关于性能优化

作者：ptteng
原文链接：https://ld246.com/article/1498126039545
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

文章目录：

1.为什么要做性能优化

2.性能优化之前，先定位问题

3.Nginx和Tomcat

4.Controller和Service又做了什么

5.Memcache和DB做了什么

6.听说他们久经沙场

一 为什么要做性能优化

“这破网站慢死了。”果冻咬着牙说，“这些程序员都是干嘛吃的？”

毛毛看着果冻笑，没说话，毕竟果冻说的破网站是“修真院自己的官网”，毕竟果冻说的这些“干嘛的”程序员就在他们UE组对面。果冻是一个萌妹子，怎么说都行，毛毛做为一个PM，可是经常注意宇桌子下面常备两块板砖的。

培宇有一点点脸红，果冻对培宇是一直有偏见的，毕竟当众表白被拒是谁都难以接受的。自那以后，着“没有打击培宇的机会，就创造打击培宇的机会”的原则，果冻让培宇生生见识到了什么叫做“由生恨两重天”。

旁边的小马哥不干了，当场就“哐“的一下拍桌子，整个二楼的人都吓了一跳，看着小马哥。

果冻也一样吓了一跳，回过神来之后就当场发飚了：”伟江你要死吗？“

伟江本来想说”你行你上啊“，但是看了看果冻渐渐变冷的眼神，终究还是说不出口，可是气氛一度尬啊，只好小声说了一句：”我去找老大解决去。。。。。。“

所有人：”。。。。。“

培宇笑的不行：”伟江就该这么有霸气，在果冻的淫威下终于爆发了，一怒而起，然后一溜烟的跑去老大了~“

二 性能优化之前，先定位问题

说到性能之前，先谈量化。

说网站慢，那是普通用户的说的话。

而对程序员来讲，必须是到多少MS。

多少MS才算慢？通常来说，人的眼睛对于200MS以上的时延是有反应的，所以一般而言，一整个页都应该在200MS之内完成。

对于复杂的请求，可以再稍微慢一点，毕竟大家还能忍，能忍的程度跟网站的价值成正比。反正我看黄片的时候是挺能忍的~

简单的用户个人信息这种请求，应该在50MS左右，List的数据，差不多在100MS左右。这是比较正的数据。

以上这些全是暗灭大人骂伟江的原话。

”所以你做为一个程序员，过来跟我说网站慢？你应该直接告诉我说现在哪一个请求慢，慢到多少M，应该达到多少MS。“

暗灭大人一边狠狠的骂了伟江一顿，一边继续在QQ上教小师妹学习：“小师妹啊，我听说最近刚上了一部电影叫做《人民的名义》，我看了看，感觉对于写代码很有帮助啊，强烈推荐给你看，这样，好有人买了两张电影票给我，我带你一起去学习一下吧。”

伟江赶紧的拿出手机来，两张票买后，然后截图发给老大。

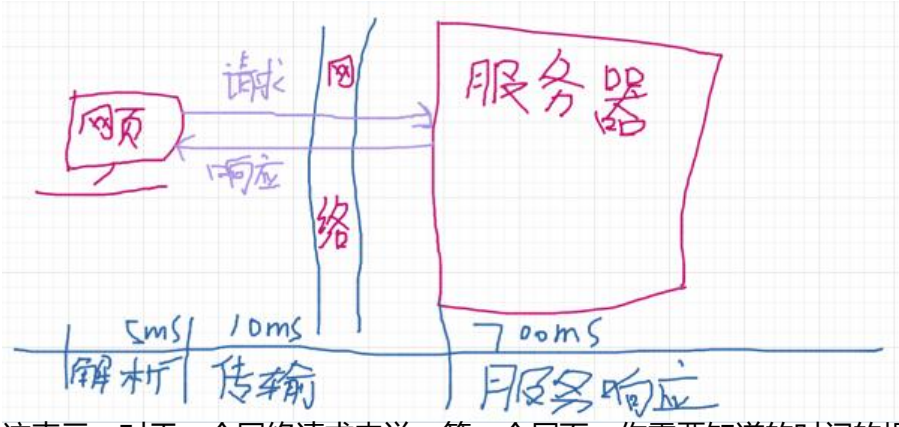
暗灭大人看到截图，这才满意，跟小师妹又回了句：“稍等啊，我要开个董事会。”

