

## 1. 存储技术分类和特点

- SCSI 小型计算机系统接口：作为输入/输出接口，主要用于硬盘光盘、磁带机等设备
- DAS 直连式存储：将存储设备通过 SCSI 接口或光纤通道直接连接到计算机上，不能实现数据与其他主机的共享，占用服务器操作系统资源，如 CPU、IO 等，数据量越大，性能越差
- NAS 网络技术存储：一种专用数据存储服务器，以数据为中心，将存储设备与服务器彻底分离，集中管理数据，从而释放带宽、提高性能、降低总拥有成本、保护投资用户通过 TCP/IP 协议访问数据，采用标准的 NFS/HTTP/CIFS 等
- SAN 存储区域网络：通过光纤交换机、光纤路由器、光纤集线器等设备将磁盘阵列、磁带等存储设备与相关服务器连接起来，形成高速专网网络，组成部分如路由器、光纤交换机接口：如 SCSI、FC，通信协议：如 IP、SCSI，一种适合于千兆数据传输的、成熟而安全解决方案，与传统的 SCSI 相比，FC 提供更高的数据传输速率、更远的传输距离、更多的设备连接支持以及更稳定的性能、更简易的安装
- FC 光纤通道：FC 主要组件，光纤，HBA（主机总线适配器），FC 交换机

## 2. iscsi 优点

基于 IP 协议技术的标准，允许网络在 TCP/IP 协议上传输 SCSI 命令，相对 FC SAN，iSCSI 实现的 IP SAN 投资更低，解决了传输效率、存储容量、兼容性、开放性、安全性等方面的问题，没有距离限制

## 3. iSCSI 操作流程

- Target 端：选择 target 名称，安装 iSCSI target
  - 准备用于 target 的存储
  - 配置 target
  - 启用服务
- Initiator 端
  - 安装 initiator
  - 配置 initiator 并启动服务

## 3. udev 的作用

从内核收到添加/移除硬件事件时，udev 将会分析：

/sys 目录下信息

/etc/udev/rules.d 目录中的规则

基于分析结果，udev 会：

处理设备命名，决定要创建哪些设备文件或链接，决定如何设置属性，决定触发哪些事件

## 4. 什么是多路径，功能是什么

当服务器到某一存储设备有多条路径时，每条路径都会识别为一个单独的设备，多路径允许您将服务器节点和储存阵列间的多个 I/O，路径配置为一个单一设备，这些 I/O 路径是可包含独立电缆、交换器和控制器的实体 SAN 链接，多路径集合了 I/O 路径，并生成由这些集合路径组成的新设备，功能：冗余：主备模式，高可用，改进的性能：主主模式，负载均衡

## 5. 什么是集群

一组通过高速网络互联的计算组，并以单一系统的模式加以管理，将很多服务器集中

起来一起，提供同一种服务，在客户端看来就象是只有一个服务器可以在付出较低成本的情况下获得在性能、可靠性、灵活性方面的相对较高的收益，任务调度是集群系统中的核心技术

## 6. 集群的目的

- 1.提高性能：如计算密集型应用，如：天气预报、核试验模拟
- 2.降低成本：相对百万美元级的超级计算机，价格便宜
- 3.提高可扩展性：只要增加集群节点即可
- 4.增强可靠性：多个节点完成相同功能，避免单点失败

## 7. 集群分类

- 高性能计算集群 HPC
  - 通过以集群开发的并行应用程序，解决复杂的科学问题
- 负载均衡（LB）集群
  - 客户端负载在计算机集群中尽可能平均分摊
- 高可用（HA）集群
  - 避免单点故障，当一个系统发生故障时，可以快速迁移

## 8.LVS 集群组成

- 前端：负载均衡层
  - 由一台或多台负载调度器构成
- 中间：服务器群组层
  - 由一组实际运行应用服务的服务器组成
- 底端：数据共享存储层
  - 提供共享存储空间的存储区域

## 9.LVS 工作模式有哪些

- VS/NAT
  - 通过网络地址转换实现的虚拟服务器
  - 大并发访问时，调度器的性能成为瓶颈
- VS/DR
  - 直接使用路由技术实现虚拟服务器
  - 节点服务器需要配置 VIP，注意 MAC 地址广播
- VS/TUN
  - 通过隧道方式实现虚拟服务器

