获取教材和讲义 PPT 等各种课程资料请访问 http://dblab.xmu.edu.cn/node/422

=课程教材由林子雨老师根据网络资料编著=

厦门大学计算机科学系教师 林子雨 编著

http://www.cs.xmu.edu.cn/linziyu

2013 年 9 月

前言

本教程由厦门大学计算机科学系教师林子雨编著，可以作为计算机专业研究生课程《大数据技术基础》的辅助教材。

本教程的主要内容包括：大数据概述、大数据处理模型、大数据关键技术、大数据时代面临的新挑战、NoSQL 数据库、云数据库、Google Spanner、Hadoop、HDFS、HBase、MapReduce、Zookeeper、流计算、图计算和 Google Dremel 等。

本教程是林子雨通过大量阅读、收集、整理各种资料后精心制作的学习材料，与广大数据库爱好者共享。教程中的内容大部分来自网络资料和书籍，一部分是自己撰写。对于自写内容，林子雨老师拥有著作权。

本教程 PDF 文档及其全套教学 PPT 可以通过网络免费下载和使用（下载地址：http://dblab.xmu.edu.cn/node/422）。教程中可能存在一些问题，欢迎读者提出宝贵意见和建议！

本教程已经应用于厦门大学计算机科学系研究生课程《大数据技术基础》，欢迎访问 2013 班级网站 http://dblab.xmu.edu.cn/node/423。

林子雨的 E-mail 是：ziyulin@xmu.edu.cn。

林子雨的个人主页是：http://www.cs.xmu.edu.cn/linziyu。

林子雨于厦门大学海韵园

2013 年 9 月

第 11 章 云数据库

厦门大学计算机科学系教师 林子雨 编著

个人主页：http://www.cs.xmu.edu.cn/linziyu

课程网址：http://dblab.xmu.edu.cn/node/422

2013 年 9 月

第 11 章 云数据库

云数据库是在 SaaS（Software-as-a-Service：软件即服务）成为应用趋势的大背景下发展起来的云计算技术，它极大地增强了数据库的存储能力，消除了人员、硬件、软件的重复配置，让软、硬件升级变得更加容易，同时，也虚拟化了许多后端功能。云数据库具有高可扩展性、高可用性、采用多租形式和支持资源有效分发等特点。可以说，云数据库是数据库技术的未来发展方向。

本章介绍云数据库的相关知识，内容要点如下：

- 云数据库概述
- 云数据库的特性
- 云数据库是海量存储需求的必然选择
- 云数据库与传统的分布式数据库
- 云数据库的影响
- 云数据库产品
- 数据模型
- 数据访问方法
- 编程模型

11.1 云数据库概述

11.1.1 云计算和 SaaS

云计算（Cloud Computing）是分布式计算（Distributed Computing）、并行计算（Parallel Computing）、效用计算（Utility Computing）、网络存储（Network Storage Technologies）、虚拟化（Virtualization）、负载均衡（Load Balance）等传统计算机和网络技术发展融合的产物。

云计算由一系列可以动态升级和被虚拟化的资源组成，这些资源被所有云计算的用户共享并且可以方便地通过网络访问，用户无需掌握云计算的技术，只需要按照个人或者团体的需要租赁云计算的资源。云计算是继 1980 年代大型计算机到客户端-服务器的大转变之后的

又一种巨变。云计算的出现并非偶然，早在上世纪 60 年代，麦肯锡就提出了把计算能力作为一种像水和电一样的公用事业提供给用户的理念，这成为云计算思想的起源。在 20 世纪 80 年代网格计算，90 年代公用计算，21 世纪初虚拟化技术、SOA、SaaS 应用的支撑下，云计算作为一种新兴的资源使用和交付模式已经被学界和产业界所广泛认知和接受。中国云发展创新产业联盟评价云计算为“信息时代商业模式上的创新”。

云计算包括三种主要类型，即 IaaS（Infrastructure as a Service）、PaaS（Platform as a Service）和 SaaS（Software as a Service）：

- **IaaS（Infrastructure as a Service）**

IaaS（Infrastructure as a Service），即“基础设施即服务”。提供给消费者的服务是对所有设施的利用，包括处理、存储、网络和其它基本的计算资源，用户能够部署和运行任意软件，包括操作系统和应用程序。消费者不管理或控制任何云计算基础设施，但能控制操作系统的选择、储存空间、部署的应用，也有可能获得有限制的网络组件（例如，防火墙、负载均衡器等）的控制。

在没有 IaaS 之前，如果你想在办公室或者公司的网站上运行一些企业应用，你需要去买服务器或者别的昂贵的硬件来控制本地应用，让你的业务运行起来。但是，现在有了 IaaS，你可以将硬件外包到别的地方去。IaaS 公司会提供场外服务器、存储和网络硬件，你可以租用这些基础设施，从而节省了维护成本和办公场地，公司可以在任何时候利用这些硬件来运行其应用。一些大的 IaaS 公司包括 Amazon、Microsoft、VMWare、Rackspace 和 Red Hat。

