对于高并发的场景来说，比如电商类，o2o，门户，等等互联网类的项目，缓存技术是Java项目中最常见的一种应用技术。然而，行业里很多朋友对缓存技术的了解与掌握，仅仅停留在掌握redis/memcached等缓存技术的基础使用，最多了解一些集群相关的知识，大部分人都可以对缓存技术掌握到这个程度。然而，仅仅对缓存相关的技术掌握到这种程度，无论是对于开发复杂的高并发系统，或者是在往Java高级工程师、Java资深工程师、Java架构师这些高阶的职位发展的过程中，都是完全不够用的。技术成长出现瓶颈，在自己公司的项目中，没有任何高并发与高可用的挑战性项目，自己不知道如何成长，自己也不知道如何让自己的技术更上一层楼。这成为了很多同学的职业发展的困惑。

同样的，高可用相关的技术以及架构，对于大型复杂的分布式系统，也是非常的重要。高可用架构中，非常重要的一个环节，就是如何将分布式系统中的各个服务打造成高可用的服务，足以应对分布式系统中各种各样的异常问题，比如服务间调用超时或者失败。这就涉及到了高可用分布式系统中的很多重要的技术，包括资源隔离，限流与过载保护，熔断，优雅降级，容错，超时控制，监控运维，等等。而行业中相当比例的同学，对高可用系统架构以及相关的技术，几乎没有太多的了解。同时也成为了你设计一个复杂的高可用系统架构，包括面试高阶的Java职位时的一个重要的阻碍。

相信很多朋友都会有这种感觉，自己的技术不知道如何成长，在公司里遇到复杂的业务场景时，瞬间又觉得自己的技术储备完全不够用。或者是在面试的时候发现自己没有任何的优势。虽然了解redis/memcached，ActiveMQ，nginx负载均衡等技术，但是了解这些技术就能让你有技术竞争力吗？掌握这些技术就足够你解决各种复杂系统中的高并发与高可用挑战吗？掌握这些技术在Java高阶职位的面试中，就能让你拥有属于自己的技术亮点吗？答案似乎都是否定的。

针对复杂的高并发、高可用相关的技术以及缓存架构，还有大型复杂的分布式系统，龙果学院独家发布的[《亿级流量电商详情页系统的大型高并发与高可用缓存架构实战》](http://www.roncoo.com/course/view/af7d501667fe4a19a60e9467e6d2b3d9" \t "_blank)视频教程中将会提供详细完整的方案供大家学习和应用。

本课程属于全网独家的大型Java高端架构项目实战课程，课程基于真实的每日上亿流量的大型电商网站中的商品详情页系统，作为项目实战。详细讲解如何实现一个复杂的缓存系统架构，去直接支撑电商背景下的高并发与高性能的访问，同时基于缓存架构本身所处的复杂分布式系统架构环境下，如何设计与实现一个高可用的分布式系统架构。期望通过本套课程能帮助大家学习到一些高阶的技术，复杂问题的解决方案，以及应对挑战性场景的大型架构设计思想。熟练掌握亿级流量电商网站的商品详情页架构如何设计与实现，能够应对各种复杂场景与挑战问题的缓存架构如何设计与实现，高阶的缓存架构以及解决方案如何应对各种棘手的高并发场景下的难题，复杂的缓存架构所处的分布式系统本身如何能够设计为一个高可用的分布式系统架构。

下面是本套课程讲解的核心技术要点。同时下面讲解的所有的架构、技术以及解决方案，在课程中，全部会采用大白话，通俗易懂的方式来讲解，同时上面的所有内容全部采用的纯手工敲代码的方式来实现，全部基于linux虚拟机搭建仿真环境来设计、开发、部署以及测试。以保证大家可以跟着课程学习以及动手练习，包括落地所有的技术以及解决方案。

