



```
server.run();  
  
// startup  
server.startup();  
System.out.println("MyCAT Server startup");  
  
} catch (Exception e) {  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
    LOGGER.error(sdf.format(new Date()) + " " + e.getMessage());  
    System.exit(-1);  
}
```

# 分布式数据库架构 及 企业实践 基于Mycat中间件

/ 周继锋 冯钻优 陈胜尊 左越宗 著 /

# 分布式数据库架构 及 企业实践

基于Mycat中间件

/ 周继锋 冯钻优 陈胜尊 左越宗 著 /



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

## 内 容 简 介

本书由资深 Mycat 专家及一线架构师、DBA 编写而成。全书总计 8 章, 首先简单介绍了分布式系统和分布式数据库的需求, 然后讲解了分布式数据库的实现原理, 并对市场上存在的各种分布式数据库中间件进行了对比, 再围绕着如何利用 Mycat 实现分布式数据库而展开。本书对 Mycat 从入门到进阶、从高级技术实践到架构剖析、从网络通信协议解析到系统工作原理的方方面面进行了详细讲解, 并剖析了 Mycat 的 SQL 路由、跨库联合查询、分布式事务及原生 MySQL、PostgreSQL 协议等核心技术。通过本书不仅可以了解 Mycat 的基本概念, 掌握 Mycat 配置等技术, 还能感受到 Mycat 的架构设计之美, 了解 Mycat 2.0 的未来规划。

无论是对于软件工程师、测试工程师、运维工程师、软件架构师、技术经理, 还是对于资深 IT 人士来说, 本书都极具参考价值。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

## 图书在版编目 (CIP) 数据

分布式数据库架构及企业实践: 基于 Mycat 中间件 / 周继锋等著. —北京: 电子工业出版社, 2016.11  
ISBN 978-7-121-30287-9

I. ①分… II. ①周… III. ①分布式数据库—数据库系统 IV. ①TP311.133.1

中国版本图书馆 CIP 数据核字 (2016) 第 268198 号

策划编辑: 张国霞

责任编辑: 徐津平

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 20 字数: 450 千字

版 次: 2016 年 11 月第 1 版

印 次: 2016 年 11 月第 1 次印刷

印 数: 3000 册 定价: 79.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式: 010-51260888-819 [faq@phei.com.cn](mailto:faq@phei.com.cn)。

# 推荐序 1

随着大数据时代的到来，海量数据存储、并行计算、异构数据互联等一系列新技术在市场上不断地涌现。相信数据库行业的很多从业者都对传统关系型数据库的单点故障及容量问题头疼不已，而“分库分表”也早已成为解决这类问题的基础，此时，Mycat 应运而生。

Mycat 是一款面向企业级应用的开源数据库中间件产品，它目前支持数据库集群、分布式事务与 ACID，被普遍视为基于 MySQL 技术的集群分布式数据库解决方案，在一些互联网、金融、运营商客户中用来替代昂贵的 Oracle。

Mycat 不仅可以轻松对接 MySQL、SQL Server 等传统关系型数据库，也融合了内存缓存、NoSQL、HDFS 等新兴大数据技术，是一款非常优秀的数据库中间件。

在如今的大数据时代，分布式架构已经成为企业级数据应用的标配，传统的关系型数据库产品已经面临一个真正的拐点：一方面，关系型数据库自身难以实现分布式，这大大限制了其数据存储能力及整体的性能表现；另一方面，商业化的传统数据库产品的成本和性价比在分布式架构崛起的状况下毫无优势可言。因此，无论是从底层全新实现分布式计算存储的 NoSQL、Hadoop，还是使用 Mycat 这样的分库分表工具，对关系型数据库大刀阔斧地进行“改装”都是大势所趋。

作为一名专注于数据库领域多年的从业者，我认为 Mycat 从中间件工具的角度成功地弥补了 MySQL 的诸多局限。

- 分布式存储：通过 Mycat，MySQL 可以实现集群化与分布式管理，使数据库容量与处理能力大大改善。
- 性能加速：通过分布式集群及 Mycat Booster 对 MySQL 数据库在集群环境下的加速，Mycat 大大提升了 MySQL 集群的性能。
- 异构数据互联互通：除了 MySQL，Mycat 同时支持如 SequoiaDB、MongoDB 这样的 NoSQL 数据库及 HDFS 分布式文件系统，实现了对非结构化数据、半结构化数据及结

