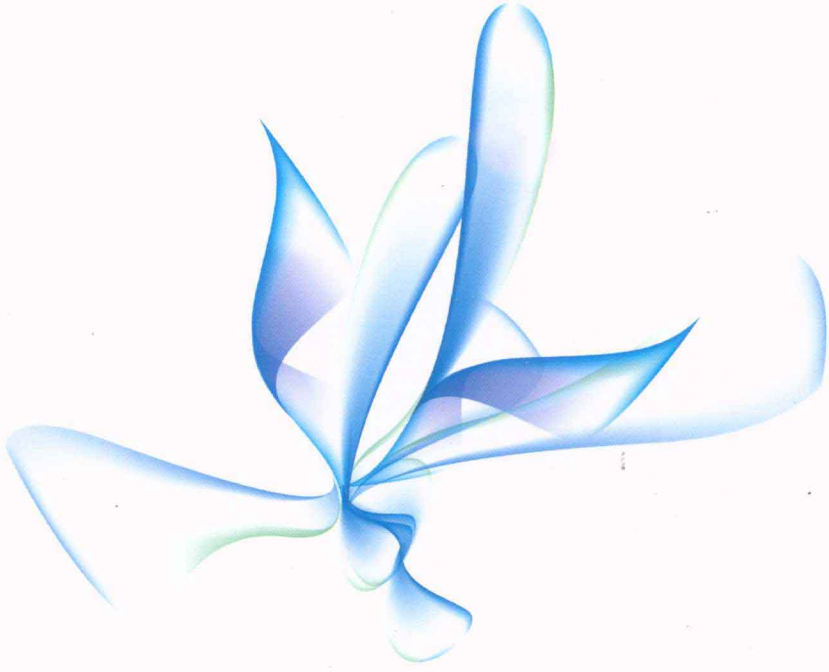
HZ BOOKS
华章科技

阿里巴巴高级技术专家（OceanBase核心开发人员）撰写，阳振坤、章文嵩、杨卫华、汪源、余锋（褚霸）、赖春波等来自阿里巴巴、新浪、网易和百度的资深技术专家联袂推荐。

