56 | 套路篇：优化性能问题的一般方法

2019-04-05 倪朋飞

Linux性能优化实战

进入课程 >

讲述：冯永吉
时长 11:43　大小 10.74M

你好，我是倪朋飞。

上一节，我带你一起梳理了，性能问题分析的一般步骤。先带你简单回顾一下。

我们可以从系统资源瓶颈和应用程序瓶颈，这两个角度来分析性能问题的根源。

从系统资源瓶颈的角度来说，USE 法是最为有效的方法，即从使用率、饱和度以及错误数这三个方面，来分析 CPU、内存、磁盘和文件系统 I/O、网络以及内核资源限制等各类软硬件资源。至于这些资源的分析方法，我也带你一起回顾了，咱们专栏前面几大模块的分析套路。

从应用程序瓶颈的角度来说，可以把性能问题的来源，分为资源瓶颈、依赖服务瓶颈以及应用自身的瓶颈这三类。

资源瓶颈的分析思路，跟系统资源瓶颈是一样的。

依赖服务的瓶颈，可以使用全链路跟踪系统，进行快速定位。

而应用自身的问题，则可以通过系统调用、热点函数，或者应用自身的指标和日志等，进行分析定位。

当然，虽然系统和应用是两个不同的角度，但在实际运行时，它们往往相辅相成、相互影响。

系统是应用的运行环境，系统瓶颈会导致应用的性能下降。

而应用程序不合理的设计，也会引发系统资源的瓶颈。

我们做性能分析，就是要结合应用程序和操作系统的原理，揪出引发问题的“真凶“。

找到性能问题的来源后，整个优化工作其实也就完成了一大半，因为这些瓶颈为我们指明了优化的方向。不过，对于性能优化来说，又有哪些常见的方法呢？

今天，我就带你一起来看看，性能优化的一般方法。同上一节的性能分析一样，我们也可以从系统和应用程序，这两个不同的角度来进行性能优化。

系统优化

首先来看系统的优化。在上一节，我曾经介绍过，USE 法可以用来分析系统软硬件资源的瓶颈，那么，相对应的优化方法，当然也是从这些资源瓶颈入手。

实际上，咱们专栏的前四个模块，除了最核心的系统资源瓶颈分析之外，也已经包含了这些常见资源瓶颈的优化方法。

接下来，我就从 CPU 性能、内存性能、磁盘和文件系统 I/O 性能以及网络性能等四个方面，带你回顾一下它们的优化方法。

CPU 优化

首先来看 CPU 性能的优化方法。在CPU 性能优化的几个思路中，我曾经介绍过，**CPU 性能优化的核心，在于排除所有不必要的工作、充分利用 CPU 缓存并减少进程调度对性能的影响。**

从这几个方面出发，我相信你已经想到了很多的优化方法。这里，我主要强调一下，最典型的三种优化方法。

第一种，把进程绑定到一个或者多个 CPU 上，充分利用 CPU 缓存的本地性，并减少进程间的相互影响。

第二种，为中断处理程序开启多 CPU 负载均衡，以便在发生大量中断时，可以充分利用多 CPU 的优势分摊负载。

第三种，使用 Cgroups 等方法，为进程设置资源限制，避免个别进程消耗过多的 CPU。同时，为核心应用程序设置更高的优先级，减少低优先级任务的影响。

内存优化

说完了 CPU 的性能优化，我们再来看看，怎么优化内存的性能。在如何“快准狠”找到系统内存的问题中，我曾经为你梳理了常见的一些内存问题，比如可用内存不足、内存泄漏、Swap 过多、缺页异常过多以及缓存过多等等。所以，说白了，内存性能的优化，也就是要解决这些内存使用的问题。

在我看来，你可以通过以下几种方法，来优化内存的性能。

第一种，除非有必要，Swap 应该禁止掉。这样就可以避免 Swap 的额外 I/O ，带来内存访问变慢的问题。

第二种，使用 Cgroups 等方法，为进程设置内存限制。这样就可以避免个别进程消耗过多内存，而影响了其他进程。对于核心应用，还应该降低 oom_score，避免被 OOM 杀死。

