第3章

大数据存储与管理基本概念

随着大数据时代的到来，需要存储与管理的数据越来越多，数据也呈现越来越复杂的结构。如何对海量数据进行存储和有效管理变得极其重要。本章主要从大数据存储的相关概念、数据存储技术的发展、分布式系统基础理论以及数据仓库、NoSQL 数据库等方面对大数据的存储和管理技术进行介绍。

3.1 大数据的数据类型

大数据的数据类型繁多，是大数据“大”的一个体现。其来源极其广泛，几乎包括人们日常生活中可以见到的所有类型。随着物联网、互联网以及移动通信网络的飞速发展，数据的格式及种类也都在不断地变化和发展。这些数据来自：

（1）企业和用户的交易数据，如 POS 机数据、信用卡刷卡数据、电子商务数据、互联网点击数据、销售系统数据、客户关系管理系统数据、公司的生产数据、库存数据、订单数据、供应链数据等。

（2）移动通信设备的移动通信数据，如聊天信息、定位信息、网络浏览信息等。

（3）日常的行为数据，如电子邮件、文档、图片、音频、视频，以及通过微信、博客、推特、维基、Facebook 等社交媒体产生的数据流。

（4）机器和传感器创建或生成的数据，如感应器、量表、智能温度控制器、智能电表、工厂机器和连接互联网的家用电器、GPS 系统数据等。

日常生活中的方方面面的数据都是大数据的数据来源。大数据按照数据结构划分，可以划分为结构化数据、半结构化数据和非结构化数据。

3.1.1 结构化数据

结构化数据通常存储在数据库中，是具有数据结构描述信息的数据，这种数据类型先有结构再有数据。例如，可以用二维表等结构来逻辑表达的数据。

结构化数据的数据特点是任何一列数据都不可再分，任何一列数据都有相同的数据类型。例如，关系数据库 SQL、Oracle 中的数据。

相对于结构化数据而言，不方便用数据结构来表达的数据即为半结构化数据和非结构

化数据，包括所有格式的文档、文本、图片、图像、音频、视频、HTML、XML、各类报表等。

3.1.2 半结构化数据

半结构化数据是介于结构化和非结构化之间的数据，这种数据的格式一般比较规范，都是纯文本文件，如 XML 文档、HTML 文档等。这种数据一般是自描述的，数据的结构和内容混在一起，没有明显的区分。使用这些数据时，需要通过特定的方式进行解析。

半结构化数据主要来源有以下三方面。

（1）在 WWW 等对存储数据无严格模式限制的情形下，常见的有 HTML、XML 和 SGML 文件。

（2）在电子邮件、电子商务、文献检索和病历处理中，存在着大量结构和内容均不固定的数据。

（3）在包含有异构信息源集成情形下，信息源上的互操作要存取的信息源范围很广，包括各类数据库、知识库、电子图书馆和文件系统等。

半结构化数据具有如下特点。

（1）隐含的模式信息。虽然具有一定的结构，但结构和数据混合在一起，没有显式的模式定义（HMTL 文件是一个典型）。

（2）不规则的结构。一个数据集合可能由异构的元素组成，或用不同类型的数据表示相同的信息。

（3）没有严格的类型约束。由于没有一个预先定义的模式，以及数据在结构上的不规则性，导致缺乏对数据的严格约束。

3.1.3 非结构化数据

非结构化数据是那些非纯文本类型的数据，这类数据没有固定的标准格式，无法对其直接进行解析。如文本文档、多媒体（视频、音频等），它们不容易收集和管理，需要通过一定数据分析和挖掘才能获得有用的数据。

3.2 数据管理技术的发展

从有文字记录开始，人类对自然和社会的认识进程就开始加快。尤其是当人类开始对数据进行有效管理时，人类的认识能力得到了进一步的提升。20 世纪 30 年代，随着工业生产和数据计算的发展，数据管理技术成为一种社会需要。数据管理的核心是对数据实现分类、组织、编码、存储、检索和维护等。

数据管理技术从诞生到现在，经过不断演变和发展，如今已经有了一整套成熟的理论体系和成熟的产业环境，引领了计算机科学的快速发展并带来了巨大的经济效益。数据管理技术中，数据库技术是核心技术，回顾数据管理技术的发展历程，可分为以下几个阶段：文件系统阶段、数据库系统阶段、数据仓库阶段、分布式系统阶段。

3.2.1 文件系统阶段

文件系统阶段是指计算机不仅用于科学计算，而且大量用于管理数据的阶段（从 20 世

纪 50 年代后期到 20 世纪 60 年代中期)。在硬件方面,外存储器有了磁盘、磁鼓等直接存取的存储设备。在软件方面,操作系统中已经有了专门用于管理数据的软件,称为文件系统。

1. 文件系统管理的优点

1) 数据需要长期保存在外存上供反复使用

由于计算机大量用于数据处理,经常对文件进行查询、修改、插入和删除等操作,所以数据需要长期保留,以便于反复操作。

2) 程序之间有了一定的独立性

操作系统提供了文件管理功能和访问文件的存取方法,程序和数据之间有了数据存取的接口,程序可以通过文件名和数据打交道,不必再寻找数据的物理存放位置,至此,数据有了物理结构和逻辑结构的区别,但此时程序和数据之间的独立性尚还不充分。

3) 文件的形式已经多样化

由于已经有了直接存取的存储设备,文件不仅有顺序文件,还有索引文件、链表文件等,因而,对文件的访问可以是顺序访问,也可以是直接访问。

4) 数据的存取基本上以记录为单位

5) 文件系统实现了记录内的结构化

尽管文件系统有上述优点,但它仍存在一些缺点。

2. 文件系统管理的缺点

1) 数据的共享性差,冗余度高

在文件系统中,数据的建立、存取仍依赖于应用程序,基本是一个(或一组)数据文件对应于一个应用程序,即数据仍然是面向应用的。当不同的应用程序具有部分相同的数据时,也必须建立各自的文件,而不能共享相同的数据,因此数据的冗余度大,浪费存储空间。同时,由于相同数据的重复存储和各自管理,容易造成数据的不一致性,给数据的修改和维护带来困难。

2) 数据的独立性不足

文件系统中的数据虽然有了一定的独立性,但是由于数据文件只存储数据,由应用程序来确定数据的逻辑结构并设计数据的物理结构,一旦数据的逻辑结构或物理结构需要改变,必须修改应用程序;或者由于语言环境的改变需要修改应用程序时,也将引起文件数据结构的改变。因此,数据与应用程序之间的逻辑独立性不强。另外,要想对现有的数据再增加一些新的应用会很困难,系统不容易扩充。

3) 并发访问容易产生异常

文件系统缺少对并发操作进行控制的机制,所以系统虽然允许多个用户同时访问数据,但是由于并发的更新操作相互影响,容易导致数据的不一致。

4) 数据的安全控制难以实现

数据不是集中管理。在数据的结构、编码、表示格式、命名以及输出格式等方面不容易做到规范化、标准化,所以其安全性、完整性得不到可靠保证,而且文件系统难以实现对不同用户的不同访问权限的安全性约束。