系统讲解构建大规模存储系统的核心技术和原理，详细分析Google、Amazon、Microsoft和阿里巴巴的大规模分布式存储系统的原理。

实战性强，通过对阿里巴巴的分布式数据库OceanBase的实现细节进行深入分析，全面讲解了大规模分布式存储系统的架构方法与应用实践。

大数据技术丛书



Large-Scale Distributed Storage System: Principles and Architectures

大规模分布式存储系统

原理解析与架构实战

杨传辉◎著

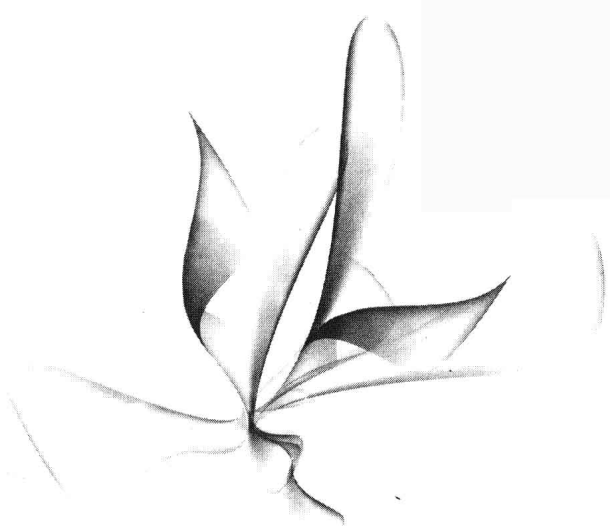
机械工业出版社
China Machine Press

大数据技术丛书

大规模分布式存储系统

原理解析与架构实战

杨传辉◎著



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

大规模分布式存储系统：原理解析与架构实战 / 杨传辉著 . —北京：机械工业出版社，2013.7

ISBN 978-7-111-43052-0

Ⅰ. 大… Ⅱ. 杨… Ⅲ. 大规模–分布式存储器–研究 Ⅳ. TP333.2

中国版本图书馆 CIP 数据核字（2013）第 141737 号

本书是分布式系统领域的经典著作，由阿里巴巴高级技术专家“阿里日照”（OceanBase 核心开发人员）撰写，阳振坤、章文嵩、杨卫华、汪源、余锋（褚霸）、赖春波等来自阿里、新浪、网易和百度的资深技术专家联袂推荐。理论方面，不仅讲解了大规模分布式存储系统的核心技术和基本原理，而且对谷歌、亚马逊、微软和阿里巴巴等国际型大互联网公司的大规模分布式存储系统进行了分析；实战方面，首先通过对阿里巴巴的分布式数据库 OceanBase 的实现细节的深入剖析完整地展示了大规模分布式存储系统的架构与设计过程，然后讲解了大规模分布式存储技术在云计算和大数据领域的实践与应用。

本书内容分为四个部分：基础篇——分布式存储系统的基础知识，包含单机存储系统的知识，如数据模型、事务与并发控制、故障恢复、存储引擎、压缩 / 解压缩等；分布式系统的数据分布、复制、一致性、容错、可扩展性等。范型篇——介绍谷歌、亚马逊、微软、阿里巴巴等著名互联网公司的大规模分布式存储系统架构，涉及分布式文件系统、分布式键值系统、分布式表格系统以及分布式数据库技术等。实践篇——以阿里巴巴的分布式数据库 OceanBase 为例，详细介绍分布式数据库内部实现，以及实践过程中的经验。专题篇——介绍分布式系统的主要应用：云存储和大数据，这些是近年来的热门领域，本书介绍了云存储平台、技术与安全，以及大数据的概念、流式计算、实时分析等。

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：吴 怡

北京市荣盛彩色印刷有限公司印刷

2013 年 9 月第 1 版第 1 次印刷

186mm × 240mm • 19 印张

标准书号：ISBN 978-7-111-43052-0

定 价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991 88361066 投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

序 言

关于分布式系统的知识，可以从大学教科书上找到，许多人还知道 Andrew S. Tanenbaum 等人在 2002 年出版的“分布式系统原理与范型”(Distributed Systems: Principles and Paradigms) 这本书。其实分布式系统的理论出现于上个世纪 70 年代，“Symposium on Principles of Distributed Computing (PODC)”和“International Symposium on Distributed Computing (DISC)”这两个分布式领域的学术会议分别创立于 1982 年和 1985 年。然而，分布式系统的广泛应用却是最近十多年的事情，其中的一个原因就是人类活动创造出的数据量远远超出了单个计算机的存储和处理能力。比如，2008 年全球互联网的网页超过了 1 万亿，按平均单个网页 10KB 计算，就是 10PB；又如，一个 2 亿用户的电信运营商，如果平均每个用户每天拨打接听总共 10 个电话，每个电话 400 字节，5 年的话费记录总量即为 0.2G×10×0.4K×365×5=1.46PB。除了分布式系统，人们还很难有其他高效的手段来存储和处理这些 PB 级甚至更多的数据。另外一个原因，其实是一个可悲的事实，那就是分布式环境下的编程十分困难。

与单机环境下的编程相比，分布式环境下的编程有两个明显的不同：首先，分布式环境下会出现一部分计算机工作正常，另一部分计算机工作不正常的情况，程序需要在这种情况下尽可能地正常工作，这个挑战非常大。其次，单机环境下的函数调用常常可以在微秒级内返回，所以除了少数访问外部设备（例如磁盘、网卡等）的函数采用异步方式调用外，大部分函数采用同步调用的方式，编译器和操作系统在调用前后自动保存与恢复程序的上下文；在分布式环境下，计算机之间的函数调用（远程调用，即 RPC）的返回时间通常是毫秒或亚毫秒（0.1~1.0 毫秒）级，差不多是单机环境的 100 倍，使用同步方式远远不能发挥现代 CPU 处理器的性能，所以分布式环境下的 RPC 通常采用异步调用方式，程序需要自己保存和恢复调用前后的上下文，并需要处理更多的异常。