- **PaaS（Platform as a Service）**

PaaS（Platform as a Service），即“平台即服务”。提供给消费者的服务是，把客户采用所提供的开发语言和工具（例如 Java，python, .Net 等）开发的、或收购的应用程序部署到供应商的云计算基础设施上去。客户不需要管理或控制底层的云基础设施，包括网络、服务器、操作系统、存储等，但客户能控制部署的应用程序，也可能控制运行应用程序的托管环境配置。

你公司所有的开发都可以在这一层进行，节省了时间和资源。PaaS 公司在网上提供各种开发和分发应用的解决方案，比如虚拟服务器和操作系统。这节省了你在硬件上的费用，也让分散的工作室之间的合作变得更加容易。一些大的 PaaS 提供者有 Google App Engine、Microsoft Azure、Force.com、Heroku、Engine Yard。最近兴起的公司有 AppFog、 Mendix 和

Standing Cloud。

- **SaaS（Software as a Service）**

SaaS（Software as a Service），即“软件即服务”。它是一种通过 Internet 提供软件的模式，厂商将应用软件统一部署在自己的服务器上，客户可以根据自己实际需求，通过互联网向厂商定购所需的应用软件服务，按定购的服务多少和时间长短向厂商支付费用，并通过互联网获得厂商提供的服务。用户不用再购买软件，而改用向提供商租用基于 Web 的软件，来管理企业经营活动，且无需对软件进行维护，服务提供商会全权管理和维护软件。对于许多小型企业来说，SaaS 是采用先进技术的最好途径，它消除了企业购买、构建和维护基础设施和应用程序的需要。

在这种模式下，客户不再像传统模式那样花费大量投资用于硬件、软件、人员，而只需要支出一定的租赁服务费用，通过互联网便可以享受到相应的硬件、软件和维护服务，享有软件使用权和不断升级服务；公司上项目不用再像传统模式一样需要大量的时间用于布置系统，多数经过简单的配置就可以使用。这是网络应用最具效益的营运模式。

Salesforce 是 SaaS 厂商的先驱，它一开始提供的是可通过网络访问的销售力量自动化应用软件。在该公司的带动下，其他 SaaS 厂商已如雨后春笋般蓬勃而起。

11.1.2 云数据库概念

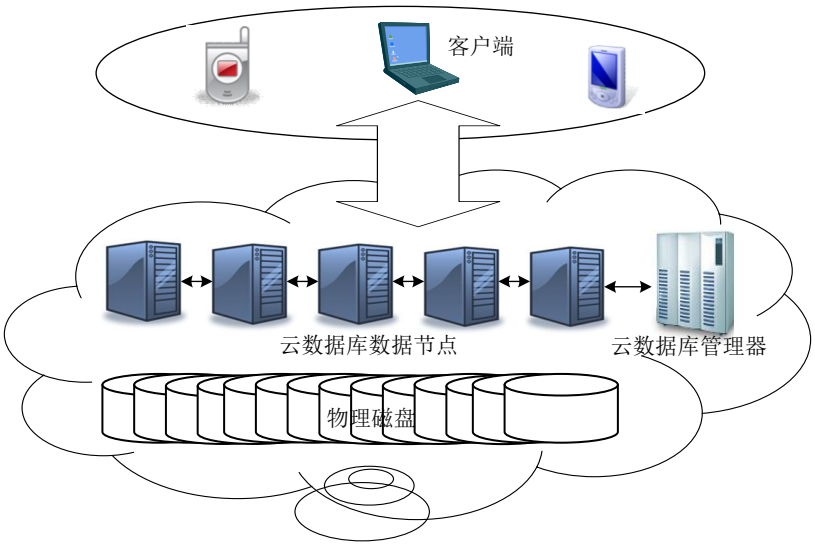

图 11-1 云数据库应用示意图

云数据库是在 SaaS（Software-as-a-Service：软件即服务）成为应用趋势的大背景下发展起来的云计算技术，它极大地增强了数据库的存储能力，消除了人员、硬件、软件的重复配置，让软、硬件升级变得更加容易，同时，也虚拟化了许多后端功能。云数据库具有高可扩展性、高可用性、采用多租形式和支持资源有效分发等特点。可以说，云数据库是数据库技术的未来发展方向。目前，对于云数据库的概念界定不尽相同，本文采用的云数据库定义是：云数据库是部署和虚拟化在云计算环境中的数据库。

如图 11-1 所示，在云数据库应用中，客户端不需要了解云数据库的底层细节，所有的底层硬件都已经被虚拟化，对客户端而言是透明的，它就像在使用一个运行在单一服务器上的数据库一样，非常方便容易，同时又可以获得理论上近乎无限的存储和处理能力。

11.2 云数据库的特性

云数据库具有以下特性：

（1）**动态可扩展**：理论上，云数据库具有无限可扩展性，可以满足不断增加的数据存储需求。在面对不断变化的条件时，云数据库可以表现出很好的弹性。例如，对于一个从事产品零售的电子商务公司，会存在季节性或突发性的产品需求变化，或者对于类似 Animoto 的网络社区站点，可能会经历一个指数级的增长阶段，这时，就可以分配额外的数据库存储资源来处理增加的需求，这个过程只需要几分钟。一旦需求过去以后，就可以立即释放这些资源。