3.2.2　数据库系统阶段

数据库系统阶段是从 20 世纪 60 年代末期开始的。在这一阶段,数据库中的数据不再

是面向某个应用或某个程序，而是面向整个企业（组织）或整个应用的。

1. 数据库的数据模型

数据库的类型是根据数据模型来划分的，而任何一个数据库管理系统（DBMS）也是根据数据模型有针对性地设计出来的，这就意味着必须把数据库设计成符合 DBMS 支持的数据模型。目前成熟地应用在数据库系统中的数据模型有层次模型、网状模型和关系模型。它们之间的根本区别在于数据之间联系的表示方式不同。层次模型以“树结构”表示数据之间的联系；网状模型以“图结构”来表示数据之间的联系；关系模型用“二维表”（或称为关系）来表示数据之间的联系。

1）层次模型

（1）定义：层次模型是用树状（层次）结构来组织数据的数据模型。

层次模型可表示为一个倒立生长的树，根据数据结构中的树（或者二叉树）的定义，每棵树都有且仅有一个根结点，其余的结点都是非根结点。每个结点表示一个记录类型与对应实体的概念，记录类型的各个字段对应实体的各个属性。各个记录类型及其字段都必须记录。

（2）特征：树的性质决定了层次模型的特征。

① 整个模型中有且仅有一个结点没有父结点，其余的结点必须有且仅有一个父结点，但是所有的结点都可以不存在子结点。

② 所有的子结点不能脱离父结点而单独存在，也就是说，如果要删除父结点，那么父结点下面的所有子结点都要同时删除，但是可以单独删除一些叶子结点。

③ 每个记录类型有且仅有一条从父结点通向自身的路径。

（3）优点。

① 层次模型的结构简单、清晰，很容易看出各个实体之间的联系。

② 操作层次类型的数据库语句比较简单，只需要几条语句就可以完成数据库的操作。

③ 查询效率较高，在层次模型中，结点的有向边表示了结点之间的联系，在 DBMS 中如果有向边借助指针实现，那么依据路径很容易找到待查的记录。

④ 层次模型提供了较好的数据完整性支持，如果要删除父结点，那么其下的所有子结点都要同时删除。

（4）缺点。

① 结构呆板，缺乏灵活性。

② 层次模型只能表示实体之间的 $1:n$ 的关系，不能表示 $m:n$ 的复杂关系，因此现实世界中的很多模型不能通过该模型直接表示。

③ 查询结点的时候必须知道其双亲结点的路径，因此限制了对数据库存取路径的控制。

2）网状模型

（1）定义。

用有向图表示实体和实体之间的联系的数据结构模型称为网状模型。

其实，网状模型可以看作放松层次模型的约束性的一种扩展。网状模型中所有的结点允许脱离父结点而存在，也就是说，在整个模型中允许存在两个或多个没有根结点的结点，同时也允许一个结点存在一个或者多个父结点，成为一种网状的有向图。因此结点之间的

对应关系不再是 1∶n，而是一种 m∶n 的关系，从而克服了层次模型的缺点。

(2) 特征。

① 可以存在两个或者多个结点没有父结点。

② 允许单个结点存在多个父结点。

③ 网状模型中的每个结点表示一个实体，结点之间的有向线段表示实体之间的联系。网状模型中需要为每个联系指定对应的名称。

(3) 优点。

① 网状模型可以很方便地表示现实世界中很多复杂的关系。

② 修改网状模型时，没有层次模型那么多的严格限制，可以删除一个结点的父结点而依旧保留该结点；也允许插入一个没有任何父结点的结点，这样的插入在层次模型中是不被允许的，除非首先插入的是根结点。

③ 实体之间的关系在底层中可以借由指针实现，因此在这种数据库中执行存取操作的效率较高。

(4) 缺点。

① 网状模型的结构复杂，使用不易，随着应用环境的扩大，数据结构越来越复杂，数据的插入、删除牵动的相关数据太多，不利于数据库的维护和重建。

② 网状模型数据之间的彼此关联比较大，该模型其实是一种导航式的数据模型结构，不仅要说明要对数据做些什么，还要说明操作的记录路径。

3) 关系模型

关系模型对应的数据库就是关系数据库了，这是目前应用最多的数据库。

(1) 定义。

使用表格表示实体和实体之间关系的数据模型称为关系模型。

使用关系模型的数据库是关系数据库，同时也是被普遍使用的数据库，如 MySQL 就是一种流行的关系数据库。支持关系模型的数据库管理系统称为关系数据库管理系统。

(2) 特征。

① 关系模型中，无论是实体还是实体之间的联系，都是被映射成统一的关系——一张二维表，在关系模型中，操作的对象和结果都是一张二维表。

② 关系数据库可用于表示实体之间的多对多的关系，只是此时要借助一个中间表——表，来实现多对多的关系。例如，学生选课系统中学生和课程之间表现出一种多对多的关系，那么需要借助选课表将二者联系起来。

③ 关系必须是规范化的关系，即每个属性是不可分割的实体，不允许表中表的存在。

④ 关系数据库通常具备 ACID 特性。

- Atomicity(原子性)：整个事务中的所有操作，要么全部完成，要么全部不完成。
- Consistency(一致性)：事务必须始终保持系统处于一致的状态，不管在任何给定的时间内并发事务有多少。
- Isolation(隔离性)：每一事务在系统中认为只有它自己在使用系统，每一个事务内部的操作及使用的数据对其他并发事务是隔离的，并发执行的各个事务之间不能相互干扰。
- Durability(持久性)：事务一旦完成，就不能返回。接下来的其他操作或故障都不应

该对其执行结果有任何影响。

(3) 优点。

① 结构简单，关系模型是一些表格的框架，实体的属性是表格中列的条目，实体之间的关系也是通过表格的公共属性表示，结构简单明了。

② 关系模型中的存取路径对用户而言是完全隐蔽的，使程序和数据具有高度的独立性。

③ 操作方便，在关系模型中操作的基本对象是集合而不是某一个元组。

④ 有坚实的数学理论作基础，包括逻辑计算、关系代数等。

(4) 缺点。

① 查询效率低。关系模型提供了较高的数据独立性和非过程化的查询功能(查询的时候只需指明数据存在的表和需要的数据所在的列，不用指明具体的查找路径)，因此加大了系统的负担。

② 由于查询效率较低，因此需要数据库管理系统对查询进行优化，加大了 DBMS 的负担。

2. 数据库系统阶段的特点

1) 采用复杂的结构化的数据模型

数据库系统不仅要描述数据本身，还要描述数据之间的联系。

2) 较高的数据独立性

数据和程序彼此独立，数据存储结构的变化尽量不影响用户程序的使用。

3) 最低的冗余度

数据库系统中的重复数据被减少到最低程度，这样，在有限的存储空间内可以存放更多的数据并减少存取时间。

4) 数据由 DBMS 统一管理和控制

DBMS 的控制功能包括：数据的安全性保护，以防止数据的丢失和被非法使用；数据的完整性检查，以保护数据的正确、有效和相容；数据的并发控制，避免并发程序之间的相互干扰；具有数据的恢复功能，在数据库被破坏或数据不可靠时，系统有能力把数据库恢复到最近某个时刻的正确状态。