基于上述原因，很多从事分布式系统相关的开发、测试、维护的朋友十分渴望了解和学习其他分布式系统的实践，可是，这些信息分散在浩瀚的知识海洋中，获取感兴趣的内容相当困难。因此，传辉的“大规模分布式存储系统”一书出现得恰如其时，这是我见过的讲解分布式系统实践最全面的一本书。它不仅介绍了当前业界最常见的分布式系统，还结合了作者自己六年多来的分布式系统开发实践。虽然这本书没有、也不可能包含分布式系统实践的所有内容，但阅读这本书的人一定会深受启发，并且还能够知道去何处获取更深层次的信息和知识。

阳振坤

阿里巴巴高级研究员，基础数据部负责人

前　言

随着社交网络、移动互联网、电子商务等技术的不断发展，互联网的使用者贡献了越来越多的内容。为了处理这些内容，每个互联网公司在后端都有一套成熟的分布式系统用于数据的存储、计算以及价值提取。Google 是全球最大的互联网公司，也是在分布式技术上相对成熟的公司，其公布的 Google 分布式文件系统 GFS、分布式计算系统 MapReduce、分布式表格系统 Bigtable 都成为业界竞相模仿的对象，最近公布的全球数据库 Spanner 更是能够支持分布在世界各地上百个数据中心的上百万台服务器。Google 的核心技术正是后端这些处理海量数据的分布式系统。和 Google 类似，国外的亚马逊、微软以及国内互联网三巨头阿里巴巴、百度和腾讯的核心技术也是其后端的海量数据处理系统。

本书的内容是介绍互联网公司的大规模分布式存储系统。与传统的高端服务器、高端存储器和高端处理器不同的是，互联网公司的分布式存储系统由数量众多的、低成本和高性价比的普通 PC 服务器通过网络连接而成。互联网的业务发展很快，而且注重成本，这就使得存储系统不能依靠传统的纵向扩展的方式，即先买小型机，不够时再买中型机，甚至大型机。互联网后端的分布式系统要求支持横向扩展，即通过增加普通 PC 服务器来提高系统的整体处理能力。普通 PC 服务器性价比高，故障率也高，需要在软件层面实现自动容错，保证数据的一致性。另外，随着服务器的不断加入，需要能够在软件层面实现自动负载均衡，使得系统的处理能力得到线性扩展。

分布式存储和当今同样备受关注的云存储和大数据又是什么关系呢？分布式存储是基础，云存储和大数据是构建在分布式存储之上的应用。移动终端的计算能力和存储空间有限，而且有在多个设备之间共享资源的强烈的需求，这就使得网盘、相册等云存储应用很快流行起来。然而，万变不离其宗，云存储的核心还是后端的大规模分布式存储系统。大数据则更近一步，不仅需要存储海量数据，还需要通过合适的计算框架或者工具对这些数据进行分析，抽取其中有价值的部分。如果没有分布式存储，便谈不上对大数据进行分析。仔细分析还会发现，分布式存储技术是互联网后端架构的“九阳神功”，掌握了这项技能，以后理解其他技术的本质会变得非常容易。

分布式存储技术如此重要，市面上也有很多分布式系统相关的书籍。然而，这些书籍往往注重理论不重实践，且所述理论也不太适合互联网公司的大规模存储系统。这是因为，虽然分布式系统研究了很多年，但是大规模分布式存储系统是在近几年才流行起来，而且起源于以 Google 为首的企业界而非学术界。笔者 2007 年年底加入百度公司，师从阳振坤老师，从事大规模分布式存储的研究和实践工作，曾经开发过类似 GFS、MapReduce 和 Bigtable 的分布式系统，后来转战阿里巴巴继续开发分布式数据库 OceanBase，维护分布式技术博客 NosqlNotes（http://www.nosqlnotes.net）。笔者在业余时间阅读并理解了绝大部

