Chapter 3

第3章 大数据存储

学习目标

- 了解大数据存储的总体情况。
- 掌握大数据存储的各种方式。
- 掌握几种常用的大数据存储技术。
- 熟悉大数据存储的可靠性与风险。

关于大数据，最容易想到的便是其数据量之庞大，如何高效地存储和管理这些海量数据是首要问题。此外，大数据还有种类结构不一、数据源繁杂、增长速度快、存取形式和应用需求多样化等特点。本章将重点介绍大数据的存储状况、方式、技术等。

3.1 大数据存储概述

目前，关于大数据存储，说得最多的热点就是存储虚拟化。对存储虚拟化最通俗的理解就是对一个或者多个存储硬件资源进行抽象，提供统一、更有效率的全面存储服务。从用户的角度来说，存储虚拟化就像是一个存储的大池子，用户看不到也不需要看到后面的磁盘，也不必关心数据是通过哪条路径存储到硬件上的，即整个过程对用户而言是透明的。

存储虚拟化有块虚拟化(Block Virtualization)和文件虚拟化(File Virtualization)两大分类。块虚拟化是指将不同结构的物理存储抽象成统一的逻辑存储，这种抽象和隔离可以让存储系统的管理员为终端用户提供更灵活的服务。文件虚拟化则是指帮助用户在一个多节点的分布式存储环境中再也不用关心文件的具体物理存储位置。

3.1.1 传统存储系统时代

计算机的外部存储系统从1956年由IBM制造出第一块硬盘算起，发展至今已经有半个多世纪了。在这半个多世纪里，存储介质和存储系统都取得了很大的发展和进步。现在，硬盘容量可高达几个TB，成本则很低。

目前,传统存储系统主要有三种架构,即 DAS、NAS 和 SAN。

(1) DAS(Direct Attached Storage,直连式存储)

顾名思义,DAS 是一种通过总线适配器直接将硬盘等存储介质连接到主机的存储方式,在存储设备和主机之间通常没有任何网络设备的参与。可以说 DAS 是最原始、最基本的存储架构,在个人计算机和服务器上也最为常见。DAS 的优势在于架构简单、成本低廉、读写效率高等,而其缺点则是容量有限、难以共享,容易形成“信息孤岛”。

(2) NAS(Network Attached Storage,网络存储系统)

NAS 是一种提供文件级别访问接口的网络存储系统,通常采用 NFS、SMB/CIFS 等网络文件共享协议进行文件存取。NAS 支持多客户端同时访问,为服务器提供了大容量的集中式存储,从而也方便了服务器之间的数据共享。

(3) SAN(Storage Area Network,存储区域网络)

SAN 通过光纤交换机等高速网络设备在服务器和磁盘阵列等存储设备之间搭设专门的存储网络,从而提供高性能的存储系统。

SAN 与 NAS 的基本区别在于 SAN 提供块(Block)级别的访问接口,一般不同时提供文件系统。通常情况下,服务器需要通过 SCSI 等访问协议将 SAN 存储映射为本地磁盘,在其上创建文件系统后进行使用。目前,主流的企业级 NAS 或 SAN 存储产品一般都可以提供 TB 级的存储容量,高端的存储产品甚至可以提供 PB 级的存储容量。

3.1.2 大数据时代的新挑战

相对于传统的存储系统,大数据存储一般与上层的应用系统结合得更加紧密。很多新兴的大数据存储都是专门为特定的大数据应用而设计和开发的,例如专门用来存放大量图片或者小文件的在线存储,支持实时事物的高性能存储等。因此,在不同的应用场景下,底层大数据存储的特点也不尽相同。但是,结合当前主流的大数据存储系统可以总结出如下一些基本特征。

1. 大容量及高可扩展性

大数据的主要来源包括社交网站、个人信息、科学研究数据、在线事物、系统日志以及传感和监控数据等。各种应用系统源源不断地产生着大量数据,尤其是社交类网站的兴起更是加快了数据增长的速度。大数据一般可达到 PB 级甚至 EB 级的信息量,传统的 NAS 或 SAN 存储很难达到这个级别的存储容量。因此,除了巨大的存储容量外,大数据存储还必须拥有一定的可扩展性。扩展包括纵向扩展(Scale-up)和横向扩展(Scale-out)两种方式。鉴于前者的扩展能力有限且成本较高,因此能够提供 Scale-out 能力的大数据存储已经成为主流趋势。

2. 高可用性

对于大数据应用和服务来说,数据是其价值所在。因此,存储系统的可用性至关重要。平均无故障时间(Mean Time Between Failures,MTBF)和平均修复时间(Mean Time To

Repair,MTTR)是衡量存储系统可用性的两个主要指标。传统存储系统一般采用磁盘阵列(RAID)、数据通道冗余等方式保证数据的高可用性和高可靠性。除了这些传统的技术手段外,大数据存储还会采用一些其他技术。例如,分布式存储系统大多采用简单明了的多副本以实现数据冗余;针对 RAID 导致的数据冗余率过高或者大容量磁盘的修复时间过长等问题,近年来学术界和工业界研究和采用了其他编码方式。

3. 高性能

在考量大数据存储性能时,吞吐率、延时和 IOPS(每秒读写次数,Input/Output Operations Per Second)是几个较为重要的指标。对于一些实时事务分析系统,存储的响应速度至关重要;而在其他大数据应用场景中,每秒处理的事务数则可能是最重要的影响因素。大数据存储系统的设计往往需要在大容量、高可扩展性、高可用性和高性能等特性之间做出权衡。