“拿白板来！”老大说。

首先，果冻朝你设了一箭。你接到了果冻射过来的这支爱之箭，看了看箭上附的小纸条，写的是什么容，按果冻的指示把她想要做的事给做完了，然后把结果又放到小纸条上，再扔回给了果冻，果冻收箭之后，打开看完。

这是一个完整的Http请求过程。

然后老大在白板上这么画了一下。



这表示，对于一个网络请求来说，第一个层面，你需要知道的时间的损耗可以分解成三大部分。

第一部分**前端的响应**，一般包括解析和渲染，这部分的性能跟前端的代码，前端硬件有关系。

第二部分就是**网络延迟**，这部分的代码正常来讲是在8~16MS左右，是的，Http请求就是差不多这个能，如果是WebSocket几乎可以做到零延迟，有暗灭大人当年写的多人在线扫雷（http://game.pttg.com 支持多人在线玩扫雷，一个人点错所有人完蛋的恶趣味）为证。

第三部分就是**服务器端的响应**，我们说的是网站慢，一般而言，也就是主要在这里，要做性能优化的方，基本上也是看这里。

伟江听的正津津有味，突然间发现老大已经转身要走了，赶紧拿出手机来又订了一个烛光晚餐。

“老大，老大，我怎么才能知道每一段时间间隔多久呢？在哪可以看得到啊？”

“F12，Fiddle都行啊，这种叫端到端的响应时间，服务器端的响应时间啊，要看Nginx的Access.log”老大满意的点了点头：“伟江的学习态度很好嘛，不像培宇，太笨，又不肯用心。”

“Access.log里会记录服务器端自己的响应时间，这是后端查看性能的重要依据，也是找运维背锅的备技能。一旦你发现服务器端的响应时间没问题，而端到端的响应有问题，那就百分百的是网络问题，可以直接找运维同学背锅了。”

“这些是做性能优化之前必须要弄明白的，问题到底出在哪儿呢？否则怎么做性能优化？所以解决性的问题，跟你泡妹子是一样的，要先脱掉妹子的外套，再脱掉外套里面的上衣，然后。。”

伟江擦了擦口水，说：“老大，如果确实是我们后端的问题呢，前端响应正常，网络传输也正常，就后端响应出问题，该怎么解决？”

三 .Nginx和Tomcat

“为什么要在前端挂一个Nginx，而不是直接用Tomcat呢？”

“你别管，总之好处多多了，不过是域名，还是跨域，还是防Ddos攻击，Nginx都是利器，很好用。在就照着来。”

