

由Elasticsearch内核开发工程师编写，从源码和设计角度分析Elasticsearch的内部原理，为合理、高效地使用Elasticsearch提供理论指导，并为大规模应用和维护过程中的常见问题提供具体的优化措施和故障诊断方法

Broadview[®]
www.broadview.com.cn



Elasticsearch

源码解析与优化实战

张超 / 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Elasticsearch

源码解析与优化实战

张超 / 著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书介绍了 Elasticsearch 的系统原理，旨在帮助读者了解其内部原理、设计思想，以及在生产环境中如何正确地部署、优化系统。系统原理分两方面介绍，一方面详细介绍主要流程，例如启动流程、选主流程、恢复流程；另一方面介绍各重要模块的实现，以及模块之间的关系，例如 gateway 模块、allocation 模块等。本书的最后一部分介绍如何优化写入速度、搜索速度等大家关心的实际问题，并提供了一些诊断问题的方法和工具供读者参考。

本书适合对 Elasticsearch 进行改进的研发人员、平台运维人员，对分布式搜索感兴趣的朋友，以及在使用 Elasticsearch 过程中遇到问题的人们。

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

Elasticsearch 源码解析与优化实战 / 张超著. —北京：电子工业出版社，2018.11
ISBN 978-7-121-35216-4

I. ①E… II. ①张… III. ①搜索引擎—程序设计 IV. ①TP391.3

中国版本图书馆 CIP 数据核字（2018）第 238923 号

责任编辑：陈晓猛

印 刷：三河市良远印务有限公司

装 订：三河市良远印务有限公司

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：22.5

字数：432 千字

版 次：2018 年 11 月第 1 版

印 次：2018 年 11 月第 1 次印刷

定 价：89.00 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 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前言

我们可以在不关心原理的情况下使用 Elasticsearch（以下简称 ES），但要想用好 ES，就必须熟知其内部原理。

为什么要阅读代码？在传统软件行业，技术文档非常丰富。当开展一个项目时，从需求分析，到概要设计、详细设计，每个步骤都有相应的文档，从项目的整体架构、技术方案选型，到流程图、类图，细化到每个接口及参数。在这种情况下，想要搞清楚系统原理，并不需要阅读代码，文档上什么都有。但是互联网产品迭代快，技术文档不全，想要搞清楚原理，只能阅读代码，相当于从代码中逆向理解设计思想。

通过分析源码，我们可以有以下收获：

理解设计思想 当我们面临要解决的问题或实现的目标时，往往有多种方案可以选择。无论表面上看起来多么简单的架构，其背后都经过了深思熟虑。思考一下为什么使用现在的方案？有没有更好的解决方案？

探究内部机制的原理 某个技术点是怎么实现的？

搞明白执行流程 某个过程是什么样的，都做了什么？有几步？先做什么，后做什么？

熟悉代码结构 如果需要进行二次开发，则给出代码入口和调用关系，有时候找到某个逻辑的代码实现要花很多时间。

学以致用 借鉴其设计理念，掌握其解决问题的方式和方法，将来面对类似的问题时可以参考。

本书结构

本书由四部分组成，第一部分为基础知识和环境准备（第 1~2 章）；第二部分介绍 ES 的主要流程（第 3~10 章），包括集群启动流程、节点启动/关闭流程、选主流程、读写流程、搜索流程和索引恢复流程；第三部分主要介绍重要内部模块（第 11~17 章），包括 gateway 模块、allocation 模块、Snapshot 模块、Cluster 模块、Transport 模块和 ThreadPool 模块等；第四部分介

绍优化和诊断方法（第 18~22 章），包括写入速度优化、搜索速度优化、磁盘使用量优化，以及在生产环境中的实际应用建议，第 22 章介绍常用的问题诊断方法，排查集群遇到的问题。

术语约定

ES 中有一些特有的概念，这些概念对应的中文翻译约定如下：

- 分片（shard）；
- 主分片（primary shard），简称 P；
- 分片副本（特指数据的一个分片，无论主分片，还是副分片）；
- 副分片（replica shard），简称 R；
- 分片分配（shard allocation）；
- 集群状态（cluster state）；
- 分配决策（allocation decision）；
- 分配感知（allocation awareness）；
- 分配标识（allocation IDs）；
- 追踪（tracking）；
- 事务日志（translog）；
- 同步集合（in-sync set）。

