

复习要点

第一章

1.1 云计算概述

- 云计算的发展历程：云计算是分布式计算、并行计算、效用计算、虚拟化、网络存储、负载均衡、热备份冗余等传统计算机和网络技术发展融合的产物。
- 云计算指 IT 资源的交付和使用模式，通过网络以按需、以扩展的方式获得所需的资源。
- 提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的，并且可以随时获取、按需使用、随时扩展、按使用付费。

1.2 云计算的服务类型

- 云计算服务的提供方式：基础设施即服务（IaaS, Infrastructure as a Service）、平台即服务（PaaS, Platform as a Service）、软件及服务（SaaS, Software as a Service）。
- 各提供方式的定义、特点、核心技术及其代表产品。
- **IaaS 定义：**把计算、存储、网络以及搭建应用环境所需的一些工具当成服务提供给用户，使得用户能够按需获取 IT 基础设施。
IaaS 特点：（1）把 IT 资源以服务的方式提交给用户；（2）基础设施可以动态扩展，即可以根据应用的需求动态增加或减少资源；（3）计费服务灵活多变，按实际使用的资源进行计费；（4）多租户，相同的基础设施资源可以同时提供给多个用户共享使用；（5）企业级的基础设施，不仅可以为个人用户提供 IT 资源，而且可以满足中小企业的 IT 资源需求，使得它们可以从聚焦的计算资源池中获利。
IaaS 核心技术：（1）虚拟化技术；（2）分布式存储技术；（3）高速网络技术；（4）超大规模资源管理技术；（5）云服务计费技术。
IaaS 服务的优势：（1）低成本；（2）免维护；（3）伸缩性强；（4）支持应用广泛；（6）灵活迁移
IaaS 代表产品：亚马逊 Amazon 的 EC2AWS，OpenStack
- **PaaS 定义：**是一种分布式平台服务，为用户提供一个包括应用设计、应用开发、应用测试及应用托管的完整计算机平台。（主要用户是开发人员）
PaaS 核心技术：（1）REST 技术；（2）多租户技术；（3）并行计算技术；（4）应用服务器；（5）分布式缓存
PaaS 平台的优势：（1）友好的开发环境；（2）丰富的服务；（3）精细的管理和控制；（4）弹性强；（5）多租户机制；（6）整合率高。
PaaS 代表产品：Force.com, Google App Engine, Windows Azure, Cloud Foundry.
- **SaaS 定义：**是软件服务提供商为了满足用户的需求提供的软件的计算能力。
SaaS 的优势：（1）使用简单（通过浏览器访问）；（2）支持公开协议；（3）成本低；（4）安全保障；

SaaS 的关键技术：(1) 大规模多租户支持；(2) 认证和安全；(3) 定价和计费；(4) 服务整合；(5) 开发和定制。

SaaS 代表产品：Microsoft Bing 在线 Office

1.3 云计算的部署模式

- 云计算的部署模式：公有云、私有云、混合云、社区云。
- 各部署模式的定义及优缺点。
- 公有云定义：是一种对公众开放的云服务，由云服务提供商运营，为最终用户提供各种 IT 资源，可以支持大量用户的并发请求。
公有云优点：所应用的程序及相关数据都存放在公有云的平台，用户无需前期的大量投资和漫长的建设过程；具有规模的优势；其运营成本比较低；只需为其所使用的付费，可节省使用成本。
公有云缺点：数据安全和隐私问题。
- 私有云定义：是指组织机构建设的专供自己使用的云平台。
私有云优点：可以支持动态灵活的基础设施，降低 IT 架构的复杂度，降低企业 IT 运营成本。
私有云缺点：企业需要大量的前期投资，需要采用传统的商业模型；私有云的规模相对于公有云来说一般要小得多，无法充分发挥规模效应。
- 混合云定义：由私有云及外部云提供商构建的混合云计算模式。使用混合云计算模式，机构可以在公有云上运行非核心应用程序，而在私有云上支持其核心程序以及内部敏感数据。
混合云优点：可以发挥出所混合的多种云计算模式各自的优势；企业可以在私有云的私密性和公有云的低廉之间做一定的权衡。
混合云缺点：混合云的部署方式对提供者的要求较高。
- 社区云定义：社区云服务的用户是一个特定范围的群体，它既不是一个单位内部的，也不是一个完全公开的服务，而是介于两者之间，所产生的成本由他们共同承担。

第二章

- 三类云计算服务的主要任务。
IaaS 的主要任务是把部署在数据中心的基础设施硬件资源通过 Web 提供给用户使用，通常包括虚拟机、网络资源和存储资源。通过使用 IaaS 服务，用户不再需要购买服务器、存储设备等硬件，而可以租用云计算服务提供的基础设施。
PaaS 的主要任务是把部署在数据中心的开发环境等平台作为一种服务提供给用户使用，通常包括操作系统、编程语言的运行环境、数据库、Web 服务器等。通过 PaaS 服务，软件开发人员可以在不购买服务器和平台软件的情况下开发和部署新的应用程序。
SaaS 主要是为用户提供被称为按需支付费用的应用软件。用户不必再去操心各种应用程序的安装、设置和运行维护，一切都由 SaaS 服务提供商来完成。总之，IaaS、PaaS 和 SaaS 都是以服务的方式为用户提供计算资源。IaaS 是为

用户提供计算所需的硬件资源，PaaS 为应用开发者提供开发应用所需的开发、测试和部署环境，而 SaaS 为最终用户提供各类应用软件。

第三章

3.1 概述

- IaaS 的根本目的

IaaS 是指将 IT 基础设施能力通过互联网提供给用户使用，并根据用户对资源的实际使用量或占用量进行计费的一种服务。IaaS 层的主要功能是使经过虚拟化后的计算资源、存储资源和网络资源能够以基础设施即服务的方式通过网络被用户使用和管理。