数据库是长期存储在计算机内有组织的、大量的、共享的数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。DBMS 在数据库建立、使用和维护时对数据库进行统一控制，以保证数据的完整性和安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对系统进行恢复。

数据库系统的出现使信息系统从以加工数据的程序为中心转向围绕共享的数据库为中心的新阶段。这样既便于数据的集中管理，又有利于应用程序的研制和维护，提高了数据的利用率和相容性，提高了决策的可靠性。

3.2.3 数据仓库阶段

数据仓库的概念是由数据仓库之父比尔·恩门(Bill Inmon)在 1991 年出版的 *Building the Data Warehouse*(《建立数据仓库》)一书中所提出的定义并被广泛接受。数据仓库(Data Warehouse)是一个面向主题的(Subject Oriented)、集成的(Integrated)、非易失

的(Non-Volatile)、随时间而变化的(Time Variant)数据集合,用于支持管理决策(Decision Making Support)。由此可以看出,数据仓库的目的很明确,就是为支持管理决策服务的。数据仓库是决策支持系统和联机分析应用数据源的结构化数据环境。

数据仓库是基于计算机技术的快速发展和大数据时代下企业的需求提出的。绝大多数数据仓库的建设目的都是为了进行数据分析与挖掘,基本都是基于关系数据库与多维数据库建立。

数据仓库系统的主要应用是 OLAP(On-Line Analytical Processing),支持复杂的分析操作,侧重决策支持,并且提供直观易懂的查询结果。

1. 数据仓库的特性

从数据仓库的定义不难看出数据仓库的 4 个特点:面向主题的、集成的、数据不可更新的、数据随时间不断变化的。

1) 数据仓库是面向主题的

数据仓库一般从用户实际需求出发,将不同平台的数据源按设定主题进行划分整合,与传统的面向事务的操作型数据库不同,具有较高的抽象性。面向主题的数据组织方式,就是在较高层次上对分析对象提供一个完整、统一并一致的描述,能完整且统一地刻画各个分析对象所涉及的有关企业的各项数据,以及数据之间的联系。

2) 数据仓库的数据是集成的

数据仓库中存储的数据大部分来源于传统的数据库,但并不是将原有数据简单地直接导入,而是需要进行预处理。这是因为事务型数据中的数据一般都是有噪声的、不完整的和数据形式不统一的。这些“脏数据”的直接导入将对在数据仓库基础上进行的数据挖掘造成混乱。“脏数据”在进入数据仓库之前必须经过抽取、清洗、转换,消除数据错误,完成数据统一,并进行数据的计算和综合,才能生成从面向事务转而面向主题的数据集合。数据集成是数据仓库建设中最重要,也是最复杂的一步。

3) 数据仓库的数据是不可更新的

数据仓库中的数据主要为决策者分析提供数据依据。决策依据的数据是不允许进行修改的。即数据保存到数据仓库后,用户仅能通过分析工具进行查询和分析,而不能修改。数据的更新升级主要在数据集成环节完成,数据一旦生成,便不可修改,同时具有一定的存储期限,过期的数据将在数据仓库中直接筛除。也正是由于其不可修改性,数据仓库的管理相对数据库管理而言要简单很多,省去了数据库管理中的完整性保护、并发控制这样的技术难点。

4) 数据仓库的数据是随时间变化的

数据仓库数据会随时间变化而定期更新,不可更新是针对应用而言,即用户分析处理时不更新数据。每隔一段固定的时间间隔后,抽取运行数据库系统中产生的数据,转换后集成到数据仓库中。随着时间的变化,数据以更高的综合层次被不断综合,以适应趋势分析的要求。当数据超过数据仓库的存储期限,或对分析无用时,从数据仓库中删除这些数据。关于数据仓库的结构和维护信息保存在数据仓库的元数据中,数据仓库维护工作由系统根据其中的定义自动进行或由系统管理员定期维护。

2. 数据仓库的组成

数据仓库通常由数据仓库的数据库、数据抽取工具、元数据、访问工具和数据集市 5 部

分组成。

1）数据仓库的数据库

数据仓库的数据库是整个数据仓库环境的核心，是数据存放的地方并提供对数据检索的支持。相对于操纵型数据库来说，其突出的特点是提供对海量数据的支持，强大的元数据管理和快速的检索技术。

2）数据抽取工具

把数据从各种各样的存储方式中抽取出来，进行必要的转换、整理，再存放到数据仓库内。

对各种不同数据存储方式的访问能力是数据抽取工具的关键，例如，COBOL 程序、MVS 作业控制语言(JCL)、UNIX 脚本和 SQL 语句等，以访问不同的数据。

数据转换包括数据的连接和合并，删除对决策应用没有意义的数据段；转换到统一的数据名称和定义；计算统计和衍生数据；给缺值数据赋予默认值；把不同的数据定义方式统一。

3）元数据

描述数据仓库内数据的结构和建立方法的数据。可将其按用途的不同分为两类：技术元数据和商业元数据。

技术元数据是数据仓库的设计和管理人员用于开发和日常管理数据仓库时用的数据。包括：数据源信息；数据转换的描述；数据仓库内对象和数据结构的定义；数据清理和数据更新时用的规则；源数据到目的数据的映射；用户访问权限，数据备份历史记录，数据导入历史记录，信息发布历史记录等。

商业元数据从商业业务的角度描述了数据仓库中的数据。包括对数据、查询、报表等进行描述的数据。

元数据为访问数据仓库提供了一个信息目录，这个目录全面描述了数据仓库中都有什么数据、这些数据是怎么得到的、怎么访问这些数据。它是数据仓库运行和维护的中心，数据仓库服务器利用它来存储和更新数据，用户通过它来了解和访问数据。

4）访问工具

为用户访问数据仓库提供手段，有数据查询和报表工具、应用开发工具、管理信息系统(EIS)工具、在线分析(OLAP)工具、数据挖掘工具。

5）数据集市

为了特定的应用目的或应用范围，而从数据仓库中独立出来的一部分数据，也可称为部门数据或主题数据。在数据仓库的实施过程中往往可以从一个部门的数据集市着手，以后再用几个数据集市组成一个完整的数据仓库。需要注意的是，在实施不同的数据集市时，同一含义的字段定义一定要相容，这样在以后实施数据仓库时才不会造成大麻烦。

3.2.4 分布式系统阶段

分布式系统是建立在网络之上的软件系统。正是因为软件的特性，所以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在于高层软件(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结点高度自治。透明性是指每一

个数据库分布结点对用户的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、数据存于哪个站点以及事务在哪个站点上执行等。