构化数据的存储及互联。

- 多样化的数据库工具：Mycat 为用户提供了丰富的管理工具，可以帮助用户更好地管理数据库系统。

本书非常适合作为 Mycat 的入门及进阶参考读物，它非常全面地阐述了分库分表的基本原理、实现机制及实践经验。本书的作者有着丰富的行业经验及技术底蕴，能够把业界非常前沿的知识用深入浅出的语言传授给各位读者。

最后，作为 SequoiaDB 的联合创始人，我十分钦佩 Mycat 团队的技术及勇气。虽然基础软件的开发难度很大，但是我们都敢于去挑战一个个技术难点，并填补国内基础软件产品的巨大空白。因此，我在这里衷心地祝愿 Mycat 前程似锦！

巨杉数据库联合创始人 王涛

## 推荐序 2

随着分布式系统的发展，应用的分布式由于无状态的特性，可以利用消息机制相对简单地进行拆分，计算的分布式也可以通过 Map、Reduce 等相关算法来解决。但是随着业务压力和并发压力的增加，我们急需一种分布式数据库解决方案来支持数据库的水平扩展，通过简单地增加服务器及线性地提升数据库的并发访问能力，为闯过分布式系统的最后一道难关铺平道路。

从阿里巴巴的 Cobar 到开源社区的 Mycat，从 Cobar 的架构师贺贤懋、朱海清、邱硕到 Mycat 的核心人员南哥、冰风影，作为一名专注于 MySQL 数据库十多年的从业者，我见证了分布式数据库的从无到有到百花齐放，在收到本书的序言邀请时，我感到非常荣幸。

现在的分布式数据库产品越来越多。YouTube 公司提供的 Vitess 功能强大，在 YouTube 的生产环境下支撑了大量的业务访问；360 公司的 Atlas 基于 MySQL Proxy 开发而成，最初主要在应用层进行透明的读写分离，于 2013 年引入了分库分表；陈菲在离开 360 公司后在 WPS 云平台用 Go 语言编写了 Kingshard；楼方鑫（黄忠）在离开支付宝后编写了 OneProxy；腾讯互娱的 DBA 团队基于 Spider 打造了自己的分布式数据库平台；淘宝在内部将 TDDL 的客户端工具作为了分库分表中间件；阿里巴巴的 B2B 开源了支撑其内部业务生产环境 3 年的 Cobar，为开源社区提供了一大助力；而基于 Cobar 开发的 Mycat 及其各种分支由于其易用性，将分布式数据库进一步推广到互联网和传统行业的各个业务领域。

Mycat 无疑是这些中间件中的佼佼者，支持百亿级别的数据分片和并行计算，支持高可用和 MySQL 的读写分离，并随着版本的更新进一步支持 Oracle、DB2、MongoDB 等后端数据库，随着周边产品的进一步成熟，在越来越多的分布式或者非分布式（仅用它的读写分离或者高可用）生产环境中得到部署，受到越来越多的企业的关注。本书恰逢其会，由 Mycat 核心开发人员撰写而成，详细讲述了 Mycat 的由来、架构特点、核心模块、实际使用案例和企业实践，是一本不可多得的好书。

沃趣科技 MySQL 负责人 李春

## 推荐序 3

作为国产开源数据库中间件——Mycat 的发起者，我不得不为本书作序。

这是一本由众多技术精英合著的数据库+中间件领域的专业书籍，这些人包括 Mycat Committer、Mycat 志愿者及资深 DBA，大家在工作之余抽出大量时间来编写和完善此书，历经一年完成了本书的编写工作，实属不易。

数据库中间件是新兴的重要的互联网中间件，目前业界仍然缺乏一本系统性介绍相关领域的软件产品、常用技术、架构等的纸质书籍。本书围绕 Mycat 开源中间件，从基础入门到架构原理，从运行机制到源码实现，从系统运维到应用实践，讲解得详尽而又完善。本书内容丰富、图文并茂、由浅入深，对数据库中间件的基本原理阐述清晰，对程序源码分析透彻，对实践经验讲解深刻。

从内容上讲，本书从一个使用者的角度去理解、分析和解决问题，通过大量的实例操作和源码解析，帮助读者深入理解 Mycat 的各种概念。读者对其中的案例只要稍做修改，再结合实际的业务需求，就可以在正在开发的项目中应用，达到事半功倍的效果。并且，通过学习书中应用实战方面的内容，不仅可以直接提高开发技能，还可以解决在实践过程中经常遇到的各种关键问题。另外，本书中的所有观点和经验均是作者们在多年建设、维护大型应用系统的过程中积累形成的，非常值得借鉴和推广。

