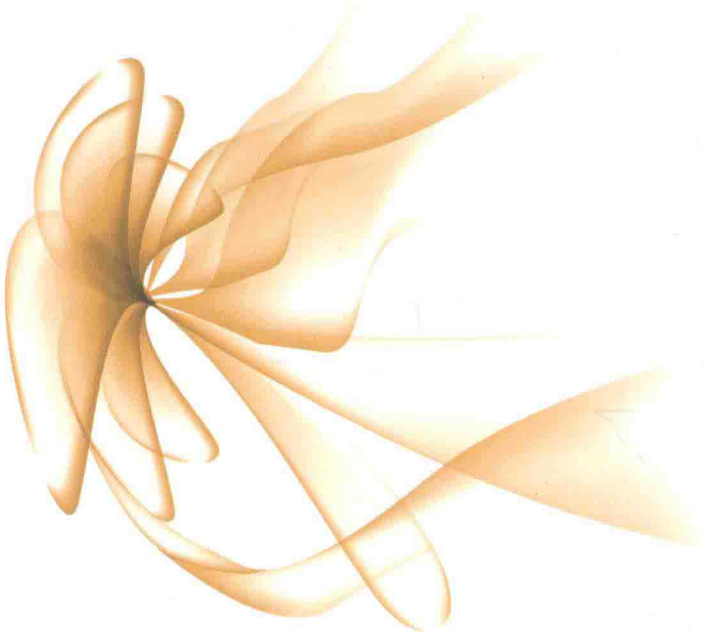
HZ BOOKS
华章科技

绝技源于江湖、将军发于卒伍，本书包含作者从程序员到首席架构师十多年职业生涯所积累的实战经验。

这不是一本讲怎么使用Hadoop的书，而是一本讲实现Hadoop功能的书，本书系统讲解构建大规模分布式系统的核心技术和实现方法，包含开源的代码，手把手教你掌握分布式技术。

大数据
技术丛书



Architecture and Design of Large Scale Distributed System

大规模分布式系统架构与设计实战

彭 渊◎著

CD-ROM

机械工业出版社
China Machine Press

技术丛书

Architecture and Design of Large Scale Distributed System

大规模分布式系统架构与设计实战

彭 渊◎著

机械工业出版社

图书在版编目（CIP）数据

大规模分布式系统架构与设计实战 / 彭渊著. —北京：机械工业出版社，2014.1
（大数据技术丛书）

ISBN 978-7-111-45503-5

I. 大… II. 彭… III. 分布式计算机系统–系统设计 IV. TP338.8

中国版本图书馆CIP 数据核字（2014）第013468号

本书是作者从程序员到首席架构师十多年职业生涯的实战经验总结，系统讲解构建大规模分布式系统的核心技术与实现方法，包含作者开源的 Fourinone 系统的设计与实现过程，手把手教你掌握分布式技术。通过学习这个系统的实现方法与相关的理论，读者可快速掌握分布式系统的理论并设计自己的分布式系统。

本书从分布式计算的基本概念开始，解剖了众多流行概念的本质，深入讲解分布式系统的基本原理与实现方式，包括 master-slave 结构、消息中枢模式、网状直接交互模式、并行结合串行模式等，以及 Fourinone 系统的架构、实现分布式功能的示例。接下来详细介绍分布式协调、分布式缓存、消息队列、分布式文件系统、分布式作业调度平台的设计与实现方法，不仅包括详细的架构原理、算法，还给出了实现步骤、核心 API、实现代码。随书附带的光盘包括书中示例代码以及 Fourinone 系统源代码。

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
2014 年 2 月第 1 版第 1 次印刷
186mm × 240mm • 15 印张
标准书号：ISBN 978-7-111-45503-5
ISBN 978-7-89405-278-0（光盘）
定 价：59.00 元（附光盘）

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

前　言

在大数据、云计算如火如荼的今天，各类技术产品顺应潮流层出不穷。大家是不是有这种感觉：Hadoop 还没学完，Storm 就来了；Storm 刚学会安装配置，Spark、Hama、Yarn 等又一起出现了；同时国内外各大云平台厂商，如 Google、亚马逊、阿里云等，还在推各自应用开发平台……要学习的东西太多了，就是这样疲于奔命地学，刚学会了某个产品的安装配置与开发步骤，没多久它又过时了。

