网络存储技术综述

December 31, 2020

Abstract

在网络存储技术中，由网络存储设备提供网络信息系统的信息存取和共享服务，其主要特征体现在超大存储容量、大数据传输率以及高的系统可用性、远程备份、异地容灾等方面。目前，网络存储技术正在成为计算机领域的研究热点，可以说，网络存储将引发继信息处理（如CPU）和信息传输（如Internet）之后IT领域的第三次技术浪潮。如何适应新的存储需求，采用什么技术来突破当前存在的存储服务的瓶颈，是人们普遍关心和迫切需要解决的问题。

本文选择了六种典型网络存储技术：DAS，SAN，HDFS，Ceph，MySQL，MongoDB进行讲解，旨在通过例子来更形象的阐述网络存储的概念与相关技术，同时指出一些在未来可供研究的点。

关键字：网络、存储、数据、传输

1 绪论

1.1 研究背景

随着计算机技术及其相关的各种网络应用的飞速发展，网络进行传输的信息量不断膨胀，对网络存储的需求也日益增多。存储系统不再是计算机系统的附属设备，而成为互联网中与计算和传输设施同等重要的三大基石之一，网络存储已成长为信息化的核心发展领域，并逐渐承担着信息化核心的责任。实际上，信息技术在任何时候都是处理、传输和存储技术的三位一体的完美结合，三者缺一不可。

数据量的迅速增长为我们提出了新的问题和要求，如何确保数据的一致性、安全性和可靠性，如何实现不同数据的集中管理，如何实现网络上的数据集中访问，如何实现不同主机类型的数据访问和保护等等。所有这些都对现有的存储技术提出了挑战，呼唤着新的网络存储技术及其产品的出现，也使得网络存储技术迅速崛起。

同时,网络技术和服务器技术也对数据处理平台的演化产生了重大的影响。随着这两项技术的逐渐成熟,以及对计算机处理能力和相关数据需求的不断增长, 更快、更好的网络存储技术得到了更多的市场驱动。

网络和存储是以两个不同的技术分别发展起来的。存储使用发起方和目标方的概念来表达，在相连的设备之间形成一种主从关系，而网络则更多的是强调连接设备之间的对等关系。存储技术的重点主要在于高效的数据组织和存放，而网络的重点主要在于高效的数据传输。

1.2 网络存储技术的发展现状

在过去的近十年中,商业模式发生了重大的改变。由于计算机和网络技术向更廉价、更有效的方向发展, 早期的“以计算机为中心”的数据处理已经演化为“以网络数据库或云为中心”的模式。

网络存储技术是最近几年IT行业最热门的技术之一。随着计算机技术和网络技术的发

展，越来越多的信息被数据化。海量的数据信息不仅需要能长时间保存, 并且需要能被快速方便地检索。电子商务、电子政务等信息化技术的推广对数据的存储容量、速度以及安全提出了更高的要求。存储技术也从本地存储发展到网络存储。网络存储技术正处于高速发展的阶段。

1.3 论文中主要英文缩写与中文对照表

Table 1: 中英对照表

缩写名称	中文对照
NAS	网络附加存储
SAN	存储区域网络
PDU	协议数据单元
iSCSI	Internet小型计算机系统接口
OLTP	联机事务处理过程
TCP/IP	传输控制协议/网际协议
RAID	磁盘阵列
LAN	局域网
SMB	服务器信息块
NFS	网络文件系统
HDFS	Hadoop分布式文件系统
Ceph	Ceph分布式文件系统
MongoDB	MongoDB数据库

2 网络存储技术架构及分析

2.1 NAS网络文件系统技术架构

NAS，全称Network Attached Storage，是一种特殊的专用数据存储服务器，是可以直接连到网络上向用户提供文件级服务的存储系统。NAS基于LAN按照TCP/IP协议进行通信，以文件IO方式进行数据传输。NAS是从传统的文件服务器发展起来的一种专有系统，它和其它节点一样直接连接到互联网上，可以像网络打印机一样被其它节点共享。NAS技术直接把存储连接到网络上，而不再挂载在服务器后面，给服务器造成负担。

简单来说NAS模型，更像是存储系统不再通过I/O总线附属于某个服务器或客户机，而直接通过网络接口与网络直接相连，由用户通过网络访问，就是简单的把某台计算机文件存储系统直接分离出来连接上LAN，形式比较简单，对外表现也是像是个简单的计算机上的存储系统。

NAS可以有多个设备，比如可以有多个硬盘相连，但是NAS文件器是其中最重要的设备，文件器直接与网络连接，提供给网络中其他设备对文件的操作。

NAS系统的逻辑构成大致分为三个部分：用户界面，协议层以及内核层。用户界面是NAS系统提供给用户用以定制存储结构，安全设置和用户管理策略等的。协议层包括了网络文件系统协议和通信协议，前者是NAS系统所能提供的数据存取服务方式，而通信协议在NAS系统中一般适用TCP/IP协议。此外，需由NAS的内核部分来处理底层数据，内核层包含对应的设备驱动模块，基本网络协议模块和卷管理模块等。

NAS系统通过实现网络文件系统功能的协议实现在网络上的其它远程主机节点上运行，而不是在本地运行，当今主流的NAS协议有NFS协议，SMB/CIFS协议等多种可行协议。通过使用NFS，用户和程序可以像访问本地文件一样访问远端系统上的文件，使得每个计算机节点都能像使用本地资源一样方便的使用网络上的资源。SMB协议是局域网上用于服务器文件访问和打印的协议。CIFS是其公共和开放的协议版本，CIFS使用客户/服务器模型，其工作原理是让CIFS协议运行于TCP/IP通信协议之上，客户端请求远程服务器上的服务器程序为它提供服务，服务器获得请求并返回响应。

SMB协议：SMB协议是Windows平台标准文件共享协议,Linux平台通过samba来支持。

运行方式：这个SMB协议运行比较灵活，SMB首先有两种运行方式：

会话层上：可以运行于NetBIOS api会话层服务上，其中NTB是一个传输层之上的一个会话层协议或者说是一个API用以提供会话层服务，而NTB的传输层接口使用的是UDP的137和138端口以及TCP的137和139的

端口。(也可以不用NTB协议而用其他的NBF, IPX/SPX等传统协议)

传输层上：可以直接运行在TCP/IP传输层的445接口上。

SMB有以下几个特点：SMB使用点对点的通讯方式，一个客户端向一个服务器提出请求，服务器相应地回答。SMB协议中的一部分专门用来处理对文件系统的访问，使得客户端可以访问一个文件服务器。SMB也有进程间通讯的部分。SMB协议尤其适用于局部子网，但是也可以被用来通过万维网来链接不同的子网。Microsoft Windows的文件和打印机分享主要使用这个功能。

