ethernet frames based on the VLAN configuration on a switch interface. This field is of 12 bits, with its value ranging from 0 to 4095.

Since VLAN tags are adopted, Ethernet frames are classified as untagged frames (without 4-byte VLAN tags) or tagged frames (with 4-byte VLAN tags).

Note: In this course, only the VLAN ID field is discussed.

4.3.2.5 VLAN Assignment Methods

PCs send only untagged frames on a network. After receiving such an untagged frame, a switch that supports the VLAN technology needs to assign the frame to a specific VLAN based on certain rules.

Available VLAN assignment methods are as follows:

1: Interface-based assignment: assigns VLANs based on switch interfaces.

- A network administrator preconfigures a port VLAN ID (PVID) for each switch interface. When an untagged frame arrives at an interface of a switch, the switch tags the frame with the PVID of the interface. The frame is then transmitted in the specified VLAN.

2. MAC address–based assignment: assigns VLANs based on the source MAC addresses of frames.

- A network administrator preconfigures the mapping between MAC addresses and VLAN IDs. After receiving an untagged frame, a switch tags the frame with the VLAN ID mapping the source MAC address of the frame. The frame is then transmitted in the specified VLAN.

3. IP subnet–based assignment: assigns VLANs based on the source IP addresses and subnet masks of frames.

- A network administrator preconfigures the mapping between IP addresses and VLAN IDs. After receiving an untagged frame, a switch tags the frame with the VLAN ID mapping the source IP address of the frame. The frame is then transmitted in the specified VLAN.

4. Protocol-based assignment: assigns VLANs based on the protocol (suite) types and encapsulation formats of frames.

- A network administrator preconfigures the mapping between protocol (suite) types and VLAN IDs. After receiving an untagged frame, a switch tags the frame with the VLAN ID mapping the protocol (suite) type of the frame. The frame is then transmitted in the specified VLAN.

5. Policy-based assignment: assigns VLANs based on a specified policy, which means VLANs are assigned based on a combination of interfaces, MAC addresses, and IP addresses.

- A network administrator preconfigures a policy. After receiving an untagged frame that matches the policy, a switch adds a specified VLAN tag to the frame. The frame is then transmitted in the specified VLAN.

4.3.2.6 Interface-based VLAN Assignment

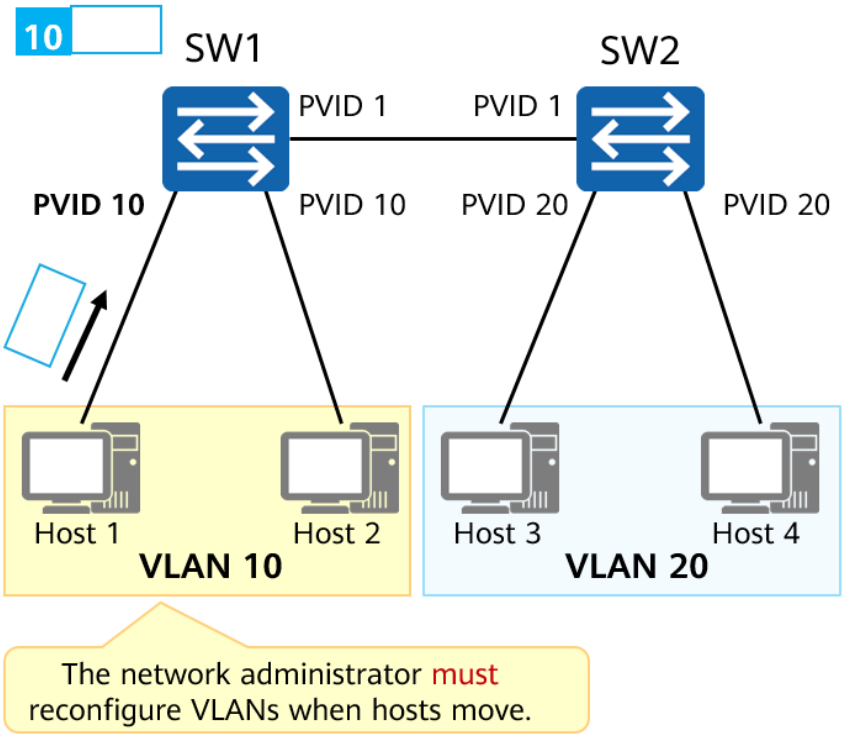


Figure 4-32 Interface-based VLAN assignment

The rule and characteristics of interface-based VLAN assignment are as follows:

- Assignment rule: VLAN IDs are configured on physical interfaces of a switch. All PC-sent untagged frames arriving at a physical interface are assigned to the VLAN corresponding to the PVID configured on the interface.
- Characteristics: This VLAN assignment method is simple, intuitive, and easy to implement. Currently, it is the most widely used VLAN assignment method. When a PC is connected to another switch interface, the frames sent by the PC may be assigned to a different VLAN.
- Port VLAN ID (PVID): default VLAN ID of an interface. The value ranges from 1 to 4094. Each switch interface must be configured with a PVID. All untagged frames arriving at a switch interface are assigned to the VLAN corresponding to the PVID configured on the interface.

The default PVID of Huawei switch interfaces is 1.

4.3.2.7 VLAN Interface Types

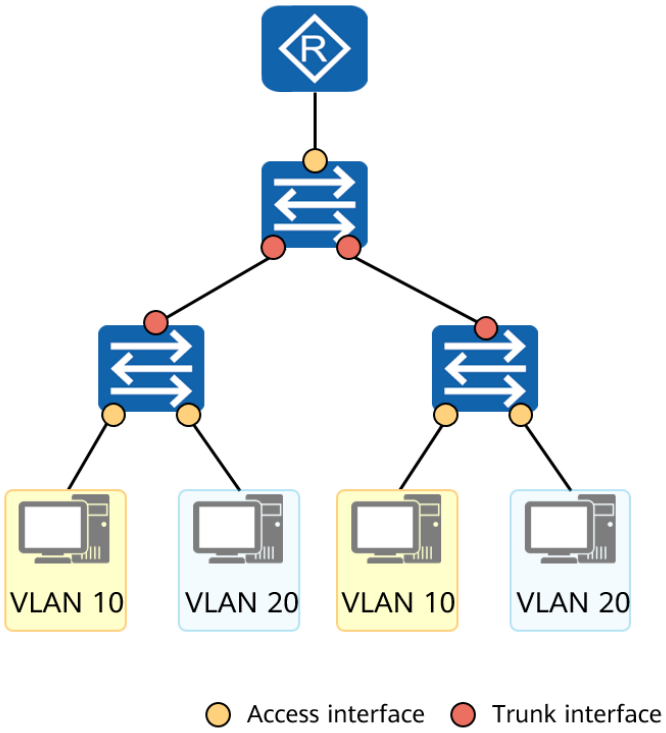


Figure 4-33 VLAN interface types

As shown in Figure 4-33, the interface-based VLAN assignment method varies according to the switch interface type.

- Access interface: An access interface often connects to a terminal (such as a PC or server) that cannot identify VLAN tags, or is used when VLANs do not need to be differentiated. In general, the NICs on such terminals receive and send only untagged frames. An access interface can be added to only one VLAN.
- Trunk interface: A trunk interface allows frames that belong to multiple VLANs to pass through and differentiates the frames using the 802.1Q tag. This type of interface often connects to a switch, router, AP, or voice terminal that can accept and send both tagged and untagged frames.
- Hybrid interface: Similar to a trunk interface, a hybrid interface also allows frames that belong to multiple VLANs to pass through and differentiates the frames using the 802.1Q tag. You can determine whether to allow a hybrid interface to send frames that belong to one or multiple VLANs VLAN-tagged. Therefore, a hybrid interface can connect to a terminal (such as a PC or server) that cannot identify VLAN tags or to a switch, router, AP, or voice terminal that can accept and send both tagged and untagged frames.

By default, hybrid interfaces are used on Huawei devices.

4.3.3 VLAN Basic Configuration

4.3.3.1 Basic VLAN Configuration Commands

4.3.3.1.1 Creating VLANs

```
[Huawei] vlan vlan-id
```

- Create a VLAN and enter the VLAN view, or enter the view of an existing VLAN.
- The value of *vlan-id* is an integer that ranges from 1 to 4094.

```
[Huawei] vlan batch { vlan-id1 [ to vlan-id2 ] }
```

Create VLANs in a batch.

- **batch**: creates VLANs in a batch.
- *vlan-id1*: specifies the start VLAN ID.
- *vlan-id2*: specifies the end VLAN ID.

4.3.3.2 Basic Access Interface Configuration Commands

- Set the interface type.

```
[Huawei-GigabitEthernet0/0/1] port link-type access
```

In the interface view, set the link type of the interface to access.

- Configure the default VLAN of the access interface.

```
[Huawei-GigabitEthernet0/0/1] port default vlan vlan-id
```

In the interface view, configure the default VLAN of the interface and add the interface to the VLAN.

vlan-id: specifies the default VLAN ID. The value is an integer that ranges from 1 to 4094.

4.3.3.3 Basic Trunk Interface Configuration Commands

- Set the interface type.

```
[Huawei-GigabitEthernet0/0/1] port link-type trunk
```

In the interface view, set the link type of the interface to trunk.

- Add the trunk interface to specified VLANs.

```
[Huawei-GigabitEthernet0/0/1] port trunk allow-pass vlan { { vlan-id1 [ to vlan-id2 ] } | all }
```

In the interface view, add the trunk interface to specified VLANs.

- (Optional) Configure the default VLAN of the trunk interface.

```
[Huawei-GigabitEthernet0/0/1] port trunk pvid vlan vlan-id
```

In the interface view, configure the default VLAN of the trunk interface.

4.3.3.4 Basic Hybrid Interface Configuration Commands

- Set the interface type.

```
[Huawei-GigabitEthernet0/0/1] port link-type hybrid
```

In the interface view, set the link type of the interface to hybrid.

- Add the hybrid interface to specified VLANs.

```
[Huawei-GigabitEthernet0/0/1] port hybrid untagged vlan { { vlan-id1 [ to vlan-id2 ] } | all }
```

In the interface view, add the hybrid interface to specified VLANs. Frames that belong to these VLANs then pass through the hybrid interface in untagged mode.

```
[Huawei-GigabitEthernet0/0/1] port hybrid tagged vlan { { vlan-id1 [ to vlan-id2 ] } | all }
```

In the interface view, add the hybrid interface to specified VLANs. Frames that belong to these VLANs then pass through the hybrid interface in tagged mode.

- (Optional) Configure the default VLAN of the hybrid interface.

```
[Huawei-GigabitEthernet0/0/1] port hybrid pvid vlan vlan-id
```

In the interface view, configure the default VLAN of the hybrid interface.

4.4 Routing Basics

4.4.1 Basic Routing Principles

4.4.1.1 Routing

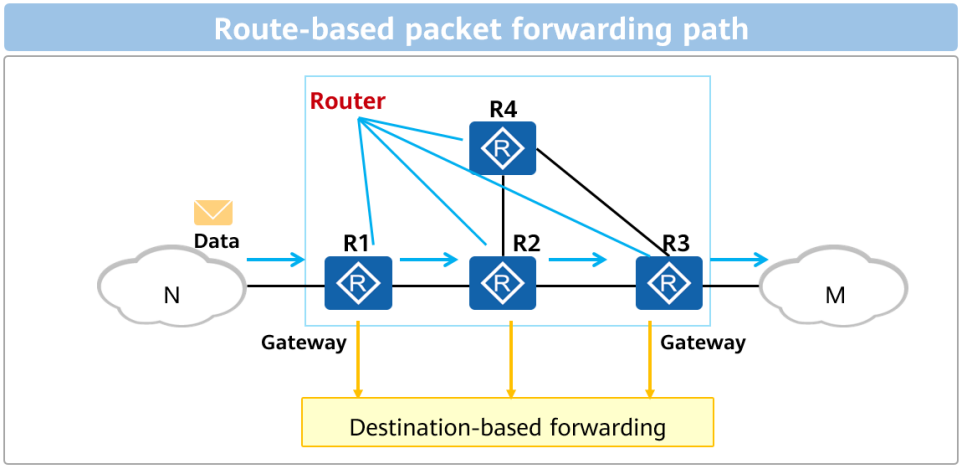


Figure 4-34 Routing

First, we need to understand some routing concepts.

- Routes are the path information that is used to guide packet forwarding.
- A routing device is one that forwards packets to a destination network segment based on routes. The most common routing device is a router.
- A routing device maintains an IP routing table that stores routing information.

As shown in Figure 4-34, a gateway and an intermediate node (a router) select a proper path according to the destination address of a received IP packet, and forward the packet to the next router. The last-hop router on the path performs Layer 2 addressing and forwards the packet to the destination host. This process is called route-based forwarding.

The intermediate node selects the best path from its IP routing table to forward packets.

A routing entry contains a specific outbound interface and next hop, which are used to forward IP packets to the corresponding next-hop device.

4.4.1.2 Routing Information

A route contains the following information:

- Destination network: identifies a destination network segment.
- Mask: identifies a network segment together with a destination IP address.
- Outbound interface: indicates the interface through which a data packet is sent out of the local router.
- Next hop: indicates the next-hop address used by the router to forward the data packet to the destination network segment.

Based on the information contained in a route, a router can forward IP packets to the destination network segment along the corresponding path.

The destination address and mask identify the destination address of an IP packet. After an IP packet matches a specific route, the router determines the forwarding path according to the outbound interface and next hop of the route.

4.4.1.3 Routing Table

Destination/ Mask	Next Hop	Outbound Interface
11.0.0.0/8	2.2.2.2	GE0/0
13.0.0.0/8	3.3.3.2	GE0/1
14.0.0.0/8	1.1.1.2	GE0/2
...		
1.1.1.0/30	1.1.1.1	GE0/2
1.1.1.1/32	127.0.0.1	GE0/2

Figure 4-35 Routing table

A router forwards packets based on its IP routing table that contains many routing entries.

An IP routing table contains only optimal routes. A router manages routing information by managing the routing entries in its IP routing table.

4.4.1.4 Checking the IP Routing Table

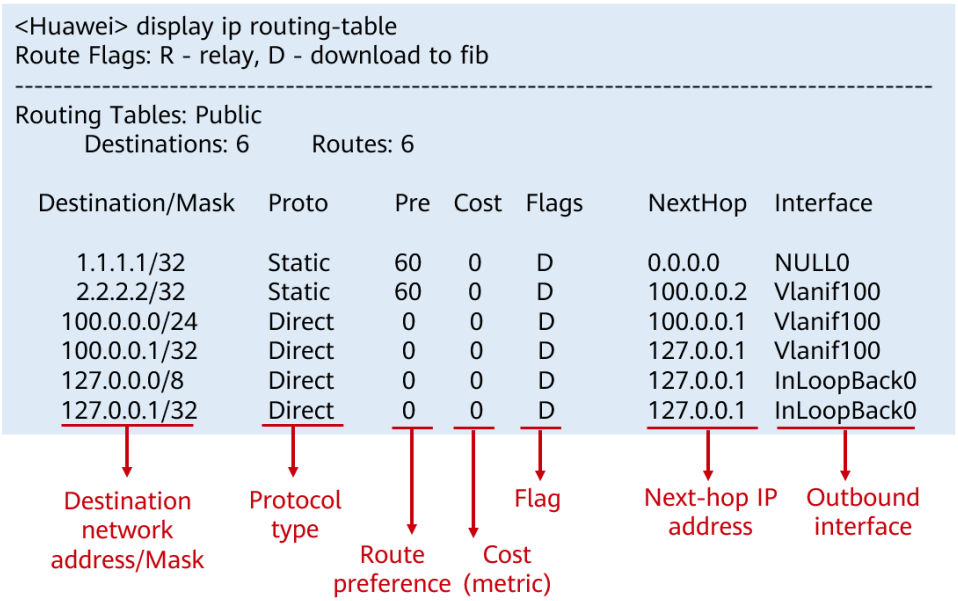
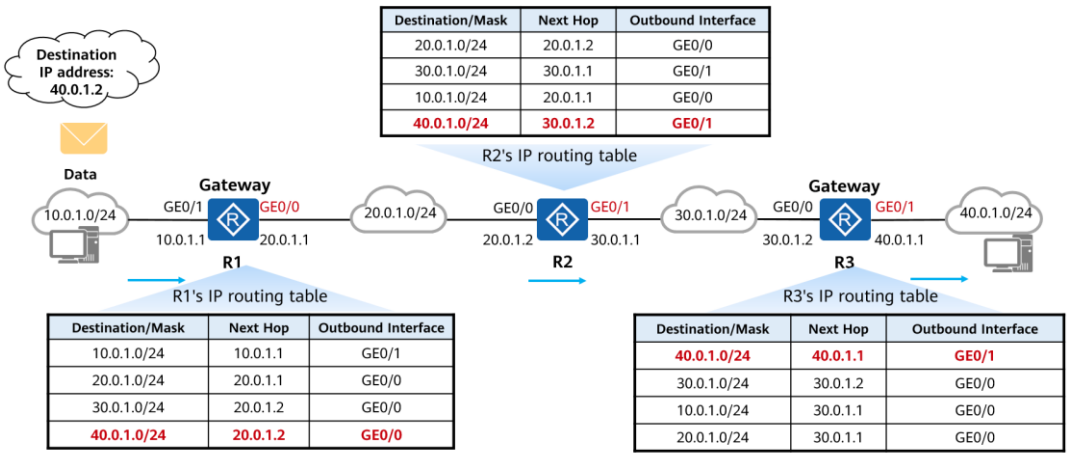


Figure 4-36 IP routing table

Figure 4-36 shows the IP routing table on a router.

- Destination/Mask: indicates the destination network address and mask of a specific route. The network segment address of a destination host or router is obtained through the AND operation on the destination address and mask. For example, if the destination address is 1.1.1.1 and the mask is 255.255.255.0, the IP address of the network segment to which the host or router belongs is 1.1.1.0.
- Proto (Protocol): indicates the protocol type of the route, that is, the protocol through which a router learns the route.
- Pre (Preference): indicates the routing protocol preference of the route. There may be multiple routes to the same destination, which have different next hops and outbound interfaces. These routes may be discovered by different routing protocols or manually configured. A router selects the route with the highest preference (with the lowest preference value) as the optimal route.
- Cost: indicates the cost of the route. When multiple routes to the same destination have the same preference, the route with the lowest cost is selected as the optimal route.
- NextHop: indicates the local router's next-hop address of the route to the destination network. This field specifies the next-hop device to which packets are forwarded.
- Interface: indicates the outbound interface of the route. This field specifies the local interface through which the local router forwards packets.

4.4.1.5 Route-based Forwarding Process



Destination/Mask	Next Hop	Outbound Interface
20.0.1.0/24	20.0.1.2	GE0/0
30.0.1.0/24	30.0.1.1	GE0/1
10.0.1.0/24	20.0.1.1	GE0/0
40.0.1.0/24	**30.0.1.2**	**GE0/1**

Destination/Mask	Next Hop	Outbound Interface
10.0.1.0/24	10.0.1.1	GE0/1
20.0.1.0/24	20.0.1.1	GE0/0
30.0.1.0/24	20.0.1.2	GE0/0
40.0.1.0/24	**20.0.1.2**	**GE0/0**

Destination/Mask	Next Hop	Outbound Interface
40.0.1.0/24	**40.0.1.1**	**GE0/1**
30.0.1.0/24	30.0.1.2	GE0/0
10.0.1.0/24	30.0.1.1	GE0/0
20.0.1.0/24	30.0.1.1	GE0/0

Figure 4-37 Route-based forwarding process

As shown in Figure 4-37, the IP packets from 10.0.1.0/24 need to reach 40.0.1.0/24. These packets arrive at the gateway R1, which then searches its IP routing table for the next hop and outbound interface and forwards the packets to R2. After the packets reach R2, R2 forwards the packets to R3 by searching its IP routing table. After receiving the packets, R3 searches its IP routing table, finding that the destination IP address of the packets belongs to the network segment where a local interface resides. Therefore, R3 directly forwards the packets to the destination network segment 40.0.1.0/24.

4.4.2 Static and Default Routes

4.4.2.1 Introduction to Static Routes

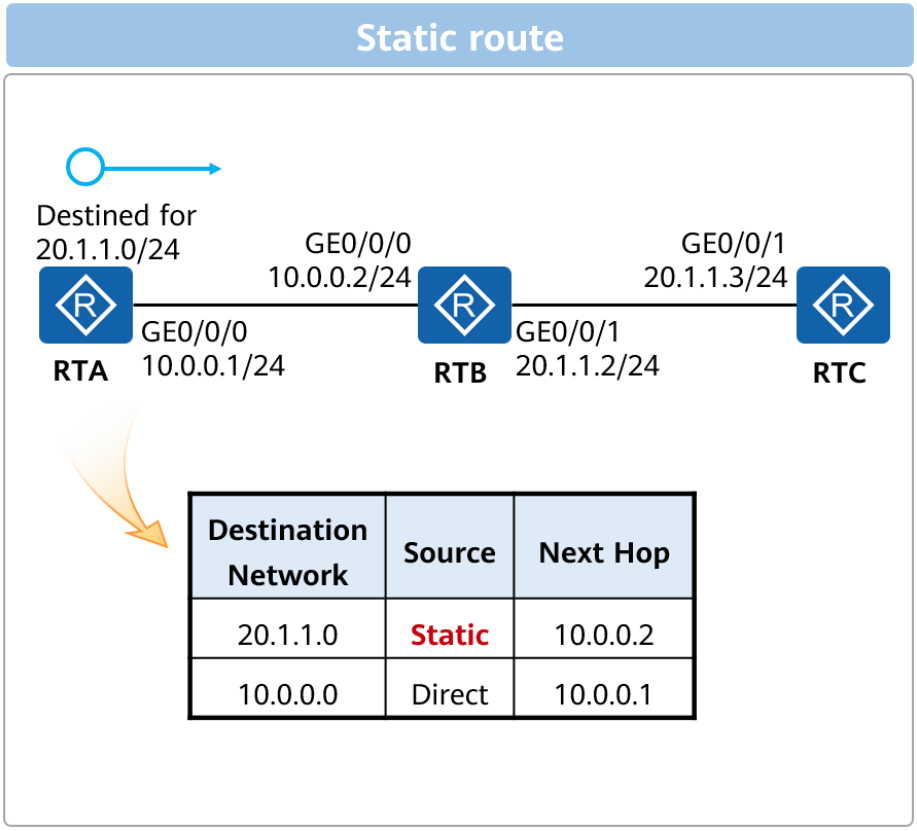


Figure 4-38 Static route

Static routes are manually configured by network administrators, have low system requirements, and apply to simple, stable, and small networks.

However, static routes cannot automatically adapt to network topology changes and so require manual intervention.

Packets destined for 20.1.1.0/24 do not match the direct route in RTA's IP routing table. In this case, a static route needs to be manually configured so that the packets sent from RTA to 20.1.1.0/24 can be forwarded to the next hop 10.0.0.2.

4.4.2.2 Static Route Configuration Example

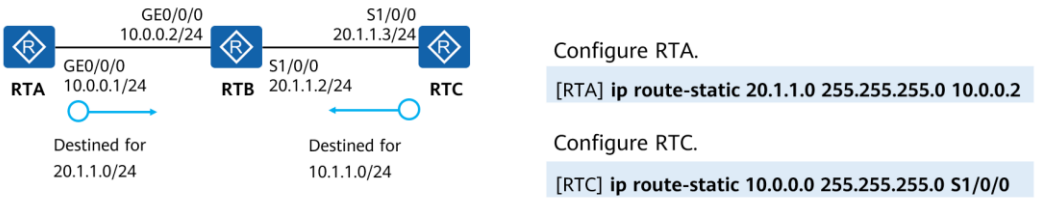


Figure 4-39 Static route configuration example

Figure 4-39 shows how to configure static routes on RTA and RTC for communication between 10.0.0.0/24 and 20.1.1.0/24.

Packets are forwarded hop by hop. Therefore, all the routers along the path from the source to the destination must have routes destined for the destination.

Data communication is bidirectional. Therefore, both forward and return routes must be available.

4.4.2.3 Default Route

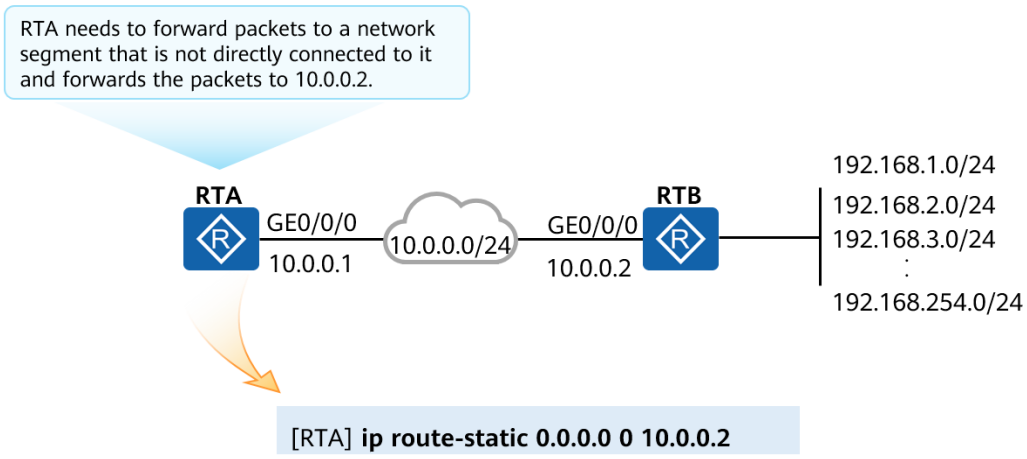


Figure 4-40 Default route

Default routes are used only when packets to be forwarded do not match any routing entry in an IP routing table.

In an IP routing table, a default route is the route to network 0.0.0.0 (with the mask 0.0.0.0), namely, 0.0.0.0/0.

4.4.2.4 Application Scenarios of Default Routes

Default routes are typically used at the egress of an enterprise network. For example, you can configure a default route on an egress device so that the device forwards IP packets destined for any address on the Internet.

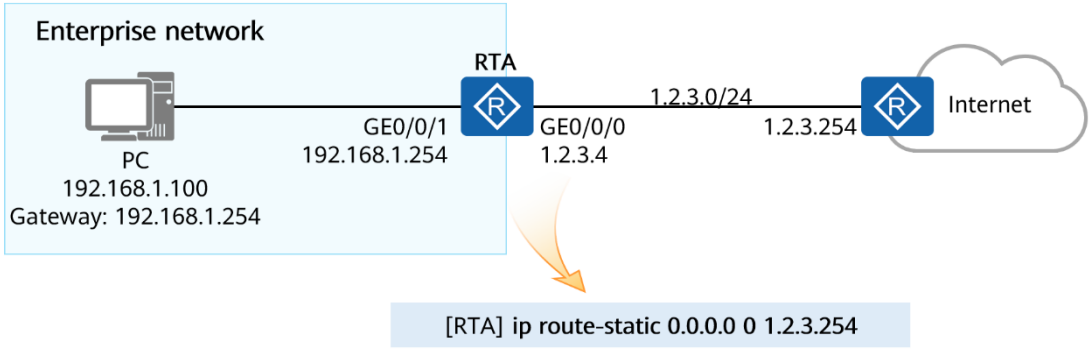


Figure 4-41 Application scenarios of default routes

4.5 Quiz

After you run the **display ip interface brief** on an existing VLANIF interface on a switch, the command output shows that the physical status and protocol status of the VLANIF interface are both Down. Why does this occur?

5 Operating System Basics

The operating system (OS) plays an important role in the interaction between the user and the computer. But what exactly is an OS? What are the types of OSs? What are the basic commands of the Linux system? This chapter explores these questions, and more, to help you better understand OSs.

5.1 Operating System Basics

5.1.1 Definition

5.1.1.1 Operating System Definition and Functions

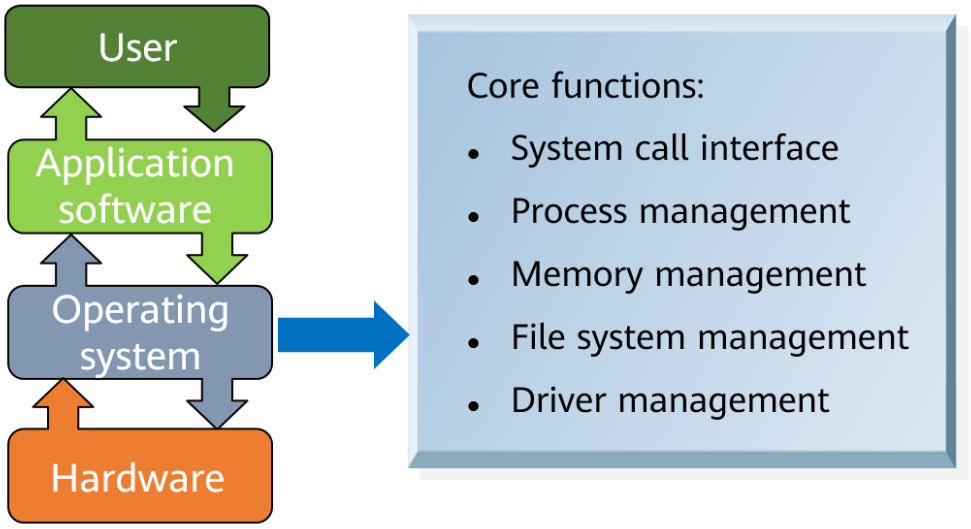


Figure 5-1 Operating system

An operating system (OS) is a special computer program that controls the computer, connects the computer and the user, and coordinates and manages hardware resources, including the CPU, drive, memory, and printer. These resources are required for running programs.

Mainstream OSs:

From the perspective of application field, OSs are classified into the following types:

- Desktop OS, server OS, host OS, and embedded OS.

Based on whether an OS is open source, it is classified as:

- Open source OS (Linux and Unix) or close source OS (Windows and Mac OS).

5.1.2 Components of an OS

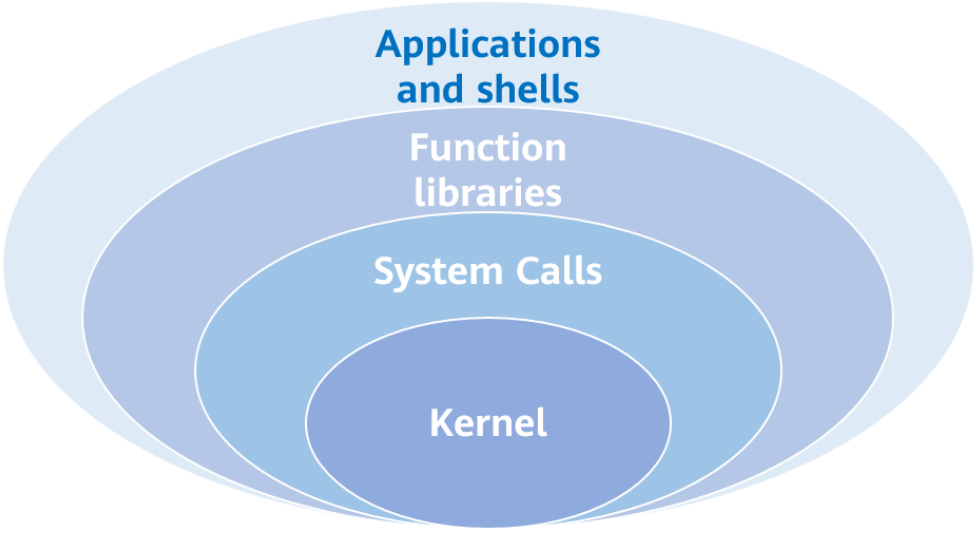


Figure 5-2 Components of an OS

From the perspective of users, an OS consists of a kernel and various applications, that is, the kernel space and user space.

The user space is where upper-layer applications run.

The kernel is essentially a software program used to manage computer hardware resources and provide a system call interface to run upper-layer application programs.

As shown in Figure 5-2, an OS consists of the kernel, system calls, function libraries, and applications, such as the shell.

1. System calls: The execution of applications depends on resources provided by the kernel, including the CPU, storage, and I/O resources. To enable upper-layer applications to access these resources, the kernel must provide an access interface, that is, the system call interface.
2. Library functions: System calls are encapsulated as library functions to provide simple service logic interfaces to users. Simple access to system resources can be completed using system calls. Library functions allow for complex access to system resources.
3. Shell: A shell is a special application program, which is also called the command line interface. It is a command interpreter in essence. It can execute texts (scripts) that comply with the shell syntax. Some shell script statements encapsulate system calls for convenient use.
4. Kernel: The kernel controls hardware resources, manages OS resources, and provides a system call interface for applications.

- Process scheduling and management: The kernel creates and destroys processes and handles their input and output.
- Memory management: The kernel creates a virtual address space for all processes based on limited available resources.

- File system management: Linux is based on the concept of file system to a large extent. Almost anything in Linux can be seen as a file. The kernel builds a structured file system on top of unstructured hardware.
- Device driver management: Drivers of all peripherals in the system, such as hard drives, keyboards, and tape drives, are embedded in the kernel.
- Network resource management: All routing and address resolution operations are performed in the kernel.

Generally speaking, user-mode applications can access kernel-mode resources using the system calls, shell scripts, and library functions.

5.1.3 Different Types of OSs

5.1.3.1 Common Server OSs

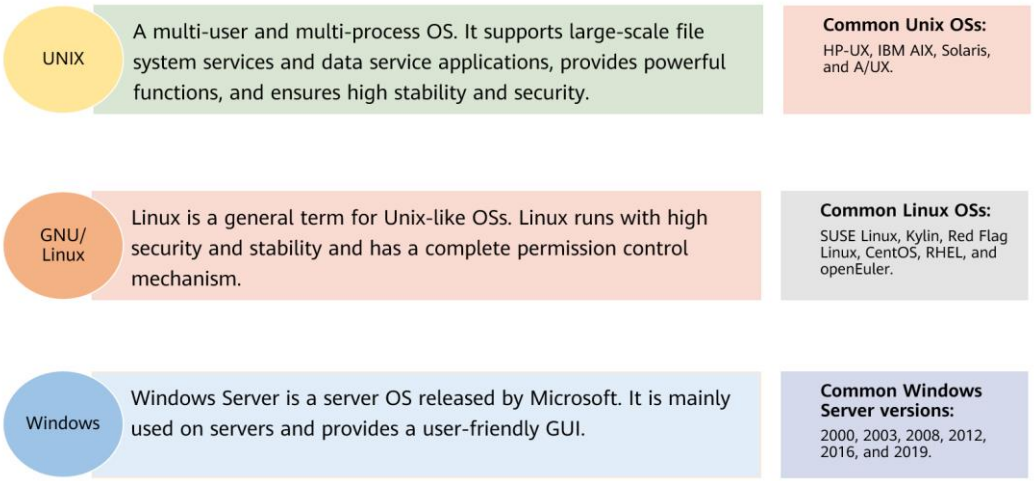


Figure 5-3 Common server OSs

Figure 5-3 shows three types of OSs commonly used on servers.

There is a history behind Linux and Unix:

- Linux is a Unix-like OS with optimized functions and user experience. Linux mimics Unix in terms of appearance and interaction.
- The Linux kernel was initially written by Linus Torvalds for a hobby when he was studying at the University of Helsinki. Frustrated by MINIX, a Unix-like OS for educational purposes, he decided to develop his own OS. The first version was released in September 1991 with only 10,000 lines of code.
- Unix systems are usually compatible only with specific hardware. This means that most Unix systems such as AIX and HP-UX cannot be installed on x86 servers or PCs. On the contrary, Linux can run on various hardware platforms.
- Unix is commercial software, while Linux is open source and free of charge.
- The following describes the Linux OS.

5.2 Linux Basics

5.2.1 Introduction to Linux

5.2.1.1 Features of Linux

Linux is a popular multitasking and multi-user OS with the following features:

- Multitasking: Linux is a multitasking operating system that allows multiple tasks to run at the same time. DOS is a single-task OS and cannot run multiple tasks at the same time. When the system executes multiple tasks, the CPU executes only one task at a time. Linux divides the CPU time into time slices and allocates them to multiple processes. The CPU runs so quickly that all programs (processes) seem to be running at the same time from the user's perspective.
- Multi-user: Linux is a multi-user OS that allows multiple users to use it at the same time. In Linux, each user runs their own or public programs as if they had a separate machine. DOS is a single-user OS and allows only one user to use it at a time.
- Pipeline: Linux allows the output of a program to be used as the input of the next program. Multiple programs are chained together as a pipeline. By combining simple tasks, you can complete complex tasks, improving the operation convenience. Later versions of DOS learned from Linux and implemented this mechanism.
- Powerful shells: Shells are the command interpreters of Linux. Linux provides multiple powerful shells, each of which is an interpreted high-level language. Users can create numerous commands through programming.
- Security protection mechanism: Linux provides a powerful security protection mechanism to prevent unauthorized access to the system and its data.
- POSIX 1.0 compatibility: The Portable Operating System Interface (POSIX) is a family of standards specified by IEEE. POSIX defines the application programming interfaces (APIs) of software runs on Unix. The family of POSIX standards is formally designated as IEEE 1003 and the ISO/IEC standard number is ISO/IEC 9945. The name of POSIX consists of the abbreviation of Portable Operating System Interface and an X that indicates the inheritance of Unix APIs.

5.2.1.2 Linux File Directory Structure

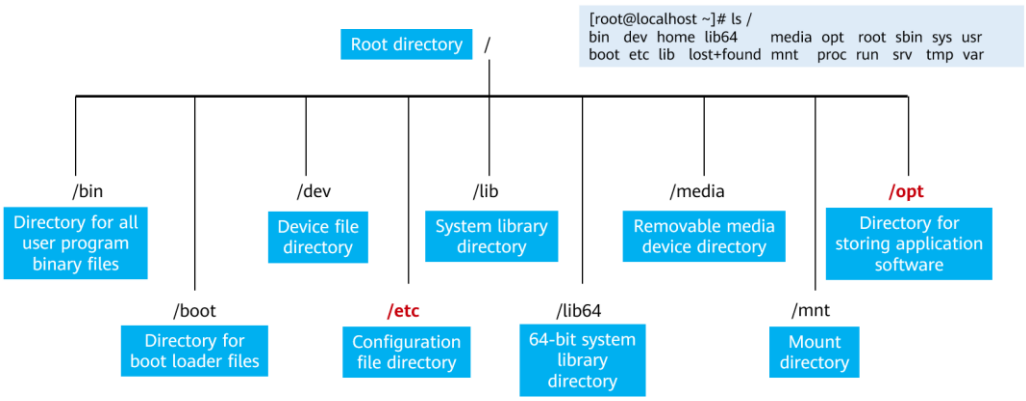


Figure 5-4 Linux file directory structure

The core philosophy of Linux is "everything is a file", which means that all files, including directories, character devices, block devices, sockets, printers, processes, threads, and pipes, can be operated, read, and written by using functions such as **fopen()**, **fclose()**, **fwrite()**, and **fread()**.

Figure 5-4 shows the file directory structure of Linux, which is a tree structure. **/** indicates the root directory.

After logging in to the system, enter the **ls /** command in the current command window. The command output similar to Figure 5-4 is displayed. The directories are described as follows:

- **/bin**: short for binary. This directory stores the frequently used commands.
- **/boot**: stores some core files used for booting the Linux OS, including some links and images.
- **/dev**: short for device. This directory stores peripheral device files of Linux. The method of accessing devices on Linux is the same as that of accessing files.
- **/etc**: stores all configuration files and subdirectories required for system management.
- **/lib**: stores basic shared libraries of the system. A library functions similarly to a dynamic link library (DLL) file on Windows. Almost all applications need to use these shared libraries.
- **/media**: Some devices, such as USB flash drives and CD-ROM drives, are automatically identified and mounted to this directory by the Linux system.
- **/mnt**: temporary mount point for other file systems. You can mount the CD-ROM drive to **/mnt** and then go to this directory to view the contents in the CD-ROM.
- **/opt**: stores additional software installed on the host. For example, if you install an Oracle database, you can save the installation package to this directory. By default, this directory is empty.

5.2.2 Introduction to openEuler

5.2.2.1 Background of openEuler

EulerOS is a server OS that runs on the Linux kernel and supports processors of multiple architectures, such as x86 and ARM. It is ideal for database, big data, cloud computing, and artificial intelligence (AI) scenarios.

Over the past decade, EulerOS has interconnected with various Huawei products and solutions. It is respected for its security, stability, and efficiency.

Cloud computing, in addition to Kunpeng processors, has sparked the growth of EulerOS to become the most powerful software infrastructure in the Kunpeng ecosystem.

To develop the Kunpeng ecosystem and build prosperity of the computing industry in China and around the world, the open source version of EulerOS was officially released as openEuler at the end of 2019.

5.2.2.2 Introduction to openEuler

openEuler is a free open source Linux distribution that supports multiple processor architectures including x86, ARM, and RISC-V. All developers, enterprises, and business organizations can simply use the openEuler community version, or use it to build, develop, and release their own OS versions.

Visit the following two links for more about openEuler. The first is the official website, and the second is the Git repository.

https://openeuler.org/

https://gitee.com/openeuler/

5.2.2.3 Relationship Between openEuler and Mainstream OSs

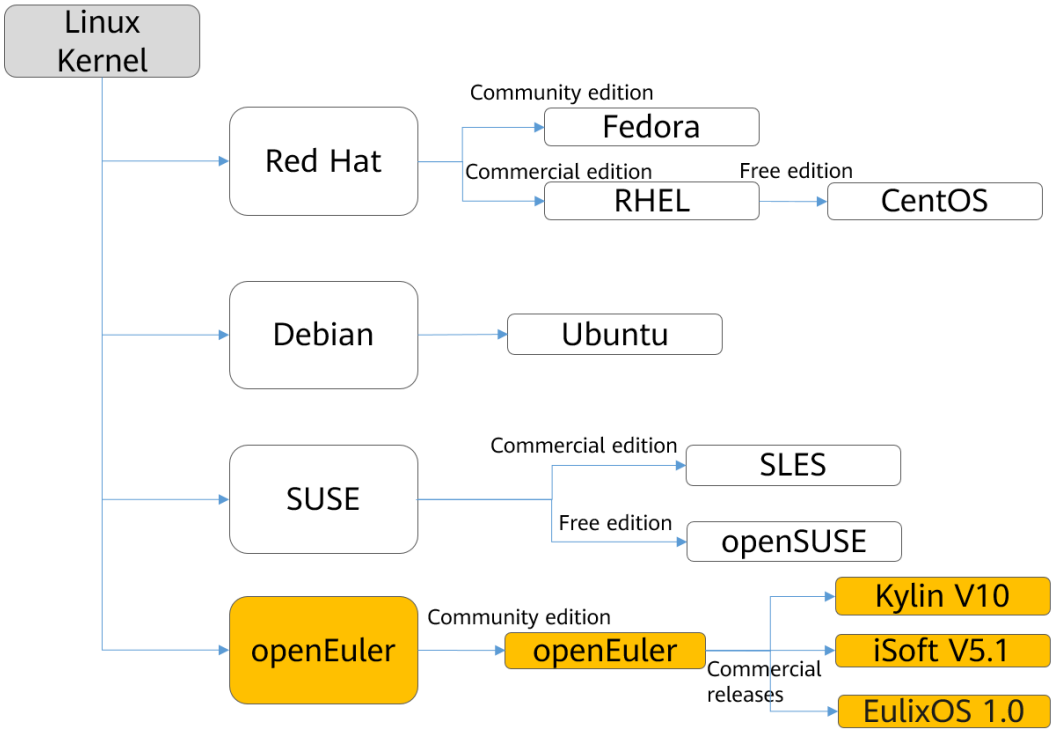


Figure 5-5 Relationship between openEuler and mainstream OSs

As shown in Figure 5-5, the upstream community of openEuler, SUSE, Debian, and Red Hat is the kernel community www.kernel.org.

The openEuler community releases free long-term support (LTS) versions, enabling operating system vendors (OSVs) such as Kylinsoft, iSoft, Sinosoft, and GreatDB to develop commercial releases.

5.2.3 Introduction to File Systems on openEuler

5.2.3.1 File System Overview

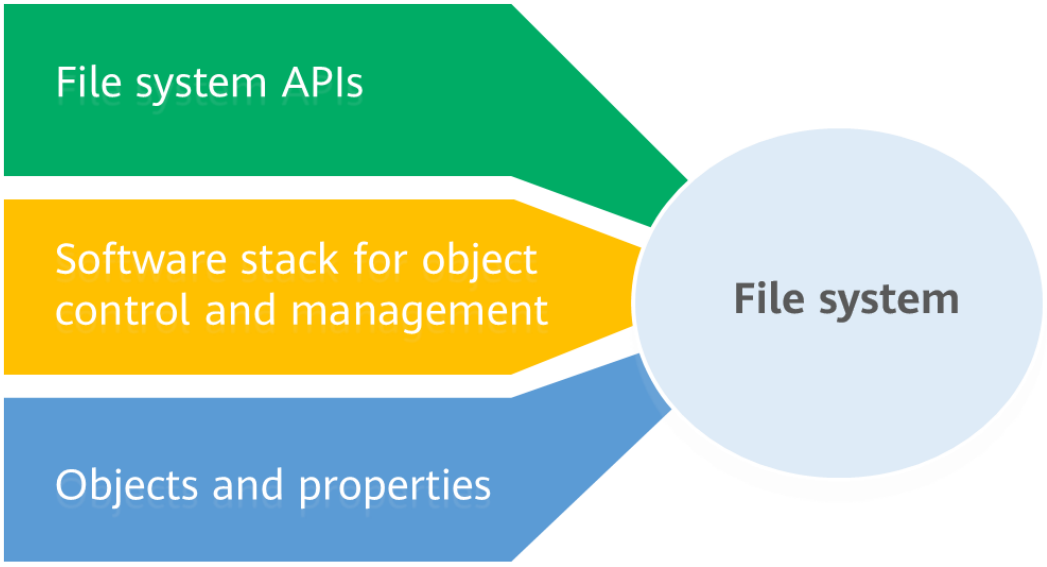


Figure 5-6 File systems

A file system is a method and a data structure used by an OS to identify files on a storage device or a partition, that is, a method of organizing files on a storage device.

In an OS, a software structure that manages and stores file data is referred to as a file management system, or file system for short.

The file system organizes and allocates the space on file storage devices, stores files, and protects and retrieves the stored files. Specifically, it is responsible for creating files for users, saving, reading, modifying, and dumping files, controlling access to files, and canceling a file that is no longer in use. Functions of a file system include: manages and schedules storage space of a file, and provides the logical structure, physical structure, and storage method of the file; maps file identifiers to actual addresses, controls and accesses files, shares file information, provides reliable file confidentiality and protection measures, and provides file security measures.

5.2.3.2 File Systems on openEuler

The openEuler kernel is derived from Linux. The Linux kernel supports more than 10 types of file systems, such as Btrfs, JFS, ReiserFS, ext, ext2, ext3, ext4, ISO 9660, XFS, Minix, MSDOS, UMSDOS, VFAT, NTFS, HPFS, SMB, SysV and PROC. The following table describes the common file systems.

The default file system on openEuler is ext4.

Common File System	Description
Ext	File system specially designed for Linux. The latest version is ext4.
XFS	A high-performance log file system developed for the IRIX OS by Silicon Graphics in 1993. Later ported to the Linux kernel, it excels in large-file processing and provides smooth data transfer.
VFAT	On Linux, VFAT is the name of the FAT (including FAT16 and FAT32) file systems in DOS and Windows.
ISO 9600	The standard file system for optical disc media. Linux supports this file system, allowing the system to read CD-ROMs and ISO image files, and burn CD-ROMs.

Figure 5-7 Common Linux file systems

5.2.4 Basic openEuler Operations

5.2.4.1 Basic Knowledge of Linux Commands

5.2.4.1.1 Linux GUI and CLI

Before getting started with the Linux commands, let's first look at the differences between the GUI and CLI.

A graphical user interface (GUI) presents all elements as graphical. The mouse is used as the main input tool, and buttons, menus, and dialog boxes are used for interaction, focusing on ease of use.

All elements on a command line interface (CLI) are character-based. The keyboard is used as the input tool to enter commands, options, and parameters for executing programs, achieving high efficiency.

Example:

Start the calculator on the Windows GUI. Choose **Start** > **Programs** > **Windows Accessories** > **Calculator**. In the calculator, click buttons to enter an expression. Similarly, a small keyboard is displayed when a certain program requires you to enter a password, asking you to click the numbers. This method is very user-friendly, and the calculator looks similar to the input device used at bank ATMs all around the world. The difference here is that you click it using a mouse, rather than using your own hands.

On the Linux CLI, enter **bc** to start the calculator. Enter the calculation **1** + **1** and press **Enter**. Result **2** is obtained.

5.2.4.1.2 Why we use CLIs

On Linux, we generally choose the CLI over the GUI for management operations.

The reasons are as follows:

Higher efficiency

- On Linux, it is faster to perform operations on a keyboard than using the mouse.
- A GUI-based operation cannot be repeated, while a CLI script can be used to complete all required tasks, for example, deleting outdated log files.

Lower overhead compared with a GUI

- Running a GUI requires a large amount of system resources. With the CLI, system resources can be released and allocated to other operations.

Sometimes, the only choice

- Most servers choose not to install a GUI.

- Tools for maintaining and managing network devices do not provide a GUI.

5.2.4.1.3 Linux CLI Shortcuts

On Linux, we need to use some shortcuts for higher work efficiency.

Examples:

Tab completion

- Use the **Tab** key to complete a command or file name, which is time-saving and accurate.
- When no command is entered, press **Tab** twice to list all available commands.
- If you have entered a part of the command name or file name, press **Tab** to complete it automatically.

Cursor control

- ↑: Press ↑ several times to display historical commands for quick execution.
- ↓: Press ↓ together with ↑ for choosing a historical command.
- **home**: Press **Home** to move the cursor to the beginning of the line.
- **Ctrl**+**A**: Press **Ctrl**+**A** to move the cursor to the beginning of the line.
- **Ctrl**+**E**: Press **Ctrl**+**E** to move the cursor to the end of the line.
- **Ctrl**+**L**: Press **Ctrl**+**L** to clear the screen.

5.2.4.1.4 Login to Linux

You can log in to Linux in either of the following ways:

Local login

- Use the keyboard, mouse, and monitor of the server to locally log in to the OS.

Remote login

- Using clients such as PuTTY or Xshell to remotely log in to openEuler.

5.2.4.1.5 Changing the Password

Passwords are used to ensure the security of the Linux system and data.

To ensure system security, you should:

- Change the password upon the first login.
- Change passwords periodically.
- Set a complex password, for example, a password containing more than eight characters and at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters.

```
[root@openEuler ~]# passwd # Change the password of the current user.
Changing password for user root.
New password:          # Enter the new password.
Retype new password:       # Enter the new password again.
passwd: all authentication tokens updated successfully
[root@openEuler ~]# passwd test1     # Change the password of a common user as the root user.
Changing password for user test1.
New password:
BAD PASSWORD: The password is a palindrome
```

```
Retype new password:
passwd: all authentication tokens updated successfully.
```

For security purposes, openEuler does not display the password when you enter it and does not use any placeholders to indicate the number of characters.

5.2.4.1.6 Types of Linux Users

On Linux, you need to use a user account to log in to the system. Linux allows multiple users to exist at the same time. On Linux, a UID is used to uniquely identify a user.

Based on different UIDs, there are three types of users in Linux (openEuler is used as an example):

The super user is also called the super administrator. Its UID is 0. The super user has all system permissions. It is similar to the administrator in Windows.

System users, also called program users, have UIDs ranging from 1 to 999. A system user is created by a program and is used to run the program or service.

Common users are generally created by the super administrator (the root user) to perform limited management and maintenance operations on the system. UIDs of common users range from 1000 to 60000.

5.2.4.1.7 Creating and Deleting a Linux User

Common commands for user management:

- Creating a user (common user by default): **useradd username**
- Viewing user information: **id username**
- Switching users: **su - username**
- Deleting a user: **userdel username**

Examples:

```
[root@openEuler ~]# useradd user01      # Create user user01.
[root@openEuler ~]# id user01      # View information about user01 as the root user.
uid=1001(user01) gid=1001(user01) groups=1001(user01)
[root@openEuler ~]# su - user01      # Switch to the user01 user. The command prompt changes to
$.
[user01@openEuler ~]$ id       # Use the id command to view information about the current user by
default.
uid=1001(user01) gid=1001(user01) groups=1001(user01)
[user01@openEuler ~]$ exit # Log out of the current user.
logout
[root@openEuler ~]# userdel user01      # Delete user user01.
```

5.2.4.2 Basic openEuler Commands

5.2.4.2.1 Power Supply Commands: shutdown and reboot

shutdown is used to shut down the computer, which requires root permissions. The **shutdown** command can safely shut down the system. It is dangerous to shut down the Linux system by directly powering off the system. Different from Windows, Linux runs many processes in the background. Therefore, forcible shutdown may cause loss of process data, making the system unstable and even damaging hardware in some systems. If you run the **shutdown** command to shut down the system, the system

administrator notifies all users who have logged in that the system will be shut down and the **login** command will be frozen, prohibiting new user logins.

Parameters of the **shutdown** command:

- **-h**: powers off the computer after it is shut down.
- **-r**: powers on the computer after it is shut down. (This operation is equivalent to restarting the computer.)
- **-p**: explicitly indicates that the system will be shut down and the main power supply will be cut off.

reboot is used to restart the computer, which requires root permissions.

Parameters of the **reboot** command:

- **-w**: writes records to the **/var/log/wtmp** file. It does not restart the system.
- **-d**: does not write records to the **/var/log/wtmp** file.
- **-i**: restarts the system with network settings disabled.

5.2.4.2.2 File Paths

File paths on Linux include absolute paths and relative paths.

- Absolute path: a path starting with the root directory (**/**), for example, **/root/Desktop**.
- Relative path: a path starting from the current path, for example, **./Desktop**. **./** or **.** indicates the current path. **../** or **..** indicates the upper-level directory of the current path.

pwd: Viewing the current path

cd: Switching paths

Syntax: **cd** *[directory]*

- **cd /usr**: goes to the **/usr** directory.
- **cd ..**: goes to the upper-level directory. Double dot indicates the parent directory.
- **cd.**: goes to the current directory.
- **cd**: goes to the home directory by default if no parameter is added.
- **cd -**: goes to the previous directory. This command is used to quickly switch between two directories.
- **cd ~**: goes to the home directory.

5.2.4.2.3 Viewing Files

On Linux, you can view directory files and common files.

- **ls**: Viewing the content of a directory
- **cat**, **tail**, or **head**: Viewing the content of a common file

The **ls** command is used to view contents of a directory.

- **-a**: lists all files including hidden files.
- **-l**: displays file details in long format.
- **-R**: lists files in the subdirectories recursively.
- **-t**: lists files by modification time.

The **cat** command is used to view contents of a small file. This command displays all lines in a file.

The **tail** command is used to view the last 10 lines of a file by default.

- **-n**: followed by a number, for example, 5, indicating that the last five lines of a file are viewed. You can also enter a number directly without the **-n** option.
- **-f**: dynamically displays file changes. This option is commonly used for viewing log files.

The **head** command is used to view the first 10 lines of a file by default.

The **less** and **more** commands are used to view large files page by page. Enter **q** to exit. Enter a slash (/) and a keyword to search for the keyword in the file.

5.2.4.2.4 Creating Files

Similarly, you can create directory files and common files on Linux.

Examples:

mkdir: Creating directories (folders)

- **-p**: cascades to create multiple directories recursively.

touch: Creating common files

5.2.4.2.5 Copying Files

cp: Copying files or directories

- **-a**: copies the files of a directory while retaining the links and file attributes.
- **-r**: If the source file is a directory, all subdirectories and files in the directories are copied recursively and the attributes are retained.

The **cp** command is used to copy files and directories. You can copy one or more files at a time. Exercise caution when running this command because data loss risks are involved.