Nginx相当于是一个大的总管，所有的请求都是先到Nginx这里来，然后再发给后面的Tomcat，或者Resin什么的。

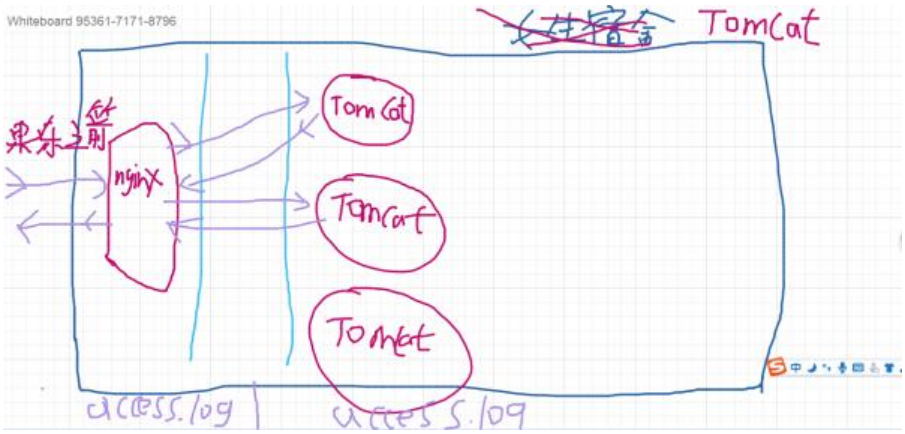
后面的Tomcat像当于是一个个漂亮女生宿舍的话：

所以Nginx就像当于是女生宿舍的门卫

对的。总之Nginx很赞，我很喜欢。Nginx会记录有所有的出入时间。

如果Nginx的耗时比较长，一般而言，问题都不太会直接出在nginx身上，可以直接再去查Tomcat/resn/jetty这些女生宿舍的岗了，哦，不是，去查看这些服务器的日志响应时间了。

所以，Nginx和Tomcat之间的图应该是这样的。



Tomcat本身也会有一个日志记录，也是Access.log,其他的叫别的什么日志无所谓了，反正从名字来，就是进女生宿舍的时间和离开的时间--那群Tomcat的研发人员一定是做了很多的实地调研才能有此神似的命名。

那么锅就继续甩给了Tomcat。Tomcat之后是什么呢？基本上就全部是Java人员写的代码了。所以我要不要往女生宿舍里偷偷看一眼？

“**可是最关键的就是女生宿舍吧，哪能只偷偷看一眼就走！**”伟江有点不满意。

四 .Controller和Service又做了什么

说的没错，Tomcat里的时间其实还可以再分解的更细一些。第一个接受到响应的就是Controller，而前说过，Controller通常是用来控制Service的，你可以理解为，每一个Service就是一个漂亮妹子。

Controller就是一个宿舍的宿管员。就是少林足球里的包租婆，她来负责管理这些小妹妹做什么事儿。

Controller依次调用女生Service1，女生Service2，女生Service3~然后再把结果返回给你。

所以Tomcat响应的时间可以分为：

1.Controller的处理时间

2.Service的调用时间

3.返回结果的处理时间

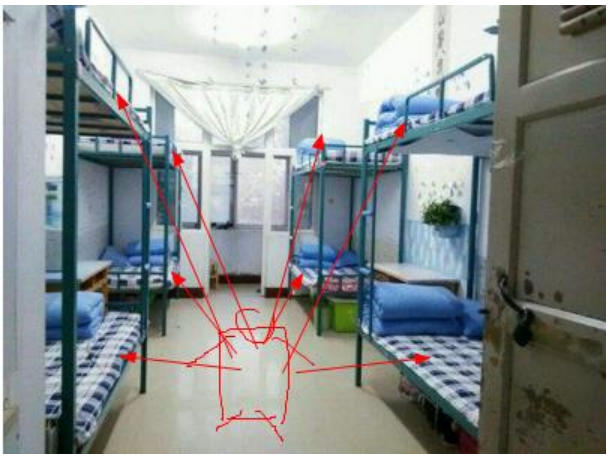
controller本身的处理事情，一般都会是在开始和结束各打一条毫秒数这是所有的业务逻辑处理的总间。调用各Service的时间包含网络传输和Service的响应时间。返回结果的时间一般都是解析成Json时间。

生成Json的时间通常很短，很短，网络传输的时间说不好，RMI的话，除第一次建立连接外，几乎可忽略不计，但是一个Controller里调用上百次的话，速度就会慢起来。Http无疑是最慢的，单纯建立要花几MS，对性能有更高要求的话，用Thrift和ProtocolBuffer等。