## 10.负载均衡调度算法有哪些

- LVS 目前实现了 10 种调度算法
- 常用调度算法有 4 种
  - 轮询（Round Robin）将客户端请求平均分发到 Real Server
  - 加权轮询（Weighted Round Robin）根据 Real Server 权重值进行轮询调度
  - 最少连接（Least Connections）选择连接数最少的服务器
  - 加权最少连接（Weighted Least Connections）根据 Real Server 权重值，选择连接数最少的服务器

## 11. 什么是 HAProxy

- 它是免费、快速并且可靠的一种解决方案

- 适用于那些负载特大的 web 站点，这些站点通常又需要会话保持或七层处理
- 提供高可用性、负载均衡以及基于 TCP 和 HTTP 应用的代理

## 12. HAProxy 的工作模式是什么

- mode http
  - 客户端请求被深度分析后再发往服务器
- mode tcp
  - 客户端与服务器之间建立会话，不检查第七层信息
- mode health
  - 仅做健康状态检查，已经不建议使用

## 13. HTTP 事务模型

- HTTP close
  - 客户端向服务器建立一个 TCP 连接
  - 客户端发送请求给服务器
  - 服务器响应客户端请求后即断开连接
  - 如果客户端到服务器的请求不只一个，那么就要不断的去建立连接
  - TCP 三次握手消耗相对较大的系统资源，同时延迟较大
- Keep-alive
  - 一次连接可以传输多个请求
  - 客户端需要知道传输内容的长度，以避免无限期的等待传输结束
  - 降低两个 HTTP 事务间的延迟
  - 需要相对较少的服务器资源
- Pipelining
  - 仍然使用 Keep-alive
  - 在发送后续请求前，不用等前面的请求已经得到回应
  - 适用于有大量图片的页面
  - 降低了多次请求之间的网络延迟

## 14.什么是 Keepalived?

- Keepalived 实现了高可用集群
- Keepalived 最初是为 LVS 设计的，专门监控各服务器节点的状态
- Keepalived 后来加入了 VRRP 功能，防止单点故障

## 15.Keepalived 运行原理

- Keepalived 检测每个服务器节点状态
- 服务器节点异常或工作出现故障，Keepalived 将故障节点从集群系统中剔除
- 故障节点恢复后，Keepalived 再将其加入到集群系统中
- 所有工作自动完成，无需人工干预

## 16. HAProxy 优缺点是什么

- 优点
  - 支持 session、cookie 功能
  - 可以通过 url 进行健康检查
  - 效率、负载均衡速度，高于 Nginx，低于 LVS
  - HAProxy 支持 TCP，可以对 MySQL 进行负载均衡
  - 调度算法丰富
- 缺点

- 正则弱于 Nginx
- 日志依赖于 syslogd, 不支持 apache 日志

## 17. 什么是分布式文件系统

- 分布式文件系统 (Distributed File System) 是指文件系统管理的物理存储资源不一定直接连接在本地节点上, 而是通过计算机网络与节点相连
- 分布式文件系统的设计基于客户机/服务器模式

## 18. 什么是 ceph

- ceph 是一个分布式文件系统
- 具有高扩展、高可用、高性能的特点
- ceph 可以提供对象存储、块存储、文件系统存储
- ceph 可以提供 PB 级别的存储空间
- 软件定义存储(Software Defined Storage)作为存储行业的一大发展趋势, 已经越来越受到市场的认可

## 19. 什么是块存储

- Ceph 块设备也叫做 RADOS 块设备
  - RADOS block device: RBD
- RBD 驱动已经很好的集成在了 Linux 内核中
- RBD 提供了企业功能, 如快照、COW 克隆等等
- RBD 还支持内存缓存, 从而能够大大提高性能
- 单机块设备
  - 光盘
  - 磁盘
- 分布式块存储
  - Ceph
  - Cinder

