Continue

FileCatalyst is a file transfer acceleration tool. Like any software, it is still limited by the performance of the hardware and the network on which it is used. If you are trying to achieve speeds faster than 1 Gbps, with FileCatalyst, read on. If you already know youll need to transfer files faster than 1Gbps, before installing FileCatalyst use IPERF3 to ensure your system is clean and has no bottlenecks. A tool like IPERF3 can be used to identify the bottlenecks or problems that might occur during transfer. IPERF3 is a free open-source tool that is widely used for measurements of the maximum achievable throughput between point-to-point connections and it can be used with TCP and UDP protocols. Bottlenecks can occur when trying to transfer files to and from different locations. Poor throughput can happen either at the network level or the storage level (normally when writing and reading from disk). IPERF3 helps troubleshoot these problems, before you even install FileCatalyst. IPERF3 can also be used to validate that your firewall is properly configured to allow FileCatalyst to work at optimal speed. IPERF3 probing will save a lot of time during the installation and configuration of FileCatalyst, firewall configuration, and storage configuration. Therefore, we recommend having clean IPERF3 logs before attempting to install FileCatalyst. Prerequisites Install IPERF3 tools in the machines you want to test. Have access to both machines where you will run the tests. Also we recommend using the same machines where FileCatalyst Server and FileCatalyst Client will be installed. Make sure you allow UDP traffic (to guarantee we will able to use acceleration). We recommend running IPERF3 under this two scenarios: For new installations and when leading a proof of concept (POC). To troubleshoot performance problems. Related Reading:What is IPERF and Whats the Difference Between IPERF and IPERF3? IPERF3 Command Line Syntax The general syntax to run the test is described below: Server side commands:iperf3 -s[options] Client side commands:iperf3 -c [options] Where some of the options are listed: General IPERF3 commands: IPERF3 Command Line Option Description -f, format [kmKM] A letter specifying the format to print bandwidth numbers in. Supported formats are k = Kbits/sec K = KBytes/sec m = Mbits/sec M = MBytes/sec -J, json Output in JSON format Server-only IPERF3 commands: IPERF3 Command Line Option Description -F, file name Server-side: read from the network and write to the file, instead of throwing the data away. Client-only IPERF3 commands: IPERF3 Command Line Option Description -F, file name Client-side: read from the file and write to the network, instead of using random data; -u, udp Use UDP rather than TCP. -b, bandwidthn[KM] Set target bandwidth to n bits/sec (default 1 Mbit/sec for UDP, unlimited for TCP). If there are multiple streams (-P flag), the bandwidth limit is applied separately to each stream.A must when testing UDP -R, reverse Run in reverse mode (server sends, client receives). -t, timen The time in seconds to transmit for. Default is 10 seconds. For more information, please see the official site: Please notefor the test listed in the following section, the server used has IP 172.20.7.22 and the client has 172.20.5.200 IP. Network Tests: IPERF3 Examples UDP throughput: Server: iperf -s Client: iperf3 -c 172.20.7.22 -P 2 -t 10 -b 1G -u With these results, we can see: Packet loss If UDP traffic is allowed The network jitter (Jitter is the variation in the delay of received packets, as a result of network congestion, improper queuing, etc.) TCP throughput: Under certain circumstances, we might need to test TCP throughput, most likely to check the multi-stream TCP behavior. If this is necessary, you can use the examples below as a reference. Server: iperf3 -s Client: iperf3 -c 172.20.7.22 -P 2 -t 30. This test will run 2 TCP streams during 30s. Disk Tests: IPERF3 Examples Disk testing using IPERF3: Disk to Memory If you run the test multiple times, make sure of the following: Using a different file every time you run OR Use a command to clear the cache of the tiles: sync; echo 3 > /proc/sys/vm/drop_caches** (as root user) Use the right command for your OS For this test, 52GB and 20GB files were used, running from the client to the server machine. For the first test, run the test as we did before using the following commands: Server: iperf3 -s Client: iperf3 -c 172.20.7.22 -i2 -t30 -b 1G -u Using a 20GB file: Server: iperf3 -s Client: iperf3 c 172.20.722 -f /home/administrator/Desktop/20GB0..mov -i2 -u -b 1G -t30 Using a 52GB file: Server: iperf3 -s Client:iperf3 -c 172.20.7.22 -f /home/administrator/Desktop/52GB0.mov -i2 -u -b 1G -t30 Disk testing using IPERF3: Memory to Disk The test will be performed the same as Disk to memory test, however the main difference is instead of using a file on client side, this will be from Server side as is shown allow: Server: iperf3 -s -f /home/administrator/Desktop/52GB0.movClient:iperf3 -c 172.20.7.22 -i2 -u -b 1G -t30 Important Note:both Test 3 and Test 4 will run until the end of the file or the end of the test duration and the slowest of the results will show the bottleneck. iPerf3 is a tool for active measurements of the maximum achievable bandwidth on IP networks.It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, SCTP with IPv4 and IPv6).For each test it reports the bandwidth, loss, and other parameters.This is a new implementation that shares no code with the original iPerf andalso is not backwards compatible. iPerf was orginally developed by NLANR/DAST.iPerf3 is principally developed by ESnet/ Lawrence Berkeley National Laboratory.It is released under a three-clause BSD license. iPerf features TCP and SCTP Measure bandwidth Report MSS/MTU size and observed read sizes. Support for TCP window size via socket buffers. UDP Client can create UDP streams of specified bandwidth. Measure packet loss Measure delay jitter Multicast capable Cross-platform: Windows, Linux, Android, MacOS X, FreeBSD, OpenBSD, NetBSD, VxWorks, Solaris,... Client and server can have multiple simultaneous