1、什么是云计算

云计算=云+计算，云计算是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。

2、云计算的优势

优化产业布局

推进专业分工

提升资源利用率

减少初期投资

降低管理开销

3、云计算的动因

芯片和硬件技术

资源虚拟化

面向服务的架构soa

软件即服务saas

互联网技术

web2.0技术

4、云服务的基本层次

基础设施层（iaas），平台层（paas），应用层（saas）

5、IaaS的基本功能

– 资源抽象 – 资源监控 – 负载管理 – 数据管理 – 资源部署 – 安全管理 – 计费管理

6、PaaS的基本功能

– 开发平台– 运行时环境 – 运营环境

7、虚拟化的概念

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。

8、服务器虚拟化的特性

– 多实例

– 隔离性

– 封装性

– 高性能

9、服务器虚拟化的关键技术

– 计算虚拟化

– 存储虚拟化  
– 设备与I/O虚拟化

– 实时迁移技术

10、创建虚拟化解决方案的步骤（没有了）

– 创建基本虚拟镜像

– 创建虚拟器件镜像

– 发布虚拟器件镜像

– 管理虚拟器件镜像

– 迁移到虚拟化环境

11、其他虚拟化技术

• 网络虚拟化

• 存储虚拟化

• 桌面虚拟化

• 应用虚拟化

• 容器虚拟化

• 代表性虚拟机

11、典型虚拟机

Xen

KVM

12、虚拟化与云计算的关系

虚拟化是云计算的基础，云计算所用的软硬件资源多是通过虚拟化实现的。

13、大规模数据存储面临的新问题与挑战

成本、容量、可靠、使用问题

14、GFS体系结构

master，管理节点,逻辑上只 有一个,保存系统元数据

Chunk server，数据存储节点,大小 固定(**64MB**)

Client，给应用程序的库文 件形式访问接口

15、GFS的容错机制

– Master容错

– Chunk Server容错

16、Paxos协议

分布式环境下保持一致性的协议。

17、Chubby锁机制

Chubby系统本质上就是一个分布式的、存 储大量小文件的文件系统

– Chubby中的锁就是文件

– 在GFS的例子中,创建文件就是进行“加锁” 操作,创建文件成功的那个server其实就是抢 占到了“锁”

– 用户通过打开、关闭和存取文件,获取共享锁 或者独占锁;并且通过通信机制,向用户发送 更新信息

18、Chubby的通信协议

操作的允许授权是通过限时的、倒计时“租期”系统来处理的

19、Bigtable数据结构

数据模型  
– 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

– Bigtable对存储的数据不做解析,看做字符串

– 具体数据结构实现需要用户自行处理

– 存储逻辑(row:string, column:string, time:int64)→string

20、Bigtable优化机制

– 局部性群组

– 布隆过滤器

– 合并压缩

21、云存储应用的特点

– 通用的设备支持(云存储的浏览端)

– 数据同步与共享

– 任意格式/大小文件

– 免费+付费

22、Mapreduce算法的架构

– 源文件:GFS– Map处理结果:本地存储 – Reduce处理结果:GFS– 日志:GFS

23、Wordcount算例

24、Hadoop执行MR的过程

– Map  
• 从磁盘上读数据  
• 执行map函数  
• Combine结果  
• 将结果写到本地磁盘上

– Reduce  
• 从各个map task的磁盘上读相应的数据(shuffle)

* 排序sort
* 执行reduce函数

• 将结果写回HDFS

25、MR算法执行过程中的数据流转过程

26、Mapreduce算法设计思想

27、运用mapreduce算法解决实际问题

28、算法调优

– 更加优化的key-value对设置

– Map算法

– Combiner算法

– Partition算法

– Reduce算法

29、Mapreduce运行过程中的各种参数及其作用

map函数：缓冲区大小100MB,阈值0.8

io.sort.factor：10；

combiner溢出写次数最小值：min.num.spills.for.combine,默认3

Tasktracker的工作线程数:tracker.http.threads,默认 40

reduce任务复制线程:mapred.reduce.parallel.copies, 默认5

30、参数调优

1）mapper的数量：尽量将输入数据切分成数据块的整数倍。如有太多小文件，则考虑

      CombineFileInputFormat;

   2）reducer的数量：为了达到最高性能，集群中reducer数应该略小于reducer的任务槽数。

   3）combiner: 充分使用合并函数减少map和reduce之间传递的数据量，combiner在map后运行；

   4）中间值的压缩：对map输出值进行压缩减少到reduce前的传递量（conf.setCompressMapOutput(true)和

      setMapOutputCompressorClass(GzipCodec.class)）；

   5）自定义序列：如果使用自定义的Writable对象或自定义的comparator，则必须确保已实现RawComparator

31、案例的mapreduce算法

32、AWS模式是什么,有什么优点?

