01 | 如何制定性能调优标准？

2019-05-21 刘超

你好，我是刘超。

我有一个朋友，有一次他跟我说，他们公司的系统从来没有经过性能调优，功能测试完成后就上线了，线上也没有出现过什么性能问题呀，那为什么很多系统都要去做性能调优呢？

当时我就回答了他一句，如果你们公司做的是 12306 网站，不做系统性能优化就上线，试试看会是什么情况。

如果是你，你会怎么回答呢？今天，我们就从这个话题聊起，希望能跟你一起弄明白这几个问题：我们为什么要做性能调优？什么时候开始做？做性能调优是不是有标准可参考？

为什么要做性能调优？

一款线上产品如果没有经过性能测试，那它就好比是一颗定时炸弹，你不知道它什么时候会出现问题，你也不清楚它能承受的极限在哪儿。

有些性能问题是时间累积慢慢产生的，到了一定时间自然就爆炸了；而更多的性能问题是由访问量的波动导致的，例如，活动或者公司产品用户量上升；当然也有可能是一款产品上线后就半死不活，一直没有大访问量，所以还没有引发这颗定时炸弹。

现在假设你的系统要做一次活动，产品经理或者老板告诉你预计有几十万的用户访问量，询问系统能否承受得住这次活动的压力。如果你不清楚自己系统的性能情况，也只能战战兢兢地回答老

板，有可能大概没问题吧。

所以，要不要做性能调优，这个问题其实很好回答。所有的系统在开发完之后，多多少少都会有性能问题，我们首先要做的就是想办法把问题暴露出来，例如进行压力测试、模拟可能的操作场景等等，再通过性能调优去解决这些问题。

比如，当你在用某一款 App 查询某一条信息时，需要等待十几秒钟；在抢购活动中，无法进入活动页面等等。你看，系统响应就是体现系统性能最直接的一个参考因素。

那如果系统在线上没有出现响应问题，我们是不是就不用去做性能优化了呢？再给你讲一个故事吧。

曾经我的前前东家系统研发部门来了一位大神，为什么叫他大神，因为在他来公司的一年时间里，他只做了一件事情，就是把服务器的数量缩减到了原来的一半，系统的性能指标，反而还提升了。

好的系统性能调优不仅仅可以提高系统的性能，还能为公司节省资源。这也是我们做性能调优的最直接的目的。

什么时候开始介入调优？

解决了为什么要做性能优化的问题，那么新的问题就来了：如果需要对系统做一次全面的性能监测和优化，我们从什么时候开始介入性能调优呢？是不是越早介入越好？

其实，在项目开发的初期，我们没有必要过于在意性能优化，这样反而会让我们疲于性能优化，不仅不会给系统性能带来提升，还会影响到开发进度，甚至获得相反的效果，给系统带来新的问题。

我们只需要在代码层面保证有效的编码，比如，减少磁盘 I/O 操作、降低竞争锁的使用以及使用高效的算法等等。遇到比较复杂的业务，我们可以充分利用设计模式来优化业务代码。例如，设计商品价格的时候，往往会有很多折扣活动、红包活动，我们可以用装饰模式去设计这个业务。

在系统编码完成之后，我们就可以对系统进行性能测试了。这时候，产品经理一般会提供线上预期数据，我们在提供的参考平台上进行压测，通过性能分析、统计工具来统计各项性能指标，看是否在预期范围之内。

在项目成功上线后，我们还需要根据线上的实际情况，依照日志监控以及性能统计日志，来观测系统性能问题，一旦发现问题，就要对日志进行分析并及时修复问题。

有哪些参考因素可以体现系统的性能？

上面我们讲到了在项目研发的各个阶段性能调优是如何介入的，其中多次讲到了性能指标，那么性能指标到底有哪些呢？

在我们了解性能指标之前，我们先来了解下哪些计算机资源会成为系统的性能瓶颈。

CPU：有的应用需要大量计算，他们会长时间、不间断地占用 CPU 资源，导致其他资源无法争夺到 CPU 而响应缓慢，从而带来系统性能问题。例如，代码递归导致的无限循环，正则表达式引起的回溯，JVM频繁的 FULL GC，以及多线程编程造成的大量上下文切换等，这些都有可能导致 CPU 资源繁忙。