行文约定

虽然本书是一本源码分析类图书，但原则上尽量少贴代码，引用的代码只是为了说明原理，因此所引用的代码并不保证和源码完全一致，对非核心逻辑有所删减，同时在代码块中，函数参数可能被省略，省略的函数参数用“...”表示，如：

```
executeBulk(...);
```

在引用代码中的某个方法时，使用#号分隔类名与方法名：

类名#方法名

一个索引由许多分片组成。我们用如下方式表示索引 `website` 的第 0 个分片：

```
website[0]
```

联系

读者有任何意见和建议都可以联系作者，邮箱：elasticsearchbook@163.com。

本书配套网站：www.elasticsearchbook.cn。

致谢

感谢李欣杰和郭东东，他们带我走进搜索领域；感谢韩洪伟，他让我学到了很多搜索系统的知识。欣杰和老韩都是资深的搜索架构师，能够和优秀的团队共事是我的荣幸。感谢 ES 团队的同事段军义，我们互相学习，一起解决了很多麻烦的问题。感谢出版社的策划编辑陈晓猛先生，他为本书的写作提供了很多建设性意见，并且耐心地编校了本书，让本书得以顺利出版。

感谢我的妻子和三岁的女儿，我爱你们！

张超

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn)，扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/35216>



目录

| | |
|--------------------------------|----|
| 第 1 章 走进 Elasticsearch..... | 1 |
| 1.1 基本概念和原理 | 1 |
| 1.1.1 索引结构 | 2 |
| 1.1.2 分片 (shard) | 2 |
| 1.1.3 动态更新索引 | 4 |
| 1.1.4 近实时搜索 | 5 |
| 1.1.5 段合并 | 5 |
| 1.2 集群内部原理 | 6 |
| 1.2.1 集群节点角色 | 6 |
| 1.2.2 集群健康状态 | 8 |
| 1.2.3 集群状态 | 8 |
| 1.2.4 集群扩容 | 8 |
| 1.3 客户端 API..... | 9 |
| 1.4 主要内部模块简介 | 10 |
| 1.4.1 模块结构 | 11 |
| 1.4.2 模块管理 | 12 |
| 第 2 章 准备编译和调试环境..... | 13 |
| 2.1 编译源码 | 13 |
| 2.1.1 准备 JDK 和 Gradle | 13 |
| 2.1.2 下载源代码 | 13 |
| 2.1.3 编译项目, 打包 | 14 |
| 2.1.4 将工程导入 IntelliJ IDEA..... | 14 |
| 2.2 调试 Elasticsearch | 16 |
| 2.2.1 本地运行、调试项目 | 16 |
| 2.2.2 远程调试 | 18 |
| 2.3 代码书签和断点组 | 19 |

| | |
|-------------------------------------|----|
| 第 3 章 集群启动流程..... | 21 |
| 3.1 选举主节点 | 22 |
| 3.2 选举集群元信息 | 22 |
| 3.3 allocation 过程..... | 23 |
| 3.4 index recovery | 24 |
| 3.5 集群启动日志 | 25 |
| 3.6 小结 | 26 |
| 第 4 章 节点的启动和关闭 | 28 |
| 4.1 启动流程做了什么 | 28 |
| 4.2 启动流程分析 | 28 |
| 4.2.1 启动脚本 | 28 |
| 4.2.2 解析命令行参数和配置文件 | 29 |
| 4.2.3 加载安全配置 | 30 |
| 4.2.4 检查内部环境 | 30 |
| 4.2.5 检测外部环境 | 30 |
| 4.2.6 启动内部模块 | 33 |
| 4.2.7 启动 keepalive 线程..... | 33 |
| 4.3 节点关闭流程 | 34 |
| 4.4 关闭流程分析 | 34 |
| 4.5 分片读写过程中执行关闭..... | 36 |
| 4.6 主节点被关闭 | 36 |
| 4.7 小结 | 36 |
| 第 5 章 选主流程..... | 38 |
| 5.1 设计思想 | 38 |
| 5.2 为什么使用主从模式 | 38 |
| 5.3 选举算法 | 39 |
| 5.4 相关配置 | 39 |
| 5.5 流程概述 | 41 |
| 5.6 流程分析 | 41 |
| 5.6.1 选举临时 Master | 42 |
| 5.6.2 投票与得票的实现 | 46 |
| 5.6.3 确立 Master 或加入集群 | 46 |
| 5.7 节点失效检测 | 47 |
| 5.7.1 NodesFaultDetection 事件处理..... | 47 |