IaaS 的根本目的在于对计算资源的池化，并提供统一的、智能化的管理和调度。（计算资源的池化就是把大量的物理资源连接在一起形成一个巨大的资源库，并且能够按照较小的单元进行统一分配和管理）

3.4 服务器虚拟化技术

- 服务器虚拟化的定义

服务器虚拟化是指能够在一台物理服务器上运行多台虚拟服务器的技术，并且上述虚拟服务器在用户、应用软件甚至操作系统看来，几乎与物理服务器没有区别，用户可以在虚拟服务器上灵活地安装任何软件。同时服务器虚拟化技术还应该确保上述多个虚拟服务器之间的数据是隔离的，虚拟服务器对资源的使用是可控的。

- IaaS 具体管理的物理资源可以分为 3 类：计算资源（CPU、内存）、存储资源、网络资源。

- 服务器虚拟化的实现方式：寄宿虚拟化（性能较低，运行在宿主操作系统之上的应用程序）和原生虚拟化（性能较高，直接运行在硬件之上的不是宿主操作系统，而是虚拟化平台）。

- 服务虚拟化的核心技术及各核心技术的实现方法。

- （1）CPU 虚拟化：把物理 CPU 抽象成虚拟 CPU，任意时刻一个物理 CPU 只能运行一个虚拟 CPU 指令。每个客户操作系统可以使用一个或多个虚拟 CPU，虚拟 CPU 的运行相互隔离，互不影响。

实现原理：全虚拟化（二进制转换）和半虚拟化（超级调用）

- （2）内存虚拟化：内存虚拟化技术报物理机的真实物理内存统一管理，包装成多个虚拟的物理内存分别供若干个虚拟机使用，使得每个虚拟机拥有各自独立的内存空间。在服务器虚拟化技术中，因为内存是虚拟机最频繁访问的设备，因此内存虚拟化与 CPU 虚拟化具有同等重要的地位。

实现方法：影子页表法和页表写入法

- （3）设备与 I/O 虚拟化：设备与 I/O 虚拟化技术对物理机的真实设备进行统一管理，包装成多个虚拟设备给若干个虚拟机使用，响应每个虚拟机的设备访问请求和 I/O 请求。

实现方式：目前，主流的设备与 I/O 虚拟化都是通过软件的方式实现的。

- (4) 网络虚拟化：网络虚拟化是将多个硬件或软件网络资源及相关的网络功能集成到一个可用软件中统一管控的过程，对于网络应用而言，该网络环境的实现方式是透明的。该网络成为虚拟网络，形成该虚拟网络的过程称为网络虚拟化。

实现方式：一个网卡，每个虚拟机独立的网络配置，多个虚拟机通过这一物理网卡进行网络连接。

- (5) 实时迁移技术：在虚拟机运行过程中，将整个虚拟机的运行状态完整、快速地从原来所在的宿主机硬件平台迁移到新的宿主机硬件平台上，并且整个迁移过程是平滑的，用户几乎不会察觉到任何差异。

实现方式：需要虚拟机监视器的协助，即通过源主机和目标主机上虚拟机监视器的相互配合，来完成客户操作系统内存和其他状态信息的拷贝。

3.5 OpenStack

- OpenStack 的核心项目及其功能

OpenStack 是一个开源的云计算管理平台项目，旨在为公共及私有云的建设与管理提供软件。

6 大核心项目：

- (1) 计算：Nova（新星）
- (2) 块存储：Cinder（烬）
- (3) 网络：neutron（中子）
- (4) 镜像服务：Glance（瞰）
- (5) 对象存储：Swift（速）
- (6) 身份服务：Keystone（基石）
- (7) UI 界面：Horizon（地平线）

Nova：主要功能包括运行虚拟机实例，管理网络以及通过用户和项目来控制对云端的访问。

Cinder：实现存储服务，根据实际需要快速地为虚拟机提供块存储设备的创建、挂载、回收以及快照备份控制等。

Neutron：为 OpenStack 提供更加灵活的划分网络的能力，能够在多租户的环境下提供给每个租户独立的网络环境。

Glance：是一套虚拟机镜像查找及检索系统，支持多种虚拟机镜像格式，有创建镜像、上传镜像、删除镜像、编辑镜像的功能。

Swift：提供弹性可伸缩、高可用的分布式对象存储服务，适合存储大规模非结构化的数据。

Keystone：为 OpenStack 其他服务提供身份验证、服务规则和服务令牌的功能，管理 Domains, Projects, Users, Groups, Roles.

第四章

- PaaS 的定义

PaaS 是一种分布式平台服务，为用户提供一个包括应用设计、应用开发、应用测试及应用托管的完整计算机平台。

- PaaS 平台——Hadoop: HDFS 的结构、节点功能及其通信协议

Hadoop 可以为大数据应用提供一个可编程的、经济的、可伸缩的平台。这个分布式系统由分布式文件存储系统（HDFS, Hadoop Distributed File System）及计算框架（MapReduce）组成。Hadoop 是一个开源项目，可以为大量数据集提供批量数据处理能力。

HDFS 是一个可以存储极大数据集的分布式文件系统，它是通过向外扩展方式构建的主机集群。

HDFS 是一种主/从模式的系统结构。

主服务器，即图中的命名节点(NameNode)，管理文件系统命名空间和客户端访问，具体文件系统命名空间操作包括“打开”，“关闭”，“重命名”等，并负责数据块到数据节点之间的映射；此外，存在一组数据节点(DataNode)，它除了负责管理挂载在节点上的存储设备，还负责响应客户端的读写请求。从内部构造看，每个文件被分成一个或多个数据块，从而这些数据块被存放于一组数据节点上；数据节点会根据命名节点的指示执行数据块创建、删除和复制操作。