## 20. 什么是 CephFS

- 分布式文件系统 (Distributed File System) 是指文件系统管理的物理存储资源不一定直接连接在本地节点上, 而是通过计算机网络与节点相连
- CephFS 使用 Ceph 集群提供与 POSIX 兼容的文件系统
- 允许 Linux 直接将 Ceph 存储 mount 到本地

## 21. 什么是元数据

- 元数据 (Metadata)
  - 任何文件系统中的数据分为数据和元数据。
  - 数据是指普通文件中的实际数据
  - 而元数据指用来描述一个文件的特征的系统数据
  - 比如: 访问权限、文件拥有者以及文件数据块的分布信息(inode...)等

## 22. 什么是对象存储

- 对象存储
  - 也就是键值存储, 通其接口指令, 也就是简单的 GET、PUT、DEL 和其他扩展, 向存储服务上传下载数据
  - 对象存储中所有数据都被认为是一个对象, 所以, 任何数据都可以存入对象存储服务, 如图片、视频、音频等

- RGW 全称是 Rados Gateway
- RGW 是 Ceph 对象存储网关，用于向客户端应用呈现存储界面，提供 RESTful API 访问接口

### 23. 什么是 kvm

是 Kernel-based Virtual Machine 的简称，KVM 要求 CPU 支持硬件虚拟化技术（如 intel VT 或 AMD-V），是 [Linux](#) 下的全虚拟化解决方案。KVM 由处于内核态的 KVM 模块和用户态的 QEMU 两部分构成。KVM 虚拟化技术是当前云计算中计算虚拟化的主流技术，是云计算的核心底层能力，使用 KVM 把多台物理机虚拟化后，组成一个大的虚拟资源池，方便用户从资源池中按需分配计算能力，提升资源的使用效率，同时保证多用户资源之间的隔离性、安全性

### 24.云模式有哪些？

- 1.IaaS：基础设施即服务：用户通过网络获取虚机、存储、网络，然后用户根据自己的需求操作获取的资源
- 2.PaaS：平台即服务：将软件研发平台作为一种服务，如 Eclipse/Java 编程平台，服务商提供编程接口/运行平台等
- 3.SaaS：软件即服务：将软件作为一种服务通过网络提供给用户，如 web 的电子邮件、HR 系统、订单管理系统、客户关系系统等。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

### 25.虚拟主机 服务器 云主机 三者的区别

虚拟主机，就是把一台运行在互联网上的服务器划分成多个“虚拟”的服务器，每一个虚拟主机都具有独立的域名和完整的 Internet 服务器(支持 WWW、FTP、E-mail 等)功能。

虚拟主机是使用特殊的软硬件技术，把一台运行在因特网上的服务器主机分成一台台“虚拟”的主机，每一台虚拟主机都具有独立的域名，具有完整的 Internet 服务器(WWW、FTP、Email 等)功能，虚拟主机之间完全独立，并可由用户自行管理，在外界看来，每一台虚拟主机和一台独立的主机完全一样。

云主机是多个服务组合到快然后之间去分割的一个云主机。他可以独立的系统这个跟独立服务器差不多。他们的可以这样来大小关系这样理解：服务器>云主机>虚拟主机。

### 26.什么是 Docker

- Docker 是完整的一套容器管理系统
- Docker 提供了一组命令，让用户更加方便直接地使用容器技术，而无需过多关心底层内核技术

### 27.Docker 优缺点

Docker 的优点

- 相比于传统的虚拟化技术，容器更加简洁高效
- 传统虚拟机需要给每个 VM 安装操作系统
- 容器使用的共享公共库和程序

Docker 的缺点

- 容器的隔离性没有虚拟化强
- 共用 Linux 内核，安全性有先天缺陷
- SELinux 难以驾驭
- 监控容器和容器排错是挑战

## 28.什么是镜像

- 在 Docker 中容器是基于镜像启动的
- 镜像是启动容器的核心
- 镜像采用分层设计
- 使用快照的 COW 技术，确保底层数据不丢失

## 29.说出 docker 常用命令

- dockerimages//查看镜像列表
- dockerhistory//查看镜像制作历史
- dockerinspect//查看镜像底层信息
- dockerpull//下载镜像
- dockerpush//上传镜像
- dockerrmi//删除本地镜像
- dockersave//镜像另存为 tar 包
- dockerload//使用 tar 包导入镜像
- dockersearch//搜索镜像
- dockertag//修改镜像名称和标签

