云存储

3.1 传统存储方式

3.1.1 DAS

DAS是一个以服务器为中心的存储结构，由存储设备通过IDE、SCSI等IO总线与特定的主机或服务器连接而成，存储设备只能通过该服务器来访问和控制，其他主机的访问必须经过该服务器的存储和转发，使用过程和使用本机硬盘并无太大差别。

3.1.1.1 安全性

存储安全由服务器承担，被认为是安全性较高的存储方案

3.1.1.2 性能

DAS通过IDE、SCSI等IO总线与服务器相连，所以当客户连接数持续增多时，总线连接方式容易成为潜在的瓶颈，严重时会导致服务器崩溃。而最大的制约来自于服务器本身，当服务器发生故障时，数据将不能访问。另一方面，服务器的CPU和内存总是有限的，在服务器负荷过重时，效率很低，实时性很差。

3.1.1.3 数据可用性

直连存储无法共享，存储空间不能在服务器之间动态分配，因此经常出现的情况是某台服务器的存储空间不足，而其他一些服务器却有大量的存储空间处于闲置状态却无法利用。DAS结构下的数据保护流程相对复杂，如果做网络备份，那么每台服务器都必须单独进行备份，而且所有的数据流都要通过网络传输。如果不做网络备份，那么就要为每台服务器都配一套备份软件和磁带设备，所以说备份流程的复杂度会大大增加。

3.1.1.4 其他

没有独立的操作系统，不能提供跨平台的共享难以满足现代存储应用大容量、高性能、动态可扩展的需求。

3.1.2 NAS

NAS是可以直接连接到网络向用户提供文件级服务的存储设备，这些设备也是网络的一部分，有相应的网络地址，通过TCP/IP协议互联。与DAS相比，NAS最明显的特点就是分离了网络设备中的服务器功能和存储功能。服务器主要负责应用层的程序，而NAS主要来响应相应进程需要的数据，因此在数据相应方面会有延迟。但用户是可以直接通过网络环境来访问NAS设备的数据的，典型的NAS服务器通常会安装运行NFS文件系统的服务端软件和运行CIFS文件系统的服务端软件，分别支持Unix和Windows系统的访问。

3.1.2.1 安全性

由于服务器和数据的交互可能需要经过公共网络，容易产生数据泄露等安全问题。

3.1.2.2 性能

存储性能较低，只适用于较小网络规模或者较低数据流域量的网络数据存储，对于一些较高要求存储环境，NAS是无法胜任的。另一方面，存储只能以文件方式访问，而不能像普通文件系统一样直接访问物理数据块，因此在某些情况下会严重影响系统效率。在数据传输方面，由于存储数据要通过普通数据网络传输，因此易受网络上其它流量的影响，当网络上有其它大数据流量时会严重影响系统性能。

3.1.2.3 数据可用性

在资源的整合和NAS管理方面，NAS只能对单个存储设备的磁盘进行资源的整合，无法跨越不同的NAS设备。换句话说，新增的NAS设备与原有的NAS设备无法集成为一体，这样NAS的扩展性其实是比较差的，且一般文件服务器没有高可用配置，存在单点故障。在数据备份时，NAS需要通过网络备份，这样备份的效率不高，而且备份也会占用宝贵的网络资源，影响客户应用的顺利进行。

3.1.2.4 其他

NAS由于具有操作系统，支持扩平台的文件共享。而且易于安装，使用管理都很方便，实现即插即用。

3.1.3 SAN

SAN是一种利用光纤通道起来的可以在服务器和存储系统之间传输数据的存储网络系统。在SAN网络系统中，存储设备既不像DAS那样直接连接到服务器上，也不像NAS那样连接到网络上，而是所有的存储设备互相连接，存储设备先形成自己的一个存储网络，服务器与这样一个存储网络相连接。与DAS相似，客户端对数据的访问要先经过服务器，这有助于提高数据的安全性。服务器与客户机的数据通信通过SCSI协议而非TCP/IP协议，它的设计和实现途径为它带来了高连接速度和处理能力。

3.1.3.1 安全性

服务器承担存储的安全，安全性高。

3.1.3.2 性能

SAN通过整合各种不同的存储设备形成一个统一的存储池，向用户提供服务，可以很容易的扩展存储容量，可以满足对于大量数据的存储功能。同时存储器之间、服务器之间通过光纤通道连接，具有很高的连接速度和处理能力，性能优势明显。

3.1.3.3 数据可用性

网络用户可以通过不止一台服务器访问存储设备，当一台服务器出现故障时，其他服务器可以接管故障服务器的任务，另一方面，数据的备份不占用网络带宽，速度快且灵活，因此具有高可用性。