| | | |
|--------------|---------------------------------------|-----------|
| 5.7.2 | MasterFaultDetection 事件处理..... | 48 |
| 5.8 | 小结..... | 49 |
| 第 6 章 | 数据模型..... | 50 |
| 6.1 | PacificA 算法..... | 50 |
| 6.1.1 | 数据副本策略..... | 51 |
| 6.1.2 | 配置管理..... | 52 |
| 6.1.3 | 错误检测..... | 52 |
| 6.2 | ES 的数据副本模型..... | 53 |
| 6.2.1 | 基本写入模型..... | 53 |
| 6.2.2 | 写故障处理..... | 54 |
| 6.2.3 | 基本读取模型..... | 54 |
| 6.2.4 | 读故障处理..... | 55 |
| 6.2.5 | 引申的含义..... | 55 |
| 6.2.6 | 系统异常..... | 56 |
| 6.3 | Allocation IDs..... | 56 |
| 6.3.1 | 安全地分配主分片..... | 56 |
| 6.3.2 | 将分配标记为陈旧..... | 57 |
| 6.2.3 | 一个例子..... | 58 |
| 6.3.4 | 不会丢失全部..... | 63 |
| 6.4 | Sequence IDs..... | 64 |
| 6.4.1 | Primary Terms 和 Sequence Numbers..... | 64 |
| 6.4.2 | 本地及全局检查点..... | 66 |
| 6.4.3 | 用于快速恢复 (Recovery)..... | 68 |
| 6.5 | _version..... | 69 |
| 第 7 章 | 写流程..... | 71 |
| 7.1 | 文档操作的定义..... | 71 |
| 7.2 | 可选参数..... | 72 |
| 7.3 | Index/Bulk 基本流程..... | 72 |
| 7.4 | Index/Bulk 详细流程..... | 73 |
| 7.4.1 | 协调节点流程..... | 74 |
| 7.4.2 | 主分片节点流程..... | 79 |
| 7.4.3 | 副分片节点流程..... | 82 |
| 7.5 | I/O 异常处理..... | 82 |
| 7.5.1 | Engine 关闭过程..... | 83 |

| | | |
|---------------|------------------------|------------|
| 7.5.2 | Master 的对应处理 | 84 |
| 7.5.3 | 异常流程总结 | 84 |
| 7.6 | 系统特性 | 84 |
| 7.7 | 思考 | 85 |
| 第 8 章 | GET 流程 | 86 |
| 8.1 | 可选参数 | 87 |
| 8.2 | GET 基本流程 | 87 |
| 8.3 | GET 详细分析 | 88 |
| 8.3.1 | 协调节点 | 89 |
| 8.3.2 | 数据节点 | 91 |
| 8.4 | MGET 流程分析 | 93 |
| 8.5 | 思考 | 94 |
| 第 9 章 | Search 流程 | 95 |
| 9.1 | 索引和搜索 | 96 |
| 9.1.1 | 建立索引 | 97 |
| 9.1.2 | 执行搜索 | 97 |
| 9.2 | search type | 98 |
| 9.3 | 分布式搜索过程 | 98 |
| 9.3.1 | 协调节点流程 | 99 |
| 9.3.2 | 执行搜索的数据节点流程 | 106 |
| 9.4 | 小结 | 108 |
| 第 10 章 | 索引恢复流程分析 | 110 |
| 10.1 | 相关配置 | 111 |
| 10.2 | 流程概述 | 111 |
| 10.3 | 主分片恢复流程 | 112 |
| 10.4 | 副分片恢复流程 | 116 |
| 10.4.1 | 流程概述 | 116 |
| 10.4.2 | synced flush 机制 | 118 |
| 10.4.3 | 副分片节点处理过程 | 118 |
| 10.4.4 | 主分片节点处理过程 | 123 |
| 10.5 | recovery 速度优化 | 127 |
| 10.6 | 如何保证副分片和主分片一致 | 128 |
| 10.7 | recovery 相关监控命令 | 131 |