## 30.说出对储存卷的理解

docker 容器不保持任何数据

- 重要数据请使用外部卷存储（数据持久化）
- 容器可以挂载真实机目录或共享存储为卷

### 31.什么是共享存储，举例说明

共享存储基本概念

- 一台共享存储服务器可以提供给所有 Docker 主机使用
- 共享存储服务器（NAS、SAN、DAS 等）
- 如：-使用 NFS 创建共享存储服务器，客户端挂载 NFS 共享，并最终映射到容器中

### 32.Openstack 主要组件都有什么

•Horizon

-用于管理 Openstack 各种服务的、基于 web 的管理接口

-通过图形界面实现创建用户、管理网络、启用实例等操作

•Keystone

-为其他服务提供认证和授权的集中身份管理服务

-也提供了集中的目录服务

-支持多种身份认证模式，如果密码认证、令牌认证、以及 AWS（亚马逊 Web 服务）登陆

-为用户和其他服务提供了 SSO 认证服务

•Neutron

-一种软件定义网络服务

-用于创建网络、子网、路由器、管理浮动 IP 地址

-可以实现虚拟交换机、虚拟路由器

-可用于在项目中创建 VPN

•Cinder

-为虚拟机管理存储卷的服务

-为运行在 Nova 中的实例提供永久的块存储

-可以通过快照进行数据备份

-经常应用在实例存储环境中，如果数据库文件

•Nova

-在节点上用于管理虚拟机的服务

-Nova 是一个分布式的服务，能够不 Keystone 交互实现认证，不 Glance 交互实现镜像管理

-Nova 被设计成在标准硬件上能够进行水平扩展

-启动实例时，如果有需要则下载镜像

- Glance

-扮演虚拟机镜像注册的角色

-允许用户为直接存储拷贝服务器镜像

-这些镜像可以用于新建虚拟机的模板

### 33. 简述什么是 Horizon，它的功能和特点

Horizon 是一个用以管理、控制 OpenStack 服务的 Web 控制面板，也称之为 Dashboard 仪表盘

- 可以管理实例、镜像、创建密钥对，对实例添加卷、操作 Swift 容器等。除此之外，用户还可以在控制面板中使用终端（console）或 VNC 直接访问实例

- 基于 python 的 djangoweb 框架进行开发

#### 功能与特点

- 实例管理：创建、终止实例，查看终端日志，VNC 连接，添加卷等

- 访问不安全管理：创建安全群组，管理密钥对，设置浮动 IP 等

- 偏好设定：对虚拟硬件模板可以进行不同偏好设定

- 镜像管理：编辑或删除镜像

- 用户管理：创建用户等

- 卷管理：创建卷和快照

- 对象存储处理：创建、删除容器和对象

### 34.什么是云主机类型

- 云主机类型就是资源的模板

- 它定义了一台云主机可以使用的资源，如内存大小、磁盘容量和 CPU 核心数等

- Openstack 提供了几个默认的云主机类型

- 管理员还可以自定义云主机类型



### 35. 举例说明云主机类型参数

- Name: 云主机类型名称
- ID: 云主机类型 ID, 系统自动生成一个 UUID
- VCPUs: 虚拟 CPU 数目
- RAM(MB): 内存大小
- Root disk(GB): 外围磁盘大小。如果希望使用本地磁盘, 设置为 0
- 临时磁盘: 第二个外围磁盘
- swap 磁盘: 交换磁盘大小

### 36. Glance 磁盘格式都是什么

- raw: 非结构化磁盘镜像格式
- vhd: VMware、Xen、Microsoft、VirtualBox 等均支持的通用磁盘格式
- vmdk: 另一个通用的磁盘格式
- vdi: VirtualBox 虚拟机和 QEMU 支持磁盘格式
- iso: 光盘数据内容的归档格式
- qcow2: QEMU 支持的磁盘格式。空间自动扩展, 并支持写时复制 copy-on-write

