TURING 图灵程序设计丛书

Apress®

Practical Hive

A Guide to Hadoop's Data Warehouse System

Hive实战

[美] 斯科特·肖 [南非] 安德烈亚斯·弗朗索瓦·弗穆尔恩
[印] 安库尔·古普塔 [美] 戴维·杰鲁姆加德 著

唐富年 译

实践Hadoop数据仓库解决方案
在大数据时代“炼数成金”

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Practical Hive

A Guide to Hadoop's Data Warehouse System

Hive实战

[美] 斯科特·肖 [南非] 安德烈亚斯·弗朗索瓦·弗穆尔恩
[印] 安库尔·古普塔 [美] 戴维·杰鲁姆加德 著

唐富年 译

人民邮电出版社
北京

图书在版编目（CIP）数据

Hive实战 / （美）斯科特·肖等著 ; 唐富年译. --
北京 : 人民邮电出版社, 2018.11
（图灵程序设计丛书）
ISBN 978-7-115-49391-0

Ⅰ. ①H… Ⅱ. ①斯… ②唐… Ⅲ. ①数据库系统－程序设计 Ⅳ. ①TP311.13

中国版本图书馆CIP数据核字(2018)第216737号

内 容 提 要

Hive“出身名门”，是最初由Facebook公司开发的数据仓库工具。它简单且容易上手，是深入学习Hadoop技术的一个很好的切入点。本书由数据库专家和大数据专家共同撰写，具体内容包括：Hive的安装和配置，其核心组件和架构，Hive数据操作语言，如何加载、查询和分析数据，Hive的性能调优以及安全性，等等。本书旨在为读者打牢基础，从而踏上专业的大数据处理之旅。

本书面向数据科学家以及对大数据技术感兴趣的读者。

◆ 著 [美] 斯科特·肖
[南非] 安德烈亚斯·弗朗索瓦·弗穆尔恩
[印] 安库尔·古普塔 [美] 戴维·杰鲁姆加德
译 唐富年
责任编辑 谢婷婷
责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市祥达印刷包装有限公司印刷

◆ 开本：800×1000 1/16
印张：15.5
字数：366千字 2018年11月第1版
印数：1－3 000册 2018年11月河北第1次印刷

著作权合同登记号 图字：01-2018-5948号

定价：69.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

版权声明

我把这本书献给我的家人。他们并不知道我从事何种工作，却容忍我一天到晚“玩”电脑。爱你们！

——斯科特·肖

我将这本书献给我的家人和我睿智的导师们，感谢他们的支持。特别感谢丹尼丝和劳伦斯。

——安德烈亚斯·弗朗索瓦·弗穆尔恩

我要向许多见证我完成这本书的人表达我的感激之情。最重要的是，我要感谢我的妻子贾丝玟和其他家人，尽管这段时间我几乎无暇照顾他们，他们却一直支持和鼓励着我。

——安库尔·古普塔

“只要锲而不舍、不断学习和永恒追求，任何人都能成为伟人。”——乔治·巴顿

——戴维·杰鲁姆加德

前　言

第一次听说 Hive 的时候，我是两个数据仓库项目的顾问。其中一个项目已经开发了 6 个月。我们的团队有 12 名顾问，但是进展甚微。源数据库是关系型的，但是由于某些未知原因，所有的约束（例如主键和外键参照关系）都被关闭了。总而言之，这样的源数据库实际上是非关系型的，而我们的团队则努力将这样的数据迁移到高度结构化的数据仓库中。我们努力解决空值问题并构建约束，还纠结于主数据管理问题和数据质量问题。该项目最终的目标就是建立一个数据仓库，重建早已经有的报表。

第二个项目的规模相对较小，但是涉及层级关系。例如，电视机有品牌名称、存货单位（SKU）、产品代码和数量各异的其他描述性特征。这些特征中有一些是动态的，而另一些则可适用于一个或多个不同的产品或品牌。各个品牌在特征的层次结构上都有所不同。同样，我们努力在一个关系型数据仓库中描述这些业务需求。

