TURING 图灵程序设计丛书

Cassandra

权威指南

Cassandra: The Definitive Guide

[美] *Eben Hewitt* 著
Jonathan Ellis 序
王旭 译

O'REILLY®

人民邮电出版社
POSTS & TELECOM PRESS


Cassandra
O'REILLY

TURING 图灵程序设计丛书

Cassandra权威指南

Cassandra: The Definitive Guide

[美] Eben Hewitt 著
Jonathan Ellis 序
王 旭 译

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo
O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北 京

图书在版编目（CIP）数据

Cassandra权威指南 /（美）休伊特（Hewitt,E.）著；
王旭译. -- 北京 ：人民邮电出版社，2011.8
（图灵程序设计丛书）
书名原文：Cassandra : The Definitive Guide
ISBN 978-7-115-25854-0

Ⅰ. ①C… Ⅱ. ①休… ②王… Ⅲ. ①关系数据库－数据库管理系统 Ⅳ. ①TP311.138

中国版本图书馆CIP数据核字(2011)第129426号

内 容 提 要

本书是一本广受好评的Cassandra图书。与传统的关系型数据库不同，Cassandra是一种开源的分布式存储系统。书中介绍了它无中心架构、高可用、无缝扩展等引人注目的特点，讲述了如何安装、配置Cassandra及如何在其上运行实例，还介绍了对它的监控、维护和性能调优手段，同时还涉及了Cassandra相关的集成工具Hadoop及其类似的其他NoSQL数据库。

本书适合数据库开发人员与网站开发者阅读。

图灵程序设计丛书

Cassandra 权威指南

◆ 著 [美] Eben Hewitt
序 [美] Jonathan Ellis
译 王 旭
责任编辑 傅志红
执行编辑 李 盼

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：19
字数：394千字 2011年8月第1版
印数：1－3 500册 2011年8月北京第1次印刷
著作权合同登记号 图字：01-2011-0812号
ISBN 978-7-115-25854-0

定价：59.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223
反盗版热线：(010)67171154

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”，创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会聚集了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O’Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O’Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

第 1 章

Cassandra 概况

如果一个概念一开始不是荒诞不羁的话，那它也没什么前途。

——阿尔伯特·爱因斯坦

欢迎阅读《Cassandra 权威指南》。本书的目标是帮助开发者和数据库管理员们理解这种重要的新型数据库，探索它与传统的关系型数据库系统有何异同，并帮助你在自己的系统中使用 Cassandra。

1.1 关系型数据库有什么问题

如果当初我问人们到底想要什么的话，他们会说想要快些的马。

——亨利·福特

请你考虑一种数据模型，它由一个拥有数千名雇员的大公司里的一个小团队发明。这个模型可以通过 TCP/IP 访问，并且可以使用包括 Java 和 Web Service 在内的多种语言访问。在被广泛采纳之前，这个模型起初若非最顶尖的计算机科学家都很难理解，最终因为用的人越来越多，才被大家认识。要使用这种模型构建出数据库，就必须学习许多新术语，换一种方式考虑数据存储。随着相关的产品层出不穷地出现，很多公司和政府部门开始广泛地使用它，因为它非常快——可以在一秒钟内执行上千次操作。这种模型产生了让人难以置信的收益。

然后，又一种新的模型出现了。

这个新模型是一种可怕的异端，主要有两点原因。首先，它最受非议的地方就在于新模型与旧模型完全不同，它们简直是格格不入。这很可怕，因为全新且完全不同的东西通常难以理解。随之而来的争论更进一步巩固了人们的看法，而这些看法主要来自人们已有的工作环境和习得的技术。其次，或许是更重要的原因，新模型所面临的阻碍更多来自商业考虑，它威胁到了在旧模型上的大量投资，这些投资正在带来大量收入。在这个时候改变似乎是可笑的，也是不可能的。

