# 第一讲：云计算概述

## 一、什么是云计算：

定义 1：云计算(Cloud Computing)是网格计算、分布式计算、并行计算、效用计算、网络存储、虚拟化、负载均衡等传统计算机技术和网络技术发展融合的产物。它旨在通过网络把多个成本相对较低的计算实体整合成一个具有强大计算能力的完美系统，并借助 SaaS、PaaS、IaaS、MSP 等先进的商业模式把这强大的计算能力分布到终端用户手中。

定义 2：云计算（cloud computing）是基于互联网的相关服务的增加、使用和交付模式，通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源。（百度）

定义 3：一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括网络，服务器，存储，应用软件，服务），这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。（NIST）

定义 4：一种能够将动态伸缩的虚拟化资源通过互联网以服务的方式提供给用户的计算模式。云计算可分为云和计算两部分理解，而云即指网络，网络的作用除了接入和路由，还包括计算、存储和服务，将分布的资源整合起来，使用户可以透明的只关注使用而不受当前设备限制。（课件）

友情提示：

① 推荐使用定义 3 和定义 4

② 云计算按照服务类型分为基础设施云、平台云和应用云。也即在不同的层次上提供相应的服务。

基础设施云：提供底层、操作硬件资源的接口，获取计算和存储能力

平台云：为应用的开发和部署提供平台

应用云：提供共有的应用，通过网络访问，用户即可使用相应的功能

③ 狭义：通过网络将庞大的运算处理程序自动拆分成若干个较小的子程序，再交给多个服务器所组成的庞大系统经过搜索、运算分析后将处理结果返回给用户

## 二、云计算的优势：

答案一（来自百度）：

1.资源灵活

以并行计算为核心，按需调度计算任务分配和计算资源，并提供从数据导入整合处理、计算模型设定到计算结果输出、多形式展现、应用 API 等完整的数据处理服务。可为科学研究、公共事业、政府等提供可靠灵活的平台。

2. 安全可控

按组隔离访问，自定义防火墙策略，天然防 ARP 欺骗，具有防 DDOS 攻击能力。

3. 数据可靠

采用分布式存储系统，数据互备，快速备份和恢复。支持各种数据处理、计算模型，满足不同领域、不同特点的计算需求。多副本容错，数据安全无忧；海量存储，空间无限。

4. 节约成本

简单的配置，完整的平台，即取即用，无需花费大量的时间搭建、维护计算环境，以服务的方式使用计算及存储资源，按需取用，按需付费，再需要购买大量设备。

5. 提高现有计算力的使用率

针对计算力分布及需求不均衡的情况，通过虚拟化技术，即使在不添加新的计算能力的前提下，通常也能有效地提高物理机硬件利用率。

6. 统一的管理

通过云计算的统一整合，实现了把物理资源池化的机制，通过云平台的统一引擎调度，从而实现了统一的管理入口。实现简单统一的管理模式。

7. 更廉价的容错性

同样，在传统的 IT 架构的模式下，对业务系统的高可用保障，通常都是通过基于本业务系统的高可用机制（双机 HA、备份等）来实现容错，没有从全局的角度考虑，从造成了大量的物理计算力的冗余。而现在的基础设施云技术已经具备了各种高可用性方案，使得容错可以被放在虚拟机级别。在相同的容错级别下，后者实现容错的代价要小很多。或者，花相同的代价，基础设施云所能够实现的容错级别又会高很多。

8.附加的社会效应

虚拟化使多用户共享共同资源成为现实，有效避免现有数据中心初期投入过大而造成的计算资源浪费。现有基础设施云上还有成熟的技术，能根据系统总体负荷，灵活地实现对系统资源的启停，在不影响业务应用的前提条件下，有效地实现数据中心的节能环保，在为企业省下电费的同时又实现了企业对社会的环保责任。

答案二（来自课件）：

1. 优化产业布局：
   成本关键由硬件成本转为电力成本和散热成本
   分散的高能耗模式转为集中的资源友好模式
   自给自足资源作坊转为规模效应工业化资源工厂
2. 推进专业分工：
   专业云计算厂商；实力雄厚的科研技术团队；丰富经验的维护管理团队；
3. 减少初期投资：
   更少的基础设施；更少的软件投入；更少的人力；更短的培训周期；更灵活的转型支持
4. 降低管理开销：
   服务化管理；不需自己维护；随需应变的解决方案
5. 提升资源利用率：
   高效解决常规业务；更好的应对突发事件；更加平衡的资源分配和负载

友情提示：

① 理解下云计算特征，针对特征就能得出云计算优势，总的来说就是资源共享，节约成本，按需提供服务

② 云计算特征：
   硬件、软件都是资源；资源可按需进行动态扩展与配置；按用计费，无需管理；
   物理上分布式共享，逻辑上以单一整体呈现

三、云计算的动因：

答案一（网络）：

1. 云计算发展的必然性

在云计算领域中许多技术并不是最新的概念，但云计算将他们统整合起来使得许多原先在实验室中的或在某些特定领域内使用的技术逐渐被广泛应用，并显示了其商业价值。

在技术层面推动云计算发展的重要因素是分布式应用的不断成熟和大力发展。当前计算机网络应用中，分布式计算越来越普遍，逐渐发展成主流的计算模式而取代集中式的大型计算机，而这也正是云计算所需要的。 云计算模式在技术上和应用上比传统集中式的 IT 模型具有更好的性能价格比。用户不需要花几十万美元就能获得高效能计算。而且多数应用场景本身就是分布式的，如工业企业的应用，他们的管理部门和工业现场可能不在同一个地理区域。这些分散的应用环境和工作模式都使得云计算成为一种广受欢迎的 IT 服务模型。

2. 应用需求方面的前景

粗略地计算，目前的个人计算机每个 CPU 芯片的处理能力是 200MIPS，即每秒钟执行两亿次指令。最近 Yahoo 公司报道他们已经实现了一万个节点（node）的应用部署，就是一万台计算机连接的分布式系统，总的处理能力是 2,000,000MIPS，最快的芯片也达不到这个速度。从当前科技发展状况来看，在一定面积上设计的芯片的速度存在极限，这个极限不可逾越。当前世界著名的超级计算机--TOP500，达到每秒执行几百万亿次指令的速度，采用分布式设计的，世界第一的 IBM BlueGene 超级计算机采用了 32 部机架，每部机架部署有 768 个 PowerPC440 CPU。

