

亿级流量网站架构核心技术



[亿级流量网站架构核心技术_下载链接1](#)

著者:张开涛

出版者:电子工业出版社

出版时间:2017-4

装帧:平装

isbn:9787121309540

《亿级流量网站架构核心技术》一书总结并梳理了亿级流量网站高可用和高并发原则，通过实例详细介绍了如何落地这些原则。本书分为四部分：概述、高可用原则、高并发原则、案例实战。从负载均衡、限流、降级、隔离、超时与重试、回滚机制、压测与预

案、缓存、池化、异步化、扩容、队列等多方面详细介绍了亿级流量网站的架构核心技术，让读者看后能快速运用到实践项目中。

不管是软件开发人员，还是运维人员，通过阅读《亿级流量网站架构核心技术》都能系统地学习实现亿级流量网站的架构核心技术，并收获解决系统问题的思路和方法。

作者介绍:

张开涛，现就职于京东，“开涛的博客”公众号作者。写过《跟我学Spring》《跟我学Spring MVC》《跟我学Shiro》《跟我学Nginx+Lua开发》等系列教程，博客现有1000多万访问量。

目录: 第1部分 概述 / 1	
1 交易型系统设计的一些原则 / 2	
1.1 高并发原则 / 3	
1.1.1 无状态 / 3	
1.1.2 拆分 / 3	
1.1.3 服务化 / 4	
1.1.4 消息队列 / 4	
1.1.5 数据异构 / 6	
1.1.6 缓存银弹 / 7	
1.1.7 并发化 / 9	
1.2 高可用原则 / 10	
1.2.1 降级 / 10	
1.2.2 限流 / 11	
1.2.3 切流量 / 12	
1.2.4 可回滚 / 12	
1.3 业务设计原则 / 12	
1.3.1 防重设计 / 13	
1.3.2 幂等设计 / 13	
1.3.3 流程可定义 / 13	
1.3.4 状态与状态机 / 13	
1.3.5 后台系统操作可反馈 / 14	
1.3.6 后台系统审批化 / 14	
1.3.7 文档和注释 / 14	
1.3.8 备份 / 14	
1.4 总结 / 14	
第2部分 高可用 / 17	
2 负载均衡与反向代理 / 18	
2.1 upstream配置 / 20	
2.2 负载均衡算法 / 21	
2.3 失败重试 / 23	
2.4 健康检查 / 24	
2.4.1 TCP心跳检查 / 24	
2.4.2 HTTP心跳检查 / 25	
2.5 其他配置 / 25	
2.5.1 域名上游服务器 / 25	
2.5.2 备份上游服务器 / 26	
2.5.3 不可用上游服务器 / 26	
2.6 长连接 / 26	
2.7 HTTP反向代理示例 / 29	

2.8 HTTP动态负载均衡 / 30	
2.8.1 Consul+Consul-template / 31	
2.8.2 Consul+OpenResty / 35	
2.9 Nginx四层负载均衡 / 39	
2.9.1 静态负载均衡 / 39	
2.9.2 动态负载均衡 / 41	
参考资料 / 42	
3 隔离术 / 43	
3.1 线程隔离 / 43	
3.2 进程隔离 / 45	
3.3 集群隔离 / 45	
3.4 机房隔离 / 46	
3.5 读写隔离 / 47	
3.6 动静隔离 / 48	
3.7 爬虫隔离 / 49	
3.8 热点隔离 / 50	
3.9 资源隔离 / 50	
3.10 使用Hystrix实现隔离 / 51	
3.10.1 Hystrix简介 / 51	
3.10.2 隔离示例 / 52	
3.11 基于Servlet 3实现请求隔离 / 56	
3.11.1 请求解析和业务处理线程池分离 / 57	
3.11.2 业务线程池隔离 / 58	
3.11.3 业务线程池监控/运维/降级 / 58	
3.11.4 如何使用Servlet 3异步化 / 59	
3.11.5 一些Servlet 3异步化压测数据 / 64	
4 限流详解 / 66	
4.1 限流算法 / 67	
4.1.1 令牌桶算法 / 67	
4.1.2 漏桶算法 / 68	
4.2 应用级限流 / 69	
4.2.1 限流总并发/连接/请求数 / 69	
4.2.2 限流总资源数 / 70	
4.2.3 限流某个接口的总并发/请求数 / 70	
4.2.4 限流某个接口的时间窗请求数 / 70	
4.2.5 平滑限流某个接口的请求数 / 71	
4.3 分布式限流 / 75	
4.3.1 Redis+Lua实现 / 76	
4.3.2 Nginx+Lua实现 / 77	
4.4 接入层限流 / 78	
4.4.1 ngx_http_limit_conn_module / 78	
4.4.2 ngx_http_limit_req_module / 80	
4.4.3 lua-resty-limit-traffic / 88	
4.5 节流 / 90	
4.5.1 throttleFirst/throttleLast / 90	
4.5.2 throttleWithTimeout / 91	
参考资料 / 92	
5 降级特技 / 93	
5.1 降级预案 / 93	
5.2 自动开关降级 / 95	
5.2.1 超时降级 / 95	
5.2.2 统计失败次数降级 / 95	
5.2.3 故障降级 / 95	
5.2.4 限流降级 / 95	