connections (-P option). Server handles multiple connections, rather than quitting after a single test. Can run for specified time (-t option), rather than a set amount of data to transfer (-n or -k option). Print periodic, intermediate bandwidth, jitter, and loss reports at specified intervals (-i option). Run the server as a daemon (-D option) Use representative streams to test out how link layer compression affects your achievable bandwidth (-F option). A server accepts a single client simultaneously (iPerf3) multiple clients simultaneously (iPerf2) New: Ignore TCP slowstart (-O option). New: Set target bandwidth for UDP and (new) TCP (-b option). New: Set IPv6 flow label (-L option) New: Set congestion control algorithm (-C option) New: Use SCTP rather than TCP (--sctp option) New: Output in JSON format (-J option). New: Disk read test (server: iperf3 -s / client: iperf3 -c testhost -i1 -F filename) New: Disk write tests (server: iperf3 -s -F filename / client: iperf3 -c testhost -i1) The default IP protocol is IPv4 Your public IPv4 address is : 91.105.45.108 (Reverse DNS: 91.105.45.108) Your public IPv6 address is : You do not have IPv6 connectivity (Reverse DNS: n/a) Mailing list To post a message to all the list members, you need to subscribe to Iperf-users.To see the collection of prior postings to the list, visit the Iperf-users Archives. Bug Report for iPerf3 Before submitting a bug report, try checking out the latest version of the code, and confirm that its not already fixed.Then submit to the iPerf3 issue tracker on GitHub: Known Issues The following problems are notable known issues, which are probably of interest to a large fraction of users or have high impact for some users, and for which issues have already been filed in the issue tracker. These issues are either open (indicating no solution currently exists) or closed with the notation that no further attempts to solve the problem are currently being made:UDP performance: Some problems have been noticed with iperf3 on the ESnet 100G testbed at high UDP rates (above 10Gbps). The symptom is that on any particular run of iperf3 the receiver reports a loss rate of about 20%, regardless of the ``-b`` option used on the client side. This problem appears not to be iperf3-specific, and may be due to the placement of the iperf3 process on a CPU and its relation to the inbound NIC. In some cases this problem can be mitigated by an appropriate use of the CPU affinity (``-A``) option. (Issue #55)Interval reports on high-loss networks: The way iperf3 is currently implemented, the sender write command will block until the entire block has been written. This means that it might take several seconds to send a full block if the network has high loss, and the interval reports will have widely varying interval times. A solution is being discussed, but in the meantime a work around is to try using a small block size, for example ``-l 4K``. (Issue #125, a fix will be released in iperf 3.1)The ``-Z`` flag sometimes causes the iperf3 client to hang on OSX. (Issue #129)When specifying the TCP buffer size using the ``-w`` flag on Linux, the Linux kernel automatically doubles the value passed in to compensate for overheads. (This can be observed by using iperf3's ``--debug`` flag.) However, CWND does not actually ramp up to the doubled value, but only to about 75% of the doubled value. Some part of this behavior is documented in the tcp(7) manual page. (Issue #145) Bug Report for iPerf2 The best way to get help with iPerf2 is by using its forum To update this site, please report them to vivien16@gueant.org and we will try to fix them quickly. The main authors of iPerf3 are (in alphabetical order): Jon Dugan, Seth Elliott, Bruce A. Mah, Jeff Poskanzer, Kaustubh Prabhu.Additional code contributions have come from (also in alphabetical order): Mark Ashley, Aaron Brown, Aeneas Jaile, Susant Sahani, Bruce Simpson, Brian Tierney. iPerf3 contains some original code from iPerf2. The authors of iPerf2 are (in alphabetical order):Jon Dugan, John Estabrook, Jim Ferbuson, Andrew Gallatin, Mark Gates, Kevin Gibbs,Stephen Hemminger, Nathan Jones, Feng Qin, Gerrit Renker, Ajay Tirumala, Alex Warshavsky. Acknowledgements for iPerf1: Thanks to Mark Gates (NLANR), Alex Warshavsky (NLANR) and Justin Pietsch (University of Washington)who were responsible for the 1.1.x releases of Iperf.For iPerf 1.7, we would like to thank Bill Cerveny (Internet2), Micheal Lambert (PSC), Dale Finkelson (UNL)and Matthew Zekauskas (Internet2) for help in getting access to IPv6 networks / machines.Special thanks to Matthew Zekauskas (Internet2) for helping out in the FreeBSD implementation.Also, thanks to Kraemer Oliver (Sony) for providing an independent implementation of IPv6version of Iperf, which provided a useful comparison for testing our features. Thanks to ESnet for re-rolling iperf from the ground up. iPerf3 is a killer piece of software. Thanks to for hosting iPerf.fr. Iperf3 is a powerful open source tool for analyzing network performance and diagnosing issues. In this comprehensive guide, well cover the key concepts and commands for using iperf3 to measure throughput, loss, and other metrics.Introduction to iperf3Iperf3 is the latest version of iperf, a commonly used network testing tool that originated in 2003. It allows engineers to measure maximum network throughput/bandwidth, jitter, and packet loss between two hosts.Some key capabilities of iperf3:Test TCP or UDP performanceClient/server architecture for easy deploymentMulti-threaded for testing multiple streamsCustomizable for advanced/edge casesWide range of statistics reported Iperf3 is included in most Linux distros and Mac OS, but can also be compiled from source on Unix, Windows, and other platforms. Its one of the essential tools for testing networks and prototyping new configurations.Installing iperf3 on LinuxTo check if iperf3 is already installed, run:$ iperf3If not installed, use your package manager to install:$ sudo apt install iperf3 or$ sudo yum install iperf3Once installed, you should see the usage instructions printed when running iperf3.TCP Network Throughput