### 37. Openstack 网络工作原理

- 实例被分配到子网中, 以实现网络连通性
- 每个项目可以有一到多个子网
- 在红帽的 Openstack 平台中, OpenStack 网络服务是缺省的网络选项, Nova 网络服务作为备用
- 管理员能够配置丰富的网络, 将其他 Openstack 服务连接到这些网络的接口上
- 每个项目都能拥有多个私有网络, 各个项目的私有网络互相不受干扰

### 38. Openstack 网络类型都有什么

- 项目网络: 项目拥有的网络由 Neutron 提供。网络间采用 VLAN 隔离
- 外部网络: 访问虚拟机实例的流量, 通过外部网络进入。实例需要配置浮动 IP 地址
- 提供商网络: 将实例连接到现有网络, 实现虚拟机实例不外部系统共享同一二层网络

### 39. 浮动 IP 地址的作用

- 浮动 IP 地址用于从外界访问虚拟机实例
- 浮动 IP 地址只能从现有浮动 IP 地址池中分配
- 创建外部网络时, 浮动 IP 地址池被定义

- 虚拟机实例启动后，可以为其关联一个浮动 IP 地址
- 虚拟机实例也可以解除 IP 地址绑定
- 解除绑定后，再绑定时，不保证绑定原来的 IP 地址

#### 40.安全组规则

- 安全组规则定义了如何处理网络访问
- 规则基于网络或协议定义
- 每个规则都有出和入两个方向
- 规则也可以指定 ip 协议版本
- 默认的安全组规则，允许虚拟机实例对外访问，但是阻止所有对虚拟机实例的访问

#### 41.什么是 ansible，实现什么功能

•Ansible 是 2013 年推出的一款 IT 自动化和 DevOps 软件，目前由 Redhat 已签署 Ansible 收购协议。其是基于 Python 研发，揉合了很多老运维工具的优点实现了批量操作系统配置，批量程序的部署，批量运行命令等功能

- ansible 可以实现：
  - 自动化部署 APP
  - 自动化管理配置项
  - 自动化的持续交付
  - 自动化的（AWS）云服务管理

#### 42. ansible 的优缺点以及特性

- ansible 优点
  - 是仅需要 ssh 和 Python 即可使用
  - 无客户端
- ansible 功能强大，模块丰富
- 上手容易门槛低
- 基于 python 开发，做二次开发更容易
- 使用公司比较多，社区活跃
- ansible 缺点
  - 对于几千台、上万台机器的操作，还不清楚性能、效率情况如何，需要进一步了解。
- ansible 特性
  - 模块化设计，调用特定的模块来完成特定任务

- 基于 python 实现—paramiko—PyYAML(半结构化语言)—jinja2
- 其模块支持 JSON 等标准输出格式，可采用任何编程语言重写
- 部署简单
- 主从模式工作
- 支持自定义模块
- 支持 playbook
- 易于使用
- 支持多层部署
- 支持异构 IT 环境

### 43. json 是什么？包含哪些特性？

—json 是 JavaScript 对象表示法，它是一种基于文本，独立于语言的轻量级数据交换格式。

—JSON 中的分隔符限于单引号 '、小括号 ()、中括号 []、大括号 {}、冒号 : 和逗号 ,

•json 特性

—JSON 是纯文本

—JSON 具有"自我描述性"（人类可读）

—JSON 具有层级结构（值中存在值）

—JSON 可通过 JavaScript 进行解析

### 44. yaml 是什么？

—是一个可读性高，用来表达数据序列的格式。

—YAML 参考了其他多种语言，包括：C 语言、Python、Perl，并从 XML、电子邮件的数据格式（RFC 2822）中获得灵感。目前已经有数种编程语言或脚本语言支持（或者说解析）这种语言。

### 45.yaml 中“ - ”，“ : ”，“ ” 缩进 “， “#” 代表什么？

—数组使用“-”来表示

—键值对使用“:”来表示

—YAML 使用一个固定的缩进风格表示数据层级结构关系

—一般每个缩进级别由两个以上空格组成

-# 表示注释

#### 46.如何定义 playbook ?

-playbook 是 ansible 用于配置，部署，和管理托管主机剧本。通过 playbook 的详细描述，执行其中的一系列 tasks，可以让远端主机达到预期的状态。