Syntax: **cp** *[OPTION]... SOURCE... DIRECTORY*

- **-a**: copies the files of a directory while retaining the links and file attributes.
- **-p**: copies the file content, modification time, and access permissions to the new file.
- **-r**: if the source file is a directory, all subdirectories and files in the directories are copied
- **-l**: creates a hard link of the source file instead of copying it.
- **-s**: creates a soft link of the source file instead of copying it.

cp command examples:

- **cp f1 f2**: copies file f1 and renames it to f2.
- **cp f1 d1/**: copies f1 to the d1 directory without renaming it.
- **cp f1 f2 f3 d1/**: copies multiple files to a directory.
- **cp -i f1 f2**: waits for the user's confirmation before overwriting f2 if f2 already exists.
- **cp -r d1 d2**: copies a directory recursively if the **-r** option is added.
- **cp -rv d1 d2**: displays the copy process if the **-v** option is added.
- **cp -s d1 d2**: creates a soft link d2 of the source file d1 instead of copying it.

- **cp -a f1 f2**: if the **-a** option is added, the attributes of the source file are retained. This option is used to copy block devices, character devices, and named pipes.
- By default, the **cp** command does not ask the user before overwriting files. Therefore, many shells have made **cp** as an alias for **cp -i**. The **-f** option in the **cp** command does not indicate forcible overwriting.

5.2.4.2.6 Moving and Renaming Files

mv: Moving or renaming a file

- The **mv** command is used to move a file or directory. Exercise caution when running this command because data loss risks are involved.
- If the source file and target file are in the same directory, the **mv** command is used to rename the file.

Syntax: **mv** *[option] source_file_or_directory target_file_or_directory*

- **-b**: backs up a file before overwriting it.
- **-f**: forcibly overwrites the target file without asking the user.
- **-i**: overwrites the target file after obtaining the user's consent.
- **-u**: updates the target file only when the source file is newer than the target.

5.2.4.2.7 Deleting Files

rm: Deleting files or directories

- The **rm** command is a high-risk command. No tool can guarantee recovery of files deleted by the **rm** command, which doesn't move a file to a recycle bin like in GUIs. Therefore, you cannot undo the deletion.

Syntax: **rm** *[OPTION] file_or_directory*

- **-f, --force**: ignores the files that do not exist and does not display any message.
- **-i, --interactive**: performs interactive deletion.
- **-r, -R, --recursive**: recursively deletes all directories listed as arguments and their subdirectories.
- **-v, --verbose**: displays the detailed progress.

5.2.4.2.8 Obtaining Help Information About a Command

To navigate the massive number of commands on Linux, you can run the **help** command to obtain help information.

help: Obtaining simple help information about a command

Syntax: *[command]* **--help** or **help** *[command]*.

- **-d**: displays a brief description of the command topic.
- **-s**: displays a brief description of the command syntax.

5.2.4.3 Text Processing on openEuler

5.2.4.3.1 Linux Text Editor - Vim

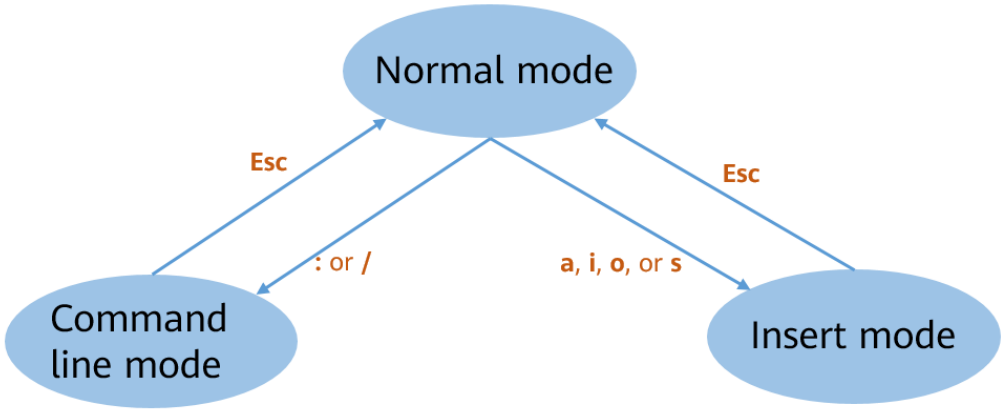


Figure 5-8 Common Operations of Vim

Vim is a customizable text editor derived from Visual Editor (vi) that inherits, improves and adds many features to vi's original base. Vim is not installed on openEuler 20.03 LTS by default. You need to manually install it.

As shown in Figure 5-8, the common operations of Vim are as follows:

- Normal mode: used to copy, paste, and delete text, undo previous operations, and navigate the cursor.
- Insert mode: used to edit and modify text.
- Command line mode: used to save, exit, search for, or replace text. Enter a colon (**:**) to switch to this mode.

5.2.4.3.2 Normal Mode of Vim

By default, Vim begins to run in normal mode after you open a file with the **vim** command.

Example:

```
vim   [options]   [file]...      Edit specified files.
```

5.2.4.3.3 Common Operations in Vim Normal Mode

In the normal mode, some shortcut keys can help improve editing efficiency:

Cursor control

- Arrow keys or **k**, **j**, **h**, and **l** keys move the cursor up, down, left, and right, respectively.
- **0**: moves the cursor to the beginning of the current line.
- **g0**: moves the cursor to the leftmost character of the current line that is on the screen.
- **:***n*: moves the cursor to line *n*.
- **gg**: moves the cursor to the first line of the file.
- **G**: moves the cursor to the last line of the file.

Data operations

- **yy** or **Y**: copies an entire line of text.
- **y***[n]***w**: copies 1 or *n* words.
- **d***[n]***w**: deletes (cuts) 1 or *n* words.
- *[n]* **dd**: deletes (cuts) 1 or *n* lines.

5.2.4.3.4 Insert Mode of Vim

Use the **vim** *filename* command to open a file and enter the normal mode by default. Type **i**, **I**, **a**, **A**, **o**, or **O** to enter the insert mode.

- If the *filename* file exists, the file is opened and the file content is displayed.
- Otherwise, Vim displays **[New File]** at the bottom of the screen and creates the file when saving the file for the first time.

Press **Esc** to exit the insert mode and return to the normal mode.

5.2.4.3.5 Command Line Mode of Vim

You can search for and replace text, and save a file in the command line mode.

1. Search

- **:/***word* or **/***word*: searches for a *word* string after the cursor. Press **n** to continue to search forwards or press **Shift+n** to search backwards.

2. Replace

- **:1,5s/***word1***/***word2***/g**: replaces all occurrences of *word1* in lines 1 to 5 with *word2*. If **g** is not specified, only the first occurrence of *word1* in each line is replaced.
- **%s/***word1***/***word2***/gi**: replaces all occurrences of *word1* with *word2*. **i** ignores the case of matches.

3. Save and Exit

- **:w**: Save the file and do not exit.
- **:wq**: Save the file and exit.
- **:q**: Exit without saving the file.
- **:q!**: Exit forcibly without saving changes to the file.
- **:wq!**: Forcibly save the file and exit.

5.2.4.4 Network Management on openEuler

5.2.4.4.1 Important Network Concepts in openEuler

Before managing the network on openEuler, you need to understand the network concepts in Linux.

Examples:

- Host network device: network adapter on the host
- Interface: Interfaces on devices are created by drivers for the system access.
- Broadcast address: an IP address used to send packets to all hosts on the network segment
- Subnet mask: a number that distinguishes the network address and the host address within an IP address

- Route: next-hop IP address when IP packets are transmitted across network segments
- Link: connection between the device and the network

5.2.4.4.2 Commands for Querying IP Addresses

ip and **ifconfig** commands are used to view IP addresses of the current host.

Viewing information about all network adapters on a host:

```
[root@openEuler ~]# ifconfig –a
[root@openEuler ~]# ip addr show
```

Viewing information about a specified interface on a host:

```
[root@openEuler ~]# ifconfig enp0s3
[root@openEuler ~]# ip addr show enp0s3
```

5.2.4.4.3 Configuring Static IP Addresses

On openEuler, the NIC configuration file and the **nmcli** command are used to configure static IP addresses.

Before configuring the static IP address by modifying the NIC configuration file, find the path to the configuration file, for example, **/etc/sysconfig/network-scripts/ifcfg-enp0s3**. Then, run the **vim** command to open the configuration file for modification.

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
IPADDR=192.168.56.100
NETMASK=255.255.255.0
```

Figure 5-9 NIC configuration example

Figure 5-9 shows the configuration of enp0s3 whose static IP address is 192.168.56.100/24. After the configuration is complete, restart the network as follows:

```
[root@openEuler ~]# nmcli connection reload enp0s3
[root@openEuler ~]# nmcli connection up enp0s3
```

Alternatively, use the nmcli command to configure the static IP address. The **nmcli** command usage is as follows:

- Check network connections of the current host.

```
[root@openEuler ~]# nmcli connection show
NAME     UUID                                  TYPE      DEVICE
enp0s3   3c36b8c2-334b-57c7-91b6-4401f3489c69  ethernet  enp0s3
enp0s8   00cb8299-feb9-55b6-a378-3fdc720e0bc6  ethernet  enp0s8
```

- Configure a static IP address.

```
[root@openEuler ~]# nmcli connection modify enp0s3 ipv4.method manual ipv4.addresses
"10.0.2.10/24" ipv4.gateway "10.0.2.2"
```

- Restart the network.

```
[root@openEuler network-scripts]# nmcli connection reload enp0s3
[root@openEuler network-scripts]# nmcli connection up enp0s3
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/18)
```

- View the IP address.

```
[root@openEuler ~]# ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 08:00:27:7d:e1:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.10/24 brd 10.0.2.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
```

5.2.4.4.4 Route Management and Configuration

On openEuler, the **route** command is used to view, configure, and manage local routes.

In addition to the **route** command, the **ip** command can also be used to manage system routes.

These commands will modify the routing table of the system. When the system is started, the routing table is loaded to the memory and maintained by the kernel.

Run the **route** command to view the routing table.

```
[root@openEuler ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.110.254 0.0.0.0         UG    100    0        0 enp4s0
192.168.110.0   0.0.0.0         255.255.255.0   U     100    0        0 enp4s0
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

Figure 5-10 Linux routing table

5.2.4.4.5 Adding a Route Using the route Command

Add a (temporary) route to a network segment or host on openEuler.

```
route [-f] [-p] [Command [Destination] [mask Netmask] [Gateway] [metric Metric]] [if Interface]]
```

Example:

```
[root@openEuler ~]# route add -net 192.168.101.0 netmask 255.255.255.0 dev enp4s0
[root@openEuler ~]# route add -host 192.168.100.10 dev enp4s0
[root@openEuler ~]# route
Kernel IP routing table
Destination     Gateway     Genmask             Flags   Metric   Ref     Use   Iface
default         _gateway          0.0.0.0             UG      100      0         0     enp4s0
192.168.100.10  0.0.0.0         255.255.255.255   UH      0        0         0     enp4s0
192.168.101.0   0.0.0.0         255.255.255.0     U       0        0         0     enp4s0
192.168.110.0   0.0.0.0         255.255.255.0     U       100      0         0     enp4s0
```

```
192.168.122.0     0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

In the commands:

```
route add -net 192.168.101.0 netmask 255.255.255.0 dev enp3s0
```

Adds a route to the 192.168.101.0/24 segment through the enp3s0 device.

```
route add -host 192.168.101.100 dev enp3s0
```

Adds a route to the 192.168.101.100 host through the enp3s0 device.

The output of the **route** command shows that routes to hosts have a higher priority than routes to network segments.

5.2.4.4.6 Deleting a Route Using the route Command

Use the **route del** command to delete a route to a network segment or host.

Syntax:

```
route del [-net|-host] [netmask Nm] [gw Gw] [[dev] If]
```

Example:

```
[root@openEuler ~]# route del -host 192.168.100.10 dev enp4s0
[root@openEuler ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp4s0
192.168.101.0   0.0.0.0         255.255.255.0   U     0      0        0 enp4s0
192.168.110.0   0.0.0.0         255.255.255.0   U     100    0        0 enp4s0
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

In the commands:

```
route del -net 192.168.101.0 netmask 255.255.255.0 dev enp3s0
```

Deletes the route to the 192.168.101.0/24 segment. To delete a route to a network segment, the network segment and subnet mask parameters are mandatory, while the device parameter is optional.

```
route del -host 192.168.101.100 dev enp3s0
```

Deletes the route to the 192.168.101.100 host. The device parameter is optional.

To delete routes in the route file, use the vi editor to edit the file and restart the network.

5.2.4.4.7 Host Name

A host name identifies a device in a local area network (LAN).

The device can be a physical or virtual machine.

The host name is stored in the **/etc/hostname** file.

```
[root@openEuler ~]# cat /etc/hostname
openEuler
```

5.2.4.4.8 Setting the Host Name

On openEuler, you can change the host name using any of the following methods:

- Setting a temporary host name: **hostname new-name**
- Setting a permanent host name: **hostnamectl set-hostname new-name**
- Setting a host name by modifying the file: write **new-name** to the **/etc/hostname** file.

Examples:

```
[root@openEuler ~]# hostname
openEuler
[root@openEuler ~]# hostname huawei
[root@openEuler ~]# hostname
huawei
[root@openEuler ~]# hostnamectl set-hostname openEuler01
[root@openEuler ~]# hostname
openEuler01
[root@openEuler ~]# echo "HCIA-openEuler" > /etc/hostname
```

Note that the host name does not take effect immediately after being set. To make the setting take effect, log in again or run the **source .bashrc** command. Run the **hostname** command to view the host name of the current system.

5.2.4.4.9 Introduction to the hosts File

Hosts in a LAN can be accessed through IP addresses. IP addresses are difficult to remember when a large number of hosts exist in the LAN. Therefore, we want to access the hosts directly through their host names.

In this case, the hosts can be located using a table that records the mapping between host names and IP addresses. This table is the **hosts** file. The **hosts** file is a system file without a file name extension. Its basic function is to establish a "database" of frequently used domain names and their corresponding IP addresses.

When a user enters a website URL in the web browser, the system searches for the corresponding IP address in the **hosts** file. Once the IP address is found, the system opens the corresponding web page.

If the URL is not found, the system sends the URL to the DNS server for IP address resolution.

Run the **cat /etc/hosts** command to view the **hosts** file:

```
[root@openEuler ~]# cat /etc/hosts
127.0.0.1       localhost    localhost.localdomain localhost4 localhost4.localdomain4
::1                   localhost    localhost.localdomain localhost6 localhost4.localdomain6
```

5.2.4.4.10 Modifying the hosts File

You can edit the **hosts** file in the following format:

```
# Ip domain.com
192.168.10.20       www.example.com
```

To delete an entry, add **#** to comment it out. For example:

```
#ip domain.com
#192.168.10.20       www.example.com
```

5.3 Quiz

On Linux, how to view the updated contents of a log file in real time?

6 Virtualization

Virtualization is the foundation of cloud computing, so what is virtualization? What is the essence of virtualization? What are mainstream virtualization technologies? This course will answer these questions and give you a brief introduction to virtualization.

6.1 Overview

6.1.1 Virtualization

6.1.1.1 What Is Virtualization?

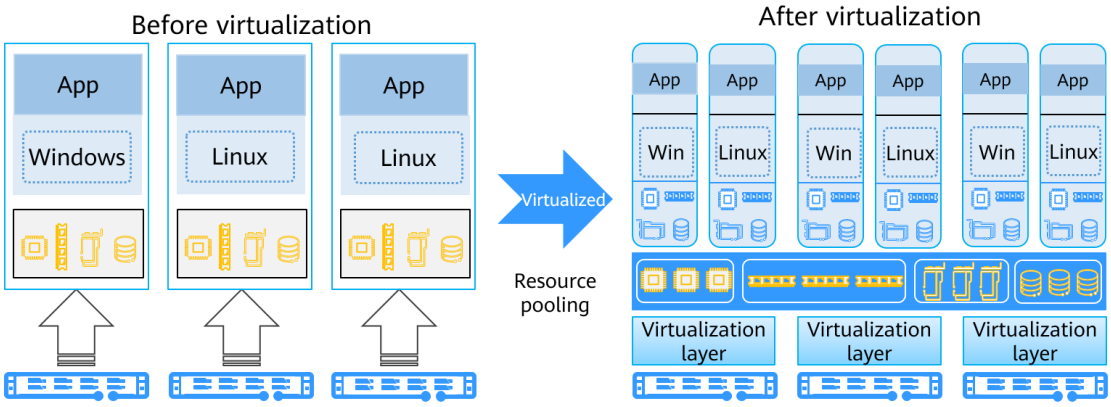


Figure 6-1 What is virtualization?

The basic concepts of virtualization have been described in section 1.2.2.1. Details are not described herein again.

6.1.1.2 Important Concepts of Virtualization

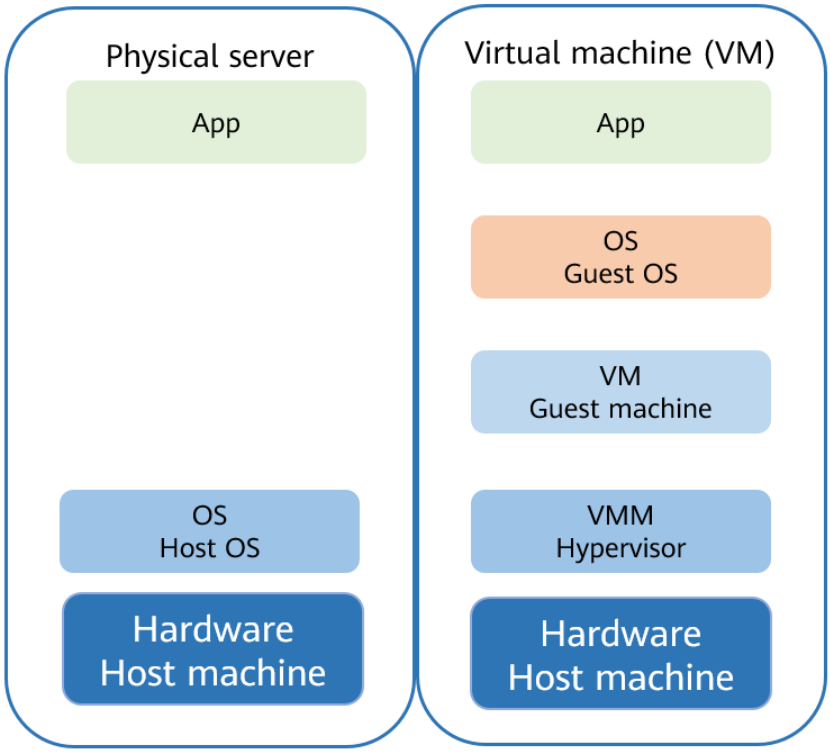


Figure 6-2 Important concepts of virtualization

As shown in Figure 6-2, there is a hypervisor, also called a virtual machine monitor (VMM), is deployed on a virtualized host machine. A hypervisor is a software layer running between physical servers and operating systems (OSs). It allows multiple OSs and applications to share hardware. In short, a host can virtualize hardware with a hypervisor, then a VM (guest machine) can be created from a virtualized resource, and an OS (guest OS) needs to be installed on the created VM.

6.1.1.3 Virtualization Types

Type	Description
Full virtualization	The VMM virtualizes the CPU, memory, and device input/output (I/O) without modifying the guest OS and hardware. Full virtualization gives you excellent compatibility, but increases the load on the host machine's CPU.
Paravirtualization	The VMM virtualizes CPU and memory and the guest OS virtualizes device I/O. The guest OS needs to be modified to coordinate with the VMM. Paravirtualization provides high performance but poor compatibility.
Hardware-assisted virtualization	Efficient full virtualization is realized with the help of hardware. Compatibility is good, and guest OSs do not need to be modified. This type of virtualization has been slowly eliminating differences between different software virtualization.

Figure 6-3 Virtualization types

Virtualization types include full virtualization (using binary translation), paravirtualization (OS-assisted), and hardware-assisted virtualization. For details about these virtualization types, see section 6.1.1.6 CPU Virtualization.

6.1.1.4 Virtualization Characteristics

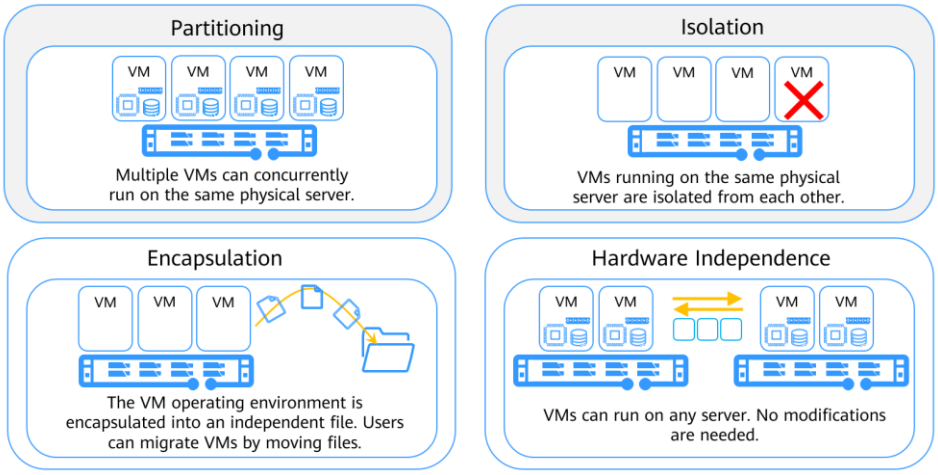


Figure 6-4 Virtualization characteristics

As shown in Figure 6-4, virtualization features partitioning, isolation, encapsulation, and hardware independence.

Partitioning: The virtualization layer allocates physical server resources to multiple VMs on the same physical server. These VMs have independent OSs which can be the same or different, so applications applicable for any OS can run on the physical server. Each OS gains access only to its own virtual hardware, such as the virtual network interface card (NIC), virtual CPUs, and virtual memory, provided by the virtualization layer, so that it misunderstands that it runs on its own dedicated server. Partitioning has the following advantages:

- Resource quotas are allocated to each partition to prevent resource overuse by virtualization.
- Each VM has an independent OS.

Isolation: VMs created in a partition are logically isolated from each other. Isolation has the following advantages:

- Even if one VM crashes or fails due to an OS failure, application crash, or driver failure, other VMs can still run properly.
- If one VM is infected with worms or viruses, other VMs will not be affected as if each VM runs on an independent physical machine.

Through isolation, resources can be controlled to provide performance isolation. That is, the minimum and maximum resources are specified for each VM to ensure that a VM does not occupy all resources and other VMs in the same system have no available resources. Multiple loads, applications, or OSs can concurrently run on a single physical machine, preventing problems (such as application conflicts and DLL conflicts) mentioned in the limitations of traditional x86 servers.

Encapsulation: All VM data including the hardware configuration, BIOS configuration, memory status, disk status, and CPU status is stored into a group of files that are independent of physical hardware. You can copy, save, and move a VM by copying only a few files. Let's use VMware Workstation as an example. You can copy a set of VM files to another computer where VMware Workstation is installed and restart the VM. Of all virtualization characteristics, encapsulation is the most important feature for VM migration. A VM becomes a hardware-independent file and then it can have features such as migration and hot swap, which are closely related to its encapsulation feature.

Hardware independence: After a VM is encapsulated into a group of independent files, the VM is completely independent from its physical hardware. You can migrate the VM by copying its device file, configuration file, or disk file to another host. The physical hardware devices are shielded by the VMM running on it. The VM running on the VMM only needs to check whether the same VMM exists on the destination host, regardless of the physical hardware specifications and configurations. This is similar to editing a Word file by using Office 2007 on computer A that runs a Windows 7 system and then copying the Word file to computer B that runs a Windows 10 system. You only need to check whether Office 2007 is installed on computer B and do not need to check the CPU model or memory size of the computer.

6.1.1.5 Advantages of Virtualization

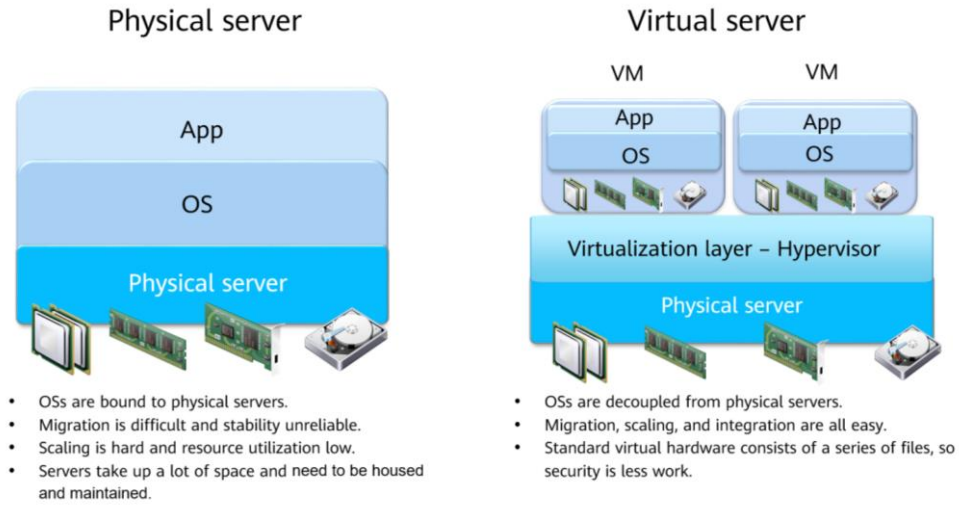


Figure 6-5 Advantages of virtualization

As shown in Figure 6-5, physical servers have many limitations when they are used as IT infrastructure. For example, OSs are bound to physical servers; when required, service migration is difficult and the stability is unreliable; resource scaling is hard, and overall resource utilization low; physical servers take up a lot of space and need to be housed and maintained.

After virtualization, the preceding problems are basically solved. The hypervisor (virtualization layer) decouples OSs from physical servers. Based on the virtualization granularity, services can be easily migrated and expanded, and resources can be easily integrated. In addition, standard virtual hardware consists of a series of files, so security is less work.

In cloud computing, virtualization includes CPU virtualization, memory virtualization, and I/O virtualization. Next, let's learn about each of these three virtualization.

6.1.1.6 CPU Virtualization

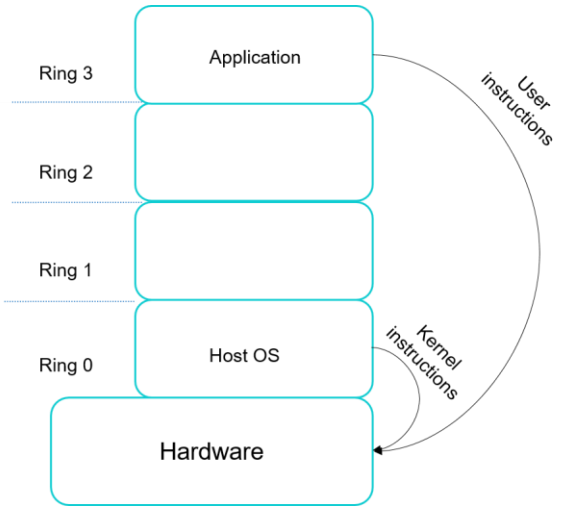


Figure 6-6 CPU hierarchical protection domain

Before CPU virtualization, let's briefly introduce the hierarchical protection domain of the CPU. In this protection mode, there are 4 privilege levels ranging from Ring 0 to Ring 3. Among them, Ring 0 is the most privileged, and Ring 3 is the least privileged. Ring 0 has direct access to the hardware. Generally, only OSs and drivers have this privilege. Ring 3 is the least privileged ring, so all applications can have this privilege. To protect computers, some dangerous instructions can only be executed by OSs, preventing malicious software from randomly calling hardware resources. For example, if an application needs to enable a camera, it must request the operation permission from a Ring 0 driver, or the operation will be rejected.

The instructions given by a host OS are classified into two types: privileged instructions and common instructions.

- Privileged instructions: are instructions used to operate and manage key system resources. These instructions can be executed only at the highest privilege level (Ring 0).
- Common instructions: are instructions that can be executed at the non-privilege level (Ring 3).

In virtualization, there is also a special type of instructions called sensitive instructions. A sensitive instruction is used for changing the operating mode of a VM or the state of a host machine. The instruction is handled by a VMM after a privileged instruction that originally needs to be run in Ring 0 on the guest OS is deprived of the privilege.

Virtualization was first applied in IBM mainframes. How do mainframes implement CPU sharing? First, let's learn about the CPU virtualization of mainframes. Mainframe CPU virtualization uses privilege deprivileging and trap-and-emulate, which are also called classic virtualization. The basic principles are as follows: The guest OS runs at the non-privilege level (that is, privilege deprivileging) and the VMM runs at the highest privilege level (or fully controls system resources).

In this case, the following problem occurs: how is a privileged operation instruction given by a guest OS of a VM implemented? The privileges of all VM systems are deprived, trap emulation takes effect. After the privileges of the guest OS are deprived, most instructions of the guest OS can still run directly on the hardware. Only when privileged instructions are executed, the VMM emulation execution (trap-and-emulate) is performed. The VMM, in place of the VM, gives the privileged instruction to the physical CPU.

The classic CPU virtualization works with the timer interrupt mechanism of the physical OS to solve the CPU virtualization problem. Figure 6-7 is used as an example. VM 1 gives privileged instruction 1 to a VMM, and in this case, an interrupt is triggered. The VMM traps privileged instruction 1 given by VM 1 into the VMM for simulation, and then converts privileged instruction 1 into privileged instruction 1' of a CPU, the VMM schedules the request to the hardware CPU based on the scheduling mechanism and returns the result to VM 1.

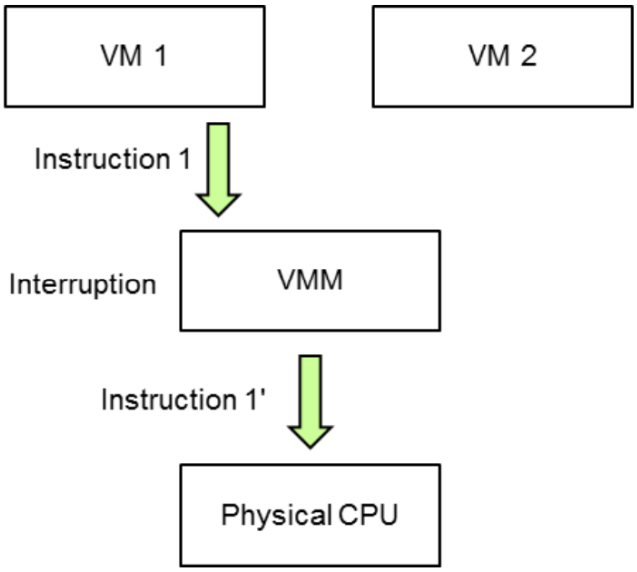


Figure 6-7 Unified scheduling of all instructions

Figure 6-8 is used as an example. When VM 1 and VM 2 simultaneously give privileged instructions to a VMM, the instructions are trapped and emulated, and a VMM scheduling mechanism performs unified scheduling. Instruction 1' is executed first, and then instruction 2' is executed. CPU virtualization function can be implemented by using the timer interrupt mechanism and privilege deprivileging and trap-and-emulate methods.

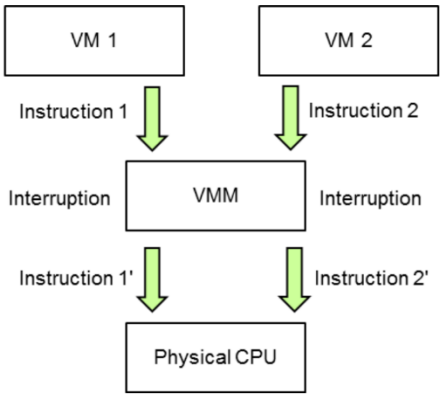


Figure 6-8 Special instructions

Why is the timer interruption mechanism required? If an emergency occurs outside the system, inside the system, or in the currently running program when the CPU runs the program, the CPU immediately stops the currently running program, automatically switches to a corresponding processing program (interrupt service program), and returns to the original program after the processing is complete. This entire process is called program interruption. Of course, the interruption time is too short to be realized by you.

As the performance of x86 hosts is increasingly enhanced, applying virtualization technologies to the x86 architecture becomes a major problem for virtualizing x86 servers. At this point, people naturally think of the CPU virtualization that was once used on mainframes. Can the classic CPU virtualization applied in mainframes be reused on x86 servers? The answer is no. Why? To answer this question, we need to understand the differences between the x86-architecture CPU and mainframe CPU.

Mainframes (including subsequent midrange computers) use the PowerPC architecture, that is, the reduced instruction set computer (RISC) architecture. In the CPU instruction set of the RISC architecture, sensitive instructions specific to VMs are included in privileged instructions, as shown in Figure 6-9.

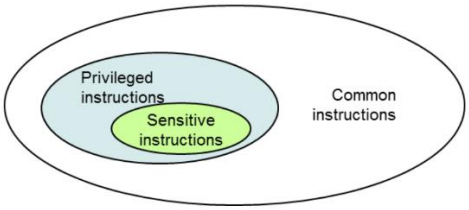


Figure 6-9 CPU instruction set of RISC architecture

After the privilege of the VM OS is removed, the privileged instructions and sensitive instructions can be trapped, emulated, and executed. Because the privileged instructions include sensitive instructions, the CPU with the RISC architecture can properly use the privilege deprivileging and trap-and-emulate methods. However, the CPU instruction set of the x86 architecture is different from that of the instruction set computer (CISC) architecture and the RISC architecture, as shown in 6-10.

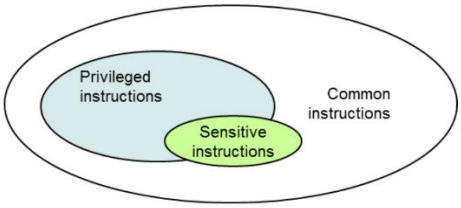


Figure 6-10 CPU instruction set of CISC architecture

As shown in figure 6-9 and 6-10, the privileged instructions and sensitive instructions in CISC instruction set do not completely overlap. Specifically, 19 sensitive instructions in the CISC instruction set based on the x86 architecture are not privileged instructions. These sensitive instructions run in the Ring 1 user mode of the CPU. What problems will this bring about? Obviously, when a VM gives the 19 sensitive instructions, these sensitive instructions cannot be trapped and emulated by the VMM. Therefore, x86 servers cannot use the classic virtualization for virtualizing the x86 architecture using privilege deprivileging and trap and emulate methods. This problem is called virtualization vulnerability. Since mainframe-based CPU virtualization cannot be directly transplanted to the x86 platform, what CPU virtualization should the x86 architecture use?

Smart IT architects came up with three ways to solve this problem. They are full virtualization, paravirtualization, and hardware-assisted virtualization (proposed by hardware vendors).

- Full virtualization

The x86-architecture CPUs are not suitable for the classic virtualization solution, and the root cause is that the CPU cannot identify and emulate 19 sensitive instructions. If these sensitive instructions are identified and can be trapped and emulated by the VMM, CPU virtualization can be implemented. But how can these 19 instructions be identified?

A fuzzy identification method can be used. All OS requests given by VMs are forwarded to a VMM, and the VMM performs binary translation on the requests. When the VMM detects privileged or sensitive instructions, the requests are trapped into the VMM for emulation. Then, the requests are scheduled to the CPU privilege level for execution. When the VMM detects program instructions, the instructions are executed at the CPU non-privilege level. This technique is called full virtualization because all request instructions given by VMs need to be filtered. Figure 6-11 shows how full virtualization is implemented.

Full virtualization was first proposed and implemented by VMware. VMs have high portability and compatibility. A VMM translates the binary code of the VM OS (guest OS) without modifying the VM OS. However, binary translation causes the performance overhead of VMM. On one hand, full virtualization has the following advantages: The VM OS does not need to be modified. VMs are highly portable and compatible, and support a wide range of OSs. On the other hand, it has the following disadvantages: Modifying the guest OS binary code during running causes large performance loss and increase the VMM development complexity. Xen developed the paravirtualization technique, which compensates for the disadvantages of full virtualization.

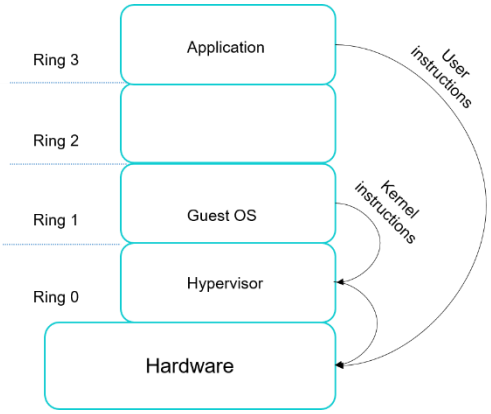


Figure 6-11 Full virtualization

- Paravirtualization

The virtualization vulnerability comes from the 19 sensitive instructions. If we can modify the VM OS (guest OS) to avoid the virtualization vulnerability, then the problem can be solved.

As shown in figure 6-12, paravirtualization modifies the guest OS so that the OS is able to aware that it is virtualized and uses the Hypercall to replace sensitive instructions in the virtualization with the hypervisor to implement virtualization. Non-sensitive or unauthorized program instructions are directly executed at the CPU non-privilege level. Paravirtualization has the following advantages: Multiple types of guest OSs can run at the same time and deliver performance similar to that of the original non-virtualized system. However, the disadvantage is that only open-source OSs such as Linux can be paravirtualized. In addition, the modified guest OS has poor portability.

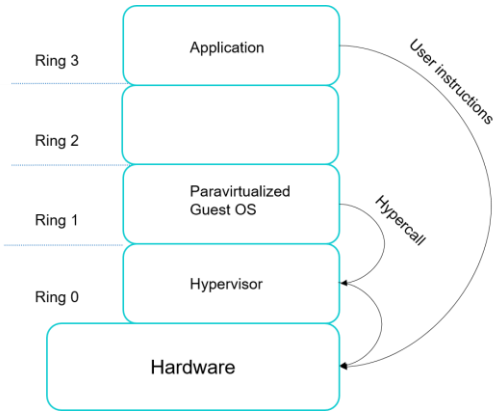


Figure 6-12 Paravirtualization

- Hardware-assisted Virtualization

In full virtualization and paravirtualization, the physical hardware does not support virtualization identification by default. The 19 sensitive instructions must be identified and the trap-emulation functions must be performed using the VMM. If physical CPUs support virtualization and are able to identify sensitive instructions, it will be a revolutionary change to CPU virtualization.

Fortunately, the CPUs of mainstream x86 hosts support the hardware virtualization. That is, Intel has launched the CPU with Intel Virtualization Technology (Intel VT-x), and AMD-V has launched the CPU with AMD-V. Both Intel VT-x and AMD-V add a new execution

mode (root mode) to the CPU. The VMM can run in the root mode, which is located at the CPU instruction level Ring 0. Privileged and sensitive instructions are automatically executed on the hypervisor. Full virtualization or paravirtualization is not required. This method that uses hardware-assisted virtualization to resolve virtualization vulnerabilities, simplify VMM work, and do not require paravirtualization and binary translation is called hardware-assisted virtualization of CPUs. Figure 6-13 shows the hardware-assisted virtualization.

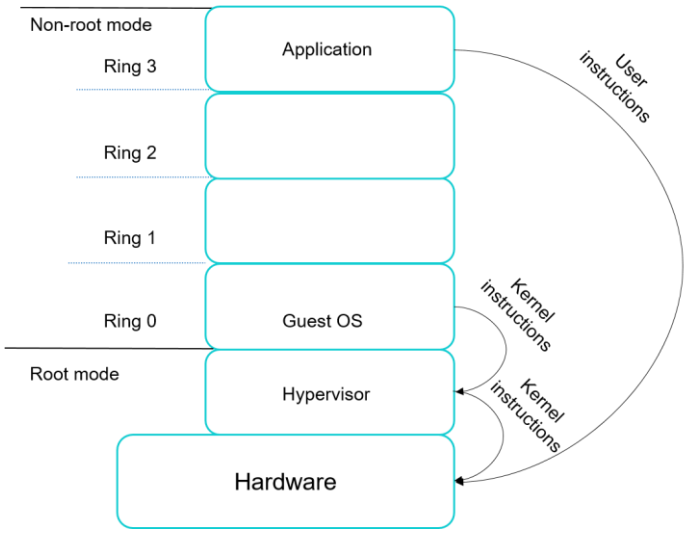


Figure 6-13 Hardware-assisted virtualization

6.1.1.6.1 Mappings Between CPUs and vCPUs

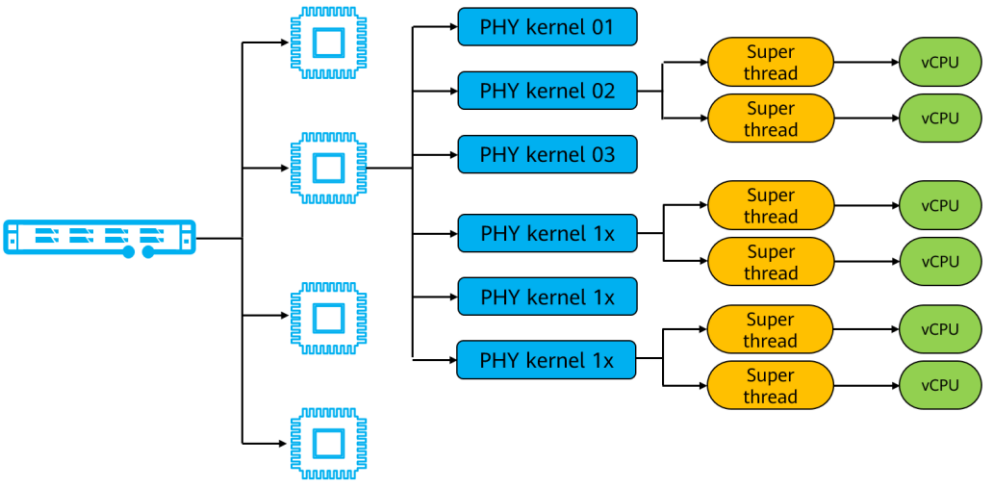


Figure 6-14 Mappings between CPUs and vCPUs

Figure 6-14 shows mappings between vCPUs and CPUs.

Let's take an RH server with the CPU frequency of 2.6 GHz as an example. A single server has two physical CPUs, each of which has eight cores. The hyper-threading technology provides two processing threads for each physical core. Each CPU has 16 threads, and the total number of vCPUs is 32 (2 x 8 x 2). The total resource of CPUs is calculated as follows: 32 x 2.6 GHz = 83.2 GHz.

The number of vCPUs on a VM cannot exceed the number of available vCPUs on a CNA node, where CNA indicates a computing node agent. Multiple VMs can reuse the same CPU, and the total number of vCPUs running on a CNA node can exceed the actual number of vCPUs.

6.1.1.7 Memory Virtualization

Beyond CPU virtualization, the next is memory virtualization. Why does CPU virtualization lead to memory virtualization?

With the emergence and application of CPU virtualization, VMs running on the VMM layer replace physical hosts to carry services and applications. In addition, multiple VMs run on the same physical host at the same time. A physical host usually has one or more memory modules. How can memory resources be allocated to multiple VMs properly? Memory virtualization was introduced to address this issue. One problem encountered by memory virtualization is how to allocate memory address space. Generally, when a physical host uses the memory address space, the following requirements must be met:

- The memory address space starts from the physical address 0.
- Addresses in the memory address space are assigned contiguously.

However, after virtualization is introduced, the following problems occur:

- There is only one memory address space that can start with the physical address 0, so it is impossible to ensure that the memory address space of all VMs start from the physical address 0.
- Although contiguous physical addresses can be assigned, this way of memory allocation leads to poor efficiency and flexibility in memory usage.

Memory virtualization was introduced to resolve issues involving memory sharing and dynamical allocation of memory addresses. Memory virtualization centrally manages and divides the physical memory of a physical machine into virtual memories for VMs. Memory virtualization provides a new address space, that is, physical address space of a client. The guest machine considers that it runs in a physical address space. Actually, the guest machine accesses a physical address through VMMs. VMMs store mappings between guest machine address spaces and physical machine address spaces, as shown in the figure 6-15.

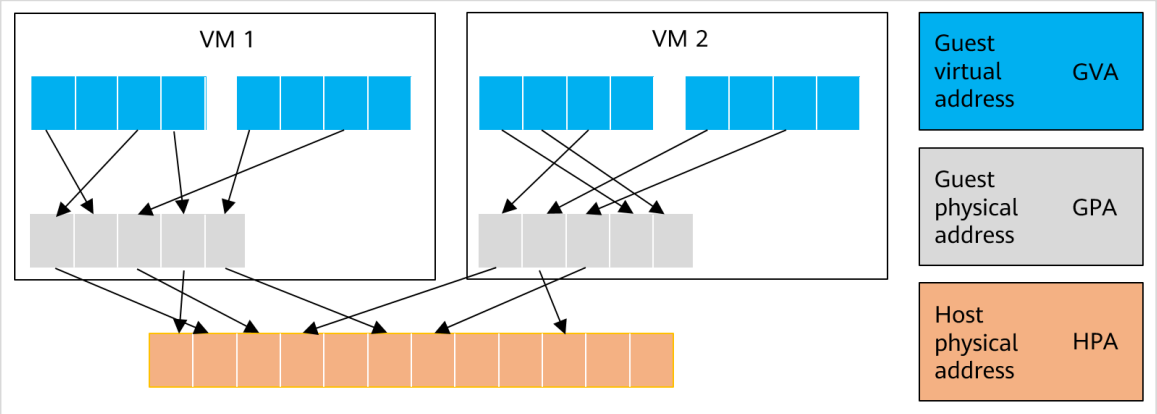


Figure 6-15 Memory virtualization

Memory address translation of memory virtualization involves three types of memory addresses: guest virtual addresses (GVAs), guest physical addresses (GPAs), and host physical addresses (HPAs). To run multiple VMs on the same physical host, the following address translation must be implemented: GVA to GPA to HPA. The guest OS on a VM controls the mapping from the GVA to the GPA. However, the guest OS cannot directly access actual host memory. The hypervisor needs to map the GPA to the HPA.

Note: We can use an example to explain the difference between HPA and GPA. If a server has a total of sixteen 16-GB memory bars, its GPA is 256 GB and HPA is sixteen memory bars distributed across different memory slots.

6.1.1.8 I/O Virtualization

Due to the emergence of compute virtualization, a large number of VMs are created on a physical host, and these VMs needs to access the I/O devices of the physical host. However, the number of I/O devices is limited. I/O device sharing among multiple VMs requires VMMs. VMM intercepts access requests from VMs to I/O devices, simulates physical I/O devices using software, and responds to I/O requests. In this way, multiple VMs have access to limited I/O devices. I/O virtualization can be implemented in the following modes: full virtualization, para-virtualization, and hardware-assisted virtualization (mainstream).

- Full virtualization: VMMs virtualize I/O devices for VMs. When a VM initiates an I/O request to an I/O device, the VMM intercepts and sends the request given by the VM to the physical device for processing. No matter which type of OS is used by the VM, the OS does not need to be modified for I/O virtualization. Multiple VMs can directly use the I/O device of the physical server. However, the VMM needs to intercept I/O requests delivered by each VM in real time and emulates these requests to a physical I/O device. Real-time monitoring and emulation are implemented by software programs on the CPU, which causes severe performance loss to the server.
- Paravirtualization: Unlike full virtualization, paravirtualization needs a privileged VM. Paravirtualization requires each VM to run a frontend driver. When VMs need to access an I/O device, the VMs send I/O requests to the privileged VM through the frontend driver, and then the backend driver of the privileged VM collects the I/O requests given by each VM. After that, the backend driver processes multiple I/O requests by time and by channel. The privileged VM runs the physical I/O device driver and sends the I/O request to the physical I/O device. After processing the request, the I/O device returns the processing result to the privileged VM. VMs send I/O requests to a privileged VM and then the privileged VM accesses a physical I/O device. This reduces the performance loss of the VMM. However, the VM OS needs to be modified. Specifically, the I/O request processing method of the OS needs to be changed so that all the I/O requests can be given to the privileged VM for processing. The VM OS (usually Linux) must be modifiable.

  Let's compare full virtualization and paravirtualization. A fully virtualized VMM is equivalent to an investigator who needs to collect and summarize each customer's opinion requests. A paravirtualized VMM is equivalent to an opinion collection box (privileged VM) where each user puts opinion request actively, and then the VMM processes these requests in a unified manner. Paravirtualization significantly reduces

the performance loss of VMM and delivers better I/O performance. Full virtualization and paravirtualization have one feature in common: VMM is responsible for I/O access processing, which causes performance loss when VMs access I/O devices.

- Hardware-assisted virtualization: With this type of virtualization, I/O device drivers are directly installed on the VM OS, without any modification to the OS. This method is equivalent to direct access from the host OS to hardware. In this way, the time required for a VM to access the I/O hardware is the same as that in the traditional way. As shown in the preceding example, hardware-assisted virtualization is equivalent to an intelligent information collection and processing platform. Users' requests can be directly submitted to the platform and then the platform processes them automatically without manual intervention. Hardware-assisted virtualization provides much higher I/O performance than full virtualization and paravirtualization. However, hardware-assisted virtualization requires special hardware support.

6.1.2 Mainstream Virtualization Technologies

6.1.2.1 Xen Virtualization

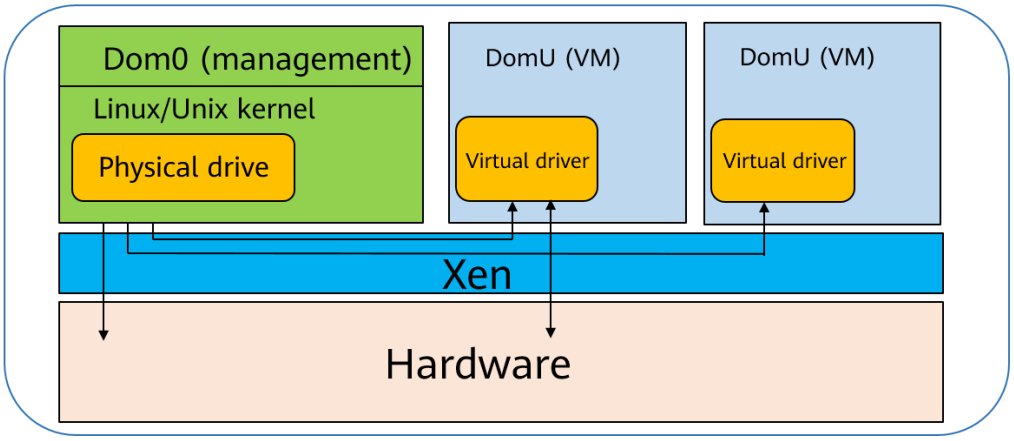


Figure 6-16 Xen architecture

Xen was initially an open-source research project of Xensource founded by Cambridge University. In September 2003, Xen 1.0 was released. In 2007, Xensource was acquired by Citrix, and then Xen was promoted by Xen Project (www.xen.org), whose members include individuals and companies (such as Citrix and Oracle). In March 2011, the organization released Xen 4.1.

Xen not only supports the x86/x86_64 CPU architecture of CISC that both ESX and Hyper-V support but also RISC CPU architectures (IA64 and ARM).

Xen supports two types of virtualization: paravirtualization (PV) or hardware virtual machine (HVM). PV requires OSs with specific kernels, for example, the Linux kernel based on the Linux paravirt_ops (a set of compilation options of the Linux kernel) framework. However, Xen PV does not support Windows due to its closeness. There is something special for Xen PV: CPUs are not required to support hardware-assisted virtualization, which is applicable to the virtualization of old servers produced before 2007. Xen HVM supports native OSs, especially Windows, and Xen HVM requires CPUs to support hardware-assisted virtualization. It can modify all hardware (including the BIOS, IDE controllers, VGA video cards, USB controllers, and NICs) emulated by QEMU. To

improve I/O performance, paravirtualized devices replace emulated devices for disks and NICs in full virtualization. Drivers of these devices are called PV on HVM. To maximize performance of PV on HVM, the CPU must support MMU hardware-assisted virtualization.

As shown in Figure 6-16, the Xen hypervisor (similar to a microkernel) directly runs on the hardware and starts before Domain 0. Domain 0 serving as a privileged domain in Xen can be used to create new VMs and access native device drivers. Figure 6-16 shows the early Xen architecture. Each virtual domain has a frontend component. Xen needs to modify guest OS. To access a device, the guest OS must interact with the backend in Domain 0 through the frontend component, and then Domain 0 accesses the device.

The Xen hypervisor layer has less than 150,000 lines of code. In addition, it, similar to Hyper-V, does not include any physical drives. The physical drive loaded in Dom0 can reuse the existing drivers in Linux. Xen is compatible with all hardware Linux supports.

6.1.2.2 KVM Virtualization

KVM is short for Kernel-based Virtual Machine. It was originally an open source project developed by Qumranet. In 2008, Red Hat acquired Qumranet. However, KVM is still an open-source project supported by providers such as Red Hat and IBM.

KVM is a kernel-based VM because KVM is a Linux kernel module. After this module is installed on a physical machine running Linux, the physical machine becomes a hypervisor without affecting other applications running on Linux. KVM supports CPU architectures and products, such as x86/x86_64 CPU architecture (also for Xen), mainframes, midrange computers and ARM architecture.

KVM makes full use of the hardware-assisted virtualization of CPU and reuses many functions of the Linux kernel. As a result, KVM consumes a few resources. Avi Kivity, the founder of KVM, claimed that the KVM module had only about 10,000 lines of code. However, we cannot naturally conclude that KVM hypervisor just had that amount of code, because KVM is actually a module that can be loaded in the Linux kernel. It is used to turn the Linux kernel into a hypervisor.

A Linux kernel is converted into a hypervisor by loading a KVM module. The Linux runs in kernel mode, a host process runs in user mode, and a VM runs in guest mode, so the converted Linux kernel can perform unified management and scheduling on the host process and the VM. This is why KVM got its name.

6.1.2.3 KVM and QEMU

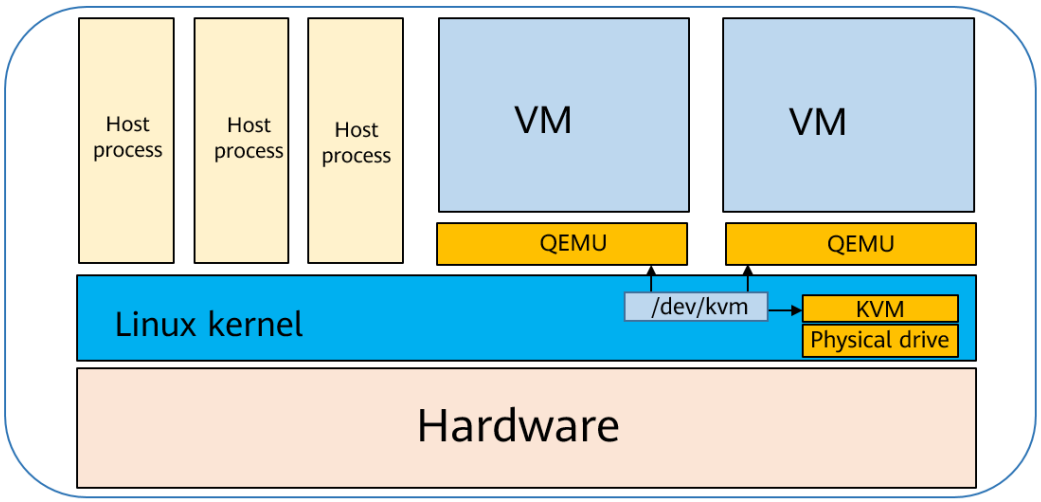


Figure 6-17 KVM and QEMU

In the KVM virtualization solution, KVM virtualizes CPU and memory, and QEMU virtualizes I/O devices.

QEMU is software-based open-source (emulation) software. It can emulate all resources required by VMs, including the CPU, memory, I/O device, USB, and NIC.

KVM is used to emulate CPU running, but does not support networks and I/O. QEMU-KVM is a complete KVM-based emulator and supports complete I/O simulation. To achieve cross-VM performance, OpenStack does not directly control QEMU-KVM but uses the Libvirt library to control QEMU-KVM. We will introduce Libvirt later.

KVM cannot be separated from QEMU. To simplify development and reuse code, KVM was modified based on QEMU at the early stage. CPU virtualization and memory virtualization that consume much CPU performance are transferred and implemented in the kernel, while the I/O virtualization module is reserved for implementation in the user space. This avoids frequent switching between the user mode and kernel mode and optimizes performance.