希望 Mycat 在大家的支持下走得更远，成为中国软件的骄傲。

Leader-us

# 前言

随着移动互联网的兴起和大数据的蓬勃发展，系统的数据量正呈几何倍数增长，系统的压力也越来越大，这时最容易出现的问题就是服务器繁忙，我们可以通过增加服务器及改造系统来缓解压力，然后采用负载均衡、动静分离、缓存系统来提高系统的吞吐量。然而，当数据量的增长达到一定程度时，增加应用服务器并不能明显地提高系统的效率，因为所有压力都会传导到数据库层面，而大多数系统都是用一个数据库来存储和管理系统数据的。这时，Mycat 应运而生。

谈到 Mycat 就不得不谈谈 Cobar，Cobar 是阿里巴巴开源的数据库中间件，由于其存在使用限制及一些比较严重的问题，Leader-us 在其基础上于 2013 年年底实现了 Mycat 1.0 版本，Mycat 一经发布便引起了很多人的关注。之后 Mycat 社区对 Cobar 的代码进行了彻底重构，使用 NIO 重构了网络模块，并且优化了 Buffer 内核，增强了聚合、Join 等基本特性，同时兼容了绝大多数数据库，使之成为通用的数据库中间件。Mycat 在 1.4 版本以后完全脱离了 Cobar 内核，同时采用了 Mycat 集群管理、自动扩容及智能优化，成为了高性能的数据库中间件。Mycat 从诞生至今已有三年多了，一直在坚持做最好的开源数据库中间件产品。

本书总计 8 章，涵盖了 Mycat 入门、进阶、高级技术实战、企业运维、架构剖析、核心技术分析、多数据库支持原理与实现等内容，内容详尽、图文并茂，几乎囊括了 Mycat 所涉及的方方面面，无论是对于软件工程师、测试工程师、运维工程师、软件架构师、技术经理，还是对于资深 IT 人士来说，本书都极具参考价值。

第 1 章：介绍了分布式系统和分布式数据库系统的原理，介绍 Mycat 的起源和发展状况，并对各种数据库中间件做了简要介绍和对比。

第 2 章：讲解了 Mycat 的入门知识，介绍了 Mycat 的安装环境、核心概念和分库分表的原理，以及 Mycat 源码开发调试的过程。

第 3 章：讲解了 Mycat 的进阶知识，主要介绍 Mycat 的各种配置和分片算法。

第4章：讲解了 Mycat 和 MySQL 实战案例，由拥有丰富的 Mycat 线上实战经验的专家和 DBA 共同编写而成，有很高的参考价值。

第5章：简要介绍了用于 Mycat 性能监控的工具——Mycat-web，详细讲解了 Mycat 和 MySQL 的优化技术，是 DBA 的亲身总结和经验之谈。

第6章：重点阐述了 Mycat 的架构，包括网络、线程、连接池、内存管理及缓存实现等，是了解 Mycat 框架的基础。

第7章：介绍了 Mycat 的核心技术，包括分布式事务的实现、跨库 Join 的三种实现方式等，介绍了多节点数据汇聚和排序的原理，并详细阐述了在 Mycat 1.6 版本中实现的一致性分布式事务的功能。

第8章：介绍了 MySQL 和 PostgreSQL 的通信协议及 Mycat 对这些通信协议的实现，然后介绍了 Mycat 对 JDBC 及多种数据库的支持，例如 Oracle、SQL Server、MongoDB 等。

本书的编写和校对历经一年，参与编写的作者都是 Mycat 开源项目中参与度比较高、提交过不少代码或有丰富的实战经验的资深人士。非常感谢参与本书编写、指导或校对的专家：Leader-us、南哥（曹宗南）、从零开始（宋伟）、小张哥（张超）、yuanfang（杨鹏飞）、顽石神（张治春）、冰麒麟（杨峰）、望舒（胡雅辉）、明明 Ben（朱阿明）、零（章爱国）、little-pan（潘自朋）、CrazyPig（陈建欣）、毛茸茸的逻辑（王成瑞）、海王星（林志强）、石头狮子（林晔）、HanSenJ（姬文刚）、武（王灯武）、战狼（刘胡波）、KK（刘军）、董海雄（易班网）、arx（李秋伟）、正能量（王金剑）、吉光（李伟）。