分分布式系统原理和各大互联网公司的系统范型相关论文，深知分布式存储系统的复杂性，也能够体会到广大读者渴望弄清楚分布式存储技术本质和实现细节的迫切心情，因而集中精力编写了这本书，希望对从事分布式存储应用的技术人员有所裨益。

本书的目标是介绍互联网公司的大规模分布式存储系统，共分为四篇：

- **基础篇**。基础知识包含两个部分：单机存储系统以及分布式系统。其中，单机存储系统的理论基础是数据库技术，包括数据模型、事务与并发控制、故障恢复、存储引擎、数据压缩等；分布式系统涉及数据分布、复制、一致性、容错、可扩展性等分布式技术。另外，分布式存储系统工程师还需要一项基础训练，即性能预估，因此，基础篇也会顺带介绍硬件基础知识以及性能预估方法。
- **范型篇**。这部分内容将介绍 Google、亚马逊、微软、阿里巴巴等各大互联网公司的大规模分布式存储系统，分为四章：分布式文件系统、分布式键值系统、分布式表格系统以及分布式数据库。
- **实践篇**。这部分内容将以笔者在阿里巴巴开发的分布式数据库 OceanBase 为例详细介绍分布式数据库内部实现以及实践过程中的经验总结。
- **专题篇**。云存储和大数据是近年来兴起的两大热门领域，其底层都依赖分布式存储技术，这部分将简单介绍这两方面的基础知识。

本书适合互联网行业或者其他从事分布式系统实践的工程人员，也适合大学高年级本科生和研究生作为分布式系统或者云计算相关课程的参考书籍。阅读本书之前，建议首先理解分布式系统和数据库相关基础理论，接着阅读第一篇。如果对各个互联网公司的系统架构感兴趣，可以选择阅读第二篇的某些章节；如果对阿里巴巴 OceanBase 的架构设计和实现感兴趣，可以顺序阅读第三篇。最后，如果对云存储或者大数据感兴趣，可以选择阅读第四篇的某个章节。

感谢阳振坤老师多年以来对我在云计算和分布式数据库这两个领域的研究实践工作的指导和鼓励。感谢在百度以及阿里巴巴与我共事多年的兄弟姐妹，我们患难与共，一起实现共同的梦想。感谢机械工业出版社的吴怡编辑、新浪微博的杨卫华先生、百度的侯震宇先生以及支付宝的童家旺先生在本书撰写过程中提出的宝贵意见。

由于分布式存储技术涉及一些公司的商业机密，加上笔者水平有限、时间较紧，所以书中难免存在谬误，很多技术点涉及的细节描述得还不够详尽，恳请读者批评指正。可将任何意见和建议发送到我的邮箱 knuthocean@163.com，本书相关的勘误和技术细节说明也会发布到我的个人博客 NosqlNotes。我的新浪微博账号是“阿里日照”，欢迎读者通过邮件、博客或者微博与我交流分布式存储相关的任何问题。我也将密切跟踪分布式存储技术的发展，吸收您的意见，适时编写本书的升级版本。

杨传辉

2013 年 7 月于北京

目　录

第1章 概 述

Google、Amazon、Alibaba等互联网公司的成功催生了云计算和大数据两大热门领域。无论是云计算、大数据还是互联网公司的各种应用，其后台基础设施的主要目标都是构建低成本、高性能、可扩展、易用的分布式存储系统。

虽然分布式系统研究了很多年，但是，直到近年来，互联网大数据应用的兴起才使得它大规模地应用到工程实践中。相比传统的分布式系统，互联网公司的分布式系统具有两个特点：一个特点是规模大，另一个特点是成本低。不同的需求造就了不同的设计方案，可以这么说，Google等互联网公司重新定义了大规模分布式系统。本章介绍大规模分布式系统的定义与分类。