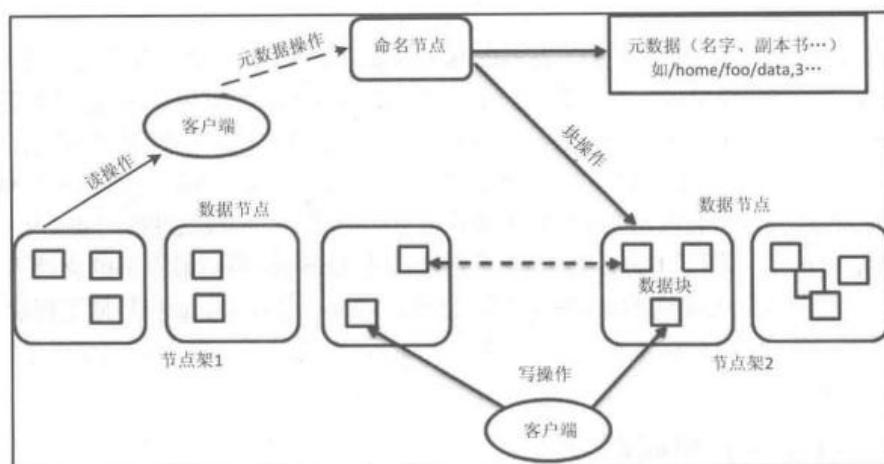


图 4.12 HDFS 架构

- MapReduce 的实现原理

MapReduce 建立在数学和计算机科学的众多基础工作之上，尤其是适用于每个数据元素的数据操作的描述方法。map 和 reduce 函数的概念直接来自于函数式编程语言。另一个关键的基本概念是“分而治之”，这个概念的基本原则是将单个问题分解成多个独立运行的子任务。如果多个子任务能够并行执行，这个方式更为强大。

MapReduce 是一个基于上述原理的处理模式，它实现了从源数据集到结果数据集的一系列转换。在最简单的情况下，作业的输入数据作为 map 函数的输入，所得到的临时数据作为 reduce 函数的输入。开发人员只需要定义数据转换形式，Hadoop 的 MapReduce 作业负责并行地对集群中的数据实施所需的转换。（把文

件分成多个数据片,然后把每个数据片交给一个 Map 函数处理,最后将每个 Map 函数统计的结果交给 Reduce 函数进行合并。如果有足够多的 Map 函数并行进行处理,那么就可以大大的提高统计的速度。)

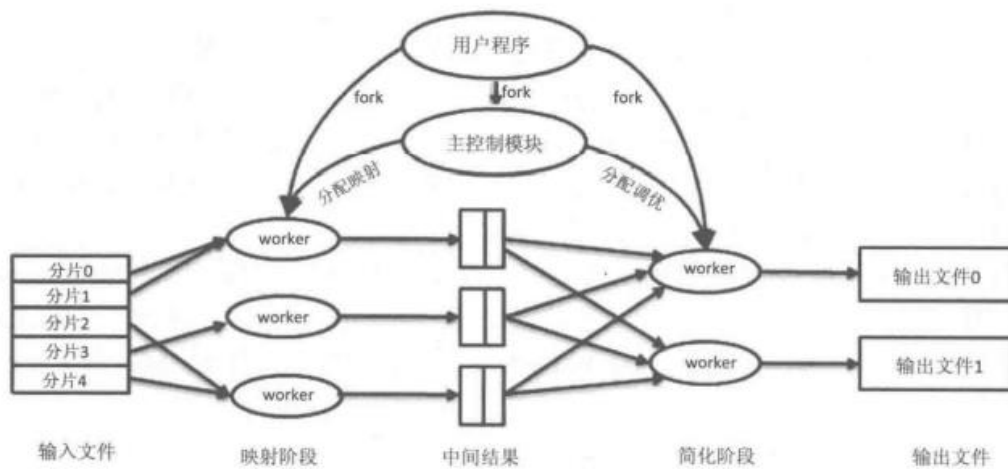


图 4.13 MapReduce 计算框架



MapReduce 计算框架

作为一个大规模数据存储平台，HDFS 并非必须与 MapReduce 配套使用。将 HDFS 和 MapReduce 组合使用是最为常见的情况。

- Hadoop MapReduce 的设计目标：(1) 易于编程；(2) 良好的扩展性；(3) 高容错性
- Hadoop 是一个批量处理系统，由于采用了大的数据集群，Hadoop 会相对快速地生成结果，但事实上，结果生成的速度还不足以满足缺乏耐心的用户的需求。因此，单独的 Hadoop 不适用于低时延查询，例如网站、实时系统或者类似查询。(是大数据框架而不是实时处理框架)

第五章

5.1 概述

- SaaS 的定义及其基本功能

SaaS 的定义：是一种新的软件交付模式，用户不需要购买和安装软件到本地的计算机，而是通过互联网远程租用运行在云数据中心的软件。

SaaS 的基本功能：为用户提供尽可能丰富的应用，为企业和机构用户化简 IT 流程，为个人用户提高日常生活方方面面的效率。

- SaaS 的特征

- (1) 多租户特征：**SaaS** 通常基于一套标准软件系统为成百上千的不同租户提供服务；
- (2) 互联网特性：**SaaS** 服务通过互联网为用户提供服务；
- (3) 服务特性：考虑服务合约的签定、服务使用的计量、在线服务质量的保证等问题；
- (4) 按需付费：用户可以根据需求按需订购软件应用服务；
- (5) 成本低：客户只要付出个人计算机和互联网服务所需的费用；
- (6) 开放性：平台提供应用功能的集成、数据接口的集成、组件的集成。