QEMU cannot be separated from KVM. QEMU is emulated by pure software and runs on user controls, so it has poor performance. QEMU uses KVM virtualization to accelerate its VMs and provide resources for them.

The **/dev/kvm** interface bridges QEMU and KVM. **/dev/kvm** is a device file. You can use the ioctl function to control and manage this file to implement data interaction between user space and kernel space. The communication process between KVM and QEMU is a series of ioctl system calls for **/dev/kvm**.

6.1.2.4 Working Principles of KVM

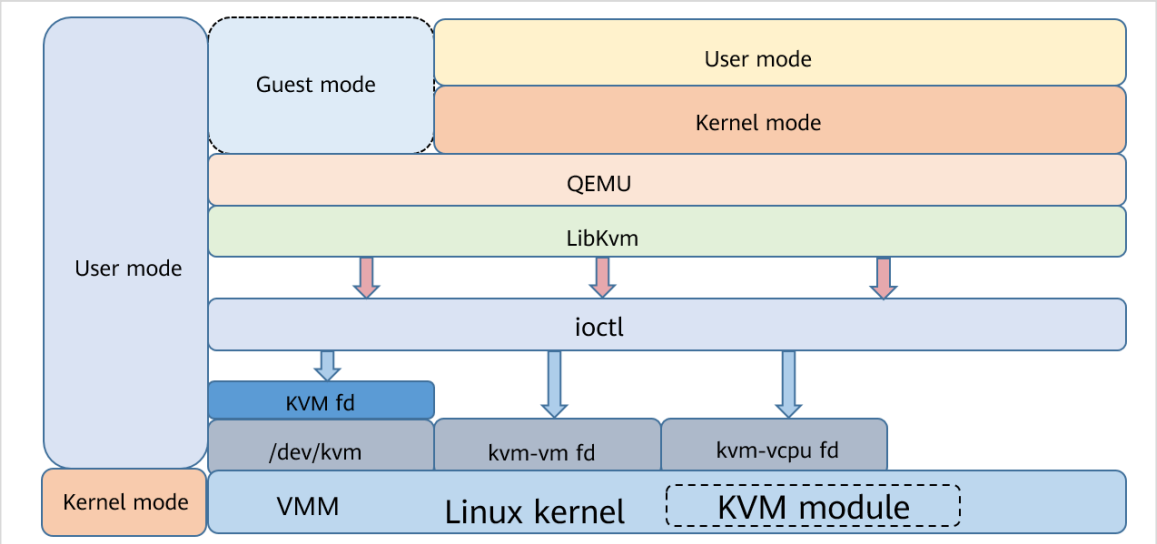


Figure 6-18 Working principles of KVM

Figure 6-18 shows the basic structure of KVM. KVM is a kernel module and is regarded as a standard Linux character set device (**/dev/kvm**). QEMU uses the file descriptor (**fd**) and ioctl to send VM creation and running commands to the device driver through the Libkvm API. KVM can parse commands.

The KVM module enables the Linux host to function as a VMM. The guest mode is added to the existing modes. There are three working modes for VMs:

- Guest mode: executes non-I/O guest codes. VMs run in this mode.
- User mode: executes I/O instructions on behalf of a user. QEMU runs in this mode to simulate I/O operation requests for VMs.
- Kernel mode: It can switch to the guest mode and process VM-Exit caused by I/O or other instructions. The KVM module works in the kernel mode where hardware can be operated. To this end, the guest OS needs to submit a request to the user mode when performing an I/O operation or executing a privileged instruction, and then the user mode initiates a hardware operation to the kernel mode.

6.1.2.5 Virtualization Platform Management Tool - Libvirt

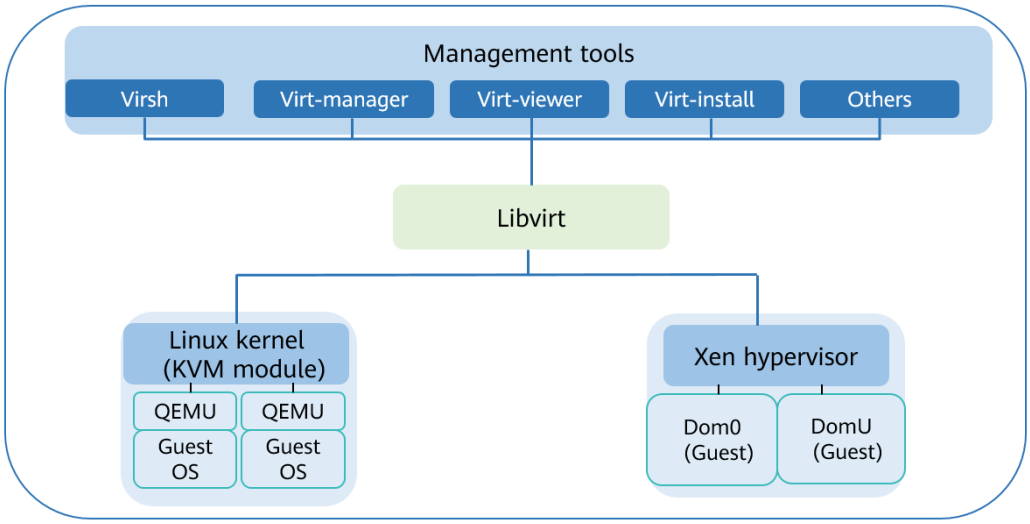


Figure 6-19 Virtualization platform management tool - Libvirt

In different virtualization scenarios, many solutions (such as KVM and Xen) are proposed. To support more vendors and service areas, many IaaS solutions need to integrate lots of virtualization. To this end, Libvirt provides a platform management tool for users, supporting multiple virtualization solutions.

Libvirt is an open-source API, daemon, and management tool designed for managing platform virtualization technologies. It manages not only virtualized clients, but also virtualized networks and storage.

Through a driver-based architecture, Libvirt supports different hypervisors. Loaded drivers vary for hypervisors: Xen driver for Xen and QEMU driver for QEMU or KVM.

A Libvirt works as an intermediate adaptation layer. It shields details of hypervisors, so the hypervisors are completely transparent to the management tool of the user space. By doing so, Libvirt provides a unified and stable API for the management tool.

6.1.2.6 Xen vs. KVM

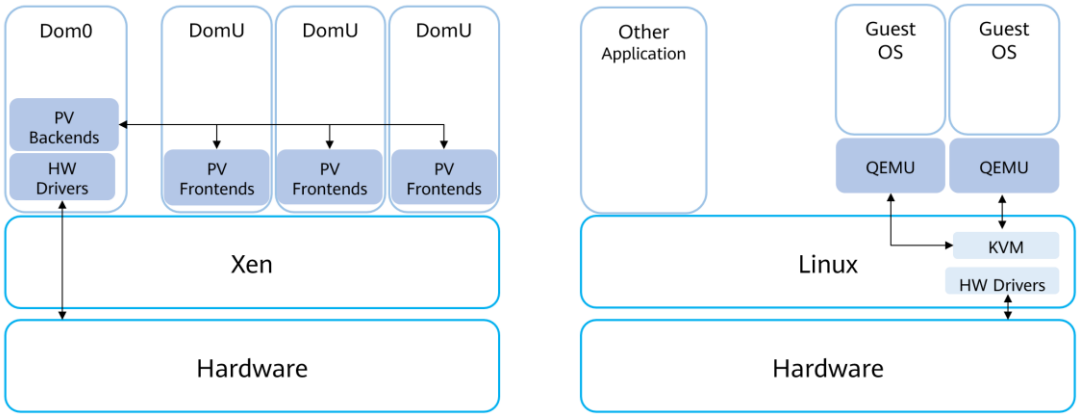


Figure 6-20 Xen vs. KVM

In general, the Xen platform and KVM platform has their own advantages.

The Xen architecture focuses on security. To ensure security, the access of domains to the shared zone must be authorized by the hypervisor.

The KVM architecture focuses on performance. The access and mapping of the shared zone between VMs or between VMs and the host kernel do not need to be authorized by the hypervisor, so the access path is short. No performance loss occurs due to the usage of the Linux Baremetal kernel.

6.2 Quiz

In KVM virtualization, what are the differences between networks using NAT gateways and bridges in allocating VM addresses?

7 Huawei Virtualization Platform

This chapter describes the basic concepts, architecture, positioning, functions, planning, and deployment of Huawei FusionCompute virtualization platform.

7.1 Introduction to FusionCompute

7.1.1 FusionCompute Virtualization Suite

Huawei FusionCompute virtualization suite is an industry-leading virtualization solution that provides the following benefits:

- Improving data center infrastructure utilization
- Significantly accelerating service rollout
- Substantially lowering power consumption in data centers
- Achieving rapid automatic fault recovery based on high availability and powerful restoration features, lowering data center cost, and increasing system uptime

FusionCompute virtualization suite virtualizes hardware resources using the virtualization software deployed on physical servers, so that one physical server serves as multiple virtual servers. It consolidates existing workloads and uses available servers to deploy new applications and solutions, realizing a high integration rate.

Single-hypervisor applies if an enterprise only uses FusionCompute as a unified operation, maintenance, and management platform to operate and maintain the entire system, including monitoring resources, managing resources, and managing the system.

FusionCompute virtualizes hardware resources and centrally manages the virtual resources, service resources, and user resources. Specifically speaking, it virtualizes compute, storage, and network resources using the compute, storage, and network virtualization technologies. In addition, it centrally schedules and manages virtual resources over a unified interface, thereby ensuring high system security and reliability and reducing the OPEX.

7.1.2 FusionCompute Positioning and Architecture

7.1.2.1 FusionCompute Positioning

FusionCompute is a cloud OS software product that virtualizes hardware resources and centrally manages the virtual resources, service resources, and user resources. It virtualizes compute, storage, and network resources using the compute, storage, and network virtualization technologies. By centrally scheduling and managing virtual

resources over a unified interface, it provides high system security and reliability while reducing OPEX. It helps carriers and other enterprises build secure, green, and energy-saving cloud data centers.

7.1.2.2 Position of FusionCompute in the Virtualization Suite

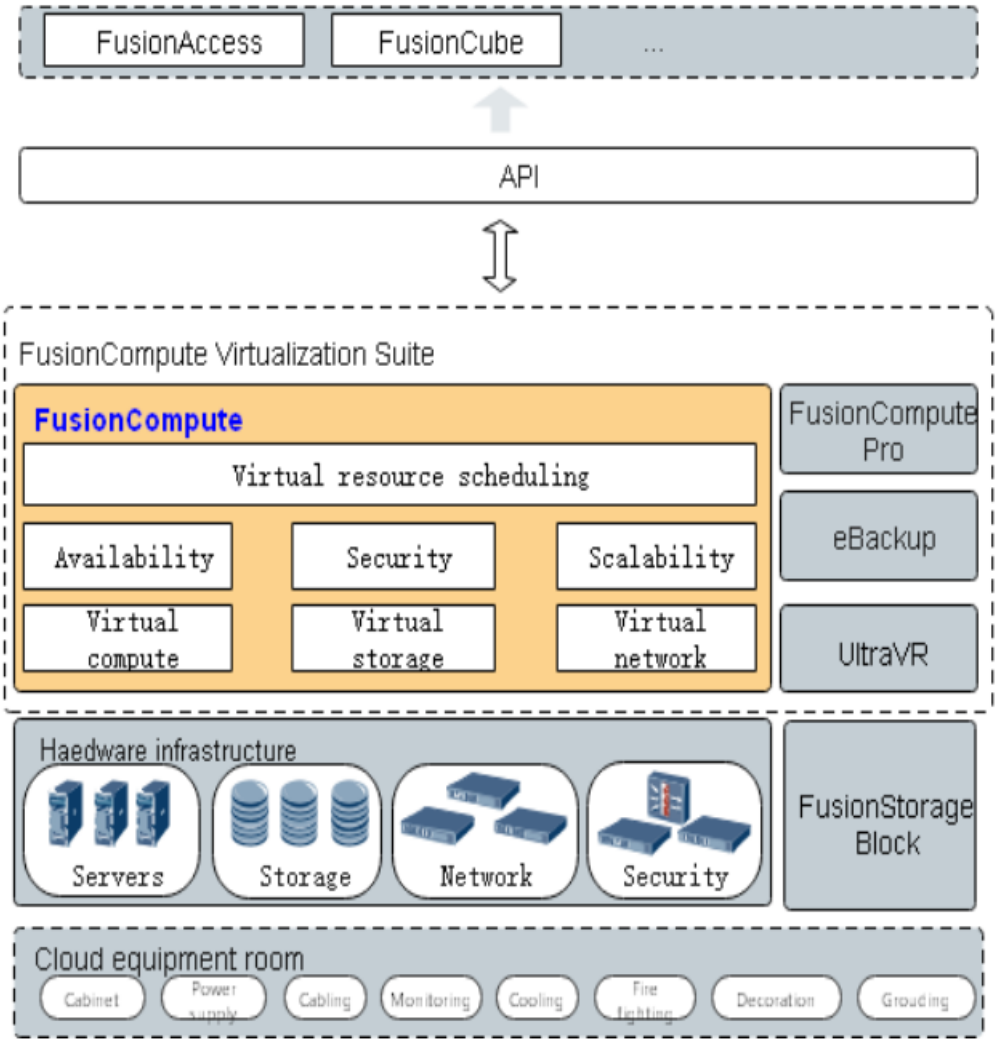


Figure 7-1 FusionCompute position

Figure 7-1 shows the position of FusionCompute in the virtualization suite. The modules in the preceding figure are described as follows. This chapter focuses on FusionCompute.

- Cloud facilities

Cloud facilities refer to the auxiliaries and space required by the cloud DC, including the power system, fire-fighting system, cabling system, and cooling system.

- Hardware infrastructure layer

Hardware infrastructure consists of servers, storage devices, network devices, and security devices. These resources allow customers to build different scales of systems and expand

its capacity based on actual needs and to use applications ranging from the entry level to the enterprise level. Various devices provide customers with multiple and flexible choices.

- FusionStorage Block

FusionStorage Block, also called Huawei Distributed Block Storage, is a distributed storage software product that integrates storage and compute capabilities. It can be deployed on general-purpose servers to consolidate the local disks on all the servers into a virtual storage resource pool, aiming to provide the block storage function.

- FusionCompute Virtualization Suite

FusionCompute virtualization suite virtualizes hardware resources using the virtualization software deployed on physical servers, so that one physical server serves as multiple virtual servers. It consolidates existing workloads and uses available servers to deploy new applications and solutions, realizing a high integration rate.

- FusionCompute

FusionCompute is a cloud OS software product that virtualizes hardware resources and centrally manages the virtual resources, service resources, and user resources. It virtualizes compute, storage, and network resources using the compute, storage, and network virtualization technologies. By centrally scheduling and managing virtual resources over a unified interface, it provides high system security and reliability while reducing OPEX. It helps carriers and other enterprises build secure, green, and energy-saving cloud data centers.

- eBackup

eBackup is a virtualized backup software product, which works with the FusionCompute snapshot function and the Changed Block Tracking (CBT) function to back up VM data.

- UltraVR

UltraVR is a DR management software product, which provides data protection and DR for key VM data using the asynchronous remote replication feature provided by the underlying SAN storage system.

- FusionCompute Pro

FusionCompute Pro is a component for unified management of resource sites in different regions. It uses virtual data canters (VDCs) to provide domain-based resource management capabilities for different users.

7.1.2.3 FusionCompute Architecture

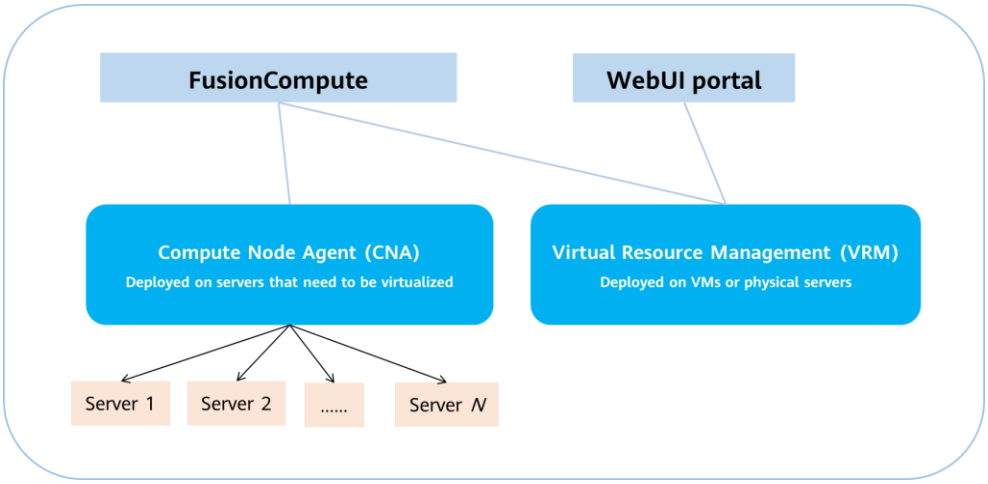


Figure 7-2 FusionCompute architecture

FusionCompute consists of two parts: Compute Node Agent (CNA) and Virtual Resource Management (VRM). In addition to CNA and VRM, Unified Virtualization Platform (UVP) is a unified virtualization platform developed by Huawei. UVP is a hypervisor, like KVM and Xen. The FusionCompute hypervisor adopts the bare-metal architecture and can run directly on servers to virtualize hardware resources. With the bare-metal architecture, FusionCompute delivers VMs with almost server-level performance, reliability, and scalability.

The architecture of FusionCompute is similar to that of KVM. VRM functions as the management tool of KVM. Administrators and common users can manage and use FusionCompute on the WebUI provided by VRM. VRM is based on the Linux OS. Therefore, many Linux commands can be used after you log in to VRM.

VRM provides the following functions:

- Managing block storage resources in a cluster
- Managing network resources, such as IP addresses and virtual local area network (VLAN) IDs in a cluster, and allocating IP addresses to VMs
- Managing the life cycle of VMs in a cluster and distributing and migrating VMs across compute nodes
- Dynamically adjusting resources in a cluster
- Implementing centralized management of virtual resources and user data and providing elastic computing, storage, and IP address services
- Allowing O&M engineers to remotely access FusionCompute through the unified WebUI to perform resource monitoring and management and view resource statistics reports

CNA is similar to the QEMU+KVM module in KVM. CNA provides the virtualization function. It is deployed in a cluster to virtualize the compute resources, storage resources, and network resources in the cluster into a resource pool for users to use. CNA is also based on the Linux OS.

CNA provides the following functions:

- Implementing the virtual computing function
- Managing VMs running on compute nodes
- Managing compute, storage, and network resources on compute nodes

CNA manages VMs and resources on the local node. VRM manages resources in a cluster or resource pool. When users modify a VM or perform other VM lifecycle operations on VRM, VRM sends a command to the CNA node. Then, the CNA node executes the command. After the operation is complete, CNA returns the result to VRM, and VRM records the result in its database. Therefore, do not modify VMs or other resources on CNA. Otherwise, the records in the VRM database may be inconsistent with the actual operations.

In addition to Huawei's hardware products, FusionCompute also supports other servers based on the x86 hardware platform and is compatible with multiple types of storage devices, allowing enterprises flexibly choose appropriate devices. A cluster supports a maximum of 128 hosts and 8000 VMs. FusionCompute provides comprehensive rights management functions, allowing authorized users to manage system resources based on their specific roles and assigned permissions.

7.1.2.4 FusionCompute Functions - Virtual Computing

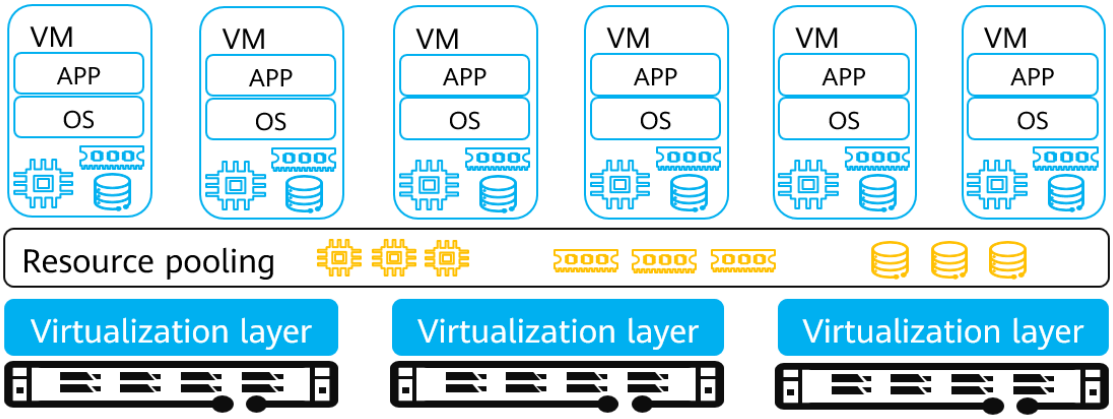


Figure 7-3 Virtual computing

FusionCompute provides the following compute virtualization functions: server virtualization, VM resource management, distributed resource scheduling, power management, and VM live migration.

1. Server virtualization

Server virtualization enables physical server resources to be converted to logical resources. With virtualization technologies, a server can be divided into multiple isolated virtual compute resources, while CPU, memory, disk, and I/O resources become pooled resources that are dynamically managed. This increases the resource utilization and simplifies system management. In addition, the hardware-assisted virtualization technology increases virtualization efficiency and enhances VM security. In FusionCompute, server virtualization supports the following features:

- Bare-metal architecture

The FusionCompute hypervisor adopts the bare-metal architecture and can run directly on servers to virtualize hardware resources. With the bare-metal architecture,

FusionCompute delivers VMs with almost server-level performance, reliability, and scalability.

- CPU virtualization

FusionCompute converts physical CPUs to virtual CPUs (vCPU) for VMs. When multiple vCPUs are running, FusionCompute dynamically allocates CPU capabilities among the vCPUs.

- Memory virtualization

FusionCompute adopts the hardware-assisted virtualization technology to reduce memory virtualization overhead.

- GPU passthrough

In FusionCompute, a Graphic Processing Unit (GPU) on a physical server can be directly attached to a specified VM to improve graphics and video processing capabilities. With this feature enabled, the system can meet user requirements for high-performance graphics processing capabilities.

- USB passthrough

In FusionCompute, a USB device on a physical server can be directly attached to a specified VM. This feature allows users to use USB devices in virtualization scenarios.

2. VM resource management

Based on FusionCompute, VM resource management allows administrators to create VMs using a VM template or in a custom manner, and manage cluster resources. This feature provides the following functions: automated resource scheduling (including the load balancing mode and dynamic energy-saving mode), VM life cycle management (including creating, deleting, starting, stopping, hibernating (in the x86 architecture), waking up (in the x86 architecture), and restarting VMs), storage resource management (including managing common disks and shared disks), VM security management (including using custom VLANs), and VM QoS adjustment based on the service load (including setting CPU QoS and memory QoS).

3. Distributed resource scheduling and power management

FusionCompute provides various pooled virtual resources, such as compute resources, storage resources, and network resources. FusionCompute intelligently schedules virtual resources to balance system loads, thereby ensuring high system reliability and availability, optimizing user experience, and improving the resource utilization of the data center. FusionCompute supports resource scheduling for:

- Load balancing

The system monitors the operating status of compute servers and VMs. When detecting that the load on servers in a cluster varies significantly and exceeds the preset thresholds, the system automatically migrates VMs based on the predefined load balancing policy to balance the utilization of resources, such as CPUs and memory.

- Dynamic energy saving

Resource scheduling for energy saving can be enabled only after resource scheduling for load balancing is enabled. The system monitors the operating status of compute servers and VMs in each cluster. When detecting that the service load of a cluster is light, the system automatically migrates VMs to certain servers based on the predefined energy

saving policy and powers off the idle compute servers. When detecting that the service load of a cluster is heavy, the system automatically wakes up the compute servers to beat service load.

4. VM live migration

FusionCompute allows VMs to migrate among the hosts that share the same storage. During the migration, services are not interrupted. This reduces the service interruption time caused by server maintenance and saves power consumption in data centers.

7.1.2.5 FusionCompute Functions - Virtual Storage

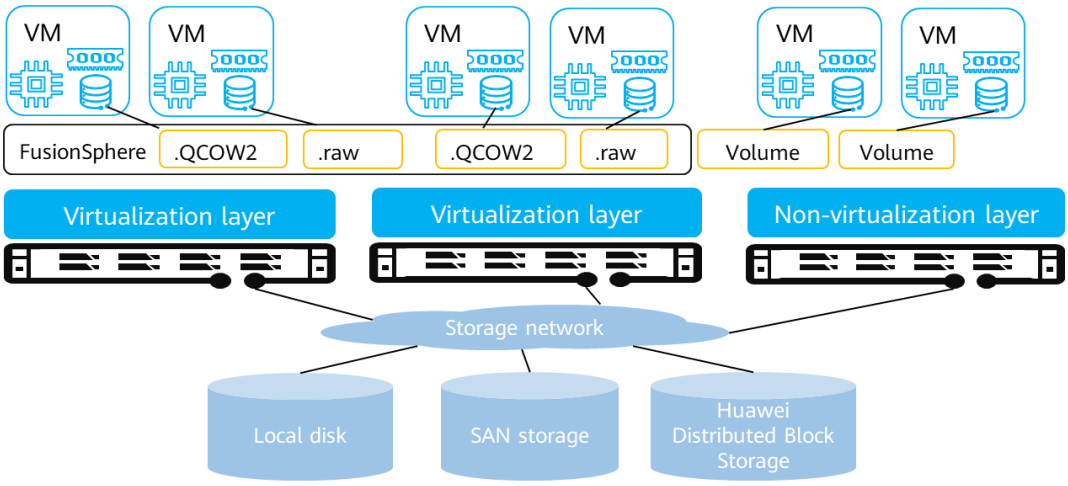


Figure 7-4 Virtual storage

The storage virtualization technology helps the system better manage the storage resources for virtual infrastructure with high resource utilization and flexibility, increasing application uptime. FusionCompute centrally manages the virtual storage resources provided by SAN devices, Huawei Distributed Block Storage, and local storage for compute nodes, and allocates the resources to VMs in the form of virtual volumes.

Storage device performance varies, and the protocols used by storage devices interfaces are different. To address the issues, storage virtualization is used to aggregate resources from different storage devices and provide data stores to manage the resources with the simplicity of a single storage device. Data stores can be used to store VM disk data, VM configuration information, and snapshots. As a result, users can implement homogeneous management on their storage resources.

VM disk data and snapshots are stored as files on data stores. All service-related operations can be converted to operations on files, which enables visibility and agility.

Based on the storage virtualization technology, Huawei provides multiple storage services to improve storage utilization, reliability, and maintainability, as well as user experience.

Huawei provides host-based storage virtualization, hiding the complexity of storage device types and bypassing the performance bottlenecks. Storage virtualization abstracts the storage devices as logical resources, thereby providing comprehensive and unified storage services. This feature hides the differences of physical storage devices and provides unified storage functions.

End users can use these virtual volumes on VMs like using local disks on x86 servers. For example, they can format these virtual volumes, read data from or write data into them,

install file systems, and install OSs. Moreover, virtual storage supports the snapshot function and resizing, which cannot be implemented on physical disks.

Administrators only need to manage the SAN devices, instead of managing specific disks. Because SAN devices are reliable, the workloads for replacing disks are significantly decreased for administrators. In addition, virtual storage supports various features that are not supported by physical disks, such as thin provisioning, quality of service (QoS), and migration. Therefore, virtual storage has distinct cost advantages over physical disks.

7.1.2.6 FusionCompute Functions - Virtual Network

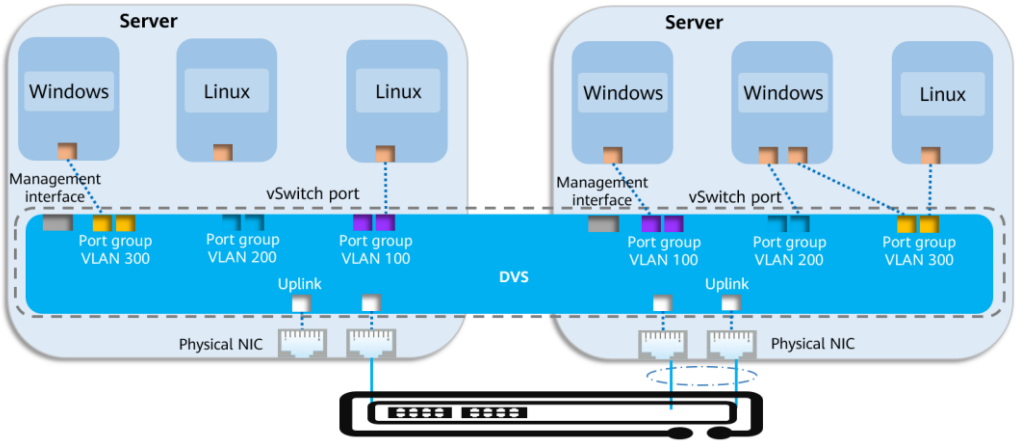


Figure 7-5 Virtual network

FusionCompute uses distributed virtual switches (DVSs) to provide independent network planes for VMs. Different network planes are isolated by VLANs, like on physical switches.

FusionCompute provides virtual network interface cards (vNICs) for VMs, virtual network I/O control, and distributed virtual switches (DVSs).

1. vNIC

You can add vNICs for each VM. Each vNIC has an IP address and a MAC address. It has the same functions as a physical NIC on a network. FusionCompute implements multiple queues, virtual swapping, QoS, and uplink aggregation to improve the I/O performance of vNICs.

2. Network I/O control

The network QoS policy enables bandwidth configuration control.

- Bandwidth control based on the sending direction and receiving direction of a port group member port.
- Traffic shaping and bandwidth priority are configured for each member port in a port group to ensure network QoS.

3. DVS

Each host connects to a DVS that functions as a physical switch. In the downstream direction, the DVS connects to VMs through virtual ports. In the upstream direction, the DVS connects to physical Ethernet adapters on hosts where VMs reside. The DVS implements network communication between hosts and VMs. In addition, a DVS functions as a single virtual switch between all associated hosts. This function ensures network configuration consistency during cross-host VM migration.

7.1.2.7 Benefits of FusionCompute (1)

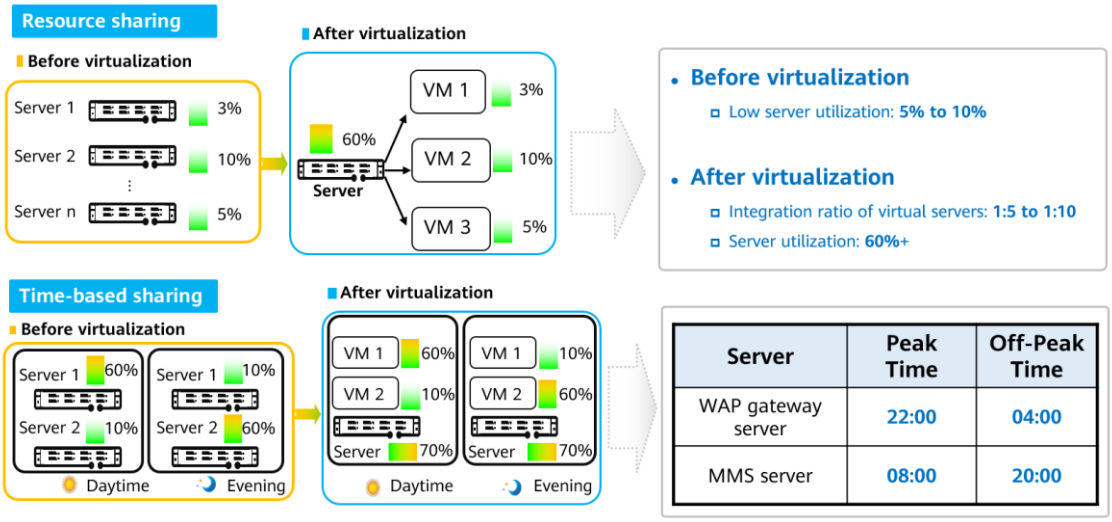


Figure 7-6 Resource sharing and time-based sharing

Resource sharing

- Before virtualization, each physical machine runs an independent application. The resource utilization and return on investment (ROI) is low.
- After virtualization, the three applications run on one server, which greatly reduces the number of servers to be purchased, and improves the server resource utilization as well as ROI.

Time-based sharing

- Before virtualization, applications run on different servers. When the service load of an application on a server is heavy, a server with light service load cannot transfer remaining computing resources to the server. As a result, servers are not used in a balanced manner.
- After virtualization, applications share all server resources in the system. If the service load of an application on a server is heavy and that on another server is light, FusionCompute allows servers to dynamically assign resources to each application, thereby fulfilling service load requirements and improving the high utilization of the server resources.

Through the resource sharing and time-based sharing, the resource virtualization can improve resource utilization, lower investment cost, and improve ROI.

7.1.2.8 Benefits of FusionCompute (2)

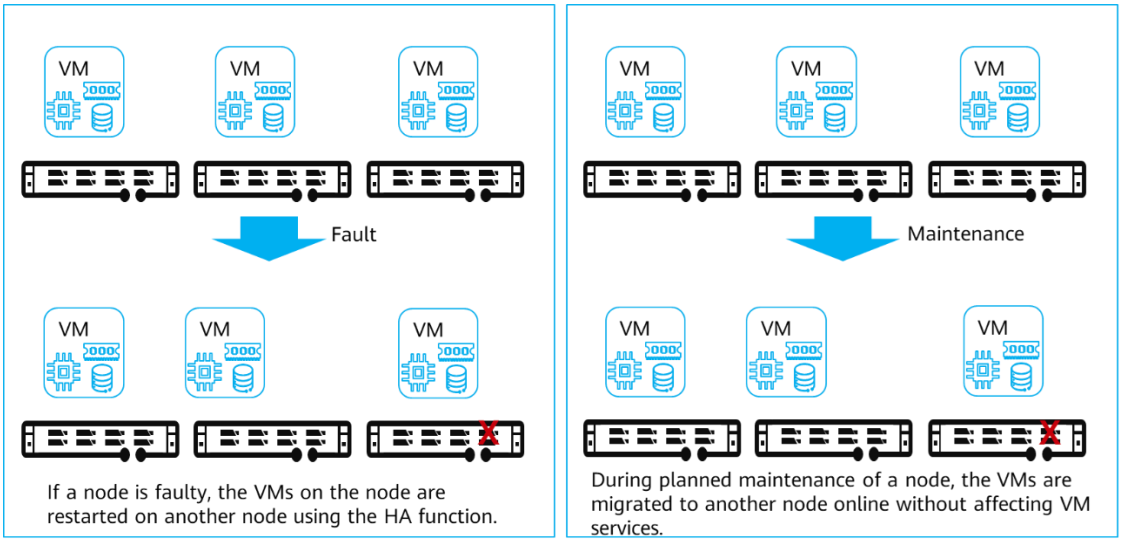


Figure 7-7 VM HA and live migration

1. VM HA

VM HA allows VMs on a server to automatically start on another properly-running server within a few minutes if their original server becomes faulty. The services running on the VMs can also automatically recover on the new server after the VMs are migrated to it. This feature ensures a VM, if faulty, can recover quickly. It allows the system to automatically recreate the VM on another normal compute node.

When detecting a VM fault, the system selects a normal compute node to create the faulty VM on the normal compute node.

- Restart or restoration of a compute node from a power failure

When a compute node restarts or recovers from a power outage, the system re-creates the HA VMs running on this node on other compute nodes.

- VM BSOD (Intel)

When detecting that BSOD occurs on a VM and the handling policy configured for this error is HA, the system re-creates the VM on another normal compute node.

2. VM live migration

This feature allows VMs to be live migrated from one server to another without interrupting user services. Therefore, planned server maintenance will not interrupt applications.

With this feature, VMs in the same cluster can be migrated from one physical server to another without interrupting services. The VM manager provides quick recovery of memory data and memory sharing technologies to ensure that the VM data before and after the live migration remains unchanged.

Before performing O&M operations on a physical server, system maintenance engineers need to migrate VMs from this physical server to another. This minimizes the risk of service interruption during the O&M process.

Before upgrading a physical server, system maintenance engineers can migrate VMs from this physical server to other servers. This minimizes the risk of service interruption during upgrade. After the upgrade is complete, migrate the VMs back to the original physical server.

System maintenance engineers can migrate VMs from a light-loaded server to other servers and then power off the server to reduce service operation costs.

In summary, if FusionCompute is used and reasonably configured, the system and service operation reliability can be significantly improved.

7.2 FusionCompute Planning and Deployment

7.2.1 Installation Preparation and Network Planning

7.2.1.1 Installation Preparation

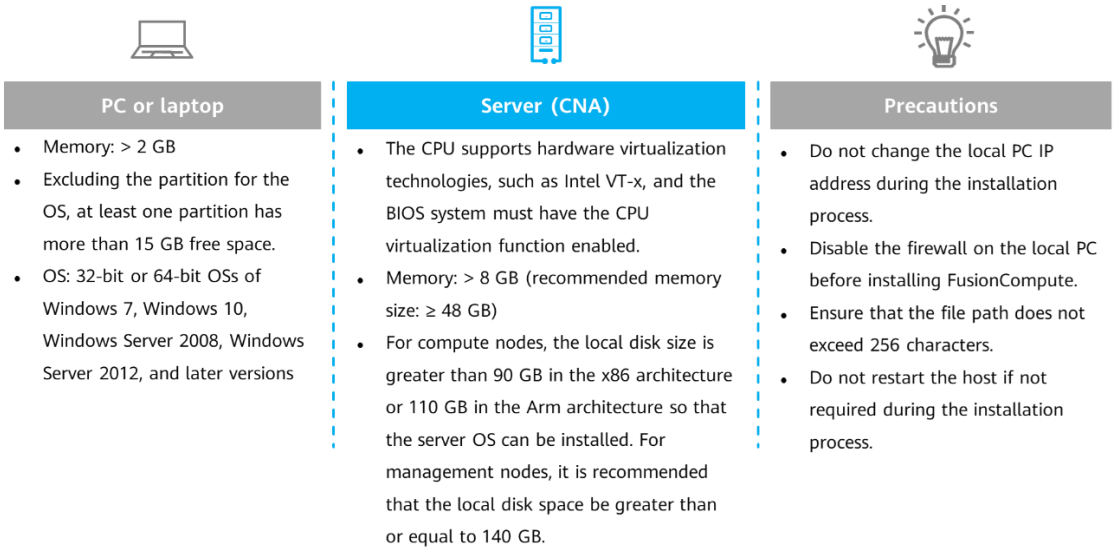


Figure 7-8 Installation preparation

Ensure that the local PC meets the FusionCompute installation requirements. Figure 7-8 shows the detailed requirements.

In addition, the software packages required for the installation can be downloaded from https://support.huawei.com/enterprise/en/index.html.

7.2.1.2 Network Planning

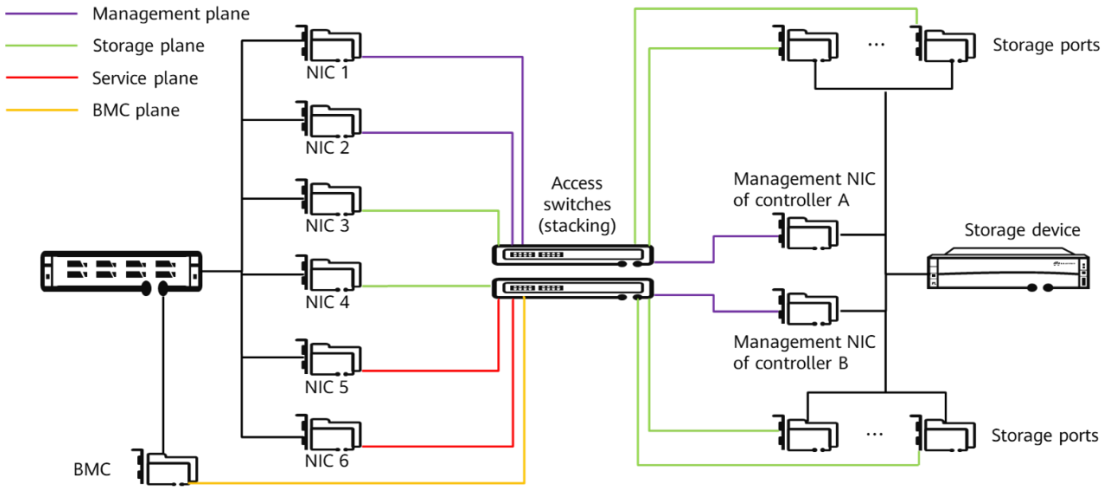


Figure 7-9 FusionCompute network planning

As shown in Figure 7-9, the FusionCompute network planning includes the following four network planes:

1: BMC plane

This plane is used by the baseboard management controller (BMC) network port on a host. It enables remote access to the BMC system of a server.

2. Management plane

This plane is used by the management system to manage all nodes in a unified manner. All nodes communicate on this plane. All nodes communicate on this plane, which provides the following IP addresses:

- Management IP addresses of all hosts, that is, IP addresses of the management network ports on hosts
- IP addresses of management VMs
- IP addresses of storage device controllers

3. Storage plane

This plane is used for the communication between hosts and storage units of storage devices. This plane provides the following IP addresses:

- Storage IP addresses of all hosts, that is, IP addresses of the storage network ports on hosts
- Storage IP addresses of storage devices

4. Service plane

- This plane is used by user VMs.

7.2.2 Installation Process and Deployment Solution

7.2.2.1 Installation Process

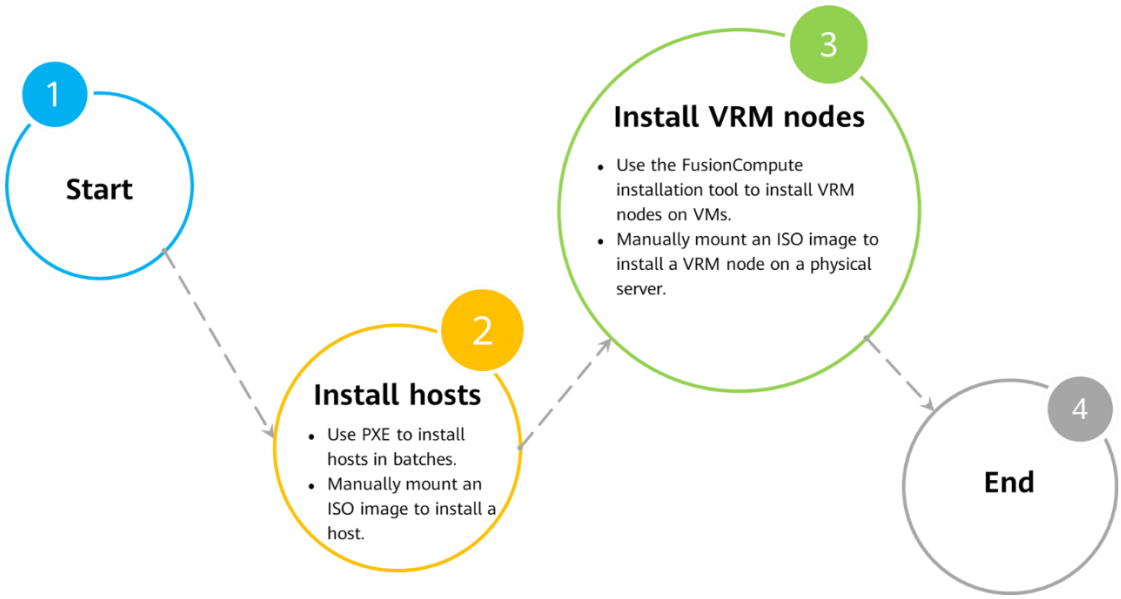


Figure 7-10 FusionCompute installation process

Figure 7-10 shows the FusionCompute installation process. In brief, you need to install CNA hosts on physical servers and then VRM nodes.

If there are a large number of CNA hosts, for example, dozens or hundreds of CNA hosts, you are advised to install the hosts in batches in pre-boot execution environment (PXE) mode. If there are a small number of CNA hosts, for example, several CNA hosts, you are advised to manually install the hosts by mounting ISO images.

After the CNA hosts are installed, install the VRM nodes. You can use the FusionCompute installation tool to deploy the VRM nodes on VMs, or manually mount ISO images to deploy the VRM nodes on physical servers.

7.2.2.2 Deployment Solution

Node	Remarks
VRM	A management node that supports centralized management of the virtual resources through a management interface.
CNA	A physical server that provides compute resources for FusionCompute. A host also provides storage resources when local disks are used.

Node	Deployment Mode	Deployment Principle
Host	Deployed on physical servers	Multiple hosts can be deployed based on customer requirements for compute resources. A host also provides storage resources when local disks are used for storage. When VRM nodes are deployed on VMs, a host must be specified for creating a VRM VM. If a small number of hosts, for example, fewer than 10 hosts, are deployed, you can add all the hosts to the management cluster to provide user services. If a large number of hosts are deployed, you are advised to add the hosts providing different user services to multiple service clusters to facilitate service management. To optimize compute resource utilization of each cluster, you are advised to configure the same types of DVSs and data stores for hosts in the same cluster.
VRM	Deployed on VMs	If the VRM nodes are deployed on VMs, you need to select two hosts in the management cluster and deploy the active and standby VRM VMs on these hosts. You are advised to deploy VRM nodes on VMs.
	Deployed on physical servers	If the VRM nodes are deployed on physical servers, the active and standby VRM nodes must be deployed on different physical servers.

Figure 7-11 FusionCompute deployment solution

Use the FusionCompute installation tool to install FusionCompute, including hosts and VRM nodes. If the active/standby mode is used, the active and standby nodes must be deployed on different hosts.

7.2.2.3 Logical View of VRM Nodes Deployed on VMs

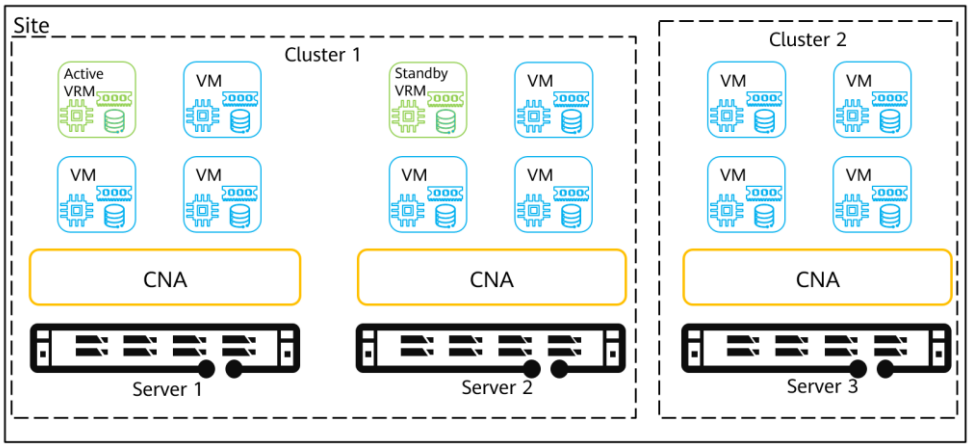


Figure 7-12 Logical view of VRM nodes deployed on VMs

As shown in Figure 7-12, VRM nodes are deployed on VMs in CNA hosts, which applies to scenarios where the FusionCompute scale is not large. To improve the availability of the FusionCompute system, the VRM nodes are deployed in active/standby mode. The active and standby VRM nodes are deployed on different CNA hosts to ensure continuous VRM availability.

7.2.2.4 Logical View of VRM Nodes Deployed on Physical Servers

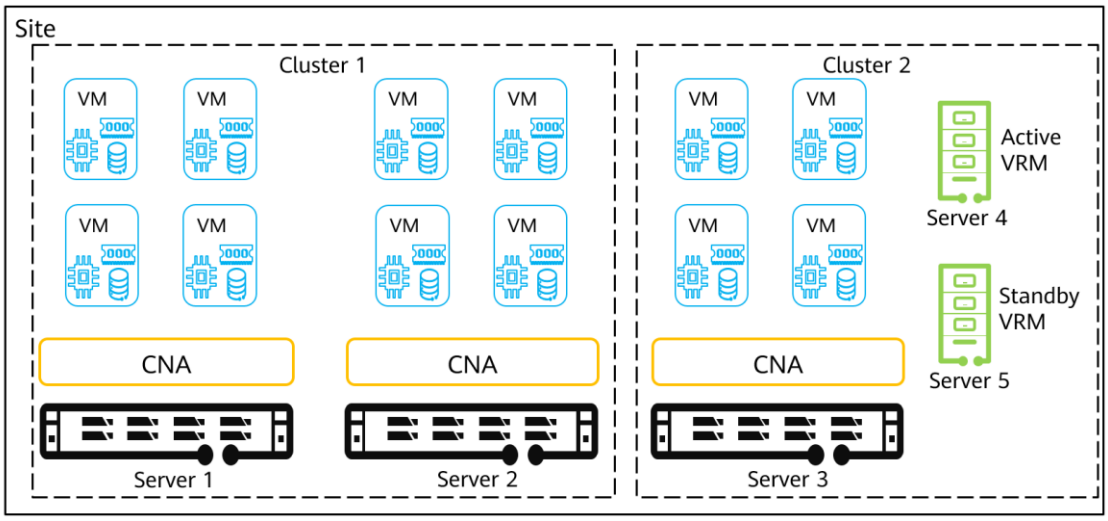


Figure 7-13 Logical view of VRM nodes deployed on physical servers

As shown in Figure 7-13, VRM nodes are deployed on physical servers. In this case, the VRM nodes and the CNA hosts are independent of each other. The physical servers where the VRM nodes reside ensure reliability of the active and standby VRM nodes.

7.3 Quiz

What are the deployment principles for hosts in FusionCompute?

8 Huawei Virtualization Platform Management and Usage

FusionCompute is a cloud OS software product that virtualizes hardware resources and centrally manages the virtual resources, service resources, and user resources. This chapter describes the compute, storage, and network virtualization features, as well as platform management and usage of FusionCompute.

8.1 Introduction to FusionCompute Compute Virtualization

8.1.1 FusionCompute Compute Virtualization Features

8.1.1.1 VM Resource Management – Online CPU and Memory Adjustment

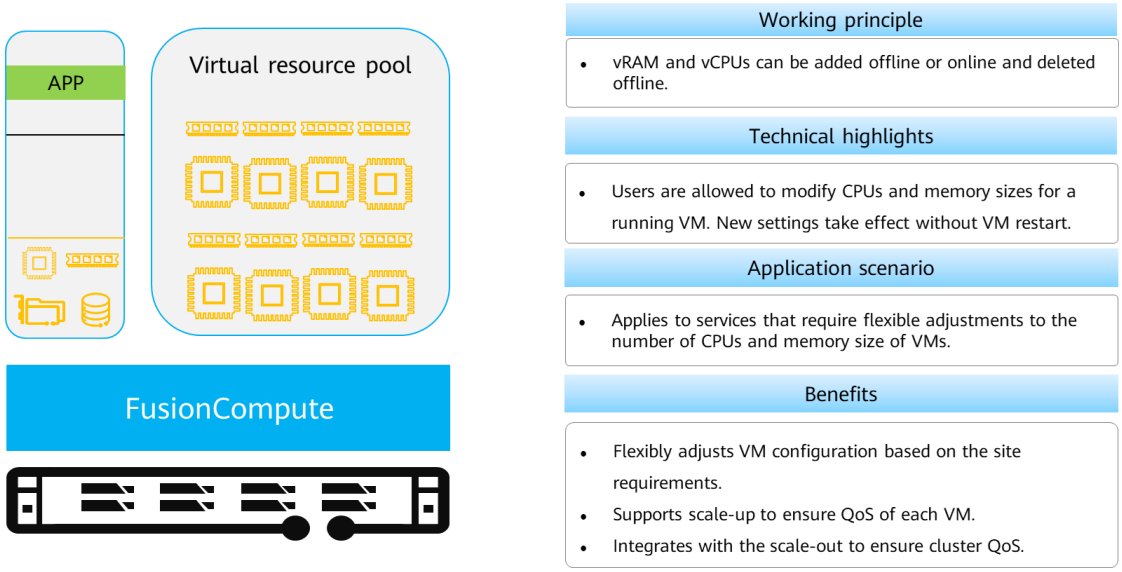


Figure 8-1 Online adjustment of VM CPU and memory resources

FusionCompute supports dynamic adjustment of VM resources. Users can dynamically adjust resource usage based on the changing workloads. FusionCompute supports online CPU and memory size modification for VMs running Linux OSs and online memory size modification for VMs running Windows OSs. New settings take effect after VMs are restarted. For example, when a user is using a VM, the CPU and memory usage reaches 75%, which may affect user experience or even affect the normal use of the VM. In this case, the VM resource hot-add feature can be used to add CPU and memory resources to

the VM online. This feature allows the resource usage to be quickly reduced to the normal level.

8.1.1.2 VM Resource Management – CPU QoS

The CPU QoS ensures allocation of compute resources for VMs and prevents cross-VM resource contention caused by service requirements. In short, it increases resource utilization and reduces costs.

When VMs are created, the CPU QoS is specified based on to-be-deployed services. The CPU QoS determines VM computing capabilities. The system ensures the CPU QoS of VMs by ensuring the minimum compute resources and resource allocation priority.

CPU QoS is determined by the following aspects:

1. CPU quota

CPU quota defines the proportion based on which CPU resources to be allocated to each VM when multiple VMs compete for the physical CPU resources.

For example, three VMs (A, B, and C) run on the host that uses a single-core physical CPU with 2.8 GHz frequency, and their quotas are set to 1000, 2000, and 4000, respectively. When the CPU workloads of the VMs are heavy, the system allocates CPU resources to the VMs based on the CPU quotas. Therefore, VM A with 1000 CPU quota can obtain a computing capability of 400 MHz. VM B with 2000 CPU quota can obtain a computing capability of 800 MHz. VM C with 4000 CPU quota can obtain a computing capability of 1600 MHz. (This example explains the concept of CPU quota and the actual situations are more complex.)

The CPU quota takes effect only when resource contention occurs among VMs. If the CPU resources are sufficient, a VM can exclusively use physical CPU resources on the host if required. For example, if VMs B and C are idle, VM A can obtain all the 2.8 GHz computing capability.

2. CPU reservation

CPU reservation defines the minimum CPU resources to be allocated to each VM when multiple VMs compete for physical CPU resources.

If the computing capability calculated based on the CPU quota of a VM is less than the CPU reservation value, the system allocates the computing capability to the VM according to the CPU reservation value. The offset between the computing capability calculated based on the CPU quota and the CPU reservation value is deducted from computing capabilities of other VMs based on their CPU quotas and is added to the VM.

If the computing capability calculated based on the CPU quota of a VM is greater than the CPU reservation value, the system allocates the computing capability to the VM according to the CPU quota.

For example, three VMs (A, B, and C) run on the host that uses a single-core physical CPU with 2.8 GHz frequency, their quotas are set to 1000, 2000, and 4000, respectively, and their CPU reservation values are set to 700 MHz, 0 MHz, and 0 MHz, respectively. When the CPU workloads of the three VMs are heavy:

According to the VM A CPU quota, VM A should have obtained a computing capability of 400 MHz. However, its CPU reservation value is greater than 400 MHz. Therefore, VM A obtains a computing capability of 700 MHz according to its CPU reservation value.

The system deducts the offset (700 MHz minus 400 MHz) from VMs B and C based on their CPU quotas.

VM B obtains a computing capability of 700 (800 minus 100) MHz, and VM C obtains a computing capability of 1400 (1600 minus 200) MHz.

The CPU reservation takes effect only when resource contention occurs among VMs. If the CPU resources are sufficient, a VM can exclusively use physical CPU resources on the host if required. For example, if VMs B and C are idle, VM A can obtain all the 2.8 GHz computing capability.

3. CPU limit

CPU limit defines the upper limit of physical CPUs that can be used by a VM. For example, if a VM with two virtual CPUs has a CPU limit of 3 GHz, each virtual CPU of the VM can obtain a maximum of 1.5 GHz compute resources.