TestsFor TCP tests, iperf3 uses a client/server model. First start the iperf3 server, then connect a client to it.Start the iperf3 serverRun the server on the host you want to connect to:$ iperf3 -sThis starts iperf3 in server mode, listening on the default port 5201.Connect the iperf3 clientOn the client host, connect to the server IP and port:$ iperf3 -c Common options:-t time to transmit for (default 10 seconds) -i interval time to report stats (default 1 second)For example, to test for 60 seconds with 5 second intervals:$ iperf3 -c 192.168.1.100 -i 5 -t 60This will connect to server 192.168.1.100 and report bandwidth statistics every 5 seconds for 60 seconds total.Understanding TCP outputThe server will show output like:[5] local 192.168.1.100 port 5201 connected to 192.168.1.4 port 51220 [ID] Interval Transfer Bandwidth[5] 0.00-5.00 sec 22.2 MBytes 37.2 Mbits/sec [5] 5.00-10.00 sec 21.3 MBytes 35.7 Mbits/sec[5] 10.00-15.00 sec 22.1 MBytes 37.1 Mbits/sec [5] 15.00-20.01 sec 21.6 MBytes 36.1 Mbits/sec...This shows the client IP and port, interval times, amount of data transferred, and bandwidth measured.The client will show:Connecting to host 192.168.1.100, port 5201[4] local 192.168.1.4 port 51220 connected to 192.168.1.100 port 5201[ID] Interval Transfer Bandwidth [4] 0.00-5.00 sec 21.5 MBytes 36.0 Mbits/sec[4] 5.00-10.00 sec 21.2 MBytes 35.5 Mbits/sec [4] 10.00-15.00 sec 21.9 MBytes 36.7 Mbits/sec[4] 15.00-20.01 sec 21.3 MBytes 35.6 Mbits/sec...This includes the connection details, intervals, transfer, and bandwidth like the server. The bandwidth is the key measurement here for TCP throughput.TCP loss and retriesBy default TCP tests show bandwidth only since loss should be minimal. To view packet loss, retries, and out-of-order info add -i 1 to print stats every 1 second:$ iperf3 -c 192.168.1.100 -i 1For high loss networks, TCP stats can help diagnose issues.UDP Network ThroughputFor UDP, the same client/server model applies but the output is different.Start the server:$ iperf3 -s -uThe -u flag specifies UDP.Connect a UDP client:$ iperf3 -c 192.168.1.100 -u -b 200M -t 60Here we added -b 200M to target 200 Mbps bandwidth and -t 60 for 60 seconds.Understanding UDP outputThe server will show:[3] local 192.168.1.100 port 5201 connected to 192.168.1.4 port 49153[ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams[3] 0.00-1.00 sec 24.4 MBytes 205 Mbits/sec 0.038 ms 0/3130 (0%) [3] 1.00-2.00 sec 24.6 MBytes 206 Mbits/sec 0.036 ms 1/3148 (0.03%)[3] 2.00-3.00 sec 24.9 MBytes 209 Mbits/sec 0.035 ms 0/3203 (0%)...Now jitter and packet loss is shown since UDP does not retry packets. Lost packets directly reduce throughput.The client will show similar stats:[3] local 192.168.1.4 port 49153 connected to 192.168.1.100 port 5201[ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams...The key metrics here are bandwidth along with jitter and loss. High jitter and packet loss will significantly degrade UDP throughput.Advanced iperf3 OptionsIperf3 supports many options for customizing tests and have more control over connections.Multiple parallel streamsUse -P to set multiple parallel client streams:$ iperf3 -c 192.168.1.100 -P 10This tests bandwidth with 10 TCP streams running over a single connection.Reverse mode client/serverBy default iperf3 connects client -> server. Use -R to reverse and push data server -> client:$ iperf3 -c 192.168.1.100 -RHelpful for testing networks with asymmetric routing or ACLs.Binding to specific interfacesBind the server or client to certain interfaces with -B :$ iperf3 -s -B 192.168.1.100$ iperf3 -c 192.168.1.100 -B 192.168.1.4This overrides the default outbound IP.Adjust TCP window sizeFor long distance links, modify TCP window size with -w:$ iperf3 -c 192.168.1.100 -w 1MSets the TCP window to 1 MB instead of the default 128K.Capture packet dumpsDump the full packet capture to a PCAP file using -B:$ iperf3 -c 192.168.1.100 -B dump.pcapAnalyze the capture in Wireshark for deep inspection.Testing IPv6Pass the -6 flag to enable IPv6:$ iperf3 -s -6$ iperf3 -c 2600::1 -6All tests will now run over IPv6 instead of IPv4.Network Troubleshooting with iperf3Here are some tips for applying iperf3 tests to identify and resolve network issues:Test at different times of day compare results to find time periods with reduced throughput. May indicate network congestion.Test from different client locations narrow down the location of congestion points.Test TCP and UDP compare to find outliers indicating protocol specific issues.Correlate results with ping and traceroute combine with other active probing tools to validate where problems occur.Verify against baseline expectations compare measured throughput to modeled provisioned bandwidth.Repeat tests over longer durations check for intermittent problems hidden by shorter tests.By strategically running iperf3 tests and correlating the data, you can pinpoint many kinds of network problems.Optimizing Networks Based on iperf3 DataUse iperf3 results to identify and resolve network bottlenecks:Upgrade overloaded network gear switches, routers, firewalls that cant handle the full traffic volume.Configure QoS policies implement QoS and traffic shaping to improve performance for critical applications.Tune TCP stack options adjust TCP buffers, windows, congestion control algorithms to match the network.Scale up capacity add links, aggregate ports, implement faster media types to increase bandwidth.Eliminate unnecessary traffic block or rate limit unneeded protocols eating up capacity.As a simple, portable throughput tester, iperf3 allows iteratively engineering networks for improved performance.Additional Iperf3 ExamplesHere are some example iperf3 commands for real-world scenarios:Test wireless network capacity:$ iperf3 -c 192.168.1.100 -i 2 -t 60Check link quality over time.Test maximum backbone throughput:$ iperf3 -c 10.1.1.100 -P 50 -t 300Use high parallel streams over 5 minutes.Verify WAN capacity before deployment:$ iperf3 -c 203.0.113.5 -r -t 1800Reverse test to remote office for 30 minutes.Validate SAN replication speed:$ iperf3 -c 10.2.32.5 -i 2 -t 120Use short intervals to find