1、亿级流量电商网站的商品详情页系统架构

面临难题：对于每天上亿流量，拥有上亿页面的大型电商网站来说，能够支撑高并发访问，同时能够秒级让最新模板生效的商品详情页系统的架构是如何设计的？

解决方案：异步多级缓存架构+nginx本地化缓存+动态模板渲染的架构

2、redis企业级集群架构

面临难题：如何让redis集群支撑几十万QPS高并发+99.99%高可用+TB级海量数据+企业级数据备份与恢复？

解决方案：redis的企业级备份恢复方案+复制架构+读写分离+哨兵架构+redis cluster集群部署

3、多级缓存架构设计

面临难题：如何将缓存架构设计的能够支撑高性能以及高并发到极致？同时还要给缓存架构最后的一个安全保护层？

解决方案：nginx抗热点数据+redis抗大规模离线请求+ehcache抗redis崩溃的三级缓存架构

4、数据库+缓存双写一致性解决方案

面临难题：高并发场景下，如何解决数据库与缓存双写的时候数据不一致的情况？

解决方案：异步队列串行化的数据库+缓存双写一致性解决方案

5、缓存维度化拆分解决方案

面临难题：如何解决大value缓存的全量更新效率低下问题？

解决方案：商品缓存数据的维度化拆分解决方案

6、缓存命中率提升解决方案

面临难题：如何将缓存命中率提升到极致？

解决方案：双层nginx部署架构+lua脚本实现一致性hash流量分发策略

7、缓存并发重建冲突解决方案

面临难题：如何解决高并发场景下，缓存重建时的分布式并发重建的冲突问题？

解决方案：基于zookeeper分布式锁的缓存并发重建冲突解决方案

8、缓存预热解决方案

面临难题：如何解决高并发场景下，缓存冷启动导致MySQL负载过高，甚至瞬间被打死的问题？

解决方案：基于storm实时统计热数据的分布式快速缓存预热解决方案

9、热点缓存自动降级方案

面临难题：如何解决热点缓存导致单机器负载瞬间超高？

解决方案：基于storm的实时热点发现+毫秒级的实时热点缓存负载均衡降级

10、高可用分布式系统架构设计

面临难题：如何解决分布式系统中的服务高可用问题？避免多层服务依赖因为少量故障导致系统崩溃？

解决方案：基于hystrix的高可用缓存服务，资源隔离+限流+降级+熔断+超时控制

11、复杂的高可用分布式系统架构设计

面临难题：如何针对复杂的分布式系统将其中的服务设计为高可用架构？

解决方案：基于hystrix的容错+多级降级+手动降级+生产环境参数优化经验+可视化运维与监控

12、缓存雪崩解决方案

面临难题：如何解决恐怖的缓存雪崩问题？避免给公司带来巨大的经济损失？

解决方案：全网独家的事前+事中+事后三层次完美缓存雪崩解决方案

13、缓存穿透解决方案

面临难题：如何解决高并发场景下的缓存穿透问题？避免给MySQL带来过大的压力？

解决方案：缓存穿透解决方案

14、缓存失效解决方案

面临难题：如何解决高并发场景下的缓存失效问题？避免给redis集群带来过大的压力？

解决方案：基于随机过期时间的缓存失效解决方案

课程大纲：

|  |  |
| --- | --- |
| 第01节 | 课程介绍以及高并发高可用复杂系统中的缓存架构有哪些东西？ |
| 第02节 | 基于大型电商网站中的商品详情页系统贯穿的授课思路介绍 |
| 第03节 | 小型电商网站的商品详情页的页面静态化架构以及其缺陷 |
| 第04节 | 大型电商网站的异步多级缓存构建+nginx数据本地化动态渲染的架构 |
| 第05节 | 能够支撑高并发+高可用+海量数据+备份恢复的redis的重要性 |
| 第06节 | 从零开始在虚拟机中一步一步搭建一个4个节点的CentOS集群 |
| 第07节 | 单机版redis的安装以及redis生产环境启动方案 |
| 第08节 | redis持久化机对于生产环境中的灾难恢复的意义 |
| 第09节 | 图解分析redis的RDB和AOF两种持久化机制的工作原理 |
| 第10节 | redis的RDB和AOF两种持久化机制的优劣势对比 |
| 第11节 | redis的RDB持久化配置以及数据恢复实验 |
| 第12节 | redis的AOF持久化深入讲解各种操作和相关实验 |
| 第13节 | 在项目中部署redis企业级数据备份方案以及各种踩坑的数据恢复容灾演练 |
| 第14节 | redis如何通过读写分离来承载读请求QPS超过10万+？ |
| 第15节 | redis replication以及master持久化对主从架构的安全意义 |
| 第16节 | redis主从复制原理、断点续传、无磁盘化复制、过期key处理 |
| 第17节 | redis replication的完整流运行程和原理的再次深入剖析 |
| 第18节 | 在项目中部署redis的读写分离架构（包含节点间认证口令） |
| 第19节 | 对项目的主从redis架构进行QPS压测以及水平扩容支撑更高QPS |
| 第20节 | redis主从架构下如何才能做到99.99%的高可用性？ |
| 第21节 | redis哨兵架构的相关基础知识的讲解 |
| 第22节 | redis哨兵主备切换的数据丢失问题：异步复制、集群脑裂 |