#### 47.使用 playbook 的好处？

执行一些简单的任务，使用 ad-hoc 命令可以方便的解决问题，但是有时一个设施过于复杂，需要大量的操作时候，执行的 ad-hoc 命令是不适合的，这时最好使用 playbook，就像执行 shell 命令与写 shell 脚本一样，也可以理解为批处理任务

-使用 playbook 你可以方便的重用编写的代码，可以移植到不同的机器上面，像函数一样，最大化的利用代码在使用 Ansible 的过程中，你也会发现，你所处理的大部分操作都是编写 playbook

#### 48. playbook 构成包含哪些？

-Target： 定义将要执行 playbook 的远程主机组

-Variable： 定义 playbook 运行时需要使用的变量

-Tasks： 定义将要在远程主机上执行的任务列表

-Handler： 定义 task 执行完成以后需要调用的任务

#### 49. 说出 playbook 几种排错调试方法？

-检测语法

ansible-playbook --syntax-check playbook.yaml

-测试运行

ansible-playbook -C playbook.yaml

-显示收到影响到主机 --list-hosts

-显示工作的 task --list-tasks

-显示将要运行的 tag --list-tags

#### 50. ELK 是什么？

•ELK 其实并不是一款软件，而是一整套解决方案，是三个软件产品的首字母缩写

-Elasticsearch： 负责日志检索和储存

-Logstash： 负责日志的收集和分析、处理

-Kibana： 负责日志的可视化

-这三款软件都是开源软件，通常是配合使用，而且又先后归于 Elastic.co 公司名下，故被简称为 ELK

#### 51. ELK 的功能？

•ELK 组件在海量日志系统的运维中，可用于解决：

-分布式日志数据集中式查询和管理

-系统监控，包含系统硬件和应用各个组件的监控

-故障排查

-安全信息和事件管理

-报表功能

## 52.描述 Elasticsearch

•ElasticSearch 是一个基于 Lucene 的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于 RESTful API 的 web 接口。

•Elasticsearch 是用 Java 开发的，并作为 Apache 许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便

## 53.Elasticsearch 特点

•主要特点

-实时分析

-分布式实时文件存储，并将每一个字段都编入索引

-文档导向，所有的对象全部是文档

-高可用性，易扩展，支持集群（Cluster）、分片和复制（Shards 和 Replicas）

-接口友好，支持 JSON

## 54.ES 没有什么功能？

•Elasticsearch 没有典型意义的事务。

•Elasticsearch 是一种面向文档的数据库。

•Elasticsearch 没有提供授权和认证特性

## 55.阐述 ES 常用插件

•head 插件：

-它展现 ES 集群的拓扑结构，并且可以通过它来进行索引（Index）和节点（Node）级别的操作

-它提供一组针对集群的查询 API，并将结果以 json 和表格形式返回

-它提供一些快捷菜单，用以展现集群的各种状态

- kopf 插件

- 是一个 Elasticsearch 的管理工具

- 它提供了对 ES 集群操作的 API

- bigdesk 插件

- 是 elasticsearch 的一个集群监控工具

- 可以通过它来查看 es 集群的各种状态，如：cpu、内存使用情况，索引数据、搜索情况，http 连接数等

## 56.kibana 是什么？有哪些特点？

- 数据可视化平台工具

- 特点：

- 灵活的分析 and 可视化平台

- 实时总结和流数据的图表

- 为不同的用户显示直观的界面

- 即时分享和嵌入的仪表板

## 57.什么是大数据以及作用？

- 大数据的定义

- 大数据由巨型数据集组成，这些数据集大小常超出人类在可接受时间下的收集、应用、管理和处理能力。

- 大数据能做什么？

- 把数据集合并后进行分析可得出许多额外的信息和数据关系性，可用来察觉商业趋势、判定研究质量、避免疾病扩散、打击犯罪或测定即时交通路况等；这样的用途正是大型数据集盛行的原因

## 58.大数据的 5V 特性是什么？

- (V)olume(大体量)