4. 安全性

大数据具有巨大的潜在商业价值,这也是大数据分析和数据挖掘兴起的重要原因之一。因此,数据安全对于企业来说至关重要。数据的安全性体现在存储如何保证数据完整性和持久化等方面。在云计算、云存储行业风生水起的大背景下,如何在多租户环境中保护用户隐私和保障数据安全成了大数据存储面临且亟待解决的新挑战。

5. 自管理和自修复

随着数据量的增加和数据结构的多样化,大数据存储的系统架构也变得更加复杂,管理和维护便成了一大难题。这个问题在分布式存储中尤其突出,因此能够实现自我管理、监测及自我修复将成为大数据存储系统的重要特性之一。

6. 成本

大数据存储系统的成本包括存储成本、使用成本和维护成本等。如何有效降低单位存储给企业带来的成本问题在大数据背景下显得极为重要。如果大数据存储的成本降不下来,动辄几个 TB 或者 PB 的数据量将会让很多中小型企业在大数据的浪潮中望洋兴叹。

7. 访问接口的多样化

同一份数据可能会被不同部门、用户或者应用访问、处理和分析,不同的应用系统由于业务不同,可能会采用不同的数据访问方式。因此,大数据存储系统需要提供多种接口以支持不同的应用系统。

3.2 大数据存储方式

本节介绍大数据的两种常用存储方式。

3.2.1 分布式存储

大数据导致了数据量的爆发式增长,传统的集中式存储(NAS 或 SAN)在容量和性能

上都无法较好地满足大数据的需求。因此，具有优秀可扩展能力的分布式存储成了大数据存储的主流架构方式。分布式存储大多采用普通的硬件设备作为基础设施，因此，单位容量的存储成本也得到了大幅降低。另外，分布式存储在性能、维护性和容灾性等方面也具有不同程度的优势。

分布式存储系统需要解决的关键技术问题包括可扩展性、数据冗余、数据一致性、全局命名空间、缓存等。从架构上来讲，大体上可以将分布式存储分为 C/S(Client/Server)架构和 P2P(Peer to Peer)架构两种。当然，也有一些分布式存储会同时存在这两种架构方式。

分布式存储面临的另外一个共同问题就是如何组织和管理成员节点，以及如何建立数据与节点之间的映射关系。成员节点的动态增加或者离开在分布式系统中基本上是一种常态。

加州大学伯克利分校的 Eric Brewer 教授于 2000 年提出的分布式系统设计的 CAP 理论指出，一个分布式系统不可能同时保证一致性(Consistency)、可用性(Availability)和分区容忍性(Partition Tolerance)这三个要素。因此，任何一个分布式存储系统只能根据其具体的业务特征和具体需求最大化地优化其中的两个要素。当然，除了一致性、可用性和分区容忍性这三个维度，一个分布式存储系统往往会根据具体业务的不同在特性设计上有不同的取舍，例如是否需要缓存模块、是否支持通用的文件系统接口等。

3.2.2 云存储

云存储是由第三方运营商提供的在线存储系统，例如面向个人用户的在线网盘和面向企业的文件、块或对象存储系统等。云存储的运营商负责数据中心的部署、运营和维护等工作，将数据存储包装成服务的形式并提供给客户。云存储作为云计算的延伸和重要组件之一，提供了“按需分配、按量计费”的数据存储服务。因此，云存储的用户不需要搭建自己的数据中心和基础架构，也不需要关心底层存储系统的管理和维护等工作，并可以根据其业务需求动态地扩大或减小对存储容量的需求。

云存储通过运营商集中、统一地部署和管理存储系统，降低了数据存储的成本，也降低了大数据行业的准入门槛，为中小型企业进军大数据行业提供了可能性。例如，著名的在线文件存储服务提供商 Dropbox 就是基于 Amazon Web 服务(Amazon Web Services，AWS)提供的在线存储系统 S3 创立起来的。在云存储兴起之前，创办类似于 Dropbox 这样的初创公司几乎是不太可能的。

云存储背后使用的存储系统其实大多是分布式架构，而云存储因其具有更多新的应用场景，在设计上也遇到了新的问题和需求。例如，云存储在管理系统和访问接口上大多需要解决如何支持多租户的访问方式的问题，而在多租户环境下就不可避免地要解决诸如安全、性能隔离等一系列的问题。另外，云存储和云计算一样，都需要解决关于信任(Trust)的问题，即如何从技术上保证企业的业务数据在第三方存储服务提供商平台上的隐私和安全，这是一个必须解决的技术问题。

云存储将存储作为服务的形式提供给用户，其在访问接口上一般都会秉承简洁易用的特性。例如，Amazon 的 S3 存储通过标准的 HTTP、简单的 REST 接口存取数据，用户分别通过 Get、Put、Delete 等 HTTP 方法进行数据块的获取、存放和删除等操作。出于操作简便方面的考虑，Amazon S3 存储并不提供修改或者重命名等操作；同时，Amazon S3 存储也并不提供复杂的数据目录结构，仅提供非常简单的层级关系；用户可以创建一个自己的数据桶(Bucket)，而所有数据则直接存储在这个 Bucket 中。另外，云存储还需要解决用户分享的问题。Amazon S3 存储中的数据直接通过唯一的 URL 进行访问和标识，因此只要其他用户经过授权，便可以通过数据的 URL 进行访问。

存储虚拟化是云存储的一个重要技术基础，是指通过抽象和封装底层存储系统的物理特性将多个互相隔离的存储系统统一为一个抽象的资源池的技术。通过存储虚拟化技术，云存储可以实现很多新的特性，例如用户数据在逻辑上的隔离、存储空间的精简配置等。