（2）**高可用性**：不存在单点失效问题。如果一个节点失效了，剩余的节点就会接管未完成的事务。而且，在云数据库中，数据通常是复制的，在地理上也是分布的。诸如 Google、Amazon 和 IBM 等大型云计算供应商，具有分布在世界范围内的数据中心，通过在不同地理区间内进行数据复制，可以提供高水平的容错能力。例如，Amazon SimpleDB 会在不同的区间内进行数据复制，因此，即使整个区域内的云设施发生失效，也可以保证数据继续可用。

（3）**较低的使用代价**：通常采用多租户（multi-tenancy）的形式，这种共享资源的形式对于用户而言可以节省开销；而且用户采用按需付费的方式使用云计算环境中的各种软、硬件资源，不会产生不必要的资源浪费。另外，云数据库底层存储通常采用大量廉价的商业服务器，这也大大降低了用户开销。

（4）**易用性**：使用云数据库的用户不用控制运行原始数据库的机器，也不必了解它身

在何处。用户只需要一个有效的连接字符串就可以开始使用云数据库。

（5）**大规模并行处理**：支持几乎实时的面向用户的应用、科学应用和新类型的商务解决方案。

11.3 云数据库是海量存储需求的必然选择

云数据库在当前数据爆炸的大数据时代具有广阔的应用前景。根据 IDC 的研究报告，在未来的 5 年中，企业对结构化数据的存储需求会每年增加 20%左右，而对非结构化数据的存储需求将会每年增加 60%左右。在小规模应用的情况下，系统负载的变化可以由系统空闲的多余资源来处理，但是，在大规模应用的情况下，不仅存在海量的数据存储需求，而且应用对资源的需求也是动态变化的，这意味着大量虚拟机的增加或减少。对于这种情形，传统的关系数据库已经无法满足要求，云数据库成为必然的选择。换句话说，大数据时代的海量存储需求催生了云数据库。

11.4 云数据库与传统的分布式数据库

分布式数据库是计算机网络环境中各场地或节点上的数据库的逻辑集合。逻辑上它们属于同一系统，而物理上它们分散在用计算机网络连接的多个节点，并统一由一个分布式数据库管理系统管理。

分布式数据库已经存在很多年，它可以用来管理大量的分布存储的数据，并且通常采用非共享的体系架构。云数据库和传统的分布式数据库有着相似的地方，比如，都把数据存放到不同的节点上。但是，分布式数据库在可扩展性方面是无法和云数据库相比的。由于需要考虑数据同步和分区失败等开销，前者随着节点的增加，会导致性能快速下降。而后者则具有很好的可扩展性，因为后者在设计的时候，就已经避免了许多会影响到可扩展性的因素，比如采用更加简单的数据模型、对元数据和应用数据进行分离以及放松对一致性的要求等等。另外，在使用方式上，云数据库也不同于传统的分布式数据库，云数据库通常采用多租户模式，即多个租户共用一个实例，租户的数据既有隔离又有共享，从而解决数据存储的问题，同时也降低了用户使用数据库的成本。

11.5 云数据库的影响

云数据库的影响主要体现在以下几个方面：

（1）**极大地改变企业管理数据的方式**。Forrester Research 分析师 Noel Yuhanna 指出，18%的企业正在把目光投向云数据库。对于中小企业而言，云数据库可以允许他们在 WEB 上快速搭建各类数据库应用，越来越多的本地数据和服务将逐渐被转移到云中。企业用户可以在任意地点通过简单的终端设备，就可以对企业数据进行全面管理。此外，云数据库可以很好地支持企业开展一些短期项目，降低开销，而不需要企业为某个项目单独建立昂贵的数据中心。但是，云数据库的成熟仍然需要一段时间。中小企业会更多地采用云数据库产品，但是，对于大企业而言，云数据库并非首选，因为大企业通常自己建造数据中心。

（2）**催生新一代的数据库技术**。IDC 的数据库分析师 Carl Olofson 认为，云模型提供了无限的处理能力以及大量的 RAM，因此，云模型将会极大改变数据库的设计方式，将会出现第三代数据库技术。第一代是 20 世纪 70 年代的早期关系数据库，第二代是 80 到 90 年代的更加先进的关系模型。第三代的数据库技术，要求数据库能够灵活处理各种类型的数据，而不是强制让数据去适应预先定制的数据结构。事实上，从目前云数据库产品中的数据模型设计方式来看，已经有些产品（比如 SimpleDB、HBase、Dynamo、BigTable）放弃传统的行存储方式，而采用键/值存储，从而可以在分布式的云环境中获得更好的性能。可以预期的是，云数据库将会吸引越来越多的学术界的目光，该领域的相关问题也将成为未来一段时间内数据库研究的重点内容，比如云数据库的体系架构和数据模型等等。