当然，我说的是 1966 年由 IBM 发明的信息管理系统（IMS）分层数据库。

IMS 是为“土星五号”登月火箭而设计的。它的架构师是 Vern Watts，他的整个职业生涯都和 IMS 联系在一起。很多读者都知道 IBM 的 DB2 数据库。而 IBM 广为人知的 DB2 数据库的名字就沿袭自它的前身 DB1，DB1 是围绕着 IMS 的数据模型构建的。IMS 于 1968 年发布，之后在用户信息控制系统（CICS）和其他应用中取得成功。至今 IMS 仍被很多应用使用着。

但是，在 IMS 发明之后的几年里就出现了一个崭新的模型，这个破坏性的、带来威胁的模型就是关系型数据库。

同样来自 IBM 的 Edgar F. Codd 博士在他 1970 年发表的论文《一种用于大规模共享数据存储系统的关系型数据模型》中，阐述了他在 IBM 圣何塞实验室的工作中提出的关系数据模型理论。这篇目前仍然可以在 http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf 访问的论文，成为了后来的关系型数据管理系统的奠基性作品。

Codd 的工作与 IMS 的层次化结构完全对立。IMS 的用户要理解并使用关系型数据库，就需要学习很多听起来十分怪异的新名词。不过，关系型模型与它的前身相比更具优势，部分因为巨人几乎总是站在其他巨人的肩膀上。

关系型数据库这个概念和它的应用已经演化了 40 多年了，毫无疑问，它是软件应用历史上最成功的一个。它既可以在个人小公司里通过微软 Access 数据库来使用，也可以在大型跨国公司的上百台经过调优的服务器上使用，构成存储数 TB 数据的数据仓库。关系数据模型存储了账单、用户记录、产品目录、账户明细、用户鉴权信息等，可以说关系型数据库里差不多存储了整个世界。毫无疑问，无论以现代的技术还是商业视角看，关系型数据库都扮演着关键角色，而且，多年后它也会以各种形式一直伴随我们存在，IMS 也是同样。关系模型虽然是 IMS 的一种替代模型，两者都各得其所。

所以，要问关系型数据库有什么问题，一言以蔽之，就是“没有问题”。

不过，实际上我还想请你考虑一个长一点的答案。这个答案需要从长计议，每个偶

然诞生的概念都在点滴改变着世界，孕育着某种革命。而这一次次的革命不过是历史的重演罢了。从 IMS、RDBMS 到 NoSQL，一如从马到汽车再到飞机，每一个都建立在前一个的基础之上，解决前一个存在的某些问题，每个都有所长，亦有所短。他们彼此共存，直到如今。

那么就来看看为什么现在我们要考虑关系型数据库的替代产品，正如四十年前 Codd 自己面对信息管理系统（IMS）的思考一样——或许 IMS 不是唯一适合用于组织信息、处理数据的方法，可能正是因为某些问题使得他必须要考虑一种 IMS 的替代品。

当基于关系型数据库的应用取得成功，访问量大幅增加的时候，我们就会遇到可扩展性问题。所有具有最基本功能的关系型数据库都会支持 join 操作，不过 join 可能会很慢。由于数据库通常依靠事务来保证一致性，而事务需要锁住数据库的一部分，使之不能被其他用户访问。因为锁本身意味着竞争同一数据的用户会被放入队列，等待获得读写权限，这在高负载的情况下可能会成为系统的死穴。