现在社会和家庭拥有的个人计算机，最多只有 30%的计算能力被利用，而其余 70%的实际上是被闲置的。这些闲置的计算机资源和计算能力只有通过分布式系统才能得到有效的利用，这样才可以大大提高一个国家的计算能力，而计算能力是衡量一个国家国力和科学研究能力的重要指标，是一个国家和地区的一种重要的战略资源，不亚于石油和其他战略物资的重要性。

云计算是把普通的服务器或者个人计算机连接起来以获得高性能和高可用性计算机的功能。云计算模式必定能大大提高我国科学计算机和商业计算能力，使得我国经济竞争力大大提升。美国和欧洲有许多分布式计算系统，他们动员和使用这些社会计算能力进行人类基因组学（Genomics）的研究、天文学问题研究、数学难题研究以及其他的科学问题研究。

云计算被关注是在因为人们考虑 IT 业到底需要什么之后，人们需要找到一种办法能够在不增加新的投资，新的人力和新的软件的情况下增加互联网的能力和容量。目前，云计算正处于一个起步的阶段，大大小小的公司提供着各式各样的云计算服务，从软件应用到网络存储再到邮件过滤，这些公司一部分是基础设备提供商，另一部分是像 Salesforce.com 之类的 SAAS（软件即服务）提供商，云计算的聚合和整合正在逐步推进。

答案二（课件）：

1. 芯片与硬件技术：
   芯片的摩尔定律遇到瓶颈；
   硬件能力激增，成本下降，使独立运作的公司集中客观的硬件能力实现规模效益成为可能
2. 资源虚拟化：
   虚拟化技术的发展；资源在云端，需要被统一管理；
3. 面向服务的架构 SOA：
   统一通信标准，更加丰富的服务，更加松散耦合、灵活的 IT 架构
4. 软件即服务思想：
   实力雄厚的大公司负责基础设施，小企业通过创新挖掘市场，转变了人们使用服务的方式，使终端用户熟悉服务的交互模式
5. 互联网技术：
   基础设施的进步，多种接入方式和更广阔的覆盖，使带宽和可靠性得到大幅提升
6. Web2.0 技术：

用户从信息的获得者变成信息的贡献者，微波等改变了人们的生活方式，为云计算提出了内在需求

友情提示：

答案一仅用来理解即可，答案二用来考试比较靠谱

## 第二讲：云服务

普及时间：

服务就是通过一系列活动，而不是实物的方式，满足对方的需求。

### 一、云服务的基本层次：

云计算通过不同层次的架构实现不同类型的服务以及满足用户对这些服务的各种需求。

包括以下几个层次的服务（通用性依次降低）：

1.基础设施层：基础设施即服务（IaaS），基础硬件，面向软件，提供了核心计算资源和网络架构的服务。消费者通过 Internet 可以从完善的计算机基础设施获得服务。

例如：操作系统访问，负载平衡，防火墙，路由

实例：AWS：EC2（亚马逊云平台弹性计算云：提供虚拟机给用户）

2.平台层：平台即服务（PaaS），开发或支撑软件，面向应用软件，具有通用性和可复用性的软件资源集合。提供平台给系统管理员和开发人员，以构建、测试及部署定制应用程序。实际上是指将软件研发的平台作为一种服务，以 SaaS 的模式提交给用户。降低了管理系统的成本。

例如：数据存储，数据库，可扩展服务

实例：GAE，亚马逊存储服务，微软 Azure

3.应用层：软件即服务（SaaS），应用软件，面向最终用户，通过 Internet 提供软件的模式，用户无需购买软件，而是向提供商租用基于 Web 的软件，来管理企业经营活动，且无需对软件进行维护。

### 二、IaaS 的基本功能：

1.资源抽象:

对硬件进行虚拟化、向下屏蔽硬件产品差异、向上对硬件进行统一的管理。

对资源进行粒度划分并管理、将物理资源放入统一的资源池中、进行管理并呈现给用户。

粒度划分是指：将资源分为虚拟机、集群、虚拟数据中心或者云

硬件资源包括：计算、存储和网络等

2.资源监控:是保障基础设施层高效工作的关键，是负载管理的前提。

对不同资源采取不同监控方式，拥有不同的监控层次（粒度），不同监控对象（物理、逻辑、解决方案）

3.负载管理：解决负载过低造成资源浪费和负载过高造成上层服务受到影响。

4.数据管理：

在多种数据并存，物理上分布式存储的基础上，保证数据的完整性，可靠性和可管理性

5.资源部署：通过自动化部署流程将资源交付给上层应用，使基础设施服务变得可用。

部署方式随基础设施层构建技术不同有巨大差异，虚拟化技术简化了资源部署的过程。

支持动态资源可伸缩性：用户的服务负载过高时，通过增加服务实例降低负载

故障恢复与硬件维护：多节点的数据冗余及快速复制环境完成物理迁移

6.安全管理：保证基础设施资源被合法地访问和使用。

7.计费管理：变买为租，对使用情况进行监控，按量、按时间计费。

三、PaaS 的基本功能：

普及时间：平台分为基于快速开发目的的技术平台；基于业务逻辑复用的业务平台；基于系统自维护和自扩展的应用平台

1. 开发平台：平台层是其上运行的应用的开发平台
   应用模型（编程语言，元数据或打包发布格式）
   API 代码库：增加代码重用，减少重复工作，缩短开发周期
   必要的开发测试环境：开发人员无需安装和配置相关软件
2. 运行时环境：
   隔离性:业务、数据、应用间、用户间隔离；
   可伸缩性：根据负载和业务规模对资源进行动态分配；
   可复用性：资源释放、回收、宏观无限、微观有限。
3. 运营环境：
   应用更新: 功能添加、版本升级；
   应用升级：提供应用升级脚本
   应用监控: 监控应用状态，通过响应时间、吞吐量、工作负载、处理的请求量监控；
   资源消耗监控: 调用基础设施层的服务进行实时分配，统计报表；
   应用卸载: 直接删除、备份后删除、卸载说明、卸载协议；
   应用计费统计：资源使用统计、业务使用统计。