| | |
|---|------------|
| 10.8 小结 | 133 |
| 第 11 章 gateway 模块分析 | 134 |
| 11.1 元数据 | 134 |
| 11.2 元数据的持久化..... | 135 |
| 11.3 元数据的恢复..... | 136 |
| 11.4 元数据恢复流程分析..... | 137 |
| 11.4.1 选举集群级和索引级别的元数据..... | 138 |
| 11.4.2 触发 allocation..... | 140 |
| 11.5 思考 | 140 |
| 第 12 章 allocation 模块分析..... | 141 |
| 12.1 什么是 allocation..... | 141 |
| 12.2 触发时机 | 142 |
| 12.3 allocation 模块结构概述..... | 142 |
| 12.4 allocators..... | 142 |
| 12.5 deciders..... | 143 |
| 12.5.1 负载均衡类 | 144 |
| 12.5.2 并发控制类 | 145 |
| 12.5.3 条件限制类 | 145 |
| 12.6 核心 reroute 实现..... | 146 |
| 12.6.1 集群启动时 reroute 的触发时机 | 147 |
| 12.6.2 流程分析 | 147 |
| 12.6.3 gatewayAllocator..... | 147 |
| 12.6.4 shardsAllocator..... | 154 |
| 12.7 从 gateway 到 allocation 流程的转换..... | 154 |
| 12.8 从 allocation 流程到 recovery 流程的转换..... | 155 |
| 12.9 思考 | 156 |
| 第 13 章 Snapshot 模块分析..... | 157 |
| 13.1 仓库 | 158 |
| 13.2 快照 | 160 |
| 13.2.1 创建快照 | 160 |
| 13.2.2 获取快照信息 | 161 |
| 13.2.3 快照 status | 163 |
| 13.2.4 取消、删除快照和恢复操作..... | 163 |

| | | |
|---------------|-----------------------------|------------|
| 13.3 | 从快照恢复 | 164 |
| 13.3.1 | 部分恢复 | 165 |
| 13.3.2 | 恢复过程中更改索引设置 | 165 |
| 13.3.3 | 监控恢复进度 | 165 |
| 13.4 | 创建快照的实现原理 | 166 |
| 13.4.1 | Lucene 文件格式简介 | 167 |
| 13.4.2 | 协调节点流程 | 168 |
| 13.4.3 | 主节点流程 | 170 |
| 13.4.4 | 数据节点流程 | 173 |
| 13.5 | 删除快照实现原理 | 184 |
| 13.5.1 | 协调节点流程 | 184 |
| 13.5.2 | 主节点流程 | 185 |
| 13.6 | 思考与总结 | 192 |
| 第 14 章 | Cluster 模块分析 | 194 |
| 14.1 | 集群状态 | 194 |
| 14.2 | 内部封装和实现 | 198 |
| 14.2.1 | MasterService | 198 |
| 14.2.2 | ClusterApplierService | 199 |
| 14.2.3 | 线程池 | 201 |
| 14.3 | 提交集群任务 | 202 |
| 14.3.1 | 内部模块如何提交任务 | 203 |
| 14.3.2 | 任务提交过程实现 | 205 |
| 14.4 | 集群任务的执行过程 | 209 |
| 14.5 | 集群状态的发布过程 | 211 |
| 14.5.1 | 增量发布的实现原理 | 213 |
| 14.5.2 | 二段提交总流程 | 214 |
| 14.5.3 | 发布过程 | 215 |
| 14.5.4 | 提交过程 | 216 |
| 14.5.5 | 异常处理 | 217 |
| 14.6 | 应用集群状态 | 217 |
| 14.7 | 查看等待执行的集群任务 | 219 |
| 14.8 | 任务管理 API | 220 |
| 14.8.1 | 列出运行中的任务 | 221 |
| 14.8.2 | 取消任务 | 222 |
| 14.9 | 思考与总结 | 222 |