5.2 支撑平台

- SaaS 支撑平台的四大类型及各自的特点

提供 SaaS 服务的应用架构由应用类型、服务用户的数量、对资源的消耗等因素决定。

四种类型由是否支持可定制、可扩展和多租户三个方面的不同组合而决定。

- (1) **定制开发**：是一种最简单的提供 SaaS 服务的类型。在这种模型下，SaaS 提供商为每个客户定制一套软件，并为其部署。每个客户使用一个独立的数据库实例和应用服务器实例。数据库中的数据结构和应用的代码可能都根据客户需求做过定制化修改。这种架构适用于快速开发的小众应用，而在开发过程中没有过多考虑可定制、可扩展等因素。
- (2) **可配置**：通过不同的配置满足不同客户的需求，而不需要为每个客户进行特定定制，以降低定制开发的成本。为了增强应用的可定制性，以实现应用功能的共享，可以将应用中可配置的点抽取出来，通过配置文件或者接口的方式开发出来。
- (3) **多租户架构**：通过运行一个应用实例，为不同租户提供服务，并且通过可配置的元数据，为不同用租户提供不同的功能和用户体验。这才是真正意义上的 SaaS 架构，它可以有效降低 SaaS 应用的硬件及运行维护成本，最大化地发挥出 SaaS 应用的规模效应。
- (4) **可伸缩的多租户架构**：通过多个运行实例来分担大量用户的访问，从而可以让应用实现近似无限的水平扩展。也就是说，SaaS 应用的运行实例运行时所使用的下层资源与当前的工作负载相适应，运行实例的规模随工作负载的变化的同态伸缩。

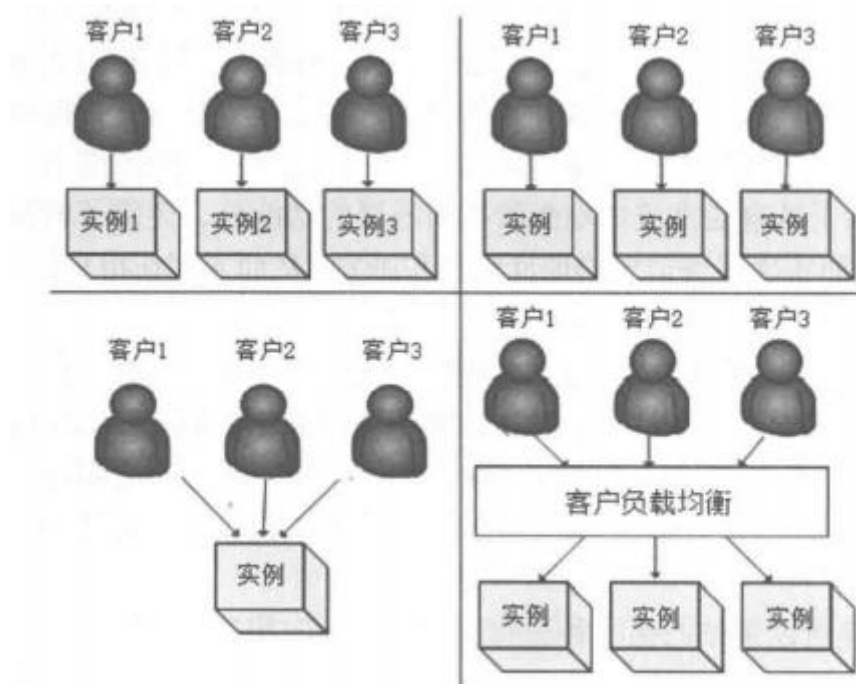


图 5.2 SaaS 层的架构类型

5.3 SaaS 应用

● 云应用的定义

SaaS 应用是运行在云端应用的集合。每一个应用都对应于一个业务需求，实现一组特定的业务逻辑，并且通过服务接口与用户交互。

- (1) 标准应用：面向大众的、采用多租户技术为数量众多的用户提供相互隔离的操作空间，提供的服务是标准的、一致的。
- (2) 客户应用：为了某个领域的客户而专门开发的，该类应用开发好标准的功能模块，允许用户进行不限于界面的深度定制。
- (3) 多元应用：一般由独立软件开发商或者是开发团队在公有云平台上搭建，满足用户某一类特定需求的创新型应用。

第六章

6.1 概述

- **虚拟桌面**是桌面云的核心技术，它可以为用户提供部署在云端的远程计算机桌面服务，即通过在云计算平台服务器上运行用户所需的操作系统应用软件，采用桌面交付协议将操作系统桌面视图以图像的方式传动到用户端设备上显示。
- 桌面云以虚拟桌面技术为基础，为用户提供访问灵活、数据安全、管理便捷的远程桌面。
- 桌面云基于 IaaS，通过桌面管理和服务模块，为用户提供良好的桌面服务，