Samba是为了使得Unix和Linux操作系统的计算机支持SMB协议而开发的一个软件，安装了Samba后，不同操作系统间的PC便可以通过SMB协议而共享文件。

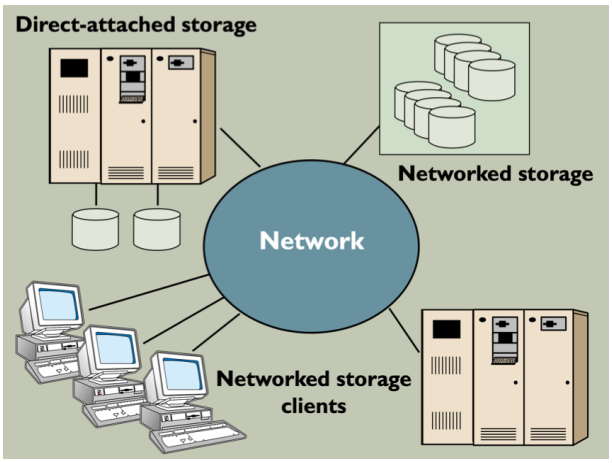

Figure 1: DAS网络存储架构

2.2 NAS系统评价及应用

NAS系统本质是一个存储管理系统，其核心是管理功能，协调和管理系统资源和共享服务，具体的共享等业务需要其他外部系统的支持，而外部系统的实现主要是依靠开源软件，而开源软件具有廉价，源代码可获取，与标准兼容，应用成熟，可靠稳定，社区资源丰富等诸多优点，适用于构建高性价比的NAS系统。NAS系统是基于开源软件的廉价且实用系统。

与SAN相比较，网络储存设备NAS使用的是基于文件的通信协议，另外，与SAN的基于块设备的共享模式不同，NAS实现了基于文件系统的共享模式。传统的NAS是在DAS设备上增加一个NAS头（或NAS网关），对外提供文件共享服务。而目前的NAS趋势是同意存储（结合NAS和SAN的特点），在SAN存储的基础上添加了NAS头（或NAS网关），对外提供NAS共享，又提供SAN共享，可以灵活的根据应用类型来配置不同的解决方案。

NAS设备上面的操作系统和软件只提供了资料存储、资料访问、以及相关的管理功能，并得以使得设备连上网络才进行远程访问；此外，NAS设备也提供了不只一种文件传输协议。NAS系统通常有一个以上的硬盘，而且和传统的文件服务器一样，通常会把它们组成RAID来提供服务，让资料更不会丢失；有了NAS以后，网络上的其他服务器就可以不必再兼任文件服务器的功能。NAS的型式很多样化，可以是一个大量生产的嵌入式设备，也可以在一般的电脑上运行NAS的软件。此外，NAS从两方面改善了数据的可用性：

首先，即使相应的应用服务器不再工作了，仍然可以读出数据。NAS让资料的使用率提升，主要的原因在于资料无需依附在网络服务器上，用户不会因为服务器死机、常态维修或是关闭而无法使用资料，因为用户可以直连NAS上的系统。

此外，NAS这样的简易服务器可以长时间不维护运作，本身不会崩溃，因为它避免了引起服务器崩溃的首要原因，即过于复杂的应用软件引起的问题。

基于NAS的产品具有如下的优点：

1. NAS可以根据需要连接到网络的任何位置，一般部署在其访问频率最高的本地网段，使NAS最靠近数据存取需求最多的用户，减小主干网的网络流量，从而更有效地节省网络带宽等资源。

2. 由于NAS设备专为文件共享功能设计，不需要键盘，显示器和光驱等部件，其价格比通用服务器更便宜；在次盘驱动器的选择方面，除在服务器领域中使用较多的SCSI外，NAS设备还支持性价比更高的IDE磁盘，进一步降低NAS的总体成本。

3. NAS产品是真正即插即用产品。NAS设备一般支持多电脑平台，用户通过网络支持协议可进入相同的文档，因而NAS设备无需改造即可用于混合UNIX／Windows NT局域网内。成熟的NAS产品，也让资料管理变得轻松及简单，让原本需要在服务器上进行的繁复设置程序，简化成几个步骤就可完成，大大的节省设置时间。

4. NAS中的一个文件可以很容易的被不同操作系统平台下的多个客户端共享。

5. NAS设备的物理位置同样是灵活的，它们可放置在工作组内，靠近数据中心的应用服务器，或者也可放在其他地点，通过物理链路与网络连接起来，进行异地的安全备份。无需应用服务器的干预，NAS设备允许用户在网络上直接存储数据，这样既可减小CPU的开销，也能显著改善网络的性能。

6. 易于维护，在需要增加存储空间时，只需在网络上添加新的NAS设备即可，不会对网络中的其它任何节点产生影响。

7. NAS具有广泛的适用性，因为支持基于TCP/IP协议以及标准的多种文件传输协议，可以适应复杂的网络环境。

8. 充分利用现有的LAN网络结构。

NAS本身具有的这些优势，决定了作为信息存储设备可以应用到各个行业和各个领域。

在云备份场景下，NAS可以轻松地实现将用户数据保存备份到云端，做到海量的数据高并发处理，易管理以易扩展的同时利于数据文件的在不同客户端间的共享。

如在大型企业中的应用：大型的NAS系统能在高容量和高流量的环境下支持几百个用户，这样的系统通常都包括专门的硬件和软件，以支持大量的用户和流量，NAS通过采用RAID技术、优化的文件系统、协议处理和完备的冗余硬件，能支持多达几百TB数量级的存储容量;它们可以提供一个或多个千兆或者万兆网络接口进行以太网连接。

又如NAS系统在网站中的应用：Web服务是Internet中最为重要的应用，它是实现信息发布、资料查询、数据处理和视频点等应用的基本平台;复杂的多媒体网站单纯依靠通用服务器的存储容量是不行的，并且随着竞争激烈，其信息存储和用户访问需要不受地理位置的限制；为了保证数据的安全，这些数据和应用也必需有备份服务的支持；Web应用需要大量数据，NAS的大容量的网络存储空间可满足存储容量的需求；NAS存储设备有着简单的管理模式，而且存取系统支持多用户和异构系统的数据共享，能集中管理数据并拥有完善的数据保护措施加之NAS系统构建的成本和维护费用较低，扩展性不错，所以在网站存储方面发挥着巨大的作用。

在数据整合场景下，对于数据分散、孤岛现象严重的情况，NAS能将进行数据集中整合的优势发挥到最好，从而解决设备种类多，管理复杂的难题。

2.3 SAN存储区域网络系统技术架构

SAN（存储区域网络）的目的是将数据的存储与处理分离，使网络中的任何主机可以访问网络中的任何一个存储设备，从而实现数据共享。SAN是一个由存储子系统组成的独立网络，将存储与服务器分离。与基于服务器的存储不同的是，SAN中的节点数可以按实际需求增长，数据能在任意服务器与任意存储之间共享。另外，SAN为存储器和主机提供了专用的数据通道，目前的SAN主要基于光纤通道。SAN系统通常包括服务器，外部存储设备，服务器适配器，集线器，交换机以及网络，存储管理工具等，是一种类似于普通局域网的高速存储网络。