| | |
|--|-----|
| 第 15 章 Transport 模块分析 | 223 |
| 15.1 配置信息 | 223 |
| 15.1.1 传输模块配置 | 223 |
| 15.1.2 通用网络配置 | 225 |
| 15.2 Transport 总体架构 | 227 |
| 15.2.1 网络层 | 227 |
| 15.2.2 服务层 | 229 |
| 15.3 REST 解析和处理 | 234 |
| 15.4 RPC 实现 | 235 |
| 15.4.1 RPC 的注册和映射 | 236 |
| 15.4.2 根据 Action 获取处理类 | 240 |
| 15.5 思考与总结 | 241 |
| 第 16 章 ThreadPool 模块分析 | 242 |
| 16.1 线程池类型 | 243 |
| 16.1.1 fixed | 244 |
| 16.1.2 scaling | 244 |
| 16.1.3 direct | 245 |
| 16.1.4 fixed_auto_queue_size | 245 |
| 16.2 处理器设置 | 245 |
| 16.3 查看线程池 | 246 |
| 16.3.1 cat thread pool | 246 |
| 16.3.2 nodes info | 247 |
| 16.3.3 nodes stats | 248 |
| 16.3.4 nodes hot threads | 248 |
| 16.3.5 Java 的线程池结构 | 250 |
| 16.4 ES 的线程池实现 | 252 |
| 16.4.1 ThreadPool 类结构与初始化 | 253 |
| 16.4.2 fixed 类型线程池构建过程 | 255 |
| 16.4.3 scaling 类型线程池构建过程 | 256 |
| 16.4.4 direct 类型线程池构建过程 | 256 |
| 16.4.5 fixed_auto_queue_size 类型线程池构建过程 | 257 |
| 16.5 其他线程池 | 258 |
| 16.6 思考与总结 | 258 |
| 第 17 章 Shrink 原理分析 | 259 |

| | | |
|---------------|-------------------------------|------------|
| 17.1 | 准备源索引 | 259 |
| 17.2 | 缩小索引 | 260 |
| 17.3 | Shrink 的工作原理 | 260 |
| 17.3.1 | 创建新索引 | 261 |
| 17.3.2 | 创建硬链接 | 261 |
| 17.3.3 | 硬链接过程源码分析 | 262 |
| 第 18 章 | 写入速度优化 | 264 |
| 18.1 | translog flush 间隔调整 | 264 |
| 18.2 | 索引刷新间隔 refresh_interval | 265 |
| 18.3 | 段合并优化 | 265 |
| 18.4 | indexing buffer | 266 |
| 18.5 | 使用 bulk 请求 | 267 |
| 18.5.1 | bulk 线程池和队列 | 267 |
| 18.5.2 | 并发执行 bulk 请求 | 267 |
| 18.6 | 磁盘间的任务均衡 | 268 |
| 18.7 | 节点间的任务均衡 | 268 |
| 18.8 | 索引过程调整和优化 | 269 |
| 18.8.1 | 自动生成 doc ID | 269 |
| 18.8.2 | 调整字段 Mappings | 269 |
| 18.8.3 | 调整_source 字段 | 269 |
| 18.8.4 | 禁用_all 字段 | 270 |
| 18.8.5 | 对 Analyzed 的字段禁用 Norms | 271 |
| 18.8.6 | index_options 设置 | 271 |
| 18.9 | 参考配置 | 271 |
| 18.10 | 思考与总结 | 272 |
| 第 19 章 | 搜索速度的优化 | 273 |
| 19.1 | 为文件系统 cache 预留足够的内存 | 273 |
| 19.2 | 使用更快的硬件 | 273 |
| 19.3 | 文档模型 | 274 |
| 19.4 | 预索引数据 | 274 |
| 19.5 | 字段映射 | 276 |
| 19.6 | 避免使用脚本 | 276 |
| 19.7 | 优化日期搜索 | 276 |
| 19.8 | 为只读索引执行 force-merge | 278 |

