

大规模分布式存储系统：原理解析与架构实战 (大数据技术丛书)

作者：杨传辉

大规模分布式存储系统

——原理解析与架构实战

杨传辉 著

ISBN: 978-7-111-43052-0

本书纸版由机械工业出版社于2013年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

目 录

[序言](#)

[前言](#)

[第1章 概述](#)

[1.1 分布式存储概念](#)

[1.2 分布式存储分类](#)

[第一篇 基础篇](#)

[第2章 单机存储系统](#)

[2.1 硬件基础](#)

[2.1.1 CPU架构](#)

[2.1.2 IO总线](#)

[2.1.3 网络拓扑](#)

[2.1.4 性能参数](#)

[2.1.5 存储层次架构](#)

[2.2 单机存储引擎](#)

[2.2.1 哈希存储引擎](#)

[2.2.2 B树存储引擎](#)

[2.2.3 LSM树存储引擎](#)

[2.3 数据模型](#)

[2.3.1 文件模型](#)

[2.3.2 关系模型](#)

[2.3.3 键值模型](#)

[2.3.4 SQL与NoSQL](#)

[2.4 事务与并发控制](#)

[2.4.1 事务](#)

[2.4.2 并发控制](#)

[2.5 故障恢复](#)

[2.5.1 操作日志](#)

[2.5.2 重做日志](#)

[2.5.3 优化手段](#)

[2.6 数据压缩](#)

[2.6.1 压缩算法](#)

[2.6.2 列式存储](#)

[第3章 分布式系统](#)

[3.1 基本概念](#)

[3.1.1 异常](#)

[3.1.2 一致性](#)

[3.1.3 衡量指标](#)

[3.2 性能分析](#)

[3.3 数据分布](#)

[3.3.1 哈希分布](#)

[3.3.2 顺序分布](#)

[3.3.3 负载均衡](#)

[3.4 复制](#)

[3.4.1 复制的概述](#)

[3.4.2 一致性与可用性](#)

[3.5 容错](#)

[3.5.1 常见故障](#)

[3.5.2 故障检测](#)

[3.5.3 故障恢复](#)

[3.6 可扩展性](#)

[3.6.1 总控节点](#)

[3.6.2 数据库扩容](#)

[3.6.3 异构系统](#)

[3.7 分布式协议](#)

[3.7.1 两阶段提交协议](#)

[3.7.2 Paxos协议](#)

[3.7.3 Paxos与2PC](#)

[3.8 跨机房部署](#)

[第二篇 范型篇](#)

[第4章 分布式文件系统](#)

[4.1 Google文件系统](#)

[4.1.1 系统架构](#)

[4.1.2 关键问题](#)

[4.1.3 Master设计](#)

[4.1.4 ChunkServer设计](#)

[4.1.5 讨论](#)

[4.2 Taobao File System](#)

[4.2.1 系统架构](#)

[4.2.2 讨论](#)

[4.3 Facebook Haystack](#)

[4.3.1 系统架构](#)

[4.3.2 讨论](#)

[4.4 内容分发网络](#)

[4.4.1 CDN架构](#)

[4.4.2 讨论](#)

[第5章 分布式键值系统](#)

[5.1 Amazon Dynamo](#)

[5.1.1 数据分布](#)

[5.1.2 一致性与复制](#)

[5.1.3 容错](#)

[5.1.4 负载均衡](#)

[5.1.5 读写流程](#)

[5.1.6 单机实现](#)

[5.1.7 讨论](#)

[5.2 淘宝Tair](#)

[5.2.1 系统架构](#)

[5.2.2 关键问题](#)

[5.2.3 讨论](#)

[第6章 分布式表格系统](#)

[6.1 Google Bigtable](#)

[6.1.1 架构](#)

[6.1.2 数据分布](#)

[6.1.3 复制与一致性](#)

[6.1.4 容错](#)

[6.1.5 负载均衡](#)

[6.1.6 分裂与合并](#)

[6.1.7 单机存储](#)

[6.1.8 垃圾回收](#)

[6.1.9 讨论](#)

[6.2 Google Megastore](#)

[6.2.1 系统架构](#)

[6.2.2 实体组](#)

[6.2.3 并发控制](#)

[6.2.4 复制](#)

[6.2.5 索引](#)

[6.2.6 协调者](#)

[6.2.7 读取流程](#)

[6.2.8 写入流程](#)

[6.2.9 讨论](#)

[6.3 Windows Azure Storage](#)

[6.3.1 整体架构](#)

[6.3.2 文件流层](#)

[6.3.3 分区层](#)

[6.3.4 讨论](#)

[第7章 分布式数据库](#)

[7.1 数据库中间层](#)

[7.1.1 架构](#)

[7.1.2 扩容](#)

[7.1.3 讨论](#)

