

Openstack 之 SDN 性能测试具体实施及结果

周威光整理^{*}

2017-07-21

^{*} 简介：恒天云 FTE

目 录

1	网络性能测试指标	3
2	网络性能测试工具选择	3
2.1	常见的网络性能测试工具简介	3
2.2	测试方法确定	4
3	网络性能测试工具 iperf 的使用	4
3.1	使用原理	4
3.2	iperf 使用方法与参数说明	4
3.3	tcp 包测试带宽	5
3.4	udp 包同时测试抖动和丢包	5
4	ping 命令的使用	6
4.1	ping 基本格式	6
4.2	icmp 包测试延迟	6

1 网络性能测试指标

常见的网络性能测试指标包含：网络吞吐量 (Throughput)、网络延迟 (latency)、抖动 (jitter)、丢包率等

1. 网络吞吐量：单位时间内通过某个网络（或信道、接口）的数据量，吞吐量受网络的带宽或者网络的额定速率限制的，例如家庭带宽为 10M 网络，表明网络吞吐量不可能超过 10Mbits/s，吞吐量的单位通常表示为位元每秒 (bit/s 或 bps)。
2. 网络延迟：通俗的讲，就是数据从电脑这边传到那边所用的时间。这儿有个问题需要确认，数据是指一个数据包的传输还是任意大小，和你传输的数据量相关。可以明显的看到，从 A 到 B 传送 1 个字节的时间和传送 100MB 的时间肯定是不一样的。标准意义上的延迟，应该仅仅指 1 个字节的传输时间，类似网络课上讲到的传播时延。（不同意见欢迎讨论）。同样存在一个名词叫做传播延时，这个应该可以标识整个数据包的传输时间，不论包大小为多少。
3. 抖动：用于描述包在网络中的传输延时的变化，抖动越小，说明网络质量越稳定越好。抖动是评价一个网络性能的最重要的因素。
4. 丢包率：测试中所丢失的数据包数量占所发送的数据包的比率，因为我们知道 TCP 协议是可靠的，所以，一般在使用 UDP 传输时，才会统计丢包率。

2 网络性能测试工具选择

2.1 常见的网络性能测试工具简介

常用的开源网络性能测试工具有两个：iperf 和 netperf，iperf 是美国伊利诺斯大学 (University of Illinois) 开发的一种开源的网络性能测试工具，netperf 是由惠普公司开发的一种网络性能的测量工具，测试网络栈。这两种工具都可以测试 TCP 协议和 UDP 协议，从可测试的网络性能指标，我们对两种工具进行下对比：

工具	带宽	网络延迟	抖动	丢包
iperf	是	否	是	是
netperf	是	是	是	否

可见 iperf 和 netperf 都可以完成基本的网络性能测试，但 netperf 更倾向于测试不同网络模式的数据传输，与本次性能测试需求不符。而 iperf 可以经过简单的参数设置，比较直观的给出带宽、抖动和丢包，并且还能设置测试时间、发送包的大小，以及带宽。虽然无法对网络延迟进行统一测试，但可以使用 ping 进行弥补

2.2 测试方法确定

测试对象	带宽	延时	抖动 (Jitter)	丢包
测试工具	iperf tcp 包测试	ping 测试	iperf UDP 测试	iperf UDP 测试

3 网络性能测试工具 iperf 的使用

3.1 使用原理

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。

3.2 iperf 使用方法与参数说明

```

1  参数说明
2  -s 以server模式启动, eg: iperf -s
3  -c host以client模式启动, host是server端地址, eg: iperf -c 222.35.11.23
4
5  通用参数
6  -f [knKM] 分别表示以Kbits, Mbits, KBytes, MBytes显示报告, 默认以Mbits为单位, eg:
   iperf -c 222.35.11.23 -f K
7  -i sec 以秒为单位显示报告间隔, eg: iperf -c 222.35.11.23 -i 2
8  -l 缓冲区大小, 默认是8KB, eg: iperf -c 222.35.11.23 -l 16
9  -m 显示tcp最大mtu值
10 -o 将报告和错误信息输出到文件 eg: iperf -c 222.35.11.23 -o ciperflog.txt
11 -p 指定服务器端使用的端口或客户端所连接的端口 eg: iperf -s -p 9999; iperf -c
    222.35.11.23 -p 9999
12 -u 使用udp协议
13 -w 指定TCP窗口大小, 默认是8KB
14 -B 绑定一个主机地址或接口 (当主机有多个地址或接口时使用该参数)
15 -C 兼容旧版本 (当server端和client端版本不一样时使用)
16 -M 设定TCP数据包的最大mtu值
17 -N 设定TCP不延时
18 -V 传输ipv6数据包
19
20 server专用参数
21 -D 以服务方式运行iperf, eg: iperf -s -D
22 -R 停止iperf服务, 针对-D, eg: iperf -s -R
23
24 client端专用参数
25 -d 同时进行双向传输测试
26 -n 指定传输的字节数, eg: iperf -c 222.35.11.23 -n 100000
27 -r 单独进行双向传输测试
28 -t 测试时间, 默认10秒, eg: iperf -c 222.35.11.23 -t 5
29 -F 指定需要传输的文件
30 -T 指定ttl值