内存：Java 程序一般通过 JVM 对内存进行分配管理，主要是用 JVM 中的堆内存来存储 Java 创建的对象。系统堆内存的读写速度非常快，所以基本不存在读写性能瓶颈。但是由于内存成本要比磁盘高，相比磁盘，内存的存储空间又非常有限。所以当内存空间被占满，对象无法回收时，就会导致内存溢出、内存泄露等问题。

磁盘I/O：磁盘相比内存来说，存储空间要大很多，但磁盘 I/O 读写的速度要比内存慢，虽然目前引入的 SSD 固态硬盘已经有所优化，但仍然无法与内存的读写速度相提并论。

网络：网络对于系统性能来说，也起着至关重要的作用。如果你购买过云服务，一定经历过，选择网络带宽大小这一环节。带宽过低的话，对于传输数据比较大，或者是并发量比较大的系统，网络就很容易成为性能瓶颈。

异常：Java 应用中，抛出异常需要构建异常栈，对异常进行捕获和处理，这个过程非常消耗系统性能。如果在高并发的情况下引发异常，持续地进行异常处理，那么系统的性能就会明显地受到影响。

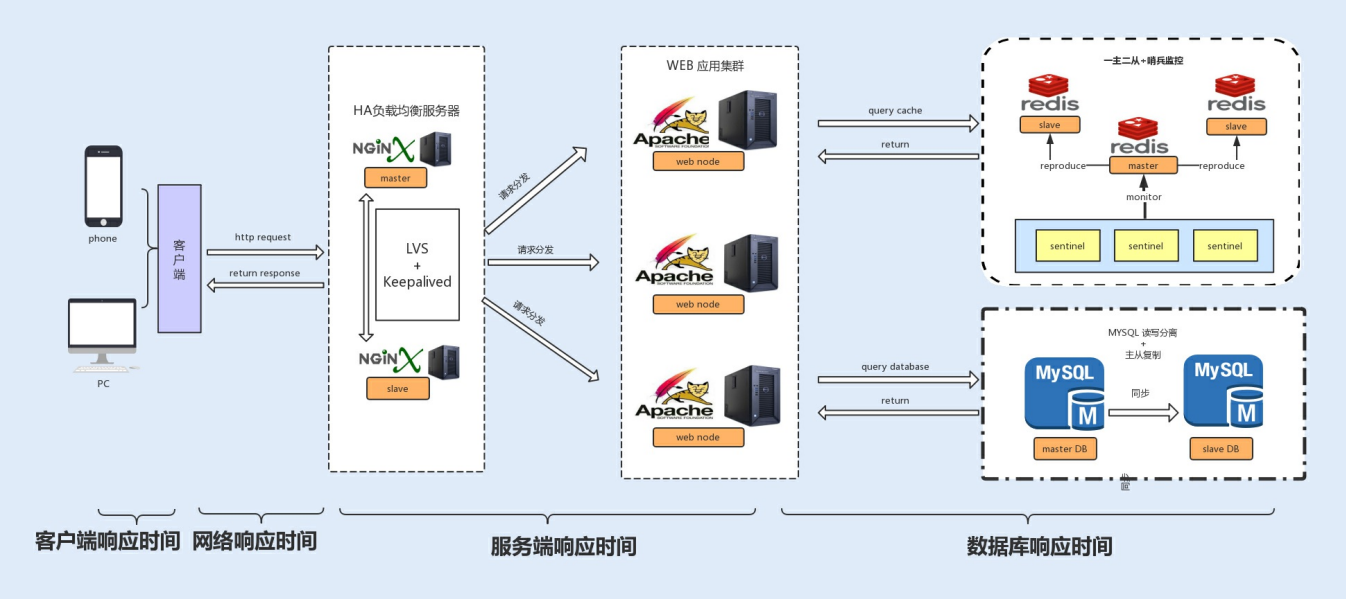
数据库：大部分系统都会用到数据库，而数据库的操作往往是涉及到磁盘 I/O 的读写。大量的数据库读写操作，会导致磁盘 I/O 性能瓶颈，进而导致数据库操作的延迟性。对于有大量数据库读写操作的系统来说，数据库的性能优化是整个系统的核心。