而我一般都喜欢用RMI，不喜欢异构的东西，Http又引入了额外的负担，所以比较喜欢用Tuscany直调用RMI。

所以性能优化的重点，就是要弄明白，如果一个大老爷们，比如说培宇，进了女生宿舍之后，总共花500多MS的时间，他在包租婆那里花了多长时间？他跑到妹子1的床前花了多久？和妹子1在一起同MS？跑到妹子2的床前是多久？在妹子2那里花了几MS？

这些数据都要清清楚楚，还是要用图来表明一下。



这个就是小培宇。如果他爬上每一个妹子的床之后都回来找包租婆登记一下，就是这样的。

如果他爬了一个妹子的床，然后赶紧的去爬了另一个妹子的床，这就是另一个服务调用服务的过程了。

这两种方式的优劣，我们随后再说。

如果他进去之后变成了8个小培宇，每个小培宇分别去上8个妹子的床，这就是多线程和异步。

如果他自己带了一个小黑板，啪啪啪写上自己的地址，然后就走。8个妹子看到地址之后各自去找小宇，这就是消息队列。

总之，先不讨论这么复杂的东西，我们就只说，培宇是一个妹子一个妹子的上。。。。。床。

那么，Controller那里应该怎么统计时间呢？

一般而言，我们会手写一个AOP的Util。并不会在Controller里手写StartAt和EndAt。

这个Aop写起来很容易，正常来讲，会在代码里这么写：

如果超过了50MS，记录日志，入参，时间。

如果不超过50MS，不记录。

这个50MS依据每个人的身体情况，哦不，依据每个应用的情况来自行设置。

所以，到了这里，你就会知道了，我们把果冻射来的爱之箭，已经穿透了包租婆，到了女生宿舍，并分解到了每一个床位的时间。

这就是一个分析和定位的过程，而通常就是在Controller的代码里，一般我们都叫做WEB工程，在这去用Aop来记录第一个Service的响应时间。

如果我们发现了，培宇在2号女生那儿花费的时间特别长呢？

五.Memcache和DB做了什么

“那就把培宇阉了！”伟江斩钉截铁的说。

“好的，现在就动手。” 老大满怀期待的看着他。

这得把培宇喊过来，问问他在2号女生那里做什么了。

写到这里我必须得说，这是公开版，公开版只能说，培宇在约2号女生看电影。

内部版，非公开发生的，培宇做的事情就比较多了。。

好了，先说看电影。

培宇说：乖，我们去看电影吧。

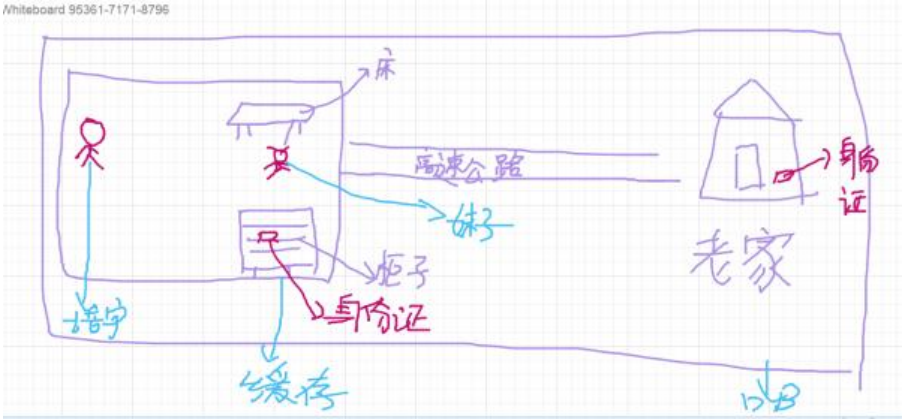
妹子说：好的，我要找找我的身份证。

这个时间发现身份证不在身边，在老家，于是妹子说，乖，你等着，我去老家把身份证拿过来。

这就叫做直接从DB里拿数据。

所以时间性能的损耗，就在于是：培宇和妹子说话，妹子回老家拿身份证。

大概的场景就是这样的。



培宇说：乖，我们看电影去吧。

妹子说：好啊，好啊，等我，我回老家拿身份证，然后一个月过去了。

培宇和妹子想了想，还是没办法看电影。

第二次培宇又说：乖，我们看电影吧。

妹子说;好啊，好啊，等我，我回家拿身份证，然后一个月又过去了。

培宇和妹子再见面之后，想了想，还是没办法看电影。