这么多千姿百态的分布式技术和产品背后有没有某些共性的东西呢？能让我们换了马甲还能认出它，让我们超越学习每个产品的“安装配置开发”而掌握背后的精髓呢？有没有可能学一反三，学一招应万招，牢牢掌握好技术的船舵，穿越一次次颠覆性的技术浪潮？本书的目的就是为你揭示分布式技术的核心内幕，透彻理解其精髓，站在浪潮之巅。

因此，这不是一本讲如何使用 Hadoop 的书，而是一本讲实现 Hadoop 功能的书，是一本讲如何简化实现分布式技术核心功能的书。这不是一本空谈概念、四处摘抄的书，而是来源于作者十多年来在私企、港企、外包、创业、淘宝、华为等企业打拼，从底层程序员一路走到首席架构师的实战经验总结。绝技源于江湖，将军发于卒伍，这本书讲的是你在课本上学不到核心技术，无论你是在中国什么样的 IT 企业做什么样的分布式应用，这本书对你都具备参考性。

本书面向千千万万战斗在一线攻城拔寨的程序员、工程师们，你可以有很多基础，也可以从头开始，本书尽量做到深入浅出和通俗易懂，希望帮助你降低分布式技术的学习成本，帮助你更容易完成工作任务，更轻松地挣钱。

本书根据分布式技术的主要应用，分别介绍分布式并行计算的基本概念、分布式协调、分布式缓存、消息队列、分布式文件系统、分布式作业调度平台等，详细阐述分布式各技术的架构原理和实现方式，并附带大量示例，便于读者实际操作运行。基于本书原理，作者用 Java 实现并开源了 Fourinone 框架，这是一个高效的分布式系统，归纳在 150KB 源码里，代码不到 1 万行，让你能够轻松掌握。学习开发核心技术的诀窍是多动手，建议读者运行本书附带的大量 DEMO，在运行后细细体会分布式的理论，进行反思和总结。本书归纳的设计思想和算法不局限于某个框架，读者领会

后可以用任何语言来实现自己的分布式系统。

本书各章有一定的独立性，阅读本书的方式比较自由，可以从头开始，也可以随性翻阅。从第2章开始，每章都有理论部分与示例，读者可以先运行DEMO，不清楚的地方再回看原理；也可以先看原理，再运行DEMO加深理解。由于时间的限制，且本书写作的时期是在作者最为忙碌和事业的转折时期，匆忙中，难免出错，请朋友们海涵，并提出意见以便于今后纠正。最后感谢机械出版社华章公司所有幕后编辑的大量工作，感谢所给予我帮助与支持的领导和朋友。