5.3	人工开关降级 / 96
5.4	读服务降级 / 96
5.5	写服务降级 / 97
5.6	多级降级 / 98
5.7	配置中心 / 100
5.7.1	应用层API封装 / 100
5.7.2	配置文件实现开关配置 / 101
5.7.3	配置中心实现开关配置 / 102
5.8	使用Hystrix实现降级 / 106
5.9	使用Hystrix实现熔断 / 108
5.9.1	熔断机制实现 / 108
5.9.2	配置示例 / 112
5.9.3	采样统计 / 113
6	超时与重试机制 / 117
6.1	简介 / 117
6.2	代理层超时与重试 / 119
6.2.1	Nginx / 119
6.2.2	Twemproxy / 126
6.3	Web容器超时 / 127
6.4	中间件客户端超时与重试 / 127
6.5	数据库客户端超时 / 131
6.6	NoSQL客户端超时 / 134
6.7	业务超时 / 135
6.8	前端Ajax超时 / 135
6.9	总结 / 136
6.10	参考资料 / 137
7	回滚机制 / 139
7.1	事务回滚 / 139
7.2	代码库回滚 / 140
7.3	部署版本回滚 / 141
7.4	数据版本回滚 / 142
7.5	静态资源版本回滚 / 143
8	压测与预案 / 145
8.1	系统压测 / 145
8.1.1	线下压测 / 146
8.1.2	线上压测 / 146
8.2	系统优化和容灾 / 147
8.3	应急预案 / 148
第3部分	高并发 / 153
9	应用级缓存 / 154
9.1	缓存简介 / 154
9.2	缓存命中率 / 155
9.3	缓存回收策略 / 155
9.3.1	基于空间 / 155
9.3.2	基于容量 / 155
9.3.3	基于时间 / 155
9.3.4	基于Java对象引用 / 156
9.3.5	回收算法 / 156
9.4	Java缓存类型 / 156
9.4.1	堆缓存 / 158
9.4.2	堆外缓存 / 162
9.4.3	磁盘缓存 / 162
9.4.4	分布式缓存 / 164
9.4.5	多级缓存 / 166

9.5 应用级缓存示例 / 167	
9.5.1 多级缓存API封装 / 167	
9.5.2 NULL Cache / 170	
9.5.3 强制获取最新数据 / 170	
9.5.4 失败统计 / 171	
9.5.5 延迟报警 / 171	
9.6 缓存使用模式实践 / 172	
9.6.1 Cache-Aside / 173	
9.6.2 Cache-As-SoR / 174	
9.6.3 Read-Through / 174	
9.6.4 Write-Through / 176	
9.6.5 Write-Behind / 177	
9.6.6 Copy Pattern / 181	
9.7 性能测试 / 181	
9.8 参考资料 / 182	
10 HTTP缓存 / 183	
10.1 简介 / 183	
10.2 HTTP缓存 / 184	
10.2.1 Last-Modified / 184	
10.2.2 ETag / 190	
10.2.3 总结 / 192	
10.3 HttpClient客户端缓存 / 192	
10.3.1 主流程 / 195	
10.3.2 清除无效缓存 / 195	
10.3.3 查找缓存 / 196	
10.3.4 缓存未命中 / 198	
10.3.5 缓存命中 / 198	
10.3.6 缓存内容陈旧需重新验证 / 202	
10.3.7 缓存内容无效需重新执行请求 / 205	
10.3.8 缓存响应 / 206	
10.3.9 缓存头总结 / 207	
10.4 Nginx HTTP缓存设置 / 208	
10.4.1 expires / 208	
10.4.2 if-modified-since / 209	
10.4.3 nginx proxy_pass / 209	
10.5 Nginx代理层缓存 / 212	
10.5.1 Nginx代理层缓存配置 / 212	
10.5.2 清理缓存 / 215	
10.6 一些经验 / 216	
参考资料 / 217	
11 多级缓存 / 218	
11.1 多级缓存介绍 / 218	
11.2 如何缓存数据 / 220	
11.2.1 过期与不过期 / 220	
11.2.2 维度化缓存与增量缓存 / 221	
11.2.3 大Value缓存 / 221	
11.2.4 热点缓存 / 221	
11.3 分布式缓存与应用负载均衡 / 222	
11.3.1 缓存分布式 / 222	
11.3.2 应用负载均衡 / 222	
11.4 热点数据与更新缓存 / 223	
11.4.1 单机全量缓存+主从 / 223	
11.4.2 分布式缓存+应用本地热点 / 224	
11.5 更新缓存与原子性 / 225	