分布式系统在数据存储方面包括分布式文件系统和分布式数据库系统两部分。

1. 分布式文件系统

计算机通过文件系统管理、存储数据，而信息爆炸时代中人们可以获取的数据呈指数级增长，单纯通过增加硬盘个数来扩展计算机文件系统的存储容量的方式，在容量大小、容量增长速度、数据备份、数据安全等方面的表现都差强人意。分布式文件系统可以有效解决数据的存储和管理难题：将固定于某个地点的某个文件系统，扩展到任意多个地点/多个文件系统，众多的结点组成一个文件系统网络。每个结点可以分布在不同的地点，通过网络进行结点间的通信和数据传输。人们在使用分布式文件系统时，无须关心数据是存储在哪个结点上或者是从哪个结点获取的，只需要像使用本地文件系统一样管理和存储文件系统中的数据。

相对于本机端的文件系统而言，分布式文件系统(Distributed File System，DFS)，或者网络文件系统(Network File System，NFS)，是一种允许文件通过网络在多台主机上分享的文件系统，可让多机器上的多用户分享文件和存储空间。在这样的文件系统中，客户端并非直接访问底层的数据存储区块，而是通过网络，以特定的通信协议和服务器沟通。借由通信协议的设计，可以让客户端和服务器端都能根据访问控制清单或者授权，来限制对于文件系统的访问。相对地，在一个分享的磁盘文件系统中，所有结点对数据存储区块都有相同的访问权，在这样的系统中，访问权限就必须由客户端程序来控制。分布式文件系统可能包含的功能有数据复制与容错。也就是说，即使系统中有一小部分的结点离线，整体来说，系统仍然可以持续运作而不会有数据损失。分布式文件系统和分布式数据存储的界限是模糊的，但一般来说，分布式文件系统是被设计用在局域网，比较强调的是传统文件系统概念的延伸，并通过软件方法来达成容错。而分布式数据存储，则是泛指应用分布式计算技术的文件和数据库等提供数据存储服务的系统。

分布式文件系统将服务范围扩展到了整个网络，不仅改变了数据的存储和管理方式，也拥有了本地文件系统所无法具备的数据备份、数据安全等优点。目前常见的分布式文件系统有 GFS、HDFS、Lustre、GridFS、mogileFS、TFS、FastDFS 等，各自适用于不同的领域。一个分布式文件系统的性能主要由以下几个方面决定。

(1) 数据的存储方式：这里涉及大容量存储、数据划分、负载均衡、可扩展性、一致性等问题。

(2) 数据的读取：这里涉及高性能、易用性、一致性等问题。

(3) 数据的安全：这里涉及容错、事务与并发控制、数据备份及恢复等问题。

虽然分布式文件系统涉及许多新的问题，但是其对于硬件资源的合理利用，对大量数据的高速处理，对于海量数据的深入挖掘，对数据安全的保障，以及各种技术的发展和新的业务需求，起着越来越重要的作用，也是当前计算机技术发展的一大热门关键技术。

2. 分布式数据库系统

随着传统的数据库技术日趋成熟、计算机网络技术的飞速发展和应用范围的扩充，数据库应用已经普遍建立于计算机网络之上。这时集中式数据库系统表现出它的不足：数据按

实际需要已在网络上分布存储，再采用集中式处理，势必造成通信开销大；应用程序集中在一台计算机上运行，一旦该计算机发生故障，则整个系统受到影响，可靠性不高；集中式处理引起系统的规模和配置都不够灵活，系统的可扩充性差。在这种形势下，集中式数据库的“集中计算”概念向“分布计算”概念发展。

分布式数据库是多个互连的数据库，它们通常位于多个服务器上，但彼此通信以实现共同的目标；通过分布式数据库管理系统（DDBMS）进行管理。

分布式数据库为数据库管理领域提供了分布式计算的优势。基本上，可以将分布式数据库定义为分布在计算机网络上的多个相关数据库的集合。

1）分布式数据库的特点

（1）物理分布性：数据不是存储在一个场地，而是存储在计算机网络的多个场地。

（2）逻辑整体性：数据物理分布在各个场地，但逻辑上是一个整体，它们被所有用户（全局用户）共享，并由一个 DDBMS 统一管理。

（3）场地自治性：各场地上的数据由本地的 DBMS 管理，具有自治处理能力，完成本场地的应用（局部应用）。

（4）场地之间协作性：各场地虽然具有高度的自治性，但是又相互协作构成一个整体。

2）分布式数据库的优点

（1）具有灵活的体系结构：允许各个场地的数据模型不同，数据库是分布透明的，隐藏每个文件在系统中物理存储的位置的细节，便于管理。

（2）系统的可靠性高、可用性好：可靠性定义为系统在特定时间运行的概率，而可用性定义为系统在一段时间内连续可用的概率。当数据和 DBMS 软件分布在多个站点上时，一个站点可能会失败而其他站点继续运行，用户仍可以访问其他站点中存在的数据，这样可以提高可靠性和可用性。

（3）可扩展性好，易于集成现有的系统：在分布式环境中，增加数据库大小或添加更多处理器比较容易。

（4）性能优越，局部响应速度快：通过将查询分解为基本上并行执行的多个子查询，提高了并行效率；各个站点可以独立工作，快速响应查询的结果。

3）分布式数据库的缺点

（1）复杂性，分布式数据库架构在设计、故障排除和管理方面要求更高。

（2）系统开销较大，尤其花在通信部分。

（3）数据的安全性和保密性较难处理。

3.3 分布式系统基础理论

在分布式系统中，有一些重要的理论引导着分布式系统的改变和发展。本节主要介绍分布式系统的 CAP 理论和 BASE 思想。

3.3.1 CAP 理论

分布式系统的一个重要理论就是 CAP 理论。2000 年，Eric Brewer 教授在 ACM 分布式计算年会上指出了著名的 CAP 理论。CAP 理论指出：在一个分布式系统中，Consistency

(一致性)、Availability(可用性)、Partition Tolerance(分区容错性)三者不可兼得。最多只能同时满足其中的两个。下面分别介绍这三个属性。

(1) 一致性(C)：在分布式系统中的所有数据备份，在同一时刻是否是同样的值(等同于所有结点访问同一份最新的数据副本)。

(2) 可用性(A)：在集群中一部分结点故障后，集群整体是否还能响应客户端的读写请求(对数据更新具备高可用性)。

(3) 分区容错性(P)：以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在 C 和 A 之间做出选择。

鉴于 CAP 理论只能满足两个属性，下面分为三种情况讨论。

(1) CA without P：如果不要求 P(不允许分区)，则 C(一致性)和 A(可用性)是可以保证的。但其实分区不是想不想分的问题，而是始终会存在，因此 CA 的系统更多的是允许分区后各子系统依然保持 CA。

(2) CP without A：如果不要求 A(可用)，相当于每个请求都需要在 Server 之间强一致，而 P(分区)会导致同步时间无限延长，如此 CP 也是可以保证的。很多传统的数据库分布式事务都属于这种模式。