8.1.1.3 VM Resource Management – Host Memory Overcommitment

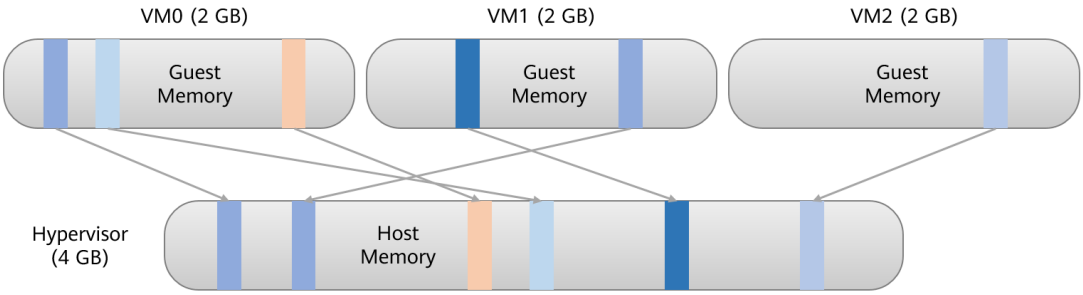


Figure 8-2 Memory overcommitment

In FusionCompute, Host Memory and Guest Memory are not in a one-to-one relationship. Host memory resources can be overcommitted to VMs, which is implemented by using the memory overcommitment technology. FusionCompute allows a server to provide virtual memory that can be larger than the server's physical memory size using various memory technologies, such as memory ballooning, memory sharing, and memory swapping.

As shown in Figure 8-2, the physical memory is 4 GB, while the memory of the three upper-layer guest OSs reaches 6 GB.

8.1.1.4 VM Resource Management – Memory Overcommitment

Memory overcommitment allows VMs to use more memory than the total physical memory of the server by leveraging specific technologies to improve VM density.

Memory overcommitment technologies include memory ballooning, memory swapping, and memory sharing. Generally, the three technologies need to be applied together and take effect at the same time.

Memory sharing: Multiple VMs share the memory page on which the data content is the same. As shown in Figure 8-3, the physical host provides 4 GB physical memory for the hypervisor and allocates the memory to three VMs. The three VMs read data from the same physical memory. According to the memory virtualization implementation principles, the hypervisor maps this memory segment to different VMs. To prevent any VM from modifying data in this memory segment, all VMs have only the read permission on this memory segment. If VMs need to write data to this memory segment, the

hypervisor needs to create a memory segment for mapping. With the memory sharing technology, 6 GB of virtual memory can be allocated to VMs based on 4 GB of physical memory.

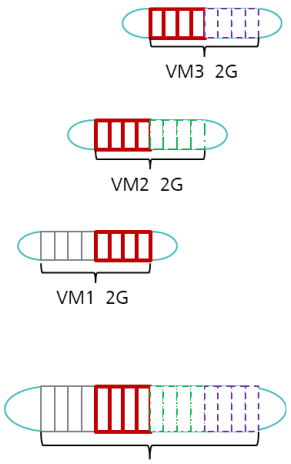


Figure 8-3 Memory overcommitment

Memory ballooning: The system automatically reclaims the unused memory from a VM and allocates it to other VMs to use. Applications on the VMs are not aware of memory reclamation and allocation. The total amount of the memory used by all VMs on a physical server cannot exceed the physical memory of the server, as shown in Figure 8-4.

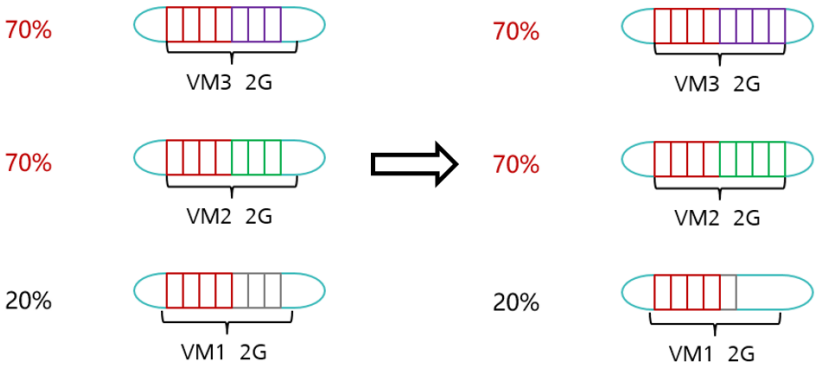


Figure 8-4 Memory ballooning

Each of the three VMs has 2 GB virtual memory. The memory usage of VM 1 is only 20%, and the memory usages of VM 2 and VM 3 are 70%. The system automatically maps physical memory allocated to VM 1 to VM 2 and VM 3 in the background to relieve the memory pressure.

Memory swapping: External storage is virtualized into memory for VMs to use. Data that is not used temporarily is stored to external storage. If the data needs to be used, it is exchanged with the data reserved on the memory, as shown in Figure 8-5.

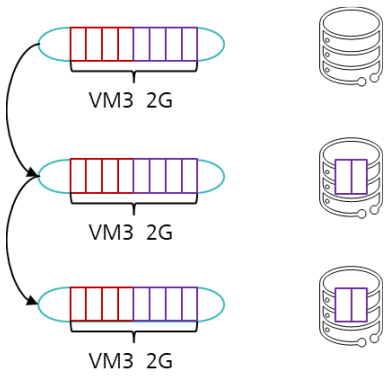


Figure 8-5 Memory swapping

Memory swapping is similar to the functions of the virtual memory of Windows and the swap partition of Linux. They simulate storage as memory to store the data that has been called to the memory but is seldom used on disks. When the data is used, the data will be called back to the memory.

In some virtualization products, such as FusionCompute, memory overcommitment is configured based on clusters. After memory overcommitment is enabled, the memory overcommitment policy replaces the physical memory allocation policy. When the memory is sufficient, VMs can use all physical memory. If the memory is insufficient, the system schedules memory resources based on the memory overcommitment policies by using memory overcommitment technologies to release free memory.

Memory overcommitment reduces customers' costs by:

- Increasing VM density when the memory size of compute nodes is fixed.
- Eliminating the need for storage devices when the VM density of compute nodes is fixed.

8.1.1.5 VM HA

Before introducing the HA feature, let's begin with a story about Kai-Fu Lee.

In the early years, Dr. Kai-Fu Lee worked for Apple, specializing in the research and development of new products.

At one time, Kai-Fu Lee and the CEO of the company, Mr. Sculley, were invited by America's most popular morning TV show, "Good Morning America." The TV station communicated with Apple in advance and hoped that they could demonstrate Apple's latest speech recognition system on live TV.

However, the success rate of the system was about 90%, and Mr. Sculley hoped to increase the success rate to 99%.

Kai-Fu Lee did not modify the program of the system, but prepared two identical computers. If one computer becomes faulty, the system will be immediately switched over to the other computer. This way, there is just a possibility of 1% (10% x 10%) that both the computers become faulty, and the success rate of the system reaches 99%.

This story shows us the basic principles of HA. The cluster technology is used to overcome the limitations of a single physical host, thereby preventing service interruption or reducing system downtime. HA in virtualization is actually the compute-layer HA, that is,

VM HA. If a compute node becomes faulty, the other compute node in the cluster can be automatically started.

A virtualization cluster usually uses shared storage. A VM consists of configuration files and data disks. The data disks are stored on shared storage, and the configuration files are stored on the compute nodes. If a compute node becomes faulty, the virtualization management system (such as vCenter and VRM) rebuilds the fault VM on other nodes based on the recorded VM configuration information.

During HA, the following issues need to be addressed:

1. Detecting host faults

2. Handling VM startup failures

First, let's look at the first issue. To check whether a compute node is faulty, the administrator needs to periodically establish communication with all nodes in the cluster. If the communication with a node fails, the node may be faulty. Use Huawei FusionCompute as an example. The heartbeat mechanism between the CNA host and VRM enables VRM to check whether CNA is faulty. The detailed process is as follows:

- The heartbeat mechanism runs in a process or service on the CNA host.
- The host sends heartbeat messages to VRM at an interval of 3s. If the host does not send heartbeat messages to VRM within 30s for ten consecutive times, the host is set to the faulty state. In this case, the "Heartbeat Communication Between the Host and VRM Interrupted" alarm is generated on the FusionCompute portal.
- A timeout mechanism is used when the host reports heartbeat messages to VRM. The timeout interval for socket connecting, receiving, and sending is 10s. If the VRM service or network is abnormal, timeout may occur. The timeout log is printed with a 13-second timestamp (Timeout detection interval 3s + Socket timeout period 10s = 13s).
- After receiving a heartbeat message from a host, VRM sets the heartBeatFreq variable to 10 (the default value is 10, which can be changed by modifying the configuration file). The detection thread decreases the value by 1 every 3s and checks the current value of this parameter. If the value is less than or equal to 0, the system regards that the host is abnormal, reports an alarm on the FusionCompute portal, and sends a host exception message to VRM. VRM then determines whether to trigger VM HA.

Next, let's look at the second issue. When a VM is started on another host, services on the VM may fail to automatically start, and even the OS may fail to start. Service recovery on the VM may fail or take a long time. In this case, HA on the service plane is needed. If the active VM is faulty or cannot be restored, services will be recovered on the standby VM using HA technologies such as the floating IP address and Keepalived.

Therefore, VM HA is usually used together with application-layer HA to improve the service recovery success rate and efficiency.

In addition to the preceding issues, how to prevent split-brain needs to be considered. Split-brain may occur when the shared storage is written by two VMs at the same time. Before HA is implemented, the management system uses the heartbeat mechanism to detect whether the compute node is faulty. If only the heartbeat network is faulty, the management system may incorrectly determine that the compute node is faulty. In this case, split-brain may occur. Therefore, before starting a VM, the system detects whether

the corresponding storage is being written. If yes, it indicates that the host may be not faulty. In this case, the system does not start the VM, but displays a message indicating that the HA is successful.

As shown in Figure 8-6, if a node in the cluster becomes faulty, VMs on this node are automatically migrated to an alternative host that is running properly.

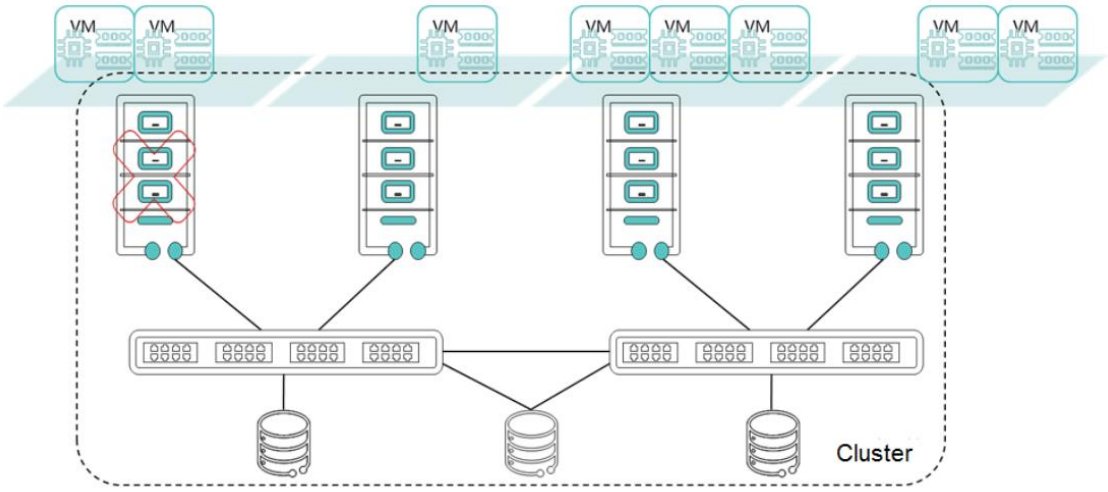


Figure 8-6 HA diagram

8.1.1.6 Dynamic Resource Scheduling

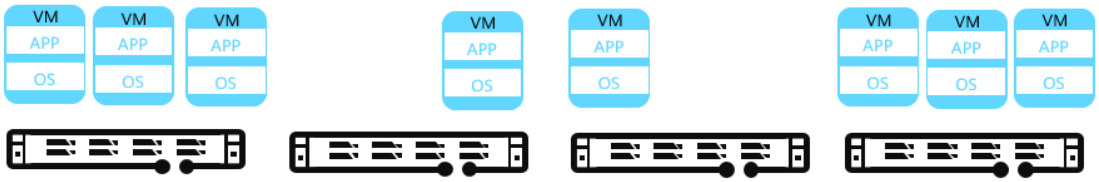


Figure 8-7 Dynamic resource scheduling

Dynamic resource scheduling (DRS) is also called compute resource scheduling automation. It uses the intelligent load balancing and scheduling algorithms and dynamic power management function to periodically monitor the loads on hosts in a cluster and migrate VMs between the hosts based on the loads, achieving load balancing between hosts in the same cluster and minimizing system power consumption.

In a virtualization environment, load balancing is generally implemented on compute nodes based on the node CPU and memory usages. The management system monitors global resource usages during VM creation and running. In the monitoring process, it uses an intelligent resource scheduling algorithm to determine the optimal host on which the VMs can run, and migrates the VMs to this optimal host by means such as live migration, improving user experience.

As shown in Figure 8-7, there are four compute nodes in the cluster and eight VMs of the same specifications are running. When the administrator creates a VM and does not specify the host for the VM, the system automatically creates the VM on the second or third node with light load. A large number of VMs are running on the first host. After a

period of time, the system automatically detects that the load on the first host is heavy and migrates VMs on this host to other light-load hosts.

The load threshold can be specified by the system administrator or defined by the system. In Huawei FusionCompute, if the CPU and memory usage of a host exceeds 60%, VRM migrates VMs on the host to other hosts. Before the migration, the administrator can set an automatic migration task. Alternatively, the administrator can manually perform the migration after receiving a notification.

The DRS working mechanism is as follows:

- When the system is lightly loaded, the system migrates some VMs to one or more physical hosts and powers off the idle hosts.
- When the system is heavily loaded, the system starts some physical hosts and allocates VMs evenly on the hosts to ensure resource supply.
- Scheduled tasks can be set to enable different resource scheduling policies at different times based on the system running status to meet user requirements in different scenarios.

For customers, this feature optimizes resource allocation in different scenarios, reduces power consumption, and improves resource utilization.

8.1.1.7 Distributed Power Management

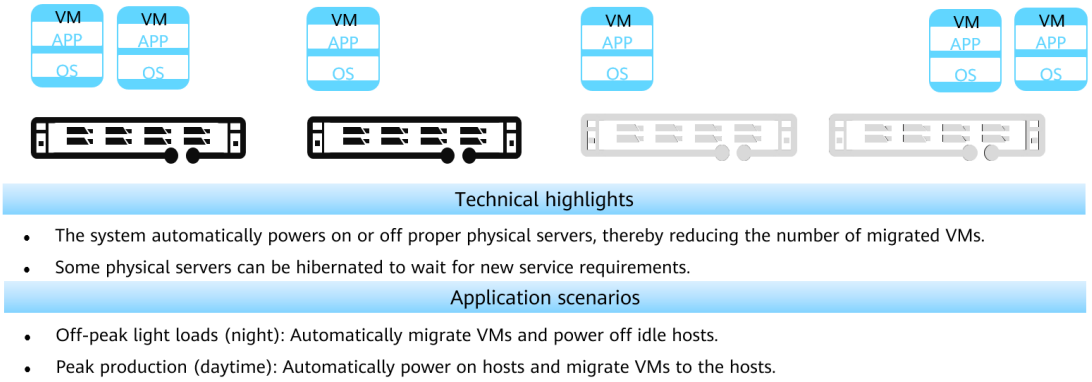


Figure 8-8 Distributed power management

Distributed power management (DPM) enables the system to periodically check resource usage on hosts in a cluster. If resources in the cluster are sufficient but service load on each host is light, the system migrates VMs to other hosts and powers off idle hosts to reduce power consumption. If the in-service hosts are overloaded, the system powers on offline hosts in the cluster to balance load among hosts.

- When DPM is enabled, automatic resource scheduling must also be enabled so that the system automatically balances VM load on hosts after the hosts are powered on.
- The time-based power management settings allow the system to manage power in different time periods based on service requirements. When services are running stably, set DPM to a low level to prevent adverse impact on services.
- With the distributed power management function enabled, the system checks resource usage in the cluster, and powers off some light loaded hosts only when the resource utilization drops below the light-load threshold over the specified time

period (40 minutes by default). Similarly, the system powers on some hosts only when the resource utilization rises above the heavy-load threshold over the specified time period (5 minutes by default). You can customize the time period for evaluating the threshold of powering on or off hosts.

DPM can be used to meet energy-saving and environmental protection requirements of enterprises. For example, if an enterprise uses the FusionCompute-based desktop cloud, all VMs used in the cloud can be migrated to certain physical hosts at 22:00 p.m. in the evening, and other physical hosts can be powered off. At 07:00 a.m. the next day, the system powers on all physical hosts and evenly distributes VMs to the physical hosts based on load balancing principles. When the physical hosts are powered off in the evening, auxiliary devices, such as the air conditioning and fresh air systems, can also be powered off to minimize energy consumption.

8.1.1.8 VM Live Migration

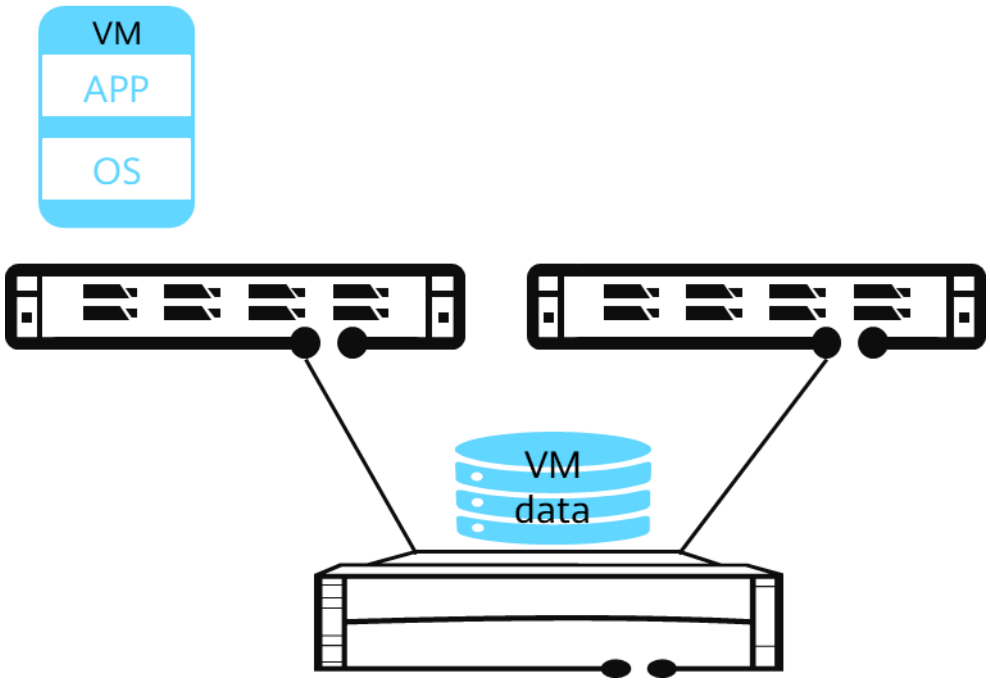


Figure 8-9 VM live migration

In FusionCompute, the VM live migration feature allows VMs to be live migrated from one physical server to another in the same cluster without interrupting services. The VM manager provides rapid memory data replication and storage data sharing technologies to ensure that the VM data before and after the live migration remains unchanged.

The VM live migration applies to the following scenarios:

- Before performing O&M operations on a physical server, system maintenance engineers need to migrate VMs from this physical server to other servers. This minimizes the risk of service interruption during the O&M process.
- Before upgrading a physical server, system maintenance engineers can migrate VMs from this physical server to other servers. This minimizes the risk of service

interruption during upgrade. After the upgrade is complete, migrate the VMs back to the original physical server.

- System maintenance engineers can migrate VMs from a light-loaded server to other servers and then power off the server to reduce service operation costs.

In addition, VM live migration is the basis of dynamic resource scheduling (DRS) and distributed power management (DPM). It can be classified into manual migration and automatic migration.

Manual migration indicates that system maintenance engineers manually migrate one VM to another server by using the VM live migration function. Automatic migration is usually based on VM resource scheduling, that is, DRS or DPM. The system automatically migrates VMs to other servers in the cluster based on the preset VM scheduling policies.

8.1.1.9 Rule Group

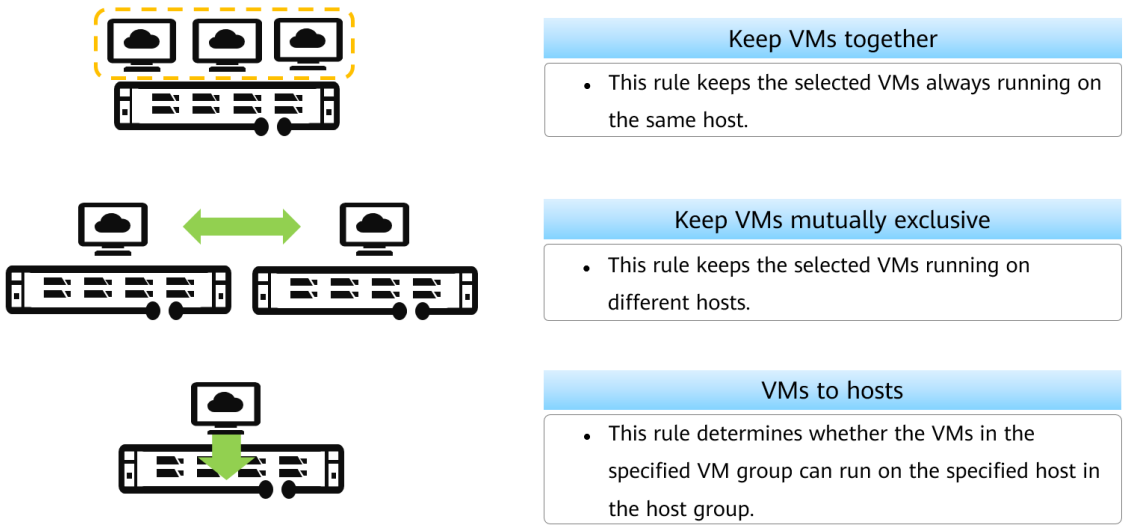


Figure 8-10 Rule group

A rule group is based on the DRS function.

DRS and DPM are part of load balancing. DRS provides a reference for system migration during load balancing.

- **Keep VMs together**: This rule keeps the selected VMs always running on the same host. One VM can be added to only one keep-VMs-together rule group.
- **Keep VMs mutually exclusive**: This rule keeps the selected VMs running on different hosts. One VM can be added to only one VM-mutually-exclusive rule group.
- **VMs to hosts**: This rule associates a VM group with a host group so that the VMs in the specified VM group can be configured to run only on the specified host in the host group.

If different rules conflict, the scheduling priorities of the rules are as follows:

- Highest priority: The rule type is **VMs to hosts**, and the rules are **Must run on hosts in group** and **Must not run on hosts in group**.
- Second priority: The rule types are **Keep VMs together** or **Keep VMs mutually exclusive**.

- Lowest priority: The rule type is **VMs to hosts**, and the rules are **Should run on host group** and **Should not run on hosts in group**.

8.2 Introduction to FusionCompute Storage Virtualization

8.2.1 Concepts Related to Storage Virtualization

8.2.1.1 Introduction to Storage Virtualization

Storage virtualization abstracts storage devices to data stores so that each VM can be stored as a group of files in a directory on a data store. A data store is a logical container that is similar to a file system. It hides the features of each storage device and provides a unified model to store VM files. The storage virtualization technology helps the system better manage virtual infrastructure storage resources with high resource utilization and flexibility.

8.2.1.2 Storage Concepts in FusionCompute

FusionCompute supports storage resources from local disks or dedicated storage devices. Dedicated storage devices are connected to hosts using network cables or fiber cables.

FusionCompute uniformly encapsulates storage units of storage resources into data stores. After storage resources are converted to data stores and associated with hosts, virtual disks can be created for VMs.

The following storage units can be encapsulated as data stores:

- Logical unit numbers (LUNs) on SAN storage, including iSCSI and FC SAN storage
- File systems on NAS devices
- Storage pools on Huawei Distributed Block Storage (FusionStorage Block)
- Local disks on hosts (virtualized)

In Huawei FusionCompute, these storage units are called storage devices, and physical storage media that provide storage space for virtualization are called storage resources, as shown in Figure 8-11.

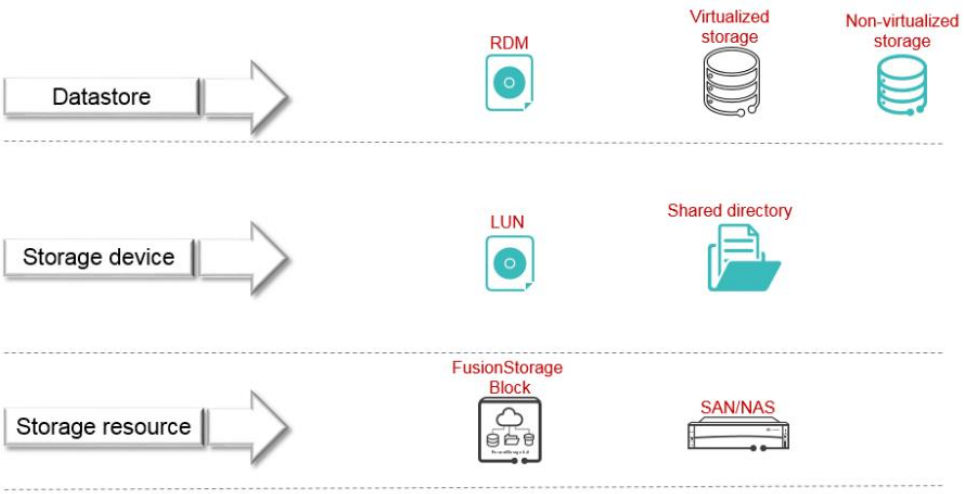


Figure 8-11 Storage concept and structure of FusionCompute

Note: When adding storage devices to FusionCompute, observe the Huawei-defined logical storage architecture and determine how devices at each logical layer are added to the system. For example, storage resources need to be manually added, and storage devices are scanned.

Before using data stores, you need to manually add storage resources. If the storage resources are IP SAN, FusionStorage Block, or NAS storage, you need to add storage ports for hosts in the cluster and use the ports to communicate with the service ports of centralized storage controller or the management IP address of FusionStorage Manager. If the storage resources are provided by FC SAN, you do not need to add storage ports.

After adding storage resources, you need to scan for these storage devices on the FusionCompute portal to add them as data stores.

Data stores can be virtualized or non-virtualized. You can use LUNs on SAN storage as data stores to be used by VMs without creating virtual disks. This process is called raw device mapping (RDM). This technology applies to scenarios requiring large disk space, for example, database server construction. RDM can be used only for VMs that run certain OSs.

8.2.1.3 FusionCompute Storage Virtualization Architecture

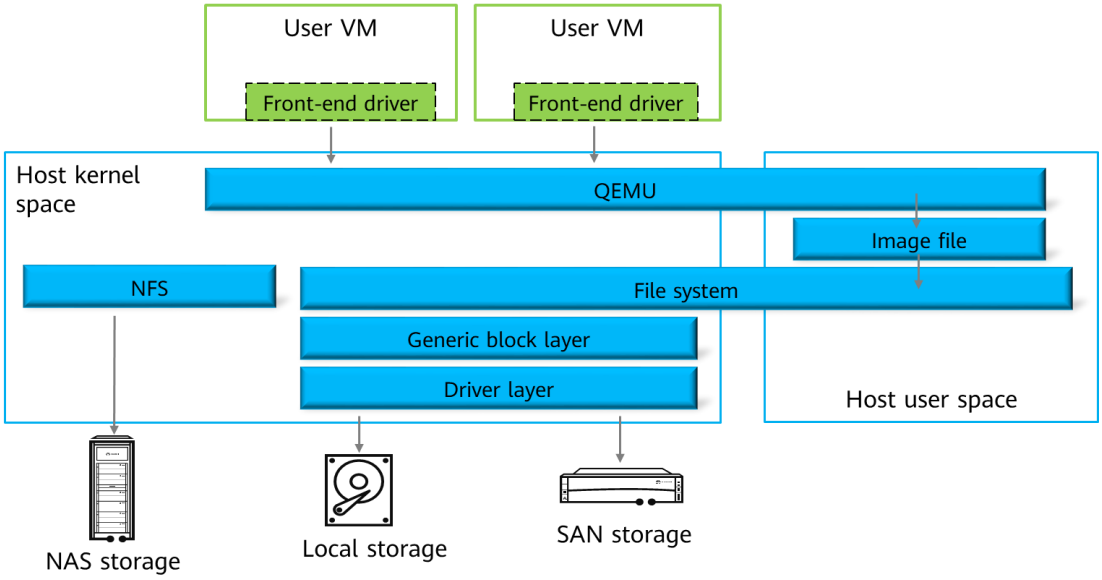


Figure 8-12 FusionCompute storage virtualization architecture

The FusionCompute storage virtualization platform consists of file systems, disk drivers, and disk tools. Block devices, such as SAN devices and local disks, are connected to servers. The block device driver layer and generic block layer offer an abstract view of the block devices and present a single storage device to hosts.

File systems are created on storage devices that can be accessed by hosts. To create a file system, the host formats storage devices, writes metadata and inode information about the file system to the storage devices, establishes mappings between files and block devices, and manages the block devices, including space allocation and reclamation. The

file system eases operation complexity by making operations on block devices invisible. VM disks are files stored in the file system.

The disk driver attaches disks to VMs only when the VMs need to use their disks. The VMs are managed through the machine emulator QEMU. The read and write I/O is received by a front-end driver, forwarded to the QEMU process, converted to read and write operations in the user-mode driver, and then written into disk files.

Attributes and data blocks are included in VM disks. The disk tool can be used to perform VM disk-related operations, including parsing disk file headers, reading and modifying disk attributes, and creating data blocks for disks.

8.2.1.4 Virtual Cluster File System - VIMS

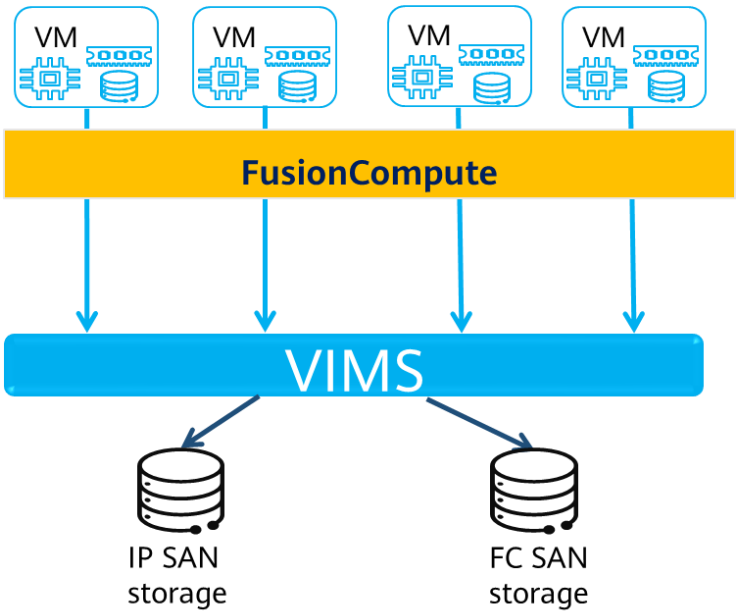


Figure 8-13 VIMS

Either a centralized or a distributed storage system forms a physical volume after the RAID or copy mechanism is used. However, in most cases, the physical volume is not mounted to upper-layer applications, for example, OSs or virtualization systems (used in this document). If the physical volume is mounted, all space is formatted by upper-layer applications. After the storage space is used up, you can add disks to expand the capacity. However, you need to reformat the physical volume after the capacity expansion, which may cause data loss. Therefore, multiple physical volumes are used as a volume group, then, the volume group is divided into multiple logical volumes (LVs). The upper-layer applications use the space of the LVs.

In cloud computing, the virtualization programs format the LVs. Vendors use different virtualization file systems. For example, VMware uses Virtual Machine File System (VMFS), and Huawei uses Virtual Image Manage System (VIMS). Both of them are high-performance cluster file systems that enable virtualization to exceed the limit of a single system and allow multiple compute nodes to access an integrated storage pool. The file system of a computing cluster ensures that no server or application software can completely control the access to the file system.

VIMS is used as an example. It is a cluster file system based on SAN storage. FusionStorage Block delivers only non-virtualized storage space. FusionCompute manages VM images and configuration files on VIMS. VIMS uses the distributed lock mechanism to ensure the data read/write consistency across the cluster.

8.2.1.5 FusionCompute Disk Technologies

After adding data stores, you can create virtual disks for VMs. Customers may have various requirements for VM disks, for example, sharing a VM disk or saving more physical space. Therefore, Huawei VM disks are classified into the following types.

1. Based on the disk type, VM disks can be classified as non-shared disks and shared disks.

- **Non-shared**: A non-shared disk can be used by only one VM.
- **Shared**: A shared disk can be used by multiple VMs.

If multiple VMs that use a shared disk write data into the disk at the same time, data may be lost. Therefore, you need to use application software to control the disk access permission.

2. Based on the configuration mode, VM disks can be classified as common disks, thin provisioning disks, and thick provisioning lazy zeroed disks.

- **Common**: In this mode, the system allocates all the configured space to the disk and zeros out the data remaining on the physical device during disk creation. The performance of the disks in this mode is better than that in the other two modes, but the creation duration may be longer than that required in the other modes.
- **Thin provisioning**: In this mode, the system allocates part of the configured disk capacity for the first time, and allocates the rest disk capacity based on the storage usage of the disk until the configured disk capacity is allocated. In this mode, data stores can be overcommitted. It is recommended that the data store overcommit rate not exceed 50%. For example, if the total capacity is 100 GB, the allocated capacity should be less than or equal to 150 GB. If the allocated capacity is greater than the actual capacity, the disk is in thin provisioning mode.
- **Thick provisioning lazy zeroed**: In this mode, the system allocates disk space based on disk capacity. However, data remaining on the physical device is zeroed out only on first data write from the VM. In this mode, the disk creation speed is faster than that in **Common** mode, and the I/O performance is medium between the **Common** and **Thin provisioning** modes. This configuration mode supports only virtualized local disks or virtualized SAN storage.

3. Based on the disk mode, VM disks can be classified as dependent disks, independent & persistent disks, and independent & non-persistent disks.

- **Dependent**: A dependent disk is included in the snapshot. Changes are written to disks immediately and permanently.
- **Independent & persistent**: In this mode, disk changes are immediately and permanently written into the disk, which is not affected by snapshots.
- **Independent & nonpersistent**: In this mode, disk changes are discarded after the VM is stopped or restored using a snapshot.

If **Independent & persistent** or **Independent & nonpersistent** is selected, the disk data will not be backed up when a snapshot is taken for the VM and will not be restored when the VM is restored using a snapshot.

After a snapshot is taken for a VM, if disks on the VM are detached from the VM and not attached to any other VMs, the disks will be attached to the VM after the VM is restored using the snapshot. However, data on the disks will not be restored.

If a disk is deleted after a snapshot is created for the VM, the disk will not be attached to the VM after the VM is restored using the snapshot.

Once set, the disk type and configuration mode cannot be changed while the disk mode can be converted.

8.2.2 FusionCompute Storage Virtualization Features

8.2.2.1 Snapshot

In virtualization, VM snapshots are similar to pictures we take in our life. A snapshot records the VM status at a certain moment and contains complete data of the VM. You can use a snapshot to restore a VM to the state at a specific time point.

Generally, VM snapshots are used to quickly restore a VM if the VM is faulty before an upgrade, patch installation, or test is performed on the VM. The VM snapshot function is implemented by the storage system. Storage Networking Industry Association (SNIA) defines a snapshot as a fully usable copy of a defined collection of data that contains an image of the data as it appeared at the point in time at which the copy was initiated. A snapshot can be a duplicate or replicate of the source data. Figure 8-14 shows the diagram of the snapshot technology.

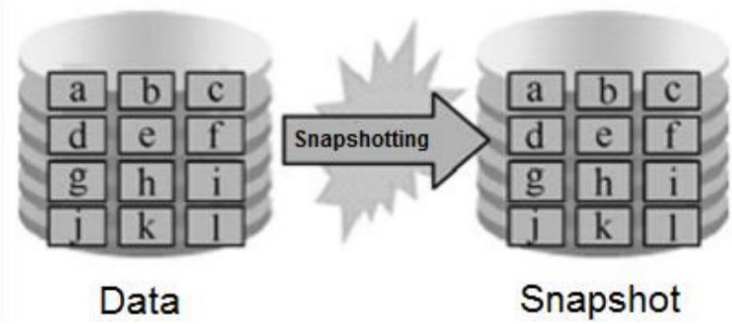


Figure 8-14 Snapshot

The snapshot technology has the following features:

Snapshots can be quickly generated and used as the data source for conventional backup and archiving, reducing or even eliminating data backup windows.

Snapshots are stored on disks and can be quickly accessed, accelerating data restoration.

Disk-based snapshots provide storage devices with flexible and frequent recovery points, enabling easy online recovery of data that is accidentally erased or damaged using snapshots of different time points.

To be specific, a snapshot creates a pointer list that indicates an address of the data to be read. When data changes, the pointer list is used to provide and replicate real-time data.

Common snapshot modes are Copy-On-Write (COW) and Redirect-On-Write (ROW). Both COW and ROW are concepts in the storage domain. Most vendors use the ROW technology to create VM snapshots. No matter whether COW or ROW is used, physical copy operations will not be performed, and only the mapping is modified. The following describes the differences between COW and ROW.

- COW

Data is recorded in data blocks. When COW is used, the system generates a new space each time a snapshot is created. Once data in a data block changes, the system copies data in the original data block to a new space and then writes new data to the original data block. The copy operation is performed before data is written. For example, there is a parking lot, and cars are parked in parking spaces. A new car (car B) can be parked in only after a car (car A) in a parking space has been moved to another parking space. If you were the owner of car A, would you feel that the parking lot management is complex when the parking lot administrator told you to make room for other cars? Why not let the new car park into a new parking space? ROW is introduced to address this issue.

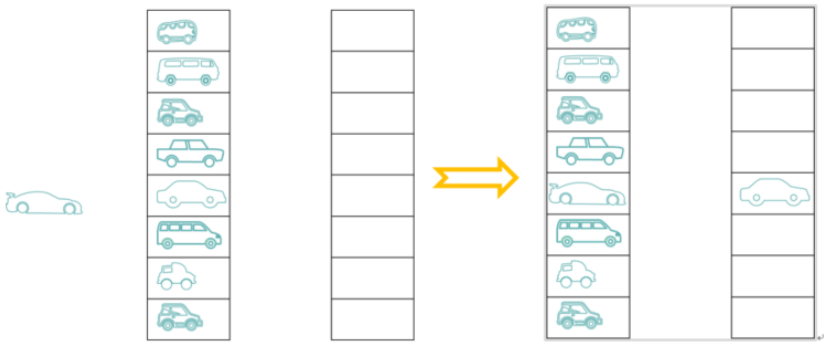


Figure 8-15 COW diagram

- ROW

Similar to COW, data is recorded in data blocks and a new space is generated when a snapshot is created. The difference from COW is as follows: if new data is written into the system, the original data remains unchanged and new data is written to a new space. Let's go back to the previous example. If a new car is parked in, the administrator can direct the car to a new parking space.

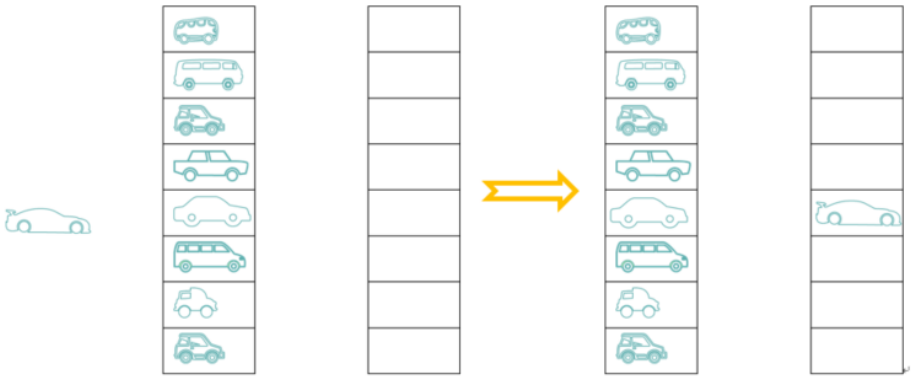


Figure 8-16 ROW diagram

COW writes data twice, whereas ROW writes data only once. Therefore, ROW has an advantage over COW in terms of creating snapshots for VMs.

Multiple snapshots can be created for a VM to form a snapshot chain. Operations on any snapshot do not affect other snapshots.

FusionCompute supports common snapshot, consistency snapshot, and memory snapshot.

Common snapshot: Only status of the dependent disk that is attached to the VM is saved

Consistency snapshot: The cache data that is not saved on the VM will be saved to the disk before a snapshot is created.

Memory snapshot: The current VM memory data and dependent disk status are saved during the snapshot creation.

8.2.2.2 Storage Live Migration

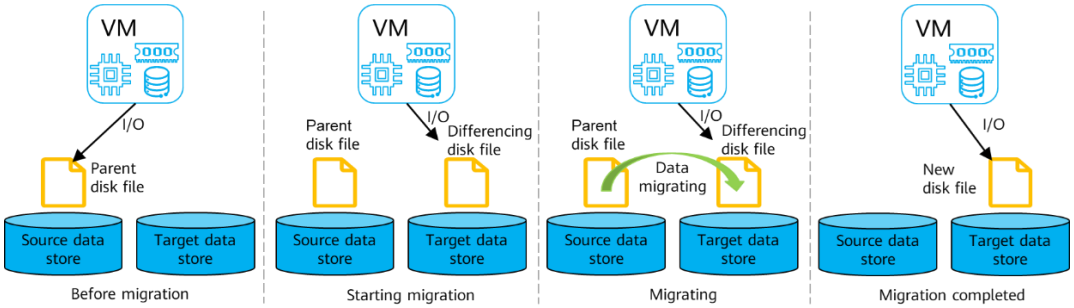


Figure 8-17 Storage live migration

FusionCompute offers cold migration and live migration for VM disks. Cold migration moves VM disks from one data store to another when the VM is stopped. Live migration moves VM disks from one data store to another without service interruption. The live migration mechanism is as follows:

- To implement live migration, the system first uses the redirect-on-write technique to write VM data to a differencing disk in the target server. Then the parent disk file becomes read-only.
- The data blocks on the parent disk are read and merged into the target differencing disk. After all data is merged, the differencing disk contains all data on the virtual disk.
- The parent disk file is removed, and the differencing disk is changed to a dynamic disk. Then, the disk in the target server can run properly.

8.2.2.3 RDM of Storage Resources

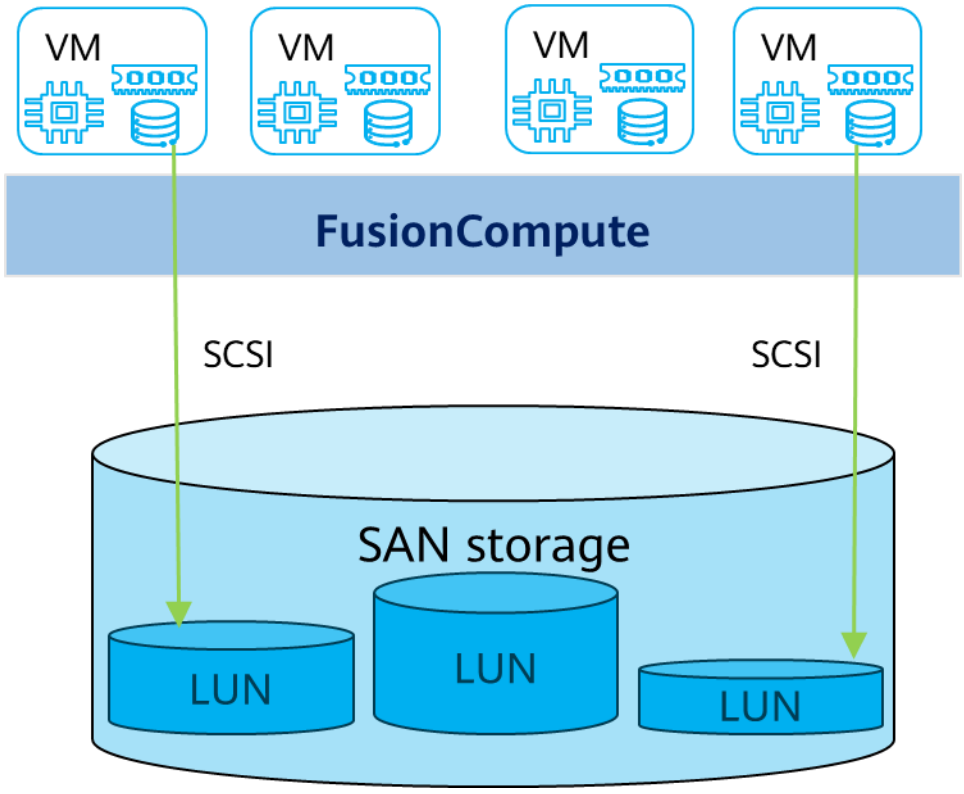


Figure 8-18 RDM of storage resources

Raw device mapping (RDM) provides a mechanism for VMs to directly access LUNs on physical storage subsystems (only Fibre Channel or iSCSI). By using physical device mapping, VMs can identify SCSI disks.

RDM can bypass a hypervisor layer to transparently transmit the SCSI commands issued by VMs to physical SCSI devices, avoiding function loss due to virtualization layer simulation.

After RDM is configured for VMs, some functions are not supported, such as linked cloning, thin provisioning, online and offline capacity expansion, incremental storage snapshot, iCache, storage live migration, storage QoS, disk backup, and VM conversion to a template.

Technical highlights

- VMs directly issue SCSI commands to operate raw devices.
- FC SAN storage and IP SAN storage support the RDM feature.

RDM applies to scenarios that require high-performance storage, such as Oracle RAC.

8.3 Introduction to FusionCompute Network Virtualization

8.3.1 Concepts Related to Network Virtualization

8.3.1.1 Development of Network Virtualization

Compute virtualization stimulates the network virtualization. In traditional data centers, a server runs an OS, connects to a switch through physical cables, and implements data exchange with different hosts, traffic control, and security control using the switch. After compute virtualization is performed, one server is virtualized to multiple virtual hosts and each virtual host has its own virtual CPU, memory, and network interface card (NIC). It is essential for virtual hosts on a single server to maintain communication, while sharing of physical equipment calls for new security isolation and traffic control. This created the demand for the virtual switching technology.

As cloud computing and virtualization become popular, layer-2 switches need to be deployed on servers and connect to VMs. In this case, virtual switches come.

Cloud computing and virtualization have advantages. Therefore, they are prevailing as a mainstream IT technology. However, an emerging technology brings with it new issues. With prosperous virtualization, it is no longer the physical server to carry services. Previously, a physical server uses at least one network cable to connect to a switch, and services running on the server share the network cable. Now, multiple VMs run on one physical server and use one network cable to carry multiple types of traffic. The new issues are how to manage various types of traffic and how to view the statuses of various types of traffic.

Previously, staff were divided into host engineers, network engineers, and program development engineers. Each of them has specific responsibilities. Currently, network engineers are responsible for virtual switches. However, virtual switches run on servers and host engineers are responsible for all components on servers. As a result, in cloud computing and virtualization, if a fault occurs on a virtual switch, neither network engineers nor host engineers can handle it. This is because virtual switches function as the real network access layer. In this case, it is necessary for both host engineers and network engineers to get familiar with the network architecture in virtualization.

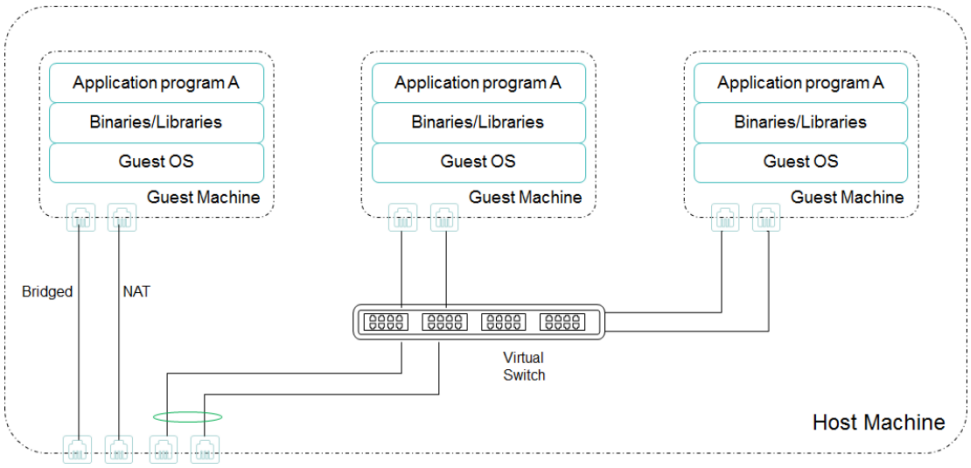


Figure 8-19 Virtual network architecture

Figure 8-19 shows a common virtual network architecture. In a personal or small-scale virtualization system, VMs are connected to physical NICs in bridged or network address translation (NAT) mode. In a large-scale virtualization system, VMs are connected to physical networks using virtual switches.

8.3.1.2 Introduction to Linux Bridge

A Linux bridge is a virtual network device that works at layer 2 and functions like a physical switch.

A bridge binds other Linux network devices and virtualizes them as ports. Binding a device to a bridge is equivalent to that a network cable connected to a terminal is inserted into the physical switch port.

In a virtualization system, the OS is responsible for interconnecting all network ports. The following figure shows the bridge-based interconnection.

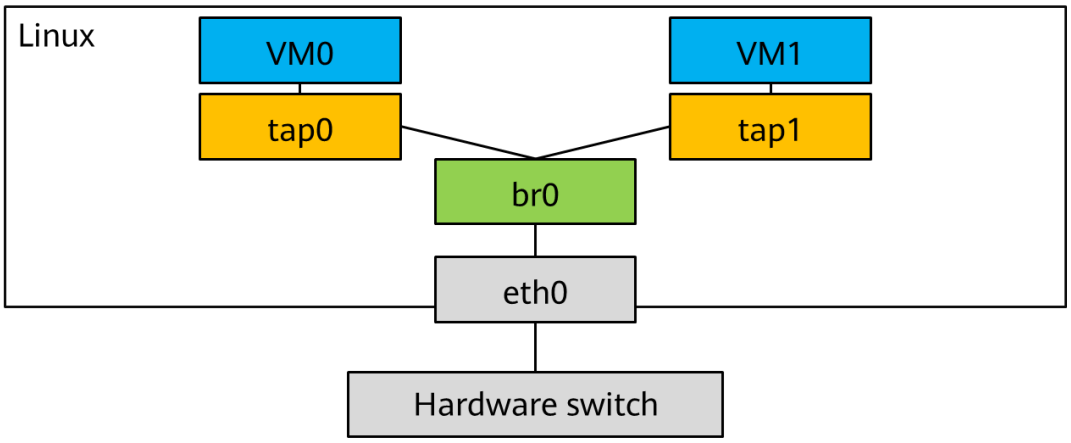


Figure 8-20 Bridge-based interconnection

The bridge **br0** binds the physical device **eth0** and the virtual devices **tap0** and **tap1**. In this case, the upper-layer network protocol stack knows only **br0** and does not need to know bridging details. When receiving a data packet, these bound devices will send the data packet to **br0** for forwarding based on the mapping between the MAC addresses and ports.

The network bridge has the following functions:

MAC learning

Initially, the network bridge has no mapping between MAC addresses and ports and sends data like a hub. When sending a data packet, the network bridge learns the MAC address and its associated port to set up a mapping between the MAC address and port (CAM table).

Packet forwarding

When sending a data packet, the network bridge obtains its destination MAC address and searches the CAM table to determine the port through which the data packet will be sent.

With only the network bridge used, VMs can communicate with external networks in bridged or NAT mode. In bridged mode, the network bridge functions as a switch, and

the virtual NIC is connected to a port of the switch. In NAT mode, the network bridge functions as a router, and the virtual NIC is connected to a port of the router.

When the virtual NIC is connected to a port of the switch, the virtual NIC and the network bridge, with the same IP address configuration, communicate with each other in broadcast mode. When the virtual NIC is connected to a port of the router, the virtual NIC and the network bridge are configured with IP addresses that belong to different network segments. In this case, the system automatically generates a network segment, the virtual NIC communicates with other networks including the network bridge in Layer 3 routing and forwarding mode, and address translation is conducted on the network bridge. In the Linux system, the default network segment is 192.168.122.0/24, as shown in Figure 8-21.

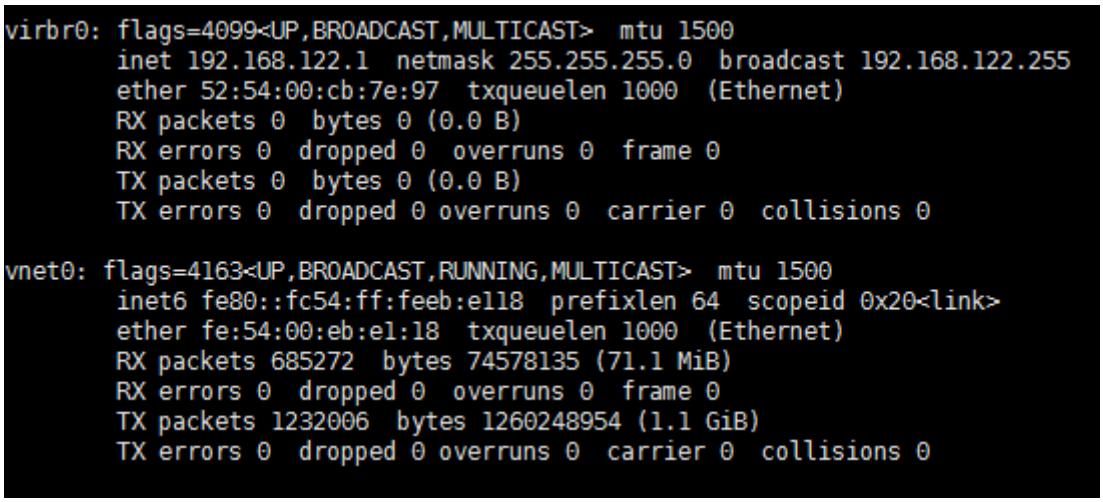


Figure 8-21 Linux bridge IP address in NAT mode

NAT is used for address translation. In NAT mode, when a VM communicates with an external network through the NAT gateway, which is **virbr0** in the Figure 8-21, the source IP address of the IP packet is translated into the IP address of the physical network bridge and a record is generated accordingly. When the external network accesses the VM, the NAT gateway forwards data packets to the VM based on the record.

NAT is widely used and has the following advantages:

- When available IP addresses in the network segment for the physical network bridge are insufficient, a new network segment can be added.
- The source IP address can be concealed. When a VM accesses an external network, the IP address of the VM is translated on the NAT gateway. Therefore, the external network will not know the IP address, protecting VM security.
- Load balancing is achieved. NAT provides the redirecting function. When multiple VMs with the same applications are deployed in active/standby mode, NAT can translate their IP addresses into one IP address used for communication with external networks. In addition, load balancing software is used for evenly distributing service access.

8.3.1.3 Introduction to OVS

The bridge and NAT are suitable for personal or small-scale systems. In bridged mode, the statuses of virtual NICs cannot be viewed and the traffic on virtual NICs cannot be

monitored. The bridged mode is supported only on the GRE tunnel. In addition, the network bridge does not support software-defined networking (SDN), which is widely used currently. Therefore, in a large-scale system, a virtual switch is used for VMs to communicate. A virtual switch acts as an upgraded network bridge, removing the defects of the network bridge.

Currently, each virtualization vendor has its own virtual switching product, such as VMware vSwitch, Cisco Nexus 1000V, and Huawei DVS. The following describes open-source Open vSwitch.

