

netperf

(July,24th,2014 by Shen Guodong)

目录

- 简介
- 编译和安装
- TCP网络性能
- UDP网络性能
- 命令行参数
- 测试网络性能
- 测试脚本
- FAQ
- 参考

简介

Netperf是一种网络性能的测量工具，主要针对基于TCP或UDP的传输。Netperf根据应用的不同，可以进行不同模式的网络性能测试，即批量数据传输（bulk data transfer）模式和请求/应答（request/reponse）模式。Netperf测试结果所反映的是一个系统能够以多快的速度向另外一个系统发送数据，以及另外一个系统能够以多快的速度接收数据。

Netperf工具以client/server方式工作。server端是netserver，用来侦听来自client端的连接，client端是netperf，用来向server发起网络测试。在client与server之间，首先建立一个控制连接，传递有关测试配置的信息，以及测试的结果；在控制连接建立并传递了测试配置信息以后，client与server之间会再建立一个测试连接，用来来回传递着特殊的流量模式，以测试网络的性能。

编译和安装

网址：<http://www.netperf.org/netperf/>

当前最新版本为：netperf-2.6.0

下载源码包->解压->./configure->make->make install。安装完成之后即可使用netperf。

TCP网络性能

由于TCP协议能够提供端到端的可靠传输，因此被大量的网络应用程序使用。但是，可靠性的建立是要付出代价的。TCP协议保证可靠性的措施，如建立并维护连接、控制数据有序的传递等都会消耗一定的网络带宽。

Netperf可以模拟三种不同的TCP流量模式：

1. 单个TCP连接，批量（bulk）传输大量数据

2. 单个TCP连接，client请求/server应答的交易（transaction）方式
3. 多个TCP连接，每个连接中一对请求/应答的交易方式

UDP网络性能

UDP没有建立连接的负担，但是UDP不能保证传输的可靠性，所以使用UDP的应用程序需要自行跟踪每个发出的分组，并重发丢失的分组。

Netperf可以模拟两种UDP的流量模式：

1. 从client到server的单向批量传输
2. 请求/应答的交易方式

由于UDP传输的不可靠性，在使用netperf时要确保发送的缓冲区大小不大于接收缓冲区大小，否则数据会丢失，netperf将给出错误的结果。因此，对于接收到分组的统计不一定准确，需要结合发送分组的统计综合得出结论。

命令行参数

在unix系统中，可以直接运行可执行程序来启动netserver，也可以让inetd或xinetd来自动启动netserver。

当netserver在server端启动以后，就可以在client端运行netperf来测试网络的性能。netperf通过命令行参数来控制测试的类型和具体的测试选项。根据作用范围的不同，netperf的命令行参数可以分为两大类：全局命令行参数、测试相关的局部参数，两者之间使用--分隔，格式如下：

```
netperf [global options]-- [test-specific options]
```

常用的命令行参数：

- H host：指定远端运行netserver的server IP地址；
- l testlen：指定测试的时间长度（秒）；
- t testname：指定测试类型，包括TCP_STREAM, UDP_STREAM, TCP_RR, TCP_CRR, UDP_RR。

其他参数请参考man netperf。

测试网络性能

根据使用传输协议的不同，批量数据传输又分为TCP批量传输和UDP批量传输。

1. TCP_STREAM

测试批量网络流量的性能，执行netperf命令：

```
# netperf -H 192.168.35.34 -l 5
Recv    Send    Send
```

Socket Size	Socket Size	Message Size	Elapsed Time	Throughput
bytes	bytes	bytes	secs.	10^6bits/sec
87380	16384	16384	5.00	940.06

从netperf的结果输出中，我们可以知道以下的一些信息：

- (1) 远端系统（即Server）使用大小为87380字节的socket接收缓冲；
- (2) 本地系统（即Client）使用大小为16384字节的socket发送缓冲；
- (3) 向远端系统发送的测试分组大小为16384字节；
- (4) 测试时间为5秒；
- (5) 吞吐量的测试结果为940.06Mbps/秒