fluctuations.ConclusionIperf3 is a flexible, cross-platform tool for network performance analysis and troubleshooting. It can measure TCP and UDP throughput, jitter, and packet loss between any two networked devices.With both basic and advanced options, iperf3 provides extensive control over test conditions to validate performance in different scenarios. The output offers administrators and engineers detailed metrics to optimize networks and identify problems.By incorporating iperf3 testing into engineering workflows, infrastructure can be proactively tuned for speed, reliability, and scalability. iperf3 is a widely used tool designed to test network bandwidth and performance between two endpoints. It is commonly employed by network administrators to measure throughput, identify bottlenecks, and validate configurations across various network environments. The tool provides a reliable means to assess and optimize network parameters, ensuring robust data flows and efficient resource utilization.Use case 1: Running iperf3 as a ServerCode:Motivation:Running iperf3 as a server is essential for establishing a test environment that allows clients to connect and assess network bandwidth between the two devices. This setup forms the foundational aspect of conducting thorough network assessments where one machine acts as the server waiting to receive data while another machine (the client) sends data.Explanation:iperf3: This is the command invoking the iperf3 tool.-s: This argument specifies that iperf3 should operate in server mode. In this mode, the tool listens for incoming connections from iperf3 clients looking to establish a performance test.Example output:---Server listening on 5201---The output indicates that the server is now listening on the default port (5201), ready to accept client connections for bandwidth testing.Use case 2: Running an iperf3 Server on a Specific PortCode:Motivation:Specifying a non-default port number for the iperf3 server can be critical when default ports are occupied or when organizational policies dictate the use of specific ports for network services. It provides flexibility and adaptability in various network environments.Explanation:iperf3: The command initiating the iperf3 tool.-s: This argument enables the server mode, enabling the tool to listen for incoming connections.-p 12345: The -p argument allows the user to define a specific port (in this case, 12345) for the server to listen on rather than the default port 5201.Example output:---Server listening on 12345---This output indicates that the server is actively listening on the specified port 12345 for any incoming client requests.Use case 3: Starting a Bandwidth TestCode:Motivation:Conducting a bandwidth test is the primary goal for utilizing iperf3. By executing this command, users can evaluate the maximum available bandwidth between a client and server, revealing potential network issues or validating the current setups performance capability.Explanation:iperf3: The primary command running the tool.-c server: The -c argument specifies client mode and requires the IP address or hostname of the server to which the client will connect. The term server here should be replaced with the actual servers address.Example output:Connecting to host server, port 5201[4] local IP -> remote IP port 5201[ID] Interval Transfer Bandwidth[4] 0.00-10.00 sec 1.25 GBytes 1.07 Gbits/secThe output details the data transferred and the bandwidth observed during the test interval, conveying the networks capacity between client and server.Use case 4: Running iperf3 in Multiple Parallel StreamsCode:Motivation:Testing network performance with multiple parallel streams can simulate realistic traffic scenarios more accurately. It is beneficial in understanding how a network handles concurrency and potential bottlenecks under increased load conditions, allowing deeper insights into performance metrics.Explanation:iperf3: Calls the iperf3 command-line utility.-c server: Denotes client mode with a connection to the specified server.-P 5: This argument specifies the number of parallel data streams (in this case, 5) the client should use in the performance test. More streams can mimic higher traffic loads.Example output:Connecting to host server, port 5201[4] local IP -> remote IP port 5201[ID] Interval Transfer Bandwidth[SUM] 0.00-10.00 sec 6.25 GBytes 5.35 Gbits/secIn this excerpt, bandwidth is measured across all streams, offering a comprehensive perspective on available network throughput when multiple connections are engaged.Use case 5: Reversing Direction of the TestCode:Motivation:Reversing the direction of the test is crucial in scenarios where the primary interest is the bandwidth from server to client rather than the client to server. This helps to understand the reverse flow of data, which is vital for applications primarily dependent on server-to-client data transmission.Explanation:iperf3: Initial command to access iperf3 features.-c server: Enable client mode and aim the test at the desired server.-R: The -R switch reverses the test traffic, letting the server send the data to the client, which is useful to check download speed from the servers perspective.Example output:Connecting to host server, port 5201Reverse mode, remote host server is sending[4] local IP -> remote IP port 5201[ID] Interval Transfer Bandwidth[4] 0.00-10.00 sec 1.10 GBytes 950 Mbits/secHere, the output captures the data transferred and bandwidth measured from server to client, providing critical insight into download performance.Conclusion:Using iperf3 effectively assists in diagnosing and optimizing network performance across various scenarios. By understanding different configurations and executing tests tailored to specific requirements, network administrators can ensure more efficient, reliable, and scalable network environments. This set of examples demonstrates the operational flexibility of iperf3 and its utility in addressing real-world networking challenges. In this article we shall see how to use iperf to very easily test the maximum throughput in network bandwidth between a server and a client. Iperf is