| 第23节 | redis哨兵的多个核心底层原理的深入解析（包含slave选举算法） |
| 第24节 | 在项目中以经典的3节点方式部署哨兵集群 |
| 第25节 | 对项目中的哨兵节点进行管理以及高可用redis集群的容灾演练 |
| 第26节 | redis如何在保持读写分离+高可用的架构下，还能横向扩容支撑1T+海量数据 |
| 第27节 | 数据分布算法：hash+一致性hash+redis cluster的hash slot |
| 第28节 | 在项目中重新搭建一套读写分离+高可用+多master的redis cluster集群 |
| 第29节 | 对项目的redis cluster实验多master写入、读写分离、高可用性 |
| 第30节 | redis cluster通过master水平扩容来支撑更高的读写吞吐+海量数据 |
| 第31节 | redis cluster的自动化slave迁移实现更强的高可用架构的部署方案 |
| 第32节 | redis cluster的核心原理分析：gossip通信、jedis smart定位、主备切换 |
| 第33节 | redis在实践中的一些常见问题以及优化思路（包含linux内核参数优化） |
| 第34节 | redis阶段性总结：1T以上海量数据+10万以上QPS高并发+99.99%高可用 |
| 第35节 | 亿级流量商品详情页的多级缓存架构以及架构中每一层的意义 |
| 第36节 | Cache Aside Pattern缓存+数据库读写模式的分析 |
| 第37节 | 高并发场景下的缓存+数据库双写不一致问题分析与解决方案设计 |
| 第38节 | 在linux虚拟机中安装部署MySQL数据库 |
| 第39节 | 库存服务的开发框架整合与搭建：spring boot+mybatis+jedis |
| 第40节 | 在库存服务中实现缓存与数据库双写一致性保障方案（一） |
| 第41节 | 在库存服务中实现缓存与数据库双写一致性保障方案（二） |
| 第42节 | 在库存服务中实现缓存与数据库双写一致性保障方案（三） |
| 第43节 | 在库存服务中实现缓存与数据库双写一致性保障方案（四） |
| 第44节 | 库存服务代码调试以及打印日志观察服务的运行流程是否正确 |
| 第45节 | 商品详情页结构分析、缓存全量更新问题以及缓存维度化解决方案 |
| 第46节 | 缓存数据生产服务的工作流程分析以及工程环境搭建 |
| 第47节 | 完成spring boot整合ehcache的搭建以支持服务本地堆缓存 |
| 第48节 | redis的LRU缓存清除算法讲解以及相关配置使用 |
| 第49节 | zookeeper+kafka集群的安装部署以及如何简单使用的介绍 |
| 第50节 | 基于kafka+ehcache+redis完成缓存数据生产服务的开发与测试 |
| 第51节 | 基于“分发层+应用层”双层nginx架构提升缓存命中率方案分析 |
| 第52节 | 基于OpenResty部署应用层nginx以及nginx+lua开发hello world |
| 第53节 | 部署分发层nginx以及基于lua完成基于商品id的定向流量分发策略 |
| 第54节 | 基于nginx+lua+java完成多级缓存架构的核心业务逻辑（一） |
| 第55节 | 基于nginx+lua+java完成多级缓存架构的核心业务逻辑（二） |
| 第56节 | 基于nginx+lua+java完成多级缓存架构的核心业务逻辑（三） |
| 第57节 | 分布式缓存重建并发冲突问题以及zookeeper分布式锁解决方案 |
| 第58节 | 缓存数据生产服务中的zk分布式锁解决方案的代码实现（一） |
| 第59节 | 缓存数据生产服务中的zk分布式锁解决方案的代码实现（二） |
| 第60节 | 缓存数据生产服务中的zk分布式锁解决方案的代码实现（三） |
| 第61节 | Java程序员、缓存架构以及Storm大数据实时计算之间的关系 |
| 第62节 | 讲给Java工程师的史上最通俗易懂Storm教程：大白话介绍 |
| 第63节 | 讲给Java工程师的史上最通俗易懂Storm教程：大白话讲集群架构与核心概念 |
| 第64节 | 讲给Java工程师的史上最通俗易懂Storm教程：大白话讲并行度和流分组 |
| 第65节 | 讲给Java工程师的史上最通俗易懂Storm教程：纯手敲WordCount程序 |
| 第66节 | 讲给Java工程师的史上最通俗易懂Storm教程：纯手工集群部署 |
| 第67节 | 讲给Java工程师的史上最通俗易懂Storm教程：基于集群运行计算拓扑 |
| 第68节 | 缓存冷启动问题：新系统上线、redis彻底崩溃导致数据无法恢复 |
| 第69节 | 缓存预热解决方案：基于storm实时热点统计的分布式并行缓存预热 |
| 第70节 | 基于nginx+lua完成商品详情页访问流量实时上报kafka的开发 |
| 第71节 | 基于storm+kafka完成商品访问次数实时统计拓扑的开发 |
| 第72节 | 基于storm完成LRUMap中topn热门商品列表的算法讲解与编写 |