## 第四讲：虚拟化

一、虚拟化的概念：

答案一：虚拟化，原本是指资源的抽象化，也就是单一物理资源的多个逻辑表示，或者多个物理资源的单一逻辑表示。具体到服务器虚拟化，就是多个物理资源的单一逻辑表示。指计算元件在虚拟的基础上而不是真实的基础上运行，是一个为了简化管理，优化资源的解决方案。

答案二（来自课件）：
虚拟化是表示计算机资源的抽象方法，通过虚拟化可以用与访问抽象前资源一致的方法来访问抽象后的资源。这种资源的抽象方法并不受实现、地理位置或底层资源的物理配置限制。

虚拟化的对象是各种各样的资源，解除了真实环境中各个层次（比如硬件、操作系统、应用程序）的耦合关系，经过虚拟化后的逻辑资源对用户隐藏了不必要的细节，用户可在虚拟环境中实现在真实环境中的部分或全部功能。

补充：广义虚拟化（信息知识虚拟化，能力虚拟化，实物虚拟化）
对象脱离原有环境，在计算机上被表示，通过计算机控制按需获取

二、服务器虚拟化的特性：

多实例：一个物理服务器上可以运行多个虚拟服务器，还支持多个客户操作系统，物理系统资源以可控的方式分配给虚拟机。

隔离性：虚拟机之间完全隔离，一个虚拟机崩溃不会对其他虚拟机造成影响，虚拟机之间的数据相互独立、不会泄露，虚拟机之间如果需要互相访问，方式等同于独立物理服务器之间的互相访问。

封装性：与硬件无关，对外表现为单一的逻辑实体，一个虚拟机可以方便地在不同硬件之间复制、移动，将不同访问方式的硬件封装成统一标准化的虚拟硬件设备，保证了虚拟机的兼容性。

高性能：可通过扩展获得“无限”的性能，虚拟化抽象层需要一定的管理开销。

补充：服务器虚拟化

服务器面临着利用率低下，基础设施老化，成本功耗增加等一系列问题，因此将虚拟化技术应用于服务器上，将物理服务器虚拟化为多个虚拟服务器。以建立动态、自动化虚拟 IT 环境，方便部署，具有高性能、可扩展性和稳定性。

三、服务器虚拟化的关键技术：

计算虚拟化：

1. CPU 虚拟化：一个 cpu 上装多个虚拟机，可以模拟出多个操作系统
2. 计算负载的动态分配
3. 能耗管理。

存储虚拟化：

1. 内存虚拟化：虚拟地址
2. 磁盘存储动态分配。

设备与 I/O 虚拟化：

软件方式实现，统一、标准化的接口，操作指令转移。

实时迁移技术：

将整个虚拟机的运行状态完整、快速地从原宿主机的硬件平台转移到新的宿主机硬件平台；具有实时性；内存页面不断的从源虚拟机监视器拷贝到目标虚拟机监视器；拷贝结束后，目标虚拟机开始运行，虚拟机监视器切换到目标虚拟机上，源虚拟机终止；广泛应用于实时系统的硬件维护。

四、创建虚拟化解决方案的步骤：

创建基本虚拟镜像：

创建虚拟机，安装操作系统，关停虚拟机（保存配置文件及虚拟镜像）。

创建虚拟器件镜像（虚拟器件包括 linux，Apache，MySQL，PHP）：

1. 分析调研：分析解决方案应用模块，各模块关联关系，确定器件多种形态
2. 编制元数据脚本：编制配置脚本，抽象配置元数据，测试
3. 制作虚拟器件：创建虚拟镜像，安装中间件和应用，安装配置脚本和元数据

发布虚拟器件镜像：OVF 格式（硬件信息和软件属性）

管理虚拟器件镜像：（虚拟器件镜像大小：几 G-几十 G，数量巨大）

管理目标要达到快速检索、减少公共仓库磁盘使用量、版本控制。

迁移到虚拟化环境：迁移技术：P2V（物理到虚拟）,V2P。

五、什么是数据中心：

答案一：数据中心(DataCenter)通常是指在一个物理空间内实现信息的集中处理、存储、传输、交换、管理。

计算机设备、服务器设备、网络设备、存储设备等通常认为是网络核心机房的关键设备。 关键设备运行所需要的环境因素，如供电系统、制冷系统、机柜系统、消防系统、监控系统等通常被认为是关键物理基础设施。

答案二：数据中心在系统上分为硬件（支撑系统和计算机设备）和软件（程序及服务），从逻辑上分为建筑设施，支撑系统，计算机设备以及信息服务。

答案三：数据中心是一整套复杂的设施。它不仅仅包括计算机系统和其它与之配套的设备（例如通信和存储系统），还包含冗余的数据通信连接、环境控制设备、监控设备以及各种安全装置。（wiki）

答案四：多功能的建筑物，能容纳多个服务器以及通信设备。这些设备被放置在一起是因为它们具有相同的对环境的要求以及物理安全上的需求，并且这样放置便于维护”，而“并不仅仅是一些服务器的集合”。

补充：新一代数据中心需求

1. 合理规划（寿命、更换周期、冗余性）
2. 流程化
3. 可管理性、可伸缩性、可靠性
4. 降低成本、节能环保

友情提示：

答案一为主，其他几个为辅作为理解，课件上只有答案二的分类情况，大致上理解它的概念即可。也可以从它的需求来理解它到底是个什么东西，个人理解其实就是把存放数据信息等的服务器放在一起，从而实现数据的统一管理，降低成本 balabala 的。

六、虚拟化与云计算的关系：

云计算是一种基于互联网的计算方式，通过这种方式，共享的软硬件资源和信息可以按需求提供给计算机和其他设备，主要是基于互联网的相关服务的增加、使用和交付模式，通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源。