使用户可以通过 PC，笔记本电脑，平板电脑甚至手机在任何有网络接入的地方访问自己的桌面，包括用户的应用软件，配置和数据。

6.2 业务价值和缺点

- 桌面云的业务价值（优势）

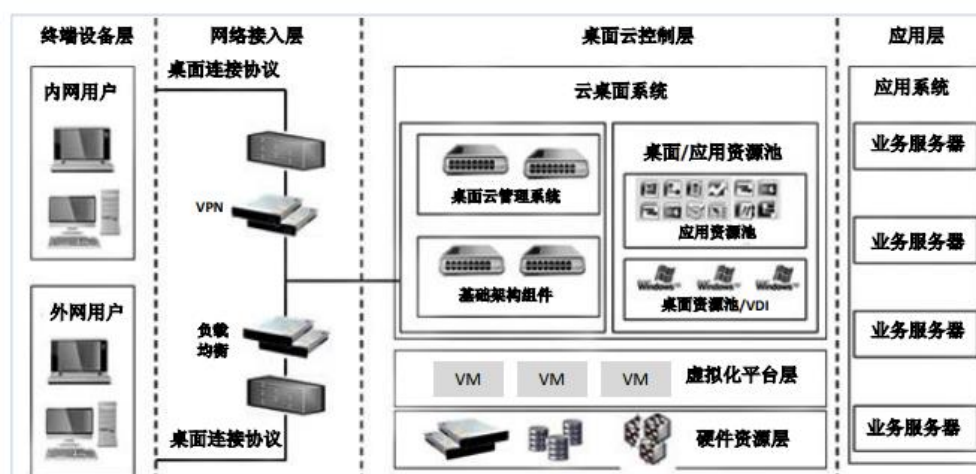
- （1）集中化管理
- （2）安全性高
- （3）绿色环保
- （4）总拥有成本减少

6.4 桌面云架构

- 桌面云架构的 6 个部分及功能

桌面云不能简单地理解为一个产品，而应该是一种基础设施，其组成架构较为复杂，通常可以分为终端设备层、网络接入层、桌面云控制层、虚拟化平台层、硬件资源层和应用层 6 个部分。

- （1）终端设备层：虚拟桌面终端主要负责给用户显示虚拟桌面视图，并通过外设接收用户侧的输入，再将其发送到服务侧。
- （2）网络接入层：将远程桌面输出到显示器，以及将键盘、鼠标以及语音等输入传递到虚拟桌面。
- （3）桌面云控制层：负责整个桌面云系统的调度，例如新虚拟桌面的注册以及将虚拟桌面的请求指向可用的系统。
- （4）虚拟化平台层：虚拟化平台层是云计算平台的核心，也是虚拟桌面的核心，承担着虚拟桌面“主机”功能，可以实现动态的硬件资源分配和回收。
- （5）硬件平台（资源）层：由多台服务器、存储和网络设备组成，为了保证桌面云系统正常工作，硬件基础设施组件应该同时满足三个要求：高性能、大规模、低开销。
- （6）应用层：主要用于向虚拟桌面部署和发布各类用户所需的软件应用，从而节约系统资源，提高应用灵活性。



6.5 虚拟桌面架构（VDI）技术

- VDI（Virtual Desktop Infrastructure）架构的基础是**服务器虚拟化**。
基于 VDI 架构的虚拟桌面解决方案的原理就是在服务器侧为每个用户准备专用的虚拟机并在其中部署用户所需的操作系统和各种应用，然后通过桌面显示协议将完整的虚拟机桌面交付给远程用户使用。

6.6 虚拟桌面交付协议

- 几种主流桌面连接协议：RDP 协议、ICA/HDX 协议、PCoIP 协议、SPICE 协议
RDP 协议（Remote Display Protocol）是 Microsoft 虚拟桌面产品采用的交付协议，是在国际电信联盟 ITU-T T.120 协议族的基础上进行的扩展，通过建立多个独立的虚拟通道，承载不同的数据传输和设备通信。（GPU 虚拟化是 RemoteFX 增强技术的核心）

ICA/HDX 协议：ICA（Independent Computing Architecture）是最老牌的虚拟桌面交付协议之一；HDX（High Definition eXperience）技术是对 ICA 协议的改进和增强，提供更好的高清使用体验。

PCoIP 协议：Teradici&VMware 在现有的标准 IP 网络的基础上研发的以显示压缩方式连接远程虚拟桌面的协议。PCoIP 的最大特点就是：将用户的会话以图像的方式进行压缩传输，对于用户的操作，只传输变化部分，保证在低带宽下也能高效的使用。

SPICE（Simple Protocol for Independent Computing Environment）协议最早由 Qumranet 开发，是一个具有三层架构的协议：QXL 驱动，SPICE 客户端，QXL 设备。

6.7 应用发布（区分是第一代还是第二代的重要标志）

- 应用发布的两大关键技术：**应用流(Application Streaming)**和**应用虚拟化(Application Virtualization)**。

应用软件发布是确保用户桌面能够被充分个性化的重要手段，其关键在于如何根据实际需要及时、有效地在操作系统桌面上发布应用软件。

- 两大关键技术的原理

应用流技术是一个集中的按需软件传送模式，其主要功能就是将应用程序及其运行环境打包成不需要安装即可运行的单一可执行程序，实现瘦客户端和应用程序的快速部署及管理，从而降低应用程序交付的成本或复杂性。（一个致命缺点是并非所有的应用程序都可以流化）

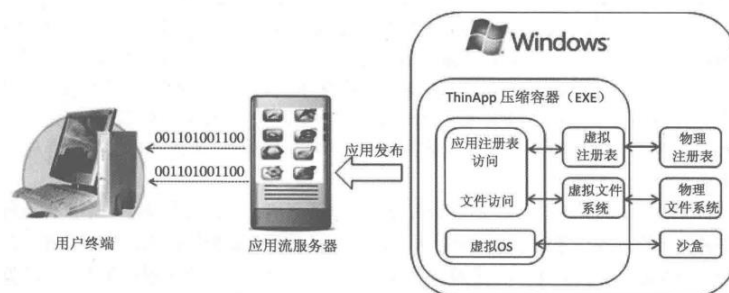


图 6.15 应用化技术工作原理

应用**虚拟化技术**提供了一种使应用无须在本地计算机进行安装就可被使用的能力，并且可以为用户提供有着与本地应用相近的用户体验。
应用虚拟化的原理是基于应用/服务器计算架构，采用类似虚拟终端技术，把应用程序的人机交互逻辑与计算逻辑隔离开来。