3.2.3 大数据存储的其他需求

1. 去重

数据快速增长是数据中心面临的最大挑战。显而易见，爆炸式的数据增长会消耗巨大的存储空间，迫使数据提供商购买更多的存储空间，然而却未必能赶上数据的增长速度。这里有几个相关问题值得考虑：产生的数据是否都被生产系统循环使用了？如果不是，那么是否可以把这些数据放到廉价的存储系统中？如何让数据备份消耗的存储空间更少？如何让备份的时间更快？数据备份后能保存的时间有多久(物理介质原因)？备份后的数据能否正常取出？

数据去重可以分为基于文件级别的去重和基于数据块级别的去重。一般来讲，将数据切块(Chunk)有两种方式：定长(Fixed Size)和变长(Variable Size)。所谓定长，就是把一个接收到的数据流或者文件按照相同的大小切分，每个 Chunk 都有一个独立的“指纹”。从实现角度来讲，定长文件的切片在实现和管理起来比较简单，但是数据去重复的比率较低，这也是容易理解的，因为每个 Chunk 在文件中都有固定的偏移。但是在最坏的情况下，如果某个文件在文件一开始新增加或者减少一个字符，则将导致所有 Chunk 的“指纹”发生变化。最差的结果是备份两个仅差一个字符的文件会导致重复数据删除率等于零，这显然是不可接受的。为此，变长 Chunk 技术应运而生，它不是简单地根据文件偏移划分 Chunk，而是根据 Anchor(某个标记)对数据进行分片。由于寻找的是特殊标记而不是数据的偏移，因此变长技术能完美地解决定长 Chunk 中由于数据偏移略有变化而导致的低数据去重比例。

2. 分层存储

众所周知，性能好的存储介质往往价格也很高。如何通过组合高性能、高成本的小容量存储介质和低性能、低成本的大容量存储介质，并使其达到性能、价格、容量及功能上的最大优化是一个经典的存储难题。例如，计算机系统上通过从外部存储(如硬盘等)到内存、缓存等一系列存储介质组成的存储金字塔很好地解决了 CPU 的数据访问瓶颈问题。分层存储

是存储系统领域试图解决类似问题的一种技术手段。近年来,各种新的存储介质的诞生给存储系统带来了新希望,尤其是 Flash 和 SSD(Solid State Drive)存储技术的成熟及其量化生产使其在存储产品中得到了越来越广泛的应用。然而企业存储,尤其是大数据存储全部使用 SSD 作为存储介质的成本依然是非常高昂的。

为了能够更好地发挥新的存储介质在读写性能上的优势,同时将存储的总体成本控制在可以接受的范围内,分层存储系统便应运而生。分层存储系统集 SSD 和硬盘等存储媒介于一体,通过智能监控和分析数据的访问热度,将不同热度的数据自动适时地动态迁移到不同的存储介质上。经常被访问的数据将被迁移到读写性能更好的 SSD 上存储,不常被访问的数据则会被存放在性能一般且价格低廉的硬盘矩阵上。这样,分层存储系统在保证不增加太多成本的前提下,大幅提高了存储系统的读写性能。

3.3 大数据的存储技术

大数据存储与管理是指利用存储器把采集到的数据存储起来并建立相应的数据库,以便管理和调用这些数据。由于从多渠道获得的原始数据通常缺乏一致性,因此会导致标准处理和存储技术失去可行性。随着数据不断增长而造成的单机系统性能不断下降,即使不断提升硬件配置,也难以跟上数据增长的速度。

大数据存储和管理的发展过程中出现了以下几类大数据存储和管理系统:分布式文件存储、NoSQL 数据库、NewSQL 数据库。

3.3.1 分布式文件存储

前面已经介绍过的 Hadoop 系统是以开源形式发布的一种对大规模数据进行分布式处理的技术。特别是在处理大数据时代的非结构化数据时,Hadoop 在性能和成本方面都具有优势,而且 Hadoop 通过横向扩展进行扩容也相对容易,因此备受关注。应该说,目前 Hadoop 是最受欢迎的在 Internet 上对搜索关键字进行内容分类的工具,同时它也可以解决许多有关极大伸缩性的问题。

1. 什么是分布式系统

分布式系统(Distributed System)是建立在网络之上的软件系统,作为软件系统,分布式系统具有高度的内聚性和透明性,因此网络和分布式系统之间的区别更多的是高层软件(特别是操作系统),而不是硬件。

内聚性是指每一个数据库分布节点高度自治,有本地的数据库管理系统。透明性是指每一个数据库分布节点对应用来说都是透明的,看不出是本地还是远程。在分布式数据库系统中,用户感觉不到数据是分布的,即用户无须知道关系是否分割、有无副本、数据存储于哪个站点以及事物在哪个站点上执行等。

在一个分布式系统中,一组独立的计算机展现给用户的是一个统一的整体,就像是一个

系统一样。系统拥有多种通用的物理和逻辑资源,可以动态地分配任务,分散的物理和逻辑资源通过计算机网络实现信息交换。系统中存在一个以全局方式管理计算机资源的分布式操作系统。通常对用户来说,分布式系统只有一个模型或范型。在操作系统之上有一层软件中间件负责实现这个模型。

在计算机网络中,这种统一性、模型以及其中的软件都不存在。用户看到的是实际的机器,计算机网络并没有使这些机器看起来是统一的。如果这些机器有不同的硬件或者不同的操作系统,那么这些差异对于用户来说都是完全可见的。如果一个用户希望在一台远程机器上运行一个程序,那么他必须登录到该远程机器上,然后在那台机器上运行该程序。