3.2 云存储

云存储是在云计算概念上延伸和发展出来的一个新的概念，是指通过集群应用、网格技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。当云计算系统运算和处理的核心是大量数据的存储和管理时，云计算系统中就需要配置大量的存储设备，那么云计算系统就转变成为一个云存储系统，所以云存储是一个以数据存储和管理为核心的云计算系统。

当我们使用某一个独立的存储设备时，我们必须非常清楚这个存储设备是什么型号，什么接口和传输协议等。为了保证数据安全和业务的连续性，我们还需要建立相应的数据备份系统和容灾系统。除此之外，对存储设备进行定期地状态监控、维护、软硬件更新和升级也是必须的。

如果采用云存储，那么上面所提到的一切对使用者来讲都不需要了。云存储系统中的所有设备对使用者来讲都是完全透明的，任何地方的任何一个经过授权的使用者都可以通过一根接入线缆与云存储连接，对云存储进行数据访问。

3.3 云存储和传统存储区别

第一块，读PPT。

第二块，读PPT后补充。传统的存储系统由于没有采用分布式的文件系统，无法将所有访问压力平均分配到多个存储节点，因而在存储系统与计算系统之间存在着明显的传输瓶颈，由此而带来单点故障等多种后续问题，而存储云正可以解决这一问题。

第三块，读PPT后补充。

3.4 按服务对象分类

看PPT读。

3.5 按服务内容分类

公共云存储：供应商可以保持每个客户的存储、应用都是独立的，私有的。比较著名的公有云存储提供商有百度云盘、华为网盘、新浪微盘、腾讯微云等。

内部云存储。这种云存储和私有云存储比较类似，唯一的不同点是它仍然位于企业防火墙内部。

混合云存储。这种云存储把公共云和私有云/内部云结合在一起。主要用于按客户要求的访问，特别是需要临时配置容量的时候。从公共云上划出一部分容量配置一种私有或内部云可以帮助公司面对迅速增长的负载波动或高峰时很有帮助。尽管如此，混合云存储带来了跨公共云和私有云分配应用的复杂性。

3.6 用户的顾虑

从功能实现上来讲，异地文件存取与文件分享共步技术早在互联网形成之初就已经得到应用，上个世纪互联网刚刚进入国内时就有厂商提供过网盘服务，当时所谓的网盘并不是大家所熟知的网络虚拟磁盘，经过十多年的发展以后，融入了移动互联网营销理念与新技术的“网盘”被包装成了“云存储”，现在每天都有数以亿计的用户正在向自己云存储空间中上传下载着各种文件。用户在越来越云存储同时也存在某些隐忧。

3.6.1 数据安全

用户的操作安全：大多数的云存储都设计了多客户端数据同步机制，一般以最后一次更新为标准，其他客户端开启时自动同步。所以当任一客户端更新内容之后，其他所有客户端都会更新，如果想再回退到之前的版本就不行了。版本管理技术上并不存在问题，但是会加大用户的操作难度，云存储服务商只有少数的私有云提供商有限的支持，多数情况下这种覆盖是时常发生的。

服务端的安全操作：云存储服务器早已经成为了黑客入侵的目标，因为服务器上不仅有无穷用户数据，对此类大用户群服务的劫持更加是黑色收入的重要来源，也就是说服务器的安全性直接影响着用户上传数据的安全，多数的云存储厂商都预备安全防护方案，但是不能忽视的永远人的操作。

3.6.2 个人隐私

有很多移动平台用户喜欢随时将自己用手机拍摄的照片通过云存储快速上传到网盘中。但是上传的每一张照片或其他文件都有可能是云存储的服务端明文保存的。从运维成本上考虑实现私钥加密不太现实，管理员可以从服务端的平台中直接查看和删除用户上传的文件，这些文件中不乏用户的机密文件或用户隐私，现阶段大型服务端都是通过建立严格的制度体系来约束管理人员的职业操守，但是云盘个人隐私泄密事件还是屡有发生。

3.6.3 运营停止

在当下的互联网环境下，对私提供的云存储盈利模式还并不清晰，究竟有多少服务商可以持续永久的提供这种服务，这种服务后期是否收费？是否会因为亏损问题而被迫停止运营，在这种情况已有用户的数据向何处迁移？数据安全由谁负责？这些都云存储服务提供商并没有给出确切。

