45 | Linux 性能优化答疑（五）

2019-03-06 倪朋飞

讲述：冯永吉
时长 07:15 大小 6.65M

你好，我是倪朋飞。

专栏更新至今，四大基础模块的最后一个模块——网络篇，我们就已经学完了。很开心你还没有掉队，仍然在积极学习思考和实践操作，热情地留言和互动。还有不少同学分享了在实际生产环境中，碰到各种性能问题的分析思路和优化方法，这里也谢谢你们。

今天是性能优化答疑的第五期。照例，我从网络模块的留言中，摘出了一些典型问题，作为今天的答疑内容，集中回复。同样的，为了便于你学习理解，它们并不是严格按照文章顺序排列的。

每个问题，我都附上了留言区提问的截屏。如果你需要回顾内容原文，可以扫描每个问题右下方的二维码查看。

问题 1：网络收发过程中缓冲区的位置

安排

当一个网络帧到达网卡后，网卡会通过 DMA 方式，把这个网络包放到收包队列中；然后通过硬中断，告诉中断处理程序已经收到了网络包。

接着，网卡中断处理程序会为网络帧分配内核数据结构（sk_buff），并将其拷贝到 sk_buff 缓冲区中；然后再通过软中断，通知内核收到了新的网络帧。

接下来，内核协议栈从缓冲区中取出网络帧，并通过网络协议栈，从下到上逐层处理这个网络帧。

老师你好，上面的一段话有些疑问想请教一下。

收包队列是属于哪里的存储空间，是属于物理内存吗，还是网卡中的存储空间，通过dma方式把数据放到收包队列，我猜这个收包队列是物理内存中的空间。这个收包队列是由内核管理的吧，也就是跟某一个进程的用户空间地址没关系？

那sk_buf缓冲区又是哪里的存储空间，为什么还要把收包队列拷贝到这个缓冲区呢，这个缓冲区是协议栈维护的吗？也属于内核，跟进程的用户

写于 2019/02/26

空间地址有关系吗?

socket的接收发送缓冲区是映射到进程的用户空间地址的吗?还是由协议栈为每个socket在内核中维护的缓冲区?

还有上面说到的这些缓冲区跟cache和buf有什么关系?会被回收吗?

内核协议栈的运行是通过一个内核线程的方式来运行的吗?是否可以看到这个线程的名字?

引自:Linux性能优化实战
33 | 关于 Linux 网络,你必须知道这些(上)

识别二维码打开原文
「极客时间」App

第一点,是网络收发过程中,收发队列和缓冲区位置的疑问。

在 关于 Linux 网络,你必须要知道这些 中,我曾介绍过 Linux 网络的收发流程。这个流程涉及到了多个队列和缓冲区,包括:

网卡收发网络包时,通过 DMA 方式交互的**环形缓冲区**;

网卡中断处理程序为网络帧分配的，内核数据结构 **sk_buff 缓冲区**；

应用程序通过套接字接口，与网络协议栈交互时的**套接字缓冲区。**

不过相应的，就会有两个问题。

首先，这些缓冲区的位置在哪儿？是在网卡硬件中，还是在内存中？这个问题其实仔细想一下，就很容易明白——这些缓冲区都处于内核管理的内存中。

其中，**环形缓冲区**，由于需要 DMA 与网卡交互，理应属于网卡设备驱动的范围。

sk_buff 缓冲区，是一个维护网络帧结构的双线链表，链表中的每一个元素都是一个网络帧（Packet）。虽然 TCP/IP 协议栈分了好几层，但上下不同层之间的传递，实际上只需要操作这个数据结构中的指针，而无需进行数据复制。

套接字缓冲区，则允许应用程序，给每个套接字配置不同大小的接收或发送缓冲区。应用程序发送数据，实际上就是将数据写入缓冲区；而接收数据，其实就是从缓冲区中读取。至于缓冲区中数据的进一步处理，则由传输层的 TCP 或 UDP 协议来完成。