分布式系统和计算机网络系统的共同点是:多数分布式系统是建立在计算机网络之上的,所以分布式系统与计算机网络在物理结构上是基本相同的。分布式操作系统的设计思想和网络操作系统是不同的,这决定了它们在结构、工作方式和功能上也不同。

网络操作系统要求网络用户在使用网络资源时必须了解网络资源,网络用户必须知道网络中各个计算机的功能与配置、软件资源、网络文件结构等情况。在网络中,如果用户要读一个共享文件,则用户必须知道这个文件存放在哪一台计算机的哪一个目录下。

分布式操作系统是以全局方式管理系统资源的,它可以为用户任意调度网络资源,并且调度过程是“透明”的。当用户提交一个作业时,分布式操作系统能够根据需要在系统中选择最合适的处理器,将用户的作业提交到该处理程序,在处理器完成作业后再将结果传递给用户。在这个过程中,用户并不会意识到有多个处理器存在,这个系统就像是一个处理器。

2. Hadoop

MapReduce 指一种分布式处理的方法,而 Hadoop 则是将 MapReduce 通过开源方式进行实现的框架(Framework)的名称。这是因为 Google 在论文中仅公开了处理方法,而并没有公开程序本身。也就是说,MapReduce 指的只是一种处理方法,而 Hadoop 则是一种基于 Apache 的授权协议,是以开源形式发布的软件程序。

前面已经介绍了 Hadoop 原本是由三大部分组成的,即用于分布式存储大容量文件的 HIDES(Hadoop Distributed File System),用于对大量数据进行高效分布式处理的 Hadoop MapReduce 框架以及超大型数据表 HBase。

从数据处理的角度来看,Hadoop MapReduce 是其中最重要的部分。Hadoop MapReduce 并非用于配备了高性能 CPU 和磁盘的计算机,而是一种工作在由多台通用计算机组成的集群上的对大规模数据进行分布式处理的框架。

Hadoop 将应用程序细分为在集群中任意节点上都可以执行的成百上千个工作负载,并分配给多个节点执行,然后通过对各节点瞬间返回的信息进行重组,以得到最终的回答。虽然存在其他功能类似的程序,但 Hadoop 仍依靠其处理的高速性脱颖而出。

Hadoop 在业界已经被大规模使用。HDFS 有着高容错性的特点,并且部署在低廉的硬件上,实现了异构软硬件平台之间的可移植性。为了尽量减小全局的带宽消耗和读延迟,HDFS 尝试返回给一个读操作距离它最近的副本。HDFS 的硬件故障是常态,而不是异常,

它可以自动维护数据的多份复制，并且能够在任务失败后自动重新部署计算任务，实现了故障的检测和自动快速恢复。HDFS 放宽了可移植操作系统接口（Portable Operating System Interface，POSI）的要求，可以以流的形式访问文件系统中的数据，实现了以流的形式访问写入的大型文件的目的，其重点是数据吞吐量，而不是数据访问的反应时间。HDFS 提供了接口，以让程序自己移动到距离数据存储更近的位置，消除了网络的拥堵，提高了系统的整体吞吐量。HDFS 的命名空间是由名字节点存储的。名字节点使用叫作 EditLog 的事务日志持久地记录每一个对文件系统元数据的改变。名字节点在本地文件系统中用一个文件存储这个 EditLog。整个文件系统命名空间，包括文件块的映射表和文件系统的配置都存储在一个叫作 FsImage 的文件中，FsImage 存储在名字节点的本地文件系统中。FsImage 和 Editlog 是 HDFS 的核心数据结构。

Hadoop 的一大优势是：由于 Hadoop 集群的规模可以很容易地扩展到 PB 级甚至 EB 级，因此企业可以将分析对象由抽样数据扩展到全部数据的范围。而且，由于处理速度有了飞跃性的提升，企业可以进行若干次重复的分析，也可以用不同的查询进行测试，从而有可能获得过去无法获得的更有价值的信息。

Hadoop 是一个能够对大量数据进行分布式处理的软件框架，它是以一种可靠、高效、可伸缩的方式处理数据的。Hadoop 是可靠的，这是因为它会首先假设计算元素和存储失败，因此会维护多个工作数据副本，确保能够针对失败的节点重新进行分布处理。Hadoop 是高效的，这是因为它以并行的方式工作，可以通过并行处理加快处理速度。Hadoop 还是可伸缩的，它能够处理 PB 级的数据。此外，Hadoop 依赖于社区服务器，因此它的成本比较低，任何人都可以使用。

总而言之，Hadoop 是一个能够让用户轻松架构和使用的分布式计算平台。用户可以轻松地在 Hadoop 上开发和运行处理海量数据的应用程序。Hadoop 主要具有以下几个优点。

（1）高可靠性

Hadoop 按位存储和处理数据的能力值得人们信赖。

（2）高扩展性

Hadoop 是在可用的计算机集簇之间分配数据并完成计算任务的，这些集簇可以方便地扩展到数以千计的节点中。

（3）高效性

Hadoop 能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此其处理速度非常快。

（4）高容错性

Hadoop 能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。

Hadoop 带有用 Java 语言编写的框架，因此其运行在 Linux 平台上是非常理想的。Hadoop 上的应用程序也可以使用其他语言编写，如 C++ 。

3.3.2 NoSQL 数据库

作为支撑大数据的基础技术，能和 Hadoop 一样受到越来越多关注的就是 NoSQL 数据库了。

传统关系数据库在数据密集型应用方面显得力不从心，主要表现在灵活性差、扩展性差、性能差等方面。而 NoSQL 数据库摒弃了传统关系数据库管理系统的设计思想，采用不同的解决方案满足扩展性方面的需求。由于 NoSQL 数据库没有固定的数据模式且可以水平扩展，因此它能够很好地应对海量数据的挑战。相对于关系数据库而言，NoSQL 数据库最大的不同是其不使用 SQL 作为查询语言。NoSQL 数据库的主要优势有：可以避免不必要的复杂性、吞吐量高、水平扩展能力强、适用于低端硬件集群、避免了昂贵的对象-关系映射。

