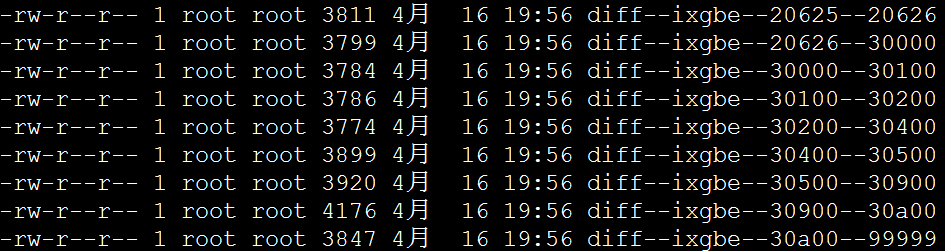
TMAS安装与配置

**一、Netmap安装与配置**

1. 确定内核版本

TMAS系统底层使用netmap报文捕获框架，能够实现10Gbps的线速抓包，不过，最新开源代码中，只包含了指定内核版本的驱动补丁，因此，在使用netmap的时候，要确认下内核版本是否在支持列表中。

以ixgbe驱动为例，可以查看/netmap/LINUX/patches/目录下的补丁列表：



补丁覆盖的内核版本由文件名的最后两段表示：

dif--ixgb--20625--20626表示适用的内核版本范围为2.6.25-2.6.26。

如果当前linux系统的内核版本不在此列表中，则可能需要升级内核版本。

1. 编译驱动和内核模块

请参考/netmap/LINUX/README文档，里面描述的非常清晰。

完全编译结果：

1. /netmap/LINUX/netmap\_lin.ko：netmap内核模块。
2. /netmap/LINUX/xxx/xxx.ko：支持netmap的网卡驱动。
3. /netmap/examples/\*：netmap例子程序，其中pkt-gen可作为测试程序。
4. 加载驱动和内核模块

Ixgbe.ko与netmap\_lin.ko两者有依赖关系，需要先加载netmap\_lin.ko再加载ixgbe.ko。

1. 使用rmmod ixgbe卸载原有驱动。
2. 使用insmod netmap\_link.ko加载netmap内核mok。
3. 使用insmod ixgbe.ko加载支持netmap的网卡驱动。

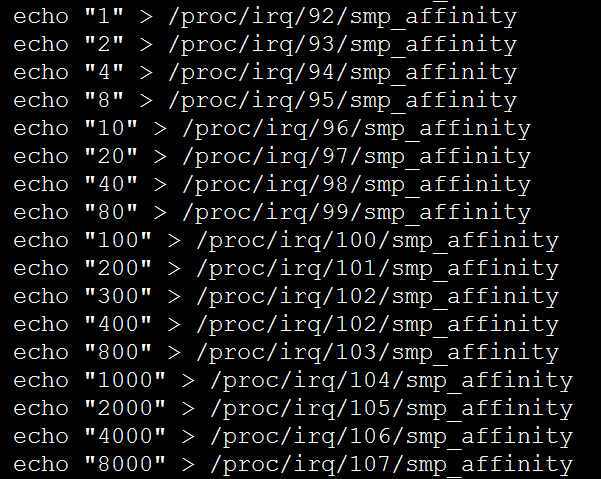
注意，安装驱动和内核模块时需要制定模块的路径。

1. 使用ifconfig eth0 up启动对应网卡。
2. 使用ifconfig eth0 promisc配置网卡工作在混杂模式。
3. 设置网卡中断分担。

一般情况下，驱动的报文收发软中断都会聚集到某一个CPU核心，在大流量情况下，中断数会非常多，当中断数达到一定限度后，CPU处理不过来，就会出现丢包。解决方案就是将驱动产生的软中断尽量分担到多个CPU核心上。

