NAME

ovs−benchmark − flow setup benchmark utility for Open vSwitch

SYNOPSIS

ovs−benchmark latency −−remote *ip*[**:***ports*] [**−−sockets** *nsocks*] [**−−batches** *nbatches*] [**−−local** [*ip*][**:***ports*]]

ovs−benchmark rate −−remote *ip*[**:***ports*] [**−−max−rate** *rate*] [**−−timeout** *maxsecs*] [**−−sockets** *nsocks*] [**−−batches** *nbatches*] [**−−local** [*ip*][**:***ports*]]

ovs−benchmark listen [**−−local** [*ip*]**:***ports*]

ovs−benchmark help

DESCRIPTION

ovs−benchmark tests the performance of Open vSwitch flow setup by setting up a number of TCP connections and measuring the time required. It can also be used with the Linux bridge or without any bridging software, which allows one to measure the bandwidth and latency cost of bridging.

Each **ovs−benchmark** command is described separately below.

The "latency" command

This command initiates *nsocks* TCP connections (by default, 100) as quickly as possible, waits for each one to complete with success or failure, and prints a bar chart of completion times on standard output, followed by a summary line. Each line in the bar chart lists a time to connection completion in milliseconds followed by a number of **.** or **!** symbols, one for each TCP connection that completed in that many milliseconds. A successful connection prints a **.**, and an unsuccessful connection (e.g. to a port on which no process is listening) prints a **!**.

If *nbatches* is given, the entire procedure is repeated the specified number of times. Only a single summary line is printed at the end.

Results vary widely based on the number of sockets and whether the remote host is listening for connections on the specified ports. With a small number of sockets, all connection times typically remain within a handful of milliseconds. As the number of sockets increases, the distribution of connection times clusters around the sending TCP stack's SYN retransmission interval. (This pattern occurs with or without Open vSwitch on the network path.)

The "rate" command

This command initiates *nsocks* TCP connections (by default, 100) as quickly as possible (limited by *maxrate*, if **−−max−rate** is specified). Each time a connection completes with success or failure, it closes that connection and initiates a new one. It continues to do so either forever or, if **−−timeout** is specified, until *maxsecs* seconds have elapsed. During the test, it prints statistics about time elapsed, successful and unsuccessful connections, and the average number of completed (succeeded or failed) connections per second over the run.

Without **−−max−rate**, the **rate** command measures the maximum sustained flow setup rate for an Open vSwitch instance. This naturally tends to drive **ovs−vswitchd** CPU usage to 100% on the host receiving the traffic.

When **−−max−rate** is specified with a value below the maximum rate that an Open vSwitch instance can handle, then **rate** can also be used to measure the kernel and userspace CPU cost of flow setups at specific flow rates.

Results tend to fluctuate greatly for the first few seconds of a run, then settle down. The displayed average is calculated over the entire run and so tends to converge asymptotically on the ''correct'' value. To converge more quickly, try running for 5 to 10 seconds, then killing and restarting the run.

The "listen" command

This command listens on one or more TCP ports for incoming connections. It accepts connections and immediately closes them. It can be paired with the **rate** or **latency** commands for observing effects of successful vs. unsuccessful TCP connections.

It is easier to reproduce and interpret **ovs−benchmark** results when there is no listener (see **NOTES** below).

The "help" command

Prints a usage message and exits successfully.

OPTIONS

−r *ip*[**:***ports*]
−−remote *ip*[**:***ports*]

This option, required on **latency** and **rate** commands, minimally specifies the remote host to connect to (as an IP address or DNS name) as *ip*.

A TCP port or range of ports (separated by −) may also be specified. If a range is specified then each port in the range is used in round-robin order. The default port is 6630 if none is specified.

−l [*ip*][**:***ports*]
−−local [*ip*][**:***ports*]

On the **latency** and **rate**, without this option, outgoing connections will not bind a specific TCP port. The local TCP stack will pick a local TCP port to bind. When this option is specified, the specified port or range of ports will be used in turn. (If a port range is specified on both **−−local** and **−−remote**, then each local port in its range will be used before the remote port is incremented to the next port in its range.)

On the **listen** command, this option specifies the local port or ports and IP addresses on which to listen. If it is omitted, port 6630 on any IP address is used.

−s *nsocks*
−−sockets *nsocks*

For **latency**, sets the number of connections to initiate per batch. For **rate**, sets the number of outstanding connections attempts to maintain at any given time. The default is 100.

−b *nbatches*
−−batches *nbatches*

For **latency**, sets the number of times to initiate and wait for all of the connections to complete. The default is 1.

−c *maxrate*
−−max−rate *maxrate*

For **rate**, caps the maximum rate at which connections will be attempted to *maxrate* connections per second. By default there is no limit.

−T *maxsecs*
−−timeout *maxsecs*

For **rate**, stops the benchmark after *maxsecs* seconds have elapsed. By default, the benchmark continues until interrupted by a signal.

NOTES

ovs−benchmark uses standard POSIX socket calls for network access, so it shares the strengths and limitations of TCP/IP and its implementations in the local and remote TCP/IP stacks. Particularly, TCP and its implementations limit the number of successfully completed and then closed TCP connections. This means that **ovs−benchmark** tests tend to slow down if run for long intervals or with large numbers of sockets or batches, if the remote system is listening on the port or ports being contacted. The problem does not occur when the remote system is not listening. **ovs−benchmark** results are therefore much more reliable and repeatable when the remote system is not listening on the port or ports being contacted. Even a single listening socket (e.g. range of ports 8000 to 9000 with one listener on port 8080) can cause anomalies in results.

Be sure that the remote TCP/IP stack's firewall allows the benchmark's traffic to be processed. For Open vSwitch benchmarking purposes, you might want to disable the firewall with, e.g., **iptables −F**.

ovs−benchmark is single-threaded. A multithreaded process might be able to initiate connections more quickly.

A TCP connection consists of two flows (one in each direction), so multiply the TCP connection statistics that **ovs−benchmark** reports by 2 to get flow statistics.