通常，我们会用下面的一条或几条方法来解决这些问题，很多时候是下面这个顺序。

- 提升硬件能力来解决问题，如增加内存、用更快的处理器以及升级硬盘。这种方法称为*垂直扩展*，可以解一时之忧。
- 当问题再度出现时，解决方法很类似：既然一台机器已经不堪重负了，我们就增加新的计算机，构成数据库集群。不过，这样你就会在正常使用及出故障时遇到数据复制与一致性问题了。这些问题之前从未出现过。
- 现在我们需要更新数据库管理系统的配置。可能是要优化数据库用来写底层文件系统的通道。我们关掉了文件系统的日志，当然，这通常是不推荐的（或者说要具体情况具体分析）。
- 在数据库系统上投入了足够多的精力之后，我们转过来审视自己的应用。我们开始优化索引、优化查询。不过，当我们的应用达到这个规模的时候，恐怕不太会完全没做过索引和查询优化，可能已经优化过不少了。那么，只好重新审视所有数据库访问代码，想发现零星的可以调优的机会，这是一件相当头疼的事。优化内容可能包括减少或改写 join 操作，除去比较消耗资源的特征（如在存储过程中的 XML 处理）等。当然，我们做那些 XML 处理本身肯定是有原因的，如果我们不得不做的话，那就得把它挪到应用层去，希望那里能解决问题，并祈祷我们这么干不会同时把什么其他东西弄坏。
- 我们增加了一个缓存层。对于大系统可能会引入分布式缓存，如 memcached、EHCache、Oracle Coherence 或其他类似产品。现在，我们又需要面临更新缓存和更新数据库的一致性问题了，对于集群来说，问题更加严重。

- 现在我们把注意力重新转向数据库，由于应用已经在那里了，并且我们很了解主要的查询路径，现在我们复制那些访问频率较高的数据，让它们更接近于查询想要得到的形式。这个过程称为反范式化（denormalization），它与关系模型的关键特征里的五种形式是对立的，也违反了 Codd 对关系型数据模型的 12 条建议。我们只能安慰自己说我们是生活在现实世界之中，而非理论世界中，并保证我们所做的都是为了让应用能够在可接受的响应时间下完成工作，即使这已经不是“纯粹”的关系模型了。

我相信这一幕对你肯定非常熟悉。在互联网的规模下，工程师们已经开始考虑这个情况是不是和当年亨利·福特的境遇有些相似，你真正需要的已不仅仅是你本来想的一匹快马了。他们已经做了一些令人称道而有趣的工作。

首先我们必须认识到，关系模型只是一种数据模型而已。它是一种看待世界的有效的方法，对某些问题非常有效。但它并没打算也没被证明可以一统天下，不留任何余地终结所有其他表示数据的方法。如果我们回顾一下历史就会发现，Codd 博士的模型也曾是个破坏性的发明。它是全新的，带来了很多新词汇和像“元组”这样的术语——老词汇但有新意义。关系模型在当时引发了质疑，甚至受到了强烈的批评。即使是 Codd 博士工作的 IBM 公司也是反对者之一，因为他们拥有一系列围绕 IMS 的富有盈利能力的产品，不需要一个闯入者来分蛋糕。

不过如今，关系模型已经无可辩驳地坐上了数据世界中的头把交椅。SQL 获得了广泛的支持和很好的理解，大学的入门课程就会讲授 SQL 与关系数据库。甚至 4.95 美元 1 月就可以租到有免费数据库的互联网主机。我们最终使用什么数据库通常由组织的架构标准而定。即使没有这样的标准，学习一下组织结构已经有了什么数据库平台仍是明智的。我们的开发和架构方面的同事都有相当难得的知识积累。

仅仅是出于渗透或是惯性，我们多年来都已经习惯了关系型数据库是一个四海一家的解决之道。

因此，也许真正的问题不是“关系数据库有什么问题”，而是“你有什么问题要解决”。

也就是说，要确保你的解决方案和你的问题相匹配。关系型数据库确实在解决很多问题时非常有效。

如果你并未面对大规模、弹性扩展的问题，那么对于 Cassandra 之类的复杂系统的取舍完全没有考虑的必要。据我所知，没有任何一位 Cassandra 的支持者会要求别人丢掉他们的所有关系型数据库的知识、放弃他们多年来对于这类系统的难得的经验，或是破坏他们的员工花费数月时间精心构建起来的系统。