```

3.3 tcp 包测试带宽

- 1 iperf -s #服务器默认等待接收tcp数据包
- 2 iperf -c 172.16.133.10 -i 1 #客户端向服务器默认发送tcp数据包

服务器给出的数据如下:

```
root@hty-controller:~# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

[ 4] local 172.16.133.10 port 5001 connected with 172.16.133.15 port 54719
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.0 sec  1.10 GBytes  941 Mbits/sec
```

客户端给出的数据如下:

```
root@hty-compute1:~# iperf -c 172.16.133.10 -i 1
Client connecting to 172.16.133.10, TCP port 5001
TCP window size: 85.0 KByte (default)

[ 3] local 172.16.133.15 port 54719 connected with 172.16.133.10 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec   114 MBytes  956 Mbits/sec
[ 3] 1.0- 2.0 sec   112 MBytes  942 Mbits/sec
[ 3] 2.0- 3.0 sec   112 MBytes  942 Mbits/sec
[ 3] 3.0- 4.0 sec   112 MBytes  942 Mbits/sec
[ 3] 4.0- 5.0 sec   113 MBytes  947 Mbits/sec
[ 3] 5.0- 6.0 sec   112 MBytes  937 Mbits/sec
[ 3] 6.0- 7.0 sec   113 MBytes  948 Mbits/sec
[ 3] 7.0- 8.0 sec   112 MBytes  942 Mbits/sec
[ 3] 8.0- 9.0 sec   111 MBytes  934 Mbits/sec
[ 3] 9.0-10.0 sec   113 MBytes  950 Mbits/sec
[ 3] 0.0-10.0 sec   1.10 GBytes  943 Mbits/sec
```

3.4 udp 包同时测试抖动和丢包

- 1 iperf -s -u -i 1 #服务器等待接收udp数据包, 每隔一秒钟显示信息
- 2 iperf -c 172.16.133.10 -u -i 1 #客户端向服务器发送udp数据包, 每隔一秒钟显示信息

服务器给出的数据如下:

```

root@hty-controller:~ x root@hty-compute1:~ x root@hty-com
root@hty-controller:~# iperf -s -u -i 1
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 172.16.133.10 port 5001 connected with 172.16.133.15 port 40711
[ ID] Interval      Transfer     Bandwidth   Jitter    Lost/Total Datagrams
[ 3] 0.0- 1.0 sec   128 KBytes  1.05 Mbits/sec  0.013 ms  0/ 89 (0%)
[ 3] 1.0- 2.0 sec   128 KBytes  1.05 Mbits/sec  0.013 ms  0/ 89 (0%)
[ 3] 2.0- 3.0 sec   128 KBytes  1.05 Mbits/sec  0.012 ms  0/ 89 (0%)
[ 3] 3.0- 4.0 sec   128 KBytes  1.05 Mbits/sec  0.012 ms  0/ 89 (0%)
[ 3] 4.0- 5.0 sec   128 KBytes  1.05 Mbits/sec  0.012 ms  0/ 89 (0%)
[ 3] 5.0- 6.0 sec   129 KBytes  1.06 Mbits/sec  0.012 ms  0/ 90 (0%)
[ 3] 6.0- 7.0 sec   128 KBytes  1.05 Mbits/sec  0.014 ms  0/ 89 (0%)
[ 3] 7.0- 8.0 sec   128 KBytes  1.05 Mbits/sec  0.014 ms  0/ 89 (0%)
[ 3] 8.0- 9.0 sec   128 KBytes  1.05 Mbits/sec  0.013 ms  0/ 89 (0%)
[ 3] 9.0-10.0 sec   128 KBytes  1.05 Mbits/sec  0.012 ms  0/ 89 (0%)
[ 3] 0.0-10.0 sec   1.25 MBytes 1.05 Mbits/sec  0.013 ms  0/ 893 (0%)