1. NoSQL 数据库与关系数据库设计理念的比较

关系数据库中的表都是存储着一些格式化的数据结构，每个元组字段的组成都一样，即使不是每个元组都需要所有字段，但数据库还是会为每个元组分配所有字段，这样的结构可以便于表与表之间进行连接等操作，但从另一个角度来说，它也是造成关系数据库性能瓶颈的一个因素。而非关系数据库 NoSQL 以键值对存储，它的结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，这样就不会局限于固定的结构，可以减少一些时间和空间上的开销。

NoSQL 数据库与传统关系数据库管理系统(RDBMS)之间的主要区别如表 3-1 所示。

表 3-1 RDBMS 与 NoSQL 数据库的区别

	RDBMS	NoSQL
数据类型	结构化数据	主要是非结构化数据
数据库结构	须事先定义，是固定的	无须事先定义，可以灵活改变
数据一致性	通过 ACIO 特性保持严密的一致性	存在临时的、不保持严密一致性的状态(结果匹配性)
扩展性	基本是向上扩展。由于需要保持数据的一致性，因此性能下降明显	通过横向扩展可以在不降低性能的前提下应对大量访问，实现线性扩展
服务器	以在一台服务器上工作为前提	以分布、协作式工作为前提
故障容忍性	为了提高故障容忍性需要很高的成本	有很多无单一故障点的解决方案，成本低
查询语言	SQL	支持多种非 SQL 语言
数据量	(和 NoSQL 相比)较小规模的数据	(和 RDBMS 相比)较大规模的数据

2. NoSQL 数据库技术特点

NOSQL 数据库的诞生缘于现有 RDBMS 存在的一些问题，例如不能处理非结构化数

据、难以横向扩展、扩展性存在极限等。由表 3-1 的对比可见，NoSQL 数据库具备以下特征：数据结构简单、不需要数据库结构定义(或者可以灵活变更)、不对数据一致性进行严格保证、通过横向扩展可以实现很高的扩展性等。简而言之，NoSQL 是一种以牺牲一定的数据一致性为代价，追求灵活性、扩展性的数据库。

(1) 易扩展

NoSQL 数据库种类繁多，但是它们的一个共同特点就是去掉了关系数据库的关系特性。数据之间无关系，这样就非常容易扩展，无形之间在架构的层面上带来了可扩展的能力。

(2) 大数据量，高性能

NoSQL 数据库具有非常高的读写性能，尤其是在大数据量下同样表现优秀，这得益于它的无关系性和数据库结构的简单。一般，MySQL 数据库使用 Query Cache，每次表的更新都会使 Cache 失效，是一种大粒度的 Cache，在针对 Web 2.0 的频繁交互的应用中，Cache 性能不高。而 NoSQL 数据库的 Cache 是纪录级的，是一种细粒度的 Cache，所以 NoSQL 数据库在这个层面上的性能更高。

(3) 灵活的数据模型

NoSQL 数据库无须事先为要存储的数据建立字段，而是随时可以存储自定义的数据格式。而在关系数据库中，增删字段是一件非常麻烦的事情。如果是数据量非常大的表，增加字段简直就是一个噩梦，这一点在大数据量的 Web 2.0 时代中尤其明显。

(4) 高可用

NoSQL 数据库在尽可能不影响性能的情况下可以方便地实现高可用的架构。例如 Cassandra 和 HBase 模型通过复制模型也能实现高可用。

3. 几种主流的 NoSQL 数据库

(1) BigTable

① BigTable 简介。

BigTable 是一个分布式的结构化数据存储系统，用来处理分布在数千台普通服务器上的 PB 级数据。Google 的很多项目都使用 BigTable 存储数据，包括 Web 索引、Google Earth、Google Finance 等。

② 数据模型。

BigTable 是一个稀疏的、分布式的、持久化存储的多维度排序 Map。Map 的索引是行关键字、列关键字以及时间戳；Map 中的每个 value 都是一个未经解析的 byte 数组。

```
(row:string, column:string,time:int64)-->string
```

下面分析一个存储 Web 网页的表的片断。

- 行名：com.cnn.www。

- contents 列族：存放网页的内容。
- anchor 列族：存放引用该网页的锚链接文本。
- anchor：cnnsi.com 列表示被 cnnsi.com 引用。
- anchhor：my.look.ca 列表示被 my.look.ca 引用。
- (com.cnn.www,anchor：my.look.ca,t8)->CNN.com。

③ 技术要点。

ⓐ 基础：GFS、Chubby、SSTable。

- BigTable 使用 Google 的分布式文件系统(GFS)存储日志文件和数据文件。
- Chubby 是一个高可用的、序列化的分布式锁服务组件。
- BigTable 内部存储数据的文件是 Google SSTable 格式的。

ⓑ 元数据组织：chubby->metadata->tablet。

元数据与数据都保存在 Google FS 中，客户端通过 Chubby 服务获得表格元数据的位置。

ⓒ 数据维护与访问。

master server 将每个 tablet 的管理责任分配给各个 tablet server，tablet 的分布信息都保存在元数据中，所以客户端无须通过 master 访问数据，只需要直接和 tablet server 通信。