Open vSwitch (OVS) is an open-source, high-quality, and multi-protocol virtual switch. It is developed by Nicira Networks using the open-source Apache2.0 license protocol. Its main codes are portable C codes. Open vSwitch is designed to enable massive network automation through programmatic extension and also support the standard management interfaces and protocols, such as NetFlow, sFlow, SPAN, RSPAN, CLI, LACP, and 802.1ag. Open vSwitch supports multiple Linux virtualization technologies, such as Xen and KVM.

The OVS official website describes Why-OVS as follows:

- The mobility of state

  All network state associated with a network entity (say a virtual machine) should be easily identifiable and migratable between different hosts. This may include traditional "soft state" (such as an entry in an L2 learning table), L3 forwarding state, policy routing state, ACLs, QoS policy, monitoring configuration (e.g. NetFlow, IPFIX, sFlow), etc. Open vSwitch has support for both configuring and migrating both slow (configuration) and fast network state between instances. For example, if a VM migrates between end-hosts, it is possible to not only migrate associated configuration (SPAN rules, ACLs, QoS) but any live network state (including, for example, existing state which may be difficult to reconstruct). Further, Open vSwitch state is typed and backed by a real data-model allowing for the development of structured automation systems.

- Responding to network dynamics

  Virtual environments are often characterized by high-rates of change. VMs coming and going, VMs moving backwards and forwards in time, changes to the logical network environments, and so forth. Open vSwitch supports a number of features that allow a network control system to respond and adapt as the environment changes. This includes simple accounting and visibility support such as NetFlow, IPFIX, and sFlow. But perhaps more useful, Open vSwitch supports a network state database (OVSDB) that supports remote triggers. Therefore, a piece of orchestration software can "watch" various aspects of the network and respond if/when they change. This is used heavily today, for example, to respond to and track VM migrations. Open vSwitch also supports OpenFlow as a method of exporting remote access to control traffic. There are a number of uses for this including global network discovery through inspection of discovery or link-state traffic (e.g. LLDP, CDP, OSPF, etc.).

- Maintenance of logical tags

  Distributed virtual switches often maintain logical context within the network through appending or manipulating tags in network packets. This can be used to

uniquely identify a VM (in a manner resistant to hardware spoofing), or to hold some other context that is only relevant in the logical domain. Much of the problem of building a distributed virtual switch is to efficiently and correctly manage these tags. Open vSwitch includes multiple methods for specifying and maintaining tagging rules, all of which are accessible to a remote process for orchestration. Further, in many cases these tagging rules are stored in an optimized form so they don't have to be coupled with a heavyweight network device. This allows, for example, thousands of tagging or address remapping rules to be configured, changed, and migrated. In a similar vein, Open vSwitch supports a GRE implementation that can handle thousands of simultaneous GRE tunnels and supports remote configuration for tunnel creation, configuration, and tear-down. This, for example, can be used to connect private VM networks in different data centers.

- Hardware integration

  Open vSwitch's forwarding path (the in-kernel datapath) is designed to be amenable to "offloading" packet processing to hardware chipsets, whether housed in a classic hardware switch chassis or in an end-host NIC. This allows for the Open vSwitch control path to be able to both control a pure software implementation or a hardware switch. There are many ongoing efforts to port Open vSwitch to hardware chipsets. These include multiple merchant silicon chipsets (Broadcom and Marvell), as well as a number of vendor-specific platforms. The advantage of hardware integration is not only performance within virtualized environments. If physical switches also expose the Open vSwitch control abstractions, both bare-metal and virtualized hosting environments can be managed using the same mechanism for automated network control.

In many ways, Open vSwitch targets a different point in the design space than previous hypervisor networking stacks, focusing on the need for automated and dynamic network control in large-scale Linux-based virtualization environments.

The OVS can be used to transmit traffic between VMs and implement communication between VMs and the external network.

8.3.1.4 FusionCompute Distributed Virtual Switch

The virtual switch can be a common virtual switch or a distributed virtual switch (DVS). A common virtual switch runs only on a single physical host. All network configurations apply only to VMs on the physical host. Distributed virtual switches are deployed on different physical hosts. You can use the virtualization management tool to configure distributed virtual switches in a unified manner. A distributed virtual switch is required for VM live migration.

Huawei virtualization products use distributed virtual switches. This section describes Huawei distributed virtual switches.

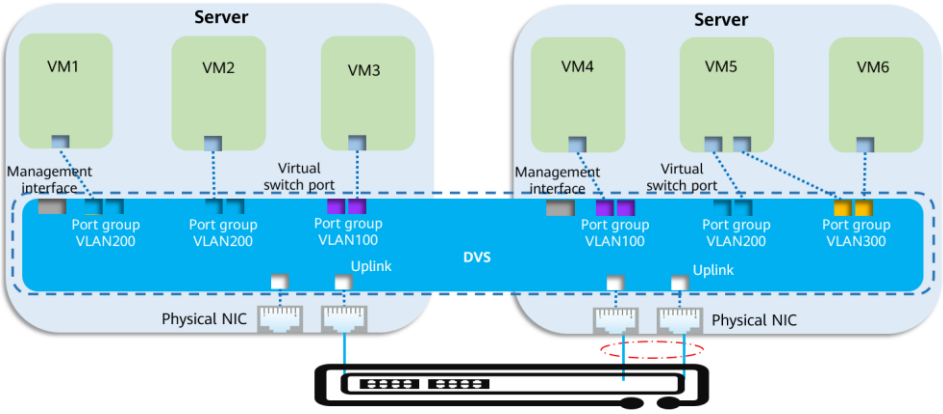


Figure 8-22 FusionCompute DVS

Huawei virtual switches are used for centralized virtual switching and use a unified portal for configuration management, simplifying user management. The virtual switch on each physical server provides VMs with capabilities, such as layer-2 communication, isolation, and QoS.

8.3.1.5 Virtual Switching Model

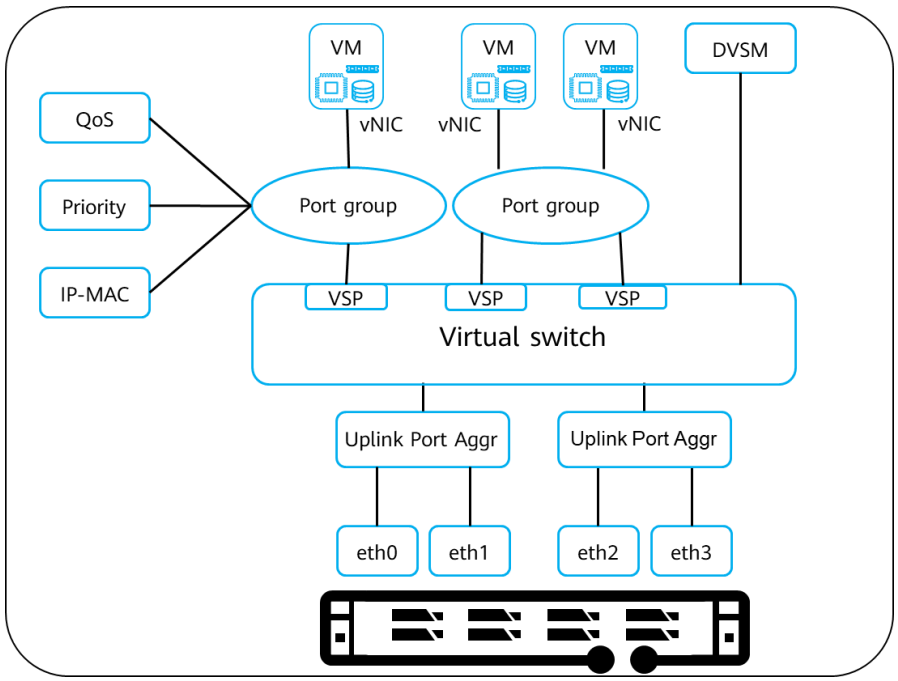


Figure 8-23 DVS model

The DVS model has the following characteristics:

1. Multiple DVSs can be configured, and each DVS can serve multiple CNA nodes in a cluster.
2. A DVS provides several virtual switch ports (VSP) with their own attributes, such as the rate. The ports with the same attributes are assigned to a port group for management. The port groups with the same attributes use the same VLAN.
3. Different physical ports can be configured for the management plane, storage plane, and service plane. An uplink port or an uplink port aggregation group can be

configured for each DVS to enable external communication of VMs served by the DVS. An uplink aggregation group comprises multiple physical NICs working based on load-balancing policies.

4. Each VM provides multiple virtual NIC (vNIC) ports, which connect to VSPs of the switch in one-to-one mapping.
5. A server allowing layer-2 migration in a cluster can be specified to create a virtual layer-2 network based on service requirements and configure the VLAN used by this network.

The VM port attributes setting can be simplified by configuring port group attributes, including security and QoS. The port group attributes setting has no impact on the proper running of VMs.

A port group consists of multiple ports with the same attributes. The VM port attributes setting can be simplified by configuring port group attributes, including bandwidth QoS, layer-2 security attributes, and VLAN. Port group attribute changes do not affect VM running.

An uplink connects the host and the DVS. Administrators can query information about an uplink, including its name, ratio, mode, and status.

Uplink aggregation allows multiple physical ports on a server to be bound as one port to connect to VMs. Administrators can set the bound port to load balancing mode or active/standby mode.

8.3.1.6 VM Communication in FusionCompute

- VMs run on a host but belong to different port groups.

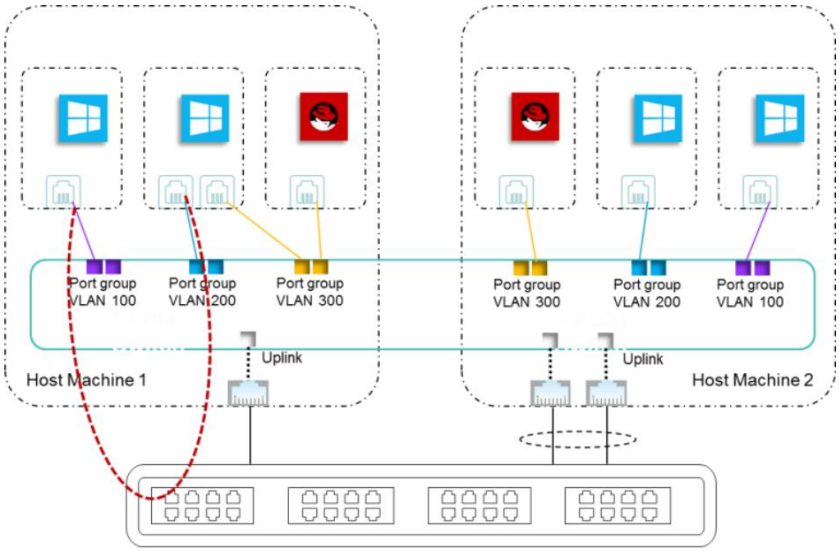


Figure 8-24 Traffic flow when VMs run on a host but belong to different port groups

A virtual switch is essentially a layer-2 switch. The port group has a vital parameter, which is VLAN ID. If two VMs belong to different port groups, they are associated with different VLANs, and cannot detect each other using broadcast. Generally, when two VMs are associated with different VLANs, they are configured with IP addresses that belong to

different network segments. To enable communication between the two VMs requires a layer-3 device, layer-3 switch, or router. In Huawei FusionCompute, the layer-3 function is provided by the physical layer-3 device. Therefore, when the two VMs intend to communicate with each other, the access traffic needs to come from a host and arrives at the physical access switch. Then, the access traffic is forwarded to the layer-3 device and routed to another host.

- VMs run on a host and belong to a port group.

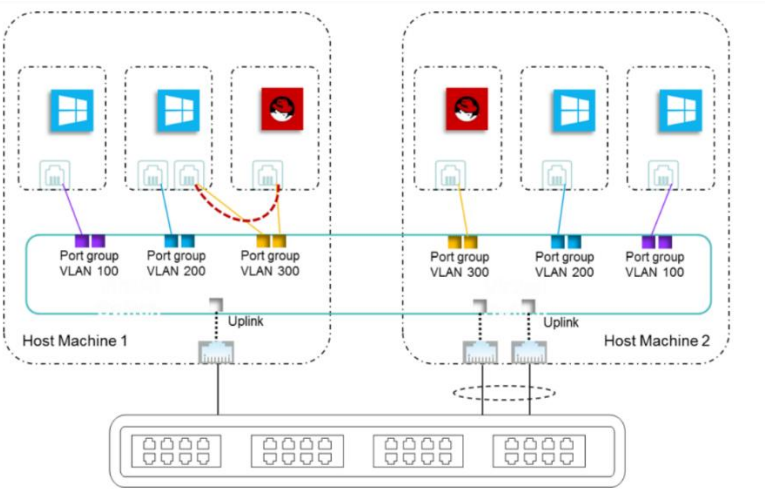


Figure 8-25 Traffic flow when VMs run on a host and belong to a port group

When two VMs run on a host and belong to a port group, they can communicate with each other using the virtual switch and the access traffic will not enter the physical network. The reason is that the two VMs belong to a port group and a broadcast domain and the virtual switch supports broadcast.

- VMs run on different hosts but belong to a port group.

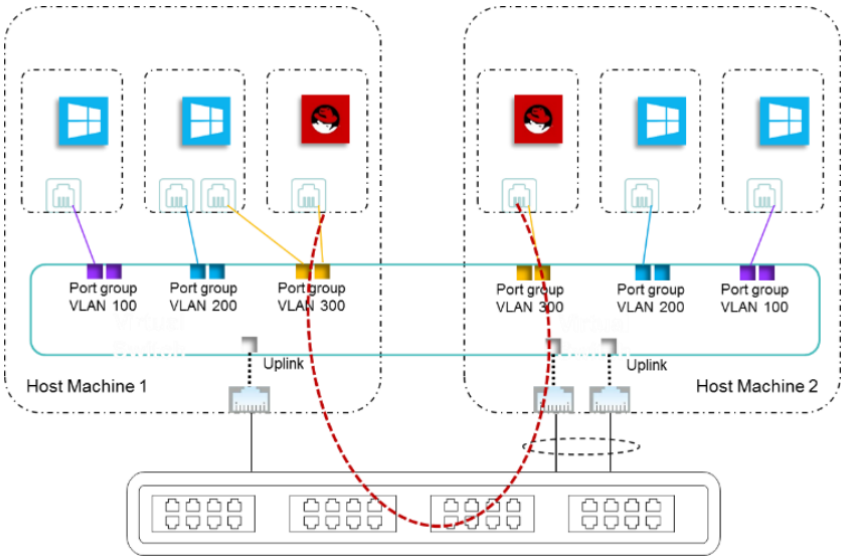


Figure 8-26 Traffic flow when VMs run on different hosts but belong to a port group

When two VMs belong to a port group, they may detect each other over broadcast. However, because the two VMs run on different hosts, they need the physical switch to connect to the network. An exception is that the two physical servers are directly interconnected. Therefore, when two VMs that run on different hosts but belong to a port group intend to communicate with each other, the access traffic needs to pass through the network port of the physical server to reach the physical switch. The two VMs can communicate with each other without using the layer-3 device, which is different from the situation when the two VMs run on different physical servers and belong to different port groups.

- Different DVSs run on a physical server.

Multiple DVSs usually run on a physical server. In this case, when two VMs are connected to different DVSs, how will the access traffic be transmitted? Generally, when two VMs are connected to different DVSs, the port groups associated with the two DVSs have different VLAN IDs, which means that the two VMs use different IP addresses. In this case, the access traffic will first reach the layer-3 device and then be routed.

8.3.2 FusionCompute Network Virtualization Features

8.3.2.1 Layer 2 Network Security Policy

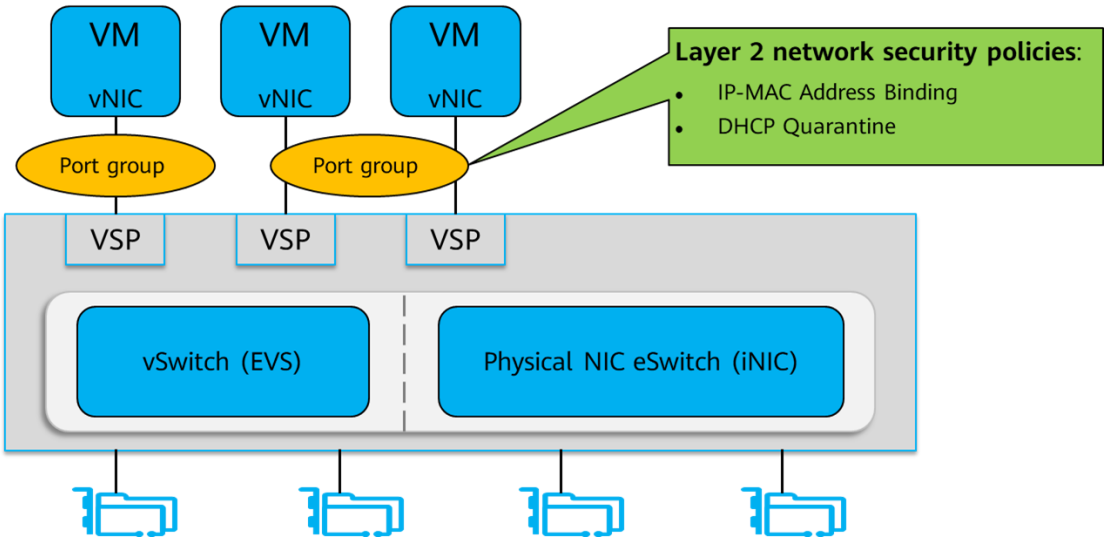


Figure 8-27 Layer 2 network security policy

In FusionCompute, the layer 2 network security policies include **IP-MAC Address Binding** and **DHCP Quarantine** to prevent IP or MAC address spoofing and DHCP server spoofing for user VMs, respectively.

IP-MAC Address Binding prevents IP address or MAC address spoofing initiated by changing the IP address or MAC address of a virtual NIC (vNIC), and therefore enhances network security of user VMs. After the binding, the packets from untrusted sources are filtered through IP Source Guard and dynamic ARP inspection (DAI).

DHCP Quarantine blocks users from unintentionally or maliciously enabling the DHCP server service for a VM, ensuring common VM IP address assignment.

8.3.2.2 Broadcast Packet Suppression

In server consolidation and desktop cloud scenarios, broadcast packet attacks caused by network attacks or virus may interrupt network communication. To prevent this, broadcast packet suppression is enabled for virtual switches.

Virtual switches support suppression of broadcast packets sent from VM ports and suppression threshold configuration. You can enable the broadcast packet suppression switch of the port group where VM NICs locate and set thresholds to reduce Layer 2 bandwidth consumption of broadcast packets.

The administrator can log in to the system portal to configure the packet suppression switch and packet suppression threshold for port groups of a virtual switch.

8.3.2.3 Security Group

Users can create security groups based on VM security requirements. A set of access rules can be configured for each security group. VMs that are added to a security group are protected by the access rules of the security group. Users can add VMs to security groups for security isolation and access control when creating VMs. A security group is a logical group that consists of instances that have the same security protection requirements and trust each other in the same region. All VM NICs in a security group communicate with VM NICs outside the security group by complying with the security group rules. Each VM NIC can be added to only one security group.

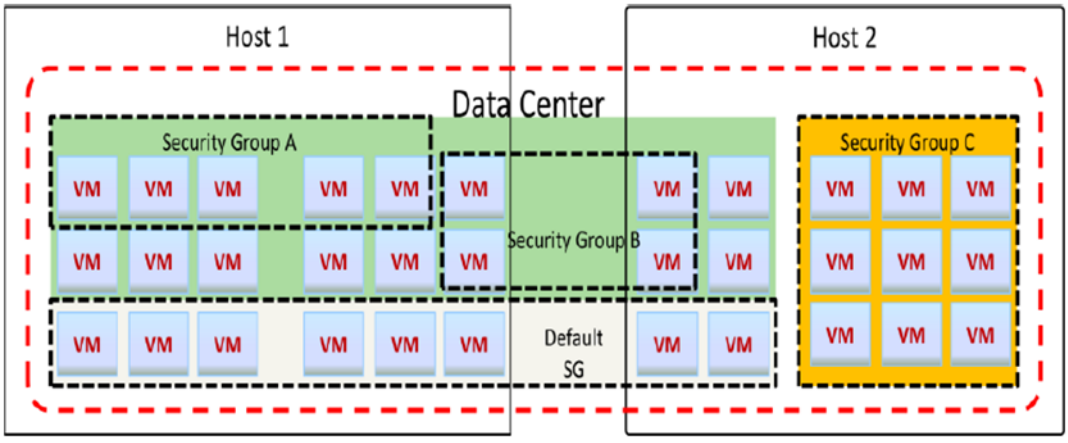


Figure 8-28 Security group

The security group provides a similar function as the firewall does. They both use iptables to filter packets for access control.

Netfilter/iptables (iptables for short) functions as the firewall filtering packets on the Linux platform. iptables is a Linux userspace module located in **/sbin/iptables**. Users can use iptables to manage the rules of the firewall. Netfilter is a Linux kernel module that implements the firewall for packet filtering.

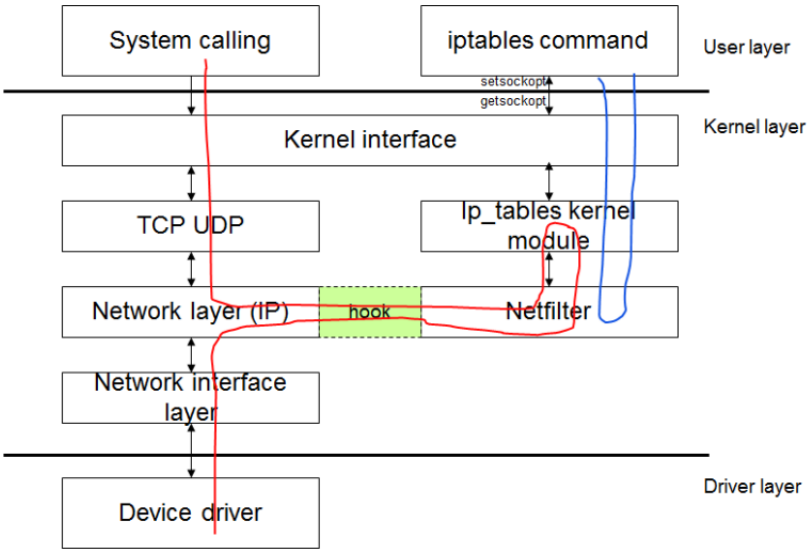


Figure 8-29 Working principles of the security group

In the preceding figure, Netfilter is a data packet filtering framework. When processing IP data packets, it hooks five key points, where services can mount their own processing functions to implement various features.

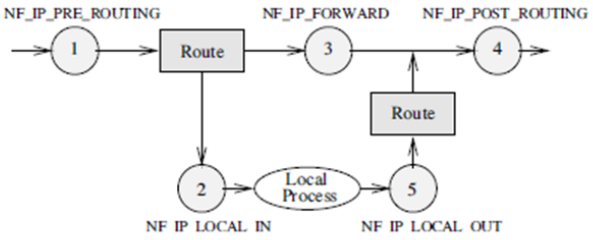


Figure 8-30 Filtering packets using rules

iptables is processed based on chains and rules in the configuration tables. Network packets that enter the chain match the rules in the chain in sequence. When a packet matches a rule, the packet is processed based on the action specified in the rule. iptables contains four tables, which are RAW, Filter (filtering packets), NAT (translating network addresses), and Mangle (changing TCP headers). These tables generate their own processing chains at the five key points as required and mount the entry functions for chain processing to the corresponding key point.

8.3.2.4 Trunk Port

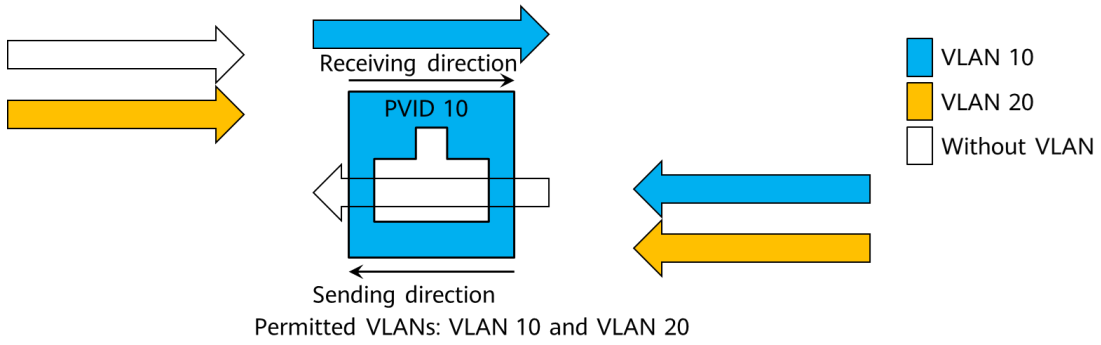


Figure 8-31 Trunk port of the virtual switch

A vNIC connects to a virtual switch through a virtual port to send and receive network data packets. The virtual port of the virtual switch can be set to the trunk mode to send and receive network data packets tagged with specified VLAN IDs. In this way, the vNIC is capable of supporting trunk ports.

An access port can be added to only one VLAN. A trunk port can receive and send packets from multiple VLANs. Select **Access** for a common VM, and select **Trunk** if a VLAN device is used for the vNIC. Otherwise, the VM network may be disconnected.

If the ports added to a port group are set to the trunk mode on a Linux VM, multiple VLAN tagging devices can be created on the VM to transmit data packets from different VLANs over one vNIC, exempting the VM from using multiple vNICs.

8.3.2.5 Network QoS

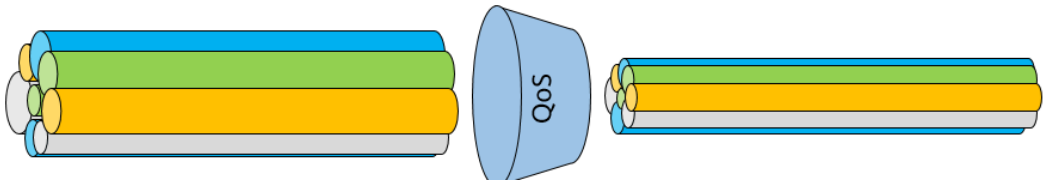


Figure 8-32 Network QoS

In FusionCompute, network QoS can be configured for port groups of VMs. The network QoS policy enables bandwidth configuration control, including:

- Bandwidth control based on the sending direction and receiving direction of a port group member port.
- Traffic shaping and bandwidth priority are configured for each member port in a port group to ensure network QoS.

8.3.2.6 Binding Network Ports

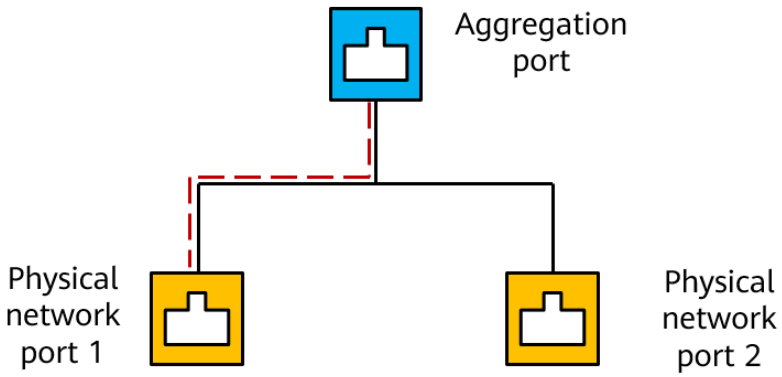


Figure 8-33 Host network port bonding

In FusionCompute, administrators can bind network ports on a CNA host to improve network reliability.

Port binding can be configured for common NICs and DPDK-driven NICs. The binding mode varies with the NIC type. The details are as follows.

The following binding modes are available for common NICs:

- Active-backup
- Round-robin
- IP address and port-based load balancing
- MAC address-based load balancing
- MAC address-based LACP
- IP address-based LACP

The following binding modes are available for DPDK-driven NICs:

- DPDK-driven active/standby
- DPDK-driven LACP based on the source and destination MAC addresses
- DPDK-driven LACP based on the source and destination IP addresses and ports

8.4 FusionCompute Virtualization Platform Management

8.4.1 Maintenance and Management

8.4.1.1 FusionCompute Account Management

FusionCompute users include local users, domain users, and interface interconnection users. A local user can log in to and manage the system. After a domain is created, a domain user can log in to the system. An interface interconnection user supports FusionCompute to interconnect with other components.

The following table lists the FusionCompute login accounts. (For details about the default passwords, see the related product documentation.)

Login Mode	Default Username/Password	Permission
Common mode	admin/XXXXXX	Has permissions of the system administrator.
Rights separation mode	System administrator: sysadmin/XXXXXX Security administrator: secadmin/XXXXXX Security auditor: secauditor/XXXXXX	System administrator: has the permissions to operate and maintain system services and create and delete users. Security administrator: has the permission to manage the rights for users and roles but does not have the permission to create users. Security auditor: has the permission to query and export operation logs of other users.

Figure 8-34 FusionCompute account table

When installing FusionCompute, you can specify the login mode. Once the mode is determined, it cannot be changed.

8.4.1.2 Alarm Management

FusionCompute alarm severities include critical, major, minor, and warning. After alarms are generated, handle alarms of high severity and then alarms of low severity. Figure 8-35 describes the alarm severities.

Alarm Severity	Icon	Description
Critical		Indicates a fault that affects the service at present, and needs to be rectified promptly.
Major		Indicates a fault that affects the service at present, and if not rectified, could result in serious consequences.
Minor		Indicates a fault that does not affect the service at present, but if not rectified, could result in more severe faults.
Warning		Indicates a potentially or imminently hazardous fault, that does not affect the service at present.

Figure 8-35 Alarm description

Maintenance engineers need to master alarm-related operations, including querying alarms, manually clearing alarms, configuring alarms, and handling alarms. For example, maintenance engineers manually clear alarms that require human intervention on FusionCompute.

Some alarms in the system need to be manually recovered. If alarms have been recovered but alarms still exist, or alarm objects have been deleted and related alarms cannot be automatically cleared, system maintenance engineers can manually clear alarms to prevent them from interfering follow-up maintenance. For details about the operation procedure, see the FusionCompute product documentation.

8.4.1.3 Monitoring Management

Administrators can query cluster, host, and VM information to obtain the cluster running status in a specified period of time.

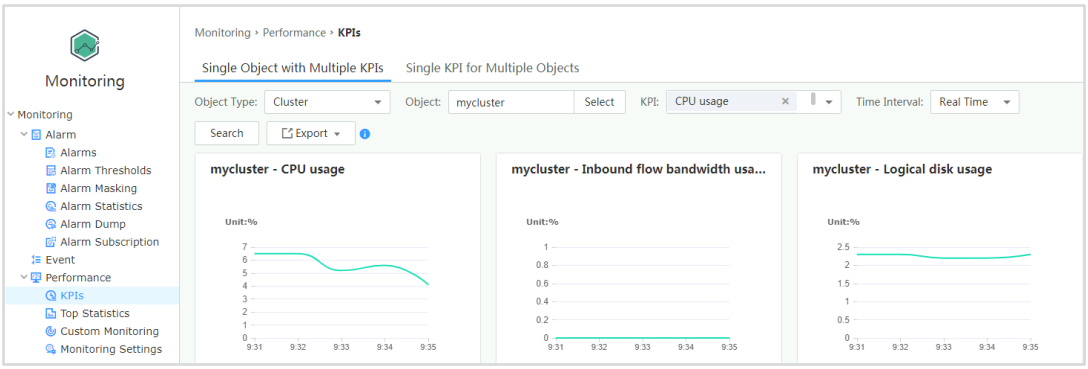


Figure 8-36 Monitoring management

As shown in Figure 8-36, FusionCompute can monitor usage of cluster, host, storage, and VM resources. You can choose **Single Object with Multiple KPIs** or **Single KPI for Multiple Objects** to view the resource usage charts.

8.4.2 Configuration Management

8.4.2.1 System Configuration

Administrators can modify FusionCompute configurations as required.

- Configuring domain authentication
- Updating the license
- Changing the system logo
- Configuring a login timeout period
- Configuring the resource scheduling interval
- Configuring an SNMP station
- Changing the VRM deployment mode from standalone to active/standby

Figure 8-37 shows the operation page.

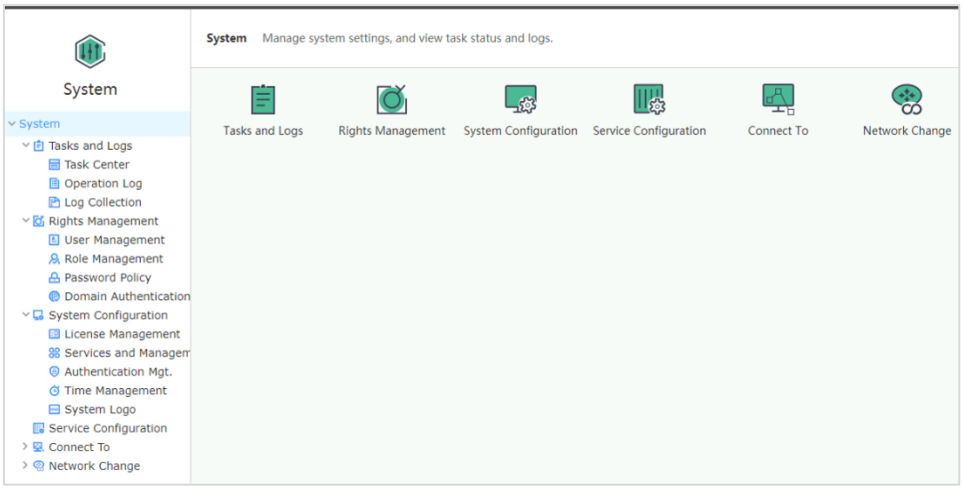


Figure 8-37 System configuration

In FusionCompute, you can query tasks and logs, modify system rights, system configuration, service configuration, and third-party interconnection, and change the network.

8.4.2.2 Task Management

On the FusionCompute web client, administrators can query the task progress on the system management page.

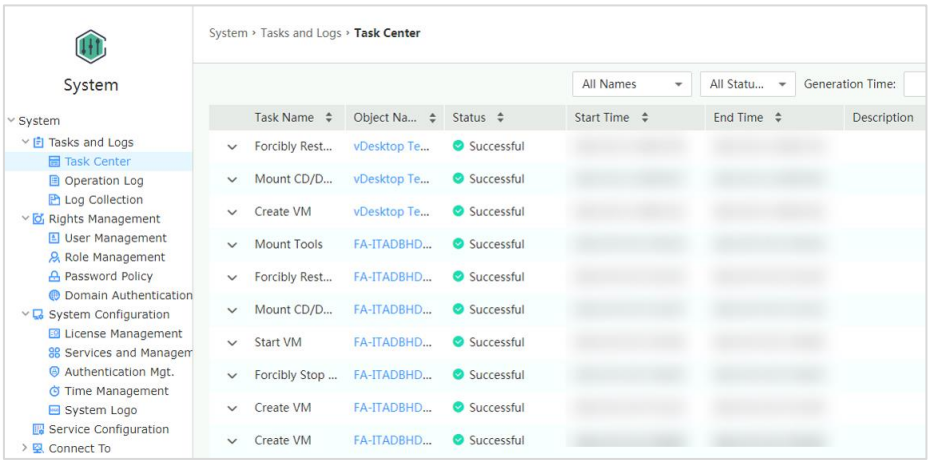


Figure 8-38 Task management

8.4.3 Cluster Resource Management

8.4.3.1 Cluster Management

Function	Description	Navigation Path
Monitoring clusters	Query cluster monitoring information (for example, running status) of a cluster within the specified time period.	**Homepage** > **Resource Pool** > **Cluster**
Configuring cluster attributes	Configure cluster attributes, such as the HA policy, memory overcommitment policy, and VM start policy.	
Configuring the resource scheduling policy	Configure the policy for scheduling compute resources in a cluster to implement dynamic resource scheduling and load balancing.	

Figure 8-39 Cluster management

FusionCompute resources include host, cluster, network, and storage resources. Host and cluster management involves the following operations on FusionCompute: creating a cluster or host as well as adjusting and scheduling host or cluster resources. Figure 8-39 shows the cluster management functions.

8.4.3.2 Cluster Configuration

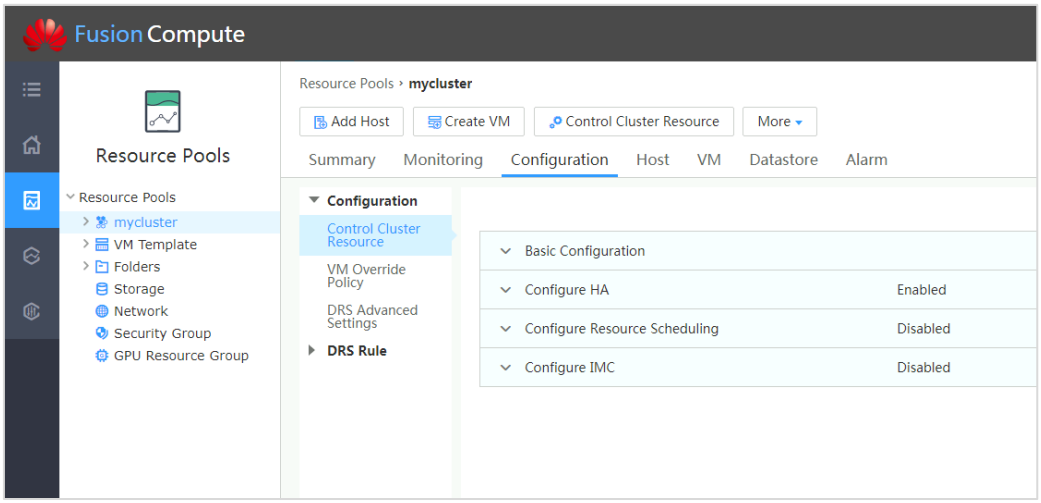


Figure 8-40 Cluster configuration

As shown in Figure 8-40, choose **Resource Pools**, select a cluster object, and configure the cluster-related functions on the **Configuration** tab page. The cluster-related functions include HA, cluster compute resource scheduling, IMC, and DRS rule. If you want to configure the DRS rule, enable the cluster compute resource scheduling.

8.4.3.3 Host Management

Function	Description	Navigation Path
Monitoring hosts	Query cluster monitoring information (for example, running status) of a cluster within the specified time period.	**Homepage** > **Resource Pool** > **Host**
Configuring host attributes	Configure host attributes, such as time synchronization policy, baseboard management controller (BMC) configuration, multi-path storage type, and maintenance mode.	
Maintaining and managing host ports	Manage and maintain host network ports, such as binding network ports and associating storage ports.	**Homepage** > **Resource Pool** > **Host** > **Configuration**
Associating storage resources with the host	Associate storage resources with the host to provide storage space for the VM on the host.	

Figure 8-41 Host management

Hosts provide compute resources to the system. Engineers must be familiar with configurations, routine monitoring operations, and maintenance operations of hosts. Figure 8-41 shows the host management.

8.4.3.4 Host Management Configuration

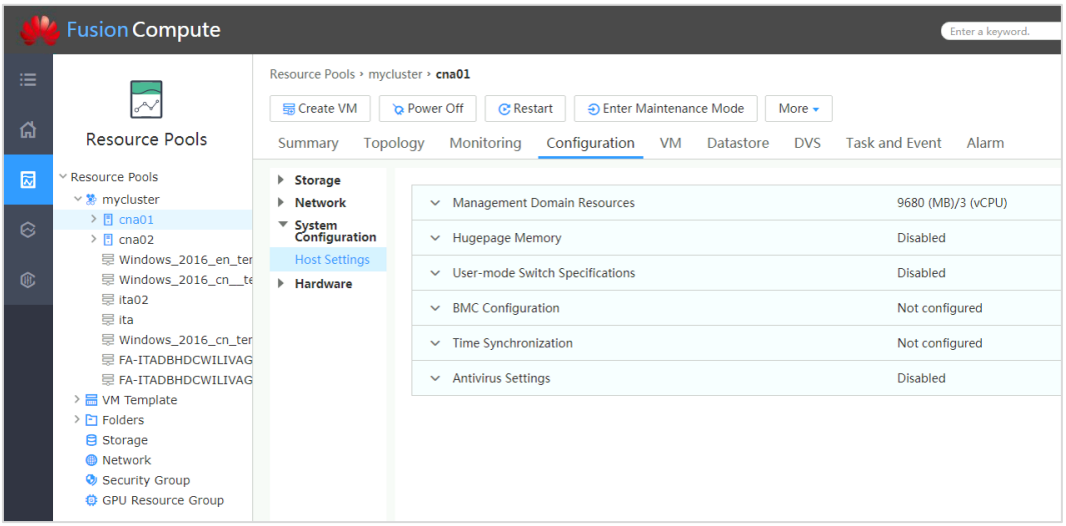


Figure 8-42 Host management configuration page

On the page shown in Figure 8-42, you can configure the BMC IP address, username, and password for a host. The host can be powered on or off by the system for the purpose of scheduling resources only after BMC parameters have been configured.

Put a host in maintenance mode. In this mode, the host is isolated from the entire system. This means that maintenance operations, such as parts replacement, power-off, or restart, can be performed on the host without affecting system services. Once a host is in maintenance mode, you must stop or migrate all VMs on the host before performing maintenance.

Configure logical ports of the host to define different network planes.

Host settings:

1. Host resources

- Configure the resources reserved for hosts in different scenarios.

2. Hugepage memory configuration

- Configure the host hugepage memory to optimize memory access efficiency and improve performance.

3. User-mode switching specifications

- When high network performance is required for a VM, configure the user-mode switching specifications for the host accommodating the VM in advance.

4. BMC configuration

5. Time synchronization

6. Antivirus settings

- Enable the antivirus function to provide user VMs running on the host with the following services: virus scanning and removal, real-time virus monitoring, network intrusion detection, network vulnerability scanning, and firewall.

8.4.3.5 Storage Resource Management

FusionCompute supports storage resources from local disks on hosts or dedicated storage devices. Dedicated storage devices are connected to hosts using network cables or fiber cables. Figure 8-43 shows the adding and management process of IP SAN storage resources.

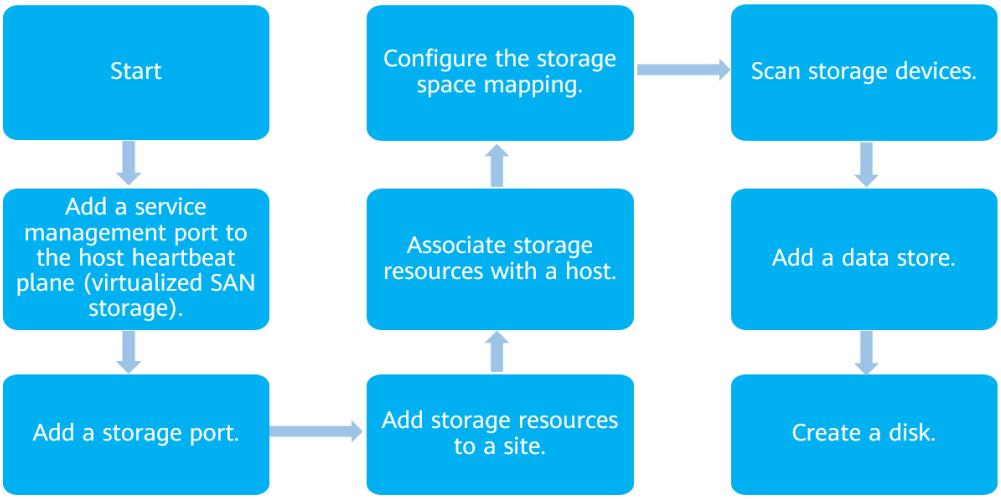


Figure 8-43 Storage resource management

The operations vary with storage resources or devices. For details, see the procedure for adding storage resources in the corresponding lab guide.

8.4.3.6 Network Resource Management

Network resource management involves the following operations for FusionCompute: create network resources, such as a distributed virtual switch (DVS) or a port group, and adjust network resource configurations.

Network Element (NE)	Description
DVS	A DVS is similar to a switch used for communication on the layer 2 network. A DVS links the port group to the VM and connects to the physical network through the uplink.
Port group	A port group is a virtual logical port similar to a template with network attributes. A port group is used to define VM NIC attributes and uses a DVS to connect to the network. **VLAN**: Users must manually assign IP addresses to VM NICs. VMs connect to the VLAN defined by the port group.
Uplink	An uplink connects the DVS to the physical network. An uplink is used for VM upstream data transmission.

Figure 8-44 Network resources

In brief, network resource management includes management of DVSs, uplink groups, and port groups. The main management operations are as follows:

- DVS management: includes the operations of creating a DVS, querying DVS information, modifying attributes of a DVS, deleting a DVS, adding a VLAN pool, deleting a VLAN pool, and modifying a Jumbo frame of the DVS.

- Uplink group management: includes the operations of adding an uplink, querying uplink information, and removing an uplink.
- Port group management: includes the operations of adding a port group, querying port group information, modifying attributes of a port group, and deleting a port group.

8.4.3.7 VM Lifecycle Management

On the FusionCompute client, you can manage VM life cycles, from creating and using to deleting VMs.

FusionCompute supports multiple VM creation methods, such as creating a bare VM, creating a VM from a template, and creating a VM using a VM.

1. Creating a bare VM

Description

A bare VM is a software computer without any OS installed. You can create a VM on a host or in a cluster, and configure VM specifications, including the specifications of CPUs, memory, disks, and NICs. After a bare VM is created, install an OS on it. The procedure for installing an OS on a bare VM is the same as that for installing an OS on a physical computer.

Suggested scenarios

- After the system is installed, the first VM is needed.
- The OSs or specifications of existing VMs or templates in the system do not meet user requirements.
- A bare VM is required to be converted or cloned to a template for VM creation. Before cloning or converting it to a template, install an OS on it.

2. Creating a VM from a template

Description

Use an existing VM template to create a VM that has similar specifications with the template. The methods are as follows:

- You can convert an existing template on the site to VM, or deploy a VM from this template.
- You can also export a template from a peer site, and import the template to the current site to create a VM.

After a template is converted to a VM, the template disappears. All attributes of the VM are identical to the template attributes.

The new VM inherits the following attributes from the template. You can customize other attributes.

- VM OS type and version
- Number and size of VM disks and bus type
- Number of VM NICs

Suggested scenarios

- The OSs and specifications of templates in the system meet user requirements.

- You can also export a template from a peer site, and import the template to the current site to create a VM.

3. Creating a VM using a VM

Description

Clone an existing VM to a VM that has similar specifications with the VM.

The new VM inherits the following attributes of the used VM. You can customize other attributes.

- VM OS type and version
- Number and size of VM disks and bus type
- Number of VM NICs

If a VM is planned to be frequently used to create VMs using cloning, you can convert or clone it to a template.

Suggested scenario

If multiple similar VMs are required, you can install different software products on one VM and clone it for multiple times to obtain required VMs.

8.5 Quiz

FusionCompute provides three disk configuration modes: common, thin provisioning, and thick provisioning lazy zeroed. What are the characteristics and differences between the common and thin provisioning modes?

9 Overview of FusionAccess

Huawei FusionAccess is a virtual desktop application based on the Huawei Cloud platform. Software deployed on the cloud platform enables end users to use a thin client or any other device that is connected to the network to access cross-platform applications and their desktops.

This chapter describes the architecture and application scenarios of FusionAccess and the principles and functions of HDP.

9.1 Overview of FusionAccess

9.1.1 Requirements for an Age of Informatization

Figure 9-1 Requirements for an age of informatization

New technologies, such as cloud computing, big data, mobility, social IT, and the Internet of Things (IoT), are transforming peoples' lives.

- The Internet is everywhere.
- Mobile Internet means you can get online anytime from anywhere.
- You can access information on the cloud, or even work on cloud.
- Enterprises use big data to analyze services and adjust their strategies.
- Chatting online has become commonplace.

In short, traditional PCs are becoming insufficient for offices that are going informatization.

9.1.2 Pain Points of a PC-Based Office

Nowadays, PCs have become essential to enterprise operations, but using PCs for everything has created some problems too.

First, there is information security. When all of your data is saved to local PCs, even though a lot of work has gone into keeping this data safe. Data loss and information leakage seem inevitable.

Second, PC management and maintenance have also become a burden. As the number of PCs keeps increasing and new applications keep being developed, it makes O&M more complicated and time consuming. And outsourcing IT O&M just drives up OPEX. Time wasted handling PC faults and provisioning new devices slow down service development.

Third, physical resources are fixed, which means they often get wasted. PCs tend to be in use only for a third of the day. Once staff leaves work and goes home, those physical PCs are left idle or shut down for the remainder. Their capacity is wasted. Furthermore, when more powerful processing capabilities are required due to service changes, physical PCs fail to keep up due to hardware constraints. PC-based office desktops have become a bottleneck for service development.

9.1.3 Advantages of the FusionAccess

FusionAccess can better meet the requirements of the age of informatization. FusionAccess brings higher information security, requires less investment, and improves office efficiency over conventional PCs. It fits into the office automation (OA) of financial institutions, enterprises, public institutions, and medical institutions, and allows you to work smoothly even when you are outdoors or on business trip. FusionAccess ensures the optimal OA user experience anytime, anywhere.

FusionAccess boasts the following advantages:

(1) Data Stored in the Cloud for Enhanced Information Security

User data of conventional desktops, which is stored in local PCs, is prone to data breach or loss caused by cyber attacks. However, user data of virtual desktops is stored and processed on servers rather than on user terminals, preventing data breach or loss. In addition, security mechanisms such as TC access authentication and encrypted data transmission are employed to ensure the security and reliability of virtual desktops.

(2) Efficient Maintenance and Automatic Control

Conventional PC desktops are failure-prone. An IT professional is required on average for managing and maintaining 400 PCs, and the maintenance procedure for each PC requires 2 to 4 hours.

With FusionAccess, resources are under automatic control, simplifying maintenance and reducing IT investments.

- Efficient maintenance: FusionAccess does not require an enterprise to designate personnel for maintaining access devices. A powerful one-click maintenance tool is provided to simplify the maintenance and increase productivity. One IT engineer can manage more than 2,000 virtual desktops, improving the maintenance efficiency by over four times.

- Automatic control: In the daytime, resource usage is automatically monitored to ensure load balancing among physical servers. At night, physical machines not in use are shut down to save energy.

(3) Services Running in the Cloud for Higher Reliability

In conventional PC desktops, all services and applications run on local PCs, with 99.5% availability and the average downtime is about 21 hours a year. However, with FusionAccess, all applications and services run in cloud data centers, Figure 9-6.9% availability. The reliable and smooth running of applications greatly reduces management and maintenance costs of office environments.

(4) Mobile Office Enabled by Seamless Application Switchover

Traditionally, users can only access their desktops from a dedicated device. However, with FusionAccess, users, either in offices or on a trip, can access their desktops at any time, from anywhere. In addition, users can move from one place to another during work without interrupting applications, because data and desktops are running and stored in cloud data centers.

(5) Reduced Noise and Power Consumption

Energy-efficient TCs lower the temperature and minimize noise in offices. After TCs are deployed, noise in offices is reduced from 50 dB to 10 dB. The total power consumption of a TC and liquid crystal display (LCD) is about 60 W, generating 70% lower electricity costs than a conventional PC. Reduced power consumption also means lower temperature control costs.

(6) Resource Elasticity and Sharing

- Resource elasticity: With FusionAccess, all resources are stored in data centers to implement the centralized management and elastic scheduling.
- Improved resource utilization: Resources are stored in a centralized manner. The average CPU utilization of conventional PCs is 5% to 20%. With FusionAccess, the CPU utilization of cloud data centers is about 60%, dramatically improving resource utilization.

The advantages of FusionAccess are built on its cutting-edge architecture.

9.1.4 VDI and IDV

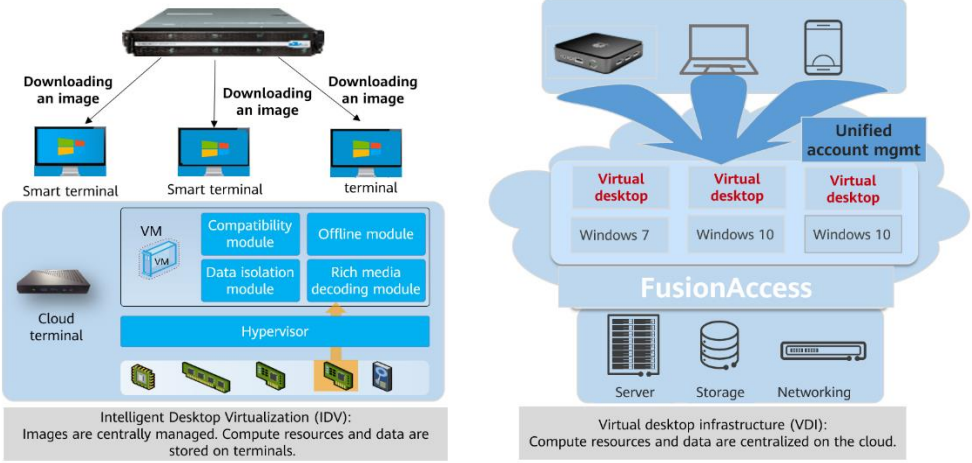


Figure 9-2 VDI and IDV

As shown in Figure 9-2, there are two virtual desktop solutions: VDI and IDV. We can compare VDI with IDV in terms of technology and industry trend.

Technology Comparison

- An IDV represents a small step forward from a traditional PC-based office architecture. Typically, it consists of an image server, management software, and local virtual desktop terminals (fat clients). A VDI, in contrast, is a completely new design. Instead of PCs, it uses TCs and cloud virtual desktops, which can be scaled out using any kind of terminals you want, as no data is stored locally. With a VDI, the security, O&M, and low resource utilization pain points of a traditional PC-based office are all addressed.
- The IDV solution uses local virtual desktops, which means there is a more complex mix of terminals in use and management and maintenance are more difficult. When a large number of terminals need to be managed, a complex server is required to centrally manage terminal images and synchronize data distributed on terminals.
- In certain scenarios, IDV can work offline, but the number of applications that can run offline has been decreasing. As investment into network infrastructure continues, we will eventually see offline applications become a thing of the past.

Industry Trends

- VDI is favored by mainstream vendors such as Huawei, Citrix, and VMware (data of IDC: top 3 vendors in China in terms of the virtual desktop market share), while IDV is used only by some Chinese vendors such as Ruijie and OS-Easy.
- VDI's integration with cloud makes it future proof.

In addition, this section compares the performance of VDI and IDV. See the following table.

Item	VDI	IDV	Remarks
Data Security	High. Centralized data storage prevents data leakage through terminal access.	Low. Data is downloaded to the terminals, which makes data leakage more likely.	Centralized data storage is more secure than local storage.
Terminal Maintenance	Easy. Terminals only provide access. No maintenance is required. Faulty terminals can be replaced at any time.	Difficult. Terminals provide services. Complex maintenance is unavoidable.	In a centralized architecture, terminal faults can only be rectified manually.
System Reliability	High. Cloud-based resources support time-based scheduling and dynamic allocation. Hardware faults can be fixed automatically.	Low. Manual intervention is required when a terminal fault occurs.	If IDV data is not synchronized to the server in a timely manner, data may be lost or inconsistent.
Terminal Requirements	None	The CPU must support virtualization and dual-OS installation. TCs are not supported.	CPUs that support virtualization are expensive and waste power.
Mobile Terminals	Support	No	With IDV, only certain terminals are supported.
Mobile Office	Support	No	In an IDV solution, if a terminal is replaced, the image needs to be pulled again.

Figure 9-3 VDI and IDV

9.1.5 FusionAccess Architecture

FusionAccess is a virtual desktop application deployed on hardware and enables end users to access cross-platform applications and virtual desktops from TCs or other networked devices.

FusionAccess brings higher information security, requires less investment, and improves office efficiency over conventional PCs. It fits into the office automation (OA) of financial institutions, enterprises, public institutions, and medical institutions, and allows you to work smoothly even when you are outdoors or on business trip. FusionAccess ensures the optimal OA user experience anytime, anywhere.