- [7.2 Microsoft SQL Azure](#)
- [7.2.1 数据模型](#)
- [7.2.2 架构](#)
- [7.2.3 复制与一致性](#)
- [7.2.4 容错](#)
- [7.2.5 负载均衡](#)
- [7.2.6 多租户](#)
- [7.2.7 讨论](#)
- [7.3 Google Spanner](#)
- [7.3.1 数据模型](#)
- [7.3.2 架构](#)
- [7.3.3 复制与一致性](#)
- [7.3.4 TrueTime](#)
- [7.3.5 并发控制](#)
- [7.3.6 数据迁移](#)
- [7.3.7 讨论](#)
- [第三篇 实践篇](#)
- [第8章 OceanBase架构初探](#)
- [8.1 背景简介](#)
- [8.2 设计思路](#)
- [8.3 系统架构](#)
- [8.3.1 整体架构图](#)
- [8.3.2 客户端](#)
- [8.3.3 RootServer](#)
- [8.3.4 MergeServer](#)
- [8.3.5 ChunkServer](#)
- [8.3.6 UpdateServer](#)
- [8.3.7 定期合并&数据分发](#)
- [8.4 架构剖析](#)
- [8.4.1 一致性选择](#)
- [8.4.2 数据结构](#)
- [8.4.3 可靠性与可用性](#)
- [8.4.4 读写事务](#)
- [8.4.5 单点性能](#)
- [8.4.6 SSD支持](#)
- [8.4.7 数据正确性](#)
- [8.4.8 分层结构](#)
- [第9章 分布式存储引擎](#)
- [9.1 公共模块](#)
- [9.1.1 内存管理](#)
- [9.1.2 基础数据结构](#)
- [9.1.3 锁](#)
- [9.1.4 任务队列](#)
- [9.1.5 网络框架](#)
- [9.1.6 压缩与解压缩](#)
- [9.2 RootServer实现机制](#)
- [9.2.1 数据结构](#)
- [9.2.2 子表复制与负载均衡](#)
- [9.2.3 子表分裂与合并](#)
- [9.2.4 UpdateServer选主](#)
- [9.2.5 RootServer主备](#)
- [9.3 UpdateServer实现机制](#)
- [9.3.1 存储引擎](#)
- [9.3.2 任务模型](#)
- [9.3.3 主备同步](#)
- [9.4 ChunkServer实现机制](#)
- [9.4.1 子表管理](#)
- [9.4.2 SSTable](#)
- [9.4.3 缓存实现](#)
- [9.4.4 IO实现](#)

- [9.4.5 定期合并&数据分发](#)
- [9.4.6 定期合并限速](#)
- [9.5 消除更新瓶颈](#)
 - [9.5.1 读写优化回顾](#)
 - [9.5.2 数据旁路导入](#)
 - [9.5.3 数据分区](#)
- [第10章 数据库功能](#)
 - [10.1 整体结构](#)
 - [10.2 只读事务](#)
 - [10.2.1 物理操作符接口](#)
 - [10.2.2 单表操作](#)
 - [10.2.3 多表操作](#)
 - [10.2.4 SQL执行本地化](#)
 - [10.3 写事务](#)
 - [10.3.1 写事务执行流程](#)
 - [10.3.2 多版本并发控制](#)
 - [10.4 OLAP业务支持](#)
 - [10.4.1 并发查询](#)
 - [10.4.2 列式存储](#)
 - [10.5 特色功能](#)
 - [10.5.1 大表左连接](#)
 - [10.5.2 数据过期与批量删除](#)
- [第11章 质量保证、运维及实践](#)
 - [11.1 质量保证](#)
 - [11.1.1 RD开发](#)
 - [11.1.2 QA测试](#)
 - [11.1.3 试运行](#)
 - [11.2 使用与运维](#)
 - [11.2.1 使用](#)
 - [11.2.2 运维](#)
 - [11.3 应用](#)
 - [11.3.1 收藏夹](#)
 - [11.3.2 天猫评价](#)
 - [11.3.3 直通车报表](#)
 - [11.4 最佳实践](#)
 - [11.4.1 系统发展路径](#)
 - [11.4.2 人员成长](#)
 - [11.4.3 系统设计](#)
 - [11.4.4 系统实现](#)
 - [11.4.5 使用与运维](#)
 - [11.4.6 工程现象](#)
 - [11.4.7 经验法则](#)
- [第四篇 专题篇](#)
- [第12章 云存储](#)
 - [12.1 云存储的概念](#)
 - [12.2 云存储的产品形态](#)
 - [12.3 云存储技术](#)
 - [12.4 云存储的核心优势](#)
 - [12.5 云平台整体架构](#)
 - [12.5.1 Amazon云平台](#)
 - [12.5.2 Google云平台](#)
 - [12.5.3 Microsoft云平台](#)
 - [12.5.4 云平台架构](#)
 - [12.6 云存储技术体系](#)
 - [12.7 云存储安全](#)
- [第13章 大数据](#)
 - [13.1 大数据的概念](#)
 - [13.2 MapReduce](#)
 - [13.3 MapReduce扩展](#)
 - [13.3.1 Google Tenzing](#)

[13.3.2 Microsoft Dryad](#)

[13.3.3 Google Pregel](#)

[13.4 流式计算](#)

[13.4.1 原理](#)

[13.4.2 Yahoo S4](#)

[13.4.3 Twitter Storm](#)

[13.5 实时分析](#)

[13.5.1 MPP架构](#)

[13.5.2 EMC Greenplum](#)

[13.5.3 HP Vertica](#)

[13.5.4 Google Dremel](#)

[参考资料](#)

[硬件基础](#)

[存储系统](#)

[分布式系统](#)

[分布式文件系统](#)

[分布式键值系统](#)

[分布式表格系统](#)

[分布式数据库](#)

[OceanBase](#)

[云存储](#)

[离线分析](#)

[OLAP](#)

[流式计算](#)

[其他](#)

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 $0.2G \times 10 \times 0.4K \times 365 \times 5 = 1.46PB$ 。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

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。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《大规模分布式存储系统：原理解析与架构实战（大数据技术丛书）》杨传辉 著.epu

请登录 <https://shgis.cn/post/302.html> 下载完整文档。

手机端请扫码查看：

