网络性能测试工具Iperf介绍

sdnlab.com/2961.html

回到花果山

【概要】Iperf是一款网络性能测试工具，可以方便的用它进行SDN网络带宽和网络质量的测试，Iperf支持协议、定时、缓冲区等参数的配置调整，报告TCP/UDP最大带宽、延迟抖动、数据包丢失等统计信息。

1 Iperf安装

Iperf安装方法有多种，可以下载源码编译安装，也可以直接使用编译好的二进制版本，在ubuntu下安装使用iperf尤为简单，apt-get install iperf 即可，值得一提的是Mininet自带Iperf，在SDN网络上测试比较便捷。

2 工作原理

使用Iperf测试时必须将一台主机设置为客户端，一台主机设置为服务器。

Iperf测试TCP

Iperf测试TCP带宽的原理比较简单，在客户端和服务端建立三次握手连接后，客户端带宽的大小等于发送的总数据除以发送的总时间。对服务端测得的带宽，则是接收的总数据除以所花时间。

TCP模式下简单举例：
Server：iperf -s
Client：iperf -c 10.0.0.1 -i 1

客户端到服务器10.0.0.1上带宽测试，每一秒钟打印一次信息。

Iperf测试UDP

Iperf测试UDP性能时，客户端可以指定UDP数据流的速率。客户端发送数据时，将根据客户端提供的速率计算数据报发送之间的时延。

客户端还可以指定发送数据报的大小。每个发送的数据报包含一个ID号，用来唯一标识报文，服务器端根据该ID号来确定数据报丢失和乱序。

当把UDP报文大小设置可以将整个报文放入IP层的包(packet)内时，那么UDP所测得的报文丢失数据即为IP层包的丢失数据，这提供了一个有效的测试包丢失情况的方法。

数据报传输延迟抖动 (Jitter)的测试由服务器端完成，客户发送的报文数据包含有发送时间戳，服务器端根据该时间信息和接收到报文的时间戳来计算传输延迟抖动。传输延迟抖动反映传输过程中是否平滑。由于它是一个相对值，所以并不需要客户端和服务器端时间同步。

UDP模式下简单举例：
Server：iperf -s -u
Client：iperf -c 10.0.0.1 -u -b 100M

在UDP模式下，客户端以100Mbps为数据发送速率，测试客户端到服务器10.0.0.1上的带宽。

综上，用以下方法测试网络连接的质量：
- 延时（反应时间或者RTT）：用ping命令量度
- Jitter（延时变化）：用Iperf UDP测试来量度
- 数据报丢失：用Iperf UDP测试来量度
- 带宽：通过TCP测试来量度

3 参数配置

表 1. 客户端/服务器端通用参数

参数	作用
-f/--format	[kmKM] 分别表示以 Kbits, Mbits, KBytes, MBytes 显示报告，默认以 Mbits 为单位
-i/--interval	以秒为单位显示报告间隔
-l/--len	读写的缓冲区大小，默认 8KB
-m/--print_mss	打印最大的 TCP 数据段大小 (MTU - TCP/IP header)
-o/--output	将报告和错误信息输出到指定文件
-p/--port	指定服务器端使用的端口或客户端所连接的端口
-u/--udp	使用 udp 协议
-w/--window	指定 TCP 窗口(socket 缓冲区)大小 ,默认是 8KB
-B/--bind	绑定一个主机地址或接口
-C/--compatibility	兼容旧版本(当 server 端和 client 端版本不一样时使用)
-M/--mss	设置 TCP 最大数据段大小 (MTU - 40 bytes)
-N/--nodelay	设置无延迟 TCP，禁用 Nagle's Algorithm
-V/--IPv6Version	设置传输 Ipv6 数据包

表 2. 服务器端参数

参数	作用
-s/--server	服务器模式下运行
-U/-- single_udp	单线程 UDP 模式下运行
-D/-- daemon	以守护进程模式运行服务器

表 3. 客户端参数

参数	作用
-b/-- bandwidth	对于 UDP，使用 bits/sec (默认 1 Mbit/sec，亦即 -u)传送带宽
-c/-- client	运行为客户端，连接到“主机”
-d/-- dualtest	同步进行双向测试
-n/-- num	传输的字节量
-r/-- tradeoff	分别进行双向测试
-t/-- time	传输持续时间 (默认 10 secs)
-F/-- fileinput	从文件中读取要传输的数据
-I/-- stdin	从标准输入（stdin）中读取要传输的数据
-L/-- listenport	双向测试接受端口
-P/-- parallel	并行客户线程数量
-T/-- ttl	多点传送的生存时间 (默认 1)

4 测试实例

测试环境

1） 已搭建好一套SDN网络环境，使用Mininet模拟交换机和主机连接到一款OpenFlow控制器
2） 使用Mininet自带的Iperf工具

应用实例

实例一：最简参数实例

服务端：

```
root@ubuntu243:~# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 33679
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  17.3 GBytes  14.8 Gbits/sec
```

客户端：

```
root@ubuntu243:~# iperf -c 10.0.0.1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 33679 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  17.3 GBytes  14.8 Gbits/sec
```

Iperf客户端连接Iperf服务器的TCP默认端口5001，否则我们可以用-p参数修改Iperf服务器的端口，客户端与服务器必须加上同样的端口。结果显示的带宽是从用户到服务器之间的带宽。

