

<<NoSQL数据库入门>>

图书基本信息

书名：<<NoSQL数据库入门>>

13位ISBN编号：9787115279507

10位ISBN编号：7115279500

出版时间：2012-5

出版时间：人民邮电出版

作者：[日]佐佐木达也

页数：199

字数：308000

译者：罗勇

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<NoSQL数据库入门>>

前言

前言 最近，到处都能听到“NoSQL”，这个词到底是什么意思呢？“NoSQL”到底能给我们带来什么好处呢？

现在，提起数据存储，一般都是针对关系型数据库来说的。

但是，关系型数据库并不是万能的，它对于某些处理依然是很吃力的。

本书所讲述的NoSQL数据库就是为了弥补关系型数据库的不足应运而生的。

在适当的情况下使用NoSQL数据库，可以为关系型数据库需要耗费大量时间才能完成的处理，提供高速、合理的解决方案。

预想的适用情况NoSQL数据库可用于：取代关系型数据库的弱势处理（比如大量数据的写入处理等）作为关系型数据库之外的另一种选择NoSQL数据库虽然可以替关系型数据库分担一些难题（如大量数据的写入等），但这其中的操作具有相当的难度。

因此本书仅仅将NoSQL定位为“关系型数据库之外的另一种选择”。

读者对象 本书面向的读者群为有1年关系型数据库开发经验的软件工程师和程序员。

因为他们对数据量的增大给关系型数据库的检索和更新处理带来的剧烈性能恶化有切身感受，也能更深刻地理解NoSQL数据库的优势。

本书将为读者介绍NoSQL数据库以及使用NoSQL数据库所带来的便利。

本书内容 本书共由5章内容组成，下面对各章内容做一个简单的介绍。

第1章首先介绍什么是NoSQL以及这个词的来源。

之后，会介绍关系型数据库的发展历史、关系型数据库的优势和不足，以及NoSQL数据库诞生的背景，以便使读者了解NoSQL数据库的发展历史。

这一章亦会对NoSQL数据库的种类和特征做简单讲解。

同时阐述“怎么样才能更好地区别使用关系型数据库和NoSQL数据库”，为NoSQL数据库的引入奠定基础。

第2章将会讲述memcached、Tokyo Tyrant、Redis和MongoDB这4种NoSQL数据库，介绍这些NoSQL数据库各自的使用背景、特征和用例，以及它们的实际应用。

通过本章的介绍，读者能够了解NoSQL数据库的基本使用方法。

第3章对上述4种NoSQL数据库的应用实例以及实现代码进行具体的介绍。

这一章将使大家对这4种NoSQL数据库能解决的具体、实际的问题有所了解，让NoSQL数据库成为解决问题的选择之一。

虽然本章涉及的各个实例都可以用关系型数据库来实现，但是使用NoSQL数据库能够获得更快的响应，同时简单的操作也给使用者带来更大的便利。

第4章对上述4种NoSQL数据库的性能进行比较。

本章不仅会进行与基本CRUD处理（创建、检索、更新、删除）相关的性能比较，而且还会着眼于一些具体实例，例如像Tokyo Tyrant的addint方法和incr方法的性能比较，Redis的list类型的插入和删除的性能比较，以及MySQL的join和MongoDB的embed的性能比较等。

第5章对NoSQL数据库在实际应用中的问题点，以及HandlerSocket解决方案进行介绍。

它虽然和NoSQL数据库略有一点不同，却是个非常有意思的解决方案，有利于读者们开阔视野。

<<NoSQL数据库入门>>

内容概要

《NoSQL数据库入门》详细地介绍了NoSQL数据库(非关系型数据库)的种类、用途以及使用方法,并对memcached、TokyoTyrant、Redis、MongoDB这4种代表性的NoSQL数据库的特征、适用范围、实现代码进行了深入探讨,并比较了它们的性能。

《NoSQL数据库入门》适合有关系型数据库开发经验的软件工程师和程序员阅读。

<<NoSQL数据库入门>>

作者简介

James Turnbull 开源拥趸，Linux Australia前任主席，经常在OSCON、Open Source Bridge、DevOpsDays等大会上发言。

现任职于Puppet Labs。

目前已有5本著作，均涉及开源软件。

James

是澳大利亚人，目前居住在美国奥勒冈州的波特兰。

他的兴趣爱好十分广泛，包括烹饪、品酒、

政治理论、新闻摄影以及哲学，最近还加入了波特兰Timbers协会足球队。

Jeffrey McCune Puppet社区成员，开源软件的支持者，经常在苹果的世界开发者大会、Macworld、Open Source Bridge、Velocity等大会上发言。