锁竞争：在并发编程中，我们经常会需要多个线程，共享读写操作同一个资源，这个时候为了保持数据的原子性（即保证这个共享资源在一个线程写的时候，不被另一个线程修改），我们就会用到锁。锁的使用可能会带来上下文切换，从而给系统带来性能开销。JDK1.6 之后，Java 为了降低锁竞争带来的上下文切换，对 JVM 内部锁已经做了多次优化，例如，新增了偏向锁、自旋锁、轻量级锁、锁粗化、锁消除等。而如何合理地使用锁资源，优化锁资源，就需要你了解更多的操作系统知识、Java 多线程编程基础，积累项目经验，并结合实际场景去处理相关问题。

了解了上面这些基本内容，我们可以得到下面几个指标，来衡量一般系统的性能。

响应时间

响应时间是衡量系统性能的重要指标之一，响应时间越短，性能越好，一般一个接口的响应时间是在毫秒级。在系统中，我们可以把响应时间自下而上细分为以下几种：



- 数据库响应时间：数据库操作所消耗的时间，往往是整个请求链中最耗时的；
- 服务端响应时间：服务端包括 Nginx 分发的请求所消耗的时间以及服务端程序执行所消耗的时间；
- 网络响应时间：这是网络传输时，网络硬件需要对传输的请求进行解析等操作所消耗的时间；
- 客户端响应时间：对于普通的 Web、App 客户端来说，消耗时间是可以忽略不计的，但如果你的客户端嵌入了大量的逻辑处理，消耗的时间就有可能变长，从而成为系统的瓶颈。

吞吐量

在测试中，我们往往会比较注重系统接口的 TPS（每秒事务处理量），因为 TPS 体现了接口的性能，TPS 越大，性能越好。在系统中，我们也可以把吞吐量自下而上地分为两种：磁盘吞吐量和网络吞吐量。

我们先来看**磁盘吞吐量**，磁盘性能有两个关键衡量指标。

一种是 IOPS（Input/Output Per Second），即每秒的输入输出量（或读写次数），这种是指单位时间内系统能处理的 I/O 请求数量，I/O 请求通常为读或写数据操作请求，关注的是随机读写性能。适应于随机读写频繁的应用，如小文件存储（图片）、OLTP 数据库、邮件服务器。

另一种是数据吞吐量，这种是指单位时间内可以成功传输的数据量。对于大量顺序读写频繁的应用，传输大量连续数据，例如，电视台的视频编辑、视频点播 VOD（Video On Demand），数据吞吐量则是关键衡量指标。

接下来看**网络吞吐量**，这个是指网络传输时没有帧丢失的情况下，设备能够接受的最大数据速率。网络吞吐量不仅仅跟带宽有关系，还跟 CPU 的处理能力、网卡、防火墙、外部接口以及 I/O 等紧密关联。而吞吐量的大小主要由网卡的处理能力、内部程序算法以及带宽大小决定。

计算机资源分配使用率

通常由 CPU 占用率、内存使用率、磁盘 I/O、网络 I/O 来表示资源使用率。这几个参数好比一个木桶，如果其中任何一块木板出现短板，任何一项分配不合理，对整个系统性能的影响都是毁灭性的。