（3）**数据库市场份额面临重新分配**。在过去的几十年里，数据库市场一直被诸如 Teradata、Oracle、IBM DB2、Microsoft SQL Server、Sybase 等传统数据库厂商所垄断。随着云数据库的出现和不断发展，市场将面临重新洗牌。首先，Amazon 和 Google 等原本并不从事数据库业务的国际知名企业，也乘着云计算的东风，开发了云中的数据库产品，加入这场新兴市场的角逐。实际上，对于云数据库市场而言，Amazon SimpleDB 和 Google BigTable 这类产品扮演了引领者的角色，传统的数据库厂商已然成为跟进者。其次，一些新的云数据库厂商开始出现，并且推出了具有影响力的产品，比如 Vertica 的 Analytic Database for the Cloud 和 EnterpriseDB 的 Postgres Plus in the Cloud。因此，数据库市场份额的重新分配不可避免。

11.6 云数据库产品

云数据库供应商主要分为三类：

- **传统的数据库厂商**：Teradata、Oracle、IBM DB2 和 Microsoft SQL Server；
- **涉足数据库市场的云供应商**：Amazon、Google 和 Yahoo；
- **新兴小公司**：Vertica、LongJump 和 EnterpriseDB。

就目前阶段而言，虽然一些云数据库产品，比如 Google BigTable、SimpleDB 和 HBase，在一定程度上实现了对于海量数据的管理，但是，这些系统暂时还不完善，只是“云数据库”的雏形。让这些系统支持更加丰富的操作以及更加完善的数据管理功能（比如复杂查询和事务处理），以满足更加丰富的应用，仍然需要研究人员的不断努力。

表 11-1 给出了市场上常见的云数据库产品，对于其中一些主要产品，下面我们会做简要介绍。

表 11-1 云数据库产品

企业	产品
Amazon	Dynamo、SimpleDB、RDS
Google	BigTable、FusionTable
Microsoft	Microsoft SQL Server Data Services 或 SQL Azure
Oracle	Oracle Cloud
Yahoo!	PNUTS
Vertica	Analytic Database v3.0 for the Cloud
EnerpriseDB	Postgres Plus in the Cloud
开源项目	HBase、Hypertable
其他	EnerpriseDB、FathomDB、ScaleDB、Objectivity/DB、M/DB:X

11.6.1 Amazon 的云数据库产品

Amazon 是云数据库市场的先行者。Amazon 除了提供著名的 S3 存储服务和 EC2 计算服

务以外，还提供基于云的数据库服务 Dynamo。Dynamo 采用“键/值”存储，其所存储的数据是非结构化数据，不识别任何结构化数据，需要用户自己完成对值的解析。Dynamo 系统中的键（key）不是以字符串的方式进行存储，而是采用 md5_key（通过 md5 算法转换后得到）的方式进行存储，因此，它只能根据 key 去访问，不支持查询。SimpleDB 是 Amazon 公司开发的一个可供查询的分布数据存储系统，它是 Dynamo“键/值”存储的补充和丰富。顾名思义，SimpleDB 的目的是作为一个简单的数据库来使用，它的存储元素（属性和值）是由一个 *id* 字段来确定行的位置。这种结构可以满足用户基本的读、写和查询功能。SimpleDB 提供易用的 API 来快速地存储和访问数据。但是，SimpleDB 不是一个关系型数据库，传统的关系型数据库采用行存储，而 SimpleDB 采用了“键/值”存储，它主要是服务于那些不需要关系数据库的 WEB 开发者。

Amazon RDS（Amazon Relational Database Service）是 Amazon 开发的一种 Web 服务，它可以让用户在云环境中建立、操作关系型数据库（目前支持 MySQL 和 Oracle 数据库）。用户只需要关注应用和业务层面的内容，而不需要在繁琐的数据库管理工作上耗费过多的时间。

此外，Amazon 和其他数据库厂商开展了很好的合作，Amazon EC2 应用托管服务已经可以部署很多种数据库产品，包括 SQL Server、Oracle 11g、MySQL 和 IBM DB2 等主流数据库平台，以及其他一些数据库产品，比如 EnerpriseDB。作为一种可扩展的托管环境，开发者可以在 EC2 环境中开发并托管自己的数据库应用。

11.6.2 Google 的云数据库产品

Google BigTable 是一种满足弱一致性要求的大规模数据库系统。Google 设计 BigTable 的目的，是为了处理 Google 内部大量的格式化及半格式化数据。目前，许多 Google 应用都是建立在 BigTable 之上的，比如 WEB 索引、Google Earth、Google Finance、Google Maps 和 Search History。BigTable 是构建在其他几个 Google 基础设施之上的。首先，BigTable 使用了分布式 Google 文件系统 GFS（Google File System）来存储日志和数据文件；其次，BigTable 依赖一个高可用的、持久性的分布式锁服务 Chubby；再次，BigTable 依赖一个簇管理系统来调度作业、在共享机器上调度资源、处理机器失败和监督机器状态。

但是，和 Amazon SimpleDB 类似，就目前而言，BigTable 实际上还不是真正的 DBMS

（Database Management System），它无法提供事务一致性、数据一致性。这些产品基本上可以被看成是云环境中的表单。

Google Cloud SQL 是谷歌公司推出的基于 MySQL 的云数据库，使用 Cloud SQL 的好处显而易见，所有的事务都在云中，并由谷歌管理，用户不需要配置或者排查错误，仅仅依靠它来开展工作即可。由于数据在谷歌多个数据中心中复制，因此它永远是可用的。谷歌还将提供导入或导出服务，方便用户将数据库带进或带出云。谷歌使用用户非常熟悉的 MySQL，因此多数应用程序不需过多调试即可运行，数据格式对于大多数开发者和管理员来说也是非常熟悉的。还有一个好处就是与应用引擎集成。