本书所有源码附带在光盘里。你也可以登录开源地址下载，开源地址：http://code.google.com/p/fourinone

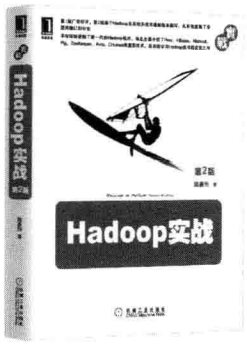
作者联系方式：邮箱：Fourinone@yeah.net

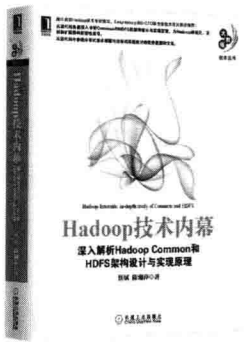
QQ群1：1313859

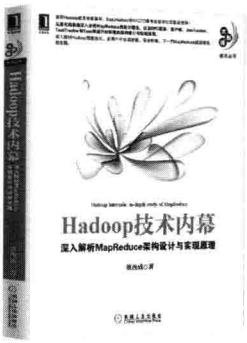
QQ群2：241116021

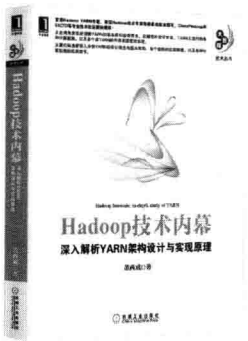
QQ群3：23321760

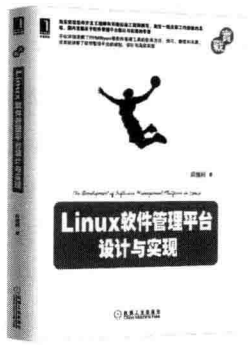
推荐阅读

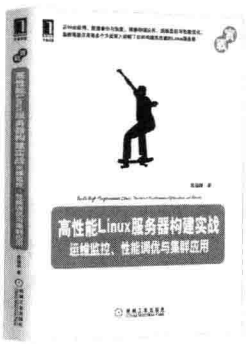

Hadoop实战
第2版
Hadoop实战

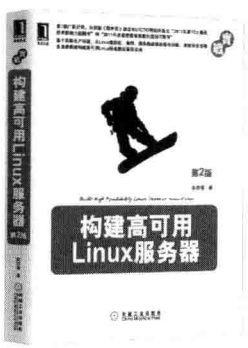

Hadoop技术内幕
深入解析Hadoop Common和
HDFS架构设计与实现原理

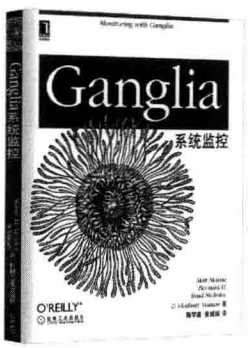

Hadoop技术内幕
深入解析MapReduce架构设计与实现原理


Hadoop技术内幕
深入解析YARN架构设计与实现原理


Linux软件管理平台
设计与实现


高性能Linux服务器构建实战
运维监控、性能调优与集群应用


构建高可用Linux服务器
第2版
构建高可用
Linux服务器


Ganglia
系统监控
O'REILLY

目　录

前　言

第1章　概述 …… 1

1.1　分布式计算、并行计算、云计算概述 …… 1
1.2　分布式产品Hadoop、ZooKeeper、HBase概述 …… 6
1.3　Fourinone的产生背景 …… 12

第2章　分布式并行计算的原理与实践 …… 14

2.1　分布式并行计算模式 …… 14
2.1.1　最初想到的master-slave结构 …… 14
2.1.2　“包工头-职介所-手工仓库-工人”模式 …… 15
2.1.3　基于消息中枢的计算模式 …… 17
2.1.4　基于网状直接交互的计算模式 …… 18
2.1.5　并行结合串行模式 …… 22
2.1.6　包工头内部批量多阶段处理模式 …… 23
2.1.7　计算集群模式和兼容遗留计算系统 …… 24
2.1.8　工人计算的服务化模式 …… 26
2.2　跟Hadoop的区别 …… 28
2.3　关于分布式的一些概念与产品 …… 30
2.4　配置文件和核心API介绍 …… 35
2.5　实践与应用 …… 36
2.5.1　一个简单的示例 …… 36
2.5.2　工头工人计算模式更完整的示例 …… 39
2.5.3　工人合并互相say hello的示例 …… 44

2.5.4 实现Hadoop经典实例Word Count …… 48
2.5.5 分布式多机部署的示例 …… 52
2.5.6 分布式计算自动部署的示例 …… 53
2.5.7 计算过程中的故障和容灾处理 …… 57
2.5.8 计算过程中的相关时间属性设置 …… 60
2.5.9 如何在一台计算机上一次性启动多个进程 …… 63
2.5.10 如何调用C/C++程序实现 …… 68
2.5.11 如何中止工人计算和超时中止 …… 68
2.5.12 使用并行计算大幅提升递归算法效率 …… 73
2.5.13 使用并行计算求圆周率π …… 81
2.5.14 从赌钱游戏看PageRank算法 …… 86
2.5.15 使用并行计算实现上亿排序 …… 96
2.5.16 工人服务化模式应用示例 …… 104
2.6 实时流计算 …… 107

第3章 分布式协调的实现 …… 111

3.1 协调架构原理简介 …… 111
3.2 核心API …… 113
3.3 权限机制 …… 115
3.4 相对于ZooKeeper的区别 …… 116
3.5 与Paxos算法的区别 …… 117
3.6 实践与应用 …… 119
3.6.1 如何实现公共配置管理 …… 119
3.6.2 如何实现分布式锁 …… 126
3.6.3 如何实现集群管理 …… 129
3.6.4 多节点权限操作示例 …… 134
3.6.5 领导者选举相关属性设置 …… 137

第4章 分布式缓存的实现 …… 139

4.1 小型网站或企业应用的缓存实现架构 …… 139
4.2 大型分布式缓存系统实现过程 …… 140
4.3 一致性哈希算法的原理、改进和实现 …… 147

4.4 解决任意扩容的问题 …… 152
4.5 解决扩容后数据均匀的问题 …… 153
4.6 分布式Session的架构设计和实现 …… 154
4.7 缓存容量的相关属性设置 …… 156
4.8 缓存清空的相关属性设置 …… 158