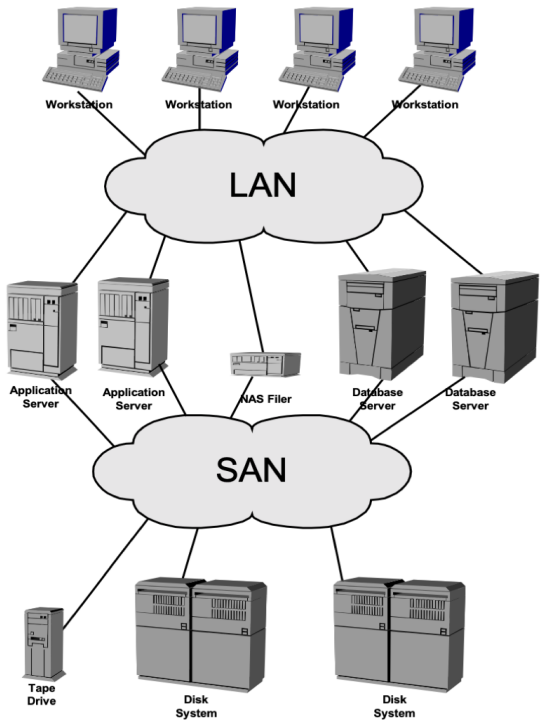

Figure 2: SAN网络存储架构

SAN（存储区域网络）是一种连接外接存储设备和服务器的架构。人们采用包括光纤通道技术、磁盘阵列、磁带柜、光盘柜的各种技术进行实现。该架构的特点是，连接到服务器的存储设备，将被操作系统视为直接连接的存储设备。除针对大型企业的企业级存储方案外，随着在2000年后价格和复杂度的降低，越来越多的中小型企业也在逐步采用该项技术。

SAN存储系统是建立在光纤通道FC（Fibre Channel）技术上的，FC是ANSI为网络和通道I/O接口建立的一个标准集成。SAN之所以能提供高性能、高扩展性且灵活的传输，是通过在各主机服务器和存储系统间实现专有连接并提供高效接口而做到的。SAN的核心是一个专用的存储网络，它允许用户把二级存储器及三级存储器合并成为一个集中管理的基础设施，利用网络及存储管理技术创建一个多平台的开放环境，以便共享存储空间。存储设备和服务器都通过专用的网络进行数据传递，然后由服务器与外部网络相连。由于专用网络上可挂接多个服务器，而且每个服务器均可以与外部网络相连，因而存储网络存在多个数据通路，从而解决了单一数据通路所带来的共享带宽问题。基于光纤通道的存储网络与LAN分离，从而使减少了存储服务对LAN带宽的占用，提升了系统性能。

iSCSI是一项标准协议，它将SCSI命令和块状数据封装到TCP/IP包中来发送接收。我们在前面说FC-SAN的实现一般是基于光纤通信网络的，光纤网络由于和我们生活的网络有很大不同，所以大部分都不太熟悉，而且光纤成本是比较高的。而且由于10G以太网的出现，就使得我们考虑IP搭建SAN的可能，由此iSCSI则是IP-SAN的协议基础。而iSCSI则是SCSI的传输层协议，就是用现有的IP网络的技术来解决SAN的实用性问题，可以说是把两方面结合了起来。

大多数SAN使用SCSI接口进行服务器和磁盘驱动器设备之间的通信。因为它们的总线拓扑结构并不适用于网络环境，所以它们并没有使用底层物理连接介质（比如连接电缆）。相对地，它们采用其它底层通信协议作为镜像层来实现网络连接。

上面说到大多数SAN使用SCSI接口进行通信，这里的SCSI指的是iSCSI，也称为SCSI over TCP/IP，正如上面所描述的，它是建立在TCP/IP协议上的端到端网络存储协议。iSCSI运行在服务器和协议传输网关设备之间，使用标准的以太网交换机或路由器，并在路由器和存储设备之间传输数据。iSCSI服务器发出的SCSI命令被封装成iSCSI协议数据单元（PDU），使用TCP协议作为底层传输层，以进行可靠，顺序的报文传递；一旦iSCSI PDU加上了TCP/IP报头，就和其它IP报文一样被路由和转发。当封装后的SCSI命令通过标准IP网传到目的地后，再把报头一层层剥去，最后把SCSI命令传送给目标存储设备进行处理。SCSI协议本质上同传输介质无关，SCSI可以在多种介质上实现，

甚至是虚拟介质。

2.4 SAN系统评价及应用

SAN提供了一个开放的，可延伸的平台来实现数据密集型环境中的存储访问，应用环境包括数据仓库，联机事务处理（OLTP），服务器集群以及存储管理等。使用光纤通道可在长达10公里的距离内连接存储设备。SAN以其良好的可靠性，可用性，可扩展性，快速数据访问能力，分布式存储架构，高速备份等优势迅速占领存储市场。

SAN在综合了网络的灵活性，可管理性及可扩展性的同时，提高了网络的带宽和存储IO的可靠性，降低了存储管理费用，并平衡了开放式系统服务器的存储能力和性能。SAN独立于应用服务器网络之外，拥有几乎无限的存储能力，以高速光纤通道作为传输媒体，FC+SCSI作为存储访问协议，将存储系统网络化，实现了真正的共享网络存储。

SAN也存在一些不足，在早期发展的时候，有一个问题是不同硬件厂商的交换机并不完全兼容。尽管基本的FCP存储协议总是兼容标准的，但是一些上层的功能却无法完成很好的互操作。与此类似的还有许多主机的操作系统，它们也会在共享某些光纤网络时候产生不良反应。在技术标准最终确定之前，市场上曾经出现了许多解决兼容性的方案，这些创新也都为标准制定提供了帮助。

SAN通常被用在大型的、高性能的企业存储操作中。通常我们不会见到只有一个磁盘驱动器的SAN，相反地，SAN通常都是链接了数个大型磁盘阵列的存储网络。因为SAN设备通常都是比较昂贵的，所以在台式机计算上，光纤通道总线适配器是比较罕见的。基于iSCSI的SAN技术曾经被寄望成为相对便宜的SAN方案，但最终它仍然没有走出企业级的大型数据中心环境。目前大多数的桌面计算机依然使用NAS协议的技术，比如CIFS和NFS。

2.5 HDFS技术架构

与前面的介绍的NAS与SAN集中式系统不同，HDFS是一个分布式系统。HDFS全称为Hadoop数据存储系统，Hadoop实现了一个分布式系统。HDFS采用了主/从架构来管理文件系统。

