

目 录 Contents

译者序	
前言	
致谢	
作者简介	
评审者简介	

第 1 章 ElasticSearch 简介	1
1.1 Apache Lucene 简介	1
1.1.1 熟悉 Lucene	2
1.1.2 Lucene 的总体架构	2
1.1.3 分析你的数据	3
1.1.4 Lucene 查询语言	4
1.2 ElasticSearch 简介	6
1.2.1 ElasticSearch 的基本概念	7
1.2.2 ElasticSearch 架构背后的 关键概念	8
1.2.3 ElasticSearch 的工作流程	9
1.3 小结	13

第 2 章 查询 DSL 进阶	14
2.1 Apache Lucene 默认评分公式解释	14
2.1.1 何时文档被匹配上	15
2.1.2 TF/IDF 评分公式	15

2.1.3 ElasticSearch 如何看评分	16
2.2 查询改写	17
2.2.1 前缀查询范例	17
2.2.2 回顾 Apache Lucene	19
2.2.3 查询改写的属性	20
2.3 二次评分	21
2.3.1 理解二次评分	21
2.3.2 范例数据	21
2.3.3 查询	22
2.3.4 二次评分查询的结构	22
2.3.5 二次评分参数配置	23
2.3.6 小结	24
2.4 批量操作	24
2.4.1 批量取	24
2.4.2 批量查询	26
2.5 排序	27
2.5.1 基于多值字段的排序	28
2.5.2 基于多值 geo 字段的排序	28
2.5.3 基于嵌套对象的排序	30
2.6 数据更新 API	31
2.6.1 简单字段更新	31
2.6.2 使用脚本按条件更新	32

2.6.3	使用更新 API 创建或删除文档	33
2.7	使用过滤器优化查询	33
2.7.1	过滤器与缓存	34
2.7.2	词项查找过滤器	36
2.8	ElasticSearch 切面机制中的 过滤器与作用域	40
2.8.1	范例数据	40
2.8.2	切面计算和过滤	41
2.8.3	过滤器作为查询的一部分	42
2.8.4	切面过滤器	44
2.8.5	全局作用域	45
2.9	小结	47

第 3 章 底层索引控制 48

3.1	改变 Apache Lucene 的评分方式	48
3.1.1	可用的相似度模型	49
3.1.2	为每字段配置相似度模型	49
3.2	相似度模型配置	50
3.2.1	选择默认的相似度模型	51
3.2.2	配置被选用的相似度模型	52
3.3	使用编解码器	53
3.3.1	简单使用范例	53
3.3.2	工作原理解释	54
3.3.3	可用的倒排表格式	55
3.3.4	配置编解码器	56
3.4	准实时、提交、更新及事务日志	58
3.4.1	索引更新及更新提交	59
3.4.2	事务日志	60
3.4.3	准实时读取	62
3.5	深入理解数据处理	62
3.5.1	输入并不总是进行文本分析	62

3.5.2	范例的使用	65
3.5.3	索引期更换分词器	67
3.5.4	搜索时更换分析器	68
3.5.5	陷阱与默认分析	68
3.6	控制索引合并	68
3.6.1	选择正确的合并策略	69
3.6.2	合并策略配置	70
3.6.3	调度	72
3.7	小结	73

第 4 章 分布式索引架构 74

4.1	选择合适的分片和副本数	74
4.1.1	分片和过度分配	75
4.1.2	一个过度分配的正面例子	75
4.1.3	多分片与多索引	76
4.1.4	副本	76
4.2	路由	76
4.2.1	分片和数据	77
4.2.2	测试路由功能	77
4.2.3	索引时使用路由	80
4.2.4	别名	83
4.2.5	多个路由值	83
4.3	调整默认的分片分配行为	84
4.3.1	分片分配器简介	84
4.3.2	even_shard 分片分配器	84
4.3.3	balanced 分片分配器	85
4.3.4	自定义分片分配器	85
4.3.5	裁决者	86
4.4	调整分片分配	88
4.4.1	部署意识	89
4.4.2	过滤	91