a free and helpful tool, available on multiple platforms, to troubleshoot network performance.It is not uncommon getting error reports similar to the network is slow, which is the assumed cause of various perceived problems. While it definitely sometimes is bandwidth congestion, my experience is that it most often is not the network being slow, but some other harder-to-spot factor being the bottleneck (disk, cpu, application logic or combinations). Being able to quickly test the physical bandwidth in such situation is often useful.Iperf is an open source tool to test network bandwidth. A pre-compiled Windows version is available from Softpedia. The tool is a 110 kB single exe file with no installation. Place a copy at server and one copy on the other end, which could be an ordinary PC. One part will be representing a server which sends data to a client. The iperf tool could be used in virtual servers to test out the virtual network both internally between virtual machines and between VMs and the rest of the network.Some confusion in the terminology exist, as the part sending the data (typically a server) is called the iperf client, and the computer receiving the data (typically a client computer) is called the server, which is reversed as how many client-server applications work.On the PC side, download the iperf file and from the command line start with the following options: (Note that all options are case sensitive.)-s (start the server which will receive the data)-w 32768 (change the TCP window size to 32 KB, default is a bit low 8 KB)iperf.exe -s -w 32768The tool will listen on TCP port 5001 at default. Depending on your Windows firewall settings you might have to accept incoming connection to this port. If wanted you can change the listening port with -p (lowercase).On the server (which could be physical or virtual) we will now start the iperf tool with different settings. Make sure the same executable file is available locally and use these settings:-c IP-ADDRESS (the IP address to the other computer)-w 32768 (to raise the TCP window size)-t SECONDS (the number of seconds to send data)-P 8 (the number of alternate streams to increase throughput, must be uppercase P)iperf.exe -c IP -P 8 -t 30 -w 32768When the test has been running for t seconds, perhaps 30-60 seconds at least, we will see the network result.Observe the final line, the [SUM] and the rightmost column for the average throughput in Mbit per second. If wanted the -f option could be used to express the values in bits, kilobits or gigabits instead. When doing longer tests, like running the tool for several minutes you could use -i option to display results every X second.You could experiment with the -w for TCP Window Size and the -P for parallel datastreams to find the optimal throughput in your situation.If you want to test the full duplex network performance, that is, transmitting and receiving at the same time, just add the -d option at the client.iperf.exe -c IP -P 8 -t 30 -w 32768 -dThis will monitor the bandwidth capacity in both directions simultaneous and also put a higher stress on the network interface card, CPUs and the network devices between the client and the server. In the test above we can see that the servers and the network are delivering great full duplex throughput with over 870 Mbit/s both incoming and outgoing. IPerf is an open-source command line tool designed to test network throughput between two network hosts. The iPerf allows to generate TCP and UDP traffic (load) between two hosts. You can use iPerf to quickly measure the maximum network bandwidth (throughput) between a server and a client, and conduct stress testing of the ISP link, router, network gateway (firewall), your Ethernet, or Wi-Fi network performance.In this article, we will show you how to install and use the iPerf tool to test the network speed (throughput) on Windows. Linux and VMware ESXi (there are iPerf versions for Android, macOS, RouterOS from MikroTik, and other platforms).IPerf is a cross-platform tool and doesnt require installation in Windows. Just download the executable and run it on two devices, the network throughput between which needs to be tested. IPerf works in a client-server mode. On the first device, the iPerf starts in server mode (it is waiting for traffic from the iPerf client). On the second computer, the iPerf starts in client mode, generates TCP or UDP traffic, and measures the maximum data transfer rate.In most cases, the iPerf3 version is used now (it supports high-speed UDP testing, port 5201 TCP/UDP is used by default).How to Install and Use iPerf on Windows?You can download iPerf 3.1.3 for Windows by the following link . If you want to use iPerf 2.0, you can get it here. You dont need to install the iPerf on Windows, it is enough to download the iPerf archive and extract it to a local directory on the drive. There are only two files in the archive: cygwin1.dll and iperf3.exe.Iperf is a console tool and you need to run it in the command line. Open a command prompt (cmd.exe) and go to the directory with the utility. For example:cd c:\tools\iperfIf you run the iperf3.exe program without parameters, it will list the available options.The iPerf utility can run in server mode (-s option) or client mode (-c). If you are running an iPerf server on Windows, you need to open inbound port 5201 for TCP and UDP protocols. You can open ports using the Windows Defender Firewall GUI or with PowerShell commands. Create and enable firewall rules like this:New-NetFirewallRule -DisplayName 'iPerf-Server-Inbound-TCP' -Direction Inbound -Protocol TCP -LocalPort 5201 -Action Allow | Enable-NetFirewallRule New-NetFirewallRule -DisplayName 'iPerf-Server-Inbound-UDP' -Direction Inbound -Protocol UDP -LocalPort 5201 -Action Allow | Enable-NetFirewallRuleThere are several iPerf GUI implementations for Windows. For example, Iperf3-Cygwin-GUI or jperf.The jperf is written on Java (in order to run jperf, the Java VM must be installed on the computer). In addition to graphical form for the CLI interface, Jperf can build real-time graph bandwidth charts of the communication channel load. To