第5章 消息队列的实现 …… 162

5.1 闲话中间件与MQ …… 162
5.2 JMS的两种经典模式 …… 163
5.3 如何实现发送接收的队列模式 …… 164
5.4 如何实现主题订阅模式 …… 168

第6章 分布式文件系统的实现 …… 173

6.1 FTTP架构原理解析 …… 174
6.2 搭建配置FttpAdapter环境 …… 177
6.3 访问集群文件根目录 …… 179
6.4 访问和操作远程文件 …… 181
6.5 集群内文件复制和并行复制 …… 184
6.6 读写远程文件 …… 187
6.7 解析远程文件 …… 189
6.8 并行读写远程文件 …… 191
6.9 批量并行读写远程文件和事务补偿处理 …… 194
6.10 如何进行整型读写 …… 198
6.11 基于整型读写的上亿排序 …… 205

第7章 分布式作业调度平台的实现 …… 219

7.1 调度平台的设计与实现 …… 219
7.2 资源隔离的实现 …… 224
7.3 资源调度算法 …… 226
7.4 其他作业调度平台简介 …… 227
7.4.1 其他MPI作业资源调度技术 …… 227
7.4.2 Mesos和Yarn简介 …… 229

第 1 章
概　　述

在概述分布式核心技术之前，我们有必要先概括阐述一下分布式计算、并行计算、云计算等相关概念，以及市场上流行的相关技术产品，如 Hadoop 生态体系，然后再结合背景引出我们为什么要归纳出一个轻量级的分布式框架。本章为后续章节的背景。本章意在使读者对分布式技术话题的前因后果先有所了解。

由于只是概述，我们对涉及的分布式计算概念和 Hadoop 生态体系只是蜻蜓点水地带过，目的仅是让读者了解到这些内容大致是什么，详细的使用方法和开发方法可以参考其他书籍。

1.1　分布式计算、并行计算、云计算概述

1. 什么是分布式计算

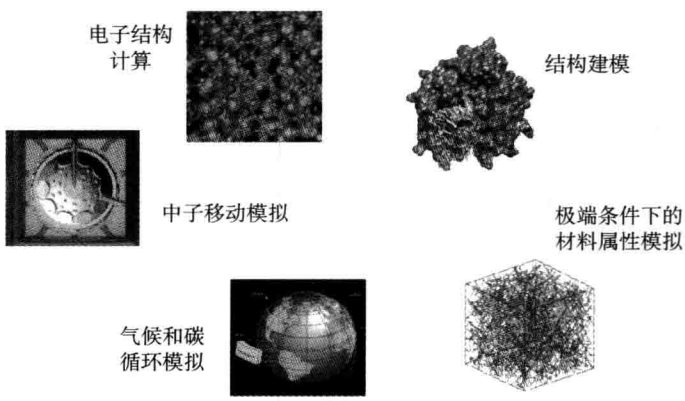


图 1-1　科学中的分布式计算

经科学研究发现：目前存在很多万亿次计算实例，其中涉及的问题都需要非常巨大的计算能力才能解决。这类问题很多还是跨学科的、极富挑战性的、人类亟待解决的科研课题，如图 1-1 所示。

除此之外还有很多研究项目需要巨大的计算能力，例如：

1）解决较为复杂的数学问题，例如：GIMPS（寻找最大的梅森素数）。

梅森素数 (Mersenne Prime）是指形如 2^p − 1 的正整数，其中指数 p 是素数，常记为 Mp。若 Mp 是素数，则称为梅森素数。p=2，3，5，7 时，Mp 都是素数，但 M11=2047=23×89 不是素数，是否有无穷多个梅森素数是数论中未解决的难题之一。截至 2012 年 7 月已累计发现 47 个梅森素数，最大的是 p=43,112,609，此时 Mp 是一个 12,978,189 位数。

值得一提的是，中国的一位数学家算出了梅森素数的分布规律图，并用简练的数学公式描述了出来。如果借助计算机的并行计算，也许会对寻找该数字分布规律有所帮助。

2）研究寻找最为安全的密码系统，例如：RC-72（密码破解）。

3）生物病理研究，例如：Folding@home（研究蛋白质折叠、误解、聚合，以及由此引起的相关疾病）。

4）各种各样疾病的药物研究，例如：United Devices（对抗癌症的有效药物）。

5）信号处理，例如：SETI@Home（寻找地球外文明）。

由上不难看出，这些项目都很庞大，都需要惊人的计算量，仅由单个电脑或个人在一个能让人接受的时间内计算完成是决不可能的。在以前，这些问题都应该由超级计算机来解决。但是，超级计算机的造价和维护非常昂贵，这不是一个普通的科研组织能承受的。随着科学的发展，一种廉价的、高效的、维护方便的计算方法应运而生——分布式计算！