第三种，使用大页、内存池等方法，减少内存的动态分配，从而减少缺页异常。

磁盘和文件系统 I/O 优化

接下来，我们再来看第三类系统资源，即磁盘和文件系统 I/O 的优化方法。在磁盘 I/O 性能优化的几个思路 中，我已经为你梳理了一些常见的优化思路，这其中有三种最典型的方法。

第一种，也是最简单的方法，通过 SSD 替代 HDD、或者使用 RAID 等方法，提升 I/O 性能。

第二种，针对磁盘和应用程序 I/O 模式的特征，选择最适合的 I/O 调度算法。比如，SSD 和虚拟机中的磁盘，通常用的是 noop 调度算法；而数据库应用，更推荐使用 deadline 算法。

第三，优化文件系统和磁盘的缓存、缓冲区，比如优化脏页的刷新频率、脏页限额，以及内核回收目录项缓存和索引节点缓存的倾向等等。

除此之外，使用不同磁盘隔离不同应用的数据、优化文件系统的配置选项、优化磁盘预读、增大磁盘队列长度等，也都是常用的优化思路。

网络优化

最后一个是网络的性能优化。在网络性能优化的几个思路中，我也已经为你梳理了一些常见的优化思路。这些优化方法都是从 Linux 的网络协议栈出发，针对每个协议层的工作原理进行优化。这里，我同样强调一下，最典型的几种网络优化方法。

首先，从内核资源和网络协议的角度来说，我们可以对内核选项进行优化，比如：

你可以增大套接字缓冲区、连接跟踪表、最大半连接数、最大文件描述符数、本地端口范围等内核资源配额；

也可以减少 TIMEOUT 超时时间、SYN+ACK 重传数、Keepalive 探测时间等异常处理参数；

还可以开启端口复用、反向地址校验，并调整 MTU 大小等降低内核的负担。

这些都是内核选项优化的最常见措施。

其次，从网络接口的角度来说，我们可以考虑对网络接口的功能进行优化，比如：

你可以将原来 CPU 上执行的工作，卸载到网卡中执行，即开启网卡的 GRO、GSO、RSS、VXLAN 等卸载功能；

也可以开启网络接口的多队列功能，这样，每个队列就可以用不同的中断号，调度到不同 CPU 上执行；

还可以增大网络接口的缓冲区大小以及队列长度等，提升网络传输的吞吐量。

最后，在极限性能情况（比如 C10M）下，内核的网络协议栈可能是最主要的性能瓶颈，所以，一般会考虑绕过内核协议栈。

你可以使用 DPDK 技术，跳过内核协议栈，直接由用户态进程用轮询的方式，来处理网络请求。同时，再结合大页、CPU 绑定、内存对齐、流水线并发等多种机制，优化网络包的处理效率。

你还可以使用内核自带的 XDP 技术，在网络包进入内核协议栈前，就对其进行处理。这样，也可以达到目的，获得很好的性能。

应用程序优化

说完了系统软硬件资源的优化，接下来，我们再来看看应用程序的优化思路。

虽然系统的软硬件资源，是保证应用程序正常运行的基础，但你要知道，**性能优化的最佳位置，还是应用程序内部**。为什么这么说呢？我简单举两个例子你就明白了。

第一个例子，是系统 CPU 使用率（sys%）过高的问题。有时候出现问题，虽然表面现象是系统 CPU 使用率过高，但待你分析过后，很可能会发现，应用程序的不合理系统调用才是罪魁祸首。这种情况下，优化应用程序内部系统调用的逻辑，显然要比优化内核要简单也有用得多。

再比如说，数据库的 CPU 使用率高、I/O 响应慢，也是最常见的一种性能问题。这种问题，一般来说，并不是因为数据库本身性能不好，而是应用程序不合理的表结构或者 SQL 查询语句导致的。这时候，优化应用程序中数据库表结构的逻辑或者 SQL 语句，显然要比优化数据库本身，能带来更大的收益。

所以，在观察性能指标时，你应该先查看**应用程序的响应时间、吞吐量以及错误率**等指标，因为它们才是性能优化要解决的终极问题。以终为始，从这些角度出发，你一定能想到很多优化方法，而我比较推荐下面几种方法。

第一，从 CPU 使用的角度来说，简化代码、优化算法、异步处理以及编译器优化等，都是常用的降低 CPU 使用率的方法，这样可以利用有限的 CPU 处理更多的请求。