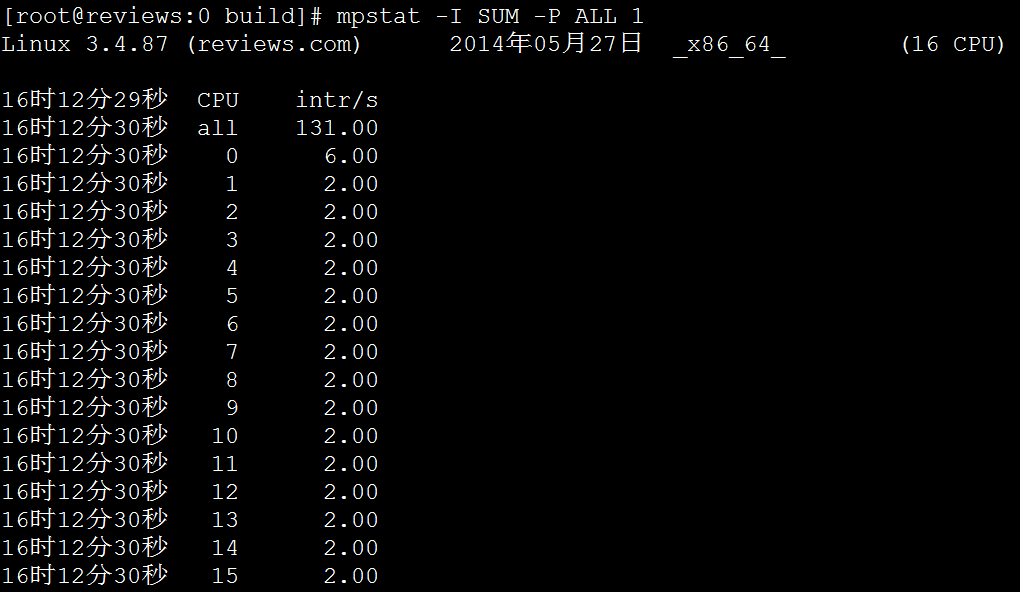
1) 使用cat /proc/interrupts | grep eth0(抓包网卡)获取多队列网卡的队列中断号。

