

开篇词 | 怎样才能做好性能调优？

2019-05-20 刘超



你好，我是刘超，目前在金山西山居担任技术经理，很高兴通过这个专栏，跟你一起讨论 **Java** 性能调优这个话题。

先给你讲个故事吧。多年前我加入了一家大型互联网公司，刚进入就以 **996** 标准，参与新品研发。公司业务发展急需互联网产品，因此我们的时间很紧张，**4** 个月新产品就上线了。

开始还算顺利，但不久后的一天晚上，系统突然就瘫痪了，重启之后，问题仍然无规律地重现。当时运维同事马上写了一个重启脚本，定时排队重启各个服务，但也只能做到“治标不治本”。

作为主力开发，我和公司的系统架构师一起排查问题。架构师轻车熟路地通过各种 **Linux** 命令在线上环境查看性能指标，也 **dump** 出日志文件，走查代码，最后定位到了问题，后面就是分析原因、制定解决方案、更新版本等一系列操作。那是我第一次深刻感受到性能调优的重要性。

后来的几年里，我又陆续参与过物流、电商、游戏支付系统的研发，这些项目都存在一个共性，就是经常会运营一些大促以及抢购类活动。活动期间，系统不仅要保证处理请求业务的严谨性，还要历经短时间内高并发的考验。我也一直处于性能调优的一线。

正所谓“实践出真知”。这些年在生产环境中遇到的事故不少，很多坑一点点踩平，就走出了一条路，这个过程中我收获了很多实打实的经验，希望能分享给更多的人，我们一起学习、交流和探讨。

关于性能调优，我先来说说的我的感受。**Java** 性能调优不像是学一门编程语言，无法通过直线

式的思维来掌握和应用，它对于工程师的技术广度和深度都有着较高的要求。

互联网时代，一个简单的系统就囊括了应用程序、数据库、容器、操作系统、网络等技术，线上一旦出现性能问题，就可能要你协调多方面组件去进行优化，这就是技术广度；而很多性能问题呢，又隐藏得很深，可能因为一个小小的代码，也可能因为线程池的类型选择错误..可归根结底考验的还是我们对这项技术的了解程度，这就是技术深度。

显然，性能调优不是一件容易的事。但有没有什么方法能把这件事情做好呢？接下来跟你分享几点我的心得。

1.扎实的计算机基础

我们调优的对象不是单一的应用服务，而是错综复杂的系统。应用服务的性能可能与操作系统、网络、数据库等组件相关，所以我们需要储备计算机组成原理、操作系统、网络协议以及数据库等基础知识。具体的性能问题往往还与传输、计算、存储数据等相关，那我们还需要储备数据结构、算法以及数学等基础知识。

2.习惯透过源码了解技术本质

我身边有很多好学的同学，他们经常和我分享在一些技术论坛或者公众号上学到的技术。这个方式很好，因为论坛上边的大部分内容，都是生产者自己吸收消化后总结的知识点，能帮助我们快速获取、快速理解。但是只做到这个程度还不够，因为你缺失了自己的判断。怎么办呢？我们需要深入源码，通过分析来学习、总结一项技术的实现原理和优缺点，这样我们就能更客观地去学习一项技术，还能透过源码来学习牛人的思维方式，收获更好的编码实现方式。

3.善于追问和总结

很多同学在使用一项技术时，只是因为这项技术好用就用了，从来不问自己：为什么这项技术可以提升系统性能？对比其他技术它好在哪儿？实现的原理又是什么呢？事实上，“知其然且知所以然”才是我们积累经验的关键。知道了一项技术背后的实现原理，我们才能在遇到性能问题时，做到触类旁通。

综合这三点心得，我也就想清楚了该怎么交付这个专栏。

在这个专栏里，我将从实战出发，精选高频性能问题，透过 **Java** 底层源码，提炼出优化思路 and 它背后的实现原理，最后形成一套“学完就能用的调优方法论”。这也是很多一线大厂对于高级工程师的要求，希望通过这个专栏帮助你快速进阶。

那这个专栏具体是怎么设计的呢？结合 **Java** 应用开发的知识点，我将内容分为七大模块，从上到下依次详解 **Java** 应用服务的每一层优化实战。

模块一，概述。 为你建立两个标准。一个是性能调优标准，告诉你可以通过哪些参数去衡量系

统性能；另一个是调优过程标准，带你了解通过哪些严格的调优策略，我们可以排查性能问题，从而解决问题。

模块二，Java 编程性能调优。JDK 是 Java 语言的基础库，熟悉 JDK 中各个包中的工具类，可以帮助你编写出高性能代码。这里我会从基础的数据类型讲起，涉及容器在实际应用场景中的调优，还有现在互联网系统架构中比较重要的网络通信调优。