负载承受能力

当系统压力上升时，你可以观察，系统响应时间的上升曲线是否平缓。这项指标能直观地反馈给你，系统所能承受的负载压力极限。例如，当你对系统进行压测时，系统的响应时间会随着系统并发数的增加而延长，直到系统无法处理这么多请求，抛出大量错误时，就到了极限。

总结

通过今天的学习，我们知道性能调优可以使系统稳定，用户体验更佳，甚至在比较大的系统中，还能帮公司节约资源。

但是在项目的开始阶段，我们没有必要过早地介入性能优化，只需在编码的时候保证其优秀、高效，以及良好的程序设计。

在完成项目后，我们就可以进行系统测试了，我们可以将以下性能指标，作为性能调优的标准，响应时间、吞吐量、计算机资源分配使用率、负载承受能力。

回顾我自己的项目经验，有电商系统、支付系统以及游戏充值计费系统，用户级都是千万级别，且要承受各种大型抢购活动，所以我对系统的性能要求非常苛刻。除了通过观察以上指标来确定系统性能的好坏，还需要在更新迭代中，充分保障系统的稳定性。

这里，**给你延伸一个方法，**就是将迭代之前版本的系统性能指标作为参考标准，通过自动化性能测试，校验迭代发版之后的系统性能是否出现异常，这里就不仅仅是比较吞吐量、响应时间、负载能力等直接指标了，还需要比较系统资源的 CPU 占用率、内存使用率、磁盘 I/O、网络 I/O 等几项间接指标的变化。

思考题

除了以上这些常见的性能参考指标，你是否还能想到其他可以衡量系统性能的指标呢？

期待在留言区看到你的见解。也欢迎你点击“请朋友读”，把今天的内容分享给身边的朋友，邀请他一起讨论。

精选留言

杨军

12

首先很高兴终于有人开Java性能优化的课程，我在工作中需要接触很多这方面的工作，希望通过学习这次课程能带来更多收获。
然后我想请教老师一个问题，CPU利用率和系统负载这两个指标之间是什么关系？网上很多资料讲的不清不楚，看不明白。

2019-05-21

作者回复

杨军你好，系统负载代表单位时间内正在运行或等待的进程或线程数，代表了系统的繁忙程度，CPU利用率则代表单位时间内一个线程或进程实时占用CPU的百分比。我们知道，一个进程或者线程在运行时，未必都在实时的利用CPU的。

比如，在CPU密集型的情况下，系统的负载未必会高，但CPU的利用率肯定会高，一个线程/进程一直在计算，它对CPU的实时利用率是100%，而系统负载是0.1;

又比如，而对于I/O密集型的程序来说，有可能CPU的利用率不高，但系统的负载却会非常高，这是因为I/O经常引起阻塞，这样导致很多线程/进程被处于阻塞等待状态，处于等待的线程或进程也是属于负载线程/进程的。

通过以上两个例子，不知道有没有让你分清楚两个指标的区别，有问题保持沟通。

2019-05-21

QQ怪 11

我觉得还有接口返回200的成功率吧

2019-05-20

作者回复

晚上好 QQ怪，你说的很对。我们平时在使用AB进行压测时，会有一个failed requests指标，本身接口没有异常的情况下，压测出现了异常，也是说明这个接口有性能问题。还有就是percentage of the requests served within a certain time这个指标，这个指标对金融交易系统来说是非常重要的，如果有99%的请求是1ms返回，但有1%是500ms返回，这对于某些对交易时间要求极致的金融系统来说也是性能问题。感谢你的回答，期望我的回答能让你满意！

2019-05-20

遇见阳光 7

老师，tps qps这块还是有点不太清楚。

2019-05-21

作者回复

遇见阳光 你好，TPS(transaction per second)是单位时间内处理事务的数量，QPS(query per second)是单位时间内请求的数量。TPS代表一个事务的处理，可以包含了多次请求。很多公司用QPS作为接口吞吐量的指标，也有很多公司使用TPS作为标准，两者都能表现出系统的吞吐量的大小，TPS的一次事务代表一次用户操作到服务器返回结果，QPS的一次请求代表一个接口的一次请求到服务器返回结果。当一次用户操作只包含一个请求接口时，TPS和QPS没有区别。当用户的一次操作包含了多个服务请求时，这个时候TPS作为这次用户操作的性能指标就更具有代表性了。