3.6.4 版权风险

有关版权问题已经大范围的出现在了国内的网盘服务中，一些个人或团体会将以影视音乐为主体的文件通过云存储的客户端上传至网盘中，然后通过分享的方式对圈子内提供下载，大量的有版权的视频音乐被这种特殊盗版方式进行传播，而且这种传播方式暂时属于监管的空白，部分云存储提供商在版权单位的压力下开始限制链接分享的范围，加强文件的过滤。但是这些手段不能从根本上解决云存储中用户上传文件的盗版传播。而要建立起一整套影视文件数字指纹签名检验系统除了庞大的研发的运维成本外，各个利益团体之间的技术标准统一也是短期内难以实现的，但是在问题得到解决之前，此刻这种分享还在进行中，面临侵权问题不仅是用户还有云存储的提供商。

3.7 服务提供商的挑战

读PPT。（上半部分）

3.8 海量数据管理技术

读PPT。

当前云数据领域成熟的产品有：GFS、HBase、Sector/Sphere、Amazon S3、OpenStack Swift、HDFS等。

3.8.1 HDFS

HDFS被设计成适合运行在通用硬件上的分布式文件系统，是一个高度容错性的系统，数据默认保存三个副本，且提供容错机制，副本丢失或宕机可自动恢复。它适合部署在廉价的机器上，能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。HDFS在最开始是作为Apache Nutch搜索引擎项目的基础架构而开发的，是Apache Hadoop Core项目的一部分。

下面简单介绍一下几个重要概念。

数据块：HDFS默认会将文件分割成64M的数据块。然后将数据块按键值对存储在HDFS上，并将键值对的映射存到内存中。如果小文件太多，那内存的负担会很重，所以HDFS不太适合小文件。

元数据节点:元数据节点的作用是管理文件目录结构，接受用户的操作请求，是管理数据节点的。元数据节点维护两套数据，一套是文件目录与数据块之间的关系，另一套是数据块与节点之间的关系。前一套数据是静态的，是存放在磁盘上的，通过fsimage和edits文件来维护 ；后一套数据是动态的，不持久放到到磁盘的，每当集群启动的时候，会自动建立这些信息，所以一般都放在内存中。它是整个文件系统的管理节点，维护着整个文件系统的文件目录树、文件根目录的元信息和每个文件对应的数据块列表。接收用户的操作请求。

数据节点:数据节点的作用是HDFS中真正存数据的。

从元数据节点:由于元数据节点要接受用户的操作请求，必须能够快速响应用户请求，为了保证元数据节点的快速响应给用户，所以将此项工作交给了从元数据节点，所以他也备份一部分fsimage的一部分内容。

3.3 实例：Hadoop文件操作

3.3.1读文件过程

客户端(client)用FileSystem的open()函数打开文件 DistributedFileSystem用RPC调用元数据节点，得到文件的数据块信息。

对于每一个数据块，元数据节点返回保存数据块的数据节点的地址。 DistributedFileSystem返回FSDataInputStream给客户端，用来读取数据。

客户端调用stream的read()函数开始读取数据。 DFSInputStream连接保存此文件第一个数据块的最近的数据节点。

Data从数据节点读到客户端(client)

当此数据块读取完毕时，DFSInputStream关闭和此数据节点的连接，然后连接此文件下一个数据块的最近的数据节点。

当客户端读完数据的时候，调用FSDataInputStream的close函数。 在读取数据的过程中，如果客户端在与数据节点通信出现错误，则尝试连接包含此数据块的下一个数据节点。

失败的数据节点将被记录，以后不再连接。

3.3.2 写文件过程

客户端调用create()来创建文件。

DistributedFileSystem用RPC调用元数据节点，在文件系统的命名空间中创建一个新的文件。

元数据节点首先确定文件原来不存在，并且客户端有创建文件的权限，然后创建新文件。

DistributedFileSystem返回DFSOutputStream，客户端用于写数据。

客户端开始写入数据，DFSOutputStream将数据分成块，写入data queue。

Data queue由Data Streamer读取，并通知元数据节点分配数据节点，用来存储数据块(每块默认复制3块)。分配的数据节点放在一个pipeline里。

Data Streamer将数据块写入pipeline中的第一个数据节点。第一个数据节点将数据块发送给第二个数据节点。第二个数据节点将数据发送给第三个数据节点。

DFSOutputStream为发出去的数据块保存了ack queue，等待pipeline中的数据节点告知数据已经写入成功。

当客户端结束写入数据，则调用stream的close函数。此操作将所有的数据块写入pipeline中的数据节点，并等待ack queue返回成功。最后通知元数据节点写入完毕。