4.4.3 运行时更新分配策略	92	6.2 关于 I/O 调节	136
4.4.4 确定每个节点允许的总分片数	93	6.2.1 控制 IO 节流	136
4.4.5 更多的分片分配属性	96	6.2.2 配置	136
4.5 查询执行偏好	97	6.3 用预热器提升查询速度	138
4.6 应用我们的知识	99	6.3.1 为什么使用预热器	138
4.6.1 基本假定	99	6.3.2 操作预热器	138
4.6.2 配置	100	6.3.3 测试预热器	141
4.6.3 变化来了	104	6.4 热点线程	144
4.7 小结	105	6.4.1 澄清热点线程 API 的用法 误区	145
第 5 章 管理 Elasticsearch	106	6.4.2 热点线程 API 的响应信息	145
5.1 选择正确的目录实现 – 存储模块	106	6.5 现实场景	146
5.2 发现模块的配置	109	6.5.1 越来越差的性能	146
5.2.1 Zen 发现	109	6.5.2 混杂的环境和负载不平衡	148
5.2.2 亚马逊 EC2 发现	111	6.5.3 我的服务器出故障了	149
5.2.3 本地网关	114	6.6 小结	150
5.2.4 恢复配置	115	第 7 章 改善用户搜索体验	151
5.3 索引段统计	116	7.1 改正用户拼写错误	151
5.3.1 segments API 简介	116	7.1.1 测试数据	152
5.3.2 索引段信息的可视化	118	7.1.2 深入技术细节	152
5.4 理解 Elasticsearch 缓存	119	7.1.3 completion suggester	168
5.4.1 过滤器缓存	119	7.2 改善查询相关性	172
5.4.2 字段数据缓存	121	7.2.1 数据	172
5.4.3 清除缓存	126	7.2.2 改善相关性的探索之旅	174
5.5 小结	127	7.3 小结	188
第 6 章 故障处理	129	第 8 章 Elasticsearch Java API	189
6.1 了解垃圾回收器	129	8.1 Elasticsearch Java API 简介	189
6.1.1 Java 内存	130	8.2 代码	190
6.1.2 处理垃圾回收问题	131	8.3 连接到集群	191
6.1.3 在类 UNIX 系统中避免内存 交换	135	8.3.1 成为 Elasticsearch 节点	191

8.3.2 使用传输机连接方式	192	8.7.4 Multi Search	212
8.3.3 选择合适的连接方式	193	8.8 Percolator	213
8.4 API 剖析	194	8.9 explain API	214
8.5 CRUD 操作	195	8.10 构造 JSON 格式的查询和文档	214
8.5.1 读取文档	195	8.11 管理 API	216
8.5.2 索引文档	197	8.11.1 集群管理 API	216
8.5.3 更新文档	199	8.11.2 索引管理 API	219
8.5.4 删除文档	201	8.12 小结	226
8.6 Elasticsearch 查询	203		
8.6.1 准备查询请求	203	第 9 章 开发 Elasticsearch 插件	227
8.6.2 构造查询	203	9.1 建立 Apache Maven 项目结构	227
8.6.3 分页	206	9.1.1 了解基本知识	228
8.6.4 排序	207	9.1.2 Maven Java 项目的结构	228
8.6.5 过滤	207	9.1.3 POM 的理念	228
8.6.6 切面计算	208	9.1.4 运行构建过程	229
8.6.7 高亮	209	9.1.5 引入 Maven 装配插件	230
8.6.8 查询建议	209	9.2 创建一个自定义 river 插件	232
8.6.9 计数	210	9.2.1 实现细节	232
8.6.10 滚动	211	9.2.2 测试 river	238
8.7 批量执行多个操作	211	9.3 创建自定义分析插件	240
8.7.1 批量操作	211	9.3.1 实现细节	240
8.7.2 根据查询删除文档	212	9.3.2 测试自定义分析插件	247
8.7.3 Multi GET	212	9.4 小结	249