2) 将所有中断号分别绑定到不同的CPU核心上。

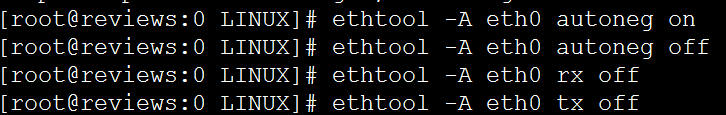


其中“1000”等是CPU核心的掩码16进制表示，换算成二进制为“1 0000 0000 0000”表示核心12。其他核心以此类推。

3) 使用mpstat -I SUM -P ALL 1查看中断分配情况。



1. 关闭网卡的流量控制功能，否则可能会有丢包。



1. 使用测试程序确认收发报文是否OK

Netmap自身带有很多测试程序，最常用的是pkt-gen，可以用来进行发包、收包测试。

可以使用pkt-gen -h查看使用帮助，最常用的几个选项为：

-i：指定收发包网卡。

-f：指定是发包(tx)还是收包(rx)。

-n：指定发送报文数。

-R：指定发送报文速率。

程序退出后，pkt-gen会统计收发的报文数及收发包速率。

**二、TMAS的安装与配置**

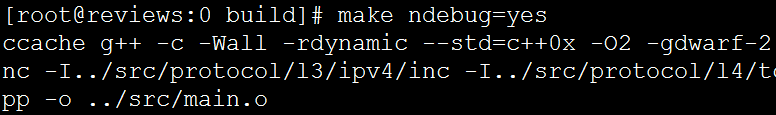
1. 构建TMAS

TMAS的构建非常简单，只需要在build路径下运行make即可，不过有有几个需要注意的地方：

1. 发布版本中，请确认makefile中已经开启了O2优化：

FLAGS = -rdynamic --std=c++0x -O2 -gdwarf-2

1. 如果不需要debug信息，请在make命令中使用ndebug=yes参数：



1. 的
2. 配置TMAS

TMAS的配置文件分为两部分：主配置文件和协议配置文件。

主配置文件为tmas.conf，当前协议配置文件包括tcp.conf和http.conf。

协议配置文件主要用来配置记录数据的规则和内容，除此之外，其他的配置信息都包含在tmas.conf中。

2.1 主配置文件tmas.conf配置详解

1. log-record-type：设置数据记录方式，支持文件记录和数据库记录两种方式。
2. packet-capture-type：报文捕获方式，支持pcap和netmap两种方式，pcap捕获方式一般用来进行功能测试，无序netmap支持；netmap方式为线上运行方式，支持大流量报文捕获。
3. capture-interface：抓包网卡设置，支持配置多张网卡，用空格分隔。
4. capture-filter：报文捕获规则，当前只有pcap报文捕获方式有效。
5. bind-capture-thread-to-cpu：配置报文捕获线程是否绑定到特定CPU核心上，一般来说，设置CPU亲和性后，会减少核间切换开销，性能会有提升（未对比测试）。
6. packet-process-thread-count：设置报文处理线程数量，此参数需要根据当前系统的CPU核心数来配置，请参考《TMAS逻辑架构》。
7. packet-process-fifo-size：无锁队列大小，请参考《TMAS逻辑架构》。
8. packet-dispatch-policy：报文分发策略，当前支持轮询、IP和IP+PORT三种分发方式。分发策略用来解决将报文分发到多个线程处理，为了保证多个线程处理的完全并发，最好将不同连接的报文分发到某一个处理线程中。**注意**：在真实环境中，不要配置轮询分发策略。
9. bind-process-thread-to-cpu：配置报文处理线程是否绑定到特定CPU核心上。
10. batch-dispatch-packet：此配置用来配置报文处理线程从无锁环形队列读取报文的方式，当前支持每次读取一个报文和每次读取尽可能多报文两种方式，经测试，批量读取方式效率更高，所以，每次读取一个报文的方式其实没有存在的价值，建议后续将此配置删除。
11. tcp/udp/http：用来配置支持的协议。
12. enable-ip-checksum：配置是否支持ip层校验和。
13. handshake-timeout：TCP握手超时时间，从收到第一个报文开始，如果在指定时间内未完成握手，则将此连接删除。
14. connection-timeout：TCP连接建立后，如果在指定时间内没有报文交互，则认为连接异常，将此连接关闭。
15. max-cached-unordered-pkt：配置TCP连接最多缓存的无序报文个数，连接如果出现丢包，则可能导致后续报文无法在重构流，此时，再缓存报文已经没有意义。
16. max-cached-pkt-before-handshake：配置TCP握手完成前最多缓存的报文数。
17. [database]域的配置项当前不支持。
18. print-captured-packet-number：打印从网卡上读取的报文数开关。
19. print-received-packet-number：打印分发到处理线程报文数开关。
20. print-interval：打印时间间隔。注意，如果指定了make参数ndebug=yes，则开关不生效。

2.2 协议配置文件详解：

协议配置文件主要用来配置数据记录的规则和内容，当前，tcp和http遵循统一的配置格式——配置域：

filter1 | filter2 | filter3

{

recorder1 on

recorder1-para xxx

recorder2 on

recorder2-para xxx

}

一个配置域包含filter和recorder两部分，filter用来决定哪些数据需要记录，recorder用来觉得记录什么内容，他们二者一起构成了一个配置域。一个配置域可以配置多个filter，中间用“|”分割，不过，有些filter是不能重复配置的，后面会说明。一个配置域中也可以配置多个recorder，用来记录不同的内容，多个recorder各自使用自己的开关，默认是off的。

2.2.1 tcp.conf配置详解

当前支持的filter有：

port：端口过滤器，当前只支持基于范围的配置，如果是单个端口，也需要采用端口范围的配置方式，比如端口80，需要使用80-80的配置方式。TCP连接中，只要有一端的端口包含在此端口范围中，则过滤器通过。

ip：IP地址过滤器，当前只支持基于范围的配置，如果是单个IP地址，也需要采用IP地址范围的配置方式，比如IP地址192.168.0.100，需要使用192.168.0.100-192.168.0.100的配置方式。TCP连接中，只要有一端的IP地址包含在此IP地址范围中，则过滤器通过。