FusionAccess boasts the following advantages:

- Administrators can control and manage virtual desktops in a centralized manner because the desktops are running in the data center.
- Users can easily access personalized virtual desktops and obtain the PC-like user experience.
- Total cost of ownership (TCO) is reduced because virtualized desktops require less management and resources.
- FusionAccess supports GPU passthrough and GPU hardware virtualization, allowing users to remotely access graphics desktops and helping reduce the TCO of graphics desktops.

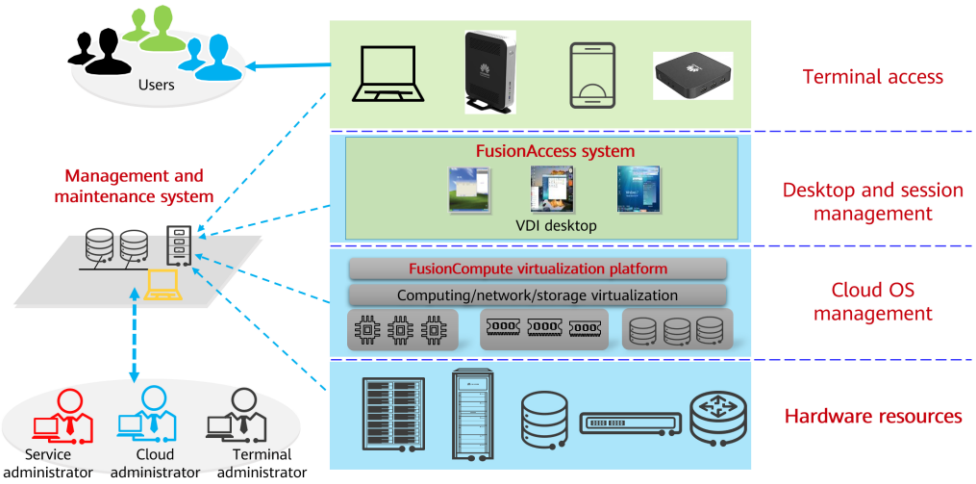


Figure 9-4 FusionAccess architecture (1)

As shown in Figure 9-4, the FusionAccess architecture consists of four layers: terminal access, desktop and session management, cloud OS management, and hardware resources. FusionAccess belongs to the desktop and session management layer and depends on the virtualization resource platform, such as the FusionCompute virtualization platform. Therefore, a virtualization platform such as FusionCompute needs to be set up using hardware resources for FusionAccess. Administrators can create and provision VDI desktops on FusionAccess. Users can use terminals and the terminal access layer to access and use VDI desktops for routine office work.

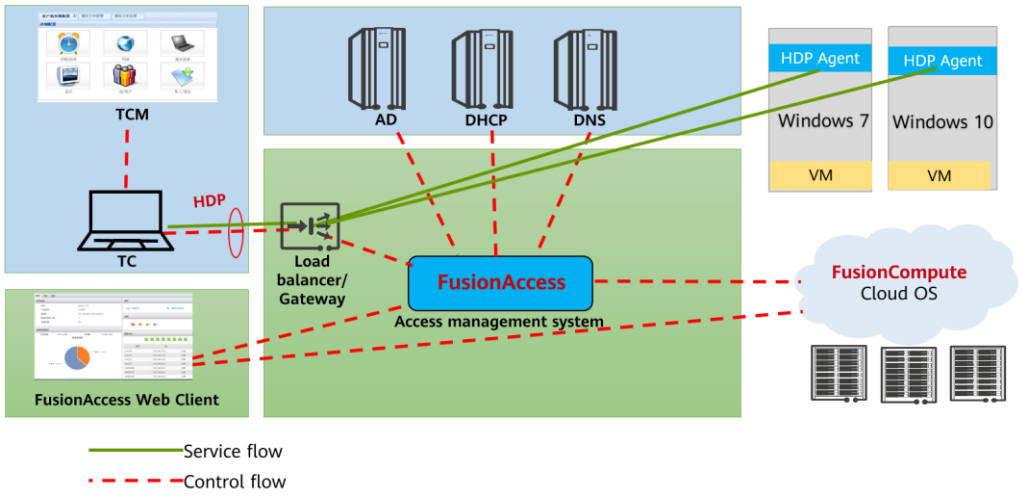


Figure 9-5 FusionAccess architecture (2)

We can also look at the FusionAccess architecture from the user perspective.

- A user uses a client to connect to the load balancer and gateway of the FusionAccess system. The user accesses the login page after FusionAccess forwards the access request.
- The user enters the account and password on the login page.
- The user account and password are forwarded to the AD component for identity authentication and the authentication result is returned.
- If the verification is successful, the user can view the VDI desktop on the login page.
- Click the VDI desktop. After the internal pre-connection and connection of FusionAccess, the user can log in to and use their VDI desktop. After the connection is successful, the client communicates with the HDA of the VDI desktop using the HDP.

The HDP Agent, part of the FusionAccess system, transmits desktop display information of VMs to clients using the HDP and receives peripheral information about the keyboard and mouse of clients.

Huawei Desktop Protocol (HDP) is a next-generation cloud access desktop protocol.

9.2 Introduction to FusionAccess Components

9.2.1 FusionAccess Overview

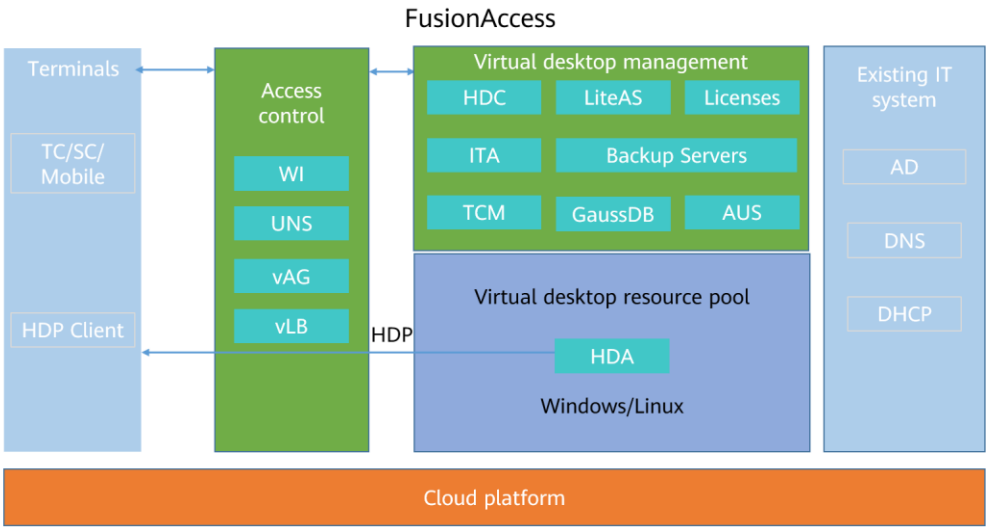


Figure 9-6 FusionAccess component overview

As shown in Figure 9-6, the green parts indicate the components of FusionAccess, which consist of the access control and virtual desktop management layers. The details about FusionAccess components will be described in the following sections. The terminal access layer consists of thin clients (TCs), software clients (SCs), and mobile terminals.

9.2.2 Access Control Layer

Components of the access control layer include WI, UNS, vAG, and vLB. The functions of these components are described as follows:

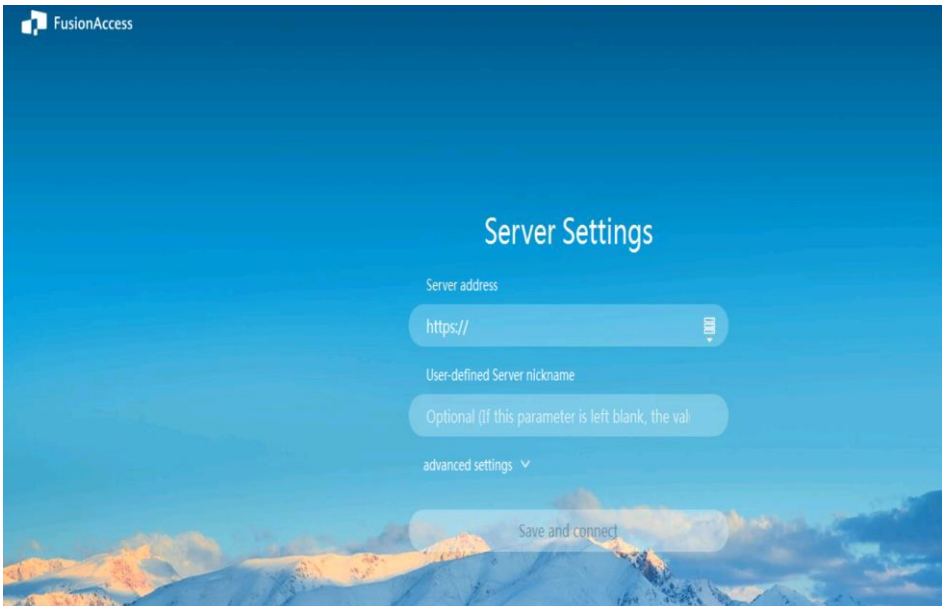


Figure 9-7 WI

Web Interface (WI)

The WI provides a web login page for users. After a user initiates a login request, the WI forwards the user login information (the encrypted username and password) to the LiteAS for authentication. If the authentication succeeds, the WI displays a computer list provided by the HDC to the user. The user can log in to any computer in the list.

Unified Name Service (UNS)

The UNS allows users to access multiple FusionAccess systems with different WI domain names using a unified domain name or IP address. Users do not need to switch between WI domain names.

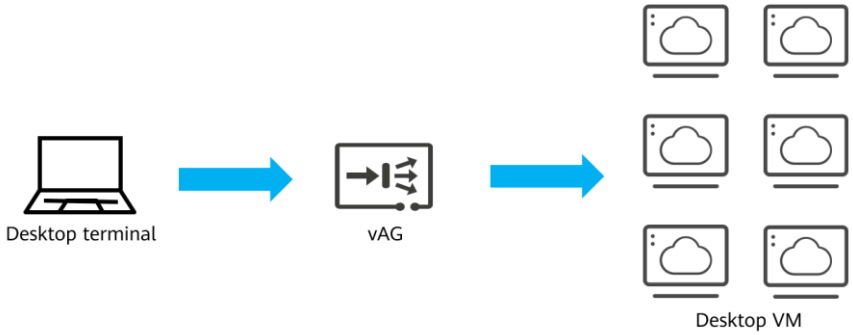


Figure 9-8 vAG

Virtual Access Gateway (vAG)

The vAG is used as desktop access gateway and self-service console gateway. When a user computer is faulty, the user cannot log in to it using a desktop protocol. In this case, the user can use the VNC self-service console to log in to the faulty computer and perform self-service maintenance.

Figure 9-9 vLB

Virtual Load Balancer (vLB)

Terminals access user computers using the vLB and vAG at the access layer. The vAG serves as an access gateway for services based on the Huawei Desktop Protocol (HDP) and self-maintenance. The vLB is used to balance the load of multiple WIs.

The LB implements load balancing between WIs to prevent a large number of users from accessing the same WI. This can be implemented by deploying the vLB.

The vLB implements load balancing between WIs as follows: The IP addresses of multiple WIs are bound to one domain name. When users enter the domain name to send login requests, the vLB resolves the domain name to WI IP addresses according to the IP address binding sequence, and evenly allocates the users' login requests to the WIs whose IP addresses have been resolved. In this way, the vLB ensures the reliability of the WIs and accelerates WI response.

9.2.3 Virtual Desktop Management Layer

The virtual desktop management layer consists of the ITA, HDC, License, GaussDB, TCM, Backup Server, LiteAS and AUS.

The functions of these components are described as follows:

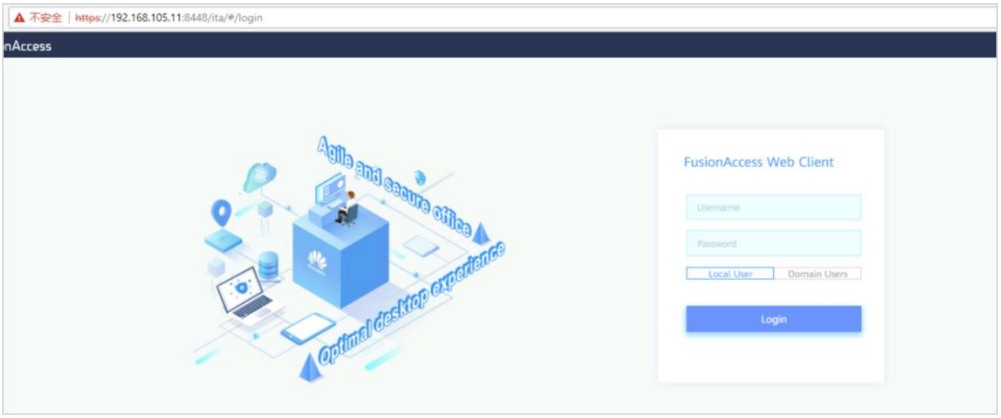


Figure 9-10 ITA

IT Adapter (ITA)

The ITA provides interfaces for users to manage VMs. It interacts with the HDC and the FusionCompute cloud platform software to create and assign VMs, manage VM statuses and templates, and operate and maintain VMs. The ITA is a Tomcat-based Web service. Specifically, the ITA provides unified interfaces for the IT portal and interfaces for the HDC, FusionCompute, VMs, and DNSs.

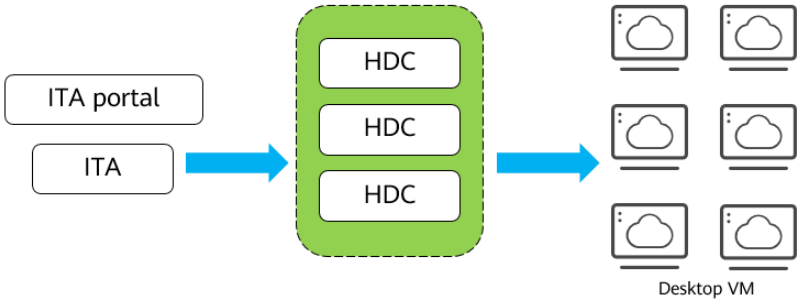


Figure 9-11 HDC

Huawei Desktop Controller (HDC)

The HDC is a core component for virtual desktop management. It manages desktop groups and assigns or unassigns VMs to or from users at the request of the ITA, and also processes VM login requests.

One HDC can manage up to 5,000 desktops. One ITA can manage multiple HDCs and a maximum of 20,000 desktops. The HDC:

- Implements and maintains the mapping between users and virtual desktops.
- Exchanges access information with the WI and helps users access desktops.
- Exchanges information with the HDA in the VM and collects information about the VM operating status and access status reported by the HDA.

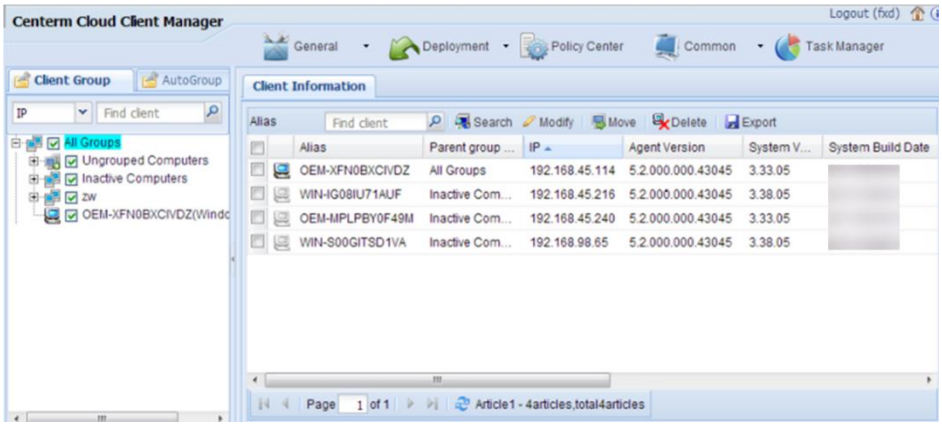


Figure 9-12 TCM

Thin Client Management (TCM)

The TCM is a desktop management system that allows administrators to perform routine management on TCs. A management server provides centralized management for TCs, including version upgrade, status management, information monitoring, and log management.

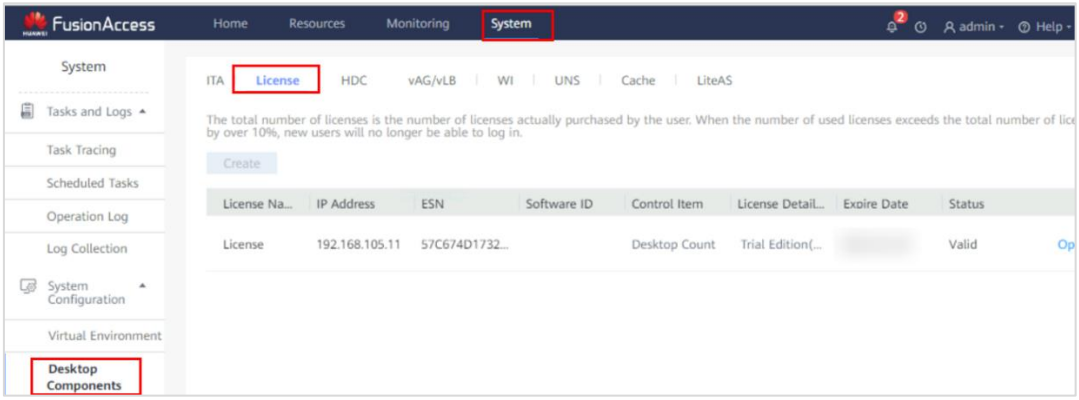


Figure 9-13 License server

License Server

The License server manages and distributes licenses for the HDC.

FusionAccess uses the license for HDP connections. When a user attempts to connect to a VM, FusionAccess checks the license on the license server to determine whether the user can connect to the VM.

The access licenses of FusionAccess are controlled by the license server.

The total number of licenses is the number of purchased licenses. When the number of used licenses reaches 1.1 times the total number of licenses, new users cannot log in to the desktops.

GaussDB

GaussDB stores data for the ITA and HDC.

BackupServer

1. The BackupServer is used to back up important files and data of components.
2. BackupServer backup policy:

- The backup operation is performed at 01:00 a.m. every day and the backups are uploaded to **/var/ftpsite/*<ITA name>*/*<folder of a component>*** on the BackupServer.
- The Backup Server retains backup data for 10 days. If backup space is insufficient, the system automatically deletes backups on a first-in first-out basis.

LiteAS

The LiteAS is a unified authentication server. It manages and authenticates desktop users and desktop user groups who access FusionAccess.

AUS

The AUS is used to upgrade the HDA. Install and configure the AUS on FusionAccess to upgrade the HDA properly.

9.2.4 Core Component of the Desktop VM - HDA

Huawei Desktop Agent (HDA): installed on the virtual desktop to connect terminals.

Thin client/Software client (TC/SC): According to the HDP protocol, a TC or SC can be connected to a VM only when the VM is configured with an HDA.

HDA provides services for TCs or SCs to use VMs.

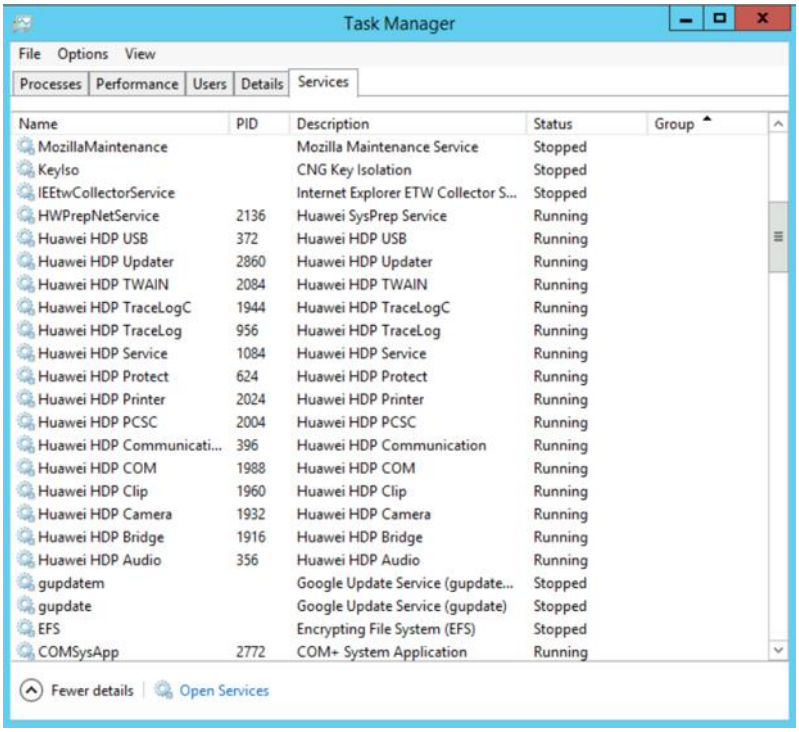


Figure 9-14 HDA

9.3 Introduction to HDP

9.3.1 Huawei Desktop Protocol

Huawei Desktop Protocol (HDP) is a next-generation cloud access desktop protocol with the following features:

- Up to 64 virtual channels. Each virtual channel bears different upper-layer application protocols. The virtual channels ensure communication security and good user experience based on Quality of Service (QoS) priorities (for example, the keyboard and mouse virtual channel can be given the top priority).
- Different compression algorithms can be used for different application types. HDP flexibly switches between server or local rendering as needed. Hardware interfaces of chips are used to accelerate video decoding and smoothen video playback. It supports 4K video playback.
- Smooth clear video playback
- Lossless compression algorithms HDP adopts lossless compression algorithms for non-natural images, and does not require transmission of repeated images. When HDP is used to display non-natural images, such as characters, icons, and OA desktops, the peak signal to noise ratio (PSNR) of HDP exceeds 50,000 dB, and the structural similarity (SSIM) reaches 0.999955, providing close-to-lossless video display quality.
- High fidelity audio HDP automatically detects voice scenarios, implements denoising when detecting noises, supports transparent voice transmission on TCs, provides

more clear sound in real time, and accurately restores sound. The perceptual evaluation of speech quality (PESQ) is over 3.4.

- Robust protocol management policies. Multiple protocol management policies are available. It provides independent channel policies for different users and user groups to ensure communication security.

9.3.2 HDP Architecture

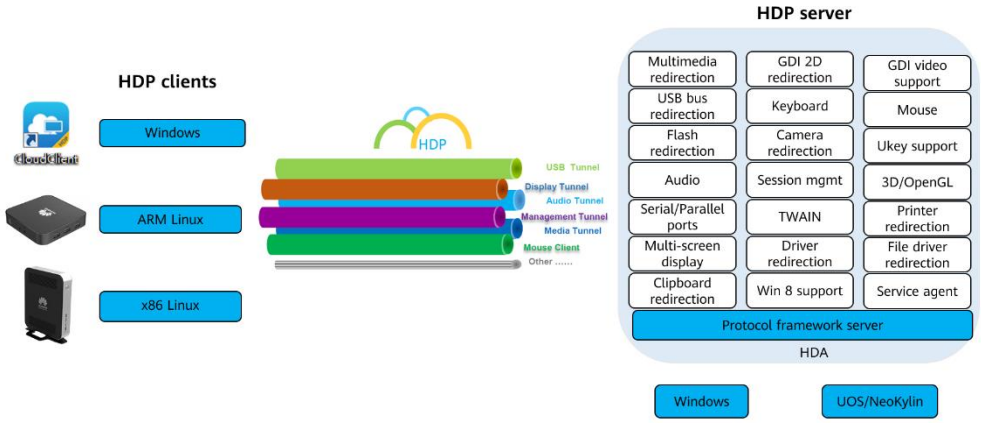


Figure 9-15 HDP architecture

As shown in Figure 9-15, HDP complies with the C/S architecture. The HDP server is deployed in the VDI desktop system as the HAD. The system can be Windows or Linux. Note that Windows desktops can only run on the x86 architecture. ARM-based Linux desktops are supported. The HDP client (TC or SC) connects to the HDP server using the HDP channel.

9.3.3 Common Desktop Protocols

In addition to Huawei's desktop communication protocols, there are other mainstream desktop protocols in the industry.

9.3.3.1 ICA/HDX

Citrix Independent Computing Architecture (Citrix ICA) is one of the most popular virtual desktop protocols. In addition to complete functions, ICA provides the following functions:

- Support for a wide range of mobile terminals.
- Network protocol independence. ICA supports TCP/IP, network basic input/output system (NetBIOS), and Internet Packet Exchange/Sequenced Packet Exchange (IPX/SPX).
- ICA supports XenServer, vSphere and Hyper-V virtualization platforms.
- ICA requires little bandwidth, so it can be used in networks of poor quality (for example where there is high latency).

High Definition Experience (HDX) is an enhanced edition of ICA. HDX improves user experience with video, audio, multimedia, and 3D services. HDX supports H.264.

ICA is the core of Citrix, connecting the platform application client operating environment and remote terminals. The I/O data (such as data about mouse, keyboard, image, sound, port, printing) of the former is redirected to the I/O devices of the latter through 32 ICA virtual channels. This provides the same user experience as using local applications.

9.3.3.2 PC over IP (PCoIP)

PCoIP was developed by Teradici for high-end graphics design. In 2008, VMware joined Teradici in developing PCoIP to develop its own virtual desktop infrastructure (VDI) solution VMware View.

- PCoIP works closely with hardware. With PCoIP, data encoding and decoding and graphics processing are implemented by dedicated hardware resources, so CPU resources are freed up for other uses. Monitors equipped with PCoIP display chips are provided.
- PCoIP is based on UDP. UDP cannot ensure reliable transmission, but it does not require the three-way handshake that TCP does for complex verification and data restoration, so it is faster and more appropriate for multimedia transmission.
- PCoIP does not support redirection of peripherals, such as serial and parallel ports. Some TC vendors provide port redirection plug-ins to make up for this.

PCoIP compresses and transmits user sessions as images and transmits only the changed parts, ensuring efficiency even in a low-bandwidth environment. PCoIP supports a resolution of 2560 x 1600 on multiple screens and a maximum of four 32-bit screens, and the Clear Type font.

Unlike TCP-based protocols such as RDP or ICA/HDX, PCoIP is based on UDP. Why UDP? TCP requires a three-way handshake for verification, which makes it not applicable to the WAN environment. Online streaming media platforms such as Xunlei Kankan and PPLIVE use UDP to maximize the use of network bandwidth and ensure smooth video playback. UDP is simple and efficient, and is usually used to provide real-time services such as VoIP and video conferencing.

PCoIP compresses and transmits user sessions as images and transmits only the changed parts, ensuring efficiency even in a low-bandwidth environment. In the WAN environment, PCoIP is adaptive and can fully utilize network bandwidth resources.

PCoIP is a typical host-end rendering protocol with good compatibility. The line speed affects the quality of the image. With a low-speed line, PCoIP transmits a lossless image to the client. As the line speed increases, PCoIP gradually displays high-definition images. PCoIP not only supports VMware solutions, but also supports hardware encoding and decoding on blade PCs and rack workstations equipped with Teradici host cards.

9.3.3.3 SPICE

Simple Protocol for Independent Computing Environments (SPICE)

- SPICE is a virtual desktop protocol developed by Qumranet. Later, it was purchased by Red Hat who provides it as an open protocol. After years of community development, SPICE is maturing.
- SPICE is good for video services, largely because video is compressed using a Kernel-based Virtual Machine (KVM), which puts less pressure on the guest OS. SPICE uses lossless compression to provide an HD experience, but that also means it needs a lot of bandwidth.

SPICE is a high-performance, dynamic, and adaptive telepresence technology, providing the same user experience as using local PCs. SPICE is designed and created for Red Hat Enterprise edition users to remotely access virtual desktops.

It uses a multi-layer architecture to meet the diverse multimedia requirements of desktop users. SPICE aims to realize intelligent access of the available system resources (such as CPUs and RAM) on client devices and virtual hosts. As a result of the access, the protocol dynamically determines whether to present the desktop application on the client device or the host server to provide the optimal user experience regardless of network conditions.

SPICE provides the optimal customer experience and has huge market potential. It is favored by Chinese virtualization vendors, such as Shenzhen Jing Cloud, CloudTop Network Technology, and NaCloud Era. Virtual desktops based on SPICE have earned the trust of customers.

9.3.3.4 RDP/RemoteFX

- Remote Desktop Protocol (RDP) is a Microsoft protocol which was developed by Citrix. RDP provides few functions and is mainly used for Windows. Mac RDP clients and Linux RDP clients RDesktop are now available as well. The latest RDP version supports printer redirection, audio redirection, and clipboard sharing.
- RemoteFX is an enhanced edition of RDP. RemoteFX supports virtual graphics processing units (vGPUs), multi-point touch, and USB redirection.

In RDP, any authorized terminal can be a terminal server. Client can log in to the server and use the corresponding resources (including software and hardware). After the protocol is upgraded, client can even use the local resources (including the printer, audio playback, disks, and hardware interfaces). All calculations are performed on the server. The client only needs to process network connections, receive data, display interfaces, and output device data. In China, virtualization vendors that use the RDP include Beijing Fangwu and Xi'an Realor.

9.3.3.5 Comparison of Common Desktop Protocols

This section compares the features of the preceding protocols. See the following table.

Feature	PCoIP	ICA	RDP	SPICE	HDP
Transmission bandwidth	High	Low	High	Medium	Low
Image display	High	Medium	Low	High	High
Two-way audio support	Low	High	Medium	High	High
Video support	Low	Medium	Medium	High	High
Peripheral support	Low	High	High	Medium	High
Transmission security	High	High	Medium	High	High

Figure 9-16 Comparison of common desktop protocols

9.3.4 HDP - 2D Graphics Display Technology

For remote display, screens of servers are captured using OS interfaces, and the screen captures are displayed on clients after processing.

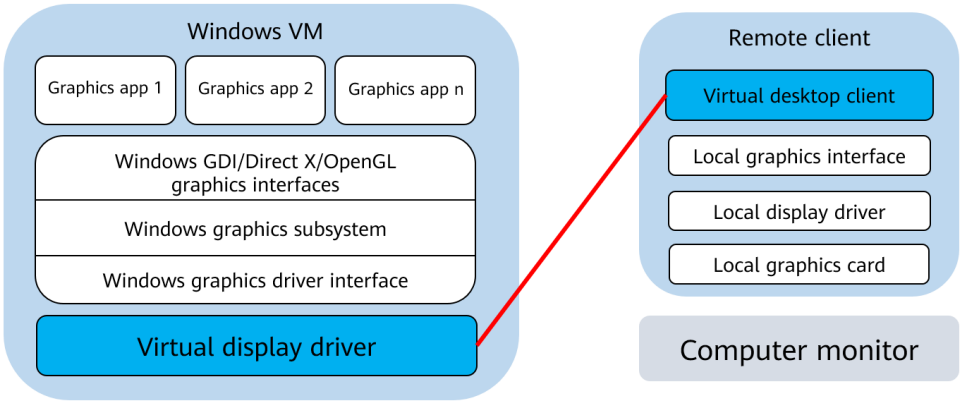


Figure 9-17 2D graphics display technology

As shown in Figure 9-17, in OS software layers, the display driver interacts with the graphics card. The upper-layer systems need to use the display driver to interact with the graphics card. Screen captures are transferred from the display driver to the graphics card of the remote TC to implement remote display.

Key Display Technologies of HDP

- Lossless compression for non-natural images: Non-natural images, such as text, Windows frames, and lines within images, are identified automatically and lossless compression applied. Natural images, like photos, are compressed at an appropriate rate.
- No transmission of redundant image data: To save bandwidth, HDP identifies what image data has changed and only transmits the changes.
- Support for multiple image compression algorithms: The most appropriate compression algorithm is selected based on different image characteristics and use cases.

9.3.5 HDP - Audio Technology

The HDP server simulates an audio driver on a VM. The audio driver interacts with the Windows audio subsystem (audio engine).

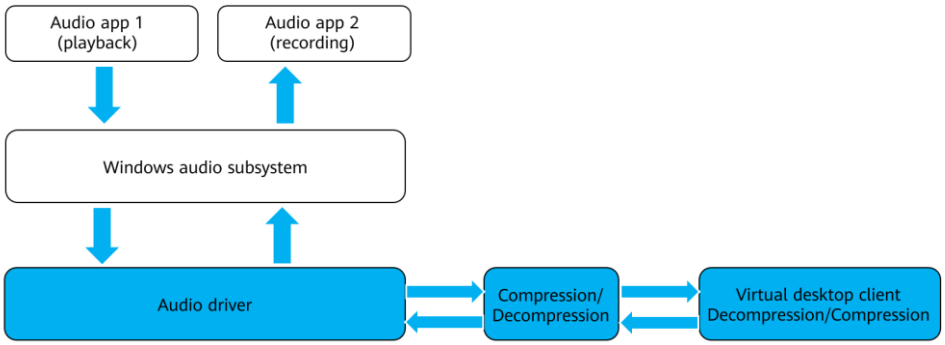


Figure 9-18 Audio technology

As shown in Figure 9-18, during audio playback, the audio driver transmits audio data received from the Windows audio subsystem to the desktop protocol client after compression, and the client plays the audio after decoding. During audio recording, the client transmits local recording data to the server after compression, the server decodes the data, and the audio driver returns the data to the Windows audio subsystem. Audio is sensitive to latency, so latency must be controlled in the whole process.

Key Audio Technologies of HDP

- High-fidelity music compression algorithm: Sound scenarios are automatically identified. A voice compression optimized for VoIP used for voices and professional high-fidelity music codecs are used for music.
- Automatic denoising: A denoising algorithm is used for VoIP to ensure excellent voice quality even in noisy environments.
- Low latency: Voice content is transmitted transparently on TCs to avoid buffering, reduce latency, and ensure real-time performance for voice communications.
- High sound quality: A default 44.1 kHz sampling rate ensures quality audio.
- Stereo mixing: All VM audio inputs and outputs can be mixed.

9.3.6 HDP - Display Technology

Currently, Huawei Desktop Protocol supports two types of video playback:

- Video recoding: Multimedia playing on the server is re-coded before being transmitted to the client for decoding and display.
- Video redirection: Video streams on the server are captured and transmitted directly to the client for decoding and display.

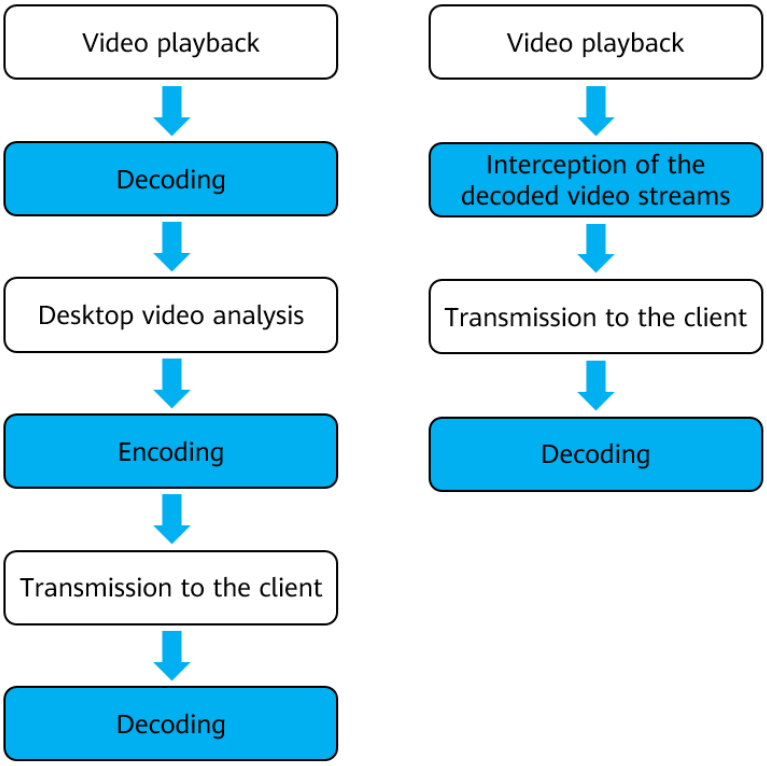


Figure 9-19 Video display technology

As shown in Figure 9-19, video redirection is more efficient because no resources are required for video decoding and re-coding on the server. However, this method has some disadvantages:

In method 1, the player running in the desktop VM consumes a large number of CPU resources for decoding videos. More CPU resources are consumed when the video area is encoded. As a result, VM density of a server is reduced. Moreover, dynamically detecting the video area is technically challenging. Usually, the image change area where the refresh rate exceeds a certain frame rate is detected as the video area.

In method 2, video code streams to be decoded are intercepted on the server only and then transmitted to the client for decoding and display, which consumes less CPU resources of the server. The multimedia redirection technology for Media Player is a popular client decoding technology. However, this technology is not popular in China because the Media Player is rarely used in China. The multimedia redirection technology for other players is emerging.

HDP supports 4K video playback. Source video files are transmitted from the server to the client, where they are decapsulated and decoded for playback.

- After decapsulation, audio and video streams are played back directly to avoid putting pressure on network bandwidth.
- Less demand is placed on the server.
- TCs can be used for 4K video playback.

Key Video Technologies of HDP

- Intelligent identification: The display server automatically distinguishes between video data and common GDI data. H.264 or MPEG2 is then used to encode the video and the TC takes care of the decoding.
- Dynamic frame rate: To ensure smooth playback, the playback frame rate is adjusted on the fly based on the network quality.
- Video data auto-adaptation: To ease pressure on the CPU and improve user experience, video streams are adjusted automatically based on the display resolution and the size of the video playback window.
- Multimedia redirection: TC hardware is used for decoding, dynamic traffic adjustment, 4K video playback, to reconnect automatically if the network connection is dropped. The TC hardware can provide smoother playback than Citrix ICA.
- Application sensitivity: Commonly-used video playback and image processing software (like Photoshop) are optimized based on customer demands.

9.3.7 HDP - Peripheral Redirection

In virtual desktops, peripherals on the TC/SC side are mapped to a remote desktop using desktop protocols. Depending on how they are used, peripheral technologies are classified as either port redirection or device redirection:

- Port redirection: The port protocols are redirected to the OSs of the remote desktops. Examples include USB port redirection, serial port redirection, and parallel port redirection.
- Device redirection: The device application protocols are redirected to the OSs of the remote desktops. Examples include camera redirection and TWAIN redirection.

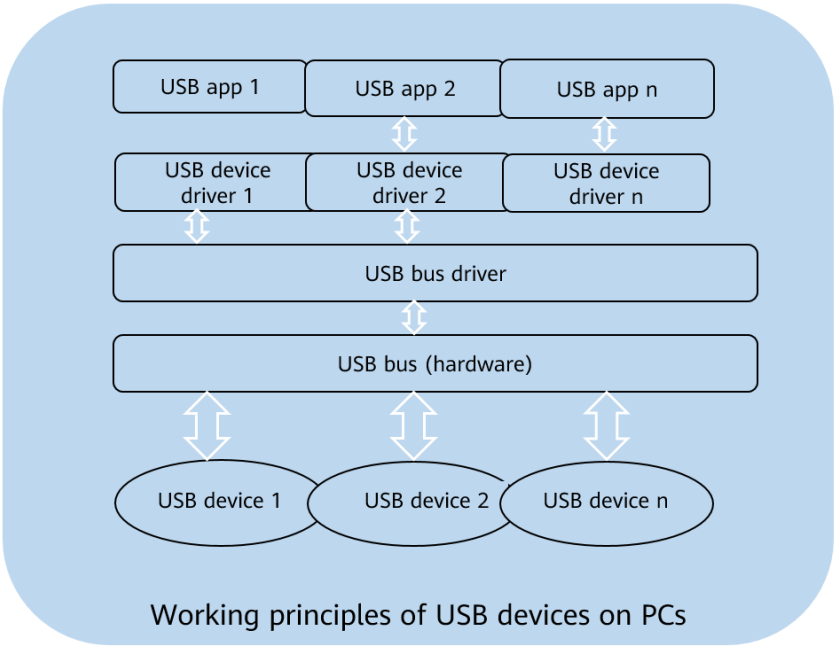


Figure 9-20 Working principles of USB devices on PCs

The preceding figure shows that the USB bus driver is essential for enabling USB devices to work normally at the software layer. When an application needs to use a USB peripheral, it must interact with the USB device driver. The USB device driver relies on the

USB bus driver to exchange data with the USB device and interacts with hardware using the bus driver as an agent.

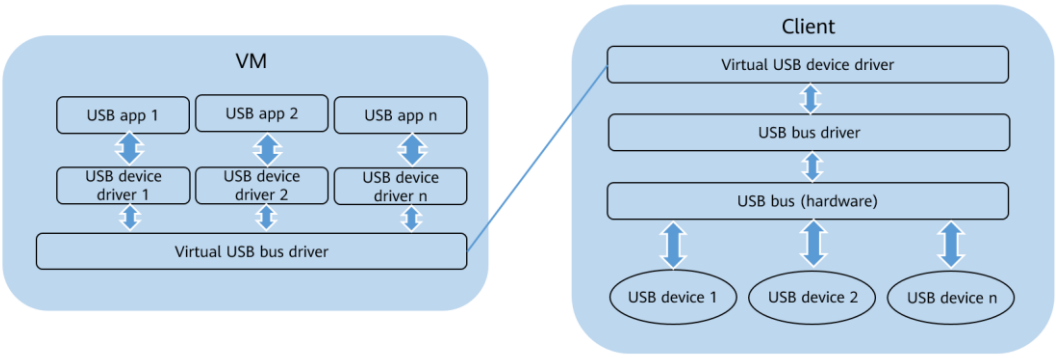


Figure 9-21 Working principles of USB port redirection on virtual desktops

The preceding figure shows the USB port redirection mode. A virtual USB bus driver is embedded in the VM and client respectively to use the remote physical USB bus driver. USB device drivers are installed and running on the VM and interact with the virtual USB bus driver. The USB device drivers and applications on the VM are not aware that the USB devices are running on a remote TC. USB port redirection is irrelevant to specific devices and applications and provides good compatibility because USB ports are redirected to desktop VMs. However, without compression and preprocessing at the device driver layer, graphics applications that are sensitive to network latency, such as applications of scanners and cameras, may consume a large amount of bandwidth. In this case, the device redirection technology must be used.

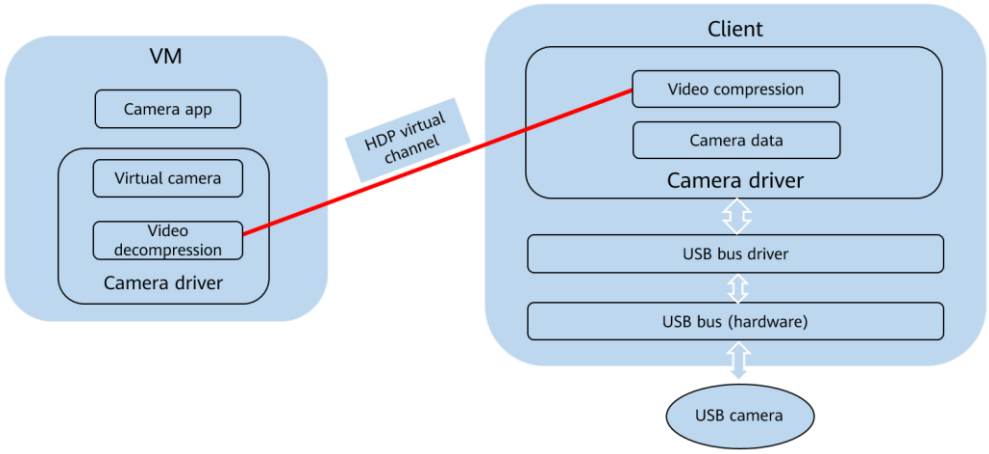


Figure 9-22 Working principles of USB port redirection on virtual desktops

The preceding figure shows the device redirection works at the device driver layer. A device driver is embedded in the TC and VM. The collected data is compressed and preprocessed by the device driver on the TC. The processed data is transmitted to the device driver on the VM using a desktop protocol. After being restored by the device driver on the VM, the data is transmitted to applications for processing.

If cameras are redirected in USB redirection mode, dozens of Mbit/s bandwidths are required, which cannot meet the requirements of commercial use. Therefore, specific optimizations are performed for USB peripherals to ensure that USB peripherals can be

commercially used in the virtual desktop system. We can use the camera redirection mode in Figure 9-22 to optimize cameras.

As shown in Figure 9-22, the client obtains camera data (bitmap data or YUV data) using an application-level interface, compresses the data using a video compression algorithm (such as H.264), and sends the compressed data to the server. The server decodes the camera data and provides the data to the application through the virtual camera. Compared with USB bus redirection mode, this camera redirection mode reduces bandwidth tenfold.

9.3.8 HDP - 3D Graphics Display Technology

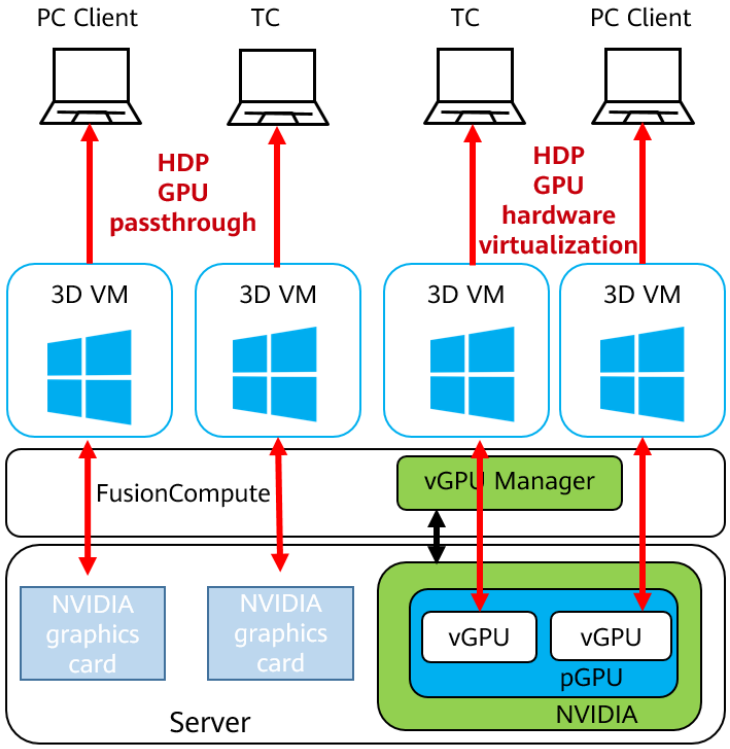


Figure 9-23 3D graphics technology

Huawei HD graphics desktop supports multiple HD graphics software products. There are three types of 3D graphics technologies used:

- GPU passthrough is used to bind a VM to each GPU, and each VM uses the GPU exclusively and accesses it using a driver. Equipped with GPU passthrough and HDP, the Huawei GPU passthrough HD graphics processing feature enables users to remotely access VMs to use GPU 3D acceleration. GPU passthrough is compatible with various types of graphics cards and supports the latest 3D applications that comply with DirectX and OpenGL.
- GPU hardware virtualization: Equipped with the vGPU technology, GPU hardware virtualization is used to virtualize a NVIDIA GRID graphics card into several vGPUs. Each vGPU is bound to a VM and the VM accesses the vGPU just like it accesses the physical GPU. By using the FusionCompute virtualization platform, the Huawei GPU hardware virutalization HD graphics processing feature allows virtualizing one physical GPU into several vGPUs. Each vGPU is bound to a VM and the VM

exclusively uses the vGPU. In this way, multiple VMs share a physical GPU, improving resource usage. A GPU can be shared by a maximum of 32 users and supports 3D applications that comply with the latest DirectX and OpenGL standards.

- Graphics workstation management: Graphics workstations are dedicated computers that specialize in graphics, static images, dynamic images, and videos. Graphics workstations are widely used in 3D animation, data visualization, CAD, CAM, and EDA that require high graphics processing capability. Graphics workstation management allows graphics workstations to be featured in FusionAccess and enables users to access the graphics workstations to use GPU 3D acceleration using HDP. It is compatible with various types of graphics cards and supports the latest 3D applications that comply with DirectX and OpenGL.

9.4 Introduction to FusionAccess Application Scenarios

9.4.1 FusionAccess Application Scenarios - Branch Offices

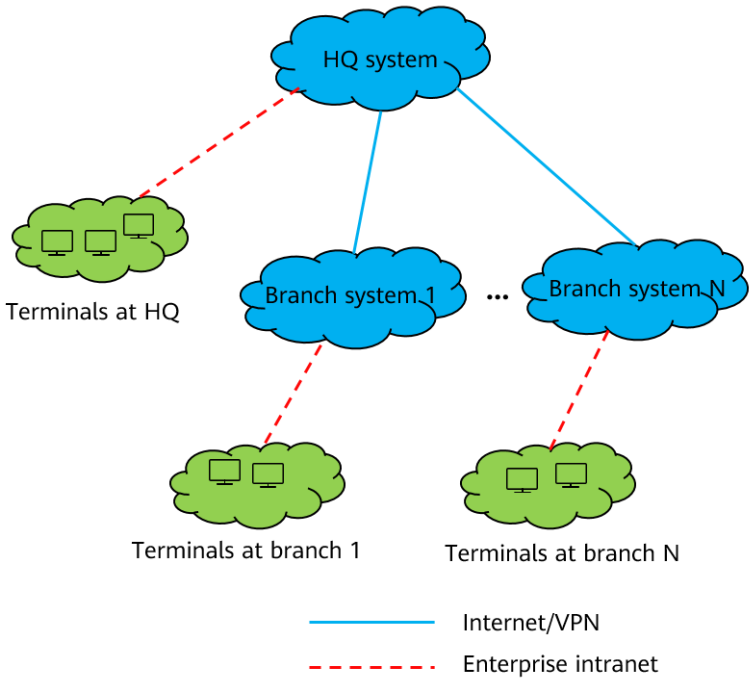


Figure 9-24 Branch offices

As shown in Figure 9-24, virtual desktops are often deployed in branch offices to improve user experience. Only management data is transmitted through the network between the headquarters and branches. Local traffic is used for VM remote desktops. This eliminates the need for network bandwidth. The minimum bandwidth required is 2 Mbit/s, and the latency is less than 50 ms. If virtual desktops are deployed in a centralized manner, high requirements are put on network bandwidth and latency for connecting to the virtual desktops remotely. If video and audio services are required, higher requirements are put on network bandwidth and latency. Deploying virtual desktops in the branch office

reduces the cost in building remote private networks and provides good VM user experience.

In addition, desktop management software is deployed in branches to ensure service reliability. Even if the WAN is disconnected, the VMs to which users have logged in can still run properly and branch services are uninterrupted.

An O&M system is deployed in the headquarters to implement centralized O&M of virtual desktops in the headquarters and branches.

9.4.2 FusionAccess Application Scenarios - Office Automation

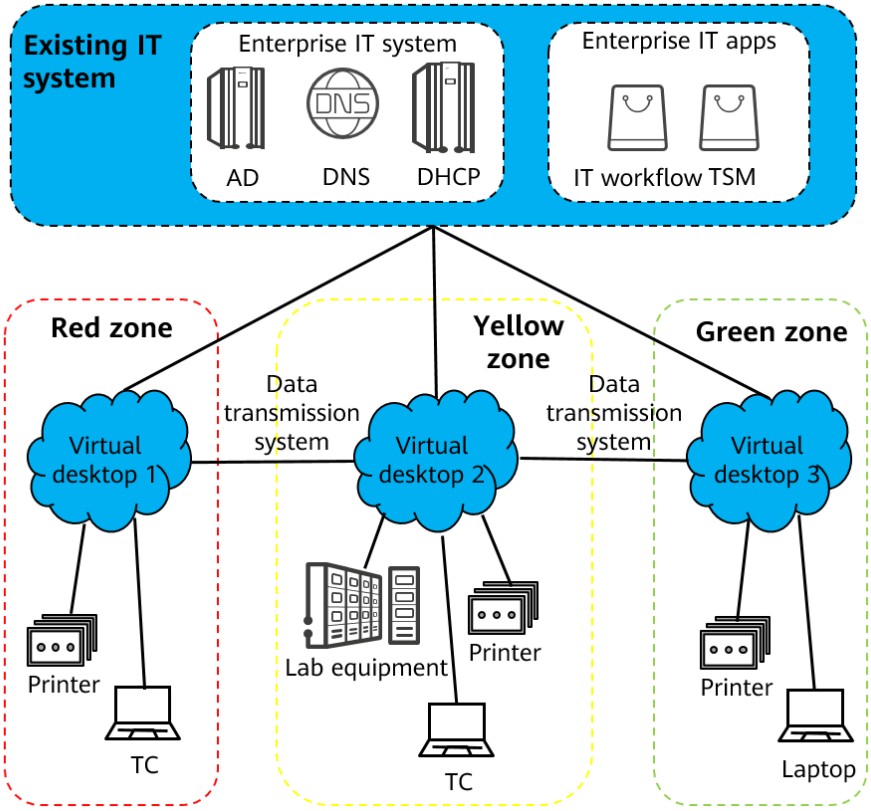


Figure 9-25 Office automation

As shown in Figure 9-25, FusionAccess allows users in an enterprise to handle work such as email processing and document editing, on a cloud computing platform and with high information security. FusionAccess can smoothly connect to enterprises' existing IT systems, allowing enterprises to make the most of their prior investments. For example, an enterprise can use the existing AD system to authenticate desktop users, and users can process existing IT workflows in FusionAccess. In addition, FusionAccess can assign IP addresses to virtual desktops using DHCP or resolve desktop domain names using an enterprises' existing DNS server.

FusionAccess uses various authentication and management mechanisms to ensure information security in workplaces.

- Users can use virtual desktops only after passing AD authentication.

- Data is stored by confidentiality in red, yellow, and green zones, which are separated from each other. Information in the red zone is top secret under the highest level of strict control. Information in the yellow zone is confidential and under a medium level of control. The green zone stores the least confidential information and is accessible by mobile users and from outside the enterprise.

The zone-based security control meets the security management requirements of most enterprises. It is easy and flexible to deploy.

Huawei has deployed about 70,000 desktops in its headquarters and branch offices and has successful OA desktop projects in commercial use worldwide.

9.4.3 FusionAccess Application Scenarios - GPU Desktop Professional Graphics

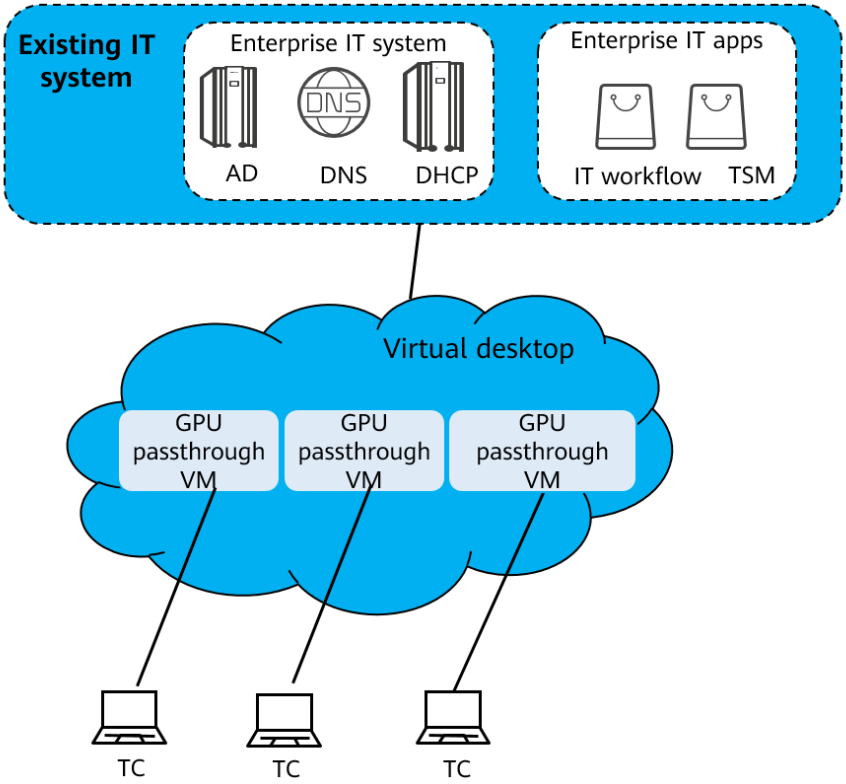


Figure 9-26 GPU desktop professional graphics

GPU virtual desktops provide powerful graphics processing capabilities in addition to CAD drawing and animation rendering.