实例二：双向带宽测试

服务端：

```
root@ubuntu243:~# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 33680
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  17.4 GBytes  14.9 Gbits/sec
------------------------------------------------------------
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 1.34 MByte (default)
------------------------------------------------------------
[  4] local 10.0.0.1 port 45677 connected with 10.0.0.2 port 5001
[  4]  0.0-10.0 sec  18.2 GBytes  15.7 Gbits/sec
```

客户端：

```
root@ubuntu243:~# iperf -c 10.0.0.1 -r
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size:  280 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.2 port 33680 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  17.4 GBytes  14.9 Gbits/sec
[  4] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 45677
[  4]  0.0-10.0 sec  18.2 GBytes  15.7 Gbits/sec
```

-r参数可以量度双向带宽，Iperf服务器会主动向客户端发起连接。

实例三：同步双向带宽测试

服务端：

```
root@ubuntu243:~# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 33682
------------------------------------------------------------
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  6] local 10.0.0.1 port 45679 connected with 10.0.0.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  6]  0.0-10.0 sec  8.14 GBytes  6.99 Gbits/sec
[  4]  0.0-10.0 sec  8.72 GBytes  7.48 Gbits/sec
```

客户端：

```
root@ubuntu243:~# iperf -c 10.0.0.1 -d
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 1.10 MByte (default)
------------------------------------------------------------
[  5] local 10.0.0.2 port 33682 connected with 10.0.0.1 port 5001
[  4] local 10.0.0.2 port 5001 connected with 10.0.0.1 port 45679
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  8.72 GBytes  7.49 Gbits/sec
[  4]  0.0-10.0 sec  8.14 GBytes  6.99 Gbits/sec
```

使用-d 参数同步测量双向带宽，而上例的-r在初始状态时，只会量度客户到服务器的带宽。

实例四：TCP窗口大小

服务端：

```
root@ubuntu243:~# iperf -s -w 3000
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 5.86 KByte (WARNING: requested 2.93 KByte)
------------------------------------------------------------
[  4] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 33685
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec   514 MBytes   431 Mbits/sec
```

客户端：

```
root@ubuntu243:~# iperf -c 10.0.0.1 -w 1500
WARNING: TCP window size set to 1500 bytes. A small window size
will give poor performance. See the Iperf documentation.
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 4.50 KByte (WARNING: requested 1.46 KByte)
------------------------------------------------------------
[  3] local 10.0.0.2 port 33685 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   514 MBytes   431 Mbits/sec
```

在连接中，如果接收方来不及验证，数据会暂时被存在一个缓冲区里，这个缓冲区的上限就是所谓的TCP窗口大小，窗口的大小可以在2到65,535 bytes之间。

实例五：UDP测试

UDP测试会得到关于Jitter和数据包丢失的重要信息。

服务端：

```
root@ubuntu243:~# iperf -s -u -i 1
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 57464
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  12.0 MBytes   101 Mbits/sec   0.007 ms    1/ 8588 (0.012%)
[  3]  0.0- 1.0 sec  29 datagrams received out-of-order
[  3]  1.0- 2.0 sec  12.0 MBytes   101 Mbits/sec   0.004 ms    0/ 8547 (0%)
[  3]  2.0- 3.0 sec  12.0 MBytes   101 Mbits/sec   0.008 ms    0/ 8547 (0%)
[  3]  3.0- 4.0 sec  12.0 MBytes   101 Mbits/sec   0.004 ms    0/ 8546 (0%)
[  3]  4.0- 5.0 sec  12.0 MBytes   101 Mbits/sec   0.003 ms    0/ 8547 (0%)
[  3]  5.0- 6.0 sec  12.0 MBytes   101 Mbits/sec   0.009 ms    0/ 8547 (0%)
[  3]  6.0- 7.0 sec  12.0 MBytes   101 Mbits/sec   0.004 ms    0/ 8547 (0%)
[  3]  7.0- 8.0 sec  12.0 MBytes   101 Mbits/sec   0.003 ms    0/ 8546 (0%)
[  3]  8.0- 9.0 sec  12.0 MBytes   101 Mbits/sec   0.003 ms    0/ 8547 (0%)
[  3]  0.0-10.0 sec   120 MBytes   101 Mbits/sec   0.002 ms   30/85468 (0.035%)
[  3]  0.0-10.0 sec  30 datagrams received out-of-order
```

客户端：

```
root@ubuntu243:~# iperf -c 10.0.0.1 -u -b 100M
------------------------------------------------------------
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 57464 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   120 MBytes   101 Mbits/sec
[  3] Sent 85469 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec   120 MBytes   101 Mbits/sec   0.001 ms   30/85468 (0.035%)
[  3]  0.0-10.0 sec  30 datagrams received out-of-order
```

良好的连接质量，数据包的丢失率要小于1%，数据包的丢失率过高会导致许多TCP数据报需要重传，从而影响带宽。 Jitter代表的是延时变化，并不依赖于延时本身，你可能拥有很长的反应时间，但是Jitter却很低。

5 结束语

Jperf与Iperf一起使用的话可以提供一个java写成的图形界面，大家可以自行学习，这里就不做介绍。如有不足之处请指正，谢谢。