use it, just specify the iPerf server address and start the network throughput test.Installing iPerf on LinuxOn CentOS/RHEL/Fedora/Rocky Linux/Oracle Linux 8, the iperf3 package is available in the base AppStream repository (in CentOS 7 you can find iperf3 in the EPEL repo).You can install it with the dnf (yum) command:# dnf install iperf3On Debian/Ubuntu/Mint distros you can install the iperf3 with the command:$ sudo apt install iperf3If you want to use your Linux host as the iperf3 server, you need to open the TCP/UDP ports 5201 in firewalld (or iptables):# firewall-cmd --permanent --add-port=5201/udp # firewall-cmd --permanent --add-port=5201/tcp # firewall-cmd --reloadUsing iPerf on VMware ESXiVMware removed the iPerf tool in ESXi 6.7 but brought it back in 6.7U1. Make sure you have iPerf installed on your ESXi host. Connect to the ESXi console via SSH, go to the /usr/lib/vmware/vsan/bin directory and check if it contains the iperf or iperf3 binary file.If iPerf is not installed, you can manually download the iperf offline bundle here (. Copy it to the ESXi host and install with the command:# esxcli software vib install -d /iperf-2.0.5-1-offline_bundle.zip no-sig-checkThe tool will be installed to the /opt/iperf/bin directory. Use the following command to run iPerf on VMware ESXi:# /opt/iperf/bin/iperf3If you are running an iPerf server on an ESXi host, you need to open port 5201. This port is disabled by default in the ESXi firewall. To open them, you can temporarily disable the ESXi firewall:# esxcli network firewall get # esxcli network firewall set --enabled=false Dont forget to enable the ESXi firewall after completing the network throughput test:# esxcli network firewall set --enabled trueWhen running iPerf on ESXi, it must be bound to the vmkernel interface:# esxcli network ip interface ipv4 get# iperf -s -B 192.168.31.50Lets now look at some examples of using iPerf to test network throughput. In this example, we will use a Linux host as the iperf server. Lets run the iperf in server mode:The iPerf server is running, it is waiting for connections on the 5201 TCP port.You can run the iPerf server as a Windows Service using the -D argument.I am using a computer with Windows 10 as an iPerf client. Open the command prompt and change to the directory with the iPerf executable: cd c:\tools\iperf In order to run a network bandwidth test from the client, specify the iPerf server address (or DNS name): iperf3.exe -c 192.168.1.200iperf3 is a free open-source, cross-platform command-line-based program for performing real-time network throughput measurements. It is one of the most powerful tools for testing the maximum achievable bandwidth in IP networks (supports IPv4 and IPv6).[You might also like: 17 Useful Bandwidth Monitoring Tools for Linux]With iperf, you can tune several parameters associated with timing, buffers, and protocols such as TCP, UDP, and SCTP. It comes in handy for network performance tuning operations.In order to acquire maximum or rather improved network performance, you need to increase the throughput as well as the latency of your networks receiving and sending capabilities.However, before you can go into actual tuning, you need to perform some tests to gather overall network performance statistics that will guide your tuning process.Its results include time interval in seconds, data transferred, bandwidth (transfer rate), loss, and other useful network performance parameters. It is primarily intended to assist in tuning TCP connections over a particular path and this is what we will focus on in this guide.Requirements:Two networked computers which both have iperf3 installed.How to Install iPerf3 in Linux SystemsBefore you start using iperf3, you need to install it on the two machines you will use for benchmarking. Since iperf3 is available in the official software repositories of most common Linux distributions, installing it should be easy, using a package manager as shown.$ sudo apt install iperf3 [On Debian, Ubuntu and Mint] $ sudo yum install iperf3 [On RHEL/CentOS/Fedora and Rocky/AlmaLinux] $ sudo emerge -a net-misc/iperf [On Gentoo Linux] $ sudo apk add iperf3 [On Alpine Linux] $ sudo pacman -S iperf3 [On Arch Linux] $ sudo zypper install iperf3 [On OpenSUSE] Once you have iperf3 installed on both machines, you can start testing network throughput.How to Test Network Speed Between Two Linux ServersFirst, connect to the remote machine which you will use as the server, and fire up iperf3 in server mode using -s flag, it will listen to port 5201 by default.You can specify the format (k, m, g for Kbits, Mbits, Gbits or K, M, G for KBytes, Mbytes, Gbytes) to report in, using the -f switch as shown.$ iperf3 -s -f K If port 5201 is being used by another program on your server, you can specify a different port (e.g 3000) using the -p switch as shown.$ iperf3 -s -p 3000Optionally, you can run the server as a daemon, using the -D flag and write server messages to a log file, as follows.$ iperf3 -s -D > iperf3log Then on your local machine which we will treat as the client (where the actual benchmarking takes place), run iperf3 in client mode using -c flag and specify the host on which the server is running (either using its IP address or domain or hostname).$ iperf3 -c 192.168.10.1 -f KAfter about 18 to 20 seconds, the client should terminate and produce results indicating the average throughput for the benchmark, as shown in the following screenshot.Test Network Throughput Between ServersFrom the benchmark results, as shown in the above screenshot, there is a variation in values from the server and client. But, you should always consider using the results obtained from the iperf client machine in every test you carry out.How to Perform Advanced Network Test Throughput in LinuxThere are a number of client-specific options for performing an advanced test, as explained below.One of the important factors that determine the amount of data in the network at a given time is the TCP window size it is important in tuning TCP connections. You can set the window size/socket buffer size using the -w flag as shown.