2019-05-22

业余草 5

优化，首先是你要知道它是什么，用了什么原理，才能更深入的去做好优化。其实最重要的优化，是优化自己的编程思想。首先要排除的是，每个需求，不是功能做完了就做好了。我们要做的是把知识变成钱，而不是把劳动力变成钱，

2019-05-21

小辉辉 3

现在的项目也是没有经过压测和调优的，上线也没多久，正好跟着老师的专栏来试着优化一下性能

2019-05-20

编辑回复

为你手动点赞

2019-05-20

Maxwell 2

请问老师最近段时间遇到端口被CLOSE_WAIT占用，重启后过了半天又重现，以前没有出现过，一般如何排查

2019-06-01

作者回复

可以通过tcpdump抓包看看连接状态，分析是否是服务端的FIN packet没有发出去。

正常的关闭流程是：服务端在接收到客户端发送的关闭请求FIN后，会进入CLOSE_WAIT状态，同时发送ACK回去。在完成与客户端直接的通信操作之后，再向客户端发送FIN，进入LAST_ACK状态。

如果连接是CLOSE_WAIT状态，而不是LAST_ACK状态，说明还没有发FIN给Client，那么可能是在关闭连接之前还有许多数据要发送或者其他事要做，导致没有发这个FIN packet。

建议确定关闭请求的四次握手，哪个环节出了问题，再去排查业务代码，可能是由于超时或者异常导致没有正常关闭连接。

2019-06-02

诸葛 2

异常：Java 应用中，抛出异常需要构建异常栈，对异常进行捕获和处理，这个过程非常消耗系统性能。如果在高并发的情况下引发异常，持续地进行异常处理，那么系统的性能就会明显地受到影响。
老师好，高并发调用方法之后第一步就是检验一堆入参，如果入参有问题立即抛出异常给前端，不再处理下边的逻辑了，这样有问题吗？我看系统性能还可以啊。压力测试也还行。如果有问题的话那应该怎么做呢，检验错误后给前端放回code码然后返回空对象吗？感谢

2019-05-26

作者回复

如果没有生成堆栈追踪信息，不会有性能问题。一般业务异常避免生成堆栈追踪信息，我们知道这个异常是什么原因，所以直接返回字符串就好了。而系统异常，一般都会生成堆栈追踪信息，以便追踪源头，更好的排查问题。

2019-05-26

随遇而安 2

希望这次能学到真正的东西

2019-05-20

编辑回复

坚持就会有收获，加油！

2019-05-20

我行我素 2

并发用户数，高可用可扩展等方面

2019-05-20

作者回复

你好 我行我素，感谢你的回答。并发用户数代表系统同一时间处理事务的并发能力，也是体现

系统性能的一个直接性能指标。当然，TPS也能间接的体现系统并发处理能力。

2019-05-21

Phoenix 2

想请教下老师，使用MySQL经常会遇到业务需要实时导出大量业务数据的需求，那么如何在不影响业务和不分库的的情况满足业务实时导出大量数据的需求呢？

2019-05-20

作者回复

你好 Phoenix，切忌在主库中操作这种报表类的导出，在写入和查询都在一个主库进行，会造成数据库性能瓶颈，严重的会导致数据库死锁。我们可以将数据库读写分离，写业务走主(写)库，导出数据可以从从(读)库导出。这种实现方式，首先能提高数据导出的性能，其次不影响写业务。

如果你们公司有大数据中心，可以考虑将需要导出的数据实时同步到大数据中心，通过实时的流计算处理生成不同需求的业务数据。

希望以上的回答能让你满意，如果有问题保持沟通！

2019-05-20

Teamillet 1

编译原理还好没忘记，少用正则.....