虚拟化是云计算发展的动因之一，构成云计算的基础。虚拟化技术使得硬件资源可以被有效的细粒度分割和管理，以服务方式提供硬件和软件资源成为可能。

虚拟化为云计算提供了很好的底层技术平台。云计算是在虚拟化出若干资源池以后的应用。

## 第五讲、 openstack

openstack 是一个旨在为公共及私有云的建设与管理提供软件的开源云平台管理项目，不是一个软件。首要任务是简化云的部署过程并为其带来良好的可扩展性。

一、AWS 模式是什么？有什么优点？

（1） AWS 模式：

Amazon Web Service 的简称，是一个典型的 IaaS 服务。提供了一组服务，包括存储（S3）、计算 EC2、网络、消息传递和数据库服务等。用户应用使用 IaaS 基础 IT 资源，将 PaaS 和通用服务作为应用架构中的组件来构建自己的服务。用户所需要的 IT 资源不在公司自己的数据中心里面，这些资源可以通过互联网获得，没有固定的投资成本。

特点：

1. 通过 Web Service 接口开放数据和功能
2. 一切以服务实现
3. 通过 SOA 架构使系统达到松耦合

（2） AWS 优点：

1. 大大降低了当今 Web 环境中的“贫富差异”， 公司不再需要承担高额的基础设施投资和维护成本，可以不购买基础设施，直接通过互联网按需租用 Amazon 的基础设施，并且一切维护工作都由 Amazon 的专业维护人员来完成，只需要为自己所租用的资源付费。企业可以把注意力集中在业务思想上，而不需要为服务器操心，不需要担心磁盘空间不足等问题。
2. 一切通过服务提供
3. 完全的 SOA 架构，各部分相互独立提供服务
4. 良好的扩展性和弹性设计

二、IaaS 模式核心需求有哪些？

1. 对于云拥有者：配置和操作基础架构
2. 对于服务提供者：注册云服务；查看服务使用和计费情况
3. 对于服务使用者：创建和存储自定义镜像；启动、监控、终止实例

三、openstack 都包含哪些核心项目？作用是什么

1. compute 计算（nova）
   一套控制器，用于为单个用户或使用群组启动虚拟机实例。将提供预制的镜像或是为用户创建的镜像提供存储机制，这样用户就能够将镜像以虚拟机的形式启动。它同样能够用于为包含着多个实例的特定项目设置网络。
2. object storage 对象存储（swift）
   一套用于在大规模可扩展系统中通过内置冗余及容错机制实现对象存储的系统。
3. image service 镜像服务（glance）
   一套虚拟机镜像查找及检索系统，是计算、存储服务的载体。
4. identity 身份验证（keystone）
   用户管理：记录用户和其权限；服务目录，提供可用服务和该服务 api 的终端地址
5. dashboard 自助门户（horizon）
   为所有服务提供了一个模块化的 web-based 用户界面。使用这个 Web GUI，可以在云上完成大多数的操作，如启动实例，分配 IP 地址，设置访问控制等
6. network connectivity 网络管理（quantum）
   在接口设备之间提供“网络连接作为一种服务”，允许用户创建自己的网络，然后连接接口，提供一个可插拔的体系架构，它能支持很多流行的网络供应商和技术

7. block storage 块存储（cinder）：提供稳定的数据块存储服务

四、镜像和实例有什么区别和联系？

镜像是一种文件形式，是冗余的一种类型，是计算、存储的载体，相当于一种模板，一个固定搭配。实例是镜像的一种具体化，一个镜像可以创建多个实例。

友情提示：

这个问题我是真没搞明白，也没查清楚。像不像对象与实例的关系呢？大家可以自行去查，然后共享答案~

五、Nova 有哪些核心模块？工作过程？

Nova 通过大量的进程合作，将最终用户的 API 请求发送到正在运行的虚拟机之上。

核心模块：

1.nova-compute负责对虚拟机实例进行创建、终止、迁移、Resize的操作。从队列中接收请求，通过相关的系统命令执行他们，再更新数据库的状态。

2.nova-volume管理映射到虚拟机实例的卷的创建、附加和取消。

3.nova-network从队列中接收网络任务，然后执行任务控制虚拟机的网络。

4.nova-scheduler 提供调度，来决定在哪台资源空闲的机器上启动新的虚拟机实例

5.Queue为守护进程传递消息。

6.SQL database 存储云基础架构中的各种数据。包括了虚拟机实例数据，网络数据等。

工作过程：

1. 用户输入命令，api 会查看这种类型的 instance 是否达到最大值 给 scheduler 发送一个消息（实际上是发送到 Queue 中）去运行这个实例。
2. 调度器接收到了消息队列 Queue 中 API 发来的消息，然后根据事先设定好的调度规则，选择好一个 host，之后，这个 instance 会在这个 host 上创建。
3. 真正创建 instance 是由 compute 完成的，通过 glance 查找镜像。
4. 根据找到的镜像，到 database 中查找相应的数据
5. volume 创建虚拟机实例的卷
6. network 为虚拟机分配 IP 等网络资源

六、keynote 权限控制过程是什么？

用户传 credential 给 keynote 进行请求，keynote 进行认证以后分配给用户一个 token（令牌），用户获得权限，将令牌和虚拟机请求传给 nova，nova 向 keynote 验证令牌，获得权限后连同对镜像的请求传给 glance，glance 向 keynote 验证令牌，把镜像传给 nova；nova 再将用户接入网络的请求传给 quantum，验证成功后，即传出成功访问的回答。

七、quantum 原理是什么？——实现网络连接管理，解决网络虚拟化问题

友情提示：

这一部分在 ppt 里面也是一张图，关键是我看不懂，所以没办法跟其他部分一样描述出来，所以大家去百度吧，或者把图看明白了也行~~

八、swift 的核心概念有哪些？

1. object：对象。基本的存储实体，所有数据按照对象进行存储。
2. container：容器。对象的装载体，组织数据的方式，存储的隔间，类似于文件夹，但不能嵌套。
3. account：账户。权限单位，一个 account 拥有若干 container。