模式：

以web service(API)管理一切服务

通过认证、授权区分用户

完全的SOA

IaaS标准

成功的商业模式

构建了完整的云计算生态系统

优点：

通过Web Service接口开放数据和功能

一切以服务实现

通过SOA的架构使系统达到松耦合

33、IaaS模式核心需求有哪些?

计算EC2、存储S3/EBS、网络服务VPC/ELB等以及数据库服务

34、Openstack都包含哪些核心项目,作用是什么?

•Openstackcompute (nova)

Nova是云组织的控制器

•Openstackojbectstorage (swift)

swift是一个可扩展的对象存储系统

•Image service (glance) 镜像

IaaS中的核心概念

计算、存储服务的载体

•Identity (keystone)

keystone的核心功能是为openstack系统提供统一的用户标识服务，使用户可以方便的接入openstack的其它服务（etc：compute，volume，glance），同时作为一个通用的云OS验证系统可以和已有的后端用户目录服务整合

•Dashboard (horizon)

为所有openstack的服务提供了一个模块化的web-based用户界面。

•Network connectivity (quantum)

在接口设备之间提供“网络连接作为一服务”。

•Block storage (cinder)

块存储，可以挂载在实例上，为实例提供稳定的数据块存储服务

35、镜像和实例有什么区别和联系?

Image：固定搭配

Instance：固定搭配的一个实例

Image：便于大量instance的管理

36、Nova有哪些核心模块,工作过程是什么?

37、Keystone权限控制过程是什么?

38、Quantum原理是什么?

39、Swift的核心概念有哪些?

Object：对象。基本的存储实体，所有数据按照对象进行存储

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，object存储于container中

Account：账户。权限单位，account拥有若干container。

40、Swift的组件有哪些,都有什么作用?

Ring 环

Replication（副本）

Updaters（更新器）

Auditors（校验器）

41、Ring算法思想是什么?

环是为了将虚拟节点（partition，分区）均衡地映射到一组物理存储设备上，

并提供一定的冗余度而设计的，集群中发生存储节点宕机、新增（删）存储节点、新增（删）zone等必须改变partition和node间的映射关系时，还可以对Ring文件通过重新平衡（rebalance）来进行更新。

当虚节点需要移动时，环会确保一次移动最少数量的虚节点数，并且一次只移动一个虚节点的一个副本。

42、Quorum协议的内容是什么?

43、HDFS的体系结构

NameNode Master

DataNode Chunksever

HDFS采用了主从（Master/Slave）结构模型，一个HDFS集群是由一个NameNode和若干个DataNode组成的。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；集群中的DataNode管理存储的数据。HDFS允许用户以文件的形式存储数据。从内部来看，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执行文件系统的命名空间操作，比如打开、关闭、重命名文件或目录等，它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请求，并在NameNode的统一调度下进行数据块的创建、删除和复制工作。

44、HDFS的运行机制

46、Hbase的运行机制

47、ZooKeeper的数据读写机制

ZooKeeper是一个由多个Server组成的集群

一个Leader，多个Follower

每个Server都保存了一份数据副本

全局数据一致

分布式读写

更新请求转发，由Leader实施

48、Yarn对Hadoop的核心改进

首先客户端不变，其调用API 及接口大部分保持兼容，这也是为了对开发使用者透明化，使其不必对原有代码做大的改变，但是原框架中核心的JobTracker和TaskTracker不见了，取而代之的是ResourceManager, ApplicationMaster与NodeManager三个部分。

我们来详细解释这三个部分，首先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。细心的读者会发现：Job 里面所在的task 的监控、重启等等内容不见了。这就是AppMst存在的原因。ResourceManager负责作业与资源的调度。接收JobSubmitter提交的作业，按照作业的上下文(Context) 信息，以及从NodeManager收集来的状态信息，启动调度过程，分配一个Container 作为App Mstr

NodeManager功能比较专一，就是负责Container 状态的维护，并向RM 保持心跳。

ApplicationMaster负责一个Job 生命周期内的所有工作，类似老的框架中JobTracker。但注意每一个Job（不是每一种）都有一个ApplicationMaster，它可以运行在ResourceManager以外的机器上。