即可从数百 TB 到数十数百 PB、甚至 EB 的规模。

- (V)ariety(多样性)

即大数据包括各种格式和形态的数据。

–(V)elocity(时效性)

即很多大数据需要在一定的时间限度下得到及时处理。

–(V)eracity(准确性)

即处理的结果要保证一定的准确性。

–(V)alue(大价值)

即大数据包含很多深度的价值，大数据分析挖掘和利用将带来巨大的商业价值。

## **59.Hadoop 是什么和特点?**

–Hadoop 是一种分析和处理海量数据的软件平台

–Hadoop 是一款开源软件，使用 JAVA 开发

–Hadoop 可以提供一个分布式基础架构

.Hadoop 特点

–高可靠性、高扩展性、高效性、高容错性、低成本

## **60.Hadoop 常用组件有哪些**

•HDFS (Hadoop 分布式文件系统)

•Mapreduce (分布式计算框架)

•Zookeeper (分布式协作服务)

•Hbase (分布式列存数据库)

•Hive (基于 Hadoop 的数据仓库)

•Sqoop (数据同步工具)

•Pig (基于 Hadoop 的数据流系统)

•Mahout (数据挖掘算法库)

•Flume (日志收集工具)

## **61.Hadoop 的部署模式有几种?**

–单机

只需要配置好环境变量即可运行，这个模式一般用来学习和测试 hadoop 的功能

–伪分布式

伪分布式的安装和完全分布式类似，但区别是所有角色安装在一台机器上，使用本地磁盘，一般生产环境都会使用完全分布式，伪分布式一般用来学习和测试方面的功能

- 完全分布式

## 62.什么是分布式文件系统

- 分布式文件系统（Distributed File System）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络不节点相连
- 分布式文件系统的设计基于客户机/服务器模式
- 一个典型的网络可能包括多个供多用户访问的服务器
- 对等特性允许一些系统扮演客户机和服务器的双重角色

## 63.Hadoop 的三大核心组件是什么？

-Hdfs

-Mapreduce

-Yarn

## 64. NFS 网关用途

- 1.用户可以通过操作系统兼容的本地 NFSv3 客户端来阅览 HDFS 文件系统
- 2.用户可以从 HDFS 文件系统下载文档到本地文件系统
- 3.用户可以通过挂载点直接流化数据。支持文件附加，但是不支持随机写

## 65.zookeeper 是什么？

-ZooKeeper 是一个分布式的，开放源码的分布式应用程序协调服务

- ZooKeeper 能干什么哪？

-ZooKeeper 是用来保证数据在集群间的事务性一致

## 66.zookeeper 角色特性

-Leader：

-接受所有 Follower 的提案请求并统一协调发起提案的投票，负责不所有的 Follower 进行内部的数据交换

-Follower：

-直接为客户端服务并参不提案的投票，同时不 Leader 中行数据交换

-Observer：



-直接为客户端服务但并不参与提案的投票，同时也不 Leader 进行数据交换

## **67.zookeeper 角色与选举**

-服务在启动的时候是没有角色的 (LOOKING)

-角色是通过选举产生的

-选举产生一个 leader，剩下的是 follower

-选举 leader 原则：

-集群中超过半数机器投票选择 leader.

-假如集群中拥有 n 台服务器，那么 leader 必须得到  $n/2+1$  台服务器投票

-如果 leader 死亡，从新选举 leader

-如果死亡的机器数量达到一半，集群挂起

-如果无法得到足够的投票数量，就重新发起投票，如果参与投票的机器不足  $n/2+1$  集群停止工作

-如果 follower 死亡过多，剩余机器不足  $n/2+1$  集群也会停止工作

-observer 不计算在投票总设备数量里面

## **68. zookeeper 可伸缩扩展性原理设计是什么**

-leader 所有写相关操作

-follower 读操作不响应 leader 提议

-在 Observer 出现以前，ZooKeeper 的伸缩性由 Follower 来实现，我们可以通过添加 Follower 节点的数量来保证 ZooKeeper 服务的读性能。但是随着 Follower 节点数量的增加，ZooKeeper 服务的写性能受到了影响。

