Low Latency Quickstart Guide

Introduction

This quick start guide demonstrates how to achieve very low latency coupled with minimum jitter on a system fitted with Solarflare network adapters and using Solarflare's kernel-bypass network acceleration middleware, Onload.

The guide will focus on the performance of the network adapter for TCP and UDP applications running on Linux. The guide uses the industry-standard Netperf network benchmark application and the Solarflare supplied open source sfnettest network benchmark application.

Please read the Solarflare LICENSE file regarding the disclosure of benchmark test results.

Procedure

1. Software Installation on page 1.
2. Test Setup on page 2.
3. Pre-Test Configuration on page 2.
4. TCP Latency on page 3.
5. UDP Latency on page 4.

Software Installation

Netperf

Netperf can be downloaded from http://www.netperf.org/netperf/

Unpack the compressed tar file using the tar command:

```
[root@system-N]# tar -zxvf netperf-<version>.tar.gz
```

This will create a sub-directory called `netperf-<version>` from which the configure and make commands can be run (as root):

```
./configure
make install
```

Following installation the `netperf` and `netserver` applications are located in the `src` subdirectory.

Solarflare sfnettest

Download the `sfnettest-<version>.tgz` source file from www.openonload.org

Unpack the tar file using the tar command:

```
[root@system-N]# tar -zxvf sfnettest-<version>.tgz
```

Run the `make` utility from the `/sfnettest-<version>/src` subdirectory to build the `sfnt-pingpong` application.

Solarflare Onload

Before Onload network and kernel drivers can be built and installed the system must support a build environment capable of compiling kernel modules. Refer to Appendix C - Build Dependencies in the Onload User Guide for more details.

Download the `Onload-<version>.tgz` file from www.openonload.org

Unpack the tar file using the tar command:

```
[root@system-N]# tar -zxvf onload-<version>.tgz
```

Run the `onload_install` command from the `Onload-<version>/scripts` subdirectory:

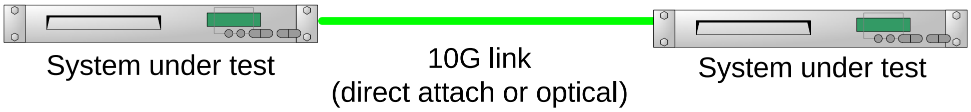
```
[root@system-N]# ./onload_install
```

This command will issue a warning if Solarflare drivers are already installed on the system. Reload the Solarflare drivers :

```
[root@system-N]# ./onload_tool reload
```

Test Setup

The diagram below identifies the required physical configuration of two servers equipped with Solarflare network adapters connected back-to-back in order to measure the latency of the adapter, drivers and acceleration middleware. If required, tests can be repeated with a 10G switch on the link to measure the additional latency delta using a particular switch.



Requirements:

- Two servers are equipped with Solarflare network adapters.
- The two machines are connected with a single cable between the Solarflare interfaces.
- The Solarflare interfaces are configured with an IP address so that traffic can pass between them. Use `ping` to verify connection.
- Onload is installed on both machines.
- The `netperf` client and `netserver` applications are installed on both machines.
- Solarflare `sfnettest` is installed on both machines.

Pre-Test Configuration

On both machines:

1 Stop the cpuspeed service to prevent power saving modes from reducing CPU clock speed.

```
[root@system-N]# service cpuspeed stop
```

2. Stop the irqbalance service to prevent the OS from rebalancing interrupts between available CPU cores.

   ```
   [root@system-N]# service irqbalance stop
   ```

3. Stop the iptables service to eliminate overheads incurred by the firewall. Solarflare recommend this step on RHEL6 for improved latency when using the kernel network driver.

   ```
   [root@system-N]# service iptables stop
   ```

4. Disable interrupt moderation.

   ```
   [root@system-N]# ethtool -C eth<N> rx-usecs 0 adaptive-rx off
   ```

   Where N is the identifier of the Solarflare adapter ethernet interface e.g. `eth4`

5. Prevent the OS from switching the CPUs into a lower power mode (C1E).

   ```
   [root@system-N]# onload_tool disable_cstates
   ```

Reference System Specification

The following latency measurements were recorded on twin Intel® Romley Servers with default factory configuration and no changes to OS or BIOS. The specification of the test systems is as follows:

- 2 x Intel® Romley Server. Equipped with Intel® Xeon® CPU E5-2687W 0 @ 3.10GHz, 32GB MEM.
- Red Hat Enterprise Linux v6.2 (x86_64 kernel, version 2.6.32-220.el6.x86_64).
- Solarflare SFN6122F NIC (driver and firmware v3.2.0). Direct-attach cable at 10G.
- Onload 201205

Solarflare BASE-T adapters will typically return a higher 1/2 RTT latency (2-2.5usecs) than SFP+ adapters. It is expected that similar results will be achieved on any Intel® or AMD based, PCIe Gen 2 server or compatible system.