第二，从数据访问的角度来说，使用缓存、写时复制、增加 I/O 尺寸等，都是常用的减少磁盘 I/O 的方法，这样可以获得更快的数据处理速度。

第三，从内存管理的角度来说，使用大页、内存池等方法，可以预先分配内存，减少内存的动态分配，从而更好地内存访问性能。

第四，从网络的角度来说，使用 I/O 多路复用、长连接代替短连接、DNS 缓存等方法，可以优化网络 I/O 并减少网络请求数，从而减少网络延时带来的性能问题。

第五，从进程的工作模型来说，异步处理、多线程或多进程等，可以充分利用每一个 CPU 的处理能力，从而提高应用程序的吞吐能力。

除此之外，你还可以使用消息队列、CDN、负载均衡等各种方法，来优化应用程序的架构，将原来单机要承担的任务，调度到多台服务器中并行处理。这样也往往能获得更好的整体性能。

小结

今天，我带你一起，从系统和应用程序这两个角度，梳理了常见的性能优化方法。

从系统的角度来说，CPU、内存、磁盘和文件系统 I/O、网络以及内核数据结构等各类软硬件资源，为应用程序提供了运行的环境，也是我们性能优化的重点对象。你可以参考咱们专栏前面四个模块的优化篇，优化这些资源。

从应用程序的角度来说，降低 CPU 使用，减少数据访问和网络 I/O，使用缓存、异步处理以及多进程多线程等，都是常用的性能优化方法。除了这些单机优化方法，调整应用程序的架构，或是利用水平扩展，将任务调度到多台服务器中并行处理，也是常用的优化思路。

虽然性能优化的方法很多，不过，我还是那句话，一定要避免过早优化。性能优化往往会提高复杂性，这一方面降低了可维护性，另一方面也为适应复杂多变的新需求带来障碍。

所以，性能优化最好是逐步完善，动态进行；不追求一步到位，而要首先保证，能满足当前的性能要求。发现性能不满足要求或者出现性能瓶颈后，再根据性能分析的结果，选择最重要的性能问题进行优化。

思考

最后，我想邀请你一起来聊聊，当碰到性能问题后，你是怎么进行优化的？有没有哪个印象深刻的经历可以跟我分享呢？你可以结合我的讲述，总结自己的思路。

欢迎在留言区和我讨论，也欢迎把这篇文章分享给你的同事、朋友。我们一起在实战中演练，在交流中进步。

上一篇　55 | 套路篇：分析性能问题的一般步骤

下一篇　57 | 套路篇：Linux 性能工具速查

精选留言 (6)

写留言

郭江伟
2019-04-05

10

每篇都认真看过， 还要二刷 ，二刷准备联系趣谈操作系统专栏一起刷。
今年计划熟悉Linux内核

展开

我来也
2019-04-05

2

[D56打卡]
“性能优化的最佳位置，还是应用程序内部”

硬件性能摆在那，操作系统就那些。
这些都是摆在那，我们改变不了的。...

展开

付坤
2019-04-08

1

大赞👍，我在实战的时候，总会遇到突然想不起来接下来的详细指标该用哪个工具了，还要一篇一篇的翻老师的文章去找。
本来还打算自己总结一篇各个指标查看用到的工具，没想到老师已经总结好了，谢谢老师。

展开

作者回复: 👍

ninuxer
2019-04-06

1

打卡day60
从分析到优化，为啥感觉分析出来容易，优化却不太容易，分析只是一个线性事情，但优化，却是一个系统性事情

作者回复: 嗯，还要看具体场景和性能要求，有些场景比较简单，只需要修改系统配置就可以解决；不过也有很多需要调整软件架构来解决

如果
2019-04-23

DAY56，打卡

展开

玉剑冰锋
2019-04-08

说到这里想请教老师一个问题，我们经常说到磁盘I/O，单挂裸盘和做RAID，如果从读写角度单挂裸盘同时读写是不是一定比RAID好，但是从长期维护角度来说RAID更易于维护，想请教老师如何取舍？

作者回复: 从可靠性角度来说，单盘是不推荐的，磁盘损坏数据就丢失了。性能的话，要看使用什么RAID级了