Graphics software usually needs to invoke 3D instructions to achieve the optimal image display. The commonly used instructions are D3D and OPENGL that require the support of GPUs.

FusionAccess provides GPU passthrough and GPU hardware virtualization for graphics processing software.

Lower Costs: With FusionAccess, users do not need to purchase new PCs or servers, or even pay for software license upgrade. That is, instead of investing in assets that are depreciating, resources are directed towards other strategic investments.

Secure, Guaranteed, and Flexible: FusionAccess ensures that users can log in to the system using the Microsoft Remote Desktop Service protocol and restricts users' access to specific folders, applications, and files. This means that users can control data security. In addition, the virtual desktop will run on a dedicated server reserved for the user's company. This protection, together with centralized management of configuration files, helps companies improve compliance to ensure the security and privacy of user data.

Centralized Data Management: Data is centrally stored on a hosted desktop, helping users find important documents more quickly.

9.5 Quiz

Which virtual desktop architecture appeals to the industry trend, IDV or VDI?

10 FusionAccess: Planning and Deployment

FusionAccess installs various management components on VMs. These components are vAG, vLB, ITA, WI, GaussDB, and HDC. FusionAccess also interacts with other components on the live network: domain-related Active Directory (AD) components, Domain Name Service (DNS) responsible for domain name resolution, and Dynamic Host Configuration Protocol (DHCP).

This section describes both the component planning and overall installation process and the initial configuration process.

10.1 FusionAccess Component Installation Planning

10.1.1 FusionAccess Management Component Planning

10.1.1.1 Integrated Deployment

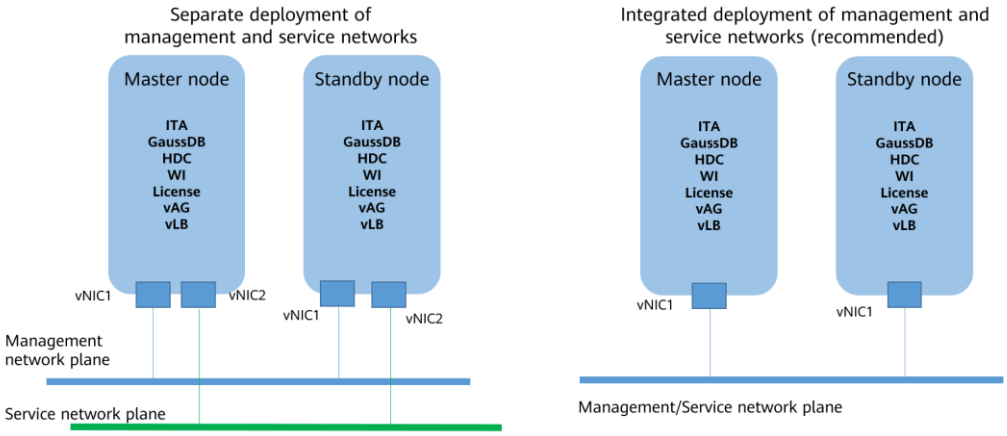


Figure 10-1 Integrated deployment

FusionAccess offers the following deployment schemes:

1. Integrated deployment (user quantity < 500)

All components are deployed together, as shown in Figure 10-1.

2. Standard deployment (500 ≤ user quantity ≤ 5,000)

The ITA/GaussDB/HDC/Cache/WI/License/LiteAS components are deployed together, and the vAG and vLB components are deployed together.

10.1.1.2 Standard Deployment

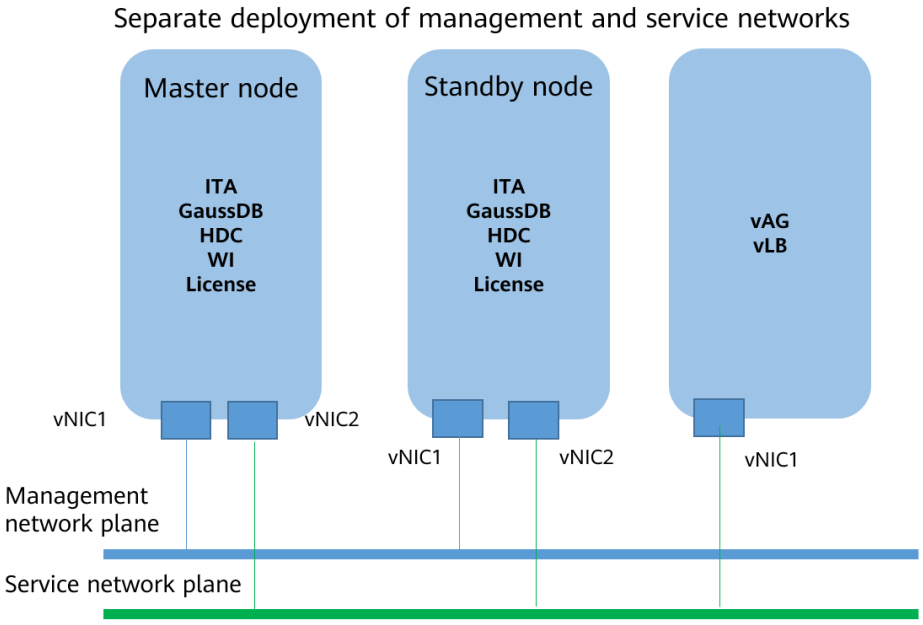


Figure 10-2 Separate deployment of management and service networks

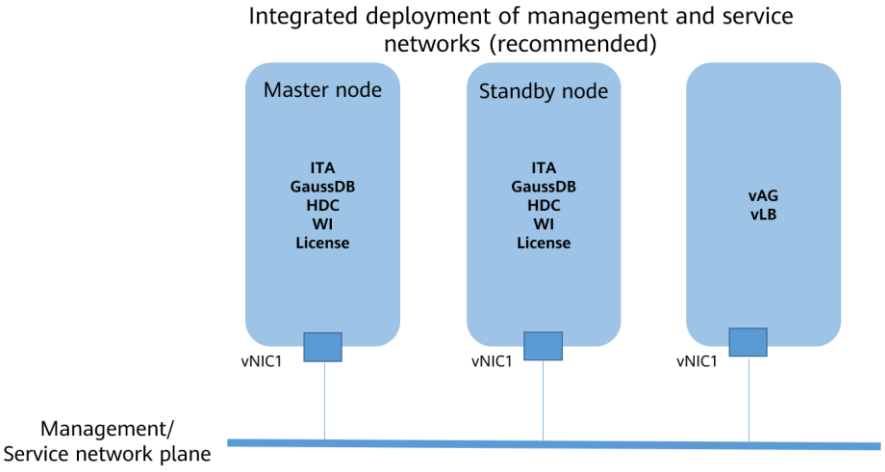


Figure 10-3 Integrated deployment of management and service networks

The preceding figures show FusionAccess component deployment schemes, which must comply with the following principles:

All management nodes must be deployed in active/standby mode at commercial sites.

The TCM must be deployed independently because its port conflicts with ITA, WI, UNS, and vLB ports.

The WI and UNS must be deployed separately because their ports conflict.

Standard deployment recommends deploying vLB and UNS separately.

High-performance storage is recommended for creating VMs with management components installed.

License should be deployed independently if multiple FusionAccess systems share a License.

10.1.1.3 Installation Details

Mode	VM	Deployment	OS	Hardware Specification	NIC
Integrated deployment	Linux infrastructure VM	Active/standby	Select **EulerOS 2.8 64 bit** for Arm architecture. Select **EulerOS 2.5 64 bit** for x86 architecture.	CPU: 8 cores Memory: 16 GB Disk: 60 GB	vNIC1: service plane vNIC2: management plane
Standard deployment	ITA/GaussDB/HDC/WI/License/LiteAS	Active/standby	Select **EulerOS 2.8 64 bit** for Arm architecture. Select **EulerOS 2.5 64 bit** for x86 architecture.	CPU: 8 cores Memory: 16 GB Disk: 60 GB	vNIC1: service plane vNIC2: management plane
	vAG	Multi-active	Select **EulerOS 2.8 64 bit** for Arm architecture. Select **EulerOS 2.5 64 bit** for x86 architecture.	CPU: 4 cores Memory: 4 GB Disk: 40 GB	vNIC1: service plane
	vLB	Active/standby	Select **EulerOS 2.8 64 bit** for Arm architecture. Select **EulerOS 2.5 64 bit** for x86 architecture.	CPU: 8 cores Memory: 4 GB Disk: 40 GB	vNIC1: service plane

Figure 10-4 Parameters of VMs in the FusionAccess component installation planning

Optional Linux infrastructure VMs include:

Backup Servers

TCM: Thin Client Manager

FusionAccess components can be installed on one VM or different VMs. The installation planning depends on the actual networking and requirements.

10.1.2 FusionAccess-associated Components

10.1.2.1 Active Directory (AD)

ADs store network resource information for query and usage. Such information includes user accounts, computer information, and printer information. AD is a directory service. It stores, searches, and locates objects and manages computer resources centrally and securely. AD provides directory management for medium- and large-sized networks on Microsoft Windows Server.

The directories in a Windows Server AD domain store user accounts, groups, printers, shared directories, and other objects.

AD manages and protects user accounts, clients, and applications, and provides a unified interface to secure information.

AD is a sophisticated service component for Windows Server OS. AD processes network objects, including users, groups, computers, network domain controllers, emails, organizational units, and trees in organizations.

The AD provides the following functions:

Basic network services: include DNS, Windows Internet Name Service (WINS), DHCP, and certificate services.

Server and client computer management: manages server and client computer accounts, and applies policies to all servers and client computers that are added to the domain.

User service: manages user domain accounts, user information, enterprise contacts (integrated with the email system), user groups, user identity authentication, and user authorization, and implements group management policies by default.

Resource management: manages network resources, such as printers and file sharing services.

Desktop configuration: allows the system administrator to centrally configure various desktop configuration policies, such as restricting portal functions, application program execution features, network connections, and security configurations.

Application system support: supports various application systems, including finance, human resources, email, enterprise information portal, office automation, patch management, and antivirus systems.

10.1.2.2 AD Objects

The smallest management unit of an AD is an object (a group of attributes). In an AD domain, the following basic objects are organized in the tree structure:

Domain controller: stores network domain controllers (equipment contexts).

Computer: stores computer objects added to the network domain.

Default account group (Builtin): stores in-house account groups.

User: stores user objects in the AD.

Organization Unit (OU): stores AD objects (users, groups, and computers) to reflect the AD organizational structure. This design enables objects to be managed using the organizational structure.

10.1.2.2.1 Domain Controller

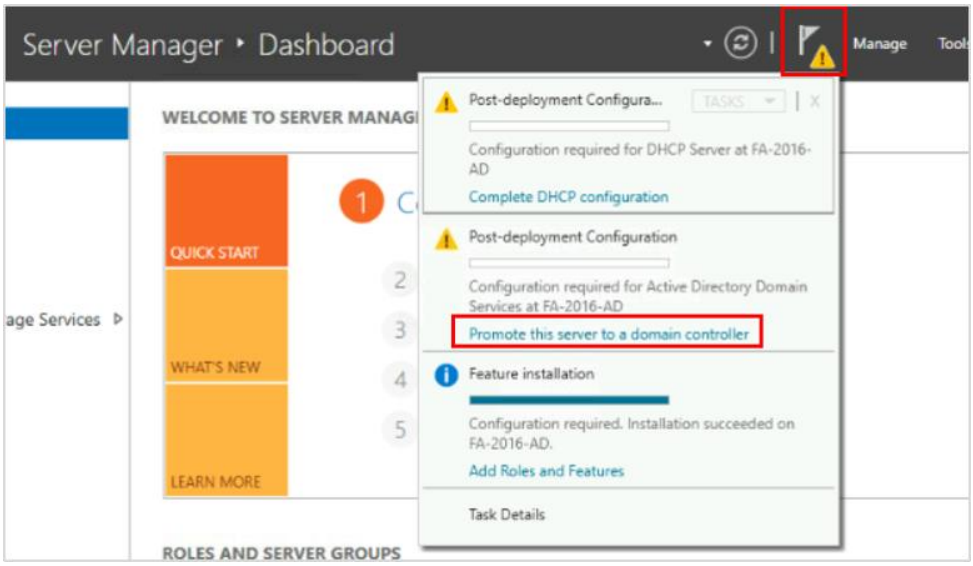


Figure 10-5 Domain controller

The directory data of the AD domain service is stored in domain controllers. There can be multiple domain controllers in a domain, and each is equally important. Data is synchronized between the domain controllers, so each domain controller stores a copy of the same AD database.

10.1.2.2.2 Domain User Account

Domain user accounts are created on domain controllers. This account is the only credential needed for domain access and is stored in the AD database of the domain as an AD object. When a user logs in to a domain from any computer in the domain, the user must provide a valid domain user account, which will be authenticated by the domain controller.

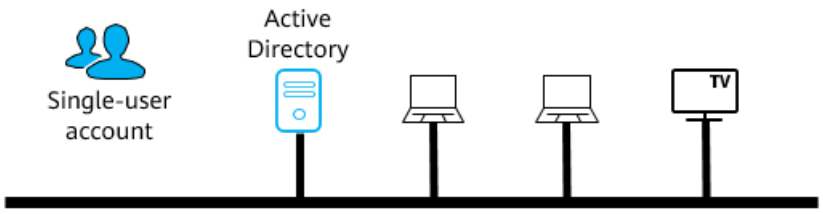


Figure 10-6 Domain users

10.1.2.2.3 Common Operations on Domain Accounts

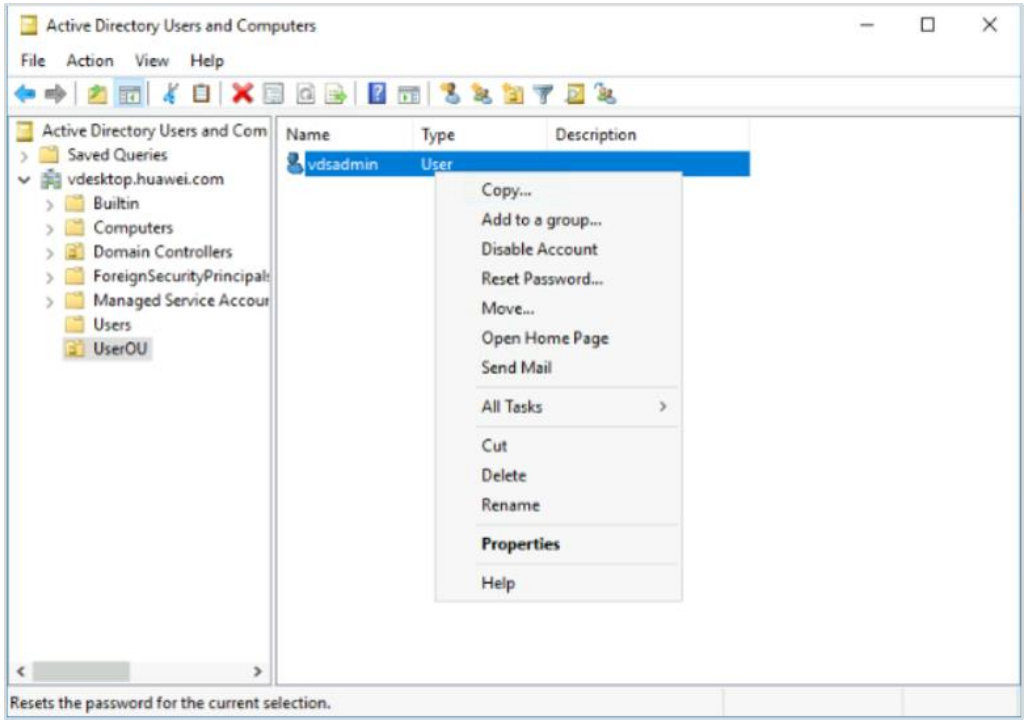


Figure 10-7 Common operations on domain accounts

Common operations on domain accounts include **Add to a group**, **Disable Account**, **Reset Password**, **Move**, **Delete**, and **Rename**.

10.1.2.2.4 User Domain Account Properties

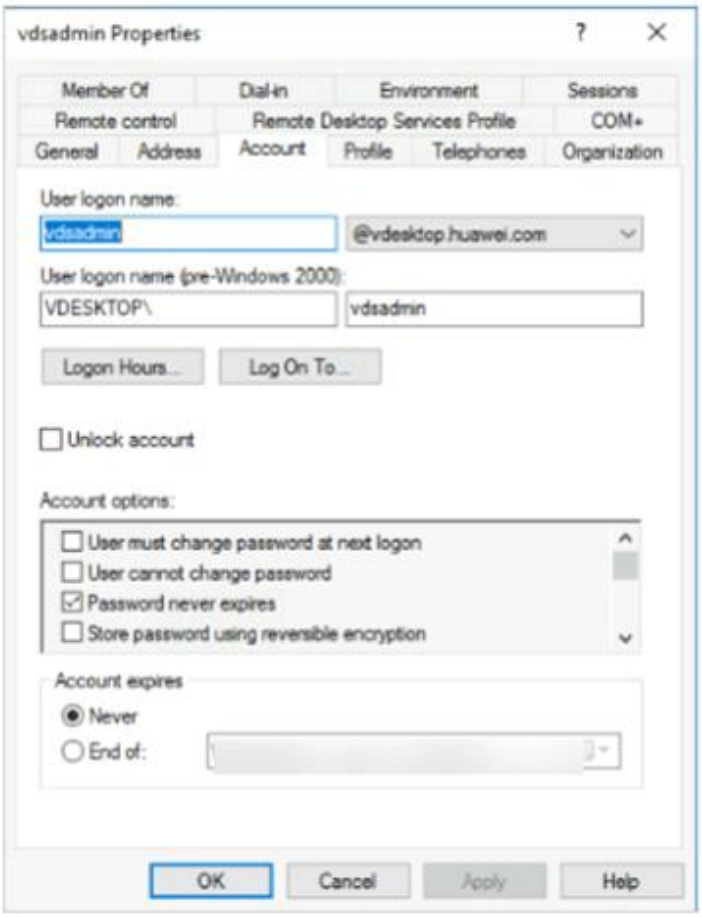


Figure 10-8 Domain account properties

You can right-click a domain account to view its properties, for example, you can view the username and change the account password options.

10.1.2.2.5 Finding a User Domain Account

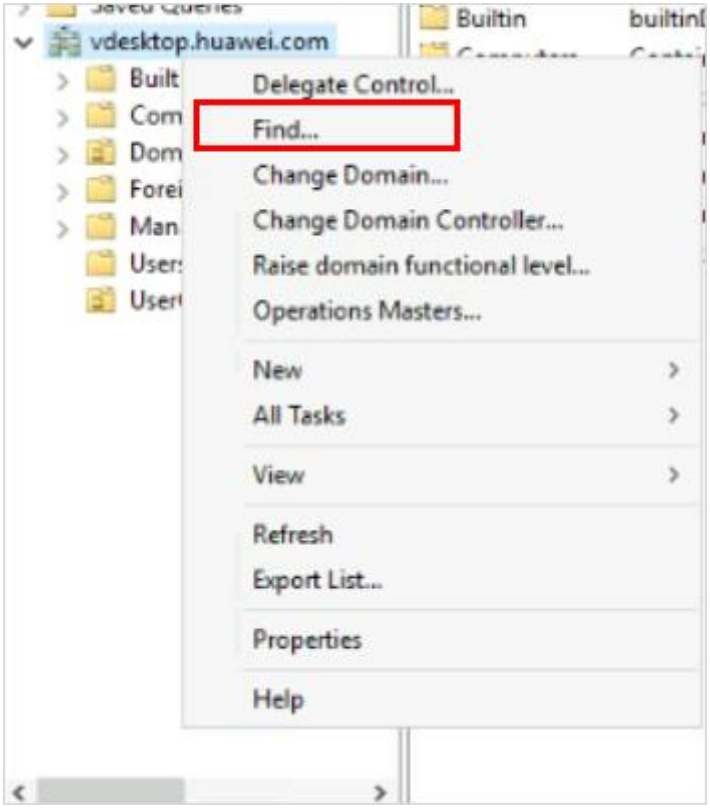


Figure 10-9 Finding a user domain account

You can right-click the domain controller, click **Find** from the shortcut menu, and enter a domain account name to quickly find the domain account.

10.1.2.2.6 User Group

A group is a logical collection of user accounts.

User groups manage accounts using in-domain resource access permissions.

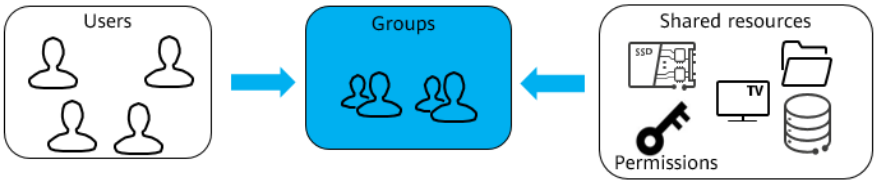


Figure 10-10 User groups

10.1.2.2.7 Groups in the AD

Groups simplify the allocation of resource permissions in the AD. You can group users who require the same resource permissions, instead of granting permissions to each user separately.

A user can join multiple groups. A group can be nested in another group.

10.1.2.2.8 Creating a User Group

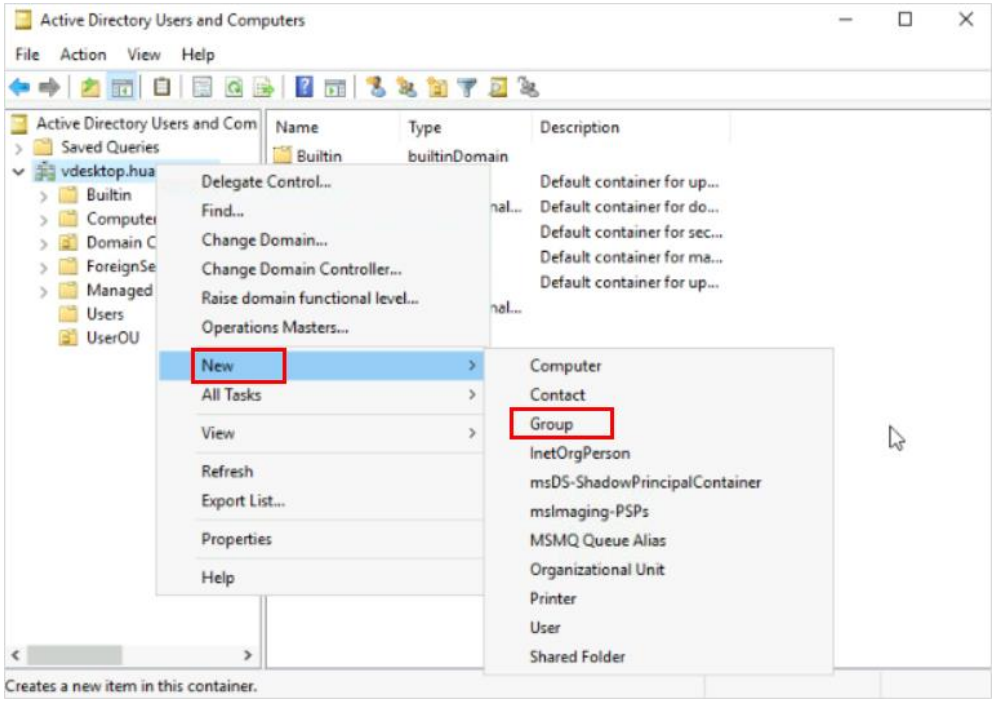


Figure 10-11 Creating a user group

The operation is similar to that of creating a user. You can right-click the AD domain controller and click **New** > **Group** to create a group.

10.1.2.2.9 Organization Unit (OU)

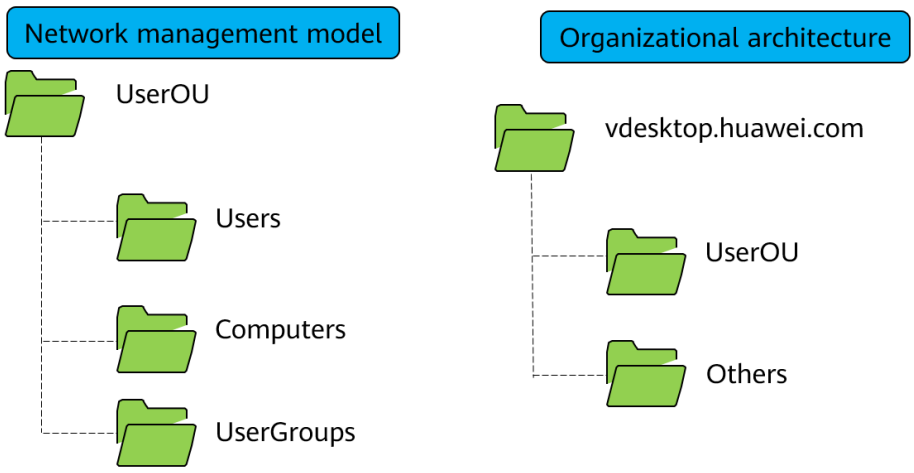


Figure 10-12 Organization unit (OU)

An OU organizes objects logically as required by an organization.

To delegate OU management and control rights, the permissions of the OU and its objects must be assigned to one or more users or groups.

10.1.2.2.10 Creating an OU

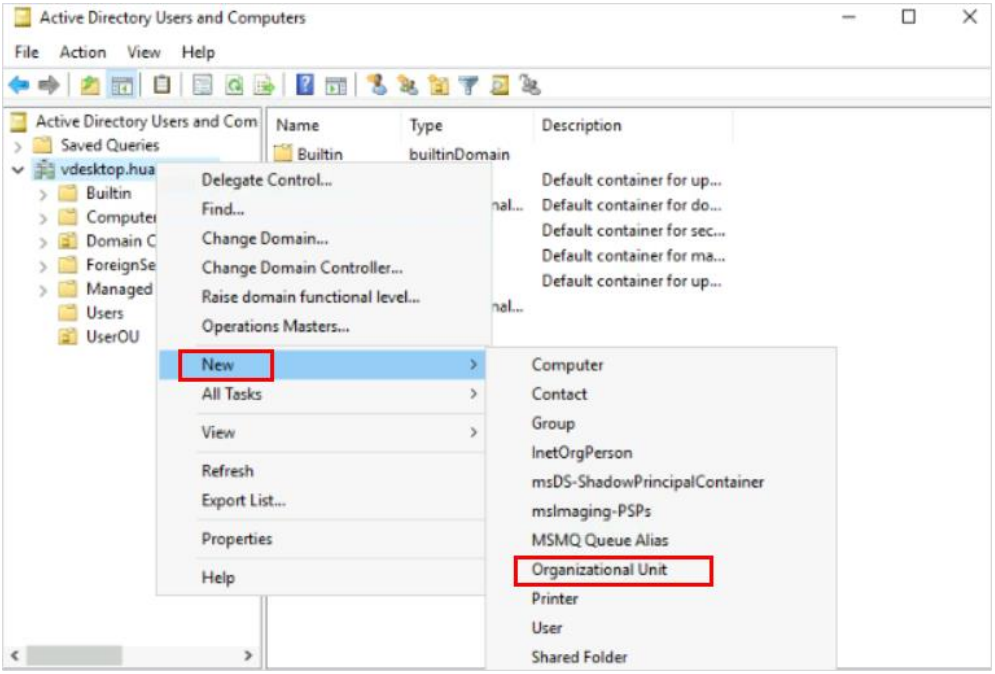


Figure 10-13 Creating an OU

Figure 10-13 shows the operations for creating an OU. When creating an OU, you should know:

OUs cannot be created in common containers.

Common containers and OUs are at the same level and do not contain each other.

OUs can be created only in domains or OUs.

10.1.2.2.11 Moving AD Objects Between OUs

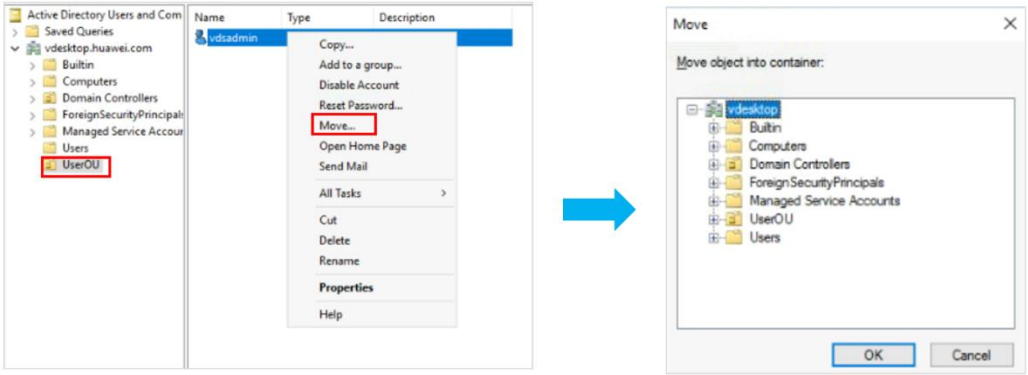


Figure 10-14 Moving AD objects between OUs

ADs allow user accounts to be moved between different OUs. After a user account is moved, the permissions assigned to the user account remain unchanged, but the user account uses the group policy of the new OU.

10.1.2.2.12 User Groups vs OUs

Similarity

OUs and user groups are AD objects.

Differences

A user group can contain only accounts.

An OU can contain accounts, computers, printers, and shared folders.

OUs have a group policy function.

10.1.2.2.13 Domains vs OUs

Similarity

OUs and domains are AD logical structures.

Both OUs and domains are the management unit of users and computers. They contain AD objects and configure group policies.

Differences

Users can log in to a domain but not an OU.

Domains are created before OUs.

OUs can exist in domains, but domains cannot exist in OUs.

A domain is at a higher level than an OU.

10.1.2.3 Adding a Computer to an AD Domain

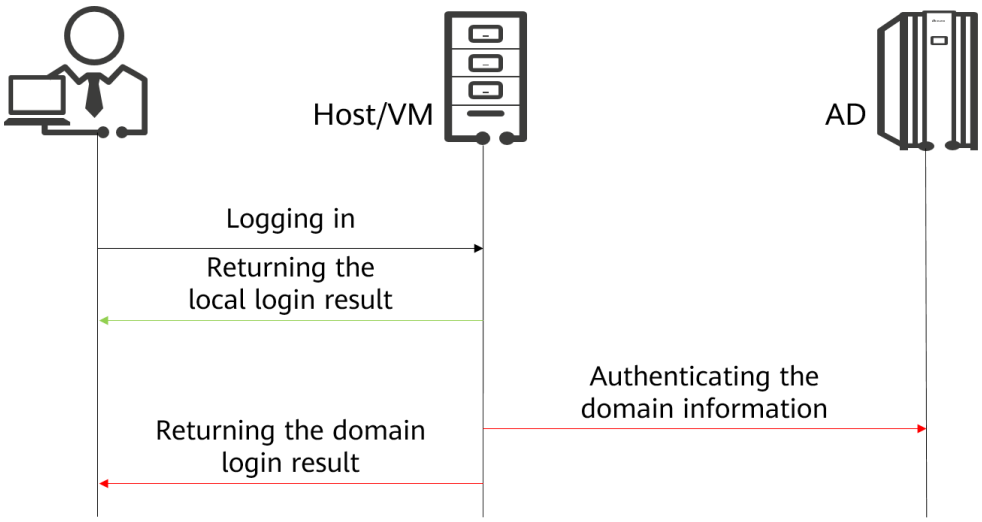


Figure 10-15 Adding a computer to an AD domain

As shown in Figure 10-15, if a user attempts to log in to the host, the system obtains the username and password, processes them with the key mechanism, and compares them with the key stored in the account database. If a match is found, the user is allowed to log in to the computer. If not, the login fails.

If a user attempts to log in to a domain, the system verifies whether the account information stored in the domain controller database is consistent with the information provided by the user. If yes, the user is allowed to log in to the domain.

10.1.2.4 Typical AD Desktop Applications

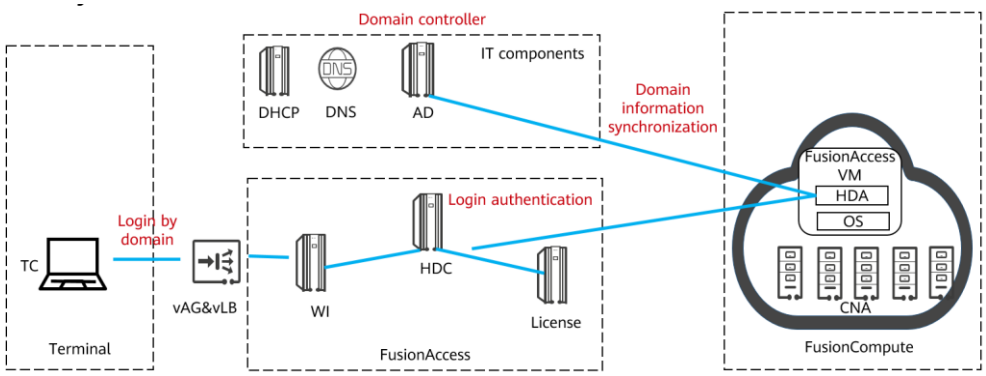


Figure 10-16 Typical AD desktop applications

As shown in Figure 10-16, the process for logging in to a VDI desktop as an AD domain user is as follows:

1. A user logs in to a desktop using the domain username.
2. The HDC sends a request to the AD for user information authentication.
3. A user VM synchronizes the domain information to a Domain Controller.

10.1.2.5 Domain Name System (DNS)

DNS is a distributed database that converts IP addresses and domain names for network access. DNS technology was created in 1983, with the original technical standards being released in RFC 882. The DNS technical standards were revised in RFC 1034 and RFC 1035 released in 1987, followed by the abolishment of RFC 882 and RFC 883. The later RFC versions have not seen any changes on the DNS technical standards.

DNS has the following advantages:

Users can access the network with easy-to-remember character strings, instead of IP numbers.

DNS cooperates with the domain controller.

A domain controller registers its role with the DNS server so that other computers can find its host name, IP address, and domain controller.

10.1.2.6 DNS Domain Name Structure

The DNS domain name management system includes: the root domain, top-level domain, second-level domain, subdomain, and host name. The structure of the domain name is like an inverted tree: The top of the structure is the roots and the highest level, while the leaves identify the lowest level.

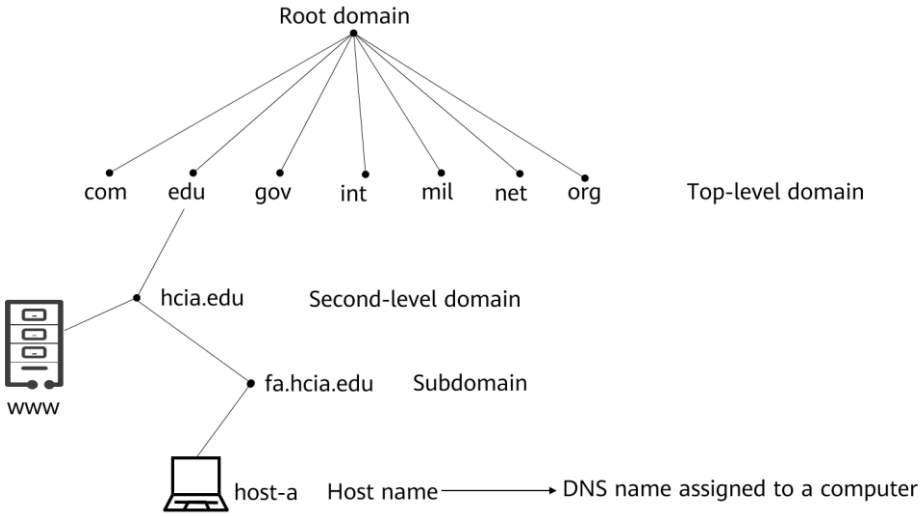


Figure 10-17 DNS domain name structure

Domain name structure resolution is as follows:

Common domains with three characters:

com: indicates a commercial organization.

edu: indicates an educational organization.

gov: indicates a governmental organization.

int: indicates an international organization.

mil: indicates a military site.

net: indicates a network.

org: indicates other organizations.

Country (region) domains with two characters:

cn: indicates the Chinese mainland.

tw: indicates Taiwan (China).

10.1.2.7 How DNS Works

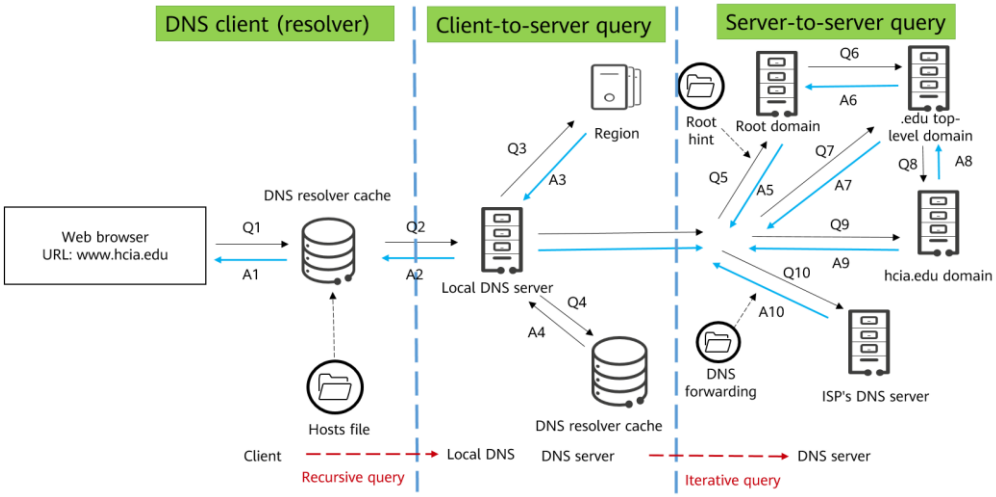


Figure 10-18 DNS work principle

The DNS server supports recursive query and iterative query.

Recursive query is a query mode of the DNS server. In this mode, after receiving a request from a client, the DNS server must return an accurate query result to the client. If the DNS server does not store the queried information locally, it queries other servers and sends the query result to the client.

Iterative query is the other query mode of the DNS server. In this mode, after receiving a request from a client, the DNS server does not directly return the query result but notifies the client of the IP address of another DNS server. The client then sends a request to the new DNS server. The procedure repeats until the query result is returned.

Terms:

hosts file: provides the mapping table between IP addresses and host names in static mapping mode, which is similar to the ARP table.

Domain: A domain is in the format of **abc.com** and can be divided into multiple zones, such as **abc.com** and **xyz.abc.com**.

The following shows three ways for the host of **www.abc.com** to query the IP address of the server of **www.xyz.abc.com**:

Recursive query:

Step 1: Query the IP address of a host of **www.xyz.abc.com** in the hosts static file and DNS resolver cache.

Step 2: If the query in step 1 fails, query the IP address in the local DNS server (domain server). That is, query the IP address in the region server and server cache.

Step 3: If the query in step 2 fails, query the IP address in the DNS server responsible for the top-level domain (.com) based on the root hints file.

Step 4: The root DNS server queries the IP address in the region server of **xyz.com**.

Step 5: The DNS server of **www.xyz.abc.com** resolves the domain name and returns the IP address to the host that sends the request along the same route.

Iterative query:

Step 1: Query the IP address of a host of **www.xyz.abc.com** in the hosts static file and DNS resolver cache.

Step 2: If the query in step 1 fails, query the IP address in all the region servers at the current level on the local DNS server (domain server).

Step 3: If the query in step 2 fails, query the IP address in all the region servers at the upper level. Repeat the query until the root DNS server.

Step 4: After reaching the root DNS server, query the IP address downwards until the IP address is found.

Combination of iterative query and recursive query:

Recursive query is a layer-by-layer query mode. For multi-layer DNS structure, the mode is inefficient. Therefore, the combination of iterative query and recursive query is generally used.

Step 1: Query the IP address of a host of **www.xyz.abc.com** in the hosts static file and DNS resolver cache.

Step 2: If the query in step 1 fails, query the IP address in the local DNS server (domain server). That is, query the IP address in the region server and server cache.

Step 3: If the query in step 2 fails, query the IP address in the DNS server responsible for the top-level domain (.com) based on the root hints file.

Step 4: The root DNS server directly returns the IP address of the DNS server in its zone to the local server, without querying in the region server of **xyz.com**.

Step 5: The local DNS server returns the result to the host sending the request.

10.1.2.8 DNS Forward Lookup

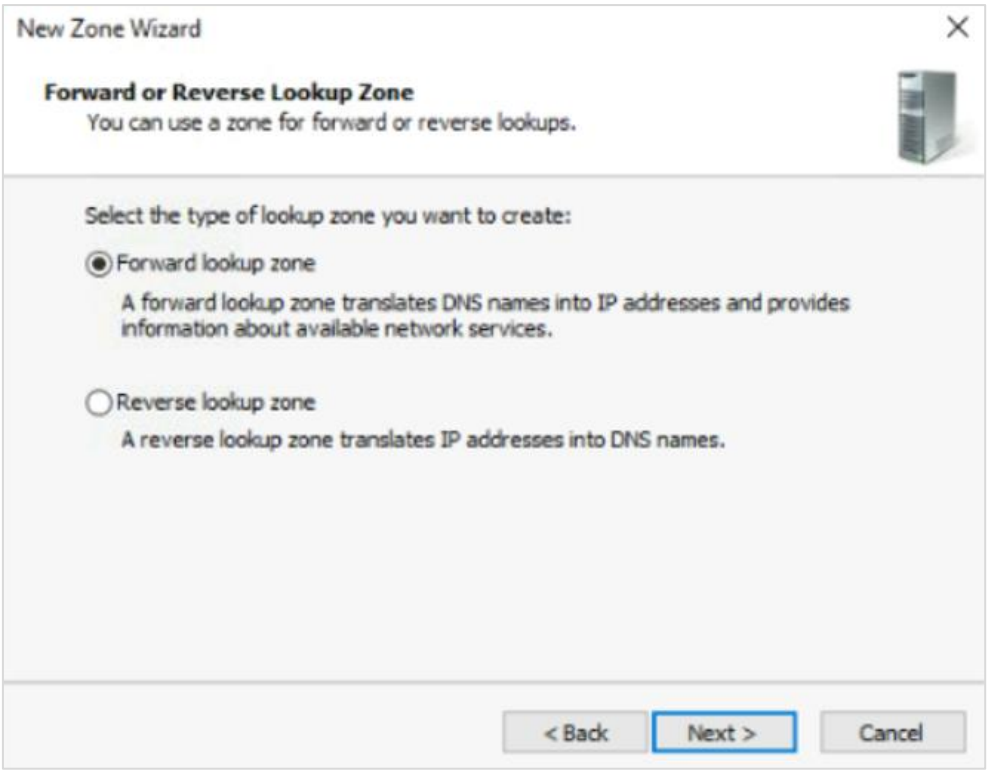


Figure 10-19 DNS forward lookup

DNS forward lookup needs a forward lookup zone - the zone where forward lookup is used in the DNS domain name space. Forward lookup resolves the domain names provided by DNS clients to IP addresses.

10.1.2.9 Adding a DNS Record

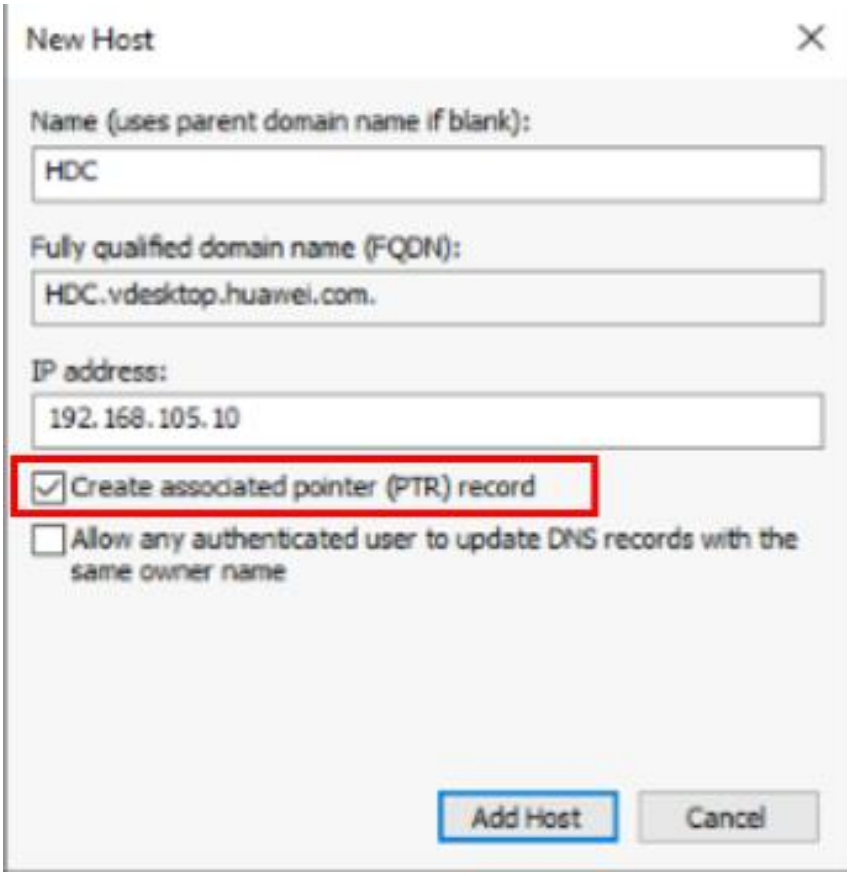


Figure 10-20 Adding a DNS record

After creating a forward lookup zone, create the host (host01) record for the zone. The record is used to map the DNS domain name to the IP address used by the computer.

If you select **Create associated pointer (PTR) record** when creating a host record in the forward lookup zone, you add a pointer to the reverse lookup zone.

10.1.2.10 DNS Reverse Lookup

Figure 10-21 DNS reverse lookup

A reverse lookup zone needs to be established to resolve an IP address to a domain name.

After creating a reverse lookup zone, create a record pointer for the zone. This pointer is used to map the IP address of the forward DNS domain name computer to the reverse DNS domain name.

10.1.2.11 Client DNS Configuration

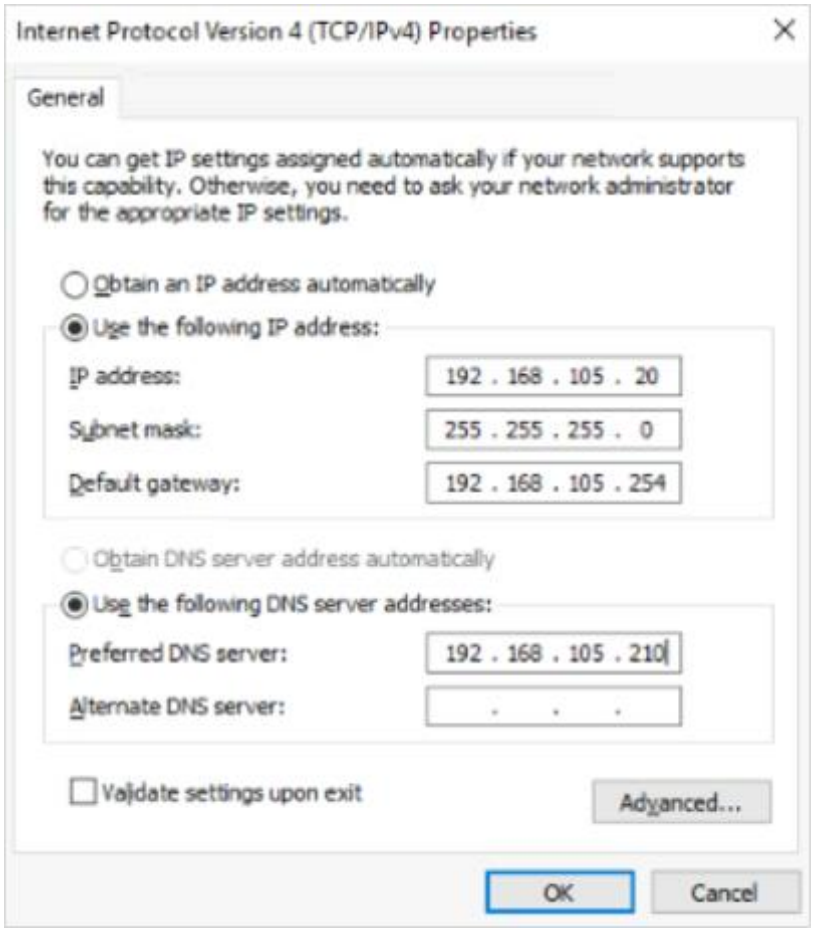


Figure 10-22 Client DNS configuration

You can configure the DNS server address for clients.

10.1.3 DNS Working Process

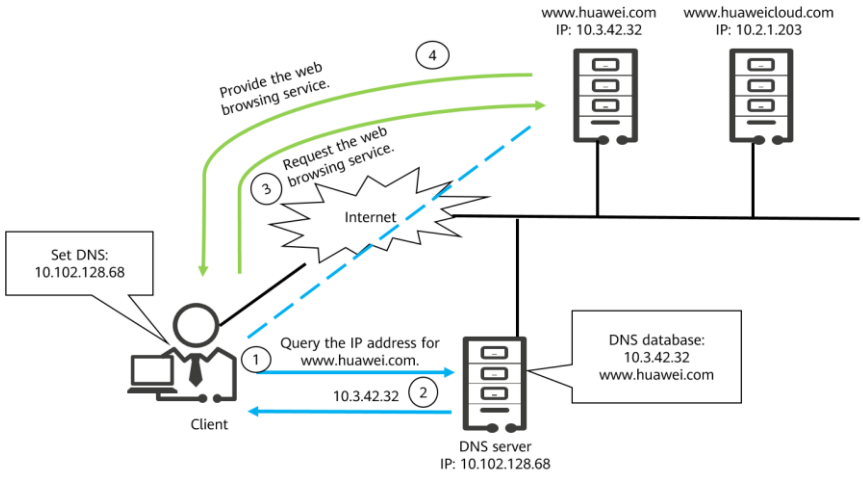


Figure 10-23 DNS working process

The process of accessing **www.Huawei.com** on a client is as follows:

The client sends a query request to the destination DNS server to query the IP address of **www.Huawei.com**.

The DNS server returns the IP address of the domain name to the client.

The client finds the corresponding web server based on the returned IP address and accesses the web page.

The web server returns the information about the deployed web page to the client.

10.1.3.1 DNS Resolution in FusionAccess

Domain name used for logging in to vLB/WI

To log in to VMs, users must configure the required domain names on the DNS server.

HDC computer name

When registering with HDC, user VMs must use the HDC domain name to find its IP address on the DNS server for authentication.

10.1.3.2 Dynamic Host Configuration Protocol (DHCP)

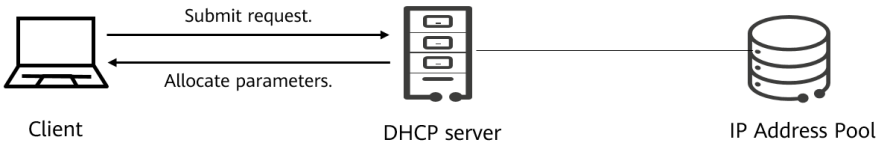


Figure 10-24 DHCP

DHCP is a network protocol used for IP networks. It is located at the application layer of the OSI model and uses the UDP protocol. DHCP provides the following functions:

Automatically assigns IP addresses to users for the intranet or network service providers.

Centrally manages all computers for the intranet administrator.

DHCP is a communication protocol that enables network administrators to centrally manage and automatically assign IP addresses. On an IP network, each device connected to the Internet must be assigned a unique IP address. With DHCP, network administrators can monitor and assign IP addresses from the central node. DHCP uses the concept of lease, which is also called the validity period of the computer IP address. The lease period depends on how long it takes a user to connect to the Internet in a place.

10.1.3.3 Necessities of DHCP

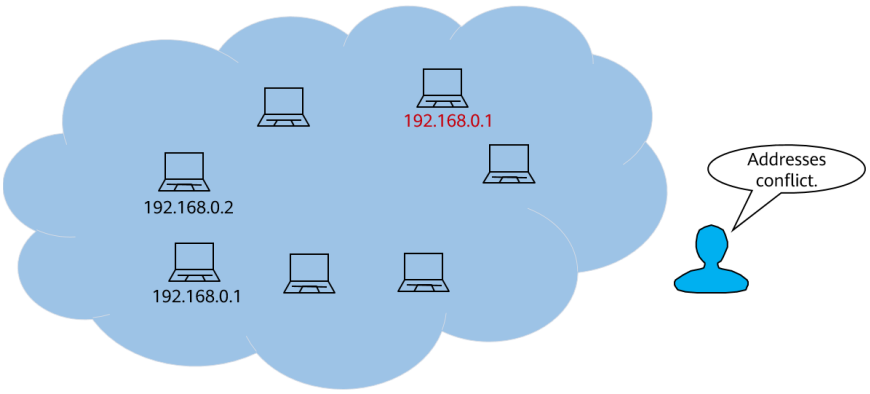


Figure 10-25 Necessities of DHCP

In larger networks, client IP addresses allocated by different users may be the same.

In a TCP/IP network, each workstation must perform basic network configurations before accessing the network and its resources. Mandatory parameters include the IP address, subnet mask, default gateway, and DNS. Required information includes IP management policies.

In larger networks, it is difficult to ensure that all hosts have correct configurations. Manual network configuration is especially difficult for dynamic networks that contain roaming subscribers and laptops. Computers are often moved from one subnet to another or out of the network. It may take a long time to manually configure or reconfigure a large number of computers. If an error occurs during the configuration of an IP host, the communication with other hosts on the network may fail.

To simplify IP address configuration and centralize IP address management, DHCP was designed by the Internet Engineering Task Force (IETF).

10.1.3.4 DHCP Advantages

DHCP reduces errors.

DHCP minimizes manual IP address misconfiguration, such as address conflict caused by the reallocation of an assigned IP address.

DHCP simplifies network management.

With DHCP, TCP/IP configuration is centralized and automatic. The network administrator defines TCP/IP configuration information of the entire network or a specific subnet. DHCP automatically allocates all additional TCP/IP configuration values to clients. Client IP addresses must be updated frequently. For example, a remote access client moves frequently, but frequent updates enable efficient and automatic configuration when it restarts at a new location. At the same time, most routers can forward DHCP configuration requests, so DHCP servers do not usually need to be configured on each subnet.

10.1.3.5 DHCP Relay Switch

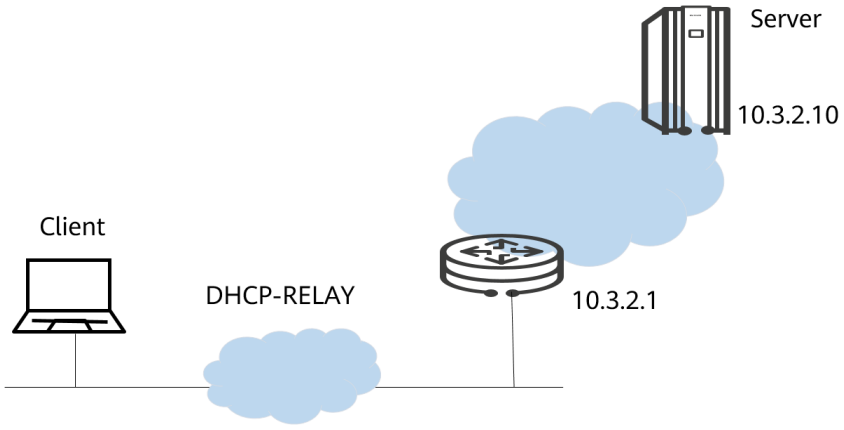


Figure 10-26 DHCP working process

By default, a DHCP server configures the network only for computers on the same network. If a DHCP server configures the network for the computers on another network, you can configure a DHCP-RELAY to process and forward DHCP information between different subnets and physical network segments.

