



Solarflare® Server Adapter User Guide

Information in this document is subject to change without notice.

Copyright © 2008-2019 SOLARFLARE® Communications, Inc. All rights reserved.

Trademarks used in this text are registered trademarks of Solarflare® Communications Inc; *Adobe* is a trademark of Adobe Systems. *Microsoft*® and *Windows*® are registered trademarks of Microsoft Corporation.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Solarflare Communications Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

The software and hardware as applicable (the “Product”) described in this document, and this document, are protected by copyright laws, patents and other intellectual property laws and international treaties. The Product described in this document is provided pursuant to a license agreement, evaluation agreement and/or non-disclosure agreement. The Product may be used only in accordance with the terms of such agreement. The software as applicable may be copied only in accordance with the terms of such agreement.

The furnishing of this document to you does not give you any rights or licenses, express or implied, by estoppel or otherwise, with respect to any such Product, or any copyrights, patents or other intellectual property rights covering such Product, and this document does not contain or represent any commitment of any kind on the part of SOLARFLARE Communications, Inc. or its affiliates.

The only warranties granted by SOLARFLARE Communications, Inc. or its affiliates in connection with the Product described in this document are those expressly set forth in the license agreement, evaluation agreement and/or non-disclosure agreement pursuant to which the Product is provided. EXCEPT AS EXPRESSLY SET FORTH IN SUCH AGREEMENT, NEITHER SOLARFLARE COMMUNICATIONS, INC. NOR ITS AFFILIATES MAKE ANY REPRESENTATIONS OR WARRANTIES OF ANY KIND (EXPRESS OR IMPLIED) REGARDING THE PRODUCT OR THIS DOCUMENTATION AND HEREBY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT, AND ANY WARRANTIES THAT MAY ARISE FROM COURSE OF DEALING, COURSE OF PERFORMANCE OR USAGE OF TRADE. Unless otherwise expressly set forth in such agreement, to the extent allowed by applicable law (a) in no event shall SOLARFLARE Communications, Inc. or its affiliates have any liability under any legal theory for any loss of revenues or profits, loss of use or data, or business interruptions, or for any indirect, special, incidental or consequential damages, even if advised of the possibility of such damages; and (b) the total liability of SOLARFLARE Communications, Inc. or its affiliates arising from or relating to such agreement or the use of this document shall not exceed the amount received by SOLARFLARE Communications, Inc. or its affiliates for that copy of the Product or this document which is the subject of such liability.

The Product is not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

A list of patents associated with this product is at <http://www.solarflare.com/patent>

SF-103837-CD

Last revised: February 2019

Issue 24



Trademarks

OpenOnload®, EnterpriseOnload®, XtremeScale™ and Flareon™ are registered trademarks of Solarflare Communications Inc in the United States and other countries.

Table of Contents

1 Introduction	1
1.1 Virtual NIC Interface	1
1.2 Product Specifications.....	5
1.3 Software Driver Support.....	14
1.4 Solarflare AppFlex™ Technology.....	15
1.5 Open Source Licenses	15
1.6 Support and Driver Download	16
1.7 Regulatory Information and Approvals	16
2 Installation	17
2.1 Solarflare Network Adapter Products	18
2.2 Fitting a Full Height Bracket (optional)	19
2.3 Inserting the Adapter in a PCI Express (PCIe) Slot	19
2.4 Attaching a Cable (RJ-45)	20
2.5 Attaching a Cable (SFP+, SFP28).....	21
2.6 Cables and Transceivers	24
2.7 QSFP+ Transceivers and Cables	31
2.8 Supported Speed and Mode.....	34
2.9 25G Link Speed	35
2.10 LED States.....	37
2.11 Port Modes	38
2.12 Single Optical Fiber - RX Configuration	44
2.13 Solarflare Mezzanine Adapter: SFN8722 OCP.....	44
2.14 Solarflare Precision Time Synchronization Adapters	45
2.15 Solarflare ApplicationOnload™ Engine	45

3 Solarflare Adapters on Linux	46
3.1 System Requirements	46
3.2 Linux Platform Driver Feature Set	47
3.3 Installing the Adapter Driver	49
3.4 SUSE Linux Enterprise Server Distributions.....	52
3.5 Installing DKMS Driver and Utilities on Ubuntu/Debian Servers.....	53
3.6 Remove 'in-tree' Driver.....	54
3.7 Configure the Solarflare Adapter.....	55
3.8 Setting Up VLANs.....	57
3.9 Setting Up Teams.....	58
3.10 NIC Partitioning	59
3.11 NIC Partitioning with SR-IOV	63
3.12 Receive Side Scaling (RSS).....	66
3.13 Receive Flow Steering (RFS)	68
3.14 Solarflare Accelerated RFS (SARFS)	70
3.15 Transmit Packet Steering (XPS).....	71
3.16 Linux Utilities RPM	73
3.17 Configuring the Boot Manager with sfboot	74
3.18 Upgrading Adapter Firmware with sfupdate.....	82
3.19 Activation key install with sfkey	87
3.20 Performance Tuning on Linux.....	90
3.21 Web Server - Driver Optimization	97
3.22 Interrupt Affinity	99
3.23 Module Parameters.....	109
3.24 Linux ethtool Statistics	111
3.25 Driver Logging Levels	120
3.26 Running Adapter Diagnostics	120
3.27 Running Cable Diagnostics	121