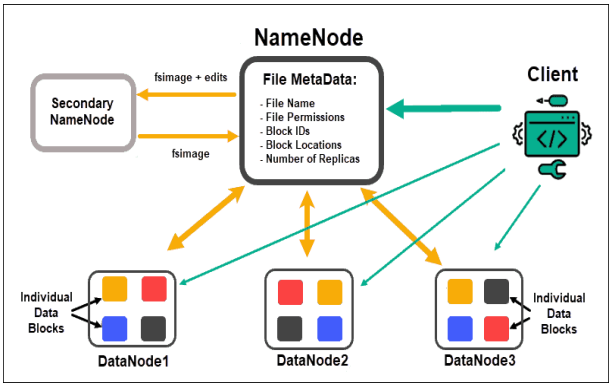

Figure 3: HDFS架构图

如图3所示，一个HDFS集群有一个名称节点(NameNode)，它是HDFS主从节点中的唯一的主节点，块复制操作都是通过NameNode中转，由NameNode代理实现。整个Hadoop集群中只有一个NameNode。

HDFS作为一个分布式文件系统，数据需要保存在多个系统上，为了保证数据的正确性，HDFS会为每个数据存储副本,在副本存放上使用了机架感知的策略尽量将副本放在不同块上来实现提高性能和可靠性。这些存放真实数据的服务器，叫做DataNode。DataNode负责管理文件系统命名空间(NameNode)和客户端节点对文件的访问以及集群中的元数据，元数据主要包括文件系统的命名空间和数据块的存储位置等。

集群中的数据节点通常是一个节点运行一个数据节点进程，负责管理它所在节点上的数据存储。HDFS系统公开了文件系统的名称空间，用户可以以文件的形式在上面存储数据。从内部看，一个文件实际上被分为一个或多个数据块，存储在一组数据范点上。名称节点负责执行文件系统的名称空间操作，同时也负责确定数据块到具体数据节点的

映射。数据节点负责处理文件系统客户端的读/写请求，并在名称节点的调度下进行数据块的创建、删除和复制操作。

由于NameNode很重要，我们需要对它进行备份，所以namenode会定期与secondarynode通信完成备份。

可以看到图中还有Client，Client通过特定的接口若NFS，HDFShttp等远程方法来完成访问。

一个典型的HDFS集群包括一个名称节点和多个数据节点名称节点负责维护命名空间:数据节负责数据块的存储和维护。一个数据块可同时备份在多个数据节点中，一个数据节点中最多只能包含该数据块的一个备份。可以简单的认为，数据节点上存储了数据块ID、数据块内容及两者之间的映射关系。

HDFS 存储系统被设计用来在大规模的廉价服务器集群上可靠地存储大规模数据，并提供高吞吐的数据读取和追加式写入。单个HDFS集群可以扩展至几千甚至上万个节点。HDFS 将所存储的文件划分为较大的数据块(data block，如128MB)，并将这些数据块分布式地存储于集群中的各个节点上。

集群通常有着多台不同的计算机，还有可能运行在不同的机架上，对于两个不同机架上的节点，若是在同一个网段下，采用交换机进行连接。在一般情况下，相同机架上节点的网络带宽远远优于处于在不同机架上的节点带宽集群中运行NameNode进程的节点称为名称节点服务器，它是整个文件系统命名空间的管理者，其中维护了很多数据结构。现阶段的HDFS集群中只有一个名称节点服务器(除此之外，一般集群中会有一个Secondary NameNode节点)。HDFS系统中的数据节点可能有成百上千个，每个数据节点定期和名称节点通信，并执行名称节点返回的指令。为了使名称节点不至于负担过大，名称节点不永久保存数据节点上的数据块信息，而是在HDFS集群系统启动时，通过数据节点的上报信息来更新名称节点上数据块与数据节点间的映射表。名称节点中存在两个至关重要的映射表，文件名同对应数据块之间的映射表和数据块与存储数据块的数据节点之间的映射表，这两个表中存储了整个文件系统的元数据信息。文件名数据块映射表负责记录并维护文件名和与之对应的数据块之间的映射关系，为整个文件系统目录的命名空间，存储在名称节点所在的硬盘之上，为需要持久化存储的元数据信息。由于数据块同时拥有多个数据副本，因此每个数据块都要由一个数据节点列表与其对应。数据块一数据节点映射表便是记录数据块鱼数据节点的映射信息。但是与“文件名一数据节点”映射表不同的是，该表并不持久存储在名称节点的硬盘上而是存储在内存中，每次HDFS系统启动的时候，名称节点根据数据节点反馈过来的块报告信息在名称节点内存中重建该表。

数据节点和名称节点建立连接后，会持续地和名称节点保持心跳。心跳返回的消息包含名称节点对数据节点的指令，如删除数据块或把数据块复制到另一个数据节点。名称节点从不主动发起到数据节点的请求，在通信过程中，两者是严格的CS架构。当然，数据节点也可以作为服务器接受来自客户端的请求，处理数据块的读写操作。数据节点间也会互相通信，执行数据块的复制操作，在客户端执行写操作的时候，数据节点间会相互配合，保证写操作的一致性。

2.6 HDFS系统评价及应用

HDFS(Hadoop distributed file system)作为面向数据追加和读取优化的开源分布式文件系统，具备可移植、高容错和可大规模水平扩展的特性。经过10 余年的发展，HDFS 已经广泛应用于大数据的存储。作为存储海量数据的底层平台，HDFS 存储了海量的结构化和非结构化数据，支撑着复杂查询分析、交互式分析、详单查询、Key-Value读写和迭代计算等丰富的应用场景

HDFS作为一种分布式文件系统，具有高容错的特点，被设计成可以部署在低成本的硬件设备上。它提供高吞吐量的应用程序数据访问，适用于拥有大数据集的应用。总的来说，HDFS系统有如下特点：

1 处理超大文件

这里的超大文件指的是TB级的文件，目前在实际应用中，HDFS 已经用来存储管理PB((PeteBytes)级的数据了，在Yahoo ，Hadoop 集群已经扩展到4000个节点，最大应用达到过20000个节点。

2 流式地访问数据

HDFS系统的设计是在“一次写入、多次读取”的数据访问模型的基础上建立的。因此，一旦数据源生成数据集之后，就会被复制到不同的存储节点，然后响应各种数据分析任务的请求一般情况下，分析任务会涉及数据集中大部分数据，即对于HDFS系统中的数据来说，访问整个数据集比访问一条记录的效率更高。

3 高吞吐

由于简单一致性模型和流式访问数据。HDFS能够很好的处理“一次写入，多次读写”的任务。也就是说，一个数据集一旦生成了，就会被复制到不同的存储节点中，然后响应各种各样的数据分析任务请求。在多数情况下，分析任务都会涉及到数据集中的大部分数据。所以，HDFS请求读取整个数据集要比读取一条记录更加高效。

4 运行在低成本的商用机器集群之上

HDFS系统的设计对硬件设备的要求比较低，只需运行于低成本的商用机器集群之上，而不需要昂贵的高可用性设备。使用低成本的商用机会导致集群中节点故障率的升高。因此HDFS系统在设计时充分考虑了系统中数据的可靠性、安全性等问题。