由于作者的写作水平有限，书中难免会有不妥或者疏漏之处，欢迎读者批评指正。

冰风影

Mycat 社区负责人

2016 年 11 月 6 日于广州番禺

# 目 录

|                        |    |
|------------------------|----|
| 第 1 章 数据库中间件与分布式数据库的实现 | 1  |
| 1.1 什么是分布式系统           | 1  |
| 1.2 为什么需要分布式数据库        | 2  |
| 1.3 分布式数据库的实现原理        | 3  |
| 1.4 Mycat 数据库中间件简介     | 5  |
| 1.4.1 Mycat 的历史与未来规划   | 5  |
| 1.4.2 Mycat 与其他中间件的区别  | 8  |
| 1.4.3 Mycat 的优势        | 10 |
| 1.4.4 Mycat 的适用场合      | 11 |
| 第 2 章 Mycat 入门         | 13 |
| 2.1 环境搭建               | 13 |
| 2.1.1 Windows 环境搭建     | 13 |
| 2.1.2 Linux 环境搭建       | 15 |
| 2.2 Mycat 核心概念详解       | 16 |
| 2.2.1 逻辑库 (schema)     | 16 |
| 2.2.2 逻辑表 (table)      | 16 |

|       |                 |    |
|-------|-----------------|----|
| 2.2.3 | 分片节点 (dataNode) | 17 |
| 2.2.4 | 节点主机 (dataHost) | 17 |
| 2.3   | Mycat 原理介绍      | 18 |
| 2.4   | 参与 Mycat 源码开发   | 19 |
| 2.4.1 | Mycat 源码环境搭建    | 19 |
| 2.4.2 | Mycat 源码调试      | 19 |

## 第 3 章 Mycat 进阶 22

|        |                     |    |
|--------|---------------------|----|
| 3.1    | Mycat 配置详解          | 22 |
| 3.1.1  | Mycat 支持的两种配置方式     | 22 |
| 3.1.2  | server.xml 配置文件     | 23 |
| 3.1.3  | schema.xml 配置文件     | 28 |
| 3.1.4  | sequence 配置文件       | 37 |
| 3.1.5  | zk-create.yaml 配置文件 | 41 |
| 3.1.6  | 其他配置文件              | 44 |
| 3.2    | Mycat 分片规则详解        | 46 |
| 3.2.1  | 分片表与非分片表            | 46 |
| 3.2.2  | ER 关系分片表            | 46 |
| 3.2.3  | 分片规则 rule.xml 文件详解  | 46 |
| 3.2.4  | 取模分片                | 47 |
| 3.2.5  | 枚举分片                | 48 |
| 3.2.6  | 范围分片                | 49 |
| 3.2.7  | 范围求模算法              | 49 |
| 3.2.8  | 固定分片 hash 算法        | 50 |
| 3.2.9  | 取模范围算法              | 52 |
| 3.2.10 | 字符串 hash 求模范围算法     | 53 |
| 3.2.11 | 应用指定的算法             | 54 |

|        |               |    |
|--------|---------------|----|
| 3.2.12 | 字符串 hash 解析算法 | 54 |
| 3.2.13 | 一致性 hash 算法   | 55 |
| 3.2.14 | 按日期(天)分片算法    | 56 |
| 3.2.15 | 按单月小时算法       | 57 |
| 3.2.16 | 自然月分片算法       | 58 |
| 3.2.17 | 日期范围 hash 算法  | 58 |
| 3.3    | Mycat 管理命令详解  | 59 |
| 3.3.1  | Reload 命令     | 61 |
| 3.3.2  | Show 命令       | 62 |

## 第 4 章 Mycat 高级技术实战

68

|       |                                  |     |
|-------|----------------------------------|-----|
| 4.1   | 用 Mycat 搭建读写分离                   | 68  |
| 4.1.1 | MySQL 读写分离                       | 69  |
| 4.1.2 | MySQL Galera Cluster 读写分离        | 73  |
| 4.1.3 | SQL Server 读写分离                  | 83  |
| 4.2   | Mycat 故障切换                       | 86  |
| 4.2.1 | Mycat 主从切换                       | 86  |
| 4.2.2 | MySQL Galera 节点切换                | 99  |
| 4.3   | Mycat+Percona+HAProxy+Keepalived | 113 |
| 4.3.1 | Mycat                            | 113 |
| 4.3.2 | Percona 集群                       | 124 |
| 4.3.3 | HAProxy                          | 131 |
| 4.3.4 | Keepalived                       | 138 |
| 4.4   | MHA+Keepalived 集群搭建              | 140 |
| 4.4.1 | 配置 MySQL 半同步方式                   | 142 |
| 4.4.2 | 安装配置 MHA                         | 150 |
| 4.4.3 | 测试重构                             | 153 |