(3) AP without C：要高可用并允许分区，则需放弃一致性。一旦分区发生，结点之间就可能会失去联系，为了高可用，每个结点只能用本地数据提供服务，而这样会导致全局数据的不一致性。现在众多的 NoSQL 都属于此类。

任何分布式系统，都必须在这三者之间进行取舍。首先就是是否选择分区，由于在一个数据分区内，根据数据库的 ACID 特性，是可以保证一致性的，不会存在可用性和一致性的问题，唯一需要考虑的就是性能问题。对于可用性和一致性，大多数应用就必须保证可用性，在互联网应用中，牺牲了可用性，相当于间接地影响了用户体验，这是不可行的，而唯一可以考虑的就是一致性了，这也是 BASE 思想的来源。

在实际应用中，关系数据库放弃了分区容错性(Partition Tolerance)，具有高的一致性(Consistency)和高可靠性(Availability)。

3.3.2　BASE 思想

BASE 是对 CAP 中一致性和可用性权衡的结果，其来源于对大规模互联网系统分布式实践的结论，是基于 CAP 理论逐步演化而来的，其核心思想是即使无法做到强一致性(Strong Consistency)，但每个应用都可以根据自身的业务特点，采用适当的方式来使系统达到最终一致性(Eventual Consistency)。BASE 是 Basically Available(基本可用)、Soft-state(软状态)和 Eventually Consistency(最终一致性)三个短语的简写。BASE 的含义包括以下三个方面。

- Basically Available：指分布式系统在出现不可预知故障的时候，允许损失部分可用性，但不影响系统的整体可用性。
- Soft-state：允许系统的不同结点的数据副本有一段时间不同步。
- Eventually Consistency：系统中所有的数据副本，在经过一段时间的同步后，最终能够达到一个一致的状态。

BASE与ACID是对立的，完全不同于ACID模型，通过牺牲强一致性，来获得基本可用性和可靠性，并要求达到最终一致性。

CAP、BASE是当前在大数据环境下非常流行的分布式数据库NoSQL的理论基础。

3.4 NoSQL数据库

NoSQL=(Not only SQL)，泛指非关系型的数据库。顾名思义，NoSQL意味着不仅是SQL，而且是一项全新的数据库革命性运动，早期就有人提出，发展至2009年趋势越发高涨。NoSQL的拥护者们提倡运用非关系型的数据存储，相对于铺天盖地的关系数据库运用，这一概念无疑是一种全新的思维的注入。下面详细介绍NoSQL的相关概念、理论基础、应用现状、数据库类型等。

3.4.1 NoSQL数据库的兴起

每一项新技术的产生与兴起都有其时代的背景，大数据时代下传统的关系数据库在面对更大规模的数据和满足更高的访问量时的疲软导致了NoSQL的提出和发展。随着互联网Web 2.0网站的兴起，传统的关系数据库在应对Web 2.0网站特别是超大规模和高并发的SNS类型的Web 2.0纯动态网站时已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。NoSQL数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战。对于NoSQL数据库的兴起，可以大概总结为以下几点。

(1) 对数据库高并发读写的要求。Web 2.0网站要根据用户个性化信息来实时生成动态页面和生成动态信息，所以基本上无法使用动态页面静态化技术，因此数据库并发负载非常高，往往要达到每秒上万次读写请求，关系数据库在应对上万次查询时还能勉强应对，但在应对上万次写SQL请求的时候就有点捉襟见肘了，硬盘的IO也难以承受了。对于一个普通的BBS网站，往往也存在着对高并发请求的需求，例如，网站需要实时统计在线用户的数量，记录热门帖子的点击次数、投票计数等，因此对数据库高并发读写的需求俨然成为一个迫切的需求。

(2) 对海量数据的高效率存储和访问需求。类似Facebook、Twitter这样的SNS网站，每天用户都产生了海量的用户动态，当系统需要在这样存储数以亿计数据的关系数据表中使用SQL查询相应的数据时，其效率是极其低下乃至难以忍受的。再如腾讯、盛大等大型的登录系统这些有着动辄以亿计的账号，关系数据库也难以应对。

(3) 对数据库的高可用性和高扩展性的要求。在基于Web的架构中，数据库是最难以进行横向扩展的，当一个应用系统的用户数量以及访问量与日俱增的时候，数据库却无法像WebServer或者AppServer那样简单地通过添加更多的硬件和服务结点来扩展性能和负载能力。对于许多需要进行24h不间断服务的网站来说，对数据库系统进行升级和扩展是非常痛苦的事情，往往需要进行停机维护或者数据迁移。NoSQL数据库的产生无疑让人们看到解决问题的希望。

当然，这里NoSQL诞生的目的并不是为了取代传统的关系数据库，NoSQL也并不能取代关系数据库的地位。对于用户的基本信息等重要信息还是需要存储到关系数据库中。

NoSQL 与传统关系数据库并存，是当下时代的需求。

3.4.2　NoSQL 数据库与关系数据库的比较

NoSQL 数据库和传统关系数据库都有各自的特点，下面从它们各自的特性出发，分析其优缺点。

1. 关系数据库的优势及存在的问题

关系数据库是建立在关系模型基础上的数据库，凭借集合代数等数学方法来处理数据库数据。把所有的数据都通过行列的二元关系表示出来，给人以直观的感受。现实世界各种实体以及实体与实体之间的关系都可以用关系模型来表示。

关系数据库具有以下优势。

(1) 适合存储结构化数据。

(2) 关系模型的二维结构接近人的逻辑思维，容易理解。

(3) 关系数据库形式规整，代数理论完整，便于数据的维护。

(4) 数据的规模与增长可以预期。

(5) 事务性强，数据一致性强，具有高稳定性。

(6) 技术成熟，有大量成功的应用案例。

凭借关系数据库的诸多优势，在 20 世纪 90 年代的互联网领域建立了庞大的应用市场，积累了大量的成功案例，同时促进了互联网的快速发展。也正是伴随着互联网的快速发展，特别是 Web 2.0 网站的快速发展，在应对这些超大规模和高并发的应用时，传统的关系数据库就遇到了很多难以克服的问题。

2. NoSQL 数据库的优势及存在的问题

NoSQL 的出现，弥补了关系数据库的很多缺陷。由于 NoSQL 数据库是基于 CAP 理论和 BASE 思想的，在没有强一致性的要求下，它对于非结构化数据和海量数据的处理有着极高的性能。

NoSQL 数据库具有以下优势。

(1) 数据模型简单且灵活。NoSQL 可以随时存储自定义的数据格式，不需要像关系数据库一样要事先定义。

(2) NoSQL 更容易扩展。与关系数据库不同，NoSQL 数据库数据之间无关系，在系统运行时，可以动态地添加或者删除结点，不需要停机维护，数据可以自动迁移。

(3) 数据库结构简单、容量更大，性能更高。

(4) 按照 Key，很容易映射复杂值的环境，且有较高的获取数据效率。

(5) 数据分区，提高了并行性，分区的同时进行了复制，防止了数据失效的问题。

(6) 和 RAID 存储系统不同的是，NoSQL 中的复制，往往是基于日志的异步复制。这样，数据就可以尽快地写入一个结点，而不会由于网络传输引起迟延。

如今很多的 NoSQL 数据存储系统都已被部署于实际应用中，表现出了不错的实用效果，但是随着其逐步的应用和研究的不断深入，NoSQL 数据库也出现了很多挑战性的问题。