ⓓ Log-structured 数据组织。

写操作不直接修改原有的数据，而是将一条记录添加到 commit log 的末尾，读操作需要从 log 中 merge 出当前的数据版本。具体实现为：SSTable、Memtable(Memtable 即内存表，用来将新数据或常用数据保存在内存表，可以减少磁盘 I/O 访问)。

④ 特点。

- 适合大规模海量数据和 PB 级数据。
- 分布式、并发数据处理，效率极高。
- 易于扩展，支持动态伸缩，适用于廉价设备。
- 适用于读操作，不适合写操作。
- 不适用于传统关系数据库。

(2) Dynamo

① Dynamo 简介。

Dynamo 最初是 Amazon 使用的一个私有的分布式存储系统。

② 设计要点。

Dynamo 采用 P2P 架构，区别于 Google FS 的 Single Master 架构，Dynamo 无须中心服务器记录系统的元数据。Dynamo 考虑了 Performance(性能)、Availability(可用性)、Durability(数据持久性)三者的平衡，可以根据应用的需求自由调整这三者的比例。

③ 技术要点。

Dynamo 将所有主键的哈希数值空间组成一个首位相接的环状序列，为每台计算机随

机赋予一个哈希值，不同的计算机就会组成一个环状序列中的不同节点，而该计算机就负责存储这一段哈希空间内的数据。数据定位使用一致性哈希；对于一个数据，首先计算其哈希值，根据其所在的某个区间顺时针进行查找，一旦找到第一台计算机，该计算机就负责存储该数据，对应的存取操作及冗余备份等操作也由其负责，以此实现数据在不同计算机之间的动态分配。

对于一个环状节点，如 M 个节点，如果一份数据需要保持 N 个备份，则该数据落在某个哈希区间内发现的第一个节点负责后续对应的 N－1 个节点的数据备份（M≥N），Vector Clock 允许数据的多个备份存在多个版本，以提高写操作的可用性（用弱一致性换取高可用性）。分布式存储系统为某个数据保存多个备份，数据写入要尽量保证备份数据同时获得更新，Dynamo 采取数据最终一致性，在一定时间窗口中对数据的更新会传播到所有备份中，但是在时间窗口内，如果有用户读取到旧的数据，则通过向量时钟（Vector Clock）容错，并非采用严格的数据一致性检查，从而实现最终一致性。当节点故障恢复时，Dynamo 可动态维护系统可用性，使系统的写入成功率大幅提升。使用 Merkle Tree 为数据建立索引，只要任意数据有变动，都将快速反馈出来。Dynamo 的网络互联采用 Gossip-based Membership Protocol 通信协议，目标是让节点与节点之间实现通信，实现去中心化。

④ 特点。

ⓐ 高可用。

Dynamo 在设计上没有单点，每个实例由一组节点组成，从应用的角度看，实例提供了 I/O 能力。一个实例上的节点可能位于不同的数据中心内，这样哪怕有一个数据中心出现问题，也不会导致数据丢失。

ⓑ 总是可写。

Hinted Handoff 确保在系统节点出现故障或节点恢复时能灵活处理，可根据应用类型优化可用性、容错性和高效性配置去中心化，人工管理工作少。

ⓒ 可扩展性较差。

由于增加机器需要给机器分配 DHT（Distributed Hash Table）算法所需的编号，操作复杂度较高，且每台机器存储了整个集群的机器信息及数据文件的 Merkle Tree 信息，机器最大规模只能到几千台。

3.3.3 NewSQL 数据库

NewSQL 数据库系统既保留了 SQL 查询的方便性，又能提供高性能和高可扩展性，而且还能保留传统的事务操作的 ACID 特性，它既能达到 NoSQL 数据库系统的吞吐率，又不需要在应用层进行事务的一致性处理。此外，NewSQL 数据库还保持了高层次结构化查询语言的优势。这类数据库系统目前主要包括 Clustrix、NimbusDB 及 VoltDB 等。

NewSQL 数据库被认为是针对 New OLTP 系统的 NoSQL 数据库或者是 OldSQL 系统的一种替代方案。NewSQL 数据库既可以提供传统的 SQL 数据库系统的事务保证，又

能提供 NoSQL 数据库系统的可扩展性。如果 New OLTP 将来能有很大的市场，那么将会有越来越多不同架构的 NewSQL 数据库系统出现。

NewSQL 数据库系统涉及很多新颖的架构设计，例如可以将整个数据库都放在主内存中运行，从而消除数据库传统的缓存管理(Buffer)；可以在一个服务器上只运行一个线程，从而去除轻量的加锁阻塞(Latching)，尽管某些加锁操作仍然需要，并且会影响性能；可以使用额外的服务器进行复制和失败恢复工作，从而取代昂贵的事务恢复操作。

NewSQL 数据库是一类新型的关系数据库管理系统，对于 OLTP 应用来说，它们可以提供和 NoSQL 数据库系统一样的扩展性和性能，还能提供和传统的单节点数据库一样的 ACID 事务保证。

NewSQL 数据库系统非常适合处理具有短事务、点查询、Repetitive(用不同的输入参数执行相同的查询)类型的事务。另外，大部分 NewSQL 数据库系统通过改进原始的 System R 设计可以达到高性能和高扩展性，例如取消重量级的恢复策略、改进并发控制算法等。