所谓**分布式计算**其实就是一门计算机科学，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得到最终的结果。

最近的一个分布式计算项目已经使用世界各地成千上万位志愿者的计算机来进行操作，利用这些闲置计算能力，通过因特网，你可以分析来自外太空的电信号，并探索可能存在的外星智慧生命；你可以寻找超过 1000 万位数字的梅森素数；你也可以寻找并发现对抗艾滋病病毒的更为有效的药物。

讨论

我们能否利用访问淘宝网的几千万个用户的电脑做一次分布式计算？

这里仅仅点拨一下，回答该问题其实涉及侵犯用户隐私。用户用电脑上网，安装了很多客户端软件，有的软件是拥有本地所有权限的，它们是否在偷偷用用户的电脑干其他私活，用户也不知道。这曾经引发过国内两大客户端巨头的官司。但是由此可以看出，千千万万用户的电脑是可

以利用起来做计算的，当然计算必须围绕一个消息中枢模式进行，因为用户电脑间的网络结构千差万别，无法直接连接。

2. 什么是并行计算

并行计算其实早就有了，所有大型编程语言都支持多线程，多线程就是一种简单的并行计算方式，多个程序线程并行地争抢 CPU 时间。

并行计算（Parallel Computing）是指同时使用多种计算资源解决计算问题的过程。并行计算的主要目的是快速解决大型且复杂的计算问题。此外还包括：利用非本地资源节约成本，即使用多个“廉价”计算资源取代大型计算机，同时克服单个计算机上存在的存储器限制问题。

传统上，串行计算是指在单个计算机（具有单个中央处理单元）上执行软件写操作。CPU 逐个使用一系列指令解决问题，但在每一个时刻只能执行一种指令。并行计算是在串行计算的基础上演变而来的，它努力仿真自然世界中的事务状态：一个序列中众多同时发生的、复杂且相关的事件。

为利用并行计算，通常计算问题表现为以下特征：

- 将工作分解成离散部分，有助于同时解决；
- 随时并及时地执行多个程序指令；
- 多计算资源下解决问题的耗时要少于单个计算资源下的耗时。

并行计算是相对于串行计算来说的，所谓并行计算分为时间上的并行和空间上的并行。时间上的并行就是指流水线技术，而空间上的并行则是指用多个处理器并发地执行计算。

3. 并行计算与串行计算的关系

并行计算与串行计算的关系如图 1-2 所示。

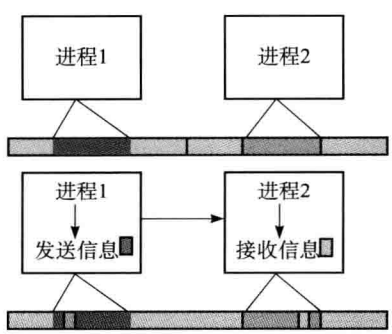


图 1-2 串行（上图），并行（下图）

结合图 1-2，对串行计算和并行计算分析如下：

- 传统的串行计算，分为“指令”和“数据”两个部分，并在程序执行时“独立地申请和占有”内存空间，且所有计算均局限于该内存空间。
- 并行计算将进程相对独立的分配于不同的节点上，由各自独立的操作系统调度，享有独立的 CPU 和内存资源（内存可以共享）；进程间相互信息交换是通过消息传递进行的。

4. 什么是云计算

云计算是一种理念，是旧瓶子装新酒，它实际上是分布式技术 + 服务化技术 + 资源隔离和管理技术（虚拟化），如图 1-3 所示。商业公司对云计算都有自己的定义，例如：

- 一种计算模式：把 IT 资源、数据、应用作为服务通过网络提供给用户（如 IBM 公司）。
- 一种基础架构管理方法论：把大量的高度虚拟化的资源管理起来，组成一个大的资源池，用来统一提供服务（如 IBM 公司）。
- 以公开的标准和服务为基础，以互联网为中心，提供安全 、快速、便捷的数据存储和网络计算服务（如 Google 公司）。