模块三，多线程性能调优。目前大部分服务器都是多核处理器，多线程编程的应用广泛。为了保证线程的安全性，通常会用到同步锁，这会为系统埋下很多隐患；除此之外，还有多线程高并发带来的性能问题，这些都会在这个模块重点讲解。

模块四，JVM 性能监测及调优。Java 应用程序是运行在 JVM 之上的，对 JVM 进行调优可以提升系统性能。这里重点讲解 Java 对象的创建和回收、内存分配等。

模块五，设计模式调优。在架构设计中，我们经常会用到一些设计模式来优化架构设计。这里我将结合一些复杂的应用场景，分享设计优化案例。

模块六，数据库性能调优。数据库最容易成为整个系统的性能瓶颈，这里我会重点解析一些数据库的常用调优方法。

模块七，实战演练场。以上六个模块的内容，都是基于某个点的调优，现在是时候把你前面所学都调动起来了，这里我将带你进入综合性能问题高频出现的应用场景，学习整体调优方法。

纵观整个专栏，以点带面，目的就是交付给你一套“学完就用的调优方法论”，让你在遇到性能问题时，能够调动所学，触类旁通，技术水平更上一层楼。

“纸上得来终觉浅，绝知此事要躬行”。这个专栏从实战中来，也希望你能应用到实战中去，最终在学习的基础上，融会贯通，补充完善，形成属于你自己的一套调优方法论。

今天的内容就到这里，期待在下一讲，和你共同开启一段新的旅程。你可以在留言区做个简单的自我介绍，聊聊你目前的工作、学习情况，以及在 Java 性能调优上的痛点，我们认识一下，也方便我后续有针对性地为你讲解。

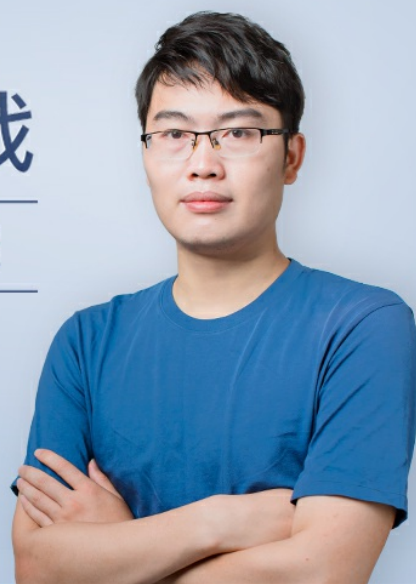
最后，感谢你的信任，我定会全力以赴，和你一起实现完美交付。

Java 性能调优实战

覆盖 80% 以上 Java 应用调优场景

刘超

金山软件西山居技术经理



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

精选留言



业余草

7

定位痛点，难定位，其实是不会定位，难重现。分析问题，不会分析，不知道怎么分析，不知道各数据代表什么等。解决问题，解决这个问题，可能引出另外一个问题等。

2019-05-20

编辑回复

期待同学能在专栏中找到这些问题的答案～

2019-05-20



luciferin

6

跟着大佬有肉吃

2019-05-20



年年

6

学完这个再等着波姐的翻译的《Java性能优化实践》出版，到时候两相印证美滋滋

2019-05-20



Jimmy(陈基明)®

4

挺好的，期待

2019-05-20



北纬91度

2

老师您好，我是做Android开发的，Java后台服务开发只是最近项目接触到一些，用的是Spring boot, springcloud，对于性能调优这块，首先我们的项目用户量很少，针对一些厂家定制项目，用户量小到出了问题可以停下来让你解决的地步，而且我也是刚接触后端这块，本身后端的很多开发知识要学，现在看到这个感觉很有用，也一边学开发一边学调优吧，也许能从摇篮中扼杀bug，对于我这种情况，老师您在学习的过程中有什么建议吗？

2019-05-22

作者回复

我总结下我过往的经历，我觉得打好基础功底，多动手实践是最重要的，其他的没有特别的建议。希望能帮助到你。

2019-06-03



何何何何何少侠

2

希望老师后面的课程能够更多的注重于实际业务场景而不只是理论

2019-05-20

作者回复

少侠你好，本次专栏第一个模块主要是给大家熟悉性能调优的一些指标和策略，后面的模块中，每一讲都是通过实际场景来讲解具体的调优。

2019-05-21



言凡

2

老师你好，我的项目上用户量少，对性能要求也不高，很难遇到性能调优的大场面。缺少实践的话，通过学习也不会有很深的感触。在这种情况下应该如何提高自己的性能调优能力呢？

2019-05-20

作者回复

言凡你好，你提的这个问题非常好，我相信很多童鞋都有跟你一样的问题。

具体的性能调优能力确实需要一些实战历练，但在具体的后面，是我们平时积累的大量基础知识。所以说，首先要保证练好扎实的基础功，到了真实战场，这些基础能帮上你的大忙。很多大公司面试的时候，对中高级开发首先要求的也是基础。