九、Swift 的组件有哪些？都有神马作用？

1. Proxy Server：是提供 Swift API 的服务器进程，负责 Swift 其余组件间的相互通信。对于每个客户端的请求，它将在 Ring 中查询 Account、Container 或 Object 的位置，并且相应地转发请求。
2. Storage Server：提供了磁盘设备上的存储服务。
3. Consistency Servers：查找并解决由数据损坏和硬件故障引起的错误
4. Ring：Swift 最重要的组件，用于记录存储对象与物理位置间的映射关系。

十、Ring 算法思想是神马？

Swift 利用一致性哈希算法构建了一个冗余的可扩展的分布式对象存储集群。主要目的是在改变集群的 Node 数量时，能够尽可能少地改变已存在 Key 和 Node 的映射关系。

首先计算每个节点的哈希值，并将其分配到一个 0~232 的圆环区间上。
其次使用相同方法计算存储对象的哈希值，也将其分配到这个圆环上。
随后从数据映射到的位置开始顺时针查找，将数据保存到找到的第一个节点上。如果超过 232 仍然找不到节点，就会保存到第一个节点上。

友情提示：
ppt 上有个示意图，偷懒木有加上~可以去看看哒~

十一、 Quorum 协议的内容是什么？ Swift 读写协议

① W+R>N ：以保证对副本的读写操作至少产生一个交集，从而保证可以读取到最新版本；

② ②W>N/2：写操作需要满足至少一半成功率

N为存储数据节点的数量，W为写入请求中写入成功数量，R为读取请求中读取成功的数量

补充：swift放弃严格一致性（ASID）模型，采用最终一致性模型，达到高可用性和无限扩展

第六讲 、云存储

一、大规模数据存储面临的新问题与挑战

1. 成本问题：数据量的增长对扩充处理能力提出要求，成本大大增加

2. 容量问题：数据量越来越大，个人难以承受，需要第三方提供存储服务
3. 可靠问题：大规模数据的共享、访问的安全性需要保障，分布性数据一致性保存问题
4. 使用问题：存储、检索、浏览以及之上的增值服务（推荐、个性化索引）已经走进我们的生活
5. 传统模式存储大规模数据受到数据库表结构的限制，并且效率从降低的量变到不可用的质变

二、GFS 体系结构——分布式文件存储系统

1. 中心服务器模式：一个 GFS 集群由一个 master 和大量的 chunkserver 构成，并被许多客户(Client)访问。通过单个 master 来协调数据访问、元数据存储，掌握 chunkserver 情况，方便负载均衡。客户与 master 的交换只限于对元数据(metadata)的操作，所有数据方面的通信都直接和 chunkserver 联系。
2. 文件被分成固定大小的块。每个块有一个不变的、全局唯一的标识，是在块创建时由 master 分配的。每个数据块至少在 3 个数据块服务器上冗余，提高可靠性
3. 不缓存数据：采用流式读写，chunkserver 上的数据存储使用本地文件系统
4. 在用户态下实现，提供专用的访问接口

三、GFS 的容错机制

1. Master 容错
   命名空间（目录结构）以及 chunk 与文件名的映射通过日志容错
   chunk 副本位置信息存储于 chunk server，master 故障时可恢复
2. ChunkServer 容错——副本方式
   每个 chunk 有多个存储副本，存储于不同的服务器上；每个 chunk 分为若干 block，对应一个 32bit 校验码

四、Paxos 协议——分布式环境下保持一致性的协议

1. 协议中基本概念：
   （1） 角色
   ①client：用户需求　　②proposer：协调者，即取得模块并希望运行的机器
   ③leader：特别的 proposer
   ④acceptor：所有的服务器，它们会对 Porposer 提出的方案进行裁决，以确认是否同意 Porposer 的请求，并视情况对 Proposer 进行回复，Acceptor 的一次回复被称为一次投票
   ⑤learner：任意希望获取模块处理情况的机器，它通过向 Acceptor 发送学习请求而获知所有模块的分配情况
   ⑥quorum：大多数 acceptor
   （2） 议案：议案内容为<序列号，方案>，序列号是 proposer 自己产生并单调递增
2. 选举过程：
   Phase1.准备阶段：
   (a)proposer 选择一个提案编号n并将prepare请求发送给 acceptors 中的一个多数派；
   (b)acceptor 收到prepare消息后，如果提案的编号大于它已经回复的所有 prepare 消息，则 acceptor 将自己上次的批准回复给 proposer，并承诺不再批准小于 n 的提案；
   Phase2. 批准阶段：

(a)当一个 proposor 收到了多数 acceptors 对 prepare 的回复后，就进入批准阶段。它要向回复 prepare 请求的acceptors 发送 accept 请求，包括编号 n 和根据 P2c 决定的 value（如果根据 P2c 没有决定 value，那么它可以自由决定 value）。