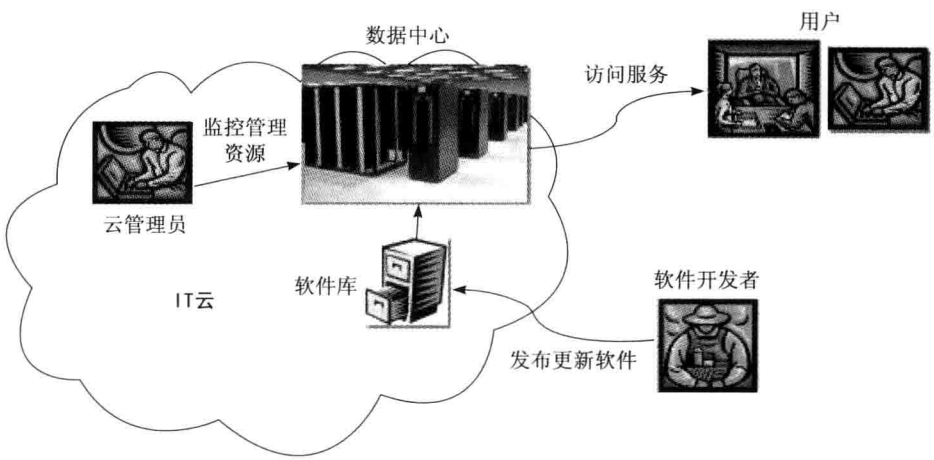


图 1-3　云计算示意图

通俗意义上的云计算往往是上面这个架构图包含的内容，开发者利用云 API 开发应用，然后上传到云上托管，并提供给用户使用，而不关心云背后的运维和管理，以及机器资源分配等问题。

虚拟化和服务化是云计算的表现形式（参见图 1-4）：

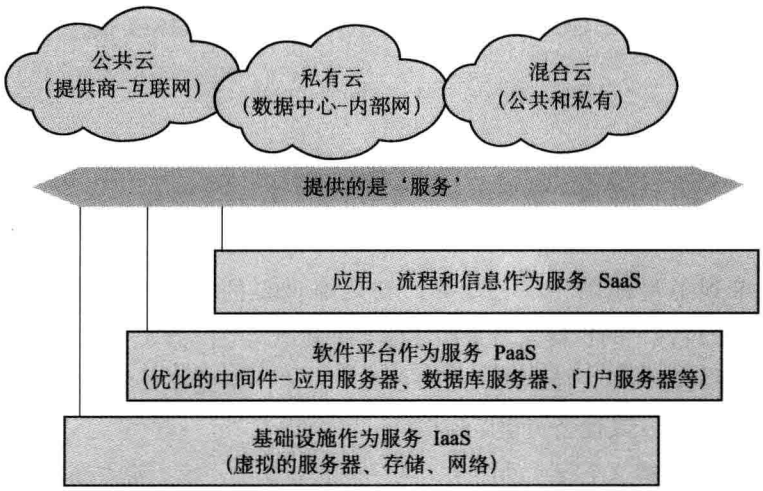


图 1-4 云计算表现形式

- 虚拟化技术包括：资源虚拟化、统一分配监测资源、向资源池中添加资源。虚拟化的技术非常多，有的是完全模拟硬件的方式去运行整个操作系统，比如我们熟悉的VMWare，可以看做重量级虚拟化产品。也有通过软件实现的，共享一个操作系统的轻量级虚拟化，比如 Solaris 的 Container、Linux 的 lxc（关于 cgroup 的方式我们在 7.2 节也会谈到）。虚拟化的管理、运维多数是通过工具完成的，比如 Linux 的 VirtManager、VMWare 的 vSphere、VMWare 的 vCloud 等等。
- 服务思想包括：
  - 软件即服务（Software-as-a-Service，SAAS）。是目前最为成熟的云计算服务模式。在这种模式下，应用软件安装在厂商或者服务供应商那里，用户可以通过某个网络来使用这些软件。这种模式具有高度的灵活性、可靠性和可扩展性，因此能够降低客户的维护成本和投入，而且运营成本也得以降低。最著名的例子就是 Salesforce.com。
  - 平台即服务（Platform-as-a-Service，PAAS）。提供了开发平台和相关组件，软件开发者可以在这个开发平台之上开发新的应用，或者使用已有的各种组件，因此可以不必购买开发、质量控制或生产服务器。Salesforce.com 的 Force.com、Google 的 App Engine 和微软的 Azure（微软云计算平台）都采用了 PAAS 的模式。
  - 基础设施作为服务（Infrastructure as a Service，IAAS）。通过互联网提供了数据中心、硬件和软件基础设施资源。IAAS 可以提供服务器、操作系统、磁盘存储、数据库和/或信息资源。用户可以像购买水电煤气一样购买这些基础设施资源使用。