$ iperf3 -c 192.168.10.1 -f K -w 500KTo run it in reverse mode where the server sends and the client receives, add the -R switch.$ iperf3 -c 192.168.10.1 -f K -w 500K -RTo run a bi-directional test, meaning you measure bandwidth in both directions simultaneously, use the -d option.$ iperf3 -c 192.168.10.1 -f K -w 500K -dIf you want to get server results in the client output, use the --get-server-output option.$ iperf3 -c 192.168.10.1 -f K -w 500K -R --get-server-outputGet Server Network Results in the ClientIt is also possible to set the number of parallel client streams (two in this example), which run at the same time, using the -P options.$ iperf3 -c 192.168.10.1 -f K -w 500K -P 2For more information, see the iperf3 man page.$ man iperf3Thats all! Remember to always perform network performance tests before going for actual network performance tuning. iperf3 is a powerful tool, that comes in handy for running network throughput tests.Do you have any thoughts to share or questions to ask, use the comment form below.When selecting a dedicated server provider for your business, one of the key components of the service price is the network bandwidth capacity available with the server plan. It is therefore important to be able to perform a reliable network bandwidth test to determine if youre getting what you paid for.To perform such a test with regards to your home bandwidth connection is easily done with many popular third-party tools. They are usually hosted on a domain with a user-friendly web interface, however, using the same speed test service for determining the network throughput of your server is a bad idea. It will always show you unreliable results.Limitations of traditional web speedtest servicesFirstly, thespeed test providersrun tests between your server and the web servers on their side. Therefore you should check what uplink ports their end servers have. The majority of todays speedtest servers run on a 1Gbps uplink,so its physically impossible to measure higher throughput than the uplink capacity.Secondly, if youredownloading a test fileplaced on your server from your home device, you are always limited by the network speed of your home connection (the speed you subscribed for with your local ISP). Given the average internet connection speed in the US is around 19 Mbps, youll never be able to test a server with 10Gbps unmetered bandwidth efficiently.Other forms of connection testing such as the wget command also present limitations (single concurrent connections) which render the method ineffective for testing network throughput inhigh-bandwidth network environments.How to effectively measure 10Gbps bandwidth?With regards, to the misconceptions regarding the speedtest methods mentioned above, we have decided to write a tutorial on how to perform a network test connection between two servers. This can be a connection betweenmultiple data centersor within the same one. To run a reliable test on 10Gbps uplink servers, you need to run multiple simultaneous test connections.To test network bandwidth, we always recommend a popular network tool called Iperf3. The Iperf3 tutorial will cover installation commands for Linux OS and CentOS.Iperf tutorialWhat is Iperf and how to install it?Iperfis a point-to-point network tool for active measurements of throughput between two servers in both directions. The tool can effectively measure the maximum server bandwidth of your uplink. Iperf measures both TCP and UDP bandwidth performance.You need at least two servers to run an Iperf3 test a source and a destination server.HOW TO INSTALL IPERF3 ON YOUR SERVER:Linux OS:Use the command:apt-get install iperf3Cent OS:Use the command:yum install iperf3How to use Iperf to test 10Gbps network bandwidthThe big advantage and main reason why we use Iperf is, that it is capable of running multiple test connections between two servers simultaneously. This is crucial for testing connections of 10Gbps or higher (opposed to a wget command to download a test file that only holds one connection unable to utilize all of the server resources usually 1Gbps maximum).Using Iperf you can test the bandwidth speed from the origin server (iperf client-server) to a destination server (iperf listening server). Both need to have sufficient bandwidth to perform the tests you are after (If youre performing a 10Gbps speed test, both servers need a 10GE uplink, the listening server should Ideally have a higher bandwidth capacity than the client server 20Gbps.).Using Iperf commands on the client-server (in this case your server), you will run a benchmark test and show the throughput data of all test connections.If you dont have an IP address of a suitable Iperf listening server, please contact our support and we will provide you with one.If you havemultiple serverswith your dedicated server provider, you can run an Iperf test between them.Iperf commandsIf youre using your own servers for both the client-server and listening server side. Activate the listening server mode using the commandiperf3 -sServer listening on 5201Youll see that the server mode is activated and what port it is listening on.Here is a command for running an Iperf test on your client server. The only other data you need is the IP address of the Iperf listening server. We recommend running more parallel streams (TCP/UDP), as one stream will be hashed to one physical uplink interface of our access switch. Use parameter -P X, where X represents the count of parallel flows.Command:iperf3 -P 20 -c $server_ipThis test will run 20 simultaneous connections against the Iperf listening server ($server_ip).Test results (Client side):These results only show the first and last four lines of the test result as showing the whole result would take up too much space and wouldnt provide any additional clarification. The columns from the above output display:Interval:It specifies the time duration for the data transfer.Transfer:This shows the total data size transferred using the Iperf test. All of the data is flushed out after completing the test.Bandwidth:The rate of speed with which the data was transferred in Mbit/sec.Each line gives the sender