2019-05-30

Geek_ebda96 1

老师你好，请问系统负载所指的正在运行或者等待的进程数和CPU的数目的百分比，但我大部分得系统来说运行的进程数目基本不会边，都是稳定的，在变的应该是应用程序本身的线程数，这个参数对于性能观察来说有用么，哪些方法能看到因为线程数过多或者线程的使用率很高导致性能问题，比如TOP查看CPU使用率？

2019-05-28

作者回复

进程或线程，不单单指的是进程。可用使用top指令查看整体的使用率以及单个进程和单个线程的使用率。

2019-05-29

张小小的席大da 1

老师 您好 我想请问个问题 您说的压测工具ab 我用过 但是我们现在想要模拟完整的项目流程去压测(直播项目) 您有什么好的建议么

2019-05-28

作者回复

建议使用jmeter或者loadrunner，可以通过录制业务流程，生成jmeter脚本。近期我会讲到测试工具的使用。

2019-05-28

行者 1

老师，我想到服务器响应时间可以进一步细分到服务器中位数响应时间，服务器最慢的响应时间；要尽可能让所有接口响应时间接近，避免长尾。

2019-05-22

作者回复

行者你好，你想到的很好，我们在做性能测试的时候有一个percentage of the requests served within a certain time指标，就是反应单位时间内，不同响应时间的占比率，例如50% 的响应时间是1ms以内，80%的响应时间是2ms以内，99%的响应时间是5ms以内。说明有19%是在2ms~5ms以内，如果这个范围太大，有可能存在性能问题，具体问题具体分析。

上述我说的这个参数应该就是你现在描述的性能指标，有问题保持沟通。

2019-05-23

tyul 1

网络吞吐量里面的字被和谐了

2019-05-22

编辑回复

感谢这位热心同学！

2019-05-22

庄风 1

希望老师在以后的课程中讲一下相关性能监测工具软件。我在项目中遇到了性能问题后，确实不知道该如何分析、监测系统状态，并进行相应的处理和优化。之前一直用的是Windows自带的性能监视器，不过并不好用。

2019-05-22

张德 1

主要是纯粹的JAVA专栏 很期待很期待

2019-05-22

丿北纬91度灬 1

老师您好，看到性能调优这块的指标内容，需要涉及到很多方面的知识，比如Linux操作，JVM虚拟机原理，Java源码基础知识等，这些东西对于一个刚接触后台开发的人来说，确实比较吃力，在学习本专栏课程的同时去学习这些东西，这些有优先级吗，哪些是需要先急需掌握一部分的

2019-05-22

作者回复

你好，在我看来Linux操作系统、JVM以及Java基础三者的学习并不会存在前后顺序，可以并行学习。对于Java初学者来说，建议可以一边了解Java的运行原理(JVM)一边学习Java基础知识，基础打好之后，我们可以进入高级篇，比如Java的并发编程，如果需要进行一些项目实践，我们可以学习Spring相关框架组件。Linux操作系统我们也可以在了解基础原理的前提下，先熟练掌握一些常用操作命令，再作深入学习。

2019-05-22

anz

1

目前在做交易系统的性能优化,最直接的感受是,在不做产品逻辑变动的情况下,合并查询,批量操作,几乎解决了我目前遇到的80%的问题;但是配置优化方面自己比较欠缺,希望跟着老师能补补自己的短板

2019-05-21

作者回复

给你赞一个

2019-05-22

Mr. Huang

1

老师您好，请问您提到的那几个响应时间测试，有没有比较好的工具推荐

2019-05-21

作者回复

你好Mr.Huang，比较常用的压力测试工具有AB，这是一个Linux系统工具，使用起来很简便，还有jmeter工具，可以在Windows环境下运行使用，个人比较习惯使用AB。

2019-05-21