TCP Latency

TCP Latency: Netperf

Run the netserver application on system-1:

```
[root@system-1]# pkill -f netserver
[root@system-1]# onload --profile=latency taskset -c 8 ./netserver
```

Run the netperf application on system-2:

```
[root@system-2]# onload --profile=latency taskset  -c  8 ./netperf  -t
TCP_RR -H <system1-ip> -l 10 -- -r 32

Socket Size   Request  Resp.   Elapsed  Trans.
Send   Recv   Size     Size    Time     Rate
bytes  Bytes  bytes    bytes   secs.    per sec
16384  87380  32       32      10.00    203309.03
```

203309 transactions/second means that each transaction takes 1/203309 seconds resulting in a RTT/2 latency of (1/203309)/2 or **2.4us.**

TCP Latency: sfnt-pingpong

Run the sfnt-pingpong application on both systems:

```
[root@system-1]# onload --profile=latency taskset -c 8 ./sfnt-pingpong

[root@system-2]# onload --profile=latency taskset -c 8 ./sfnt-pingpong --
maxms=10000 --affinity "8;8"  tcp  <system1-ip>
#
#       size    mean    min     median  max     %ile    stddev  iter
        1       2453    2380    2434    18288   2669    77      1000000
        2       2453    2379    2435    45109   2616    90      1000000
        4       2467    2380    2436    10502   2730    82      1000000
        8       2465    2383    2446    8798    2642    70      1000000
        16      2460    2380    2441    7494    2632    68      1000000
        32      2474    2399    2454    8758    2677    71      1000000
        64      2495    2419    2474    12174   2716    77      1000000
```

The output identifies mean, minimum, median and maximum (nanosecond) RTT/2 latency for increasing TCP packet sizes including the 99% percentile and standard deviation for these results. A message size of 32 bytes has a mean latency of **2.4us** with a 99%ile latency under **2.7us.**

UDP Latency

UDP Latency: Netperf

Run the netserver application on system-1:

```
[root@system-1]# pkill -f netserver
[root@system-1]# onload --profile=latency taskset -c 8 ./netserver
```

Run the netperf application on system -2:

```
[root@system-2]# onload --profile=latency taskset -c 8 ./netperf -t UDP_RR -H
<system1-ip> -l 10 -- -r 32

Socket Size   Request  Resp.   Elapsed  Trans.
Send   Recv   Size     Size    Time     Rate
bytes  Bytes  bytes    bytes   secs.    per sec
124928 124928 32       32      10.00    221175.78
```

221175 transactions/second means that each transaction takes 1/221175 seconds resulting in a RTT/2 latency of (1/221175)/2 or **2.2us**.

UDP Latency: sfnt-pingpong

Run the `sfnt-pingpong` application on both systems:

```
[root@system-1]# onload --profile=latency taskset -c 8./sfnt-pingpong
```

```
[root@system-2]# onload --profile=latency taskset -c 8./sfnt-pingpong --maxms=10000
--affinity "8;8"  udp  <system1-ip>
#        size     mean     min      median   max      %ile     stddev   iter
         0        2273     2218     2267     20082    2358     68       1000000
         1        2277     2218     2270     43863    2400     77       1000000
         2        2274     2217     2269     9887     2353     56       1000000
         4        2282     2225     2278     7474     2354     63       1000000
         8        2280     2221     2275     7906     2401     49       1000000
         16       2279     2221     2276     9565     2340     52       1000000
         32       2281     2221     2278     9800     2339     47       1000000
         64       2309     2239     2303     8051     2428     54       1000000
```

The output identifies mean, minimum, median and maximum (nanosecond) RTT/2 latency for increasing UDP packet sizes including the 99% percentile and standard deviation for these results. A message size of 32 bytes has a mean latency of **2.2us** with a 99%ile latency under **2.4us**.

Testing Without Onload

The benchmark performance tests can be run without Onload using the regular kernel network drivers. To do this remove the `onload --profile=latency` part from the command line.

To get the best response and comparable latency results using kernel drivers, Solarflare recommend setting interrupt affinity so that the network interrupts and the benchmark application run on the same CPU core. Refer to the Solarflare Server Adapter User Guide ’Interrupt and Irqbalance Service’ for configuration options.

Using regular kernel network drivers the following comparable results were recorded (usecs).

	RHEL 6.2	
	Mean	*99%ile*
TCP	6.8	7.7
UDP	5.4	5.8

Further Information

For installation of Solarflare adapters and performance tuning of the network driver when not using Onload refer to the Solarflare Server Adapter User Guide (SF-103837-CD) available from https://support.solarflare.com/

The Onload feature set and detailed performance tuning information can be found in the Onload User Guide (SF-104474-CD) available from https://support.solarflare.com/

Questions regarding Solarflare products, Onload and this user guide can be emailed to support@solarflare.com