

iperf

网络性能测试工具

补充说明

iperf命令 是一个网络性能测试工具。iperf可以测试TCP和UDP带宽质量。iperf可以测量最大TCP带宽，具有多种参数和UDP特性。iperf可以报告带宽，延迟抖动和数据包丢失。利用iperf这一特性，可以用来测试一些网络设备如路由器，防火墙，交换机等的性能。

iperf分为两种版本。Unix/Linux版和Windows版，Unix/Linux版更新比较快，版本最新。Windows版更新慢。Windows版的iperf叫jperf或者xjperf。jperf是在iperf基础上开发了更好的UI和新的功能。

Linux版本下载地址：<http://code.google.com/p/iperf/downloads/list>

安装iperf

对于windows版的iperf，直接将解压出来的iperf.exe和cygwin1.dll复制到%systemroot%目录即可，对于linux版的iperf，请使用如下命令安装：

```
gunzip -c iperf-<version>.tar.gz | tar -xvf -
cd iperf-<version>
./configure
make
make install
```

选项

```
<table>
<tbody>
<tr>
<th>命令行选项</th>
<th>描述</th>
</tr>
<tr>
<td>客户端与服务器共用选项</td>
</tr>
<tr>
<td>-f, --format [bkmaBKMA]</td>
<td>格式化带宽数输出。支持的格式有：
'b' = bits/sec 'B' = Bytes/sec
'k' = Kbits/sec 'K' = KBytes/sec
'm' = Mbits/sec 'M' = MBytes/sec
'g' = Gbits/sec 'G' = GBytes/sec
'a' = adaptive bits/sec 'A' = adaptive Bytes/sec
自适应格式是kilo-和mega-二者之一。除了带宽之外的字段都输出为字节，除非指定输出的格式，默认的参数是a
注意：在计算字节byte时，Kilo = 1024 Mega = 1024^2 Giga = 1024^3 通常，在网络中 Kilo = 1000 Mega = 1000^2 and Giga = 1000^3 所以 Iperf 也按此来计算比特（位）。如果这些困扰了你，那么请使用-f b参数，然后亲自计算一下。</td>
</tr>
```

<tr> <td>-i, --interval #</td> <td>设置每次报告之间的时间间隔，单位为秒。如果设置为非零值，就会按照此时间间隔输出测试报告。 默认值为零。</td> </tr> <tr> <td>-l, --len #[KM]</td> <td>设置读写缓冲区的长度□TCP方式默认为8KB□UDP方式默认为1470字节。</td> </tr> <tr> <td>-m, --print_mss</td> <td>输出TCP MSS值（通过TCP_MAXSEG支持□MSS值一般比MTU值小40字节。通常情况</td> </tr> <tr> <td>-p, --port #</td> <td>设置端口，与服务器端的监听端口一致。默认是5001端口，与ttcp的一样。</td> </tr> <tr> <td>-u, --udp</td> <td>使用UDP方式而不是TCP方式。参看-b选项。</td> </tr> <tr> <td>-w, --window #[KM]</td> <td>设置套接字缓冲区为指定大小。对于TCP方式，此设置为TCP窗口大小。对于UDP方式，此设置为 接受UDP数据包的缓冲区大小，限制可以接受数据包的最大值。</td> </tr> <tr> <td>-B, --bind host</td> <td>绑定到主机的多个地址中的一个。对于客户端来说，这个参数设置了出栈接口。对于服务器端来说， 这个参数设置入栈接口。这个参数只用于具有多网络接口的主机。在Iperf的UDP模式下，此参数用于 绑定和加入一个多播组。使用范围在224.0.0.0至239.255.255.255的多播地址。参考-T参数。</td> </tr> <tr> <td>-C, --compatibility</td> <td>与低版本的Iperf使用时，可以使用兼容模式。不需要两端同时使用兼容模式，但是强烈推荐两端 同时使用兼容模式。某些情况下，使用某些数据流可以引起1.7版本的服务器端崩溃或引起非预期的连接 尝试。</td> </tr> <tr> <td>-M, --mss #ip头减去40字节。在以太网中□MSS值为1460字节□MTU1500字节）。许多操作系统 不支持此选项。</td> </tr> <tr> <td>-N, --nodelay</td> <td>设置TCP无延迟选项，禁用Nagle's运算法则。通常情况此选项对于交互程序，例如telnet□是禁 用的。</td> </tr> <tr> <td>-V (from v1.6 or higher)</td> <td>绑定一个IPv6地址。 服务端□\$ iperf -s -V
--

客户端\$ iperf -c <Server IPv6 Address> -V

注意：在1.6.3或更高版本中，指定IPv6地址不需要使用-B参数绑定，在1.6之前的版本则需要。在大多数操作系统中，将响应IPv4客户端映射的IPv4地址。</td>