在默认情况下，netperf发送的测试分组大小为本地系统所使用的socket发送缓冲大小。

在TCP_STREAM方式下，相关的局部参数如下表所示：

参数	描述
-s size	设置本地系统的socket发送与接收缓冲大小
-S size	设置远端系统的socket发送与接收缓冲大小
-m size	设置本地系统发送测试分组的大小
-M size	设置远端系统接收测试分组的大小
-D	对本地与远端系统的socket设置TCP_NODELAY选项

通过修改以上的参数，并观察结果的变化，我们可以确定是什么因素影响了连接的吞吐量。

例如，如果怀疑路由器由于缺乏足够的缓冲区空间，使得转发大的分组时存在问题，就可以增加测试分组（-m）的大小，以观察吞吐量的变化：执行netperf命令：

```
# netperf -l 2 -H 192.168.35.34 -- -m 1024
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time   Throughput
bytes bytes bytes secs.  10^6bits/sec

87380 16384 1024  2.00   939.22
```

测试分组的大小由上个例子的16384减少到1024字节，而吞吐量却没有变化，说明路由器不存在缓冲区的问题。

2. TCP_RR

TCP_RR方式的测试对象是多次TCP request和response的交易过程，但是它们发生在同一个TCP连接中，这种模式常常出现在数据库应用中。

执行netperf命令

```
# netperf -l 2 -H 192.168.35.34 -t TCP_RR
Local /Remote
Socket Size   Request  Resp.   Elapsed  Trans.
Send   Recv   Size     Size     Time     Rate
bytes  Bytes  bytes    bytes    secs.    per sec

16384  87380  1         1        2.00     7099.92
16384  87380
```

输出的结果也是两行组成。第一行显示本地系统的情况，第二行显示的是远端系统的信息。平均交易率（transaction rate）为7099.92次/秒。这里的request和response分组的大小都为1个字节，不具有很大的实际意义。用户可以通过相关的参数来改变request和response分组的大小。

TCP_RR方式下的参数如下表所示：

参数	描述
-r req,resp	设置request和reponse分组的大小
-s size	设置本地系统的socket发送与接收缓冲大小
-S size	设置远端系统的socket发送与接收缓冲大小
-D	对本地与远端系统的socket设置TCP_NODELAY选项

执行netperf命令：

```
# netperf -l 2 -H 192.168.35.34 -t TCP_RR -- -r 32,1024
Local /Remote
Socket Size   Request  Resp.   Elapsed  Trans.
Send   Recv   Size     Size     Time     Rate
bytes  Bytes  bytes    bytes    secs.    per sec

16384  87380  32        1024     2.00     5908.93
```

可以看出，由于request/reponse的值增加了，导致交易率明显下降。

3. TCP_CRR

与TCP_RR不同，TCP_CRR为每次交易建立一个新的TCP连接。最典型的应用就是HTTP，每次HTTP交易是在一条单独的TCP连接中进行的。因为需要不停地建立新的TCP连接，并且交易结束后拆除TCP连接，交易率一定会受到很大影响。

执行netperf命令：

```
# netperf -l 2 -H 192.168.35.34 -t TCP_CRR
Local /Remote
Socket Size   Request  Resp.   Elapsed  Trans.
Send   Recv   Size    Size    Time     Rate
bytes  Bytes  bytes   bytes   secs.    per sec

16384  87380  1        1       2.00     2204.16
16384  87380
```

即使是使用一个字节的request/response分组，交易率也明显降低了，只有2204.16次/秒。

TCP_CRR使用与TCP_RR相同的局部参数。

测试脚本

在源码包doc\examples目录下提供了一些脚本，供网络测试使用。如tcp_stream_script, tcp_rr_script等。

FAQ

Q1: 在RHEL 6版本上装netperf，执行make时报如下错误：

```
./configure: line 6295: syntax error near unexpected token `ac_cv_sockaddr_has_sa_len'
./configure: line 6295: `AC_CHECK_SA_LEN(ac_cv_sockaddr_has_sa_len)'
```

A: 将configure文件中的AC_CHECK_SA_LEN(ac_cv_sockaddr_has_sa_len)行注释掉即可。

参考

<http://www.netperf.org/netperf/>

<http://www.cnblogs.com/bluesports/archive/2012/05/08/2489578.html>

http://blog.csdn.net/cheng_fangang/article/details/8456015