(b)在不违背自己向其他proposer的承诺的前提下，acceptor 收到 accept 请求后即批准这个请求。

友情提示：

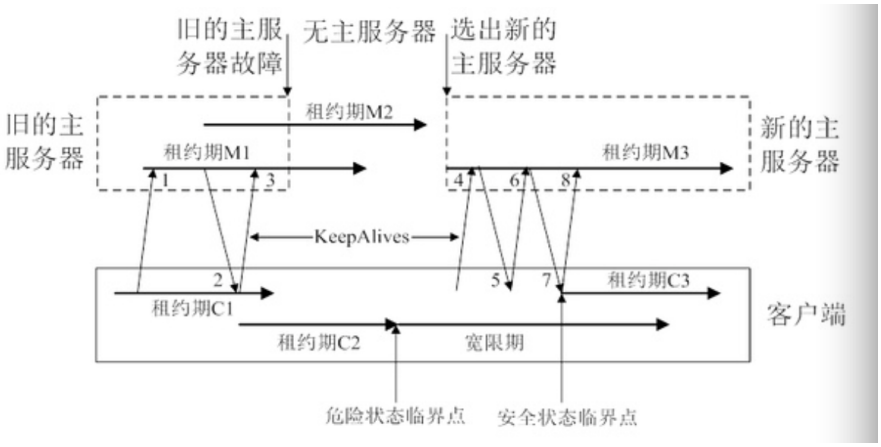
ppt中有一个例子和一张流程图，可以作为理解参考。

五、Chubby锁机制——粗粒度的分布式锁服务

1. 本质：分布式的、存储大量小文件的文件系统，锁就是文件
2. 创建文件就是“加锁”操作，创建成功的 server 就是抢到了锁
3. 用户通过打开、关闭和存取文件，获取共享锁或者独占锁，通过通信机制，发送更新信息
4. 一群机器需要选举 master 时，这些机器同时申请某个锁文件。成功获取锁得服务器当选主服务器，并在文件中写入自己的地址。其他服务器通过读取文件中的数据获取 master 的地址。

六、Chubby 通信协议：操作的允许授权是通过限时的、倒计时“租期”系统来处理的

Keep Alive 是周期性发送的一种消息，它有两方面功能：延长租约有效期，携带事件信息告诉客户端更新。正常情况下，租约会由 Keep Alive 一直不断延长。




1、 开始时客户端给主服务器发送一个 keep alive 请求

2、 主服务器等到租约 c1 快要到期时回复相应请求并更新租约期到 c2

3、 客户端收到回应后，任务主服务器活跃，更新租约期到 C2 并且发送新的请求

4、 租约期 c2 到期后没有收到新的回应，进入宽限期，不断跟主服务器联系

7、新主服务器产生批准请求，更新租约期到 C3

七、Bigtable 数据结构——基于 GFS 和 Chubby 的分布式存储系统

分布式多维映射表；通过行关键字+列关键字+时间戳进行索引；对存储的数据不做解析看成字符串，可以存储任意类型数据；无数据校验

1. 行
   每行数据有一个可排序关键字和任意列项；表按照行健有序化处理；
2. 列
   特定含义的数据集合；多个列归并为族，同一族数据压缩保存，是访问控制的基本单元
3. 时间戳
   保存不同时期的数据

八、Bigtable 优化机制

1. 局部性群组：
   根据需要，将原本不存储在一起的数据，以列族为单位存储至单独的子表
2. 布隆过滤器：（判断某个元素是否隶属于集合）
   误判率低，存储空间小，快速确定某个列键位置
3. 合并压缩
   读取多个 SSTable 创建一个新的来保持其中最新数据

九、云存储应用的特点

1.通用的设备支持　2.数据同步与共享　3.任意格式和大小的文件　4.免费+付费模式

## 第七讲、MapReduce 算法原理

一个软件架构，是一种处理海量数据的并行编程模式，只要执行简单的计算，将并行化、容错、数据分布、负载均衡等细节放在一个库里，并行编程时不必关心。

一、MapReduce 算法的架构

每个 mapreduce 任务都被初始化为一个 job，每个 job 分为两个阶段，实现了 Map 和 Reduce 两个功能：

Map：把一个函数应用于集合中所有成员，然后返回一个基于这个处理的结果集

Reduce：对结果集进行分类和归纳

二、WordCount 算例

伪代码：

```
Map(K,V){
    For each word w in V
       Collect(w , 1);
}
Reduce(K,V[]){
    int count = 0;
    For each v in V
       count += v;
    Collect(K , count);
}
```

1. 自动分割文本
2. 分割后的每一对<key,value>进行用户定义的 map 处理，生成新的<key,value>对
3. 系统自动对输出结果集归拢排序，传给 reduce
4. 通过 reduce 生成最后结果

友情提示：

源代码及注释在 ppt 中有

三、Hadoop 执行 MR 的过程

1. master 节点运行 jobTracker 实例，接收客户端 job 请求，一个 job 是对一个数据集的处理

slave 节点运行 TaskTracker 实例，一个 task 是一次 map 或者 reduce 处理过程

2. MR 程序由一个 jar 文件和一个 xml 文件组成，jar 包含程序代码，xml 包含程序配置操作
3. 客户端设定配置之后，提交 job，将 job 数据发送到 jobTracker 的文件系统中，Mapreduce 库会把所输入文件分割成 M 块，放到不同的 datanote 上。
4. JobTracker 将 job 放入队列进行调度，并把 jar 和配置文件放到共享空间
5. Hadoop 有一个机器内进程间操作通信的机制，taskTracker 周期性告诉 jobTracker 工作状态，如果空，job 给 task 分配任务，开一个进程处理
6. MR 程序被传送到各个数据所在的 HDFS 的 datanote 上运行，mapper 被输送到数据节点上进行计算，中间结果被输送到 reducer 上进行计算，Reducer 将计算结果分布式保存到 HDFS
7. 当所有 map 和 reduce 任务完成时，master 会唤醒用户程序，通知任务完成，再取执行新的任务。

四、MR 算法执行过程中的数据流转过程

1. 用户文件上传到 HDFS，由 mapreduce 库切割分别存储到 datanote 上
2. 以行号，文本形式传入 map 中，经过自定义 map 处理产生<key,value>对，构成中间结果集缓存在本地内存中
3. reducer 接到 master 的收集中间结果集任务后，通过远程调用读取中间结果，并将其排序，key 相同的 value 进行合并处理化简。
4. 最终结果写入到 GFS 文件系统中

## 第八讲、mapreduce 算法应用

一、mapreduce 案例算法

1. 词频：
   WordCount 基础上 map 函数中，加入<*,num>键值对，一次 mapreduce 过程得到每个词出现的个数及总数
2. inverted index（若干文本文件->倒排索引结果）：
   map 输出：<{token,fileID},one>
   combiner 输出：<{token,fileID},sum>
   reduce 输出：<key,value> = <word，articleId：num>
3. 计算文本相似度：
   ① 进行倒排索引
   ② 计算两两文章对的相似性
   map：同一个词对应的文章链表中，两两文章构成一组<{article1,article2},num>
   reduece 进行收集
   ③ 统计文章相似度