|                                |     |
|--------------------------------|-----|
| 4.4.4 扩展 Keepalived            | 155 |
| 4.5 用 ZooKeeper 搭建 Mycat 高可用集群 | 158 |
| 4.5.1 ZooKeeper 概述             | 158 |
| 4.5.2 ZooKeeper 的运用场景          | 161 |
| 4.5.3 ZooKeeper 在 Mycat 中的使用   | 163 |
| 4.6 Mycat 高可用配置                | 165 |
| 4.7 Mycat 注解技术                 | 170 |
| 4.7.1 balance 注解实战             | 170 |
| 4.7.2 master/slave 注解实战        | 172 |
| 4.7.3 SQL 注解实战                 | 173 |
| 4.7.4 schema 注解实战              | 176 |
| 4.7.5 dataNode 注解实战            | 176 |
| 4.7.6 catlet 注解实战              | 177 |

---

## 第 5 章 Mycat 企业运维 179

---

|                              |     |
|------------------------------|-----|
| 5.1 Mycat 性能监控——Mycat-web 详解 | 179 |
| 5.1.1 Mycat-web 简介           | 179 |
| 5.1.2 Mycat-web 的配置和使用       | 180 |
| 5.1.3 Mycat 性能监控指标           | 181 |
| 5.2 Mycat 性能优化               | 183 |
| 5.3 MySQL 优化技术               | 186 |
| 5.3.1 数据库建表设计规范              | 186 |
| 5.3.2 SQL 语句与索引              | 195 |
| 5.3.3 配置文件                   | 206 |
| 5.3.4 InnoDB 选择文件系统          | 212 |
| 5.3.5 系统架构                   | 213 |

|                        |     |
|------------------------|-----|
| 第 6 章 Mycat 架构剖析       | 215 |
| 6.1 Mycat 总体架构介绍       | 215 |
| 6.2 Mycat 网络 I/O 架构与实现 | 218 |
| 6.2.1 Mycat I/O 架构概述   | 218 |
| 6.2.2 前端通信框架           | 221 |
| 6.3 Mycat 线程架构与实现      | 224 |
| 6.3.1 多线程基础            | 224 |
| 6.3.2 Mycat 线程架构       | 226 |
| 6.4 Mycat 内存管理及缓存架构与实现 | 228 |
| 6.4.1 Mycat 内存管理       | 229 |
| 6.4.2 Mycat 缓存架构与实现    | 231 |
| 6.5 Mycat 连接池架构与实现     | 232 |
| 6.5.1 Mycat 连接池        | 232 |
| 6.5.2 Mycat 连接池架构及代码实现 | 234 |
| 6.6 Mycat 主从切换架构与实现    | 235 |
| 6.6.1 Mycat 主从切换概述     | 236 |
| 6.6.2 Mycat 主从切换的实现    | 238 |
| 第 7 章 Mycat 核心技术分析     | 241 |
| 7.1 Mycat 分布式事务的实现     | 241 |
| 7.1.1 XA 规范            | 241 |
| 7.1.2 二阶段提交            | 242 |
| 7.1.3 三阶段提交            | 243 |
| 7.1.4 Mycat 中分布式事务的实现  | 244 |
| 7.2 Mycat SQL 路由的实现    | 249 |
| 7.2.1 路由的作用            | 249 |

|       |                   |     |
|-------|-------------------|-----|
| 7.2.2 | SQL 解析器           | 250 |
| 7.2.3 | 路由计算              | 252 |
| 7.3   | Mycat 跨库 Join 的实现 | 260 |
| 7.3.1 | 全局表               | 261 |
| 7.3.2 | ER 分片             | 262 |
| 7.3.3 | catlet            | 263 |
| 7.3.4 | ShareJoin         | 264 |
| 7.4   | Mycat 数据汇聚和排序的实现  | 270 |
| 7.4.1 | 数据排序              | 270 |
| 7.4.2 | 数据汇聚              | 273 |