1.2 分布式产品 Hadoop、ZooKeeper、HBase 概述

1. Hadoop

说到云计算技术和产品，不能不提到 Google 这家企业。曾经，微软是 IT 行业的象征，号称只招最聪明的人。十年后，微软逐渐疲软了下来，Google 这家企业取而代之号称只招最聪明的人。

从搜索引擎到卫星地图，到云计算，到风靡世界的 Android，到现在的无人汽车、Google 眼镜，以及传说中的机器人之父和在他家里满地爬的机器人……这个行业总有这么一家企业成为最高科技的领袖，控制着大部分的核心技术，聚拢着大部分的一流人才，垄断着最大份额的市场，几乎让其他公司望而却步。

因此，我们不难理解这个行业的结果，第一的企业吃肉，第二喝点汤，其他都亏损。

也许在学校读书时，考试成绩前十名都是好学生，但是 IT 行业的社会竞争，必须要做到第一，成为所在领域的老大才有利可图。

再简单回顾一下云计算相关技术产品的发展史：

- 2002 ～ 2004：Apache Nutch。
- 2004 ～ 2006：Google 发表 GFS 和 MapReduce 相关论文。Apache 在 Nutch 中实现 HDFS 和 MapReduce。
- 2006 ～ 2008：Hadoop 项目从 Nutch 中分离。
- 2008 年 7 月，Hadoop 赢得 Terabyte Sort Benchmark。

从上面我们可以看到早在 2004 ～ 2006 年，Google 就发表了两篇与 GFS 和 MapReduce 相关的论文，分别是分布式文件系统和基于它的 Map/Reduce 并行计算。Apache 的开源项目 Hadoop 便是根据这两篇论文的思想实现的 Java 版本，Hadoop 引起关注是在它赢得了一次 TB 排序比赛：Hadoop Wins Terabyte Sort Benchmark：One of Yahoo's Hadoop clusters sorted 1 terabyte of data in 209 seconds, which beat the previous record of 297 seconds in the annual general purpose (Daytona) terabyte sort benchmark. This is the first time that either a Java or an open source program has won.

用了大量的机器在 209 秒完成 1TB 排序，提升了之前 297 秒的记录。

值的一提的是，Hadoop 的作者 Doug Cutting 同时也是 Lucene 和 Nutch 的作者，早年供职 Yahoo，后来担任 Apache 软件基金会主席。

Hadoop 的名字来源于 Doug Cutting 儿子的一个宠物名。Hadoop 从最初的 HDFS 分布式文件系统发展到后来的 Hadoop+ZooKeeper+Hive+Pig+HBase 生态体系。