10.1.4 FusionAccess-associated Component Installation Planning

10.1.4.1 Integrated Deployment

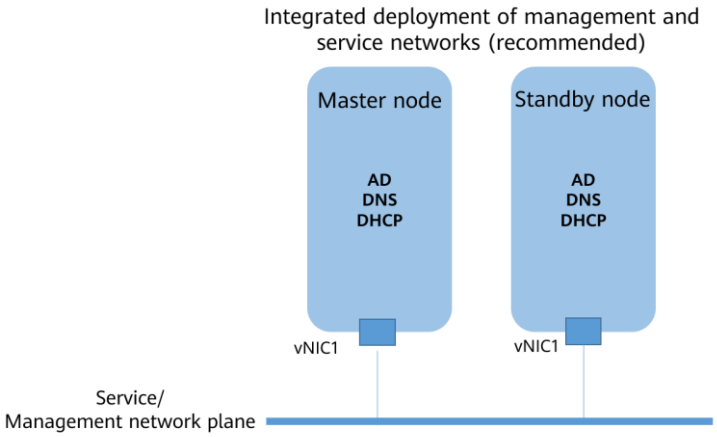


Figure 10-27 Integrated deployment of FusionAccess-associated components

Figure 10-27 shows the integrated deployment of FusionAccess-associated components. The active and standby VMs must be deployed on different CNAs. Figure 10-28 shows the specifications of FusionAccess-associated components.

VM	Deployment	OS	Hardware Specification	NIC
AD/DHCP/DNS	Active/standby	Windows Server 2016 R2 Standard 64 bit	CPU: 2 cores Memory: 2 GB Disk 1: 50 GB (system disk) Disk 2: 20 GB (backup disk)	vNIC1: service plane

Figure 10-28 Specifications of Windows components

For details about FusionAccess installation process and initial configuration, see *HCIA-Cloud Computing V5.0 Lab Guide (FusionAccess)*.

10.2 Quiz

Can FusionAccess work properly without the AD/DNS/DHCP components?

11 FusionAccess: Service Provisioning

Clone technology is important for virtual desktops. It enables templates to be batch-deployed on desktops. Clone is divided into full copy and linked clone, and further derives full memory and QuickPrep.

This section introduces full copy and linked clone, differences between virtual desktops, and the process of provisioning virtual desktops on FusionAccess.

11.1 Service Encapsulation

11.1.1 Background of Clone

Virtual desktop technologies enable the batch provisioning and O&M of office desktops, and make enterprise IT management easier. Clone is one important technology. Administrators clone a parent VM/template on more VMs, facilitating IT management and O&M. The cloned VMs have the same OS, application systems, data, and documents as the parent VM.

Clone is divided into full copy and linked clone, and QuickPrep is derived from full copy.

11.1.2 A Full Copy Desktop

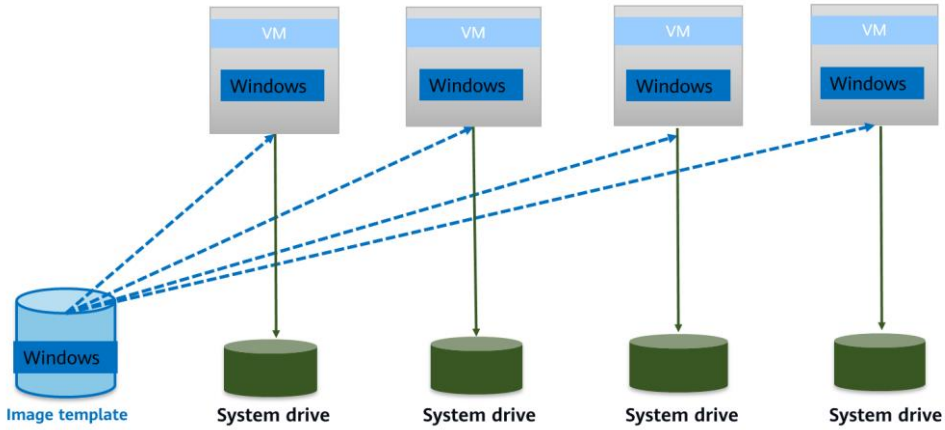


Figure 11-1 A full copy desktop

As shown in Figure 11-1, a full copy VM is an independent computer that is created using a source computer (not joining a domain) template. Full copy VMs have the following characteristics:

Users can save data changes (such as installed software) on computers.

Target computers have their own CPU, memory, and disk resources.

Each computer needs to be maintained separately (for such operations as software upgrades and antivirus database updates).

After a VM is shut down, users can save their customized data.

Restoration upon shutdown is not supported.

The one-click restoration function is supported.

11.1.3 Principles of Full Copy

A full copy VM is a complete and independent copy of a parent VM (VM template). It is independent of the parent VM - modification and deletion of the parent VM do not affect the running of the full copy VM.

11.1.4 Characteristics of Full Copy

Each full copy VM is an independent entity, and data changes (such as software installation) on each VM can be saved.

However, both the parent VM and each full copy VM use independent CPU, memory, and disk resources. Users need to maintain software (such as upgrades or antivirus updates) on each full copy VM.

11.1.5 QuickPrep VMs

The principles of QuickPrep VMs are as follows:

A QuickPrep VM is not encapsulated using Sysprep, but is renamed and added to the domain by applications in the VM.

There is no essential difference between full copy and QuickPrep.

The advantages of QuickPrep VMs are as follows:

The QuickPrep template is more efficient at VM provisioning than the full copy template.

The local SIDs of all computers provisioned using the QuickPrep template are the same. Some industry software uses local SIDs to identify computers. The software regards all computers provisioned using the QuickPrep template as one VM, causing installation and usage problems. You are advised to provision a small number of test VMs when using a QuickPrep template to ensure that the software required by customers can run properly.

11.1.6 A Linked Clone Desktop

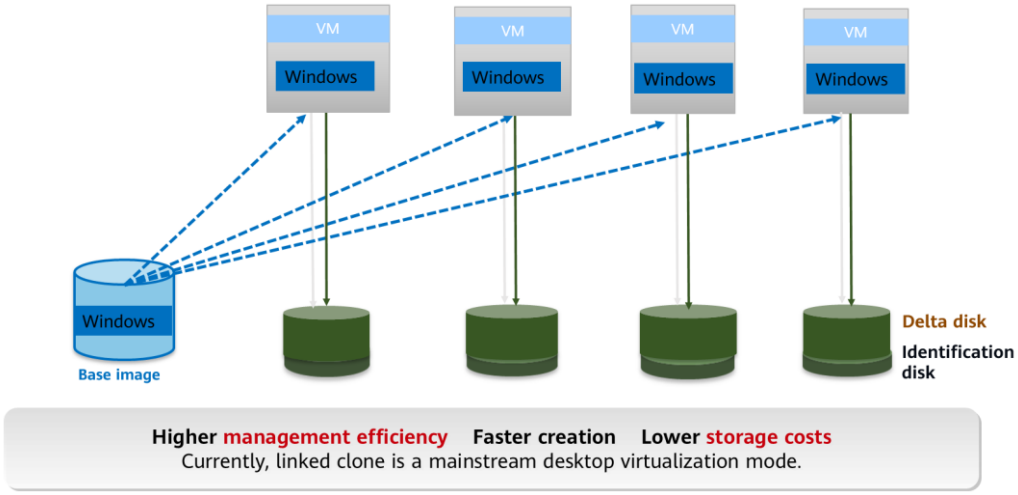


Figure 11-2 A linked clone desktop

For a linked clone computer that is created using a source computer (joining a domain) template:

Multiple linked clone computers of a source computer can share the same disk space. The linked clone base disk allows a maximum of 128 clone volumes.

With the same server resources, more virtual desktops can be created in linked clone mode than in full copy mode, reducing enterprise IT costs.

Computers can be updated in batches using a template (for such operations as software upgrades and antivirus database updates).

Automatic restoration upon shutdown is supported.

A linked clone computer can be created quickly.

11.1.7 Principles of Linked Clone

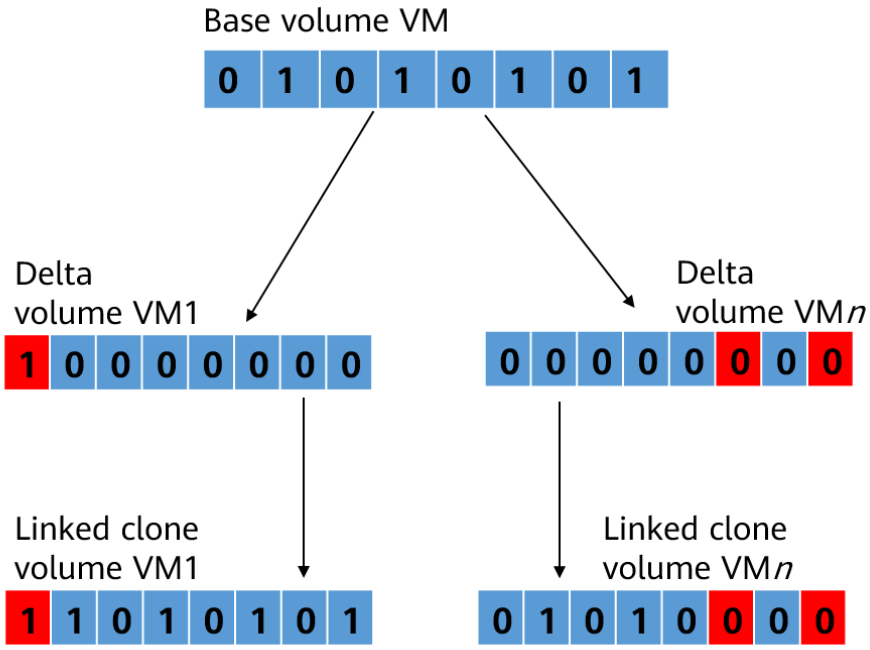


Figure 11-3 Principles of linked clone

The data in both the base and delta volumes is the data in the linked clone volume. The linked clone base volume is read-only and shared by multiple linked clone VMs.

The linked clone delta volume can be read and written. Each linked clone VM has a delta volume for storing differentiated data.

The linked clone technology features fast creation and small storage usage, and is applicable to homogeneous users and highly standardized desktops.

Because the base disk is shared by many desktops, it must offer high read performance.

11.1.8 Advantages of Linked Clone

Administrators upgrade multiple systems and install system patches and new software for linked clone VMs together.

A shared base disk means no need to copy the system disk.

The delta disks of the linked clone VMs store temporary user data, which can be automatically deleted when the VMs are stopped.

AD stores the customized configurations and data of users.

To update the base disk, the original linked clone template is cloned as a VM, and then this VM is started to update related systems. After that, this VM is converted to a template, and the function of updating VM group software is used. The O&M is simplified to ensure better IT system security and reliability.

If users of linked clone desktops need to store customized configurations and data, configure profile redirection or folder redirection on the AD for these users. The redirection storage location can be a remote file server directory, a web disk, or the data

disk of a linked clone VM. Customized configurations and data stored in remote file server directories or web disks can roam to the corresponding desktop to which the user logs in.

11.1.9 Benefits of Linked Clone

Linked clone improves efficiency and saves costs.

VMs can be created in seconds, so overall provisioning is faster.

A large amount of storage space can be saved, lowering enterprise IT costs.

Unified system updates and patch installation for linked clone VMs make O&M more efficient and cost-effective.

11.1.10 Template, Base Volume, and Delta Volume

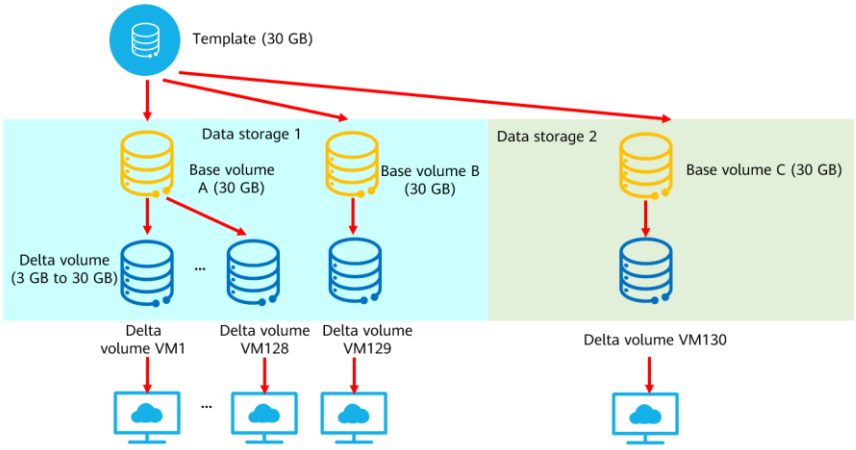


Figure 11-4 Template, base volume, and delta volume

As shown in the figure 11-4, if the size of a linked clone template is 30 GB, when a linked clone VM is created on data storage 1, the template is automatically copied to generate a 30 GB base volume A, and then the automatic snapshot function is used to create a delta volume for each VM. When there are 128 delta volumes on the base volume A, the system automatically generates a base volume B in data storage 1 and creates delta volumes for other linked clone VMs. A maximum of 128 delta volumes can be created for each base volume to prevent high I/O pressure when all VMs are running.

The delta disk created for each VM adopts thin provisioning. The initial size of each delta disk is nearly 0 GB. To store data, the estimated size of a delta disk is no less than 3 GB but not greater than that of a template. Generally, 5 GB, 10 GB, or 12 GB capacity is estimated for a delta disk based on application scenarios and restoration frequency of linked clone VMs.

As shown in the figure 11-4, the base disk and delta disk must be deployed on the same data storage. The template can be deployed on another data storage. Linked clone VMs can be created only on data storage that supports thin provisioning.

11.1.11 Full Copy Use Case: Personalized Office

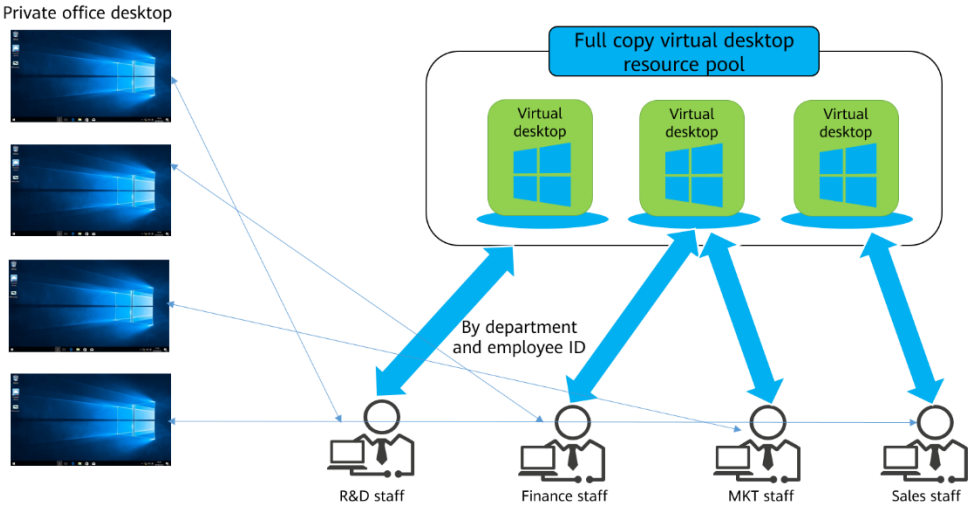


Figure 11-5 Personalized office using full copy virtual desktops

In personalized OA scenarios, each employee from different departments may have different requirements on desktop settings, so they need customized desktops.

Features of a full copy virtual desktop are as follows:

It is an independent computer that is created using a source computer (not joining a domain) template.

Users can save data changes (such as installed software) on computers.

Target computers have their own CPU, memory, and disk resources.

Each computer needs to be maintained separately (for such operations as software upgrades and antivirus database updates).

After a VM is shut down, users can save their customized data.

Restoration upon shutdown is not supported.

The one-click restoration function is supported.

11.1.12 Linked Clone Use Case: Public Reading Room

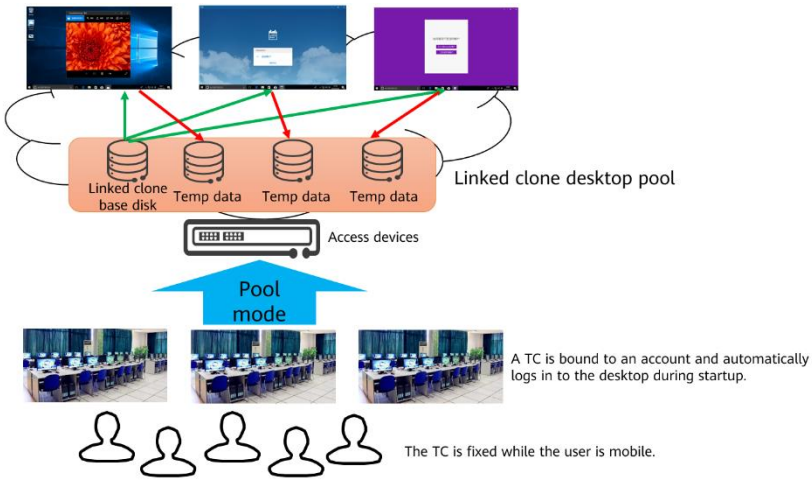


Figure 11-6 Public reading room using linked clone virtual desktops

In an electronic reading room, users only need to log in to and use VMs. The reading software has been contained in image files, and the service is simple. An electronic reading room has the following characteristics:

Users can access the Internet. The viruses and Trojan on the network are difficult to prevent.

Users are not fixed and VMs do not need to be shut down.

USB flash drives must be supported.

Maintenance is simple.

The electronic reading room has low storage requirements but faces security threats. Linked clone desktops are suitable for the electronic reading room. Linked clone VMs share a read-only base disk. The base disk is preinstalled with the required applications to prevent viruses or Trojan. When users log in to the linked clone VMs, temporary data generated during Internet access and web page browsing is stored on delta disks. If the delta disks are attacked by viruses and Trojan, you only need to restart the VMs to clear data on the delta disks. If VMs need to be upgraded or patches need to be installed on VMs, the administrator only needs to update the base disk.

VMs can be assigned to multiple users dynamically for better resource reusability. Each TC is bound to a fixed VM account, and the TC can be logged in to once powered on. Users do not need to enter accounts or passwords when they log in.

11.1.13 Full Copy vs Linked Clone

The major difference is in the system disk:

Multiple linked clone VMs share the same base disk. Each cloned VM has a delta disk to record write data to its system disk. Data includes temporary data, customized configurations saved in **C:\User**, and temporary personalized applications saved in **C:\Program Files**. Together, the base disk and the delta disk constitute the system disk (drive C) of a linked clone VM.

Administrators can forcibly reinitialize linked clone VMs and update applications for VMs in the linked clone VM group in a unified manner to update system base disks of linked clone VMs.

11.1.14 Comparison of Desktop VMs

Type	Provisioning Mode	Description	Provisioning Rate	Key Features of Desktop Components
Full copy	Full copy	Each VM has a system disk with independent storage space, repeated data results in storage wastage, VM creation is slow, and VMs lack unified update/restore.	Slow	Each VM is independent and can store personalized data.
	QuickPrep	Same as the above.	Medium	Each VM is independent and can store personalized data. VMs using the same template have the same SID.
Linked clone	Linked clone	Multiple VMs share a base disk but have an independent thin-provisioning delta disk, less storage space is used, VM creation is fast, and VMs have unified update/restore	Fast	Multiple VMs share a system volume. VMs created using the same template have the same SID. VMs can be restored after shutdown but do not save personalized data.

Figure 11-7 Comparison of desktop VMs

11.2 Template Creation

11.3 Virtual Desktop Provisioning

For details about template creation and virtual desktop provisioning, see *HCIA-Cloud Computing V5.0 Lab Guide (FusionAccess)*.

11.4 Quiz

Can the disk configuration mode be common or thick provisioning lazy zeroed when you provision a linked clone desktop? If not, why?

12 FusionAccess: Features and Management

Huawei FusionAccess lets you manage virtual desktops by customizable policies and service adjustment. In case of a fault, an alarm is generated to help you resolve the fault quickly.

The chapter describes FusionAccess policy management, service adjustment and alarm handling.

12.1 Policy Management

12.1.1 Overview

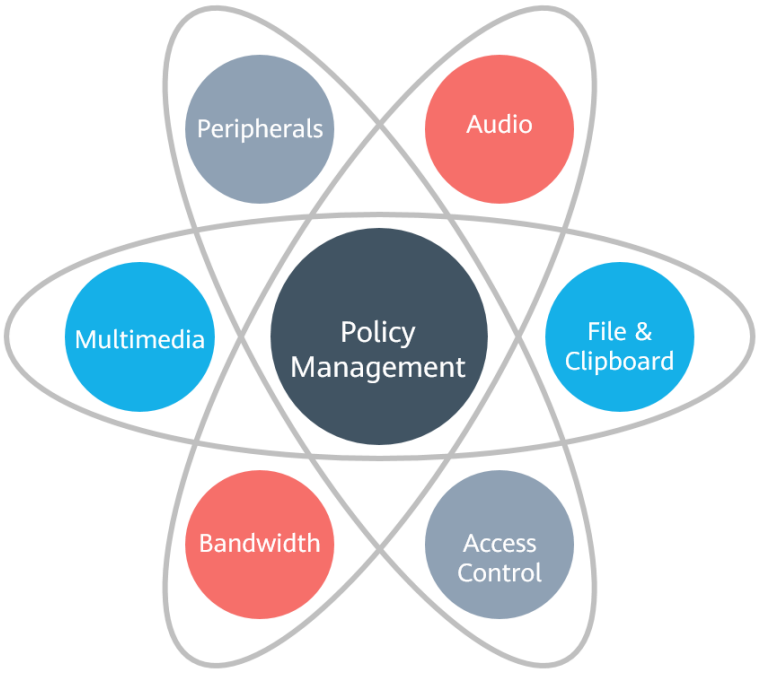


Figure 12-1 Policy management

Create application policies for all computers in a computer group, a computer, or computers of a user as needed in different scenarios.

You can create policies in terms of peripherals, audio, multimedia, client, display, file and clipboard, access control, session, bandwidth, virtual channel, watermark, keyboard and mouse, audio and video bypass, personalized data management, and customization.

12.1.2 Scenarios

Policy for Audio Scenarios

For daily work or conferences that do not allow audio recording and playback, disable **Audio Redirection**.

Set **Play Volume** only in education scenarios, such as online classrooms that need a set volume.

Policy for Display Scenarios

For desktop environments that require high definition, choose **Display** > **Display Policy Grade** to expand advanced settings and modify parameters such as **Bandwidth**, **Lossy Compression Recognition Threshold**, and **Lossy Compression Quality**.

Server decoding: playback of local and network videos. Multimedia redirection: local video playback. Flash redirection: network video playback.

12.1.3 Practices

For details about policy management practices, service adjustment, and alarm handling, see the lab guide.

12.2 Quiz

What are the precautions for adding a computer domain user?

13 Cloud Computing Trends

You have been previously introduced to how virtualization technologies integrate data center resources for better utilization. However, virtualized data centers also face challenges in unified scheduling and management of infrastructures, networks, and storage resources. What are the solutions to these challenges? What is the direction cloud computing is going?

In this section, you will learn the basic concepts and components of OpenStack and understand edge computing, blockchain, and cloud native concepts.

13.1 OpenStack Overview

13.1.1 Concepts

OpenStack is a popular open source cloud operating system (OS) framework. Since its first release in June 2010, OpenStack continues to be mature with the joint efforts of developers and users, and has been widely deployed in various fields, such as the private/public cloud and NFV. At the same time, OpenStack attracts and is supported by almost all mainstream IT vendors, and many startups that provide OpenStack-related products and services have emerged. OpenStack has become a mainstream standard in the open source cloud computing field.

Today, a prosperous and far-reaching OpenStack ecosystem has been established. OpenStack has become an unavoidable topic in the cloud computing era. Before understanding the development and core of cloud computing technologies, you must know OpenStack well. Therefore, this section describes some concepts of OpenStack.

The OpenStack official website defines OpenStack as a cloud OS. The analogy between PC OS and cloud OS can be used for better understanding.

OS is an important part of a computer system. It is a system that integrates various software and hardware of a computer system to process tasks and provide services for users. Some examples in daily life and work may be helpful for your understanding. Linux and Windows are common server/PC OSs, and Android and iOS are common mobile phone OSs. Essentially, these OSs provide five core functions: resource access and abstraction, resource allocation and scheduling, application lifecycle management, system management and maintenance, and man-machine interaction. In other words, an OS must be capable of providing these five functions before integrating software and hardware to provide services for users.

The five functions are described as follows:

1. Resource access and abstraction: Hardware devices, such as CPUs, memory, local hard disks, and NICs, are connected to a system and abstracted as logical resources that can be identified and managed by OSs.
2. Resource allocation and scheduling: OSs allocate hardware resources to system software or application software to enable efficient use of available resources.
3. Application lifecycle management: OSs help users to install, upgrade, start, stop, and uninstall application software.
4. System management and maintenance: OSs help system administrators to configure, monitor, and upgrade systems.
5. Man-machine interaction: OSs provide necessary man-machine interfaces for system administrators and users to perform operations on systems.

A complete cloud OS should also provide the preceding five functions. But the core difference is the managed object. A cloud OS manages a distributed cloud computing system consisting of a large quantity of software and hardware, while a common OS manages a server, a personal computer, or a mobile phone.

The five functions of a cloud OS are as follows:

1. Resource access and abstraction: Hardware resources, such as servers, storage devices, and network devices, are connected to a cloud computing system in virtualization or software-defined mode and abstracted into resource pools that can be identified and managed by the cloud OS.
2. Resource allocation and scheduling: Cloud OSs allocate hardware resources to tenants and their applications to enable efficient use of available resources.
3. Application lifecycle management: Cloud OSs help tenants to install, start, stop, and uninstall cloud applications.
4. System management and maintenance: Cloud OSs help system administrators to manage and maintain cloud systems.
5. Man-machine interaction: Cloud OSs provide necessary man-machine interfaces for system administrators and tenants to perform operations on cloud systems.

Although the cloud OS is much more complex than the common OS, their five key functions correspond to each other one by one.

OpenStack is a key component, or a framework, for building a complete cloud OS.

To build a complete cloud OS, a large number of software components need to be integrated so that they can work together to provide functions and services for system administrators and tenants. However, OpenStack cannot independently provide all capabilities of a complete cloud OS. Among the five functions mentioned above, OpenStack cannot independently access and abstract resources, and needs to work with underlying virtualization software, software-defined storage (SDS), and software-defined networking (SDN). OpenStack cannot independently provide comprehensive application lifecycle management capabilities, and needs to integrate various management software platforms at the upper layer. OpenStack does not have complete system management and maintenance capabilities, and needs to integrate various management software and maintenance tools when it is put into production. The man-machine interface provided by OpenStack is not powerful enough.

Therefore, to build a complete OpenStack-based cloud OS, OpenStack needs integrating with other software components that provide capabilities it does not have. So, indeed, OpenStack is a cloud OS framework. Based on this framework, different components can be integrated to build complete cloud OSs meeting different requirements.

Open source is an important attribute of OpenStack. If you do not know what open source is, you cannot truly understand the development history and future trend of OpenStack. A prominent feature of open source would be releasing source code on the Internet. However, the OpenStack community goes beyond that. In the OpenStack community, in the entire process of requirement proposal, scenario analysis, solution design, code submission, test execution, and code merge, each component, feature, and line of code are open to the public to ensure the supervision and participation of community contributors to the maximum extent. It is the effective supervision and full participation that prevent OpenStack from the absolute control of a few people, companies, or organizations and ensure the prosperity of the OpenStack community ecosystem. In addition, OpenStack complies with the most business-friendly Apache 2.0 license, which protects the business interests of enterprises in the community. This in turn fuels the business success of OpenStack products. In conclusion, OpenStack is a framework software product developed and released in open source mode for building cloud OSs in different scenarios. A deep understanding of this essence is of great significance for in-depth learning of OpenStack.

13.1.1.1 Design Ideas

In addition to the fast development of cloud computing technologies and the industry, OpenStack's unique design ideas also play a powerful role in promoting its rapid development. The design ideas of OpenStack can be summarized as openness, flexibility, and scalability. This section briefly introduces the three ideas.

- Openness

  The openness of OpenStack is rooted in its open source mode. As mentioned in the previous section, the open source concept of OpenStack is not only reflected in releasing source code, but also reflected in the entire process of design, development, test, and release. This open source mode prevents OpenStack from being controlled by individual users or enterprises and evolving to a closed architecture or system. In addition, the open northbound standard APIs and the free access to southbound software and hardware are made possible thanks to the openness of OpenStack. At the same time, OpenStack adheres to the concept of "not reinventing wheels" in the open source community. It continuously introduces and fully reuses excellent open source software in related technical fields to improve design and development efficiency and ensure software quality.

- Flexibility

  OpenStack is flexible because it uses the plug-in and configurable mode. OpenStack uses plug-ins to flexibly connect and manage different computing, storage, and network resources. It uses one architecture to implement resource pooling for different vendors and devices. For example, you can use plug-ins to access different Hypervisors (KVM, Xen, vCenter, and FusionCompute) for computing resource management, manage storage devices of different vendors and SDS (Ceph, FusionStorage, and vSAN), and access different network hardware devices, open source network components (OVS, Linux bridge, and HAProxy), and multiple SDN

controllers. The access mode using plug-ins can be configured. You can select different plug-ins by configurations to access different resources, so OpenStack does not need to be packaged and released again.

In addition, OpenStack is flexible because it is independent of any specific commercial software or hardware. In other words, any commercial software or hardware is optional and replaceable in OpenStack. In this way, users can use open source and open solutions to build OpenStack-based cloud computing systems without worrying about vendor lock-in.

- Scalability

  OpenStack has a highly scalable architecture. Specifically, its functions and system scale are scalable. OpenStack consists of multiple decoupled projects, which provide different services in a cloud computing system, such as identity authentication and authorization, computing, block storage, network, image, and object storage. For a cloud computing system in a specific scenario, system designers can use several OpenStack projects at first or introduce new OpenStack projects after system roll-out. Some OpenStack projects also have scalable functions. System designers can add new functions to these projects without affecting the existing functions. In terms of system scale, OpenStack is scalable because of its centerless and stateless architecture. Its main projects can be scaled out to meet the requirements of cloud computing systems of different scales. After a cloud computing system is constructed, the system scale can be gradually expanded by adding system management nodes and resource nodes based on the actual application loads. The architecture of OpenStack lowers initial construction cost, simplifies initial system planning, and provides plenty of expansion space for cloud computing system builders and operators.

13.1.2 Project Layering in the OpenStack Community

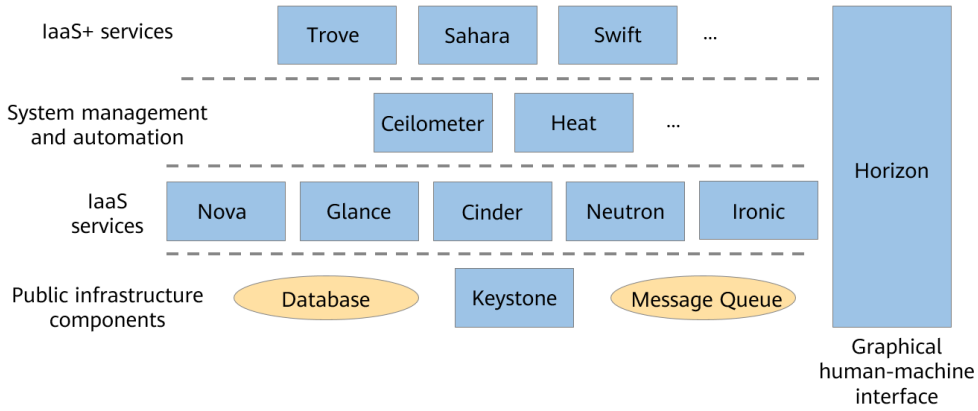


Figure 13-1 Project layering in the OpenStack community

When Austin, the first release of the OpenStack community, was released in 2010, OpenStack contained only two projects: Nova and Swift, which implemented only simple and basic functions. Today, OpenStack has become increasingly mature and powerful, and the number of OpenStack projects has increased significantly. For example, the release notes of Mitaka alone have 29 projects. Each project fulfills its own

responsibilities and cooperates with each other to form a cloud OS framework featuring flexible architecture, a wealth of functions, and high scalability.

To help you quickly and clearly understand OpenStack, this section briefly introduces some important and representative OpenStack projects.

- Keystone: identity authentication and authorization service

  A cloud OS should be capable of sharing computing, storage, and network resources and IaaS, PaaS, and SaaS services constructed based on these resources among different users while ensuring secure access and use of a same cloud computing system. A secure and reliable identity authentication and authorization service is the basis to provide this capability. Keystone is an OpenStack project providing identity authentication and authorization service. Keystone authenticates users and issues tokens to authorized users. Users can use the tokens to access other OpenStack projects and their services. The token verification and permission control mechanism of each component works with Keystone to identify users and control their permissions. This ensures proper resource operations of users and isolates and protects their resources.

- Nova: computing service

  A cloud OS should be capable of providing users with VMs of the given specifications. Nova is an OpenStack project providing such computing service. It manages a large number of physical servers with Hypervisors deployed in a unified manner to form a logical resource pool with a complete resource view. With the resource pool, Nova can manage the lifecycle of resources after receiving requests from different users. The lifecycle management mainly refers to creating, deleting, starting, and stopping VMs. After a user sends a VM creation request, Nova assembles resources such as CPUs, memory, local storage, and I/O devices in the logical resource pool into a VM of the given specifications, installs a proper OS on the VM, and provides the VM for the user.

  In addition, Nova works with Ironic to manage Bare Metal Server (BMS) resources. After receiving a BMS resource request sent by a user, Nova invokes the corresponding functions of Ironic to automatically select and allocate a BMS and install and deploy an OS on the BMS. In this way, users can use physical machine resources like using VM resources.

- Ironic: BMS management

  Ironic works with Nova to provide BMS capabilities for users.

  Ironic performs physical server management. When a physical server is added to a resource pool, Ironic records the hardware specifications of the physical server and reports the specifications to Nova. When a user sends a BMS management request, Ironic performs specific management operations on the corresponding physical server according to Nova's instructions. For example, when a user sends a BMS creation request, Ironic performs the corresponding operations, such as hardware initial configuration and OS installation, on the selected physical server based on the Nova scheduling result.

- Glance: image service

  Generally, after a VM is created, an OS needs to be installed on the VM. Therefore, a cloud computing system needs to preset OS images of different types and versions

for users to select. In some application scenarios, some common applications need to be pre-installed in images, which further increases the types and quantity of images. Therefore, a cloud OS must be capable of managing images. Glance is an OpenStack project providing image service.

Glance manages images in the system and their metadata. Image management includes creating, deleting, querying, uploading, and downloading images. However, in a normal production environment, Glance stores only the metadata of image files, but not image files. It is essentially a management frontend. To fully manage and store images, Glance needs to interconnect with a real object storage backend.

- Swift: object storage service

  Object storage service is a common data storage service in cloud computing. It is applicable when a single file with a large amount of data needs to be stored, data is less frequently accessed, requirements for data access latency are low, and data needs to be stored with low costs. Swift is an OpenStack project providing object storage service.

  Unlike most OpenStack projects that only have control functions and do not directly carry user services, Swift has a complete object storage system, so it can even be used as an object storage system independent of OpenStack.

  In the OpenStack system, Swift can also be used as the storage backend of Glance to store image files.

- Cinder: block storage service

  In a typical, KVM-based OpenStack deployment solution, VMs created by Nova use the local file system of each compute node to store data by default. For such a VM, the lifecycle of the data follows that of the VM. That is, when the VM is deleted, the data is deleted accordingly. If users want persistent block storage media whose lifecycle is independent of VMs, they can use Cinder, which provides block storage service (volume service).

  Cinder abstracts storage resources of different backend storage devices or SDS clusters into block storage resource pools, divides the pools into volumes of different sizes as required, and allocates the volumes to users.

  When using volumes provided by Cinder, users need to use Nova to attach the volumes to specified VMs. In this case, users can view and access the block device corresponding to the volume in the VM OS.

- Neutron: network service

  Network service is a key component of the IaaS layer capabilities of cloud OSs. Only with a stable, easy-to-use, and high-performance cloud virtual network can users connect various resources and services provided by cloud computing system to form an application system that meets their service requirements.

  Neutron is an OpenStack project providing network service. Neutron and the sub-projects derived form itself provide users with multiple network service functions from L2 to L7, including L2 networking, L3 networking, intranet DHCP management, Internet floating IP address management, intranet and extranet firewalls, load balancing, and VPN. In general, the L2 and L3 capabilities of Neutron are mature. Today, Neutron has replaced Nova Network and become the mainstream virtual network service implementation mode at L2 and L3 in OpenStack. But Neutron's L4

to L7 capabilities are still developing rapidly, and only some basic application capabilities are available.

It should be noted that the DNS as a service capability of OpenStack is not provided by Neutron, but by Designate.

- Heat: resource orchestration service

  One of the core benefits of cloud computing is the automatic management and use of IT resources and services. In other words, after the cloud computing technology is applied, numerous, complex management tasks that need manual operations in the traditional IT field can be automatically completed by invoking APIs provided by a cloud OS, thereby significantly improving IT system management efficiency.

  In the preceding complex management operations in the IT field, the lifecycle management operations of user service application systems, such as installation, configuration, capacity expansion, and deletion of application systems, are typical. Such operations are complex, time-consuming, and labor-consuming, and cannot meet the emerging requirements for quick service roll-out and elastic deployment. Heat is an OpenStack project providing automatic application system lifecycle management. Specifically, Heat parses the template that is submitted by users and describes the requirements of application systems on resource types, quantity, and connections. Based on the template, Heat invokes APIs of other projects (Nova, Cinder, and Neutron) to automatically deploy application systems. This process is highly automated and programmed. A template can be reused on the same or different OpenStack-based cloud computing systems, greatly improving the application system deployment efficiency. In addition, Heat can work with Aodh, a sub-project of OpenStack Ceilometer, to implement auto scaling of application systems. This further simplifies the management of some application systems that use the stateless and horizontally scalable architecture, and has typical cloud computing service features.

- Ceilometer: monitoring and metering

  In a cloud computing system, resources are provided as services for users. Users also need to pay fees based on the type and quantity of resources used. In this way, a cloud OS should be capable of monitoring and metering resource usage. This is the primary cause why OpenStack introduces Ceilometer.

  Ceilometer mainly collects information about the types and quantity of resources used by users in polling mode. The information is used as the basis for charging.

  In addition, Ceilometer can use the information to send alarms through Aodh to trigger Heat to execute the auto scaling function.

  However, Ceilometer does not support charging. The system designer needs to interconnect Ceilometer to an appropriate charging module to realize the complete charging function. Currently, the OpenStack community has created CloudKitty as its native charging component. However, CloudKitty is still in the early stage and cannot be put into commercial use.

- Horizon: graphical interface

  Horizon is a graphical man-machine interface in the OpenStack community. After long-term development and improvement in the community, Horizon has a user-friendly interface and provides rich easy-to-use functions that can meet the basic

requirements of cloud computing system administrators and common users, so Horizon can be used as the basic management interface of OpenStack-based cloud computing systems.

In addition, the Horizon architecture uses many plug-ins, so it is flexible and easy to expand and allows system designers to develop new functions on demand.

13.2 Emerging Technologies

13.2.1 Edge Computing

13.2.1.1 What Is Edge Computing?

Edge computing is a kind of distributed computing. Computing resources move downwards from the cloud to the edge. Data is processed and analyzed near the devices and data sources. Edge computing forms a distributed open platform that integrates networks, computing, storage, and applications. Edge computing provides services by the nearest edge to respond in real time, meeting basic requirements on intelligence, security, and privacy protection.

In Wikipedia, edge computing is defined as a distributed computing paradigm that moves the computation of applications, data and services from the central network node to the periphery nodes of the network. Large-scale services that are originally managed by the central node are split into smaller parts. These parts are distributed to edge nodes for easier management. An edge node refers to any node that has computing and network resources between a data source and a cloud center. For example, a mobile phone is an edge node between a person and a cloud center, and a gateway is an edge node between a smart home and a cloud center.

The edge computing architecture consists of three layers: device, edge, and cloud. Inter-layer and cross-layer communication are available.

The device layer consists of various devices. It collects and reports raw data and uses event sources as the input of application services. The edge computing layer consists of network edge nodes and is widely distributed between devices and the computing center. The computing and storage resources on each edge node vary greatly, and these resources change dynamically. The edge computing layer properly deploys and schedules the computing and storage capabilities of the edges to implement basic service response. Cloud computing center is still the most powerful data processing center. Data reported by the edge computing layer is permanently stored in the cloud computing center. Analysis tasks and comprehensive global information processing tasks that cannot be processed by the edge computing layer still need to be completed in the cloud computing center.

Amazon pioneered edge computing and has launched AWS Greengrass. Microsoft has released Azure IoT Edge, which moves cloud analytics to devices, supports offline use, and focuses on AI applications on the edge. Google has launched Edge TPU (a hardware chip) and Cloud IoT Edge (software stack) to extend data processing and machine learning capabilities to edge devices, enabling devices to process data from sensors in real time and predict results locally.

Alibaba has launched Link IoT Edge, which uses TSL to convert devices of various protocols and data formats into standard TSL models to provide secure, reliable, low-latency, cost-effective, scalable local computing services. Huawei has launched Intelligent EdgeFabric (IEF), which extends cloud applications to edge nodes and associates edge and cloud data to offer a complete edge computing solution that contains integrated services fueled by edge-cloud synergy. In the manufacturing industry, Haier and ROOTCLOUD have launched their own cloud-edge synergy platforms based on a wide range of industrial scenarios to help users quickly build industrial Internet applications and implement fast access of various industrial devices.

13.2.1.2 Concepts of Intelligent Edge

Traditional IoT is intelligent because of the data center, not edge devices.

Currently, the ideas of mainstream IoT technologies are as follows:

- Edge devices transmit the collected data to the data center.
- The data center performs data computation, processing, and analysis, and then delivers the operation instructions to edge devices.
- Edge devices execute the instructions to obtain the results required by users.

Therefore, sometimes such predicaments arise:

- Although all devices are located in the same area or even in the same building, they must communicate with the data center far away. This not only causes delay, but also reduces the availability of the entire system due to network or other causes.
- The collected raw data must be transmitted to the data center for analysis and processing. This requires higher network bandwidth and storage capacity and increases the workloads of the data center.
- The intelligence of the entire IoT depends on the data center. Edge devices can only collect and transmit data and execute instructions. Once the network communication between edge devices and the data center is interrupted, the entire IoT system may be unavailable.

Therefore, the preceding IoT technologies are still far from being truly intelligent. Intelligence does not mean that it is totally free from manual intervention.

As a result, intelligent edge computing emerges. It proposes a new mode that each edge device in the IoT is intelligent and capable of data collection, analysis, computing, and communication. Intelligent edge computing also uses cloud computing to configure security and deploy and manage edge devices at scale. It allocates intelligent resources based on device types and scenarios. In this way, intelligence can flow between the cloud and edge devices.

Let's take an IoT-based temperature monitoring system as an example:

- Edge sensors no longer need to continuously transmit temperature values to the data center. They can make decisions by themselves. They contact the data center only when the values change significantly and wait for feedback from the data center to determine what operations they should take.
- Edge devices can be more intelligent. For example, if the temperature changes abruptly, edge devices can directly determine what operations to perform based on

the applications running on the devices without contacting the data center. Even if the network is temporarily interrupted, the entire system can run normally.

In this mode, the computing and processing capabilities of edge devices can be used to process most IoT tasks nearby. This not only reduces the workloads of the data center, but also responds to different states of edge devices timely and accurately. Making edge devices truly intelligent is the charm of intelligent edge computing.

13.2.1.3 Features of Intelligent Edge

The features of intelligent edge can be summarized as data upload, capability deployment on the edge, intelligent localization, and edge hardware acceleration.
Data upload: Once processed and filtered, edge data is uploaded from the edge to the cloud.

Capability deployment on the edge: AI and other capabilities need offloading from the cloud to edge devices.

Intelligent localization: Intelligent edge is miniaturized, lightweight, and edge-prone, free from cloud and network constraints.

Edge hardware acceleration: Powerful hardware enables on-premises real-time inference.

In general, the intelligent edge needs more than a single technology. The framework and software stack of intelligent edge computing show features of different layers: the underlying hardware acceleration; the intelligent localization and lightweight technologies at the middle layer; edge-cloud synergy at the upper layer, such as capability deployment on the edge, anonymized data upload, and device management. Full-stack hardware and software and public/private/hybrid cloud computing are essential capabilities.

13.2.2 Blockchain

13.2.3 What Is Blockchain?

Blockchain, also known as distributed ledger technology, is a tamper-proof, non-repudiation accounting mechanism maintained by multiple parities. Used in conjunction with cryptography, it secures transmission and access, and ensures data storage consistency.

Blocks validated by orderers through cryptography then sequentially form a blockchain (or distributed ledger). Blockchain is a low-cost computing and collaboration paradigm used to build trust in an untrustworthy environment. It is vital for developing tomorrow's digital economy and trust system.

The blockchain technology will be more widely used in the coming years. Considered as a revolutionary and disruptive technology, it can upgrade the existing service progress with its excellent efficiency, reliability, and security. Blockchain technology empowers enterprises with the following advantages:

- Trust is built among parties by reliably sharing data.
- Decentralized, shared, and permit-required ledgers integrate data into a system to break down data silos.
- Data is highly secured.

- Lower dependency is required on third parties.
- Real-time, tamper-proof records are shared among participants.
- Authenticity and integrity of products in the business flow are ensured by participants.
- Products and services in the supply chain are tracked and traced seamlessly.

13.2.3.1 Blockchain Concepts

Blockchain is an innovation that combines multiple existing technologies.

The related technologies include:

- Distributed ledger: A database that is shared, replicated, and synchronized among network members. It records transactions (such as asset or data exchange) between these members without spending time and money for ledger reconciliation.
- Cryptography (or hash algorithm): The hash value of a digital content segment can be used to verify its integrity. Any modification to digital content significantly changes its hash value. A qualified hash algorithm easily obtains this value from digital content while preventing back-calculation of the original digital content from the value.
- Distributed consensus: A system's independent participants must achieve majority consensus on a transaction or operation. Examples include verifying double-spending transactions, validating transactions, and determining whether to write verified data to the existing ledger.
- Smart contract (or chaincode): This runs on a blockchain and is automatically triggered by specific conditions. It is an important way to implement service logic when using a blockchain. Due to the nature of blockchains, the execution results of contracts cannot be forged or tampered with, and their reliability is assured.

13.2.3.2 Blockchain System Architecture

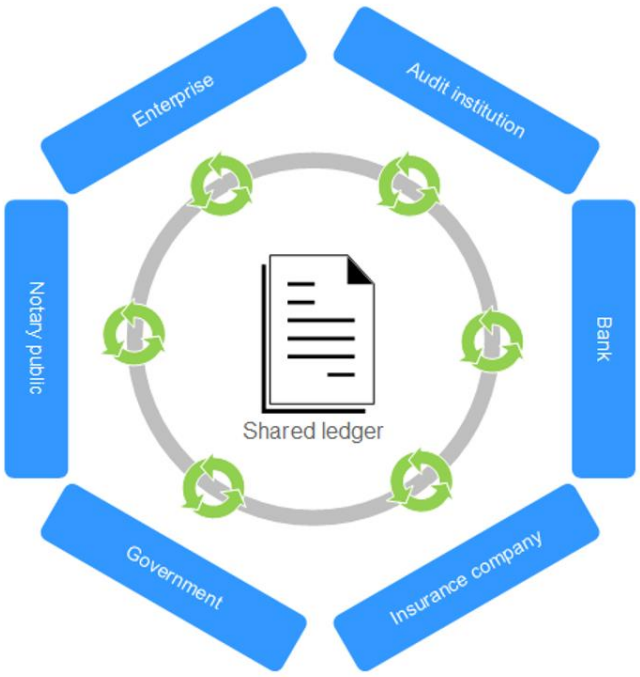


Figure 13-2 Blockchain system architecture

Ledgers are shared by participants in a business network and updated upon each transaction.

Cryptographic algorithms ensure transaction security by limiting participant access only to related ledger content.

Transaction-related contract clauses are embedded into the transaction database to form smart contracts. These clauses are automatically executed when an event meets clause conditions.

Consensus algorithms ensure that transactions are validated by all involved parties and meet supervision and audit requirements.

The application scenarios of blockchain include:

- Inter-company transactions: Blockchain technology ensures transaction consistency and accounting balance without the need for reconciliation. It supports transactions among different systems and provides traceable and immutable E2E information for internal and external audits.
- Supply chain logistics: In addition to transparent rules and automatic settlement, blockchain technology enables E2E tracking of goods all the way from production to final reception, improving the trust between consumers and partners in the supply chain. Electronic proofs of delivery (PODs) reduce the delay caused by paper works. Smart contracts enable automatic settlement to improve efficiency.
- Healthcare: The healthcare consortium blockchain connects information systems of healthcare institutions, so that regional inspection as well as ultrasound and radiological examination results can be securely exchanged for online healthcare, two-way referral, and remote consultation. Encryption and smart contract-based

authorization mechanisms offer patients access to their own healthcare data while protecting their privacy. Others can access the data only when authorized.

13.2.4 Cloud Native

13.2.4.1 Development of Cloud Technologies – Cloud Native

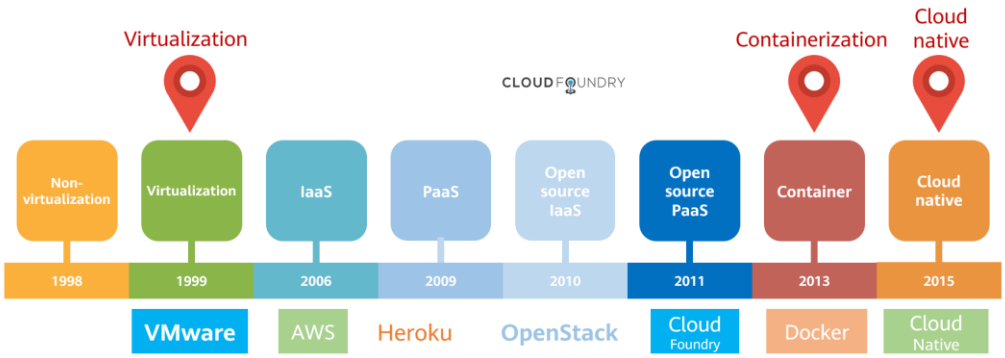


Figure 13-3 Development of cloud technologies – cloud native

The concept of cloud native was put forward by Matt Stine from Pivotal in 2013.

Matt summarized this concept based on his years of architecture design and consulting experience. It has been continuously improved by the community and gradually matures with the establishment of CNCF in 2015. Pivotal clearly defined cloud native in 2015, pointing out that cloud native is a way to build and run applications by making full use of cloud computing advantages.

13.2.4.2 Cloud Native Concepts

According to Cloud Native Computing Foundation (CNCF), cloud native is a software development method that features containers, service mesh, microservices, immutable infrastructure, and declarative APIs to build and run scalable applications in various cloud computing environments.

Cloud computing is the current popular choice for IT development and lays a foundation for cloud native. Cloud native is the next stage of cloud computing.

In terms of technical benefits, more and more enterprises will use cloud native in their businesses. In terms of software development, cloud native allows easier, faster service innovations.

According to Gartner, 70% of global enterprises will run at least three containerized applications in production by 2023.

Cloud native involves a large number of new PaaS technologies and development concepts. It is the shortest path to unleash the value of cloud computing and promotes the upgrade of cloud computing. CNCF is also working on the standardization of cloud native technologies while avoiding vendor lock-in. Cloud native is not only an upgrade of application architectures that use cloud, but also an upgrade of cloud platforms and cloud services.

- **Container**: Containerization is also called operating system level virtualization. This technology virtualizes the operating system kernel and allows user space software

instances to be divided into several independent units and run in the kernel instead of only one instance. Such a software instance can be called a container.

- **Microservice**: This software architecture style is based on small building blocks that focus on single responsibilities and functions. It combines complex, large-scale applications as modules. Functional blocks communicate with each other using language-independent/-agnostic APIs.
- **Service mesh**: A service mesh decouples communication between services from service processes, provides services programmatically, and decouples the data plane from the control plane.
- **Immutable infrastructure**: Once an instance of any infrastructure is created, it becomes read-only. If you need to modify or upgrade the instance, replace it with a new instance.
- **Declarative API**: Compared with imperative APIs, declarative APIs describe the desired state of the system to implement deployment and control.

13.2.4.3 Cloud Native Applications

Pivotal and Red Hat provide their interpretation on cloud native applications.

Pivotal:

Cloud-native applications are purpose built for the cloud model. These applications—built and deployed in a rapid cadence by small, dedicated feature teams to a platform that offers easy scale-out and hardware decoupling—offer organizations greater agility, resilience, and portability across clouds. For details, you can visit the Pivotal website at https://pivotal.io/de/cloud-native.

Red Hat:

Cloud-native applications are a collection of small, independent, and loosely coupled services. They are designed to deliver well-recognized business value, like the ability to rapidly incorporate user feedback for continuous improvement. In short, cloud-native app development is a way to speed up how you build new applications, optimize existing ones, and connect them all. Its goal is to deliver apps users want at the pace a business needs. For details, you can visit the Red Hat website at https://www.redhat.com/en/topics/cloud-native-apps.

13.2.4.4 Overall Understanding of Cloud Native Applications

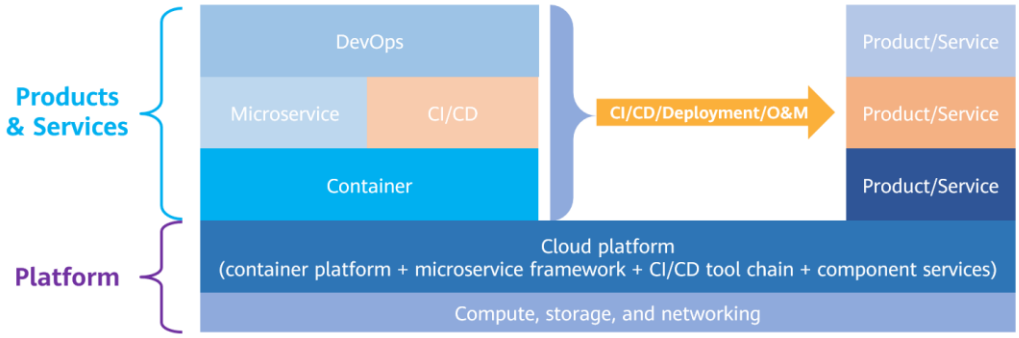


Figure 13-4 Overall understanding of cloud native applications

Cloud native applications are running on the cloud to exploit the advantages of the cloud model.

Applications are packaged, distributed, and deployed in containers. The microservice architecture is used by applications to make full use of cloud component services. The organization architecture and method of DevOps and the CI/CD tool chain are jointly used to implement continuous delivery of products and services.

In a word, cloud native is an approach and practice. Cloud native applications are the practice results of cloud native.

Platform: provides technical support on cloud native.

Products & Services: delivered and running on four key technologies and organizational formations of cloud native.

13.3 Quiz

What are the differences between edge computing and cloud computing?

14 Conclusion

This document focuses on the basics of cloud computing (server, storage, network, and OS), overview of FusionCompute, routine management and troubleshooting of virtualization resource pools, overview of FusionAccess and related components, as well as their installation, deployment, service management, and troubleshooting. Other mature technologies, such as containers and OpenStack, will be detailed in the latest HCIP and HCIE courses. You can also learn about emerging technologies such as fog computing, edge technologies, and serverless. We will update some useful documents on our official website to keep you abreast of the latest cloud developments.

Any suggestions, comments, and technical questions are welcome on the official HCIA-Cloud Computing forum:

https://forum.huawei.com/enterprise/en/index.html

Learn more at:

1. Huawei Certification: https://e.huawei.com/en/talent/#/cert
2. Huawei Talent Online: https://ilearningx.huawei.com/portal/subportal/EBG/51
3. Huawei ICT Academy: https://e.huawei.com/en/talent/#/ict-academy/home/v1?t=1561101910908
4. HUAWEI CLOUD Developer Institute: https://edu.huaweicloud.com/intl/en-us/