由于上述几点原因，使用HDFS系统处理一些问题时不但没有优势，反而有一些局限性.首先，HDFS系统不适于低延迟的数据访问。如果想要处理一些短时间、低延迟的应用请求，使用HDFS系统并不合适因为HDFS 是设计成用来处理大数据集分析任务的，主要是为了实现高数据吞吐量，这就要求系统以高延迟为代价目前有一些补充的方案，比如使用HBase ，通过上层数据管理项目来尽可能地弥补这个不足。

其次，存储小文件时效率低。HDFS系统使用名称节点(NameNode)管理文件系统的元数据，响应客户端节点的请求及返回文件位置，因此系统能存储的文件数量大小受限于NameNode节点。例如，每个文件、索引目录及块大约占100字节，如果有100万个文件，每个文件占一个块，那么至少要消耗200MB内存，这似乎还可以介绍。但如果有更多文件，那么名称节点的工作压力增大，检索处理元数据的效率会大大降低。

最后，HDFS系统不支持多用户写入及在任意位置对文件进行修改目前在HDFS系统中，一个文件只能有一个写入者，写操作只能在文件末尾完成，即只能进行追加操作。

在大数据Hadoop生态系统不断发展的过程中，HDFS 因其自身的稳定可靠、简单易用、扩展性高等优点，使得越来越多的上层应用和系统将其作为统一的底层存储。HDFS成为了事实上的“数据中心”，其上存储的数据类型和支持的分析负载越来越多元化。

HDFS作为通用的分布式文件系统，强调高可扩展和低成本地提供高可靠的海量数据存储，保证副本的强一致，高吞吐地落地快速生成的数据，以及基于文件存储多样化的数据，包括结构化数据和文档、图等半结构化和非结构化的数据。HDFS的这些特点契合了大数据的大容量、生成快和多类型的特性，未来更可能作为大数据存储和分析的基础设施，支持多模型和多应用的数据存储。另外，随着异构体系结构的大数据平台在工业界逐渐成为常态，新型的硬件设备也开始迅速在企业中得到推广和应用，HDFS的底层硬件平台也将从传统磁盘和低速网络走向异构存储和高速网络连接。在HDFS未来的发展

方向上，目前的研究工作虽然已经取得了一定的进展，但仍然存在很多问题值得深入探讨和研究。

2.7 Ceph分布式存储系统

Ceph是开源的分布式文件系统，基于网络存储，基于RADOS(Reliable Autonomic Distributed Object Store)，通过一系列API将数据以块（block），文件（file）和对象（object）的形式展现。

Ceph生态系统架构可以划分为四部分，分别是：客户端（Clients）；元数据服务器集群Metadata server cluster（mds）；对象存储集群Object storage cluster（osd）：集群监视器Cluster monitors (mon)。

Ceph的生态系统的概念架构如图4所示:

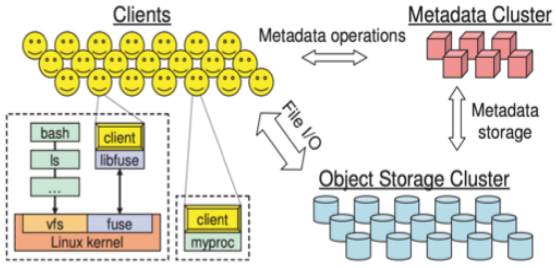

Figure 4: Ceph分布式存储系统架构图

客户端是用户使用Ceph文件系统的入口，元数据服务器mds主要用于缓存和同步分布式元数据，对象存储集群osd将数据和元数据作为对象存储，执行其他关键职能，集群监视器mon执行监视功能。

用户在客户端访问Ceph系统时，只是对文件执行输入输操作，而并不知道该系统是由元数据服务器监视器和对象存储设备等多个节点构成。也就是说这些东西对用户是不可视的。Ceph文件系统的内部结构使用的是对象存储接口，虽然对象存储比传统存储的性能有很大的提高，但是目前对象存储接口并没有得到广泛的应用。为了适应这种情况，系统把RADOS时象存储进行抽象和封装成librados库，在这一基础库的基础上提供抽象上更高层、更便于应用或客户端使用的三种上述接口：块存储接口、对象村粗接口和文件系统接口，客户端可以根据自身的需求来选择适合的接口来访问文件系统。

客户使用元数据服务器，通过执行元数据操作，来确定数据位置，以及元数据服务器管理数据位置，和在何处存储新数据元数据并没有存储在某一固定的存储设备上，而是存储在一个存储集群里，实际的文件IO发生在客户和对象存储集群之间。这样一来，更高层次的POSIX功能(例如，打开、关闭、重命名)就由元数据服务器管理。但是，POSIX功能，如读和写，则直接由对象存储集群管理。

对象存储集群是由许多个对象存储设备（OSD）构成的，Ceph对象存储设备除了拥有传统存储设备的数据存储功能外。还与其集群内的监控节点和其他OSD进行通信和合作，OSD可以是一个磁盘、SSD固态硬盘、RAID阵列或者其他物理存储设备。在该系统中可以轻松地对OSD设备进行动态的添加或删除，从而实现系统的高可扩展性。

集群监视器主要负责监视Ceph集群的状态信息，维护集群的映射关系：如Cluster Map，一种RADOS的关键数据结构，它负责管理，维护集群中的所有成员及它们之间的关系，属性以及数据的分发操作。每当一个用户连接到Ceph想要对数据进行存储或者更新操作，OSD需要先通过监视器，从中获取到最近更新状态的相关数据文件的Map信息，才能根据相关信息分布的计算出数据文件最终存储的位置和结点。

CRUSH(Controlled Replication Under Scalable Hashing)Ceph系统的核心，也是Ceph设计上最先进的部分，该算法是一个高效的伪随机数据分布算法，之所以是伪随数是因为CRUSH实现了一个一致性哈希算法，数据能够根据不同的参数得到固定的存储位置，且这些位置是均匀分布在集群中的。CRUSH算法使得Ceph的存储方式和传统的方法不同，不需

要依赖任何目录或者索引，也无需对元数据进行查询后才能找到数据的位置。

通过CRUSH算法建立的分布式存储系统相比传统分布式系统具有两个明显优势：第一，CRUSH 算法建立的分布式系统是完全分布式的，没有中心节点，任何节点都可以独立计算数据对象的存储位置；第二，极大简化了元数据的设计，所有元数据都是和集群架构相关的(静态元数据，只有集群扩展或者故障才会改变)。因此，可以说CRUSH 算法不仅充分利用了存储资源，也利用了整个集群的计算资源，这是其他分布式存储所不能比拟的。当集群出现故障时，CRUSH也支持多种数据安全备份方式，包括数据快照、奇偶校验、纠删码恢复等。