| 第73节 | 基于storm+zookeeper完成热门商品列表的分段存储 |
| 第74节 | 基于双重zookeeper分布式锁完成分布式并行缓存预热的代码开发 |
| 第75节 | 将缓存预热解决方案的代码运行后观察效果以及调试和修复所有的bug |
| 第76节 | 热点缓存问题：促销抢购时的超级热门商品可能导致系统全盘崩溃的场景 |
| 第77节 | 基于nginx+lua+storm的热点缓存的流量分发策略自动降级解决方案 |
| 第78节 | 在storm拓扑中加入热点缓存实时自动识别和感知的代码逻辑 |
| 第79节 | 在storm拓扑中加入nginx反向推送缓存热点与缓存数据的代码逻辑 |
| 第80节 | 在流量分发+后端应用双层nginx中加入接收热点缓存数据的接口 |
| 第81节 | 在nginx+lua中实现热点缓存自动降级为负载均衡流量分发策略的逻辑 |
| 第82节 | 在storm拓扑中加入热点缓存消失的实时自动识别和感知的代码逻辑 |
| 第83节 | 将热点缓存自动降级解决方案的代码运行后观察效果以及调试和修复bug |
| 第84节 | hystrix与高可用系统架构：资源隔离+限流+熔断+降级+运维监控 |
| 第85节 | hystrix要解决的分布式系统可用性问题以及其设计原则 |
| 第86节 | 电商网站的商品详情页缓存服务业务背景以及框架结构说明 |
| 第87节 | 基于spring boot快速构建缓存服务以及商品服务 |
| 第88节 | 快速完成缓存服务接收数据变更消息以及调用商品服务接口的代码编写 |
| 第89节 | 商品服务接口故障导致的高并发访问耗尽缓存服务资源的场景分析 |
| 第90节 | 基于hystrix的线程池隔离技术进行商品服务接口的资源隔离 |
| 第91节 | 基于hystrix的信号量技术对地理位置获取逻辑进行资源隔离与限流 |
| 第92节 | hystrix的线程池+服务+接口划分以及资源池的容量大小控制 |
| 第93节 | 深入分析hystrix执行时的8大流程步骤以及内部原理 |
| 第94节 | 基于request cache请求缓存技术优化批量商品数据查询接口 |
| 第95节 | 开发品牌名称获取接口的基于本地缓存的fallback降级机制 |
| 第96节 | 深入理解hystrix的短路器执行原理以及模拟接口异常时的短路实验 |
| 第97节 | 深入理解线程池隔离技术的设计原则以及动手实战接口限流实验 |
| 第98节 | 基于timeout机制来为商品服务接口的调用超时提供安全保护 |
| 第99节 | 基于hystrix的高可用分布式系统架构项目实战课程的总结 |
| 第100节 | 基于request collapser请求合并技术进一步优化批量查询 |
| 第101节 | hystirx的fail-fast与fail-silient两种最基础的容错模式 |
| 第102节 | 为商品服务接口调用增加stubbed fallback降级机制 |
| 第103节 | 基于双层嵌套command开发商品服务接口的多级降级机制 |
| 第104节 | 基于facade command开发商品服务接口的手动降级机制 |
| 第105节 | 生产环境中的线程池大小以及timeout超时时长优化经验总结 |
| 第106节 | 生产环境中的线程池自动扩容与缩容的动态资源分配经验 |
| 第107节 | hystrix的metric统计相关的各种高阶配置讲解 |
| 第108节 | hystrix dashboard可视化分布式系统监控环境部署 |
| 第109节 | 生产环境中的hystrix分布式系统的工程运维经验总结 |
| 第110节 | 高并发场景下恐怖的缓存雪崩现象以及导致系统全盘崩溃的后果 |
| 第111节 | 缓存雪崩的基于事前+事中+事后三个层次的完美解决方案 |
| 第112节 | 基于hystrix完成对redis访问的资源隔离以避免缓存服务被拖垮 |
| 第113节 | 为redis集群崩溃时的访问失败增加fail silent容错机制 |
| 第114节 | 位redis集群崩溃时的场景部署定制化的熔断策略 |
| 第115节 | 基于hystrix限流完成源服务的过载保护以避免流量洪峰打死MySQL |
| 第116节 | 为源头服务的限流场景增加stubbed fallback降级机制 |
| 第117节 | 高并发场景下的缓存穿透导致MySQL压力倍增问题以及其解决方案 |
| 第118节 | 在缓存服务中开发缓存穿透的保护性机制以及代码测试 |
| 第119节 | 高并发场景下的nginx缓存失效导致redis压力倍增问题以及解决方案 |
| 第120节 | 在nginx lua脚本中开发缓存失效的保护性机制以及代码测试 |
| 第121节 | 支撑高并发与高可用的大型电商详情页系统的缓存架构课程总结 |
| 第122节 | 如何将课程中的东西学以致用在自己目前的项目中去应用？ |
| 第123节 | 如何带着课程中讲解的东西化为自己的技术并找一份更好的工作？ |

课程观看地址：<http://www.roncoo.com/course/view/af7d501667fe4a19a60e9467e6d2b3d9>