(1) NoSQL 数据库是面向应用的，且绝大多数都是面向特定应用的自构建的开源项

目，没有权威的数据库厂商提供强有力的商业支持，缺乏通用性。一旦 NoSQL 数据库的相关产品出现故障，就只能靠自己解决，需要承担一定的技术风险。

（2）NoSQL 数据库的应用成熟度不高，实际应用较少，已有产品支持的功能有限（不支持事务特性），导致其应用具有一定的局限性。

（3）没有强一致性约束，有些场景无法适用，这也是其无法代替关系数据库的原因之一。

（4）由于缺乏类似关系数据库所具有的强有力的理论，数据库的设计很难体现业务的实际情况，也增加了数据库设计的难度。

（5）目前为止，HBase 数据库是安全特性最完善的 NoSQL 数据库产品之一，而其他的 NoSQL 数据库多数没有提供内建的安全机制，但随着 NoSQL 的发展，越来越多的人开始意识到安全的重要，部分 NoSQL 产品逐渐开始提供一些安全方面的支持。

3. NoSQL 数据库和传统关系数据库相结合

NoSQL 数据库和传统关系数据库都有各自的优势和问题。基于它们的适用范围不同，目前主流架构采用关系数据库（如 MySQL）＋NoSQL 的组合方案。目前为止，还没有出现一个能够适用各种场景的数据库，而且根据 CAP 理论，这样的数据库是不存在的。

在强一致性和高可用性的场景下，数据库基于 ACID 特性；而在高可用性和扩展性场景下，数据库采用 BASE 思想。NoSQL 数据库可以弥补关系数据库的一些缺陷，但是目前还是无法取代关系数据库，将两者结合起来使用，各取所长，才是应对当下海量数据处理问题的正确方式。

3.4.3 NoSQL 数据库的 4 大类型

目前为止，NoSQL 数据库已经超过 200 种了，对比传统的关系数据库，NoSQL 数据库主要分为以下几种：键值（Key-Value）存储数据库、列存储（Column-Orientated）数据库、文档型（Document-Oriented）数据库、图形（Graph-Oriented）数据库。

1. 键值存储数据库

键值数据库是最常用的 NoSQL 数据库，类似哈希表，它的数据是以键值对（Key-Value）的形式保存的。键值数据库的优势在于简单、易部署。但是如果只对部分值进行查询或更新的时候，键值数据库就显得效率低下了。

适用的场景：存储用户信息，如会话、配置文件、参数、购物车等。这些信息一般都和 ID（键）挂钩，这种情景下键值数据库是一个很好的选择。

不适用场景：

（1）取代通过键查询，而是通过值来查询。键值数据库中根本没有通过值查询的途径。

（2）需要存储数据之间的关系。在键值数据库中不能通过两个或以上的键来关联数据。

（3）事务的支持。在键值数据库中故障产生时不可以进行回滚。

2. 列存储数据库

一般的数据库都以行为单位，这样可以提高数据的读入性能。但如果要一次读取若干行中的很多列，则将所有行的某一列作为基本数据存储单元的存储效果会更好，列存储数据库也因此得名。列存储数据库将数据存储在列族中，一个列族存储经常被一起查询的相关数据。例如，有一个 Person 类，通常会一起查询其姓名和年龄而不是薪资。在这种情况下，姓名和年龄就会被放入一个列族中，而薪资则在另一个列族中。

适用的场景：

(1) 日志。可以将数据存储在不同的列中，每个应用程序可以将信息写入自己的列族中。

(2) 博客平台。每条信息都被存储到不同的列族中。例如，标签可以存储在一个列族，类别可以存储在一个列族，而文章则存储在另一个列族。

不适用场景：

(1) 需要 ACID 事务。有些列存储数据库不支持事务。

(2) 原型设计。列存储数据库的数据结构是基于人们期望的数据查询方式而定的。在模型设计之初，人们根本不可能去预测它的查询方式，而一旦查询方式改变，人们就必须重新设计列族。

3. 文档型数据库

文档型数据库是一种用来管理文档的数据库，它与传统数据库的不同在于，其处理的基本单位是文档。每个文档都是自包含的数据单元，是一系列数据项的集合，可长可短，甚至可以无结构。每个数据项都有一个名称与对应的值，值既可以是简单的数据类型，如字符串、数字和日期等；也可以是复杂的类型，如有序列表和关联对象。文档型数据库可以看作键值数据库的升级版，允许之间嵌套键值。而且文档型数据库比键值数据库的查询效率更高。

适用的场景：

(1) 日志。企业环境下，每个应用程序都有不同的日志信息。

(2) 分析。鉴于它的弱模式结构，不改变模式就可以存储不同的度量方法及添加新度量。

不适用场景：在不同的文档上需要添加事务。

4. 图形数据库

图形结构的数据库同其他行列以及刚性结构的 SQL 数据库不同，它是使用灵活的图形模型，并且能够扩展到多个服务器上。图形数据库以实体为顶点，实体间的关系作为边建图。对于很多应用领域来说，其事物对象本来就是一个图结构，如社交网络与交通运输网络。对于这些应用，使用图形结构的数据库进行存储就比较方便。

适用的场景：

(1) 一些关系性强的数据中。

(2) 推荐引擎。如果人们将数据以图的形式表现，将会非常有益于推荐的制定。

不适用场景：非图形的数据模型。图形数据库的适用范围很小，因为很少有操作涉及整个图。

NoSQL 数据库的 4 大类型的分析如表 3.1 所示。

表 3.1 NoSQL 数据库的 4 大类型

分类	典型代表	典型应用场景	数据模型	优点	缺点
键值存储数据库	Tokyo Cabinet/Tyrant、Redis、Voldemort、OracleBDB	内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等	Key 指向 Value 的键值对，通常用 Hashtable 来实现	查找速度快	数据无结构化，通常只被当作字符串或者二进制数据
列存储数据库	Cassandra、HBase、Riak	分布式文件系统	以列族式存储，将同一列数据存在一起	查找速度快，可扩展性强，更容易进行分布式扩展	功能相对局限
文档型数据库	CouchDB、MongoDb	Web 应用（与 Key-Value 类似，Value 是结构化的）	Key-Value 对应的键值对，Value 为结构化数据	数据结构要求不严格，表结构可变，不需要像关系数据库一样需要预先定义表结构	查询性能不高，而且缺乏统一的查询语法
图形数据库	Neo4J、InfoGrid、Infinite Graph	社交网络、推荐系统等。专注于构建关系图谱	图结构	利用图结构相关算法，如最短路径寻址、N 度关系查找等	很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群方案