11.6.3 Microsoft 的云数据库产品

2008 年 3 月，微软通过 SQL Data Service（SDS）提供 SQL Server 的 RDBMS 功能，这使得微软成为云数据库市场上的第一个大型数据库厂商。此后，微软对 SDS 功能进行了扩充，并且重新命名为 SQL Azure。微软的 Azure 平台提供了一个 WEB 服务集合，可以允许用户通过网络在云中创建、查询和使用 SQL Server 数据库，云中的 SQL Server 服务器的位置对于用户而言是透明的。对于云计算而言，这是一个重要的里程碑。SQL Azure 具有以下特性：

- 属于关系型数据库：支持使用 TSQL（Transact Structured Query Language）来管理、创建和操作云数据库；
- 支持存储过程：它的数据类型、存储过程和传统的 SQL Server 具有很大的相似性，因此，应用可以在本地进行开发，然后部署到云平台上；
- 支持大量数据类型：包含了几乎所有典型的 SQL Server 2008 的数据类型；
- 支持云中的事务：支持局部事务，但是不支持分布式事务。

11.6.4 开源云数据库产品

HBase 和 Hypertable 利用开源 MapReduce 平台 Hadoop 提供了类似于 BigTable 的可伸缩数据库实现。MapReduce 是 Google 开发的、用来运行大规模并行计算的框架。采用 MapReduce 的应用更像一个人提交的批处理作业，但是，这个批处理作业不是在单个服务

器上运行，应用和数据都是分布在多个服务器上。Hadoop 是由 Yahoo 资助的一个开源项目，是 MapReduce 的开源实现，从本质上来说，它提供了一个使用大量节点来处理大规模数据集的方式。

HBase 已经成为 Apache Hadoop 项目的重要组成部分，并且已经在生产系统中得到应用。与 HBase 类似的是 Hypertable。不过，HBase 的开发语言是 Java，而 Hypertable 则采用 C/C++开发。与 HBase 相比，Hypertable 具有更高的性能。但是，HBase 不支持 SQL（Structual Query Language）类型的查询语言。

甲骨文开源数据库产品 BerkelyDB 也提供了云计算环境中的实现。

11.6.5 其他云数据库产品

Yahoo! PNUTS 是一个为网页应用开发的、大规模并行的、地理分布的数据库系统，它是 Yahoo!云计算平台重要的一部分。Vertica Systems 在 2008 年发布了云版本的数据库。10Gen 公司的 Mongo、AppJet 的 AppJet 数据库也都提供了相应的云数据库版本。M/DB:X 是一种云中的 XML 数据库，它通过 HTTP/REST 访问。FathomDB 旨在满足基于 Web 的公司提出的高传输要求，它所提供的服务更倾向于在线事务处理而不是在线分析处理。IBM 投资的 EnerpriseDB 也提供了一个运行在 Amazon EC2 上的云版本。LongJump 是一个与 Salesforce。com 竞争的新公司，它推出了基于开源数据库 PostgreSQL 的云数据库产品。Intuit QuickBase 也提供了自己的云数据库系列。麻省理工学院研制的 Relational Cloud 可以自动区分负载的类型，并把类型近似的负载分配到同一个数据节点上，而且采用了基于图的数据分区策略，对于复杂的事务型负载也具有很好的可扩展性，此外，它还支持在加密的数据上运行 SQL 查询。

11.7 数据模型

云数据库的设计可以采用不同的数据模型，不同的数据模型可以满足不用应用类型的需求，主要包括：键/值模型和关系模型。

11.7.1 键/值模型

BigTable、Dynamo、SimpleDB、PNUTS、HBase 等产品都采用了键/值模型存储数据。

下面我们以 Google BigTable 的数据模型为例来介绍键/值模型。

Google BigTable 的数据模型：BigTable 和它的同类开源产品 HBase，提供了一个不同于以往的简单、动态的、非关系型的数据模型。BigTable 采用了键/值数据模型。在 BigTable 中，包括行列以及相应的时间戳在内的所有数据都存放在表格的单元里。BigTable 的内容按照行来划分，多个行组成一个小表（Tablet），保存到某一个服务器节点中，这就意味着，每个 Tablet 包含了位于某个区间内的所有数据。对于 BigTable 而言，一个数据簇中存储了许多表，其中，每个表都是一个 Tablet 集合。在最初阶段，每个表只包含一个 Tablet。随着表的增长，它会被自动分解成许多 Tablet，每个 Tablet 默认尺寸大约是 100 到 200MB。BigTable 使用一个类似于 B+树的三层架构来存储 Tablet 位置信息。由于 BigTable 采用了键/值数据模型，因此，不存在表间的联接操作，这也使得数据分区操作相对简单，只需要根据键的区间来划分即可。

一个 BigTable 实际上就是一个稀疏的、分布的、永久的多维排序图，它采用行键盘（row key）、列键（column key）和时间戳（timestamp）对图进行索引。图中的每个值都是未经解释的字节数组：