-客户端提交一个请求，若是读请求，则由每台 Server 的本地副本数据库直接响应。若是写请求，需要通过一致性协议 (Zab) 来处理

-Zab 协议规定：来自 Client 的所有写请求，都要转发给 ZK 服务中唯一的 Leader，由 Leader 根据该请求发起一个 Proposal。然后，其他的 Server 对该 Proposal 进行 Vote。最后，Leader 对 Vote 进行收集，当 Vote 数量过半时 Leader 会向所有的 Server 发送一个通知消息。最后，当 Client 所连接的 Server 收到该消息时，会把该操作更新到内存中并对 Client 的写请求做出回应

## **69.kafka 是什么?**

-Kafka 是由 LinkedIn 开发的一个分布式的消息系统

-kafka 是使用 Scala 编写

-kafka 是一种消息中间件

## 70.为什么要使用 kafka

-冗余、提高扩展性、缓冲

-保证顺序，灵活

-异步通信

## 71.kafka 角色和集群结构是什么

-producer: 生产者，负责发布消息

-consumer: 消费者，负责读取处理消息

-topic: 消息的类别

-Partition: 每个 Topic 包含一个或多个 Partition.

-Broker: Kafka 集群包含一个或多个服务器

-Kafka 通过 Zookeeper 管理集群配置，选举 leader

## 72.为什么 NameNode 需要高可用

-NameNode 是 HDFS 的核心配置，HDFS 又是 Hadoop 的核心组件，NameNode 在 Hadoop 集群中至关重要，NameNode 机器宕机，将导致集群不可用，如果 NameNode 数据丢失将导致整个集群的数据丢失，而 NameNode 的数据的更新又比较频繁，实现 NameNode 高可用势在必行

## 73. 说出 NameNode 高可用方案

-为了解决 NameNode 单点故障问题，Hadoop 给出了 HDFS 的高可用 HA 方案：HDFS 通常由两个 NameNode 组成，一个处于 active 状态，另一个处于 standby 状态。Active NameNode 对外提供服务，比如处理来自客户端的 RPC 请求，而 Standby NameNode 则不对外提供服务，仅同步 Active NameNode 的状态，以便能够在它失败时进行切换。

## 74.NameNode 高可用架构是什么

-一个典型的 HA 集群，NameNode 会被配置在两台独立的机器上，在任何时间上，一个 NameNode 处于活动状态，而另一个 NameNode 处于备份状态，活动状态的 NameNode 会响应集群中所有的客户端，备份状态的 NameNode 只是作为一个副本，保证在必要的时候提供一个快速的转移。为了让 Standby Node 不 Active Node 保持同步，这两个 Node 都不单独称为 JNS 的互相独立的进程保持通信 (Journal Nodes)。当 Active Node 上更新了 namespace，它将记录修改日志发送给 JNS 的多数派。Standby nodes 将会从 JNS 中读取这些 edits，并持续关注它们对日志的变更。Standby Node 将日志变更应用在自己的 namespace 中，当 failover 发生时，Standby 将会在提升自己为 Active 之前，确保能够从 JNS 中读取所有的 edits，即在 failover 发生之前 Standby 持有的 namespace 应该与 Active 保持完全同步。

NameNode 更新是很频繁的，为了保持主备数据的一致性，为了支持快速 failover，Standby node 持有集群中 blocks 的最新位置是非常必要的。为了达到这一目的，DataNodes 上需要同时配置这两个 Namenode 的地址，同时和它们都建立心跳链接，并把 block 位置发送给它们，还有一点非常重要，任何时刻，只能有一个 Active NameNode，否则将会导致集群操作的混乱，那么两个 NameNode 将会分别有两种不同的数据状态，可能会导致数据丢失，或者状态异常，这种情况通常称为“split-brain”（脑裂，三节点通讯阻断，即集群中不同的 Datanode 看到了不同的 Active NameNodes）。对于 JNS 而言，任何时候只允许一个 NameNode 作为 writer；在 failover 期间，原来的 Standby Node 将会接管 Active 的所有职能，并负责将 JNS 写入日志记录，这中机制阻止了其他 NameNode 基于处于 Active 状态的问题。