</tr>

<tr>

<td>服务器端专用选项</td>

</tr>

<tr>

<td>-s, --server</td>

<td>Iperf服务器模式</td>

</tr>

<tr>

<td>-D (v1.2或更高版本)</td>

<td>Unix平台下Iperf作为后台守护进程运行。在Win32平台下Iperf将作为服务运行。</td>

</tr>

<tr>

<td>-R(v1.2或更高版本，仅用于Windows)</td>

<td>卸载Iperf服务（如果它在运行）。</td>

</tr>

<tr>

<td>-o(v1.2或更高版本，仅用于Windows)</td>

<td>重定向输出到指定文件</td>

</tr>

<tr>

<td>-c, --client host</td>

<td>如果Iperf运行在服务器模式，并且用-c参数指定一个主机，那么Iperf将只接受指定主机的连接。此参数不能工作于UDP模式。</td>

</tr>

<tr>

<td>-P, --parallel #</td>

<td>服务器关闭之前保持的连接数。默认是0，这意味着永远接受连接。</td>

</tr>

<tr>

<td>客户端专用选项</td>

</tr>

<tr>

<td>-b, --bandwidth #[KM]</td>

<td>UDP模式使用的带宽，单位bits/sec此选项与-u选项相关。默认值是1 Mbit/sec</td>

</tr>

<tr>

<td>-c, --client host</td>

<td>运行Iperf的客户端模式，连接到指定的Iperf服务器端。</td>

</tr>

<tr>

<td>-d, --dualtest</td>

<td>运行双测试模式。这将使服务器端反向连接到客户端，使用-L 参数中指定的端口（或默认使用客户端连接到服务器端的端口）。这些在操作的同时就立即完成了。如果你想要一个交互的测试，请尝试-r参数。</td>

</tr>

<tr>

<td>-n, --num #[KM]</td>

<td>传送的缓冲器数量。通常情况Iperf按照10秒钟发送数据-n参数跨越此限制，按照指定次数发送指定长度的数据，而不论该操作耗费多少时间。参考-l与-t选项。**</td>**

</tr>

<tr>

<td>-r, --tradeoff</td>

<td>往复测试模式。当客户端到服务器端的测试结束时，服务器端通过-l选项指定的端口（或默认为客户端连接到服务器端的端口），反向连接至客户端。当客户端连接终止时，反向连接随即开始。如果需要同时进行双向测试，请尝试-d参数。**</td>**

</tr>

<tr>

<td>-t, --time #</td>

<td>设置传输的总时间Iperf在指定的时间内，重复的发送指定长度的数据包。默认是10秒钟。参考-l与-n选项。**</td>**

</tr>

<tr>

<td>-L, --listenport #</td>

<td>指定服务端反向连接到客户端时使用的端口。默认使用客户端连接至服务端的端口。**</td>**

</tr>

<tr>

<td>-P, --parallel #</td>

<td>线程数。指定客户端与服务端之间使用的线程数。默认是1线程。需要客户端与服务端同时使用此参数。**</td>**

</tr>

<tr>

<td>-S, --tos #</td>

<td>出栈数据包的服务类型。许多路由器忽略TOS字段。你可以指定这个值，使用以"0x"开始的16进制数，或以"0"开始的8进制数或10进制数。

例如，16进制'0x10' = 8进制'020' = 十进制'16' TOS值1349就是：

IPTOS_LOWDELAY minimize delay 0x10

IPTOS_THROUGHPUT maximize throughput 0x08

IPTOS_RELIABILITY maximize reliability 0x04

IPTOS_LOWCOST minimize cost 0x02**</td>**

</tr>

<tr>

<td>-T, --ttl #</td>

<td>出栈多播数据包的TTL值。这本质上就是数据通过路由器的跳数。默认是1，链接本地。**</td>**

</tr>

<tr>

<td>-F (from v1.2 or higher)</td>

<td>使用特定的数据流测量带宽，例如指定的文件。

\$ iperf -c <server address> -F <file-name>**</td>**

</tr>

<tr>

<td>-I (from v1.2 or higher)</td>

<td>与-F一样，由标准输入输出文件输入数据。**</td>**

</tr>

<tr>

<td>杂项</td>

</tr>

<tr>

<td>-h, --help</td>

```
<td>显示命令行参考并退出。</td>
</tr>
<tr>
<td>-v, --version</td>
<td>显示版本信息和编译信息并退出。</td>
</tr>
</tbody>
</table>
```

实例

带宽测试通常采用UDP模式，因为能测出极限带宽、时延抖动、丢包率。在进行测试时，首先以链路理论带宽作为数据发送速率进行测试，例如，从客户端到服务器之间的链路的理论带宽为100Mbps，先用-b 100M进行测试，然后根据测试结果（包括实际带宽，时延抖动和丢包率），再以实际带宽作为数据发送速率进行测试，会发现时延抖动和丢包率比第一次好很多，重复测试几次，就能得出稳定的实际带宽。

UDP模式

服务器端：

```
iperf -u -s
```

客户端：

```
iperf -u -c 192.168.1.1 -b 100M -t 60
```

在udp模式下，以100Mbps为数据发送速率，客户端到服务器192.168.1.1上传带宽测试，测试时间为60秒。

```
iperf -u -c 192.168.1.1 -b 5M -P 30 -t 60
```

客户端同时向服务器端发起30个连接线程，以5Mbps为数据发送速率。

```
iperf -u -c 192.168.1.1 -b 100M -d -t 60
```

以100M为数据发送速率，进行上下行带宽测试。

TCP模式

服务器端：

```
iperf -s
```

客户端：

```
iperf -c 192.168.1.1 -t 60
```

在tcp模式下，客户端到服务器192.168.1.1上传带宽测试，测试时间为60秒。

```
iperf -c 192.168.1.1 -P 30 -t 60
```

客户端同时向服务器端发起30个连接线程。

```
iperf -c 192.168.1.1 -d -t 60
```

进行上下行带宽测试。

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:iperf>

Last update: **2021/10/15 14:58**