2.8 Ceph系统评价及应用

Ceph是一种为优秀的性能、可靠性和可扩展性而设计的统一的、分布式文件系统。相比较于传统的分布式存储系统或者分布式文件系统，Ceph具有非常创新的架构设计。Ceph的核心算法CRUSH 数据分布算法帮助Ceph简化了元数据的设计和存储，数据的读和写无需进行查表或者检索索引目录，只需同步CRUSH Map 到客户端，然后客户端就可以根据CRUSH Map 计算出某个数据对象的存储位置，从而直接对该存储节点发出读或者写请求，这样的设计避免了由于中心节点的和数据服务器造成的性能瓶颈。不仅如此，Ceph通过CRUSH 算法能够灵活地管理数据副本，对于部分存储节点的损坏，能够在保证数据正确性的前提下，尽可能地减少数据的迁移。在需要对集群进行扩展时，也能够在不影响集群性能的前提下，尽可能地把数据迁移到新的节点中，进行集群reweight操作，使得数据始终保持均匀地分布在集群的各个存储节点中。

Ceph之所以被称为统一存储系统，是因为Ceph提供了文件系统，块存储和对象存储三种不同的接口，在内部则统一由对象存储对数据进行组织。

Ceph具有高扩展性，使用普通x86服务器，支持10到1000台服务器，支持TP到PB级的扩展，Ceph具有高性能：数据分布均衡，并行化高，对于objects storage和block storage，与HDFS系统相比，Ceph不需要元数据服务器。Ceph具有高可靠性，不会产生单点故障，具有多数据副本，自动管理，自动修复的能力以及非常强的容错能力，即出错恢复能力通常在分布式系统中常见的故障有网络连接失败、磁盘故障、节点死机重启或者电源断电等，Ceph在面对这些故障时能够并行地进行自主修复，无需人工参与为了实现自主修复功能，Ceph引入了Monitor 角色，该角色既可以是人工手动选择也可以由集群自己选择，一般为奇数量的Monitor，这些Monitors共同管理整个集群，其主要手段由三部分构成：故障检测，故障恢复，故障预防。

Ceph也具有一些局限性，使用不同的存储形式会有不同的缺点体现。如IO路径过长。这个问题在Ceph的客户端和服务器端都存在。以osd为例，一个IO需要经过message、OSD、FileJournal、FileStore 多个模块才能完成，每个模块之间都涉及到队列和线程切换，部分模块在对IO 进行处理时还要进行内存拷贝，导致整体性能不高。此外，存储形式为块存储时，如磁盘阵列，硬盘等，主机之间无法共享数据，且采用SAN架构组网时，光纤交换机的造价成本很高。存储形式为文件存储时，如FTP，NFS服务器，虽然造价更低，数据共享容易，但读写速度以及传输速度均不高。

存储形式的不同也导致Ceph可用于不同的场景，如存储形式为块存储时，应用场景有docker容器，虚拟机磁盘存储分配，日志存储等等；存储形式为文件存储时，其应用于日志存储，或有目录结构的文件存储等。存储形式为对象存储时，应用场景为图片存储或视频存储等。

作为分布式文件系统，其能够在维

护POSIX兼容性的同时加入了复制和容错功能。作为Linux的文件系统备选之一，Ceph不仅仅是一个文件系统，还是一个有企业级功能的对象存储生态环境。现在，Ceph已经被集成在主线Linux内核中，但只是被标识为实验性的。在这种状态下的文件系统对测试是有用的，但是对生产环境没有做好准备。但是考虑到Ceph加入到Linux内核的行列，不久的将来，它应该就能用于解决海量存储的需要了。

2.9 MySQL架构

MysQL Cluster是一种允许在无共享架构(Share Nothing Architecture)的系统中应用内存数据库的集群技术。通过这种无共享架构，MySQL集群可以运行在廉价的硬件平台上，而且它对软硬件都无特殊要求。MySQL集群一般采用分布式设计，每个组件都有自己专属的内存和磁盘，因此不存在单点故障。通过不支持任何共享存储方案的冗余设计，使得MySQL集群具有数据的高可用性。

实际上，MySQL Cluster就是把NDB (Network DataBase)存储引擎与标准的MysQL服务器集成在了一起。在术语中，"NDB"特指整个存储引擎体系的一部分，而"MySQL集群" 是指一个或更多的MySQL服务器与NDB存储引擎的组。MySQL集群是由一组被称为宿主机(host)的计算机组成，每台宿主机上运行着一个或多个进程。这些进程，在MySQL集群中也被称为节点，包括MysQL服务器(也叫SQL节点)，数据节点(也叫NDB节点)，管理服务器(也叫管理节点)，可能还有一些其它的数据访问程序。

从MySQL架构图可以看到，外部的应用程序通过SQL层mysqld进程来访问所需的数据，但其并不存储实际元数据，而只维护所有运算必须的逻辑结构。

NDB集群存储层包含了n台服务器作为数据存储的载体，其存储了所有业务数据，并通过统一的对外接口服务，管理层主要由一些独立的节点实现，这些节点通过管理工具套件来协调，监控所有的NDB存储节点。

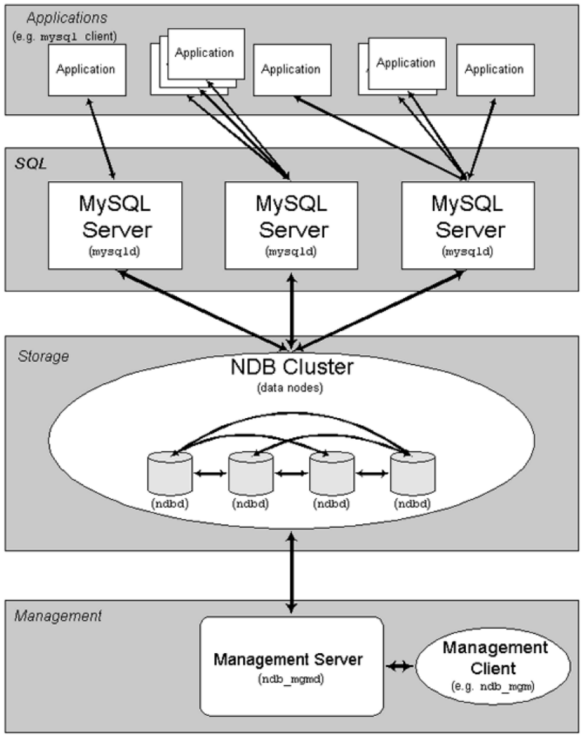

Figure 5: MySQL架构图

在MySQL集群中，包括了三类主要组件：数据节点，SQL节点和管理节点。这些组件之间的关系如图5所示。

从该架构图中我们可以清楚地看到MySQL集群中所有节点的功能，集群中的所有数据都保存在了NDB服务器的存储引擎中，而在MySQL 服务器中则存储着表结构。应用程序要通过SQL节点来访问这些数据表，管理服务器也要通过管理客户端来管理集群中的数据节点。MySQL 集群为它的用户提供了方便高效的数据管理方式。

1、数据节点：数据节点用于保存集群中的数据。在MySQL 集群中，每个数据节点保存着完整数据的一个分片，数据分片(或分区)视节点数目和配置而定。NDB引擎要根据余参数的配置来使用数据节点，再根据数据节点的数目将数据进行分片或分区存储。

