IPERF/JPERF

Network Bandwidth performance testing tool

ABSTRACT
Iperf is a tool to measure network performance for measuring bandwidth provided by the network. Iperf is a tool to measure maximum TCP bandwidth, allowing the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss.
Overview

The purpose of this document is to help students understand the fundamentals of using Iperf for analyzing network performance by calculating the bandwidth performance. This tool measures bandwidth by allowing tuning of various TCP parameters and UDP characteristics. This tool reports bandwidth, delay jitter and datagram loss while performing the tests.

In order to simplify the execution of Iperf on clients we can use many types of Graphical User Interface (GUI). JPerf is one such utility used widely which we will be using in this document to test Iperf.

Requirements

The following equipment are required to perform the tests:

- 2 or more machines provided with ethernet or Wi-Fi network interface with the following supported Operating Systems (OS):
  - Windows 2000, XP, 2003, Vista, 7, 8, 8.1
  - MacOS X
  - Linux 32 bits (i386)/64 bits (AMD64)
  - Oracle Solaris
  - Android
- Wired/Wireless Network to which the devices are connected
  - A simple wireless router would be enough as we will see in this test

Instructor Notes

JPerf utility is a java based GUI based on Iperf network bandwidth measurement tool. JPerf 2.0.2 version can be downloaded from the following links for windows and MAC:

Windows: https://code.google.com/p/xjperf/downloads/list


Also, in order to run this utility you might be required to download Java SE Runtime Environment which can be downloaded from the following links by accepting the license agreement and selecting the appropriate Operating system running on the device

http://jesterpm.net/downloads

Iperf can also run on Android devices by installing the iPerf for Android app, which can be found at the following link: https://play.google.com/store/apps/details?id=com.magicandroidapps.iperf
iPerf/JPerf Overview

Performance of a network can be tested by using iPerf for both wired and wireless interfaces. This can be done in various ways which include both devices on wired interface, both devices on wireless interface and either device on wired and wireless interface. While performing these tests we need to define one of the devices as server which transmits TCP and UDP data packets to the clients.

While there can be various combination of server and clients located on wireless or wired interface, one of the most preferred way to test wireless network is to connect server running iPerf using a wired link to the wireless router and then connecting a wireless client to this server using iPerf to test downstream traffic. However, depending on the type of network analysis to be done any type of setup can be created. In this document we have connected both the server and clients on the same wireless network such that all of them belong to the same subnet and can see each other. An initial ping from any device to all devices under test with their address is a good idea to verify this.

iPerf provides with a high set of operational functionality including connecting various clients to a single server. This is particularly important when multiple client testing is required to be performed.

JPerf Screen

Once the Jperf utility is being downloaded from the above provided link along with java runtime if required, the user might be able to run the utility by executing the jperf.bat (batch) file from the downloaded location in windows.

Similarly MAC users might just open the JPerf-2.0.2.dmg (disk image) and then opening the created drive from desktop and then running jperf.app. This will start the JPerf utility program and looks like this when running:
According to the desired role of the machine the user might select server or client from the upper part of the screen, notice the generated commands while making the selection:

**Server:**

**Client:**
Notice the command generated for server: `bin/iperf.exe -s -P 0 -i 1 -p 5001 -f k`

This command is actually the iPerf command line to be executed by the JPerf utility which can be broken down into following simple parameters:

**Server:**

`bin/iperf.exe -s -P 0 -i 1 -p 5001 -f k`

<table>
<thead>
<tr>
<th>Command line option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s</td>
<td>Run Iperf in server mode</td>
</tr>
<tr>
<td>-P 0</td>
<td>The number of connections to handle by the server before closing. Default is 0 (which means to accept connections forever)</td>
</tr>
<tr>
<td>-i 1</td>
<td>Sets the interval time in seconds between periodic bandwidth, jitter, and loss reports. If non-zero, a report is made every interval seconds of the bandwidth since the last report. If zero, no periodic reports are printed. Default is zero.</td>
</tr>
<tr>
<td>-p 5001</td>
<td>The server port for the server to listen on and the client to connect to. This should be the same in both client and server. Default is 5001, the same as ttcp.</td>
</tr>
<tr>
<td>-f k</td>
<td>A letter specifying the format to print bandwidth numbers in. here 'k' = Kbits/sec. Other formats might be specified as appropriate like: 'm' = Mbits/sec, 'b' = bits/sec or 'K' = KBytes/sec, 'M' = MBytes/sec, 'B' = Bytes/sec</td>
</tr>
</tbody>
</table>