第七章

7.0 存储知识基础

- HDD(Hard Disk Drive, 硬盘驱动器)由介质、磁头、接口；SSD(Solid State Drives, 固态硬盘)有主控、内存、缓存、固件四大部分组成。SSD 容量较小，价格贵，随机存取极快，数据恢复较难；而 HDD 容量较大，价格便宜，随机存取速度一般，数据恢复较为容易。
- 常见接口协议：SCSI(小型计算机系统接口, Small Computer System Interface)，是在存储产品中应用最广泛的主机接口类型；FC(Fiber Channel, 光纤通道)；以太网口；iSCSI(互联网小型计算机系统接口)是一种在 TCP/IP 上进行数据块传输的标准。
- **双机热备**：主服务器运行应用，备份服务器处于空闲状态，但实时监测服务器的运行状态，一旦主服务器出现异常或故障，备份服务器立即接管主服务器的应用(active/standby 方式)。
双机互备：没有主服务器和备份服务器之分，两台主机互为备份。主机各自运行不同的应用，同时还相互监测对方状况。当任一台主机宕机时，另一台主机立即接管它的应用，以保证业务的不间断运行(active/active)。
集群技术(cluster)：就是一个域内包含多台拥有共享存储空间的服务，各服务器通过内部局域网相互通信，群集内的任一服务器上运行的业务均可被所有的客户所使用。当一台服务器发生故障时，它所运行的应用将由其他服务器自动接管，这就实现了负载均衡和互为备份。

7.1 概述

- 云存储的核心是**应用软件**与**存储设备**相结合，通过应用软件来实现**存储设备**向**存储服务**的转变，是一个以**数据存储和管理**为核心的云计算系统。
- 用户使用云存储，并不是使用某一个存储设备，而是使用**整个云存储系统**带来的一种数据访问服务。
- 云存储的特征
 - (1) 易管理：少量管理员可以处理上千节点和 PB 级存储，更高效地支撑大量上层应用对存储资源的快速部署需求；
 - (2) 高可扩展性：云存储系统支持海量数据处理，资源可以实现按需扩展；
 - (3) 低成本：云存储系统应具有高性价比的特点，低成本体现在两个方面：更低的建设成本和更低的运维成本。
 - (4) 多租户：云存储架构的一个关键特征是多租户。
 - (5) 无接入限制：云存储与传统存储之间最显著的差异之一是其访问方法，相比于传统存储，云存储强调对用户存储的灵活支持，服务域内的存储资源可以随处接入，随时访问。

7.2 结构模型-7.3 云存储架构

- 云存储的结构模型自上而下分为访问层、应用接口层、基础管理层、存储层。
存储层：云存储最基础的部分；
基础管理层：云存储最核心的部分，也是云存储中最难以实现的部分；
应用接口层：不同的云存储运营单位可以根据实际业务类型，开发不同的应用程序应用服务接口，提供不同的应用服务；
访问层：任何一个授权用户都可以通过标准的公用应用接口来登录云存储系统，享受云存储服务。



- 云存储架构分为两大类：**紧耦合对称架构**和**松耦合非对称架构**。
传统的存储系统利用**紧耦合对称架构**，这种架构的设计旨在解决**HPC(高性能计算, 超级运算)**问题，现在其正在向外扩展成为云存储，从而满足快速呈现的市场需求，新的存储系统已经采用了**松耦合非对称架构**，集中元数据和控制操作，这种架构并不非常适合高性能**HPC**，但是这种设计旨在解决云部署的**大容量**存储需求。
紧耦合对称架构（Tight Coupling Symmetrical, TCS）是为了解决单一文件性能所面临的挑战，这种挑战限制了传统NAS系统的发展。为了解决这一问题，业内创建了TCS架构产品，很多节点同时伴随着分布式锁管理和缓存一致性功能，这种解决方案对于单文件吞吐量问题很有效，已经在很多HPC客户中得到了广泛应用。
松耦合非对称架构（Loose Coupling Asymmetrical, LCA）利用一个数据路径之外的中央元数据控制服务器来向外扩展。

7.4 云存储类型及其适合的应用

- 云存储分为3种类型：块存储（Block Storage）、文件存储（File Storage）、对象存储（Object Storage）。
- **块存储**（Block Storage）：把单笔的数据写到不同的硬盘，借以得到较大的单笔读写带宽。

优点是单笔数据读写很快；
缺点是成本较高，并且无法解决真正海量文件的存储。

DAS 和 SAN 两种块存储类型

块存储应用于：

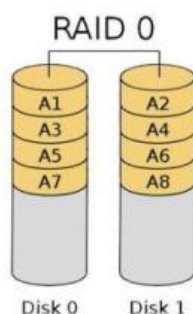
- （1）快速更改的单一文件系统；（2）针对单一文件大量读写的高性能计算
- **文件存储（File Storage）**：是基于文件级别的存储，它是把一个文件放在一个硬盘上，即使文件太大拆分时，也放在同一个硬盘上。
优点是对一个多文件、多人使用的系统，总带宽可以随着存储节点的增加而扩展，它的架构可以无限制地扩容，并且成本低廉。
缺点是对单一文件的读写会受到单一硬盘效能的限制。
适合的应用场景：
 - （1） 文件较大，总读取带宽要求较高；
 - （2） 多个文件同时写入；
 - （3） 长时间存放的文件；
- **对象存储（Object Storage）**：多台服务器内置大容量硬盘，再装上对象存储软件，然后在额外增加几台服务作为管理节点，安装上对象存储管理软件。管理软件可以管理其他服务器对外提供读写访问功能。
对象存储是为了克服块存储与文件存储各自的缺点，发扬它们各自的优点。简单而言块存储读写快，不利于共享；文件存储读写慢，利于共享。
- 存储直接裸盘映射的应用—数据库：适合使用块存储