2、 sQL节点： SQL节点是用来访问MySQL

集群数据的节点。这类节点负责的是存储层以外事情。对于数据节点来说，可以把采用NDB存储引擎的MySQL 服务器视作其客户端。除了SQL节点之外，客户端应用还可以通过数据节点的

3、管理(MGM)节点：它负责对集群中各节点进行管理，记录并保存着集群环境的详细配置和日志信息。通过它可以实现对各类节点的启动、停止、维护等操作，同时它会把获取的各个节点的状态信息反馈给集群中的所有节点。

MySQL复制（Replication）本身是一个比较简单的架构，即一台从服务器(Slave)从另一台主服务器(Master)读取二进制日志然后再解析并应用到自身。一个最简单复制环境只需要两台运行有MySQL Server的主机即可，甚至可以在同一台物理服务器主mysqld Master Slave 搭建。但是在实际应用环境中，可以根据实际的业务需求利用MySQL 复制的功能自己定制搭建出其他多种更利于横向扩展的复制架构。如主从架构，双主架构等。

主从架构指的是使用一台MySQL服务器作为Master，多台MySQL服务器作为Slave，将Master的数据复制到Slave上。在实际的应用场合中，主从架构模式是MySQL复制最常用的。一般在这种架构下，系统的写操作都在Master中进行，而读操作则分散到各个Slave上，因此此种架构特别合适对于解决目前互联网高读写比的问题。

双主架构是为解决Master端需要进行特殊的维护操作时，系统停机时间过长的问题而提出的。将Slave节点切换成Master来提供写服务。双主架构的两个服务器互为Master和Slave，双方的更改都会通过复制应用到对方。该架构能避免停机之后重新搭建复制环境的操作，因为任意一端都记录了自己当前复制到对方的位置，当系统启动之后，就会自动从之前的位置从新开始复制，而无需人为干预，很大程度上节省了维护成本。

2.10 MySQL集群系统评价及应用

MySQL集群具有如下特点：

1、在性能和可扩展能力方面

(1)自动分片：能够将数据库自动透明地部署于低成本的商用服务器上。

(2)采取多主复制为集群提供了较高的写操作扩展能力。

(3)能够实时响应。

2、在可用性和数据的完整性方面

(1)采用分布式、无共享架构保证集群的高可用性。

(2)无单点故障。

(3)数据同步复制。

(4)自动故障切换。

(5)故障自我修复式恢复。

(6)可进行数据的跨地域复制。

3、在部署灵活性方面

(1)可以部署在虚拟机环境。

(2)可以在内存中管理表或将表存储在磁盘上。

(3)可以在商用硬件间扩展集群，不用共享磁盘。

MySQL复制技术具有操作简洁，实施方便，维护成本低的特点。在网络中实际的应用中，MySQL复制是使用最普遍的一种提高系统可用性的方案。众多的MySQL用户利用复制技术，通过简单的添加经济的PC服务器的办法，得到了系统性能成本甚至指数级别的提升。这也是众多MySQL用户选择MySQL的原因，再加上其操作简单，实施方便，复制功能也成为了业界扩展MySQL集群的最佳方案。

在数据信息量飞速增加，硬件设备难以跟上应用系统对于数据处理能力的需求的今天，MySQL集群方案利用多个单独的数据库系统，采用一定的架构组成一个高性能，高可用的数据库集群系统来解决企业复杂的应用，MySQL作为开源数据库中的佼佼者，以其简单，高效，可靠的特点，早已在IT行业成为一个家喻户晓的数据库系统。小到嵌入

式系统，到中小型的互联网应用，甚至大型的企业应用系统，MySQL无处不在。

2.11 MongoDB数据库系统架构

除了传统的关系型数据库管理系统之外，还有一类数据库系统。NoSQL泛指这样一类数据库和数据存储，他们不遵循经典的数据库架构，且常常与Web规模的大型数据库相关。NoSQL数据库不需要特定的表结构，通常不支持表的连接操作，不支持完整的ACID属性，而且一般拥有强大的可扩展性。MongoDB是一种分布式键值数据库，虽然属于非关系数据库，但却是非关系数据库中功能最丰富，也最像关系数据库的产品。

MongoDB是面向文档的开源的NOSQL数据库系统,用C++语言编写。它提供一种强大、灵活、可扩展的数据存储方式。它扩展了关系型数据库的众多功能如辅助索引，范围查询和排序。MongoDB的功能非常丰富，比如内置的对MapReduce 聚合的支持,以及对地理空间索引的支持。MongoDB支持对数据库的数据进行索引查找，使得时间效率非常高。

MongoDB的核心概念是文档（document），多个键及其相应的值有序的存放在一起组成文档，文档类似于关系型数据库中的元组。多个文档组成集合（collection），集合如同关系型数据库中的表。多个集合组成数据库，一个MongoDB的实例可以承载多个数据库，每个数据库之间是完全独立的。MongoDB的文档采用BSON格式存储，BSON是Binary JSON的简写，是一种类似于JSON文档的二进制序列化方案。用BSON格式来存储数据具有轻量级，容易遍历和高效。同时，在主流的计算机语言如Java，Python中对JSON都有很好的支持，数据从MongoDB中提取之后，无需转换可直接使用。

分片是MongoDB的扩展方式。通过分片能够增加更多的机器来应对不断增加的负载和数据,同时还能不影响应用。分片是指将

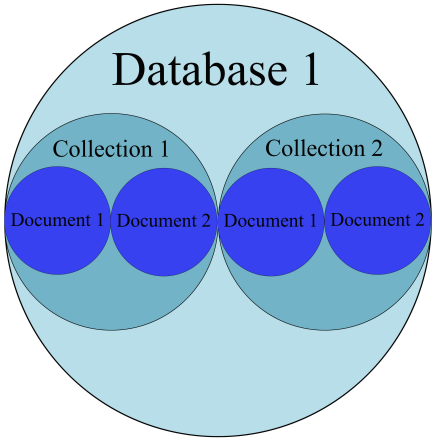

Figure 6: MongoDB数据库系统架构图

数据拆分,将其分散存储在不同的机器上的过程,这些机器不需要是功能强大的大型机,这样就能够通过廉价机器集群存储更多数据,处理更大的负载。MongoDB 支持自动分片功能,完全摆脱了手动分片带来的种种麻烦。MongoDB 分片集群最终的效果是自动切分数据,并且做到数据的负载均衡。分片主要为了实现3个简单的目标:

1.让集群“不可见”

应用程序只需知道和它打交道的是一个简单的mongodb实例就可以了。

2.保证集群总可以读写

任何集群都无法保证永远运行,但是在合理配置下,永远都不应该出现用户无法读写数据的情况。在功能明显降级前,集群应当允许尽可能多的节点失效。

3.使集群易于扩展

当系统需要更多的空间和资源时,应当可以添加。

2.12 MongoDB系统评价及应用

基于文档的灵活的数据模式，是MongoDB的一大优势，对于数据模型多样或多变的业务场景，相比MySQL等数据库，无需使用DDL语句进行表结构的修改；相比其他Key-Value数据库，由于MongoDB的Value字段对于MongoDB是非透明的，可以对其建立