1.1 分布式存储概念

大规模分布式存储系统的定义如下：

“分布式存储系统是大量普通PC服务器通过Internet互联，对外作为一个整体提供存储服务。”

分布式存储系统具有如下几个特性：

- 可扩展。分布式存储系统可以扩展到几百台甚至几千台的集群规模，而且，随着集群规模的增长，系统整体性能表现为线性增长。
- 低成本。分布式存储系统的自动容错、自动负载均衡机制使其可以构建在普通PC机之上。另外，线性扩展能力也使得增加、减少机器非常方便，可以实现自动运维。
- 高性能。无论是针对整个集群还是单台服务器，都要求分布式存储系统具备高性能。
- 易用。分布式存储系统需要能够提供易用的对外接口，另外，也要求具备完善的监控、运维工具，并能够方便地与其他系统集成，例如，从Hadoop云计算系统导入数据。

分布式存储系统的挑战主要在于数据、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。分布式存储涉及的技术主要来自两个领域：分布式系统以及数据库，如下所示：

- 数据分布：如何将数据分布到多台服务器才能够保证数据分布均匀？数据分布到多台服务器后如何实现跨服务器读写操作？
- 一致性：如何将数据的多个副本复制到多台服务器，即使在异常情况下，也能够保证不同副本之间的数据一致性？
- 容错：如何检测到服务器故障？如何自动将出现故障的服务器上的数据和服务迁移到集群中其他服务器？
- 负载均衡：新增服务器和集群正常运行过程中如何实现自动负载均衡？数据迁移的过程中如何保证不影响已有服务？
- 事务与并发控制：如何实现分布式事务？如何实现多版本并发控制？
- 易用性：如何设计对外接口使得系统容易使用？如何设计监控系统并将系统的内部状态以方便的形式暴露给运维人员？
- 压缩 / 解压缩：如何根据数据的特点设计合理的压缩 / 解压缩算法？如何平衡压缩算法节省的存储空间和消耗的 CPU 计算资源？

分布式存储系统挑战大，研发周期长，涉及的知识面广。一般来讲，工程师如果能够深入理解分布式存储系统，理解其他互联网后台架构不会再有任何困难。

1.2 分布式存储分类

分布式存储面临的数据需求比较复杂，大致可以分为三类：

- 非结构化数据：包括所有格式的办公文档、文本、图片、图像、音频和视频信息等。
- 结构化数据：一般存储在关系数据库中，可以用二维关系表结构来表示。结构化数据的模式（Schema，包括属性、数据类型以及数据之间的联系）和内容是分开的，数据的模式需要预先定义。
- 半结构化数据：介于非结构化数据和结构化数据之间，HTML 文档就属于半结构化数据。它一般是自描述的，与结构化数据最大的区别在于，半结构化数据的模式结构和内容混在一起，没有明显的区分，也不需要预先定义数据的模式结构。

不同的分布式存储系统适合处理不同类型的数据，本书将分布式存储系统分为四类：分布式文件系统、分布式键值（Key-Value）系统、分布式表格系统和分布式数据库。

1. 分布式文件系统

互联网应用需要存储大量的图片、照片、视频等非结构化数据对象，这类数据以对象的形式组织，对象之间没有关联，这样的数据一般称为 Blob（Binary Large Object，二进制大对象）数据。

分布式文件系统用于存储 Blob 对象，典型的系统有 Facebook Haystack 以及 Taobao File System（TFS）。另外，分布式文件系统也常作为分布式表格系统以及分布式数据库的底层存储，如谷歌的 GFS（Google File System，存储大文件）可以作为分布式表格系统 Google Bigtable 的底层存储，Amazon 的 EBS（Elastic Block Store，弹性块存储）系统可以作为分布式数据库（Amazon RDS）的底层存储。