第一个项目所面临的困难体现在从一个模式迁移到另一个模式时。这个问题必须在任何人提出任何疑问之前解决，即便这样，这些疑问也必须提前知晓。第二个项目的困难则出现在表达那些无法与既有的数据结构融合的业务规则时。我们最终让客户更改了他们的业务规则，以适应该结构。

当我第一次将文件复制到 HDFS 并且在该文件之上创建 Hive 表时，我为这种解决方案的简单易用以及它对数据分析的深远影响而赞叹不已。从那时起，我见过一些使用 Hive 的数据项目从设计到拥有真正的分析价值仅需几周时间，而使用传统方法则要花费数月。对于数据驱动型公司和那些需要解决关键业务问题的公司而言，Hive 和更庞大的 Hadoop 生态系统确实是规则的改变者。

本书的目的就是带你体验我所经历过的顿悟时刻，旨在为你打牢基础，进而探索和体验 Hive 和 Hadoop 所提供的功能，帮助你踏上技术之旅，学习在未来十年甚至更长一段时间里驱动创新能力的技术。要在技术领域生存，你必须不断重塑自我，因为技术是不断向前发展的。现在，这列火车即将启程，欢迎乘坐！

致　谢

早在加入 Hortonworks 公司之前，我就想写一本关于 Hive 的书。那时候有关 Hive 的书比较少，而且我看过的一些虽然技术讲得很好，但是并不面向普通用户，尤其是来自关系数据库领域的用户。到 Hortonworks 公司工作以后，我感到坐下来写这本书变得容易多了。我的手边有最优秀的资源，而且可以接触到一些我所见过的最聪明的人。我认识了像艾伦·盖茨这样的 Hive 代码提交者，他们会毫不犹豫地回复邮件或者花点时间跟我在会议上交流。我与世界上最棒的解决方案工程师团队建立了友谊并且得到了他们的支持。然而过了近两年半，我还是没有写完这本书。

我没有预料到这个市场的发展速度如此之快，也没有预料到整个团队为客户提供解决方案需要投入大量时间。这确实是一项令人钟爱的工作，但是为了兼顾工作和家庭，我不得不将这本书的写作一再拖延，而且拖了很长一段时间。我想其他任何一家出版社都会放弃我再找别人，但是 Apress 出版社一直在耐心等待（虽然我不能说他们一点都没有退却，而且理应如此），并且坚信总有一天我们会写出一本书来。

写一本关于 Hive 的书，其艰难之处在于：如果你的写作中断了 6 个月，那么你就要写一本新书了。我意识到这并不是一个人能够完成的工作，我需要帮助。安库尔是首先站出来帮助我的人。如果没有他的坚持和奉献，就不会有这本书。也是安库尔使我们与安德烈亚斯取得了联系，我相信安库尔会同意，如果没有安德烈亚斯令人惊叹的写作能力和知识水平，也不会有这本书，至少这本书会更薄一些，你从中获得的信息量会大大减少。最后，感谢戴维，他确定了本书的技术重点，对于去冗存精起到了至关重要的作用，指引我们一路向前。

还有其他很多人在自己有限的时间里尽己所能地提供了帮助。微软 CAT 团队的辛迪·格罗斯在早期曾经参加了本书的撰写，帮助推动这个项目前进。感谢安西尔承担了非常必要的技术审校工作——尤其是对我所撰写的章节。要特别感谢 Hortonworks 公司，它不仅仅支持本书的撰写，而且发自肺腑地为此激动不已。Hortonworks 团队并不仅仅因为这是一本关于 Hive 的书而激动，他们是为我们这个作者团队所取得的成就而激动。我从未被迫在工作和这本书之间做出选择，专心本职工作是我自己的选择。

最后，感谢我的家人。我的孩子们可能根本不需要 Hive，但是我知道，他们认为爸爸能参与撰写一本书是一件非常酷的事。从英语专业的学生到一家开源大数据公司的解决方案工程师，而且能够撰写技术类图书，这是一段很长的成长之旅。环顾四周，我感到非常知足。再次重申，我与业界最聪明的人一起工作，虽然他们的才智我难以望其项背，但是我深知，他们的集体智慧和见解会使我成为一个更加优秀的人。