7.5 关键技术

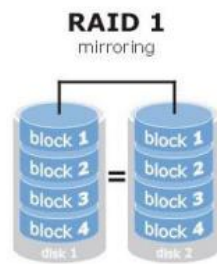
- RAID0、RAID1、RAID5 的工作模式

RAID 为廉价磁盘冗余阵列（Redundant Array of Inexpensive Disks）：RAID 技术将一个个单独的磁盘以不同的组合方式形成一个逻辑硬盘，从而提高了磁盘读取的性能和数据的安全性。不同的组合方式用 RAID 的级别来标识。

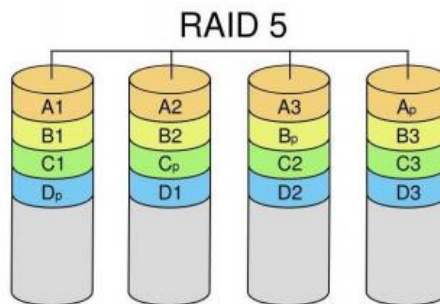
RAID0：也称为条带化(stripe)，将数据分成一定大小顺序读写到阵列的磁盘里，RAID0 可以并行的执行读写操作，可以充分利用总线的带宽，理论上讲，一个由 N 个磁盘组成的 RAID0 系统，它的读写性能将是单个磁盘读取性能的 N 倍，且磁盘存储效率最大(100%)。缺点：不提供数据冗余保护，一旦数据损坏将无法恢复。RAID0 应用于对读取性能要求较高但所存储的数据为非重要数据的情况下使用。



RAID1 称为镜像(mirror)，它将数据完全一致的分别写到工作磁盘和镜像磁盘，因此它的磁盘空间利用率为 50%，在数据写入时时间会有影响，但是读的时候没有任何影响。RAID1 应用于对数据保护极为重视的应用。



RAID5 将数据校验的信息均匀分散到阵列的各个磁盘上，这样就不存在并发写操作时的校验盘性能瓶颈。阵列的磁盘上既有数据，也有数据校验信息，数据块和对应的校验信息会存储于不同的磁盘上，当一个数据盘损坏时，系统可以根据同一带区的其他数据块和对应的校验信息来重构损坏的数据。RAID5 可以理解为 RAID0 和 RAID1 的折中方案，可以为系统提供安全保障，但保障程度比 RAID1 要低而磁盘空间利用率比 RAID1 高。



- 云存储的 3 种存储虚拟化技术及其描述：基于主机的虚拟化、基于存储设备虚拟化、基于存储网络虚拟化。

存储虚拟化（Storage Virtualization）：对存储硬件资源进行抽象化表现。

基于主机的虚拟化存储的实现，其核心技术是通过增加一个运行在操作系统下的逻辑卷管理软件将磁盘上的物理块号映射成逻辑卷号，并以此实现把多个物理磁盘阵列映射成一个统一的虚拟的逻辑存储空间（逻辑块），实现存储虚拟化的控制和管理。

基于存储设备虚拟化技术依赖于提供相关功能的存储设备的阵列控制器模块，常见于高端存储设备，其主要应用针对异构的 SAN 存储构架。

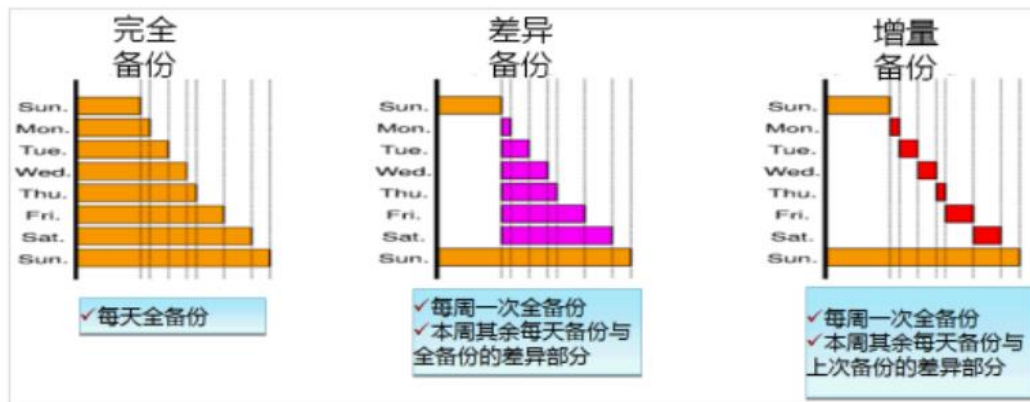
基于存储网络虚拟化技术核心是在存储区域网中增加虚拟化引擎实现存储资源的集中管理，其具体实施一般是通过具有虚拟化支持能力的路由器或交换机实现。

- 数据容错的 2 种技术方式及其定义：基于复制（Replication）的容错技术、基于纠删码（Erasure Code）的容错技术。

基于复制的容错技术：对一个数据对象创建多个相同的数据副本，并把得到的多个副本散布到不同的存储节点上。

基于纠删码的容错技术：是一类源于信道传输的编码技术，因为能够容忍多个数据帧的丢失，被引入到分布式存储领域，使得基于纠删码的容错技术成为能够容忍多个数据块同时失效的、最常用的基于编码的容错技术。

- 数据备份是容灾的基础，是为防止突发事故而采取的一种数据保护措施，根本目的是数据资源重新利用和保护，核心的工作是数据恢复。
- 数据备份有完全备份，差异备份和增量备份。