其次，这些缓冲区，跟前面内存部分讲到的 Buffer 和 Cache 有什么关联吗？

这个问题其实也不难回答。我在内存模块曾提到过，内存中提到的 Buffer ，都跟块设备直接相关；而其他的都是 Cache。

实际上，sk_buff、套接字缓冲、连接跟踪等，都通过 slab 分配器来管理。你可以直接通过 /proc/slabinfo，来查看它们占用的内存大小。

问题 2：内核协议栈，是通过一个内核线程的方式来运行的吗

第二个问题，内核协议栈的运行，是按照一个内核线程的方式吗？在内核中，又是如何执行网络协议栈的呢？

Days

写于 2019/02/09

老师春节不休息，大赞啊，老师可否讲解一下一个包从网卡接收，发送在内核协议栈的整个流程，这样性能分析的时候，更好的理解数据包阻塞在哪里？

引自：Linux性能优化实战

34 | 关于 Linux 网络，你必须知道这些（下）

识别二维码打开原文
「极客时间」 App

说到网络收发，在中断处理文章中我曾讲过，其中的软中断处理，就有专门的内核线程 ksoftirqd。每个 CPU 都会绑定一个 ksoftirqd 内核线程，比如， 2 个 CPU 时，就会有 ksoftirqd/0 和 ksoftirqd/1 这两个内核线程。

不过要注意，并非所有网络功能，都在软中断内核线程中处理。内核中还有很多其他机制（比如硬中断、kworker、slab 等），这些机制一起协同工作，才能保证整个网络协议栈的正常运行。

关于内核中网络协议栈的工作原理，以及如何动态跟踪内核的执行流程，专栏后续还有专门的文章来讲。如果对这部分感兴趣，你可以先用我们提到过的 perf、systemtap、bcc-tools 等，试着来分析一下。

问题 3：最大连接数是不是受限于 65535 个端口

我来也

写于 2019/02/12

[D35打卡]
09年那会,我所在公司的服务器端都是单进程+select.
后来把select换为了poll和epoll.
再后来还拆分成了多进程,N个网络收发层+M个业务处理层.
毕竟我们的情况是 业务处理的耗时远大于网络收发的耗时.
目前的网络收发层也只支持最大65530个并发连接,毕竟是单ip单端口的.
如果想支持更多并发连接,就另外再开一个进程.

并没有往C100K甚至是C1000K的方向上努力了.

引自：Linux性能优化实战
35 | 基础篇：C10K 和 C1000K 回顾

识别二维码打开原文
「极客时间」 App

我们知道，无论 TCP 还是 UDP，端口号都只占 16 位，也就说其最大值也只有 65535。那是不是说，如果使用 TCP 协议，在单台机器、单个 IP 地址时，并发连接数最大也只有 65535 呢？

对于这个问题，首先你要知道，Linux 协议栈，通过五元组来标志一个连接（即协议，源 IP、源端口、目的 IP、目的端口)。

明白了这一点，这个问题其实就有了思路。我们应该分客户端和服务器端，这两种场景来分析。

对客户端来说，每次发起 TCP 连接请求时，都需要分配一个空闲的本地端口，去连接远端的服务器。由于这个本地端口是独占的，所以客户端最多只能发起 65535 个连接。

对服务器端来说，其通常监听在固定端口上（比如 80 端口），等待客户端的连接。根据五元组结构，我们知道，客户端的 IP 和端口都是可变的。如果不考虑 IP 地址分类以及资源限制，服务器端的理论最大连接数，可以达到 2 的 48 次方（IP 为 32 位，端口号为 16 位），远大于 65535。

所以，综合来看，客户端最大支持 65535 个连接，而服务器端可支持的连接数是海量的。当然，由于 Linux 协议栈本身的性能，以及各种物理和软件的资源限制等，这么大的连接数，还是远远达不到的（实际上，C10M 就已经很难了）。

问题 4：“如何优化 NAT 性能”课后思考

倪朋飞

最后，给你留一个思考题。MASQUERADE 是最常用的一种 SNAT 规则，常用来为多个内网 IP 地址提供共享的出口 IP。

假设现在有一台 Linux 服务器，使用了 MASQUERADE 的方式，为内网的所有 IP 提供出口访问功能。那么，

当多个内网 IP 地址的端口号相同时，MASQUERADE 还可以正常工作吗？

如果内网 IP 地址数量或请求数比较多，这种方式有没有什么隐患呢？

摘录于 2019年03月03日

> 引自：Linux性能优化实战
> 41 | 案例篇：如何优化 NAT 性能？（上）

在 如何优化 NAT 性能 的最后， 我给你留了两个思考题。

MASQUERADE 是最常用的 SNAT 规则之一，通常用来为多个内网 IP 地址，提供共享的出口 IP。假设现在有一台 Linux 服务器，用了 MASQUERADE 方式，为内网所有 IP 提供出口访问功能。那么，