实战经验虽然是我们短板，但平时我们也可以在很多源码上学习一些调优经验，例如锁的优化的方式，减小锁粒度是优化锁常用的方式，我们可以学习和借鉴使用，像这样的优化案例有很多。也可以通过一些渠道学习大公司的优化方案以及大神的分享优化方案，强调的是学习一种思维方式，不一定能在实际项目中应用到，但是我们能在遇到问题的时候想到优化方案。

还有就是通过动手来实践，提高自己的实践能力。

2019-05-21



猫头鹰爱拿铁

2

对性能调优所说的要扎实的计算机基础实在是太认同了。之前也做过类似的事，不过涉及的不复杂，是个单体应用，在开始性能调优的时候首先要对应用的性能进行监控获知瓶颈在哪，这就涉及到磁盘、网络、内存、cpu以及数据库不仅仅是jvm这块。然后再从这些数据里面抽丝剥

茧等去找问题。

2019-05-20

编辑回复

英雄所见略同~[]

2019-05-20



SlamDunk

1

做业务开发，经历过的性能优化最蛋疼的两种:1.压力测试过程中发现内存泄露。2.业务代码质量差稍显混乱，服务响应时间太长，优化时还要小心翼翼保持原业务逻辑。

2019-06-07



胖妞

1

平常开发的时候压根就没有进行过系统调优，平常开发只要没bug就感觉完事，可是，这样的开发总感觉停留在表象！而且，日常开发也基本没有用过jdk自带的工具进行bug处理，linux系统下开发部署，只要满足不报错就行！遇到服务器报broken pipe都不知道咋处理！希望这个课程能给我一些收获！

2019-05-22



建国

1

老师好，我是跨专业到码农，已经从事JAVA开发两年半，我们公司是做培训的系统，客户大规模使用前都会报备，平时不需要多少的并发量，需要的时候做系统升级，每次做升级都需要压力测试，但是现在服务器的内存达到了瓶颈，就是刚重启完系统基本都是70%，运行几天之后基本都是85%，所以每周都晚上重启两次（遇到上线就不需要了），所以很难受，希望能学到点内存优化的方案策略，期待期待

2019-05-22

作者回复

你好 建国，看来咱们的服务存在性能问题呀，希望这个专栏能帮助你快速上手性能问题分析和调优。

2019-05-22



liu

1

一个亟须扎实根底的专栏

2019-05-21



不靠谱~

1

每次遇到性能调优就不知所措了，希望能跟随老师入门。

2019-05-21

编辑回复

加油！

2019-05-21



jack

1

性能调优涉及到对技术掌握的广度与深度。学习大佬的经验与方法，加油 ^ 0 ^ ~!

2019-05-20



NARUTO

期待

👍 0

2019-06-08



到道可道

👍 0

看到大佬的介绍，都是痛点呀，希望能跟着大佬好好学，进阶提升自己

2019-06-01



李青

👍 0

首先，感谢老师的分享，使我受益匪浅。

其实我在这想回答@业余草童靴在留言中的困惑

个人认为性能调优分为以下步骤：

1.目前现象----》2.提出猜想-----》3.验证猜想-----》4.定位到问题-----》5.解决问题

分析问题难其实就对应的是第2步骤，说白了就是你提不猜想，为什么你提出猜想，那是因为你的知识面不广，基础知识不牢固。知识面不够，就导致针对现象，提不出问题。

定位问题难对应第4步骤，这个的问题就是不会使用工具，我们在佐证我们的猜想时需要一些辅助工具。不会使用工具，就导致不能佐证猜想。

2019-05-31

作者回复

阐述的很好，点赞！

2019-05-31



半小时的天气

👍 0

目前遇到的问题是数据库查询很慢呀明明数据不多

2019-05-30

作者回复

有没有建立索引呢？如果建立了，查询时有没有使用到索引呢？

2019-05-31



ABC

👍 0

开始打卡，希望跟着这个专栏能学到更多有用的知识。

2019-05-29



-W.LI-

👍 0

老师好!我现在的情况是技术深度和广度都不够，之前也尝试过看一些底层源码。可是看起来很费劲，一头扎在实现里，没法看见全貌。单独一行代码看的懂，连一起整个类的功能就想不太明白了，更搞不清为啥这么设计了。我现在的策略是先抓广度，比较好的书和课程我都看起来。买一些课程跟着老师们学(好些理解不了的，能背的我就先背下来，背不下来的就只好放过了先)，等学完一些课程以后又开始尝试看源码这样会不会好一些？

2019-05-26

作者回复

这个想法很好！

2019-05-26