- **行键**：BigTable 在行键上根据字典顺序对数据进行维护。对于一个表而言，行区间是根据行键的值进行动态划分的。每个行区间称为一个 Tablet，它是负载均衡和数据分发的基本单位，这些 Tablet 会被分发到不同的数据服务器上。
- **列键**：被分组成许多“列家族”的集合，它是基本的访问控制单元。存储在一个列家族当中的所有数据，通常都属于同一种数据类型，这通常意味着具有更高的压缩率。数据可以被存放到列家族的某个列键下面，但是，在把数据存放到这个列家族的某个列键下面之前，必须首先创建这个列家族。在创建完成一个列家族以后，就可以使用同一个家族当中的列键。
- **时间戳**：在 BigTable 中的每个单元格当中，都包含相同数据的多个版本，这些版本采用时间戳进行索引。BitTable 时间戳是 64 位整数。一个单元格的不同版本是根据时间戳降序的顺序进行存储的，这样，最新的版本可以被最先读取。

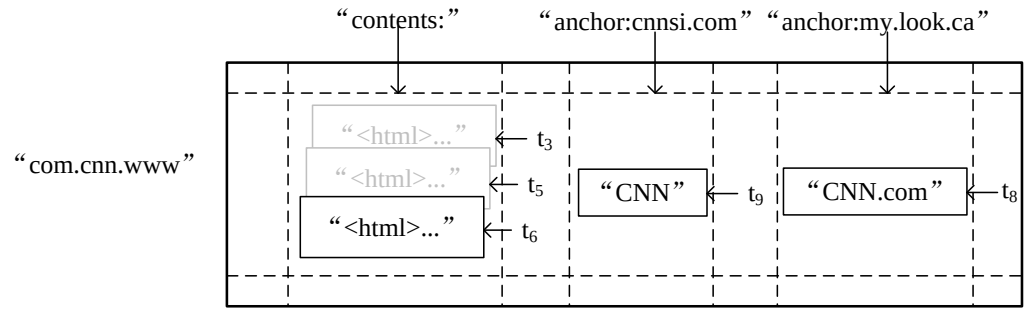

图 11-2 BigTable 数据模型的一个实例

这里以一个实例来阐释 BigTable 的数据模型。图 11-2 显示了存储了网页数据的 WebTable 的一个片段。行名称是反转的 URL，*contents* 列家族包含了网页内容，*anchor* 列家族包含了任何引用这个页面的 anchor 文本。CNN 的主页被 Sports Illustrated 和 MY-look 主页同时引用，因此，这里的行包含了名称为”anchor:cnnsi.com”和”anchor:my.look.ca”的列。每个 *anchor* 单元格都只有一个版本，*contents* 列有三个版本，分别对应于时间戳 t_3，t_5 和 t_6。

HBase 和 BigTable 一样，也采用了键/值数据模型，它是一个开放源码、分布式、面向列、多维、高可用、高性能的存储技术，采用 JAVA 语言编写。作为一个使用了 Hadoop 的分布式数据库，HBase 可以实现结构化数据的可靠存储。就像 Google BigTable 充分利用 Google File System 提供的分布式数据存储功能一样，HBase 的目的就是在 HDFS（Hadoop Distributed File System）之上提供类似 BigTable 的功能。HBase 采用多层索引表来执行键/值映射，获得了优越的主键查询性能。

HBase、BigTable 中的数据库模式和关系型模式具有很大的区别。第一，不存在表间的联接操作；第二，整个模式也只有一个索引——行键。与 RDBMS（Relational Database Management System）不同的是，开发者不需要使用 WHERE 从句的等价表达形式。通过设计，HBase 中的所有访问方法，或者通过行键访问，或者通过行键扫描，从而使得整个系统不会慢下来。由于 HBase 位于 Hadoop 框架之上，MapReduce 就可以用来生成索引表。

HBase 列家族可以被配置成支持不同类型的访问模式。一个家族也可以被设置成放入内存当中，以消耗内存为代价，从而换取更好的响应性能。

此外，Amazon Dynamo、SimpleDB 也都和 BigTable 一样采用了键/值存储。SimpleDB 中包含三个概念：domain、item 和 attribute，其中，domain 相当于一个 table，item 相当于一行，attribute 相当于一列，一列可以有多个值。但是，SimpleDB 和 BigTable 在数据划分的方式上存在一些差别：

- **SimpleDB 的数据划分**：采用静态数据划分方法，利用哈希函数把数据分发到多个数据节点，这使得 SimpleDB 更像一个哈希系统。这种方法的优点是，实现难度小；但是缺点也很明显，即用户的一些 domain 存放不连续。为了降低不连续数据存放对用户查询性能带来的负面影响，SimpleDB 对用户 domain 的大小进行限制，比如一个 domain 大小不超过 10GB。
- **BigTable 的数据划分**：采用动态划分方法，一个用户的数据可能会被划分成多个 Tablet，分发到不同的数据节点上。这种方法的优点是可以很好地支持负载均衡，缺点是需要相关的 Tablet 分拆与合并机制。