总体上看，分布式文件系统存储三种类型的数据：Blob 对象、定长块以及大文件。在系统实现层面，分布式文件系统内部按照数据块（chunk）来组织数据，每个数据块的大小大致相同，每个数据块可以包含多个 Blob 对象或者定长块，一个大文件也可以拆分为多个数据块，如图 1-1 所示。分布式文件系统将这些数据块分散到存储集群，处理数据复制、一致性、负载均衡、容错等分布式系统难题，并将用户对 Blob 对象、定长块以及大文件的操作映射为对底层数据块的操作。

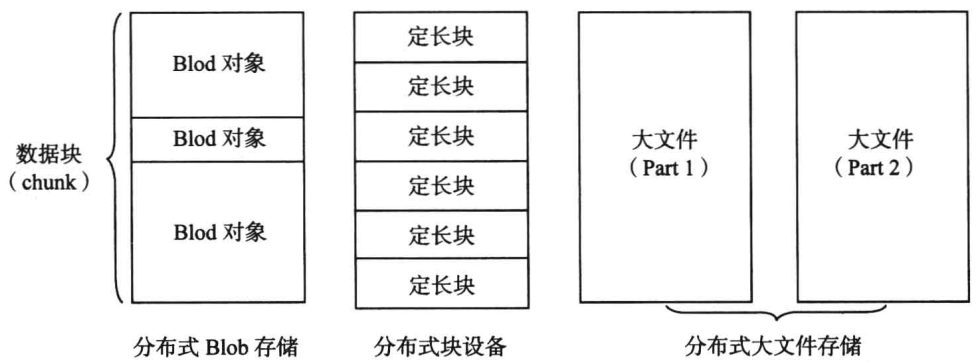


图 1-1 数据块与 Blob 对象、定长块、大文件之间的关系

2. 分布式键值系统

分布式键值系统用于存储关系简单的半结构化数据，它只提供基于主键的 CRUD（Create/Read/Update/Delete）功能，即根据主键创建、读取、更新或者删除一条键值记录。

典型的系统有 Amazon Dynamo 以及 Taobao Tair。从数据结构的角度看，分布式键值系统与传统的哈希表比较类似，不同的是，分布式键值系统支持将数据分布到集群中的多个存储节点。分布式键值系统是分布式表格系统的一种简化实现，一般用作缓存，比如淘宝 Tair 以及 Memcache。一致性哈希是分布式键值系统中常用的数据分布技术，因其被 Amazon DynamoDB 系统使用而变得相当有名。

3. 分布式表格系统

分布式表格系统用于存储关系较为复杂的半结构化数据，与分布式键值系统相比，分布式表格系统不仅仅支持简单的 CRUD 操作，而且支持扫描某个主键范围。分布式

表格系统以表格为单位组织数据，每个表格包括很多行，通过主键标识一行，支持根据主键的 CRUD 功能以及范围查找功能。

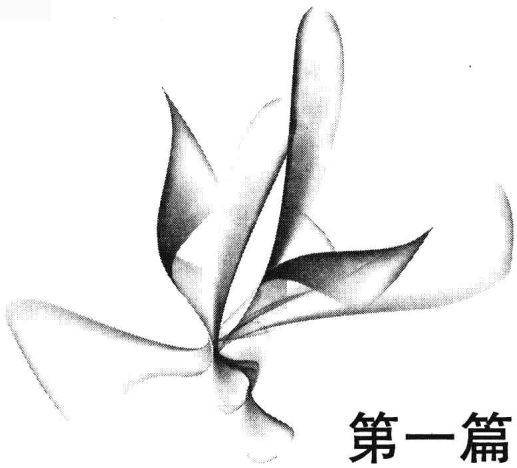
分布式表格系统借鉴了很多关系数据库的技术，例如支持某种程度上的事务，比如单行事务，某个实体组（Entity Group，一个用户下的所有数据往往构成一个实体组）下的多行事务。典型的系统包括 Google Bigtable 以及 Megastore，Microsoft Azure Table Storage，Amazon DynamoDB 等。与分布式数据库相比，分布式表格系统主要支持针对单张表格的操作，不支持一些特别复杂的操作，比如多表关联，多表联接，嵌套子查询；另外，在分布式表格系统中，同一个表格的多个数据行也不要求包含相同类型的列，适合半结构化数据。分布式表格系统是一种很好的权衡，这类系统可以做到超大规模，而且支持较多的功能，但实现往往比较复杂，而且有一定的使用门槛。