- 11.6 缓存崩溃与快速修复 / 226
 - 11.6.1 取模 / 226
 - 11.6.2 一致性哈希 / 226
 - 11.6.3 快速恢复 / 226
- 12 连接池线程池详解 / 227
 - 12.1 数据库连接池 / 227
 - 12.1.1 DBCP连接池配置 / 228
 - 12.1.2 DBCP配置建议 / 233
 - 12.1.3 数据库驱动超时实现 / 234
 - 12.1.4 连接池使用的一些建议 / 235
 - 12.2 HttpClient连接池 / 236
 - 12.2.1 HttpClient 4.5.2配置 / 236
 - 12.2.2 HttpClient连接池源码分析 / 240
 - 12.2.3 HttpClient 4.2.3配置 / 241
 - 12.2.4 问题示例 / 243
 - 12.3 线程池 / 244
 - 12.3.1 Java线程池 / 245
 - 12.3.2 Tomcat线程池配置 / 248
- 13 异步并发实战 / 250
 - 13.1 同步阻塞调用 / 251
 - 13.2 异步Future / 252
 - 13.3 异步Callback / 253
 - 13.4 异步编排CompletableFuture / 254
 - 13.5 异步Web服务实现 / 257
 - 13.6 请求缓存 / 259
 - 13.7 请求合并 / 261
- 14 如何扩容 / 266
 - 14.1 单体应用垂直扩容 / 267
 - 14.2 单体应用水平扩容 / 267
 - 14.3 应用拆分 / 268
 - 14.4 数据库拆分 / 271
 - 14.5 数据库分库分表示例 / 275
 - 14.5.1 应用层还是中间件层 / 275
 - 14.5.2 分库分表策略 / 277
 - 14.5.3 使用sharding-jdbc分库分表 / 279
 - 14.5.4 sharding-jdbc分库分表配置 / 279
 - 14.5.5 使用sharding-jdbc读写分离 / 283
 - 14.6 数据异构 / 284
 - 14.6.1 查询维度异构 / 284
 - 14.6.2 聚合数据异构 / 285
 - 14.7 任务系统扩容 / 285
 - 14.7.1 简单任务 / 285
 - 14.7.2 分布式任务 / 287
 - 14.7.3 Elastic-Job简介 / 287
 - 14.7.4 Elastic-Job-Lite功能与架构 / 287
 - 14.7.5 Elastic-Job-Lite示例 / 288
- 15 队列术 / 295
 - 15.1 应用场景 / 295
 - 15.2 缓冲队列 / 296
 - 15.3 任务队列 / 297
 - 15.4 消息队列 / 297
 - 15.5 请求队列 / 299
 - 15.6 数据总线队列 / 300
 - 15.7 混合队列 / 301

15.8 其他队列 / 302	
15.9 Disruptor+Redis队列 / 303	
15.10 下单系统水平可扩展架构 / 311	
第4部分 案例 / 323	
16 构建需求响应式亿级商品详情页 / 324	
16.1 商品详情页是什么 / 324	
16.2 商品详情页前端结构 / 325	
16.3 我们的性能数据 / 327	
16.4 单品页流量特点 / 327	
16.5 单品页技术架构发展 / 327	
16.5.1 架构1.0 / 328	
16.5.2 架构2.0 / 328	
16.5.3 架构3.0 / 330	
16.6 详情页架构设计原则 / 332	
16.7 遇到的一些坑和问题 / 339	
16.8 其他 / 347	
17 京东商品详情页服务闭环实践 / 348	
17.1 为什么需要统一服务 / 348	
17.2 整体架构 / 349	
17.3 一些架构思路和总结 / 350	
17.4 引入Nginx接入层 / 354	
17.5 前端业务逻辑后置 / 356	
17.6 前端接口服务端聚合 / 357	
17.7 服务隔离 / 359	
18 使用OpenResty开发高性能Web应用 / 360	
18.1 OpenResty简介 / 361	
18.1.1 Nginx优点 / 361	
18.1.2 Lua的优点 / 361	
18.1.3 什么是ngx_lua / 361	
18.1.4 开发环境 / 362	
18.1.5 OpenResty生态 / 362	
18.1.6 场景 / 362	
18.2 基于OpenResty的常用架构模式 / 363	
18.3 如何使用OpenResty开发Web应用 / 371	
18.4 基于OpenResty的常用功能总结 / 375	
18.5 一些问题 / 376	
19 应用数据静态化架构高性能单页Web应用 / 377	
19.1 整体架构 / 378	
19.2 数据和模板动态化 / 381	
19.3 多版本机制 / 381	
19.4 异常问题 / 382	
20 使用OpenResty开发Web服务 / 383	
20.1 架构 / 383	
20.2 单DB架构 / 384	
20.3 实现 / 387	
21 使用OpenResty开发商品详情页 / 405	
21.1 技术选型 / 407	
21.2 核心流程 / 408	
21.3 项目搭建 / 408	
21.4 数据存储实现 / 410	
21.5 动态服务实现 / 422	
21.6 前端展示实现 / 430	
• • • • • (收起)	