友情提示：

具体代码 ppt 里面有，建议重点看 map 和 reduce 里面的东西，至于 combiner 当做 reduce 处理

二、算法调优：

1. 算法级调优：
   （1） 更加优化的 key-value 对设置（比如新增加键值对*计算频率用）
   （2） Map 算法
   （3） Combiner 算法：自定义 combiner，减少网络流量
   （4） Partition 算法：控制分发策略，若 key 使多元 key，对 key 解析后再分发
   （5） Reduce 算法
2. 参数级调优：
   将 map 输出作为输入传给 reducer，确保每个 reducer 的输入都按键排序的过程叫 shuffle
   shuffle 是调优的重点
   （1） Map
   ① 产生输出时，通过缓冲写入内存：缓冲区大小，容量阈值，指定路径
   ② 按照要传送到的 reducer 对数据进行 partirion，每个 partition 内部按照 key 进行键内排序：合并流数；
   ③ 压缩 map 输出效率更高：压缩标志；压缩方式
   ④ reducer 通过 http 方式从 map 处得到输出文件分区，分区的工作线程数由 tasktracker 控制：工作线程数（tracker.http.threads）
   （2） Reduce
   ① 只要一个 map 任务完成，reduce 任务就开始复制其输出：复制线程数；获取 map 最大时间
   ② 若 map 输出小，复制到内存中；否则先写入到内存缓冲区，达到阈值后写到磁盘：内存缓冲区占堆空间百分比；溢出阈值；map 输出阈值；
   ③ 后台线程根据合并因子将其合并成更大的排好序的文件：合并因子
   ④ reduce 开始时，内存中 map 输出大小不能超过输入内存阈值，以便为 reduce 提供尽可能多的内存：输入内存阈值
   ⑤ 输出结果写入 HDFS 系统：hadoop 文件缓冲区大小

3. 调优原则：
   给 shuffle 过程尽可能多的内存空间；
   map 和 reduce 函数尽量少用内存；
   运行任务的 JVM 内存尽可能大；
   map 端尽量估算 map 输出大小，减少溢出写磁盘的次数；
   reduce 端的中间数据尽可能多的驻留在内存
   增加 hadoop 的文件缓冲区

友情提示：

算法和参数调优 ppt 里面没有讲具体应该怎么调，只说了可以调哪些，所以估计了解一下哪些地方能促进优化就好了吧

## 第九讲、Hadoop——可靠的共享存储分析系统

### 一、Hadoop 项目的由来

起源于一个开源的网络搜索引擎项目 Apache Nutch，借鉴 GFS，实现了一个开源的实现 HDFS，05 年 nutch 上实现了一个 mapreduce 系统，完成了所有主要算法的 mapreduce+HDFS 移植。

MapReduce+HDFS 从 nutch 移植出来构成了 Hadoop。

补充：Hadoop——一组相关项目的统称

1. MapReduce：分布式数据处理模型和执行环境
2. HDFS：分布式文件系统
3. HBase：分布式按列存储数据库
4. ZooKeeper：分布式、可用性高的协调服务，提供通用的分布式锁服务
5. Pig：数据流语言和运行环境，用于检索非常大的数据集
6. Hive：分布式、按列存储的数据仓库

二、HDFS 体系结构

采用了主从(Master/Slave)结构模型,一个 HDFS 集群是由一个 NameNode 和若干DataNode 组成,其中 NameNode 作为主服务器,管理文件系统的命名空间和客户端对文件的访问操作。DataNode 管理存储的数据。

HDFS 允许用户以文件的形式存储数据,文件被分成若干个数据块,而且这若干个数据块存放在一组 DataNode 上。NameNode 是整个 HDFS 的核心,它通过维护一些数据结构来记录每一个文件被分割成了多少个块、这些块可以从哪 些 DataNode 中获得,以及各个 DataNode 的状态等重要信息。NameNode 执行文件系统的 命名空间操作,如打开、关闭、重命名文件或目录等,也负责数据块到具体 DataNode 的映 射。DataNode 负责处理文件系统客户端的文件读写操作,并在 NameNode 的统一调度下进 行数据块的创建、删除和复制操作。

三、HDFS 运行机制

1. 可靠性保障：
   冗余机制——数据复制
   故障检测——datanode（心跳包，块报告，数据完整性监测）namenode（日志和镜像）
2. 读文件流程：
   ① 客户端调用分布式文件系统对象的 open 方法
   ② 分布式文件系统联系 namenode，得到所有数据块信息，返回数据块地址，并且根据与客户端的距离排序
   ③ 返回一个对象给客户端，调用 read 方法读取数据
   ④ FSDataInputStream 连接距离最近的 datanode 读数据，读完关闭 datanode 读下一个
   ⑤ 完成读取后，客户端调用 close 方法

   每个块读取都要通过校验和确认保证数据完整；
   遇到错误时，尝试从另一个读取，并记录 datanode，通知给 namenode
3. 写文件流程：
   ① 客户端调用分布式文件系统对象的 create 方法创建文件
   ② 系统联系 namenode，执行各种检查保证建立文件不存在，且客户端有权限
   ③ 检查通过则 namenode 为新文件创建一条记录，否则抛出异常
   ④ 返回一个对象给客户端，将写入数据分成数据包并写入内部队列 dataqueue
   ⑤ DataStreamer 处理队列，根据列表要求 namenode 分配适合的新块存储备份
   ⑥ Namenode 分配的数据备份 datanode 形成一个管线，dataStreamer 将数据包一次传输给三个节点
   ⑦ 对象维护一个确认队列，收到管线的确认后，从队列删除数据包。

补充：HDFS 与 GFS 区别

1. 中心服务器模式：单一服务器模式，存在单点故障&多态物理服务器，一台对外
2. 子服务器管理模式：心跳方式告知&Server 轮询从 Chubby 中获取的独占锁
3. 安全模式：副本补足则拷贝至安全数目&读取副本失败，master 负责发起拷贝任务
4. 空间回收：实际数据删除一段时间后实施，先删目录，便于恢复&删除文件