4. 分布式数据库

分布式数据库一般是从单机关系数据库扩展而来，用于存储结构化数据。分布式数据库采用二维表格组织数据，提供 SQL 关系查询语言，支持多表关联，嵌套子查询等复杂操作，并提供数据库事务以及并发控制。

典型的系统包括 MySQL 数据库分片（MySQL Sharding）集群，Amazon RDS 以及 Microsoft SQL Azure。分布式数据库支持的功能最为丰富，符合用户使用习惯，但可扩展性往往受到限制。当然，这一点并不是绝对的。Google Spanner 系统是一个支持多数据中心的分布式数据库，它不仅支持丰富的关系数据库功能，还能扩展到多个数据中心的成千上万台机器。除此之外，阿里巴巴 OceanBase 系统也是一个支持自动扩展的分布式关系数据库。

关系数据库是目前为止最为成熟的存储技术，它的功能极其丰富，产生了商业的关系数据库软件（例如 Oracle，Microsoft SQL Server，IBM DB2，MySQL）以及上层的工具及应用软件生态链。然而，关系数据库在可扩展性上面临着巨大的挑战。传统关系数据库的事务以及二维关系模型很难高效地扩展到多个存储节点上，另外，关系数据库对于要求高并发的应用在性能上优化空间较大。为了解决关系数据库面临的可扩展性、高并发以及性能方面的问题，各种各样的非关系数据库风起云涌，这类系统成为 NoSQL 系统，可以理解为“Not Only SQL”系统。NoSQL 系统多得让人眼花缭乱，每个系统都有自己的独到之处，适合解决某种特定的问题。这些系统变化很快，本书不会尝试去探寻某种 NoSQL 系统的实现，而是从分布式存储技术的角度探寻大规模存储系统背后的原理。



第一篇
基 础 篇

本篇内容

第 2 章　单机存储系统

第 3 章　分布式系统

第 2 章　单机存储系统

单机存储引擎就是哈希表、B 树等数据结构在机械磁盘、SSD 等持久化介质上的实现。单机存储系统是单机存储引擎的一种封装，对外提供文件、键值、表格或者关系模型。单机存储系统的理论来源于关系数据库。数据库将一个或多个操作组成一组，称作事务，事务必须满足原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）以及持久性（Durability），简称为 ACID 特性。多个事务并发执行时，数据库的并发控制管理器必须能够保证多个事务的执行结果不能破坏某种约定，如不能出现事务执行到一半的情况，不能读取到未提交的事务，等等。为了保证持久性，对于数据库的每一个变化都要在磁盘上记录日志，当数据库系统突然发生故障，重启后能够恢复到之前一致的状态。

本章首先介绍 CPU、IO、网络等硬件基础知识及性能参数，接着介绍主流的单机存储引擎。其中，哈希存储引擎是哈希表的持久化实现，B 树存储引擎是 B 树的持久化实现，而 LSM 树（Log Structure Merge Tree）存储引擎采用批量转储技术来避免磁盘随机写入。最后，介绍关系数据库理论基础，包括事务、并发控制、故障恢复、数据压缩等。

2.1　硬件基础

硬件发展很快，摩尔定律告诉我们：每 18 个月计算机等 IT 产品的性能会翻一番；或者说相同性能的计算机等 IT 产品，每 18 个月价钱会降一半。但是，计算机的硬件体系架构保持相对稳定。架构设计很重要的一点就是合理选择并且能够最大限度地发挥底层硬件的价值。