| | | |
|---------------|------------------------------|------------|
| 19.9 | 预热全局序号 (global ordinals) | 279 |
| 19.10 | execution hint | 279 |
| 19.11 | 预热文件系统 cache | 280 |
| 19.12 | 转换查询表达式 | 280 |
| 19.13 | 调节搜索请求中的 batched_reduce_size | 281 |
| 19.14 | 使用近似聚合 | 281 |
| 19.15 | 深度优先还是广度优先 | 281 |
| 19.16 | 限制搜索请求的分片数 | 281 |
| 19.17 | 利用自适应副本选择 (ARS) 提升 ES 响应速度 | 282 |
| 第 20 章 | 磁盘使用量优化 | 285 |
| 20.1 | 预备知识 | 285 |
| 20.1.1 | 元数据字段 | 285 |
| 20.1.2 | 索引映射参数 | 286 |
| 20.2 | 优化措施 | 287 |
| 20.2.1 | 禁用对你来说不需要的特性 | 287 |
| 20.2.2 | 禁用 doc values | 290 |
| 20.2.3 | 不要使用默认的动态字符串映射 | 290 |
| 20.2.4 | 观察分片大小 | 291 |
| 20.2.5 | 禁用 _source | 291 |
| 20.2.6 | 使用 best_compression | 291 |
| 20.2.7 | Force Merge | 292 |
| 20.2.8 | Shrink Index | 292 |
| 20.2.9 | 数值类型长度够用就好 | 292 |
| 20.2.10 | 使用索引排序来排列类似的文档 | 292 |
| 20.2.11 | 在文档中以相同的顺序放置字段 | 292 |
| 20.3 | 测试数据 | 293 |
| 第 21 章 | 综合应用实践 | 294 |
| 21.1 | 集群层 | 294 |
| 21.1.1 | 规划集群规模 | 294 |
| 21.1.2 | 单节点还是多节点部署 | 295 |
| 21.1.3 | 移除节点 | 295 |
| 21.1.4 | 独立部署主节点 | 296 |
| 21.2 | 节点层 | 296 |
| 21.2.1 | 控制线程池的队列大小 | 296 |

| | |
|--|------------|
| 21.2.2 为系统 cache 保留一半物理内存 | 297 |
| 21.3 系统层 | 297 |
| 21.3.1 关闭 swap | 297 |
| 21.3.2 配置 Linux OOM Killer | 297 |
| 21.3.3 优化内核参数 | 298 |
| 21.4 索引层 | 304 |
| 21.4.1 使用全局模板 | 304 |
| 21.4.2 索引轮转 | 304 |
| 21.4.3 避免热索引分片不均 | 305 |
| 21.4.4 副本数选择 | 306 |
| 21.4.5 Force Merge | 306 |
| 21.4.6 Shrink Index | 306 |
| 21.4.7 close 索引 | 307 |
| 21.4.8 延迟分配分片 | 307 |
| 21.4.9 小心地使用 fielddata | 307 |
| 21.5 客户端 | 308 |
| 21.5.1 使用 REST API 而非 Java API | 308 |
| 21.5.2 注意 429 状态码 | 308 |
| 21.5.3 curl 的 HEAD 请求 | 308 |
| 21.5.4 了解你的搜索计划 | 309 |
| 21.5.5 为读写请求设置比较长的超时时间 | 309 |
| 21.6 读写 | 309 |
| 21.6.1 避免搜索操作返回巨大的结果集 | 309 |
| 21.6.2 避免索引巨大的文档 | 309 |
| 21.6.3 避免使用多个 _type | 310 |
| 21.6.4 避免使用 _all 字段 | 310 |
| 21.6.5 避免将请求发送到同一个协调节点 | 310 |
| 21.7 控制相关度 | 311 |
| 第 22 章 故障诊断 | 316 |
| 22.1 使用 Profile API 定位慢查询 | 317 |
| 22.2 使用 Explain API 分析未分配的分片 (Unassigned Shards) | 320 |
| 22.2.1 诊断未分配的主分片 | 320 |
| 22.2.2 诊断未分配的副分片 | 324 |
| 22.2.3 诊断已分配的分片 | 326 |
| 22.3 节点 CPU 使用率高 | 328 |

| | | |
|--------|-----------------|-----|
| 22.4 | 节点内存使用率高 | 330 |
| 22.5 | Slow Logs | 333 |
| 22.6 | 分析工具 | 334 |
| 22.6.1 | I/O 信息 | 334 |
| 22.6.2 | 内存 | 335 |
| 22.6.3 | CPU 信息 | 337 |
| 22.6.4 | 网络连接和流量 | 339 |
| 22.7 | 小结 | 341 |
| 附录 A | 重大版本变化 | 342 |