当多个内网 IP 地址的端口号相同时，MASQUERADE 还能正常工作吗？

内网 IP 地址数量或者请求数比较多的时候，这种使用方式有没有什么潜在问题呢？

对于这两个思考题，我来也、ninuxer 等同学，都给出了不错的答案：

我来也

写于 2019/02/25

[D41打卡]
在已有的项目经验中,还未涉及到过NAT.
倒是本地的虚拟机环境下,或者路由器上,会看到nat相关选项.

问题一:当多个内网 IP 地址的端口号相同时，MASQUERADE 还可以正常工作吗?
我觉得是可以正常工作的,要不然就不会允许设置ip地址段了.😄[纯属猜测哈]
在路由器上做端口映射时,一个外网端口只能对应一个内网的IP.
但是反方向,nat在转换源地址时,应该会记录原来的连接信息吧.要不然收到包该给谁发呢.

问题二:如果内网 IP 地址数量或请求数比较多，这种方式有没有什么隐患呢?
根据之前的经验,在请求数过多时,会导致CPU软中断上升.
再谷歌了下,有看到说:
iptables的conntrack表满了导致访问网站很慢.[https://my.oschina.net/jean/blog/189935]
```kernel 用 ip_conntrack 模块来记录 iptables 网络包的状态，并保存到 table 里（这个 table 在内存里），如果网络状况繁忙，比如高连接，高并发连接等会导致逐步占用这个 table 可用空间。```
```

优化Linux NAT网关[https://tech.youzan.com/linux_nat/]

```net.netfilter.nfconntrackbuckets 这个参数，默认有点小，连接数多了以后，势必造成“哈希冲突”增加，“哈希处理”性能下降。（是这样吗？）```

引自：Linux性能优化实战

41 | 案例篇：如何优化 NAT 性能？（上）

识别二维码打开原文
「极客时间」App
```

ninuxer

写于 2019/02/27

打卡day43
工作场景没用到nat，基本都是基于4层或7层的反代
针对第一个问题，是可以的，第二个问题不可以，我认为是有连接追踪表，文件数量，端口数量的限制

引自：Linux性能优化实战
41 | 案例篇：如何优化 NAT 性能？（上）

识别二维码打开原文
「极客时间」 App

先看第一点，当多个内网 IP 地址的端口号相同时，MASQUERADE 当然仍可以正常工作。不过，你肯定也听说过，配置 MASQUERADE 后，需要各个应用程序去手动配置修改端口号。

实际上，MASQUERADE 通过 conntrack 机制，记录了每个连接的信息。而在刚才第三个问题 中，我提到过，标志一个连接需要五元组，只要这五元组不是同时相同，网络连接就可以正常进行。

再看第二点，在内网 IP 地址和连接数比较小时，这种方式的问题不大。但在 IP 地址或并发连接数特别大的情况下，就可能碰到各种各样的资源限制。

比如，MASQUERADE 既然把内部多个 IP ，转换成了相同的外网 IP（即 SNAT），那么，为了确保发送出去的源端口不重复，原来网络包的源端口也可能会被重新分配。这样的话，转换后的外网 IP 的端口号，就成了限制连接数的一个重要因素。

除此之外，连接跟踪、MASQUERADE 机器的网络带宽等，都是潜在的瓶颈，并且还存在单点的问题。这些情况，在我们实际使用中都需要特别注意。

今天主要回答这些问题，同时也欢迎你继续在留言区写下疑问和感想，我会持续不断地解答。希望借助每一次的答疑，可以和你一起，把文章知识内化为你的能力，我们不仅在实战中演练，也要在交流中进步。

上一篇　44 | 套路篇：网络性能优化的几个思路（下）

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。