**Client:**

`iperf -c 192.168.1.193 -P 1 -i 1 -p 5001 -f k -t 10 -T 1`

<table>
<thead>
<tr>
<th>Command line option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c 192.168.1.193</td>
<td>Run Iperf in client mode, connecting to an Iperf server running on host.</td>
</tr>
<tr>
<td>-P 1</td>
<td>The number of simultaneous connections to make to the server. Default is 1. Requires thread support on both the client and server.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>i 1</code></td>
<td>Sets the interval time in seconds between periodic bandwidth, jitter, and loss reports. If non-zero, a report is made every interval seconds of the bandwidth since the last report. If zero, no periodic reports are printed. Default is zero.</td>
</tr>
<tr>
<td><code>-p 5001</code></td>
<td>The server port for the server to listen on and the client to connect to. This should be the same in both client and server. Default is 5001, the same as ttcp.</td>
</tr>
<tr>
<td><code>-f k</code></td>
<td>A letter specifying the format to print bandwidth numbers in. Here 'k' = Kbits/sec. Other formats might be specified as appropriate like: 'm' = Mbits/sec, 'b' = bits/sec or 'K' = KBytes/sec, 'M' = MBytes/sec, 'B' = Bytes/sec</td>
</tr>
<tr>
<td><code>-t 10</code></td>
<td>The time in seconds to transmit for. Iperf normally works by repeatedly sending an array of len bytes for time seconds. Default is 10 seconds. See also the <code>-l</code> and <code>-n</code> options.</td>
</tr>
<tr>
<td><code>-T 1</code></td>
<td>The time-to-live for outgoing multicast packets. This is essentially the number of router hops to go through, and is also used for scoping. Default is 1, link-local.</td>
</tr>
</tbody>
</table>

Additional parameters can be set from the left part of the screen.
Notice the change in parameter ‘–f k’ to ‘-f m’ when the option from left pane is selected for Output frame format to be in Mbits. This clearly shows that there is no need for user to remember any parameter and can just select from user interface and continue with execution of the test.

One additional benefit of JPerf over simple iPerf command line execution is that it provides us with a Bandwidth & Jitter Graph using the results generated by running iPerf which are shown at the bottom right part of the JPerf screen. The Bandwidth & Jitter Graph shows a real-time information of the sent/received bytes of any number of devices connected to the device at that very time.

iPerf Screen for Android

iPerf is also adapted to run on portable devices like android phones and tablets with an app known as iPerf for Android and looks similar to the following screenshot:
This app is simple command line interface which requires us to input iPerf commands in order to execute the test. As can be seen from the screenshot, unlike JPerf this doesn’t automatically generate commands and they need to be manually fed on the device by typing it in the command line text input provided at the center of the screen. However, since it utilizes the same set of commands and hence we can key in the same command which we utilized for JPerf.

**Procedure**

Once the commands have been generated or keyed in the device, the user can press ““Run Iperf” button located on the top right corner on the screen on the JPerf interface or the ON/OFF button located at the left top in the android Iperf App on the server. We then need to perform the same action on the client device. The generated statistics will be displayed on the lower part of the screen on the JPerf/Iperf utility on the devices as shown below:
Server:

![JPerf 2.0.2 - Network performance measurement graphical tool](image)

Bandwidth & Jitter

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
<th>5.0</th>
<th>6.0</th>
<th>7.0</th>
<th>8.0</th>
<th>9.0</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU usage</td>
<td>5.0%</td>
<td>6.0%</td>
<td>7.0%</td>
<td>8.0%</td>
<td>9.0%</td>
<td>10.0%</td>
<td>11.0%</td>
<td>12.0%</td>
<td>13.0%</td>
<td>14.0%</td>
</tr>
</tbody>
</table>

Output:

- [280] 8.0-10.0 sec 1.22 MBytes 10.2 Mbits/sec
- [248] 9.0-12.0 sec 0.73 MBytes 6.12 Mbits/sec
- [280] 9.0-10.0 sec 1.27 MBytes 10.7 Mbits/sec
- [TD] Interval Transfer Bandwidth
- [280] 0.0-10.5 sec 10.5 MBytes 8.56 Mbits/sec
- [248] 10.0-11.0 sec 0.77 MBytes 3.09 Mbits/sec
- [280] 0.0-11.1 sec 9.63 MBytes 7.28 Mbits/sec

---

Bhavik N. Patel
Client:

![Image of JPerf 2.0.2 network performance measurement graphical tool]

Iperf command: `iperf -c 192.168.1.193 -P -i 1 -p 5001 -f m -t 11 -T 1`

Choose iPerf Mode: Client

Server address: 192.168.1.193

Port: 5001

Parallel Streams: 1

Client Limit: [ ]

Num Connections: 0

Bandwidth

Sat, 5 Apr 2014 05:02:00

15.0

10.0

5.0

2.5

0

5

10

15

Mbps (b/s)

Time (sec)

Output

[ 4] 7.0-8.0 sec 0.88 MBytes 7.29 Mbits/sec
[ 4] 8.0-9.0 sec 0.75 MBytes 6.29 Mbits/sec
[ 4] 9.0-10.0 sec 0.38 MBytes 3.15 Mbits/sec
[ 4] 10.0-11.0 sec 0.38 MBytes 3.15 Mbits/sec
[ 4] 0.0-11.1 sec 9.62 MBytes 7.29 Mbits/sec

Done.

Bhavik N. Patel
JPerf also allows exporting the generated log file of detailed report to be exported in text format and graph to be exported as image. This can be done by right clicking and using the ‘save as’ option.

Reference:

This was a basic guide for using iPerf/JPerf. For further or detailed information you can refer to:

http://iperf.fr/