上一篇讲过Dao。其实Service里对于Dao的访问时间分析，也是这么记录的。同理，如果你要用到多Dao，可能是先回老家拿身份证，再去商场买件衣服等等等等。

而我们统计妹子去拿身份证的时间，也是同理可以用刚刚说过的Aop的性能统计Util.

这样一个Service处理事情的时间，就分成了本地运算和获取数据的时间。

通常本地运算的时间会比较快，而获取数据的时间会比较慢。

如果说你发现了妹子总是回老家，而且时间刚好是一个月，你怎么办？

给妹子提个醒，可不可以把身份证放到旁边的衣柜里？这样我们就可以直接从衣柜里拿身份证了啊。

这个衣柜，就叫做缓存。

而一般而言，我们的包租婆是不管妹子从哪拿身份证的，总之能接待好客人，呃，接待好培宇就行了。

所以如果要继续区分，性能问题。必须要知道，数据有没有走缓存？从缓存中的数据取出来要多久？常情况下应该几MS，为什么妹子花了20多MS？是不是柜子前面的人比较多？还是从床到柜子中间的坏了？柜子是在哪里放着的？

妹子回老家取身份证，会不会堵车？

这些都是需要靠日志来记录时间，也简单提醒一下，假设不是培宇一个人来呢。10000个人同时来呢你必须得在日志里记录一些关键信息，确保知道哪个时间是哪个老板，呃，哪个客户的。

六.听说他们久经沙场

其实性能优化有很多很多方案，不同的应用场景，解决方案是不同的。但是最根本的就是要明白，时倒底损耗在什么地方？

对于果冻来说，就是简单的射了一支爱之箭，后面做了哪些，是不需要知道的。

可是伟江就必须得明白，培宇对每一个女生宿舍的妹子都做了什么。在保安那里花了多久，在包租婆里花了多久，在每一个女生床上花了多久，妹子从柜子里取身份证要多久，从老家取身份证又要多久。

我们一直说的，一个程序员的真正工作的地方不在笔记本上，也不是在开发环境，也不是在测试环境而是在线上环境，就是这个道理。

做为一个老程序员来说，对于线上有哪几台服务器，IP是多少，内存是多大，硬盘是多大，CPU是几的，常用负载是多少，IO有多少，每天增长的日志是多少，DB里的数据库大小是多大，每天的增量多少，系统中最慢的请求是什么，每天会访问多少次，最快的请求是什么，每天应该访问多少次，每访问的时候正常应该是在多少MS内返回，一次会取多少条数据，分解到哪几个Service，每个Service不同时间请求是多少，有多少穿透DB的次数。

所有的这些数据都要了如指掌，这样才能够非常快的定位线上的问题。

我们不能说这样的程序员是一个水平多厉害的高手，但是他如果做到了这么一点，他就是一个久经沙的老兵，这样才像是一个职业的工程师。

所以这篇文章的重点就在于是，简单的讲解一下时间之箭的运转，中间会有哪些环节，每个环节应该费多长时间。

简单说，只有先找到妹子，才会知道应该在妹子身上花多少时间，对吧？

** 技能树.IT修真院 http://www.jnshu.com**

“我们相信人人都可以成为一个工程师，现在开始，找个师兄，带你入门，掌控自己的学习节奏，学的路上不再迷茫”。

** 这里是技能树.IT修真院，成千上万的师兄在这里找到了自己的学习路线，学习透明化，成长可见，师兄1对1免费指导。**