

RedHawk 9.2 OpenOnload Installation Instructions Release Notes

November 30, 2023



This page intentionally left blank

1. Introduction

This document provides instructions for installing OpenOnload version 8.1.2.26 for Onload compatible cards software onto an existing system installed with RedHawk Linux 9.2.x.

2. Requirements

- An x86_64 host system running RedHawk 9.2.x
- A *RedHawk OpenOnload* version 9.2 DVD
- A supported Onload adapter card listed on page 43 of the *Onload User Guide*, available on the DVD.

3. Installation

3.1. Boot to RedHawk 9.2

Ensure that the system is booted with a RedHawk 9.2 kernel and invoke the following command to verify:

```
# uname -r
```

3.2. Install required base distribution packages

Mount the *RedHawk OpenOnload* DVD and issue the following commands to install the various base distribution packages that are required by OpenOnload:

```
# mkdir -p /tmp/mount
# mount /dev/cdrom /tmp/mount
# cd /tmp/mount
# ./install
```

NOTE

This installation script can run for as long as 30 minutes as it compiles and installs kernel modules, and several user-space packages and then updates the firmware in the Onload adapter card installed in the system. The many warnings produced during installation can be safely ignored.

When the installation is complete, reboot the system with the same RedHawk kernel. After the reboot, proceed with the steps below.

3.3. Verify installation

Invoke the following command to verify that OpenOnload has been successfully installed:

```
# onload --version
```

Output like the following should be displayed:

```
Onload 8.1.2.26
Copyright (c) 2002-2023 Advanced Micro Devices, Inc.
```

Built: Nov 30 2023 09:36:04 (release)
Build profile header: <ci/internal/transport_config_opt_extra.h>
Kernel module: 8.1.2.26

NOTE

Onload will only be installed on the current running kernel. If the running kernel is switched after installation, you must re-run the install script with the --kernelfiles option to use Onload on the new kernel. See --help for a list of valid options for the install script.

3.4. Simple Onload configuration test

This section provides steps for verifying that connections between two directly connected compatible Onload adapter cards can be established successfully.

1. On both systems, modify the `/etc/hosts` file to define two addresses. In this simple example, one system is named *server* and the other system is named *client*. Add entries like the following: note that the text added must be identical on both systems:

```
10.1.1.1 server-oo
10.1.1.2 client-oo
```

2. On both systems, create a `/etc/sysconfig/network-scripts/ifcfg-<interface>` file, where `<interface>` is the name of the interface of the Onload adapter card.

```
NAME=<interface>
DEVICE=<interface>
BOOTPROTO=none
TYPE=Ethernet
IPADDR=10.1.1.1           ← Use 10.1.1.1 for server and 10.1.1.2 for client
NETMASK=255.255.255.0
NETWORK=10.1.1.0
BROADCAST=10.1.1.255
ONBOOT=no
```

3. Reboot both systems, ensuring that the same RedHawk kernels are booted.
4. Issue the following command on both systems to verify that the Onload network is configured correctly and that both hosts are active and connected:

```
# ifconfig <interface>
```

5. Finally, verify that you can ping and ssh from both systems to the other system:
 - ping `client-oo` from `server-oo`
 - ping `server-oo` from `client-oo`
 - ssh into `client-oo` from `server-oo`
 - ssh into `server-oo` from `client-oo`

This concludes the simple Onload configuration and connection verification.

Please contact Concurrent Real-Time technical support if you had any problems during this installation (<http://www.concurrent-rt.com/support> or 1-800-245-6453).

3.5. IP over OpenOnload Latency Testing

This section provides steps for verifying the connection between two directly connected Onload cards can achieve advertised latency.

Sfnettest is a set of benchmark tools and test utilities created by Solarflare for benchmark and performance testing of network servers and network adapters. On both systems, issue the following commands to build and install Sfnettest from source:

```
# git clone https://github.com/Xilinx-CNS/cns-sfnettest
# cd cns-sfnettest/src
# make
# cp sfnt-pingpong /usr/bin
```

First, set some configuration options that decrease latency for Onload acceleration technologies. On both machines:

```
# systemctl stop firewalld
# sysctl -w vm.nr_hugepages=1024
# ethtool -C <interface> rx-usecs 0 adaptive-rx off
```

It is also recommended to shield the CPU core and its NUMA node the test will run on. For example, on both systems CPU 4 and its NUMA node were shielded:

```
# shield -a C4
```

1. On the server run the following to start sfnt-pingpong on CPU 4:

```
# onload run -b4 sfnt-pingpong
```

2. On the client run the following to start the test on CPU 4:

```
# onload run -b4 sfnt-pingpong --affinity "4;4" tcp 10.0.0.1
```

Output like the following should appear on the client once the test establishes a connection to the server:

size	mean	min	median	max	%ile	stddev	iter
1	10213	9273	10185	29074	10902	251	147000
2	10139	9266	10121	30681	10745	229	148000
4	10137	9289	10119	32408	10737	234	148000
8	10140	9203	10123	31292	10737	226	148000
16	10137	9193	10118	30494	10735	227	148000
32	10156	9243	10137	29063	10752	225	148000

The output identifies mean, minimum, median and maximum (nanosecond) ½ RTT latency for increasing packet sizes, including the 99% percentile and standard deviation for these results.