NewSQL 数据库主要包括以下两类系统。

① 拥有关系数据库产品和服务，并将关系模型的好处带到分布式架构上。

② 提高关系数据库的性能，使之达到不用考虑水平扩展问题的程度。

3.3.4 云存储技术

面对大数据的海量异构数据，传统存储技术面临建设成本高、运维复杂、扩展性有限等问题，成本低廉、扩展性高的云存储技术日益得到关注。

1. 云存储的定义

由于业内对云存储没有统一的标准，各厂商的技术发展路线也不尽相同，结合云存储技术发展背景及主流厂商的技术方向，可以得出如下定义：云存储是指通过集群应用、网格技术或分布式文件系统等将网络中大量不同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系统。

2. 云存储的架构

云存储是由网络设备、存储设备、服务器、应用软件、公用访问接口、接入网络和客户端程序等组成的复杂系统。云存储以存储设备为核心，通过应用软件对外提供数据存储和业务访问服务。云存储的架构如图 3-1 所示。

(1) 存储层

存储设备数量庞大且分布在不同地域，彼此通过广域网、互联网或光纤通道网络连接在一起。在存储设备之上是一个统一存储设备管理系统，可以实现存储设备的逻辑虚拟化管理、多链路冗余管理以及硬件设备的状态监控和故障维护。

(2) 基础管理层

通过集群、分布式文件系统和网格计算等技术实现云存储设备之间的协同工作，使多个

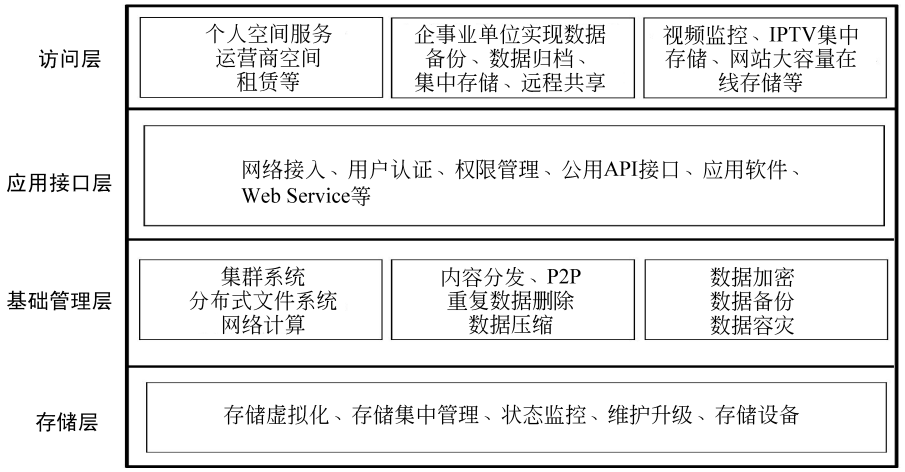

图 3-1　云存储的架构

存储设备可以对外提供同一种服务，并提供更强大的数据访问性能。数据加密技术保证云存储中的数据不会被未授权的用户访问，数据备份和容灾技术可以保证云存储中的数据不会丢失，以保证云存储自身的安全和稳定。

（3）应用接口层

不同的云存储运营商根据业务类型开发了不同的服务接口并提供不同的服务。例如视频监控、视频点播应用平台、网络硬盘、远程数据备份应用等。

（4）访问层

授权用户可以通过标准的公共应用接口登录云存储系统，享受云存储服务。

3. 云存储中的数据缩减技术

大数据时代，云存储的关键技术主要有存储虚拟化、分布式存储、数据备份、数据缩减、内容分发、数据迁移、数据容错等技术。其中，数据缩减技术能够满足海量信息爆炸式增长的趋势，可以在一定程度上节约企业存储成本，提高工作效率，因此该技术成为了人们关注的重点。

（1）自动精简配置技术

传统配置技术为了避免重新配置可能造成的业务中断，通常会过度配置容量。在这种情况下，一旦存储分配给某个应用，就不可能再重新分配给另一个应用，因此造成了已分配的容量无法得到充分利用的情况，造成资源的极大浪费。自动精简配置技术利用虚拟化方法减少物理存储空间的分配，最大限度地提升存储空间的利用率，其核心原理是“欺骗”操作系统，让操作系统误认为存储设备中有很大的存储空间，而实际的物理存储空间则没有那么大。自动精简配置技术会减少已分配但未使用的存储容量的浪费，在分配存储空间时，需要多少存储空间，系统就分配多少存储空间。随着数据存储的信息量越来越多，实际存储空间也可以及时扩展，无须用户手动处理。

(2) 自动存储分层技术

自动存储分层(AST)技术是在存储上减少数据量的另外一种机制,主要用来帮助数据中心最大限度地降低成本和复杂性。在过去,移动数据主要依靠手工操作,由管理员判断这个卷的数据访问压力或大或小,在迁移的时候也只能一个整卷一起迁移。自动存储分层技术的特点则是其分层的自动化和智能化。利用自动存储分层技术,一个磁盘阵列能够把活动数据保留在快速、昂贵的存储上,把不活跃的数据迁移到廉价的低速层上,使用户数据保留在合适的存储层级,减少了存储需求的总量,降低了成本,提升了性能。随着固态存储在当前磁盘阵列中的应用以及云存储的来临,自动存储分层技术已经成为大数据时代补充内部部署的存储的主要方式。

(3) 重复数据删除技术

物理存储设备在使用一段时间后必然会出现大量重复的数据。重复数据删除技术(De-duplication,一般称为 De-dupe 技术)作为一种数据缩减技术,可以对存储容量进行优化。该技术可以删除数据集中重复的数据且只保留其中一份,从而消除冗余数据。使用 De-dupe 技术可以将数据缩减到原来的 1/20～1/50。由于大幅减少了物理存储空间的信息量,从而达到减少传输过程中的网络带宽、节约设备成本、降低能耗的目的。重复数据删除技术按照消重的粒度可以分为文件级和数据块级。可以同时使用 2 种以上的 Hash 算法计算数据指纹,以获得非常小的数据碰撞概率。具有相同指纹的数据块即可认为是相同的数据块,其在存储系统中仅需要保留一份。这样,一个物理文件在存储系统中就只对应一个逻辑表示。