——斯科特·肖

目　录

第1章 为Hive打好基础：Hadoop

到现在为止，任何有一丝好奇心的技术人员都听说过 Hadoop，这是在办公室闲聊时被反复提及的一个词。对于 Hadoop 的看法不一，从“Hadoop 就是浪费时间”到“它很了不起，将解决我们当前的所有问题”。你可能也听说过你们公司的董事、经理甚至首席信息官让团队开始实现这种新生的大数据事物，并且找到一个要用它来解决的问题。在谈到大数据时，非技术人士的第一反应通常是：“噢，你是说像 NSA[①]那样吗？”毫无疑问，大数据带来了重大责任，但是显然，如果对大数据的使用及其好处缺乏认识，将会滋生不必要的恐惧、不确定和怀疑。

实际上，你在看这本书就说明你对 Hadoop 感兴趣。你可能已经知道 Hadoop 能让你存储和处理大量数据。我们猜测，你也认识到了 Hive 是一款强大的工具，允许你使用 SQL 来实现熟悉的数据访问操作。从书名可知，这本书是关于 Hive 的，它会告诉你 Hive 在访问大型数据存储时是多么重要。牢记这一点有助于理解我们为何撰写本书。我们已经有了像 T-SQL 和 PL/SQL 这样的工具，以及其他能够检索数据的分析工具，为什么还需要 Hive 呢？在现有环境中增加更多需要掌握新技能的工具，难道没有额外的资源成本吗？事实上，我们认为可用的数据是不断变化的，而且变化得很快。这种快速的变化迫使我们扩展自己的工具集，不能局限于过去 30 年所依赖的工具。我们将在后面的章节中看到，我们确实需要改变，但是也需要利用那些早已获得的成就和技能。

Hadoop 与“大数据”这个术语几乎同义。在我们看来，“大数据”正在慢慢地走向其他术语（例如决策支持系统和电子商务）的命运。当人们将“大数据”作为一种解决方案来谈论时，他们通常是从市场营销的视角来看问题的，而不是从一种工具或者能力的视角。我记得与一位高层管理人员会面时，他坚决要求我们不要在讨论中使用“大数据”这个术语。我同意了，因为我觉得这个术语会冲淡谈话的主题，使我们更关注于通用术语而没有触及技术的变革本质。但是话又说回来，数据确实在变大，而我们不得不从某个地方开始讨论这个话题。

我的观点是：Hadoop 最初是一种用于解决特定问题的技术。它在不断演化，而且演化的速度比罐子里果蝇的繁殖速度还快。它已经演变成一种核心技术，正在改变企业看待其数据的方式——如何使用数据，如何深入理解所有数据——以解决特定业务需求并获得竞争优势。用于处理数据的现有模型和方法论正不断受到挑战。随着不断演进并且越来越被认可，Hadoop 从一种

① 美国国家安全局。——译者注

小众的解决方案转变为每个企业都能从中获取价值的解决方案。再从另外的角度想想。现在的日常技术都是基于专门的需求创造出来的，例如军事需求。我们认为理所当然的东西，比如胶带和 GPS，都是首先针对特定军事需求而开发的。为什么会这样？创新至少需要 3 个要素：一种迫在眉睫的需求，一个可以识别的问题，还有金钱。军队是庞大而复杂的组织，拥有人才、金钱、资源以及发明这些日常用品的需求。显然，军队为自己使用而发明的产品和零售商店里的产品往往不太一样。后者经过改良、推广和精心打磨，以供日常使用。随着我们深入了解 Hadoop，要注意与此相同的过程：那些独特且紧紧聚焦于某一需求的发明将不断演进，以满足更广泛的企业需求。