```

客户端给出的数据如下:

```

root@hty-controller:~ x root@hty-compute1:~ x root@hty-c
root@hty-compute1:~# iperf -c 172.16.133.10 -u -i 1
-----
Client connecting to 172.16.133.10, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 172.16.133.15 port 40711 connected with 172.16.133.10 port 5001
[ ID] Interval      Transfer     Bandwidth   TCP window size: 65.3 KByte (default)
[ 3] 0.0- 1.0 sec   129 KBytes  1.06 Mbits/sec
[ 3] 1.0- 2.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 2.0- 3.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 3.0- 4.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 4.0- 5.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 5.0- 6.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 6.0- 7.0 sec   129 KBytes  1.06 Mbits/sec
[ 3] 7.0- 8.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 8.0- 9.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 9.0-10.0 sec   128 KBytes  1.05 Mbits/sec
[ 3] 0.0-10.0 sec   1.25 MBytes 1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec   1.25 MBytes 1.05 Mbits/sec  0.012 ms  0/ 893 (0%)
root@hty-compute1:~#

```

4 ping 命令的使用

4.1 ping 使用方法与参数说明

- 1 语法:
- 2 ping(选项)(参数)
- 3
- 4 选项:
- 5 -d: 使用Socket的SO_DEBUG功能;
- 6 -c<完成次数>: 设置完成要求回应的次数;
- 7 -f: 极限检测;
- 8 -i<间隔秒数>: 指定收发信息的间隔时间;
- 9 -I<网络界面>: 使用指定的网络界面送出数据包;

```

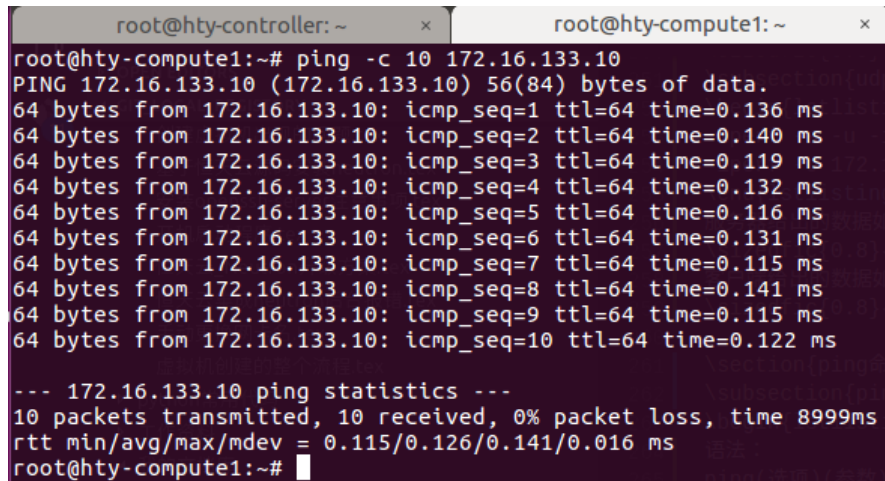
10 -l<前置载入>: 设置在送出要求信息之前, 先行发出的数据包;
11 -n: 只输出数值; -p<范本样式>: 设置填满数据包的范本样式;
12 -q: 不显示指令执行过程, 开头和结尾的相关信息除外;
13 -r: 忽略普通的Routing Table, 直接将数据包送到远端主机上;
14 -R: 记录路由过程; -s<数据包大小>: 设置数据包的大小;
15 -t<存活数值>: 设置存活数值TTL的大小;
16 -v
17
18 参数:
19 目的主机: 指定发送ICMP报文的的目的主机。

```

4.2 icmp 包测试延迟

```
1 ping -c 10 172.16.133.10
```

ping 服务器显示如下:



```

root@hty-controller: ~ x root@hty-compute1: ~ x
root@hty-compute1:~# ping -c 10 172.16.133.10
PING 172.16.133.10 (172.16.133.10) 56(84) bytes of data.
64 bytes from 172.16.133.10: icmp_seq=1 ttl=64 time=0.136 ms
64 bytes from 172.16.133.10: icmp_seq=2 ttl=64 time=0.140 ms
64 bytes from 172.16.133.10: icmp_seq=3 ttl=64 time=0.119 ms
64 bytes from 172.16.133.10: icmp_seq=4 ttl=64 time=0.132 ms
64 bytes from 172.16.133.10: icmp_seq=5 ttl=64 time=0.116 ms
64 bytes from 172.16.133.10: icmp_seq=6 ttl=64 time=0.131 ms
64 bytes from 172.16.133.10: icmp_seq=7 ttl=64 time=0.115 ms
64 bytes from 172.16.133.10: icmp_seq=8 ttl=64 time=0.141 ms
64 bytes from 172.16.133.10: icmp_seq=9 ttl=64 time=0.115 ms
64 bytes from 172.16.133.10: icmp_seq=10 ttl=64 time=0.122 ms

--- 172.16.133.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8999ms
rtt min/avg/max/mdev = 0.115/0.126/0.141/0.016 ms
root@hty-compute1:~#

```

参考文献

- [1] [网络性能测试工具 Iperf 介绍](#)
- [2] [使用 iperf 测试网络性能](#)