2.1.1　CPU 架构

早期的 CPU 为单核芯片，工程师们很快意识到，仅仅提高单核的速度会产生过多的热量且无法带来相应的性能改善。因此，现代服务器基本为多核或多个 CPU。经典的多 CPU 架构为对称多处理结构（Symmetric Multi-Processing，SMP），即在一个计算机上汇集了一组处理器，它们之间对称工作，无主次或从属关系，共享相同的物理内存及总线，如图 2-1 所示。

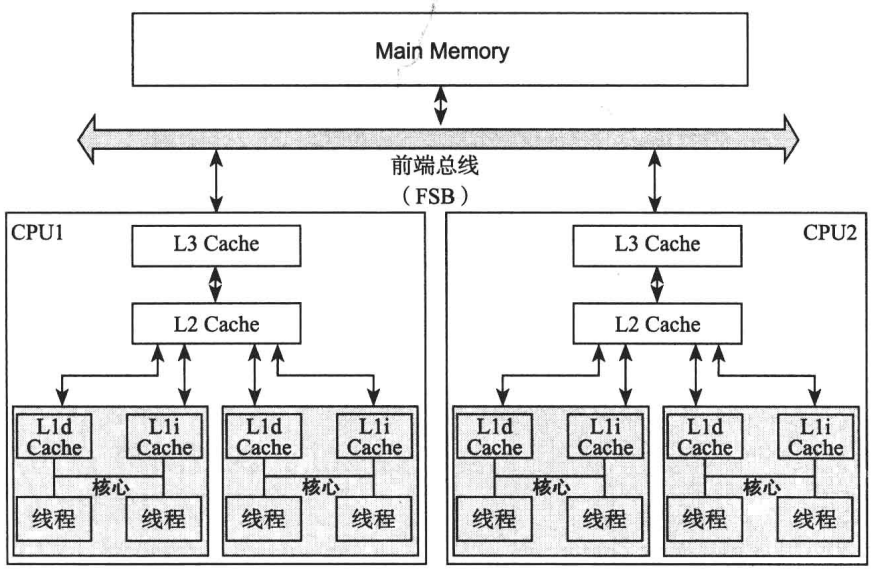


图 2-1　SMP 系统结构

图 2-1 中的 SMP 系统由两个 CPU 组成，每个 CPU 有两个核心（core），CPU 与内存之间通过总线通信。每个核心有各自的 L1d Cache（L1 数据缓存）及 L1i Cache（L1 指令缓存），同一个 CPU 的多个核心共享 L2 以及 L3 缓存，另外，某些 CPU 还可以通过超线程技术（Hyper-Threading Technology）使得一个核心具有同时执行两个线程的能力。

SMP 架构的主要特征是共享，系统中所有资源（CPU、内存、I/O 等）都是共享的，由于多 CPU 对前端总线的竞争，SMP 的扩展能力非常有限。为了提高可扩展性，现在的主流服务器架构一般为 NUMA（Non-Uniform Memory Access，非一致存储访问）架构。它具有多个 NUMA 节点，每个 NUMA 节点是一个 SMP 结构，一般由多个 CPU（如 4 个）组成，并且具有独立的本地内存、IO 槽口等。

图 2-2 为包含 4 个 NUMA 节点的服务器架构图，NUMA 节点可以直接快速访问本地内存，也可以通过 NUMA 互联互通模块访问其他 NUMA 节点的内存，访问本地内存的速度远远高于远程访问的速度。由于这个特点，为了更好地发挥系统性能，开发应用程序时需要尽量减少不同 NUMA 节点之间的信息交互。

2.1.2　IO 总线

存储系统的性能瓶颈一般在于 IO，因此，有必要对 IO 子系统的架构有一个大致的了解。以 Intel x48 主板为例，它是典型的南、北桥架构，如图 2-3 所示。北桥芯片通过前端总线（Front Side Bus，FSB）与 CPU 相连，内存模块以及 PCI-E 设备（如高端的 SSD 设备 Fusion-IO）挂接在北桥上。北桥与南桥之间通过 DMI 连接，DMI 的带宽