如果要将 Hadoop 和大数据比作什么，可以将它们视为一段旅程。很少有公司在创立之初就需要含有 1000 个节点的集群，也不会轻易决定在这样的平台上运行关键业务。企业会经历一段可预测的旅程，时间从几个月到几年不等。希望本书能够帮助你开始自己的旅程，并且有助于阐明整个旅程的具体步骤。第 1 章介绍了 Hadoop 世界为什么与众不同，以及它的由来。本章为稍后的讨论奠定了基础。在学习具体的技术之前，你将了解 Hadoop 平台，也将明白开源模型为何如此不同且具有颠覆性。

1.1　一只小象出生了

2003 年，Google 公司发表了一篇并不引人注目的论文，题目为“The Google File System”。在硅谷之外并没有多少人关注这篇论文的发表和它试图传达的信息。这篇论文所包含的信息可直接用于像 Google 这样的公司，其主要业务聚焦于对互联网进行索引，而对于大多数公司来说，这并不是一个常见的用例。这篇论文描述了一种独特的存储框架，是为解决 Google 公司今后的技术需求而设计的。本着 TL&DR[①]的精神，这里给出其最突出的观点：

- 故障是常态
- 文件很大
- 文件通过追加来更改，而不是通过更新来更改
- 紧耦合的应用程序和文件系统 API

如果你打算成就一家规模达数十亿美元的互联网搜索公司，上述假设大多都有意义。你将主要关心大文件的处理，以及以低延迟的代价连续进行长时间的读写操作。你也会想把自己的海量存储需求分散到（廉价的）商用硬件上，这样就不需要建立代价高昂的资源竖塔。数据摄入需要格外关注，在写入时对这些数据结构化（模式化）只会延迟处理过程。你还需要安排一个由世界顶级开发人员组成的团队，构建可伸缩、分布式且高可用的解决方案。

Yahoo 注意到了这篇论文。他们在互联网搜索领域面临着类似的伸缩性问题，并且使用了由 Doug Cutting 和 Mike Cafarella 创建的一个名为 Nutch 的应用程序。Google 的那篇论文为 Doug 和 Mike 提供了一个框架，可解决 Nutch 架构中很多固有的问题，其中最重要的是可伸缩性和可靠性。而接下来需要进行的就是对基于 Google 的论文设计的解决方案重新进行工程实现。

① 互联网用语，意思是“篇幅太长，不再阅读”。——译者注

注意 请记住，最初的GFS（Google文件系统）和发展成为Hadoop的这部分技术并不相同。GFS是一个框架，而Hadoop则是将该框架付诸实施。GFS仍然是Google专有的，也就是说，它并不是开源的。

当我们提起Hadoop时，经常会想到Google公司的那篇论文中有关存储器的那一部分概述。实际上，这个等式的另一半（更为重要）是Google公司在2004年发表的题为“MapReduce: Simplified Data Processing on Large Clusters”的论文。这篇论文采用一种被称作“易并行”（embarrassingly parallel）的方法，将大型分布式集群上的数据存储与该数据的处理相结合。

注意 对MapReduce的讨论将贯穿全书。MapReduce在交互式SQL查询处理中既是一个意义重大的角色，也是一个日益走向衰落的角色。

Doug Cutting和Yahoo的其他一些人看到了GFS和MapReduce对Yahoo自身用例的价值，因此从Nutch剥离出一个单独的项目。Doug用他儿子的玩具小象的名字Hadoop来命名这个项目。尽管这个名字很可爱，但是项目本身还是很严肃的，而且Yahoo打算推广它，用它来满足搜索引擎以及广告业务方面的需求。

注意 Hadoop社区流传着一个笑话：如果你将产品命名权交给工程部门而不是市场营销部门，那么就会得到类似于Hadoop、Pig、Hive、Storm、Zookeeper和Kafka这样的名字。我个人喜欢那些看起来愚蠢但是能够解决复杂实际问题的应用程序。至于那只Hadoop小象的命运嘛，Doug现在仍然随身带着它参加各种演讲活动。