(4) 数据压缩技术

数据压缩技术是提高数据存储效率最古老、最有效的方法,它可以显著降低待处理和待存储的数据量,一般情况下可以实现 2∶1～3∶1 的压缩,对于随机数据的效果更好,如数据库。该技术的原理是将接收到的数据通过存储算法存储到更小的空间中。在线压缩(RACE)是最新研发的数据压缩技术,与传统压缩技术不同。RACE 技术不仅能在数据首次写入时对其进行压缩,以帮助系统控制大量数据在主存中杂乱无章地存储的情形,还可以在数据写入存储系统前压缩数据,以进一步提高存储系统中的磁盘和缓存的性能和效率。数据压缩技术中使用的 LZS 算法是基于 LZ77 实现的,主要由滑窗(Sliding Window)和自适应编码(Adaptive Coding)构成。在进行压缩处理时,在滑窗中查找与待处理数据相同的块,并用该块在滑窗中的偏移值及块长度替代待处理数据,从而实现压缩编码。如果滑窗中没有与待处理数据块相同的字段,或偏移值及长度数据超过被替代数据块的长度,则不进行替代处理。LZS 算法的实现非常简洁,处理也比较简单,能够适应各种高速应用。

3.4 大数据存储的可靠性

大数据是一种数据集成,是指无法在可容忍的时间内用传统 IT 技术和软硬件工具对其进行感知、获取、管理、处理和服务的数据集合。大数据也是一项 IT 技术。大数据是继

云计算、物联网之后 IT 产业又一次颠覆性、革命性的技术变革。大数据时代的来临已成为不可阻挡的趋势。在现代社会,大数据正在改变着世界,改变着人们的生活,已经成为影响一个国家及其全体国民的重要事物。对现有的各种大数据进行系统集成和有效利用是现阶段信息化建设的核心任务。但大数据在给经济社会的发展带来巨大便利和商机的同时,也蕴藏着各种潜在的风险。

3.4.1 大数据可靠性的风险

1. 数据窃取

大数据采用云端存储处理海量数据,对数据的管理较为分散,无法控制用户进行数据处理的场所,难以区分合法用户与非法用户,容易导致非法用户入侵并窃取重要信息。在网络空间,大数据更容易成为攻击目标。

2. 非法添加和篡改分析结果

黑客入侵大数据系统并非法添加和篡改分析结果,可能对金融机构以及个人甚至政府的决策造成干扰。

3. 个人信息泄露

大数据面临用户移动客户端安全管理和个人金融隐私信息保护的双重安全挑战,企业较难在安全性与便利性之间达成平衡。

4. 数据存储安全

“数据大集中”在中国金融业获得了广泛认可。一些大型券商和银行纷纷建设数据种子,作为金融服务的核心和基础。大数据对数据存储的物理安全性、多副本性要求较高。一方面,各类复杂数据的集中存储容易导致存储混乱,造成安全管理违规;另一方面,安全防护手段的更新升级速度无法跟上数据量的非线性增长,大数据安全防护容易出现漏洞。

3.4.2 提高大数据可靠性的方法

1. 建立大数据金融生态系统

大数据金融生态系统是指金融大数据与从事大数据金融活动的个人、家庭、厂商、政府、非政府组织等社会行为主体之间共同形成的动态系统。

各主体在从事金融交易活动时会产生海量金融大数据,这种大数据呈几何式增长,构建海量金融大数据与大数据金融活动相互影响的大数据金融生态系统是非常有必要的,必须加强对系统内不法行为的限制,杜绝信息篡改和窃取,保护个人隐私,促进信息流的良性循环,保证数据的真实可靠。同时要引入信用系统、评级系统等,以强化金融大数据系统的安全性和可靠性。

2. 规范数据提取及交易程序

一方面,必须明确收集大数据主体。大数据的产生包括两个渠道,一是来自法律授权的

收集，二是公民使用网络设备自动形成的信息记录。两种信息源头的信息会混杂在一起，从而形成更为精准、私密的信息。针对此类信息的收集，目前尚无法做到程序化和模板化，只能秉持两个基本原则，即利益原则和知情与许可原则。

另一方面，必须明晰数据交易主体。大数据是静态的提取与存储过程，也是动态的交易过程。在金融领域，不论是个人信息、企业信息还是政府信息都非常重要，应严格审查和审批参与大数据交易的主体及其掌握的信息，从信息供给层面予以规范。

本章小结

本章首先介绍了大数据存储在新时代面临的挑战，然后介绍了常用的两种大数据存储方式，重点介绍了大数据的几种存储技术，最后分析了大数据可靠性所面临的风险及提高大数据可靠性的方法。

通过本章的学习，读者应该重点掌握和熟悉几种常用的大数据存储技术。

实验 3

熟悉大数据存储技术

1. 实验目的

(1) 了解大数据分布式存储技术。

(2) 熟悉 NoSQL 数据库技术。

2. 工具/准备工作

(1) 在开始本实验之前，请认真阅读教材的相关内容。

(2) 准备一台带有浏览器且能够联网的计算机。

3. 实验内容

(1) 请简述什么是分布式存储技术。

(2) 请简述什么是 NoSQL 数据库技术。

4. 实验总结

5. 实验评价（教师）