BigTable、Dynamo、SimpleDB、PNUTS、HBase 等产品虽然都采用了键/值模型存储数据，但是，这些系统在进行数据访问的时候都是以单个键作为访问粒度，这种方式对于一些网页应用而言无法取得好的性能，比如在线游戏、社区网络、合作编辑等包含合作性质的应用，这些应用需要对一个键组(key group)进行一致性的访问。由此，名为 G-Store 的键/值系统，并提出了“键组协议”来对一组键进行一致的、高效的访问，也就是说，在该系统中，数据访问的粒度不再是单个的键，而是一个键组。

11.7.2 关系模型

微软的 SQL Azure 云数据库采用了关系模型，它的数据分区方式和 BigTable 有些不同。关系型云数据库的数据模型涉及行组和表组等相关概念。

一个表是一个逻辑关系，它包含一个*分区键*，用来对表进行分区。具有相同分区键的多个表的集合，称为*表组*。在表组中，具有相同分区键值的多个行的集合，称为*行组*。一个行组中包含的行，总是被分配到同一个数据节点上。每个表组会包含多个行组，这些行组会被分配到不同的数据节点上。一个数据分区包含了多个行组。因此，每个数据节点都存储了位于某个分区键值区间内的所有行。

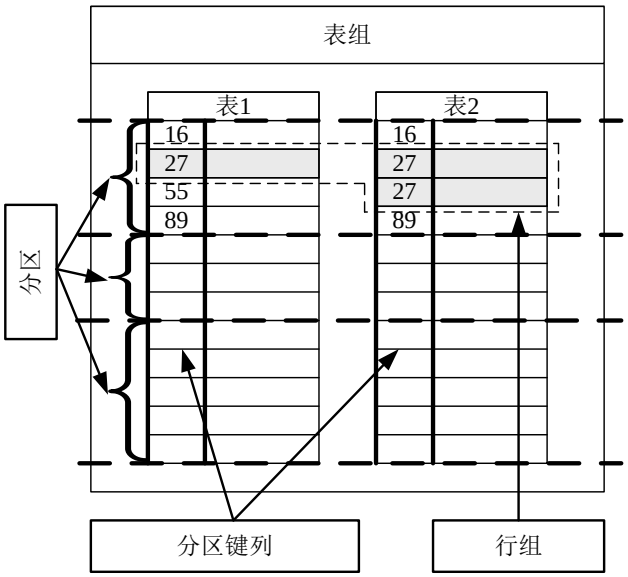

图 11-3 关系型云数据库的数据模型

这里以一个实例来解释关系型云数据库的数据模型。如图 11-3 所示，一个表组包含了两个相关的表，即图 11-3 中的表 1 和表 2。表 1 和表 2 分别包含了一个分区键列，每个分区键列是由多个分区键值组成的，这两个分区键列具有相同类型的分区键，都是 *ID* 类型的数值型数据。表 1 中的 *ID* 列和表 2 中的 *ID* 列存在主外键关联。因此，表 1 中和表 2 中的 *ID* 列值为 27 的三个行（图中用灰色底纹表示，并且用虚线框包围的部分），就属于一个行组。从图中也可以看出，表组被分割成多个分区。而且，同一行组中的内容都位于同一个数据分区内。

一个表组通常包含那些会被某个应用同时使用的多个表，因此，需要把这些表存储在一起，从而加快查询效率。比如，有两个表 *STUDENT_INFO* 和 *BOOKBORROWING_INFO*，前者记录的信息包括名字、性别、年龄、电话等等，而后者则记录了借书的相关信息。当图书馆管理员查询一个人的借书记录时，通常需要同时查看借阅者姓名、年龄、电话、借书数量、借书时间等信息，这就需要在 *STUDENT_INFO* 和 *BOOKBORROWING_INFO* 这两个表之间进行联接操作。很先让，如果把这两个表存储在同一个数据节点上，联接操作会快很多。

类似地，把一个行组中的多个行存储在一个数据分区内，也具有明显的好处。比如，图 11-3 中的表 1 和表 2 中 ID 为 27 的的多个行构成一个行组，它们之间存在主外键关联，把它们存放在一起可以加快查询的效率。

11.8 数据访问方法

这里以一个实例来阐释云数据库的数据访问方法。如图 11-4 所示，当客户端请求数据时，它首先向管理器请求一份分区映射图，管理器向客户端发送分区映射图，客户端收到以后，在图中进行搜寻，根据键值找到自己所需数据的存储位置，然后客户端到指定的数据节点请求数据，最后，由该数据节点把数据返回给客户端。实际上，为了改进性能，同时也为了避免管理器的性能瓶颈，通常会在客户端缓存常用的分区映射图，这样，客户端在很多情况下不用与管理器交互就可以直接访问相应的数据节点。