Hadoop在Yahoo公司内部的规模增长并不典型，但是对于当前很多实现的模式来说，它是一个典范。在Yahoo的案例中，最初的开发只能扩展到少数几个节点，但是几年之后，就可以扩展到数百个节点了。随着集群的增长和扩展，系统摄入了越来越多的企业数据，组织内部的壁垒开始被打破，用户也开始从数据中发现更多的价值。随着所有职能领域的壁垒都被打破，更多的数据被迁移到集群中。一个有着美好目标的事物很快成了整个组织的核心和灵魂，更恰当地说，成了整个组织的存储和分析引擎。正如一位作者所述：

> 2011年，当Yahoo将Hortonworks分拆为一家独立的、专注于Hadoop的软件公司时，Yahoo的Hadoop基础设施已经拥有了42 000个节点和数百PB的存储空间。

1.2 Hadoop的结构

Hadoop通常包含两个部分：存储和处理。存储部分就是Hadoop分布式文件系统（HDFS），处理就是指MapReduce（MR）。

注意　在本书完成之际，这一环境正在改变。MapReduce 现在只是在 HDFS 之上处理 Hive 的一种方式。MapReduce 是一种传统的面向批量任务的处理框架。像 Tez 这样的新处理引擎越来越倾向于近实时的查询访问。随着 YARN 的出现，HDFS 正日益成为一个多租户环境，允许很多数据访问模式，例如批量访问、实时访问和交互访问。

当我们考虑到常规文件系统的时候，会想到像 Windows 或 Linux 这样的操作系统。这些操作系统都安装在运行重要应用程序的单台计算机上。如果我们通过网络将 50 台计算机连接到一起，会发生什么呢？我们仍然有 50 个不同的操作系统，如果我们想要运行单个应用程序来使用它们所有的计算能力和资源，这种方式对我们没有多少帮助。

例如，我正在微软的 Word 软件中输入这些内容，该软件只能在单操作系统和单台计算机上安装和运行。如果想提高 Word 应用程序的运算性能，那么我别无选择，只能给我的计算机增加 CPU 和 RAM。问题在于，我所能增加的 CPU 和 RAM 的数量是有限制的。我很快就会达到单台设备的物理极限。

HDFS 则做了一些独特的事情。你可以选取 50 台计算机并且在每一台上都安装一个操作系统（如 Linux）。在用网络将它们连接起来之后，你在所有计算机上都安装 HDFS，并且将其中一台计算机声明为主节点，将其他所有计算机都声明为工作节点。这样你就构建了 HDFS 集群。现在，当你将文件复制到某个文件夹中时，HDFS 会自动将文件的各个部分存放在集群的多个节点上。HDFS 成为 Linux 文件系统之上的一个虚拟文件系统，它抽象了你在集群的多个节点上存储数据的事实。图 1-1 从整体上说明了 HDFS 如何从客户端抽象出多个系统。

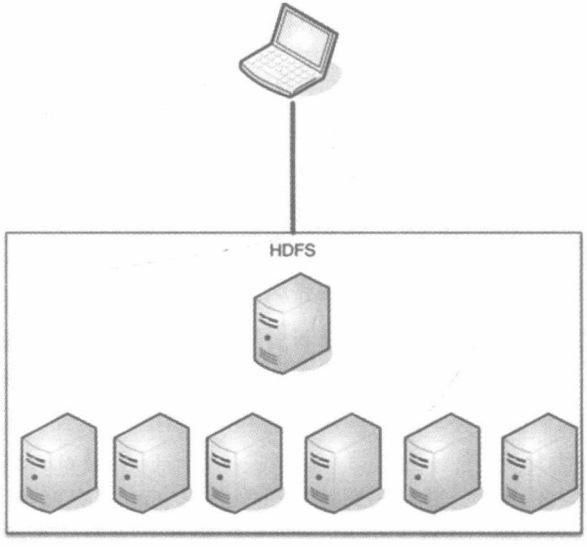

图 1-1　HDFS 的简单视图

图 1-1 非常简单，展示了最基本的视图（将在 1.3.2 节详细说明）。最重要的一点是，现在的增长能力是水平的而不是垂直的。与为单台设备添加 CPU 或 RAM 不同，你只需要增加一台设备，也就是一个节点。线性可伸缩性允许你基于自己扩大的资源需求快速扩展自己的能力。敏锐的读者很快就会争辩说，通过虚拟化也可以获得类似的优势。那么让我们用虚拟化的视角来看待同一问题。图 1-2 展示了这样一个虚拟化架构。