## 第 8 章 Mycat 多数据库支持原理与实现 275

|       |                           |     |
|-------|---------------------------|-----|
| 8.1   | MySQL 协议在 Mycat 中的实现      | 275 |
| 8.1.1 | MySQL 协议概述                | 275 |
| 8.1.2 | Mycat 的 MySQL 协议实现        | 283 |
| 8.2   | PostgreSQL 协议在 Mycat 中的实现 | 287 |
| 8.2.1 | PostgreSQL 介绍             | 287 |
| 8.2.2 | PostgreSQL 协议             | 288 |
| 8.2.3 | PostgreSQL 实现             | 293 |
| 8.3   | Mycat 对 JDBC 支持的实现        | 298 |
| 8.3.1 | Oracle 配置                 | 299 |
| 8.3.2 | SQL Server 配置             | 300 |
| 8.3.3 | MongoDB 配置                | 301 |
| 8.3.4 | 源码分析                      | 306 |

# 第 1 章

## 数据库中间件与分布式数据库的实现

---

从第一台计算机产生至今的半个多世纪里，计算机已经深入人们的生活，在各个领域得到广泛应用，并不断地改变人们的生活方式。从电子管到晶体管，从集成电路到超大规模集成电路，从单个 CPU、存储器、外设和一些终端在内的集中式计算到大量 CPU 通过高速网络连接组成的并行计算，计算机的发展有着惊人的进步。

### 1.1 什么是分布式系统

分布式系统是指其组件分布在网络上，组件之间通过传递消息进行通信和动作协调的系统。它的核心理念是让多台服务器协同工作，完成单台服务器无法处理的任务，尤其是高并发或者大数据量的任务。它的主要特点如下。

- **透明性：**分布式系统对于用户来说是透明的，一个分布式系统在用户面前的表现就像一个传统的单处理机分时系统，可让用户不必了解其内部结构就能使用。
- **扩展性：**分布式系统的最大特点是可扩展性，它能够根据需求的增加而扩展，可以通过横向扩展使集群的整体性能得到线性提升，也可以通过纵向扩展单台服务器的性能（上升空间有限，使用不多）使服务器集群的性能得到提升。
- **可靠性：**分布式系统不允许单点失效的问题存在，它的基本思想是，如果一台机器坏

了，则其他机器能够接替它进行工作，具有持续服务的特性。

- **高性能：**高性能是人们设计分布式系统的一个初衷，如果建立了一个透明、灵活、可靠的分布式系统，但是它运行起来像蜗牛一样慢，那么这个分布式系统是失败的。

分布式系统在拥有众多优点的同时自然有其缺点。

- 在节点通信部分的开销比较大，线程安全问题也变得复杂，需要在保证数据完整性的同时兼顾性能。
- 过分依赖网络，网络信息的丢失或饱和将会抵消分布式系统的大部分优势。
- 有潜在的数据安全和网络安全等安全性问题。

下面讲讲分布式系统与集中式系统的区别。首先，简单地讲，分布式系统采用并行计算，而集中式系统采用串行计算，这是二者本质上的区别；其次，分布式系统的性价比高，对于处理大规模数据而言，最节约成本的办法是在一个系统中使用集中在一起的大量的廉价 CPU；然后，分布式系统的可靠性高，通过把负载分散到众多机器上，单个芯片发生故障时最多会使一台机器停止运行，而其他机器不会受到影响；另外，分布式系统比集中式系统的可扩展性强，可根据业务量的增长逐渐扩展系统的计算能力；最后，从性能上讲，分布式系统的计算能力比单个大型主机更强。

## 1.2 为什么需要分布式数据库

随着计算机和信息技术的迅猛发展和普及，行业应用系统的规模迅速扩大，行业应用所产生的数据量呈爆炸式增长，动辄达到数百 TB 甚至数百 PB 的规模，已远远超出现有的传统计算技术和信息系统的处理能力，而集中式数据库面对大规模数据处理逐渐表现出其局限性。因此，人们希望寻找一种能快速处理数据和及时响应用户的访问的方法，也希望对数据进行集中分析、管理和维护。这已成为现实世界的迫切需求。

分布式数据库是在集中式数据库的基础上发展起来的，是计算机技术和网络技术结合的产物。分布式数据库是指数据在物理上分布而在逻辑上集中管理的数据库系统。物理上分布是指分布式数据库的数据分布在物理位置不同并由网络连接的节点或站点上；逻辑上集中是指各数据库节点之间在逻辑上是一个整体，并由统一的数据库管理系统管理。不同的节点分布可以跨不同的机房、城市甚至国家。

分布式数据库的主要特点如下。

- **透明性：**用户不必关心数据的逻辑分区和物理位置分布的细节，也不必关心重复副本