3.5 大数据存储与管理技术

云计算技术、物联网等技术快速发展，多样化已经成为数据信息的一项显著特点，为充分发挥信息应用价值，有效存储已经成为人们关注的热点。

为了有效应对现实世界中复杂多样的大数据处理需求，需要针对不同的大数据应用特征，从多个角度、多个层次对大数据进行存储和管理。下面介绍大数据存储的一些关键技术：分布式存储技术、虚拟化技术和云存储技术。其中，分布式存储技术和虚拟化技术是云存储的核心技术。

3.5.1 分布式存储技术

传统的集中式存储对搭建和管理的要求较高。由于硬件设备的集中存放，机房的空间、散热和承重等都有严格的要求；存储设备要求性能较好，对主干网络的带宽也有较高的要求。

而在信息爆炸的时代，人们可以获取的数据呈指数级增长，单纯在固定某个地点进行硬盘的扩充在容量大小、扩充速度、读写速度和数据备份等方面的表现都无法达到要求；而且大数据处理系统的数据多来自于客户，数据的种类多，存储系统需要存储各种半结构化、非

结构化的数据，如文档、图片、视频等，因此大数据的存储宜使用分布式文件系统来管理这些非结构化数据。

1. 分布式存储概念

分布式数据存储，即存储设备分布在不同的地理位置，数据就近存储，带宽上没有太大压力。可采用多套低端的小容量的存储设备分布部署，设备价格和维护成本较低。小容量设备分布部署，对机房环境要求也较低。分布式数据存储将数据分散在多个存储结点上，各个结点通过网络相连，对这些结点的资源进行统一的管理。

传统的分布式计算系统中通常计算结点与存储结点是分开的。当执行计算任务时，首先要把数据从数据结点传输至计算结点（数据向计算迁移），这种处理方式会使外存文件数据 I/O 访问成为一个制约系统性能的瓶颈。为了减少大数据并行计算系统中的数据通信开销，应当考虑将计算向数据靠拢和迁移。例如，MapReduce 模型采用了数据/代码互定位的技术方法，该方法让计算结点首先尽量负责计算其本地存储的数据，以发挥数据本地化特点；仅当结点无法处理本地数据时，再采用就近原则寻找其他可用计算结点，并把数据传送到该可用计算结点。

面对目前互联网应用 PB 级的海量存储的存储需求，频繁的数据传输，都是通过应用分布式存储系统，实现在普通 PC 上部署结点，通过系统架构设计提供强大的容错能力，针对大型的、分布式的、大量数据访问的应用给用户提供总体性能最高的服务。

2. 分布式存储系统的特性

（1）高可靠性。这是存储系统的基本保障，既要保证数据在读写过程中不能发生错误，同时还要保证数据进入系统后硬件失效不会导致数据丢失。

（2）可扩展性。分布式存储系统可以扩展到几百台甚至几千台这样的一个集群规模，系统的整体性能线性增长。系统必须具有一定的自适应管理功能，能够根据数据量和计算的工作量估算所需要的结点个数，并动态地将数据在结点间迁移，以实现负载均衡。同时，结点失效时，数据必须可以通过副本等机制进行恢复，不能对上层应用产生影响。

（3）高性能。存储系统软件的实现需要释放硬件技术进步带来的性能提升。现在高速存储设备在不断降低延迟，增加吞吐，如果还使用传统的 TCP 网络和内核的 CPU 调度，将不能充分发挥硬件的性能。

（4）低成本。分布式存储系统的自动容错、自动负载均衡的特性，允许分布式存储系统可以构建在低成本的服务器上。线性的扩展能力也得以增加，实现分布式存储系统的自动运维。

（5）易用性。分布式存储系统需要对外提供方便易用的接口，另外，也需要具备完善的监控、运维工具，并且可以方便地与其他系统进行集成。

3. 分布式存储系统的关键技术

实现分布式存储系统的关键主要在于数据、状态信息的持久化，要求在自动迁移、自动容错、并发读写的过程中保证数据的一致性。其主要关键技术如下。

（1）数据划分。在分布式环境中，数据存储在多个存储单元，数据的划分是影响系统性能、负载均衡以及扩展性的关键因素之一。系统必须在用户请求到来之前完成数据的合理分发，才能降低系统时延。

（2）数据冗余技术。分布式存储系统需要使用多台服务器共同存储数据，而随着服务器数量的增加，服务器出现故障的概率也在不断增加。要保证在有服务器出现故障的情况下系统仍然可用，通过牺牲一定的数据一致性，采用异步复制的方式确保数据的可用性。

（3）容错。如何快速检测到服务器故障，并自动地将在故障服务器上的数据进行迁移？结点的失效侦测和失效恢复是关键技术。

（4）负载均衡。新增的服务器要在集群中保障负载均衡，数据迁移过程中要保障不影响现有的服务。

（5）事务并发控制技术。实现分布式事务并发控制及其事务故障恢复。

3.5.2 虚拟化技术

虚拟化技术和分布式存储技术是云存储的基础技术。虚拟化技术是将人们可以用到的资源组成资源池来创建虚拟化的资源提供给用户。虚拟化技术主要应用在基础设施即服务的服务模式中，大多数资源都可以通过虚拟化技术对其进行统一管理。

虚拟化技术的发展离不开计算机技术的发展。虚拟化技术的目的是简化管理、优化资源，通过将实际环境运行的软件、硬件各种资源放在虚拟的环境来运行，透明化底层物理硬件，从而最大化地利用物理硬件。

1. 虚拟化技术的概念

虚拟化技术是一门应用十分广泛的基础技术。其作用如下：虚拟化技术是一种逻辑简化技术，实现物理层向逻辑层的变化。当一个系统采用虚拟化技术后，其对外表现的就是一种逻辑结构，人们无须了解其内部结构就可以通过其相对简单的逻辑结构对其进行使用。

在计算机领域，虚拟化技术是将计算机的各种实体资源，如服务器、网络、内存及存储等，通过抽象、转换后呈现出来，打破实体结构间不可切割的障碍，使用户可以比原本的组态更好的方式来应用这些资源。这些资源的新虚拟部分是不受现有资源的架设方式、地域或物理组态所限制的，从而最大化地利用物理硬件。

总的来看，虚拟化技术包含如下三个方面。

（1）虚拟化的对象是各种各样的资源。

（2）经过虚拟化后的逻辑资源对用户隐藏了不必要的细节。

（3）用户可以在虚拟环境中实现其在真实环境中的部分或者全部功能。

2. 虚拟化技术的发展

虚拟化技术随着计算机发展的需求也在不停地发展。早期的计算机价格昂贵，硬件利用率低，提出了分时系统来提高硬件利用率。为了实现分时系统，克里斯托弗（Christopher Strachey）提出了虚拟化的概念，发表了一篇名为《大型高速计算机中的时间共享》的学术报告。随后10年里，为了使硬件资源得到充分利用，IBM发明了一种操作系统虚拟机技术，可以使用户在一台主机上运行多个操作系统。而后随着科技水平的提高，计算机硬件资源价格降低，VMware率先实施了以虚拟机监视器为中心的软件解决方案，之后Intel公司和AMD公司在其x86处理器中增加了硬件虚拟化功能。自从2008年以后，云计算技术的发展推动了虚拟化技术成为研究热点。由于虚拟化技术能够屏蔽底层的硬件环境，充分利用