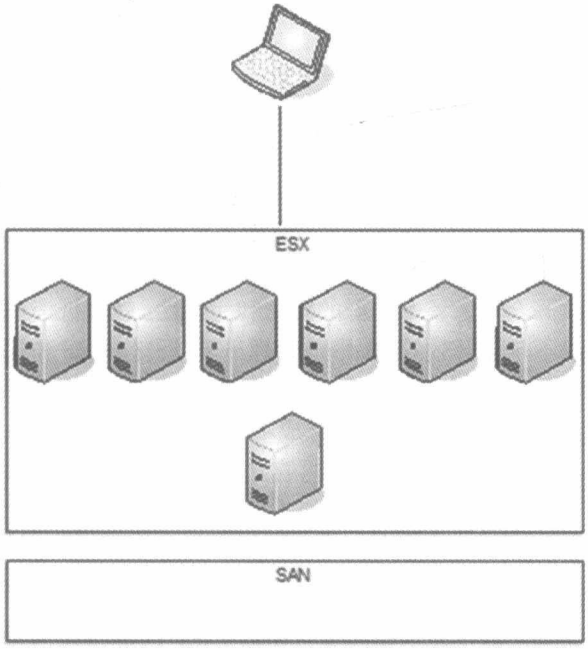

图 1-2 虚拟化架构

管理员在服务器（大多数情况下是一个服务器集群）上安装虚拟管理软件。该软件集中了 CPU 和内存这样的资源，这样看起来就像有一台有着大量资源的服务器。在虚拟操作系统层上有多个客户，可以将可用的资源池划分给每个客户使用。这样做的好处包括：IO 资源的最大化、资源的动态供应，以及物理集群层的高可用性。存在的问题则包括：对 SAN 存储器的依赖、不能进行水平扩展、垂直扩展存在限制，以及对多操作系统安装存在依赖等。目前大多数数据中心都采用这种模式，而且在过去 10 多年里，虚拟化一直是 IT 界的主流趋势。

注意 图 1-2 中用到了术语 ESX。我们当然并不特指 VMWare。我们给出虚拟化的架构只是为了说明 Hadoop 如何从根本上改变了数据中心的模式，以满足独特的现代数据需求。对于很多用例来说，私有云虚拟化仍然是一种可行的技术，而且应该与其他架构（如设备或公有云）放在一起考虑。

Hadoop 的其他优势还包括降低能源消耗以及减少物理服务器的占用和动态供应。Hadoop 需要与保持了长达 10 多年的发展趋势的虚拟化架构相抗衡，这是一项艰难的任务。企业这些年来一直都在远离物理架构，在减少数据中心的物理服务器数量上取得了很大进展。如果在扩展文件系统之时，Hadoop 所能提供的只是添加另一个物理节点，那么我们也就没有必要写这本书了，而 Hadoop 也将重蹈 Pets.com 的覆辙[①]。Hadoop 的架构还有更多特性，在商业上具有变革性意义，值得在物理架构上投入。

1.3　数据冗余

大规模的数据也必须是高可用的。Hadoop 以高效且廉价的方式存储数据。Hadoop 软件架构内置了一些机制，允许我们采用廉价的硬件。正如 Google 公司在其论文中所说的，最初的设计假定节点会发生故障。当集群水平扩展到数百、数千甚至数万个节点时，我们别无选择，只能假定在任意给定时间集群中都会有少量服务器出现故障。