现任职于Puppet Labs，致力于编写代码和帮助客户改进他们的Puppet部署。

Jeff目前居住在奥勒冈州的波特兰，喜爱钻研微控制器、动画、摄影、音乐，爱好徒步旅行以及沙滩长途步行。

<<NoSQL数据库入门>>

书籍目录

- 第1章 NoSQL数据库的基础知识 1
 - 1.1 关系型数据库和NoSQL数据库 2
 - 1.1.1 什么是NoSQL 2
 - 1.1.2 关系型数据库简史 2
 - 1.1.3 数据库的分类 3
 - 1.1.4 关系型数据库的优势 5
 - 1.1.5 关系型数据库的不足 5
 - 1.1.6 NoSQL数据库 9
 - 1.2 NoSQL数据库是什么 12
 - 1.2.1 键值存储 13
 - 1.2.2 面向文档的数据库 14
 - 1.2.3 面向列的数据库 14
 - 1.3 如何导入NoSQL数据库 16
 - 1.3.1 始终只是其中一种选择 16
 - 1.3.2 在何种程度上信赖它？
18
- 第2章 数据库的种类和特征 19
 - 2.1 memcached(临时性键值存储) 20
 - 2.1.1 什么是memcached 20
 - 2.1.2 为什么要使用memcached 20
 - 2.1.3 特征和用例 21
 - 2.1.4 安装步骤 27
 - 2.1.5 动作确认 29
 - 2.1.6 各种开发语言需要用到的程序库 36
 - 2.1.7 相关工具 37
 - 2.2 Tokyo Tyrant(永久性键值存储) 44
 - 2.2.1 什么是Tokyo Tyrant 44
 - 2.2.2 为什么要使用Tokyo Tyrant 44
 - 2.2.3 特征和用例 44
 - 2.2.4 安装步骤 48
 - 2.2.5 动作确认 50
 - 2.2.6 各种开发语言需要用到的程序库 58
 - 2.2.7 相关工具 58
 - 2.3 Redis(临时性/持久性键值存储) 61
 - 2.3.1 什么是Redis 61
 - 2.3.2 为什么要使用Redis 61
 - 2.3.3 特征和用例 67
 - 2.3.4 安装步骤 71
 - 2.3.5 动作确认 72
 - 2.3.6 各种开发语言需要用到的程序库 81
 - 2.4 MongoDB(面向文档的数据库) 82
 - 2.4.1 什么是MongoDB 82
 - 2.4.2 为什么要使用MongoDB 82
 - 2.4.3 特征和用例 84
 - 2.4.4 安装步骤 87

<<NoSQL数据库入门>>

- 2.4.5 动作确认 88
- 2.4.6 各种开发语言需要用到的程序库 100
- 2.4.7 相关工具 100
- 第3章 试用NoSQL数据库 103
 - 3.1 memcached的具体使用实例 104
 - 3.1.1 例 关系型数据库的缓存 104
 - 3.1.2 例 音乐视听排行网站 112
 - 3.1.3 例 外部API的缓存 119
 - 3.2 Tokyo Tyrant的具体使用实例 120
 - 3.2.1 例 在每个页面显示访问量 120
 - 3.3 Redis的具体应用实例 130
 - 3.3.1 例 时间线(Time Line)形式的Web应用 130
 - 3.3.2 例 查询历史记录 144
 - 3.4 MongoDB的具体使用实例 151
 - 3.4.1 例 问卷调查数据的保存 151
 - 3.4.2 例 解析数据的存储 165
- 第4章 性能验证 167
 - 4.1 基本的插入和查询处理的性能 168
 - 4.1.1 假定案例 168
 - 4.1.2 准备工作 171
 - 4.1.3 插入处理的性能 172
 - 4.1.4 查询的性能 172
 - 4.2 不同实例的性能比较 175
 - 4.2.1 Tokyo Tyrant的addint方法和incr方法 175
 - 4.2.2 对Redis的列表类型的数据进行添加和删除 177
 - 4.2.3 MySQL的JOIN和MongoDB的embed 178
- 第5章 NoSQL化的关系型数据库 183
 - 5.1 关于NoSQL数据库 184
 - 5.1.1 各种NoSQL数据库的特征 184
 - 5.1.2 运行时的开销以及经验不足的问题 185
 - 5.1.3 将MySQL数据库NoSQL化的方法 185
 - 5.2 尝试使用HandlerSocket 187
 - 5.2.1 特征 187
 - 5.2.2 为MySQL安装HandlerSocket 188
 - 5.2.3 动作确认 191
 - 5.2.4 HandlerSocket的性能 197

<<NoSQL数据库入门>>

章节摘录

版权页：插图：让我来看另外一个例子，尝试把MongoDB作为解析数据的存储器，也就是把日志数据进行解析，然后使用MongoDB来保存解析结果。

为了检查服务的使用情况，提供更好的服务，可以通过解析日志数据，获取PV 和UU ，以及其他各类数据，这些数据应该如何保存呢？

使用关系型数据库 当然使用关系型数据库来管理这些数据也不是不可以，但是，对于分析日志数据来说，我们并不清楚哪些字段是必须的。

可能有些数据刚开始并不是必须的，但在进行某些分析时就变成必须的了。

因此，可以考虑在创建解析用表时把所有解析数据都保存起来，在必要的时候添加一列，或者把每条解析数据保存到单独的表中。

如果把解析数据全都保存在同一个表中，每添加一列需要花费很多工夫，而且我们还要考虑到，只有在某些特定分析中才用得到的字段会越来越多。

如果把每条解析数据都保存到单独的表中，表的数量就会在短时间内骤然增加，也很让人头疼。

使用MongoDB 这种情况下尝试MongoDB不失为明智之选。

解析日志数据的时候会得到各种各样的数据，若使用MongoDB便可以不用考虑表结构的问题，直接把所有数据都保存下来即可。

需要添加新数据的时候，也无需变更表结构，直接就能进行添加，这就是使用MongoDB的好处。

对于那些以后可能会用到的临时数据，保存起来也很简单。

<<NoSQL数据库入门>>

编辑推荐

- 1、了解当今最炙手可热的NoSQL新型数据库技术；
- 2、介绍memcached、Tokyo Tyrant、Redis、MongoDB这四种代表性的NoSQL数据库

<<NoSQL数据库入门>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>