and receiver results for every stream. The most important results to look at are the last two [SUM] lines. They show the average result of the network bandwidth test. In this case, each stream used up approximately 460 Mbps. Total network bandwidth reaching 9.33 Gbits/sec. The client-server has a 10Gbps uplink, the listening server a bonded 20Gbps uplink.This tutorial focused solely on the use of Iperf for testing the network throughput between two servers. The tool, of course, allows many other configurations and application via a range of other Iperf commands.If you are a customer of DataPacket and would like any additional help with the setup, do not hesitate tocontact DataPacket support. IPerf is an open-source command line tool designed to test network throughput between two network hosts. The iPerf allows to generate TCP and UDP traffic (load) between two hosts. You can use iPerf to quickly measure the maximum network bandwidth (throughput) between a server and a client, and conduct stress testing of the ISP link, router, network gateway (firewall), your Ethernet, or Wi-Fi network performance.In this article, we will show you how to install and use the iPerf tool to test the network speed (throughput) on Windows. Linux and VMware ESXi (there are iPerf versions for Android, macOS, RouterOS from MikroTik, and other platforms).IPerf is a cross-platform tool and doesnt require installation in Windows. Just download the executable and run it on two devices, the network throughput between which needs to be tested. IPerf works in a client-server mode. On the first device, the iPerf starts in server mode (it is waiting for traffic from the iPerf client). On the second computer, the iPerf starts in client mode, generates TCP or UDP traffic, and measures the maximum data transfer rate.In most cases, the iPerf3 version is used now (it supports high-speed UDP testing, port 5201 TCP/UDP is used by default).How to Install and Use iPerf on Windows?You can download iPerf 3.1.3 for Windows by the following link . If you want to use iPerf 2.0, you can get it here. You dont need to install the iPerf on Windows, it is enough to download the iPerf archive and extract it to a local directory on the drive. There are only two files in the archive: cygwin1.dll and iperf3.exe.Iperf is a console tool and you need to run it in the command line. Open a command prompt (cmd.exe) and go to the directory with the utility. For example:cd c:\tools\iperfIf you run the iperf3.exe program without parameters, it will list the available options.The iPerf utility can run in server mode (-s option) or client mode (-c). If you are running an iPerf server on Windows, you need to open inbound port 5201 for TCP and UDP protocols. You can open ports using the Windows Defender Firewall GUI or with PowerShell commands. Create and enable firewall rules like this:New-NetFirewallRule -DisplayName 'iPerf-Server-Inbound-TCP' -Direction Inbound -Protocol TCP -LocalPort 5201 -Action Allow | Enable-NetFirewallRule New-NetFirewallRule -DisplayName 'iPerf-Server-Inbound-UDP' -Direction Inbound -Protocol UDP -LocalPort 5201 -Action Allow | Enable-NetFirewallRuleThere are several iPerf GUI implementations for Windows. For example, Iperf3-Cygwin-GUI or jperf.The jperf is written on Java (in order to run jperf, the Java VM must be installed on the computer). In addition to graphical form for the CLI interface, Jperf can build real-time graph bandwidth charts of the communication channel load. To use it, just specify the iPerf server address and start the network throughput test.Installing iPerf on LinuxOn CentOS/RHEL/Fedora/Rocky Linux/Oracle Linux 8, the iperf3 package is available in the base AppStream repository (in CentOS 7 you can find iperf3 in the EPEL repo).You can install it with the dnf (yum) command:# dnf install iperf3On Debian/Ubuntu/Mint distros you can install the iperf3 with the command:$ sudo apt install iperf3If you want to use your Linux host as the iperf3 server, you need to open the TCP/UDP ports 5201 in firewalld (or iptables):# firewall-cmd --permanent --add-port=5201/udp # firewall-cmd --permanent --add-port=5201/tcp # firewall-cmd --reloadUsing iPerf on VMware ESXiVMware removed the iPerf tool in ESXi 6.7 but brought it back in 6.7U1. Make sure you have iPerf installed on your ESXi host. Connect to the ESXi console via SSH, go to the /usr/lib/vmware/vsan/bin directory and check if it contains the iperf or iperf3 binary file.If iPerf is not installed, you can manually download the iperf offline bundle here (. Copy it to the ESXi host and install with the command:# esxcli software vib install -d /iperf-2.0.5-1-offline_bundle.zip no-sig-checkThe tool will be installed to the /opt/iperf/bin directory. Use the following command to run iPerf on VMware ESXi:# /opt/iperf/bin/iperf3If you are running an iPerf server on an ESXi host, you need to open port 5201. This port is disabled by default in the ESXi firewall. To open them, you can temporarily disable the ESXi firewall:# esxcli network firewall get # esxcli network firewall set --enabled=false Dont forget to enable the ESXi firewall after completing the network throughput test:# esxcli network firewall set --enabled trueWhen running iPerf on ESXi, it must be bound to the vmkernel interface:# esxcli network ip interface ipv4 get# iperf -s -B 192.168.31.50Lets now look at some examples of using iPerf to test network throughput. In this example, we will use a Linux host as the iperf server. Lets run the iperf in server mode:The iPerf server is running, it is waiting for connections on the 5201 TCP port.You can run the iPerf server as a Windows Service using the -D argument.I am using a computer with Windows 10 as an iPerf client. Open the command prompt and change to the directory with the iPerf executable: cd c:\tools\iperf In order to run a network bandwidth test from the client, specify the iPerf server address (or DNS name): iperf3.exe -c 192.168.1.200

Iperf test. Average throughput in performance testing. What is throughput in performance testing. Iperf options. Iperf latency test. Iperf test speed. Iperf max throughput test.