四、Htable 数据结构

行；列（族）；时间戳；

友情提示：

与 bigTable 数据结构相同，可返回参看相关内容

五、HBase 的运行机制

数据存储实体为区域，表按照水平的方式划分为一个或多个区域，每个区域有一个随机 id，且区域内行为键值有序的。区域以分布式方式存储在集群内。通过区域服务器运行：

1. 写：写数据首先写入“预写日志”；先缓存，再批量写入；完成后在日志中做标记
2. 读：区域服务器先在缓存中查找，找到则直接服务；
3. 合并：映射文件数量超过阈值，则区域服务器进行合并
4. 分割：区域文件大过阈值时，按照行方式对半分割；在元信息表中生成子元信息表；主服务器在得知分割后，将子表分配给新的区域服务器服务
5. 失效恢复：将失效服务器的区域分配给其他服务器，原“预写”日志进行分割并分配给新的区域服务器

六、ZooKeeper 的数据读写机制——层次化目录的数据模型

1. ZooKeeper 是一个由多个 server 组成的集群：一个 leader，多个 follower。
2. 每个 server 保存一个数据副本，全局数据一致。
3. 更新请求进行转发，由 leader 实施。
4. 使用约定：
   ① 来自同一个 client 的更新请求顺序执行
   ② 数据更新原子性
   ③ 全局唯一数据视图：无论连接哪个 server，视图都一样
   ④ 实时性

七、Yarn 对 Hadoop 的核心改进

1. 原框架中的 JobTracker 和 TaskTracker 被 ResourceManager, ApplicationMaster 与 NodeManager 取代。
2. ResourceManager 是一个中心的服务,它做的事情是调度、启动每一个 Job 所属的 ApplicationMaster、另外监控 ApplicationMaster 的存在情况。
3. NodeManager 功能比较专一,就是负责 Container 状态的维护, 并向 RM 保持心跳
4. ApplicationMaster 负责一个 Job 生命周期内的所有工作,类似 老的框架中 JobTracker。可以运行在resourceManager以外的机器上

改进后的优点：

1. 减少了jobTracker也即resourceManager的资源消耗，让监测每一个 Job 子任务 (tasks) 状态的程序分布式化

2. ApplicationMaster是一个可变更的部分,用户可以对不同的编程模型 写自己的 AppMst, 让更多类型的编程模型能够跑在 Hadoop 集群中

3. 对于资源的表示以内存为单位，比以task任务数目分配更加合理

4. 客户端调用API或者接口，程序框架改变后不再需要被强制更新。

## 第十讲、云计算高级话题

一、什么是云安全？

有两种理解：

1. 利用云的强大计算、存储能力进行安全保障，包括安全云和云查杀
   木马病毒的特征越来越细，识别对存储和计算要求越来越高，利用云进行计算
2. 云的安全保障
   数据放在云端，操作通过网络，通过一定机制保障云是安全的。

二、云计算的安全威胁

1. 共享技术漏洞 2.数据丢失与数据泄露 3.恶意内部用户 4.账户服务与传输劫持
5. 不安全的 APIs 6.服务的恶意使用 7.不确定风险预测

三、云计算的安全优势

1. 数据可访问性大大提高
2. 分布式存储：实现数据的备份和容错机制
3. 数据高度安全：数据在云端集中存储实现专业的数据保护和备份，更容易实现安全监测
4. 事件快速反应

四、物联网与云计算的关系

1. 物联网指的是将各种信息传感设备与互联网结合起来而形成一个巨大网络。其目的是让所有的物品与网络连接在一起，方便识别和管理。在这个整合的网络中，存在能力超级强大的中心计算机群，能够对整合网络中的人员、机器、设备和基础设施实施实时的管理和控制。具有三个特点：全面感知；可靠传送；智能控制
2. 物联网和云计算之间是应用与平台的关系。物联网的发展依赖于云计算系统的完善，从而为海量物联信息的处理和整合提供可能的平台条件，云计算的集中数据处理和管理能力将有效的解决海量物联信息存储和处理问题。

五、什么是云主机？什么是云终端？

1. 云主机：新一代的共享主机。
   主机公司将它的硬件和网络线路，做成一朵云，向客户提供一些通向这朵云的网络接口API，供其使用。每个用户共享的不是某一台特定的服务器，而是云里所有的服务器。

2. 云终端：
   云的接入终端。核心在云，终端只是云的访问接口。

六、云计算中的信任机制如何构建

建立用户使用云计算服务所需要的信任的社会关系，最基本、最重要的保证在于互联网的民主性所形成的由下而上的力量。事实上，信任不是一次性测试出来的，也不是依靠一套固定指标测出来的，它是云计算运作过程中累积出来的品质，是消除一个个不可信要素的过程。如何更好地抽象、应用这种应用演化中所涌现出来的信任，是云安全中信任管理的关键问题之一。云计算中信任的建立、维持和管理，可以通过社会与技术手段相结合的方式来推动信任机制的完善。

云提供者不但要保障网络传输等的安全性，还要加强数据处理和存储时的安全性，防止外部入侵数据的同时防止内部人员的泄露。在存储方面，通过冗余等方法，加强节点的可信度。

总体来讲，在不可信的基础上构建可信的服务

友情提示：
这一个问题 ppt 没有讲，也没有查到靠谱的答案。看来我得回去练一下百度的功力~~

七、云计算的价值何在？

1. 帮助用户降低 IT 服务成本：
   （1） 云计算服务的构建方式能够充分利用 IT 资源，并大量采用自动化的手段降低管理成本。具有相当好的规模效应.
   （2） 通过运用云计算的技术，可以让用户实现均衡的、优化的硬件资源管理，从而使企业不再需要进行过度的投资。
2. 提升用户体验：云计算使他们从底层复杂的技术细节中解放了出来，可以把注意力放在相对高级、更靠近业务的工作中去。
3. 绿色环保：云计算可以极大地提高资源利用率，有效降低能源消耗。通过资源的整合，可以减少许多重复的 IT 建设。

终于写完了~咩哈哈~各位考试加油~