当前支持的recorder有：

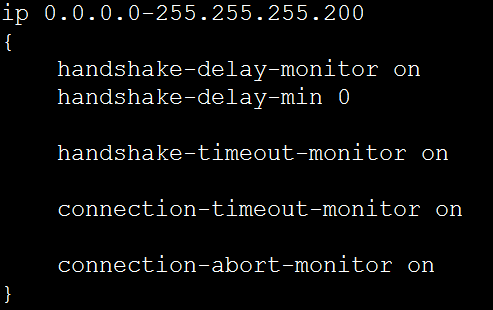
handshake-delay-monitor：用来记录TCP连接的握手时延，只有正常建立连接并正常关闭的连接才会记录，其包含一个记录参数handshake-delay-min，用来设置大于此时延的连接才记录，如果采用文件记录的方式，对应记录文件为tcp\_hs\_delay.log。

handshake-timeout-monitor：用来记录握手超时的连接，记录参数在tmas.conf中，为handshake-timeout。如果采用文件记录的方式，对应的记录文件为tcp\_hs\_timeout.log。

connection-timeout-monitor：用来记录连接建立成功后，但是长时间没有报文交互的连接，记录参数在tmas.conf中，为connection-timeout。如果采用文件记录的方式，对应的记录文件为tcp\_conn\_timeout.log。

connection-abort-monitor：用来记录异常断开的连接，一般是指收到RST的连接。如果采用文件记录的方式，对应的记录文件为tcp\_conn\_abort.log。

配置示例：



2.2.2 HTTP配置详解

当前支持的filter

port：与TCP的port过滤器一致。

ip：与TCP的ip过滤器一致

uri：配置HTTP请求行中的PATH字段，支持正则表达式。

host：配置HTTP请求头中的host字段，支持正则表达式。

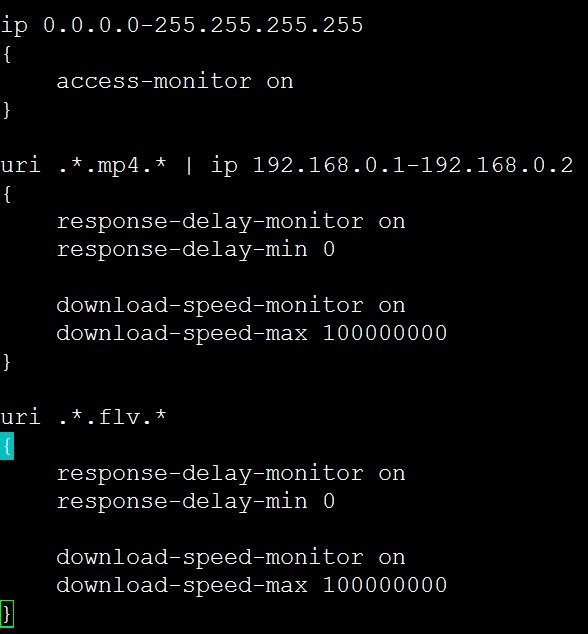
当前支持的recorder

response-delay-monitor：记录HTTP响应时延，其包含一个记录参数response-delay-min，只有响应时延大于此值的HTTP会话才会被记录。如果采用文件记录的方式，对应的记录文件为http\_resp\_delay.log。

download-speed-monitor：记录HTTP下载速度，其包含一个记录参数download-speed-max，只有平均下载速度小于此值的HTTP会话才会被记录。如果采用文件记录的方式，对应的记录文件为http\_dl\_speed.log。

access-monitor：记录访问信息，无记录参数。

配置示例：



1. 运行TMAS

运行TMAS非常简单，当前不需要参数，直接运行即可，不支持deamon。