HDFS 提供了一个可扩缩的、容错的、可以在廉价机器上运行的分布式文件系统，按行

进行存储，按 64MB 块进行文件拆分。

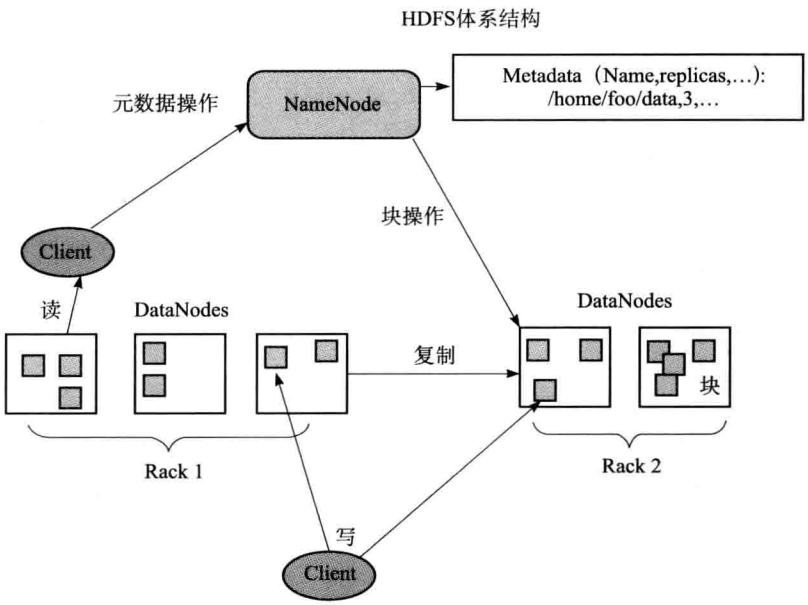


图 1-5 HDFS 体系结构

我们可以看到，HDFS 的架构是一个 NameNode 和多个 DataNode 的结构（参见图 1-5）：

❑ NameNode

存储 HDFS 的元数据 (metadata)。

管理文件系统的命名空间（ namespace)。

创建、删除、移动、重命名文件和文件夹。

接收从 DataNode 来的 Heartbeat 和 Blockreport。

❑ DataNode

存储数据块。

执行从 NameNode 来的文件操作命令。

定时向 NameNode 发送 Heartbeat 和 Blockreport。

除了提供分布式文件存储外，Hadoop 还提供基于 Map/Reduce 的框架，进行按行的并行分析，可以用来查询和计算。

图 1-6 是以 Word Count 为例子演示 Map/Reduce 机制的图，学习过 Hadoop 的人一般都看到过，有趣的是，我们在 2.1.4 节也会基于 Fourinone 的方式实现一个 Word Count。

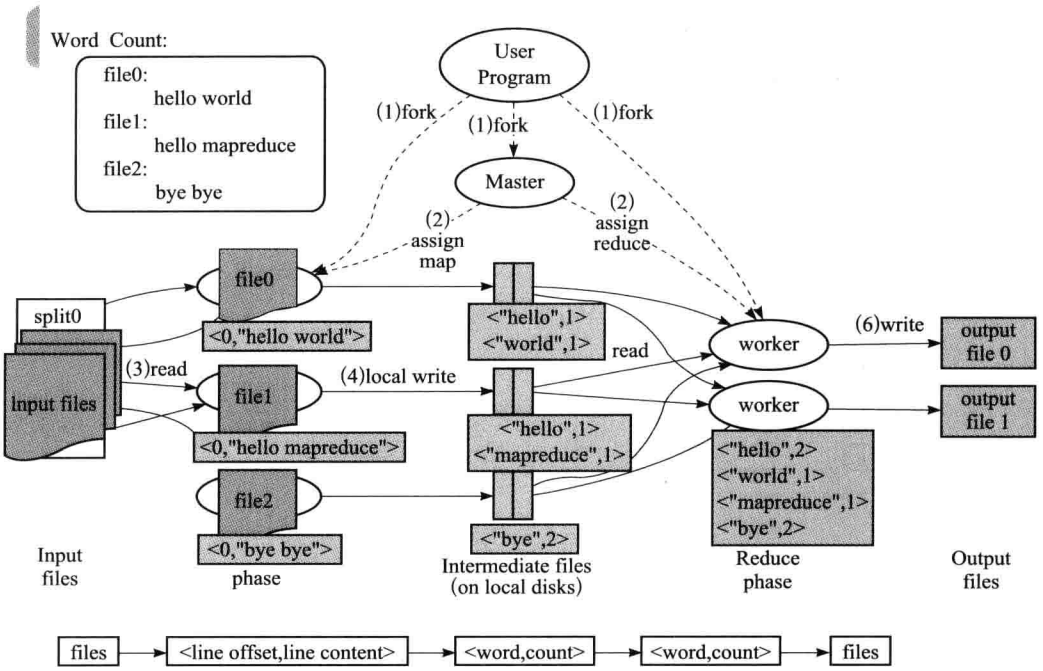


图 1-6 Word Count

图 1-7 中 Hadoop 的 Map/Reduce 实际上是 1.0 版，也许 Hadoop 项目组也意识到该框架的局限性，在目前 Map/Reduce2.0（Yarn）版中实际上已经完全进行了重构，整个设计思想是做成一个资源和任务的调度框架，再也不是 Google 的 Map/Reduce 相关论文阐述的内容了。关于 Yarn 我们在 7.4.2 节也会谈到。

关于 Hadoop 的详细资料可以参考以下信息：

http://hadoop.apache.org/hdfs/docs/current/hdfs_design.html

http://labs.google.com/papers/gfs.html

2. ZooKeeper

ZooKeeper 在 Hadoop 生态体系中是作为协同系统出现的，为什么会独立出一个协同系统呢？我们看看跟分布式协同相关的一些重要概念。

- 分布式协同系统：大型分布式应用通常需要调度器、控制器、协同器等管理任务进程的资源分配和任务调度，为避免大多数应用将协同器嵌入调度控制等实现中，造成系统扩充困难、开发维护成本高，通常将协同器独立出来设计成为通用、可伸缩的协同系统。
- ZooKeeper：Hadoop 生态系统的协同实现，提供协调服务，包括分布式锁、统一命