如果有些服务器的故障会危害整个集群的安全和完整性，就会抵消 HDFS 所提供的任何其他好处，更不用说由于睡眠不足而导致的 Hadoop 管理员流失。Google 和 Yahoo 的工程师面临着艰巨的任务，既要降低成本又要增加正常运行时间。目前已有的高可用性解决方案在满足他们的需求时，不可避免地会使公司陷入硬件成本、软件成本和维护成本的深渊中。为了满足他们的需求，有些事情必须改变。Hadoop 成了解决这一问题的方案，但是首先我们需要了解为什么现有工具无法成为解决方案。

1.3.1　传统的高可用性

当我们想到冗余的时候，通常都会想到高可用性（HA）。高可用性是一种架构，它描述了你访问环境的频次。我们通常用“9”的形式来度量高可用性。我们可以说自己的运行时间是 99.999，也就是五个九。表 1-1 显示了基于可用率的实际停机时间。

表 1-1　可用率概览

可用率	每年停机时间	每月停机时间	每周停机时间
90%（一个九）	36.5 天	72 小时	16.8 小时
95%	18.25 天	36 小时	8.4 小时
97%	10.96 天	21.6 小时	5.04 小时
98%	7.3 天	14.4 小时	3.36 小时
99%（两个九）	3.65 天	7.2 小时	1.68 小时
99.5%	1.83 天	3.6 小时	50.4 分
99.8%	17.52 小时	86.23 分	20.16 分
99.9%（三个九）	8.76 小时	43.8 分	10.1 分

① Pets.com 是一家短命的互联网公司。——译者注

（续）

可用率	每年停机时间	每月停机时间	每周停机时间
99.95%	4.38 小时	21.56 分	5.04 分
99.99%（四个九）	52.56 分	4.32 分	1.01 分
99.995%	26.28 分	2.16 分	30.24 秒
99.999%（五个九）	5.26 分	25.9 秒	6.05 秒
99.9999%（六个九）	31.5 秒	2.59 秒	0.605 秒
99.99999%（七个九）	3.15 秒	0.259 秒	0.0605 秒

成本通常与正常运行时间成比例。正常运行时间越多意味着成本越高。尽管有少量解决方案也依赖于软件，但是大多数高可用性解决方案都聚焦于硬件。大多数解决方案的理念都是采用一组被动系统，以备在主系统出现故障时使用。大多数集群基础设施都采用这种模式。你可能有一个主节点和任意数量的副节点，其中含有复制的应用程序二进制文件以及用于集群的特定软件。一旦主节点发生故障，副节点就会马上接管。

注意 你可以选择建立一个双活式集群，这其中的两个系统都在用。从资源的视角来看，你的成本仍然很高，因为当出现故障之后，你可能需要解决两个系统的应用程序在同一台服务器上运行的问题。

快速故障恢复可使停机时间最小化，而且如果正在运行的应用程序是集群感知型的，可以解决会话终止的情形，最终用户根本就不会意识到系统已经出现了故障。虚拟化就采用了这种模型。物理主机一般是由 3 个或者更多系统构成的集群，其中一个系统是被动式的，它在活动系统出现故障时负责接管控制。虚拟客户可以跨系统迁移，而且客户端甚至并未意识到操作系统已经迁移至其他服务器。该模型也有助于维护工作，例如应用更新、打补丁或者更换硬件。管理员在副系统上进行维护，然后将副系统切换成主系统，再对原系统进行维护。私有云采用了类似的框架，在大多数情况下，在集群中都有一个空闲服务器，主要用于替换出现故障的集群节点。图 1-3 展示了典型的集群配置。

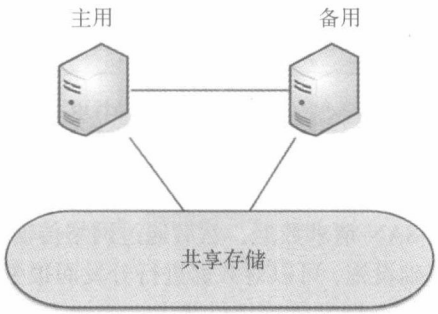

图 1-3 带有共享存储的两节点集群配置