- 数据缩减技术：自动精简配置，自动存储分层，重复数据删除，数据压缩（无损压缩和有损压缩），内容分发网络（Content Distribute Network, CDN）
- 典型的云储存服务：EMC ATOMS, Amazon S3, Amazon EBS, Google File System, Google Big Table（构建于 GFS 之上的分布式数据库系统）

第八章

- 典型的 IaaS 平台和 PaaS 平台

Amazon: Amazon Web Services (AWS) 是 Amazon Web 服务的总称，共包括了 12 个门类共 33 种云计算产品与服务。

Google: Google AppEngine (GAE) Web 运行平台，是用户的业务系统能够运行在 Google 分布式基础设施上，具有易用性、可伸缩性、低成本的特点。Gmail 是一个电子邮箱的 SaaS 平台，Google Docs 是一个界面类似于微软 office 产品的 SaaS 平台。

Microsoft: Microsoft Azure 是 Microsoft 面向云计算推出的平台即服务(PaaS)产品，包括四大部分：Windows Azure, SQL Azure, Windows Azure AppFabric, Windows Azure Marketplace。

典型 IaaS 平台：

(1) **OpenStack** 是一个开源的基础设施即服务(IaaS)云计算平台，可以为公有云和私有云服务提供云计算基础架构平台，开发语言是 Python，采用 Apache 许可证发布该项目源代码。

(2) **CloudStack** 是一个开源的云操作系统，它可以帮助用户利用自己的硬件建立自己的云服务，CloudStack 可以帮助用户更好地协调服务器、存储、网络资源，从而构建一个 IaaS 平台。

(3) **Eucalyptus**(Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems)是一种开源的软件基础架构，用来通过计算集群或工作站群实现弹性的、实用的云计算，其主要由五个核心的组件构成：节点控制器(Node Controller, NC)，集群控制器(Cluster Controller, CC)，云群控制器(Cloud

Controller, CLC), 存储控制器(EBS)和存储服务(Walrus, W), 它们能够相互协作共同提供所需的云服务。
其中 OpenStack 优势显著。

(1) 综合比较

IaaS开源平台对比

开源平台	分层架构	SOA	组件化解耦	扩展性	开放平台
Eucalyptus	较好 云、集群和节点分层; 缺乏API层	较好 CLC采用Mule编排服务; 服务间使用Web Services接口交互	较好 计算和存储服务解耦; 网络没有与计算解耦	弱 只在虚拟化技术和调度算法上支持	弱 采用Mule作为服务总线
CloudStack	好 分为API层、核心编排层和资源层	弱 整体为一体化设计, 无SOA设计	一般 整体一体化设计, 通过代理来解耦	好 通过插件框架和代理支持扩展	较好 初步形成开发平台, 覆盖基本功能
OpenStack	好 分为访问层、核心服务层和共享服务层; 每个服务分为API层、核心控制层和资源层	好 完全按照SOA设计; 服务间使用API接口	好 计算、网络、存储、镜像和身份认证等均为独立服务	好 各服务器内部组件均支持框架和插件设计	好 采用统一开发语言、类似的软件架构和实现技术

典型 PaaS 平台:

- (1) **Cloud Foundry** 是 VMware 推出的业界第一个开源 PaaS 云平台, 它支持多种框架、语言、运行时环境、云平台及应用服务。是由相对独立的多个模块构成的分布式系统, 每个模块单独存在和运行, 各模块之间通过消息机制进行通信, 本身是基于 Ruby 语言开发的, 每个部分可以认为拿来即可运行, 不存在编译等过程。
- (2) **OpenShift 3**: 是一个私有的平台即服务解决方案, 主要用来在容器中搭建、部署以及运行应用程序, 是基于 Apache2.0 许可的开源的软件。OpenShift 广泛支持多种编程语言和框架, OpenShift 3 基于 Docker 容器技术, 是基于 Red Hat 的 Linux 基础之上, 其运行单元是"POD", 其架构与前面的版本完全不同。

1. 技术方面

- OpenShift 3底层容器技术是Docker, Cloud Foundry提出了一套兼容Docker格式的技术, 但并不是以Docker作为平台的实现技术。
- 到目前为止, Cloud Foundry解决问题是云化的应用程序, 但是对于传统的有状态的应用程序, Cloud Foundry还没有一个清晰的技术性方案。相比之下OpenShift 3通过使用Docker技术可以运行传统的有状态的应用。
- Cloud Foundry提供了一个重量级的应用, 也就是大数据套件 (Big Data Suite)。
- Cloud Foundry提供了一个移动服务套件 (Mobile Services Suite), 可以和大数据服务以及通用服务进行整合, 通过集成在PaaS上建立移动应用程序。

2. 社区基础

在Docker社区里面, 除了Docker公司之外, Red Hat在企业里的贡献程度是位居第二的。Cloud Foundry背后的支持公司Pivotal还在做另一套容器技术。但是目前来讲, Docker技术已经被大家认可。另外, Cloud Foundry自己有一些编排技术, 但是Red Hat用的是Google的编排技术, 这个编排技术在整个社区里面被公认为是最好的。

3. 产品就绪

目前OpenShift 3已经是正式发布版 (GA) 了, 已经可以发行给企业客户来使用, 但是Cloud Foundry还没有正式发布版。从目前来看, OpenShift 3适合于应用程序密集的部署, 而Cloud Foundry更适合于大数据和移动应用的部署。

从目前看来, OpenShift 3 适合于应用程序密集的部署, 而 Cloud Foundry 更适合于大数据和移动应用的部署。

说明

这次考试分为 A、B 两卷，每份试卷包括选择题、判断题、填空题和简答题。其中，选择题共 20 道，每道 2 分，共 40 分。判断题 10 道，每道 2 分，共 20 分。填空题 5 道，每空 1 分，共 10 分。简答题 5 道，每道 6 分，共 30 分。