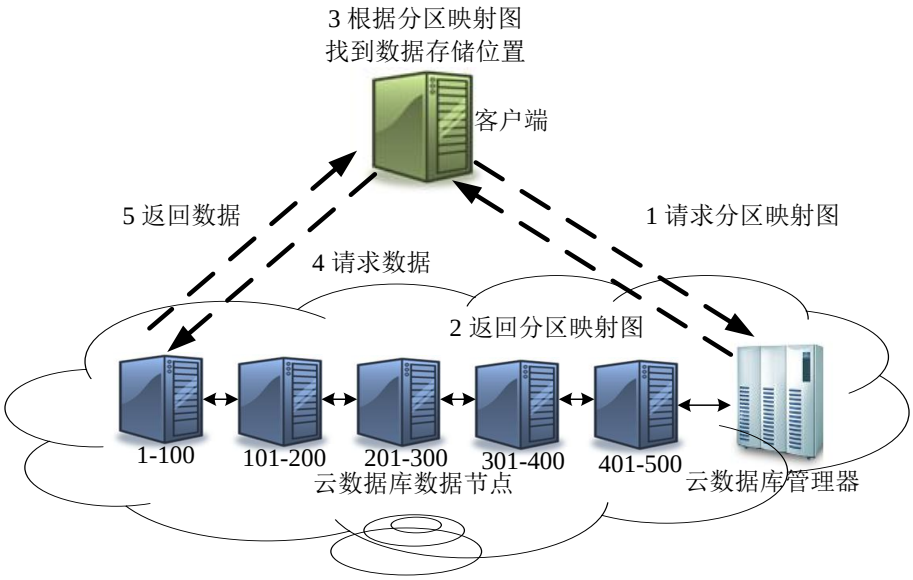

图 11-4 云数据库中的数据访问方法

11.9 编程模型

云数据库存储了海量数据，涉及到大量的数据运算，如果仍然采用传统的数据处理方法，将无法充分发挥云环境的优点。因此，采用一种机制简单而又具有高可扩展性的编程方法就显得尤为重要。查询这种大型的数据集，可以采用 Google 发明的一种新的编程模型，称为 MapReduce。

MapReduce 编程模型用来处理跨越成百上千个机器的大规模数据集，该软件架构隐藏了并行计算、数据分布和失败处理的细节。它背后的思想是，在分布式环境下处理数据，总是包括两个基本步骤：(1)映射所需的数据；（2）对这些数据进行聚集（aggregate）操作。它让用户定义一个映射函数，这个映射函数把一个“key/value”对的集合转换成一个中间临时

的“key/value”对的集合，然后使用一个 reduce 函数，把所有那些与同一个键相关的中间临时值都进行合并。许多现实世界的任务都可以采用这个模型来表达。MapReduce 可以运行在大规模商用服务器簇上，并且具有很高的可扩展性。一个典型的 MapReduce 计算会在成千上万的机器上处理许多 TB 的数据。

但是，传统的关系型数据库和 MapReduce 对数据的处理方式存在很大的区别，因此，需要把关系数据库中的一些操作转换成 MapReduce 操作。这其中一类典型的操作就是联接(join)操作。

总的来说，在 MapReduce 环境下执行两个关系的联接操作的方法如下：假设关系 R(A，B)和 S(B,C)都存储在一个文件中。为了联接这些关系，必须把来自每个关系的各个元组都和一个 key 关联，这个 key 就是属性 B 的值。可以使用一个 Map 进程集合，把来自 R 的每个元组(a,b)转换成一个 key-value 对，其中的 key 就是 b，值就是(a,R)。注意，这里把关系 R 包含到 value 中，这样做使得我们可以在 Reduce 阶段，只把那些来自 R 的元组和来自 S 的元组进行匹配。类似地，可以使用一个 Map 进程集合，把来自 S 的每个元组(b,c)，转换成一个 key-value 对，key 是 b，value 是(c,S)。这里把关系名字包含在属性值中，可以使得在 Reduce 阶段只把那些来自不同关系的元组进行合并。Reduce 进程的任务就是，把来自关系 R 和 S 的具有共同属性 B 值的元组进行合并。这样，所有具有特定 B 值的元组必须被发送到同一个 Reduce 进程。假设使用 k 个 Reduce 进程。这里选择一个哈希函数 h，它可以把属性 B 的值映射到 k 个哈希桶，每个哈希值对应一个 Reduce 进程。每个 Map 进程把 key 是 b 的 key-value 对，都发送到与哈希值 h(b)对应的 Reduce 进程。Reduce 进程把联接后的元组(a,b,c)，写到一个单独的输出文件中。

本章小结

本章首先介绍了云数据库的概念和特性，然后指出云数据库是海量存储需求的必然选择；接下来，比较了云数据库和传统的分布式数据库，并阐述了云数据库的影响；然后，介绍了具有代表性的云数据库产品，包括 Amazon、Google 和 Microsoft 的产品；最后，介绍了云数据库的数据模型和数据访问方法。

参考文献

[1] 林子雨,赖永炫,林琛,谢怡,邹权. 云数据库研究. 软件学报.2012,23(5):1148-1166.

（注：本章内容全部来自林子雨发表的《软件学报》论文“云数据库研究”）

附录 1:任课教师介绍

林子雨(1978－),男,博士,厦门大学计算机科学系助理教授,主要研究领域为数据库,数据仓库,数据挖掘.

主讲课程：《大数据技术基础》

办公地点：厦门大学海韵园科研 2 号楼

E-mail: ziyulin@xmu.edu.cn

个人网页：http://www.cs.xmu.edu.cn/linziyu