索引，还可以进行全文检索，在查询效率上更具优势。

MongoDB还具有强大的索引能力，支持创建唯一索引、二级索引、TTL索引和地理位置索引等，这在NoSQL数据库中是数一数二的，在此基础上，MongoDB还提供了执行计划功能，通过explain和hint。命令可以查看执行计划、强制查询某个索引，这些特性相比关系型数据库也不成多让。

MongoDB的复制集是数据库领域领先的高可用和读写负载均衡解决方素，提供了数据自动（异步/同步）复制能力，一个新节点加入到复制集中会自动进行数据初始同步随后进行复制，无需人工干预。

MongoDB 的主要特性有:

(1)丰富的数据类型

MongoDB是面向文档的数据库,放弃关系模型的一个主要原因是为了获得更加灵活的扩展性。它是无模式的,文档的键不会事先定义也不会固定不变,应用层可以方便地处理新增的键或丢失的键,为开发者变更数据模型提供极大的便利。

(2)容易扩展

MongoDB 在设计时考虑了系统扩展的问题,面向文档的数据模型可以自动在多台服务器之间进行分割。通过其Auto-Sharding 机制,可以自动实现集群的数据和负载均衡。

(3)功能丰富

支持辅助索引、存储JavaScript 和MapReduce等其他聚合工具的独特功能。

(4)卓越的性能

MongoDB 对文档进行自动动态填充,预分配数据文件,用空间换取性能的稳定。默认的存储引擎中使用了内存映射文件,将内存的管理工作交给操作系统去处理。

(5)简便的管理

尽可能的让服务器自动配置,通过复制机制来提升系统的可靠性

随着NoSQL得到两大领先Web巨人Google和Amazon的支持，大量开发者开始在他们的应用或者公司中尝试这些产品。小到刚创业的公司,大到大型企业, 不到五年的时间里面,NoSQL用于管理大型数据集得到了广泛的传播和应用。众多知名的企业都用到了NoSQL 数据库，其中包括FaceBook 、Yahoo 、eBuy、BM、TaoBao 等等。其中许多公司也通过开源向全世界贡献出了他们的研究成果和扩展组件。

即时通讯应用平台的主要数据是消息,而消息的主体是文本、图片、音频、视频等,数据格式变化多样,而且数据量大。对这些特点,传统的关系型数据库显得有些吃力。对比了NoSQL 的四种数据模型,采用了文档型数据库中的MongoDB来存储消息数据,主要原因是MongoDB在保证海量数据存储的同时, 还具有高效率的查询性能。MongoDB与传统数据库相比有以下这些优势,以MySQL为例。

1. 扩展性

数据之间的关系弱化，因此从架构层上讲非常易于扩展。数据之间有关系，扩展比较困难。

2. 读写性能

数据结构简单，所以在大数据下能有非常高的读写性能。在大数据下，要保证良好的读写性能需要很高的代价。

3. 数据模型

数据模型灵活简单，无需事先设计好数据结构，更加需要可随时添加字段。需要事先定义好表结构，如果后续开发需要增删字段，会非常麻烦，特别是表的数据量超大时。

4. 数据库扩展性

MongoDB是为集群环境设计的，很容易实现数据库的扩展。数据库扩展很难，当单数据增大时，性能会下降。

5. 负载均衡

MongoDB有自己的自动分片功能。MySQL需要采用第三方的均衡器，很难完美整合。

MongoDB也有一定的局限性，如不宜进

行大数据文件的存储，计算分析能力远不及Hadoop提供的运算分析工具，同时，MongoDB不支持远程事务管理，以及需要跨表和跨文档原子性更新操作，因为MongoDB的事务支持仅限于本机的单文档事务。

MongoDB的应用场景非常广泛，如在高伸缩性的场景下，由于Mongo 的高可伸缩性的特点和功能，十分适合有多台服务器组成的分布式数据库文件存储，同时，因为MongoDB的高效的存储性能以及可伸缩性，非常适合网页数据的存储，插入，更新及删除。

3 结束语

本文对当前的最典型的六种网络存储技术的架构，优劣以及应用作了简要介绍。随着网络的应用和需求不断深入以及技术的不断发展，将涌现更多更好的网络存储新技术，同时企业数据提供更加安全可靠、速度更快、更易于管理的网络存储平台。

未来的世界是网络存储世界，存储作为服务器非常重要的一方面，无论在硬件还是软件方面都已经从主机系统中脱离出来，成为完全独立的系统。而作为未来存储方向的网络存储，更因为其低成本、高可靠性和高智能化，将越来越被众多用户所重视。

随着网络存储技术的发展，各种网络存储技术在功能上将会相互融合，各种网络存储设备的互联性也会得到极大的改善。此外，硬件介质的选取，软件管理方式的不同，都决定着网络存储技术的不同发展方向。

【参考文献】

[1] MongoDB的云数据管理技术的研究与应用 刘一梦（北京交通大学） 2012-06-01

[2] 基于Ceph的云存储系统设计与实现 程靓坤（中山大学） 2014-06-30

[3] 基于ceph文件系统的元数据缓存备份技术的研究与实现 方协云（华中科技大学） 2015-05-01

[4] 基于Hadoop的海量网络数据处理平台的关键技术研究 林文辉（北京邮电大学） 2014-04-25

[5] 基于HDFS的存储技术的研究 王永洲（南京邮电大学） 2013-02-01

[6] 基于iSCSI协议的IP SAN网络存储技术研究 陈大恒（国防科学技术大学） 2006-06-01

[7] 基于Linux的NAS系统设计 程延锋（西安电子科技大学） 2009-01-01

[8] 基于MongoDB的传感器数据分布式存储的研究与应用 郭匡宇（南京邮电大学) 2013-04-01

[9] 基于MongoDB的应用平台的研究与实现 吕林（北京邮电大学） 2015-03-14

[10] 基于MySQL复制技术的数据库集群研究 韦一鸣（杭州电子科技大学） 2013-12-01

[11] 基于MySQL集群实现的高性能数据库架构设计 朱红（上海交通大学） 2013-09-01

[12] 基于NDB引擎的MySQL Cluster的部署规则及测试 李红艳（山东大学） 2015-04-20

[13] Ceph存储技术中CRUSH算法的研究与改进 穆彦良（成都信息工程大学） 2016-06-30

[14] Ceph分布式文件系统的研究及性能测试 李翔（西安电子科技大学） 2014-03-01

[15] NAS网络存储技术研究 刘金柱（华中科技大学） 2009-05-01

[16] HDFS存储和优化技术研究综述 金国栋; 卞昊穹; 陈跃国; 杜小勇（软件学报） 2019-08-12