计算机的软硬件条件，使之成为切分型云计算技术的核心技术。

目前来看，通过服务器虚拟化技术实现资源整合是主要发展驱动力。对于服务器，虚拟化技术发展的热点在安全、存储和管理等方面。而在技术应用方面，虚拟化的性能、环境的部署、设备的兼容等方面是关键。

3. 虚拟化技术的分类

虚拟化技术已经比较成熟，其形式多种多样，实现的应用也形成体系。按照不同的应用领域可将虚拟化技术分为应用虚拟化、桌面虚拟化、服务器虚拟化、网络虚拟化、存储虚拟化。

1）应用虚拟化

应用虚拟化安装在一个虚拟的环境里，拥有与应用程序相关的所有共享资源，极大地方便应用程序的部署、更新和维护。通常应用虚拟化与应用程序生命周期管理结合起来。

2）桌面虚拟化

桌面虚拟化将大量的终端资源集中到后台数据中心，方便对用户的众多终端统一管理，实现资源的调配，终端用户可以通过登录认证、访问数据中心、自取数据，完成自己的业务，极大地提高了数据使用的灵活性。

3）服务器虚拟化

服务器虚拟化是指能够在一台物理服务器上运行多台虚拟服务器的技术，多个虚拟服务器之间的数据是隔离的，虚拟服务器可控地占有物理资源。被虚拟出来的服务器称为虚拟机（Virtual Machine，VM）。

4）网络虚拟化

网络虚拟化是基础设施即服务的基础。网络虚拟化让一个物理网络可以支持多个逻辑网络，虚拟化保留了网络设计中原有的层次结构、数据通道和所能提供的服务，让使用虚拟化网络的用户体验和独享物理网络一样，同时网络虚拟化技术还有效地提高了网络资源的利用率。常见的网络虚拟化技术如 VPN、VLAN。

5）存储虚拟化

存储虚拟化也是基础设施即服务的基础。存储虚拟化技术通过将底层设备进行抽象化统一管理，对服务器层屏蔽了底层存储设备的异构性、特殊性，只保留了其统一的逻辑特性，从而实现了存储系统资源的集中。存储虚拟化是存储整合的重要组成部分，它能减少管理问题，提高存储利用率，降低新增存储的费用。存储虚拟化技术主要分为基于主机的虚拟化存储、基于网络的虚拟化存储、基于存储设备的虚拟化存储。

3.5.3　云存储技术

云存储不是指存储数据的设备，而是一种服务，具体来说，它是把数据存储和访问作为一种服务，并通过网络提供给用户。云存储与云计算不同，云计算提供计算能力，而云存储提供存储能力。

云存储是在云计算概念上延伸和衍生发展出来的一个新的概念。云存储专注于向用户提供以网络为基础的在线存储服务，它是指通过集群应用、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，保证数据的安全性，并节约存储空间。用户可以在任

何时间、任何地方，通过任何可联网的装置连接到云上方便地存取。数据用户无须考虑存储的各个细节，如存储容量、存储类型、存储格式、存储的安全与完整等问题，只需按需付费就可以从云存储中获取近乎无限大的存储空间和企业级的服务质量。

1. 云存储的特性

云存储具有以下特性，用来解决海量数据的增长而带来的存储难题。

1）动态可扩展

理论上，云存储就有无限的可扩展性，可以满足不断增加的数据存储需求。可以随着容量的增长，线性地扩展性能和存取速度。

2）可用性高

具有数据存储的高度适配性和自我修复能力，当一个结点失效了，其余的结点会接管未完成的事务。并且数据的保存是冗余的，在不同的地理位置之间会有数据备份，用来提高数据的容错能力。

3）使用方便

云存储的最大优点之一是允许用户在任何时间、任何地点、任何设备上自由共享和访问数据，并且支持多种数据的存储格式。只需保持网络连接，就可以享受到高质量的数据存储服务。可以享用自动更新的存储技术并且不需要用户来维护。

4）服务性能高

云存储通过大型分布式服务集群可以提供强大的服务质量，支持海量数据访问，数据部署速度快，备份速度快，数据安全性强，提供强大的数据计算能力。

5）价格较低

云存储采用多租用模式，同时为多个用户提供服务，所有用户共享存储资源，降低了资源浪费。用户采用“按需付费”的方式来使用云存储提供的各种服务，比在本地存储数据具有更高的性价比。

2. 云存储的分类

按照云存储的所有者来说，可以将云存储分为公共云存储、私有云存储、混合云存储3类。

1）公共云存储

公共云存储的供应商提供大量的文件存储。供应商可以保持每个客户的存储、应用都是独立的、私有的。所有的组件都是放在共享的基础存储设施里，设置在用户端的防火墙外部，用户直接通过安全的互联网访问。公共云存储可以直接通过增加服务器来增加存储空间。常见的公共云存储如亚马逊的云存储服务、百度云盘、360云盘、搜狐企业云盘等。

2）私有云存储

私有云实现的功能和公有云类似，不同点是它为某一企业或社会团体独有，位于企业防火墙内部，可以使用所有授权的硬件和软件。私有云相对保密性、安全性更高，不过使用费用和维护费用也更高一些。

3）混合云存储

这种云存储把公共云和私有云结合在一起。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云可以帮助公司面对迅速

增长的负载波动或高峰。相对而言，混合云存储对云存储的分配带来一定的复杂性。

3. 云存储实现技术

云存储实现技术包括两个层面：分布式存储层以及存储访问层。

分布式存储层管理存储服务器集群，实现各个存储设备之间的协同工作，保证数据可靠性，对外屏蔽数据所在位置、数据迁移、数据复制、机器增减等变化，使得整个分布式系统看起来像一台服务器。分布式存储层是云存储系统的核心，也是整个云存储平台中最难实现的部分。CDN 结点将云存储系统中的热点数据缓存到离用户最近的位置，从而减少用户的访问延时并节约带宽。

存储访问层位于分布式存储层的上一层，该层的主要作用是将分布式存储层的客户端接口封装为 WebService(基于 RESTful 等协议)服务。另外，该层通过调用公共服务实现用户认证、权限管理以及计费等功能。存储访问层不是必需的，云存储平台中的计算实例也可以直接通过客户端 API 访问分布式存储层中的存储系统。

习　　题

1. 简述大数据的数据类型。

2. 简述数据管理技术的发展分为哪些阶段，每个阶段的特点是什么？

3. 数据库系统中的数据模型有哪几种？

4. 简述分布式系统的 CAP 理论和 BASE 思想。BASE 和 ACID 的各自特点和适用场景是什么？

5. 简述 NoSQL 数据库的 4 大类型。

6. 简述分布式存储系统。

7. 简述虚拟化技术。

8. 简述云存储技术。