标签

架构

高可用

高并发

互联网

性能

计算机

服务器

Web

评论

都是一笔带过的常识，夹杂大量配置文件和 demo 代码……
懂的人不用看，不懂的人看了也没用那种书。

用一半代码注水的技术书，干货不多，启发不大，国产书就是这么不用心的注水吗？

了解电商，颇有“原来如此”之感。但案例讲解不够细致，且贴配置和代码注水。

介绍了很多实用的技术以及应用场景，偏实战，配置感觉多了点，如果更深入点还不错。
。 btw
好多引用的文章就贴了一个二维码，但是真的看就很麻烦，建议放到GitHub上面做个
本书引用，也方便大家。 ~again one more
time~合适的码龄段读起来还是可以补充些技能点的。
感觉作者想让一本书塞很多知识点就导致了都是简单聊聊没有更加深入

注水严重，懂的人不用看，不懂的人看不懂，飞机上囫囵吞枣扫完

这本书对大流量的网站架构介绍得比较清晰和全面，不过没有很深入，一些框架的简略
介绍和示例代码占了不少篇幅

it's a good book. i used it for interviews. even though things were not going really well,
but still, it's a good book. worth to read.

偏向前端，涉及面很广，处理各类性能问题，内容点到而止

主要讲术，原理性的不多

涉及的范围很广，难免顾此失彼，想把什么都讲清楚，比较困难，总体来说还是不错的

一本膨大而自信的实战干货checklist，不重在提挈启迪

什么都想讲，什么都讲不细

科普类的图书，对于我这样前端开发，可以对后台开发技术和架构有个大致的了解，不至于抓瞎

趁双11来之前，迅速把这本书刷一遍。屠龙术已get，就差龙了

还是有不少实战干货，主要不足是接入层nginx相关内容比重太大

前三章还行，最后一百页太水了。确实是网友那句话：懂的人不用看，不懂的人看不懂

国内讲述这类内容的书籍并不多，每出一本我都会仔细阅读，这本书比想象中的要厚，内容扎实细密，比较严谨，值得阅读。

需要重构

内容很泛夹杂很多代码片段截图，断断续续翻了一遍补充下常识性的东西，细节还是需要实践才能有更深入的理解。

但入门知识可以，杂而不深入

[亿级流量网站架构核心技术 下载链接1](#)

书评

书分四个部分：指导原则，高可用，高并发，实践案例。这篇文章只说一说前三个部分。
指导原则 这里有一些并不是原则，而是技巧。我理解的原则如下：高并发原则：
1. 无状态设计：因为有状态可能涉及锁操作，锁又可能导致并发的串行化。 2.

保持合理的粒度：无论拆分还...

1.

我很早就读过作者的博文，也经常从作者的微信公众号的文章里收获很多，所以一直很期待这本书，一是相信作者的水平，二是多谢他多年来的分享； 2. 这本书的选题立意很棒，这一类的书籍国内外基本上没有，随着互联网的蓬勃发展，这一块的需求攀升。作者结合他在京东的工作和所...

除了目录整理得还行之外，内容真是看不下去，尝试了好几次强迫自己坚持看下去，看了一大半的样子实在坚持不下去了。大量的具体配置细节，这是定位于工具书么？可是书中想要涉及的点又很多，这么细的东西肯定也只能覆盖很小的一面了。而真正原理性的东西往往一笔带过或者干脆没...

《亿级流量网站架构核心技术》一书总结并梳理了亿级流量网站高可用和高并发原则，通过实例详细介绍了如何落地这些原则。本书分为四部分：概述、高可用原则、高并发原则、案例实战。从负载均衡、限流、降级、隔离、超时与重试、回滚机制、压测与预案、缓存、池化、异步化、扩容...

江湖人称”涛哥”，京东大牛，一次偶然的机会，通过中生代技术了解到了这本书，一些大牛的公众号也有推荐过此书，真正的干货实战书。去年下半年购买入手，读过了一次。当时正在紧急开发一个互联网项目，用户体量和访问量预估还不低，虽然有过类似项目的架构设计和开发经验， ...

[亿级流量网站架构核心技术_下载链接1](#)