4 Solarflare Adapters on Windows.....	123
4.1 Windows 2016 Driver	124
4.2 Legacy Driver	124
4.3 System Requirements	124
4.4 Driver Certification	124
4.5 Minimum Driver and Firmware Packages	124
4.6 Firmware Variants	125
4.7 Windows Feature Set	126
4.8 Installing Solarflare Driver Package	127
4.9 Configuration & Management	129
4.10 Adapter Configuration	130
4.11 Flow Control.....	131
4.12 Jumbo Frames	132
4.13 Checksum Offload	132
4.14 Interrupt Moderation (Interrupt Coalescing)	133
4.15 NUMA Node.....	134
4.16 Receive Side Scaling (RSS).....	135
4.17 Teaming and VLANs.....	139
4.18 Adapter Statistics	142
4.19 Performance Tuning on Windows	143
5 Solarflare Adapters on VMware.....	150
5.1 Native ESXi Driver (VMkernel API)	151
5.2 Legacy Driver (vmklinux API)	151
5.3 System Requirements	151
5.4 Distribution Packages	152
5.5 VMware Feature Set	153
5.6 Install Solarflare Drivers	155
5.7 Driver Configuration	157
5.8 Adapter Configuration	157
5.9 Granting access to the NIC from the Virtual Machine	158
5.10 NIC Teaming.....	158
5.11 Configuring VLANs.....	159
5.12 Performance Tuning on VMware	160
5.13 Interface Statistics.....	170
5.14 vSwitch/VM Network Statistics	170
5.15 CIM Provider	173
5.16 Adapter Firmware Upgrade - sfupdate_esxi	174
5.17 Adapter Configuration - sfboot_esxi	177
5.18 ESXCLI Extension	179
5.19 vSphere Client Plugin.....	188
5.20 Fault Reporting - Diagnostics	198
5.21 Network Core Dump	199
5.22 Adapter Diagnostic Selftest	199



6 Solarflare Adapters on FreeBSD	200
6.1 System Requirements	200
6.2 FreeBSD Platform Feature Set	201
6.3 Installing Solarflare Drivers	201
6.4 Unattended Installation	203
6.5 Configuring the Solarflare Adapter	205
6.6 Setting Up VLANs	206
6.7 FreeBSD Utilities Package	207
6.8 Configuring the Boot ROM with sfboot	208
6.9 Upgrading Adapter Firmware with sfupdate	214
6.10 Performance Tuning on FreeBSD	216
6.11 Module Parameters	226
6.12 Kernel and Network Adapter Statistics	228
7 SR-IOV Virtualization Using KVM	237
7.1 Introduction	237
7.2 SR-IOV	242
7.3 KVM Network Architectures	244
7.4 PF-IOV	257
7.5 General Configuration	259
7.6 Feature Summary	260
7.7 Limitations	261
8 SR-IOV Virtualization Using ESXi	262
8.1 Introduction	262
8.2 Configuration Procedure - SR-IOV	265
8.3 Configuration Procedure - DirectPath I/O	265
8.4 Install Solarflare Drivers in the Guest	265
8.5 Install the Solarflare Driver on the ESXi host	265
8.6 Install Solarflare Utilities on the ESXi host	266
8.7 Configure VFs on the Host/Adapter	268
8.8 Virtual Machine	269
8.9 List Adapters - Web Client	270
8.10 vSwitch and Port Group Configuration	271
8.11 VF Passthrough	275
8.12 DirectPath I/O	281



9 Solarflare Boot Manager	285
9.1 Introduction	285
9.2 Solarflare Boot Manager.....	286
9.3 iPXE Support	287
9.4 sfupdate Options for PXE upgrade/downgrade	287
9.5 Starting PXE Boot.....	289
9.6 iPXE Image Create	293
9.7 Multiple PF - PXE Boot	295
9.8 Default Adapter Settings.....	298
10 Unattended Installations	300
10.1 Unattended Installation - Red Hat Enterprise Linux	301
10.2 Unattended Installation - SUSE Linux Enterprise Server	303
Index	305

1

Introduction

This is the User Guide for Solarflare® Server Adapters. This chapter covers the following topics:

- [Virtual NIC Interface on page 1](#)
- [Advanced Features and Benefits on page 2](#)
- [Product Specifications on page 5](#)
- [Software Driver Support on page 14](#)
- [Solarflare AppFlex™ Technology. on page 15](#)
- [Open Source Licenses on page 15](#)
- [Support and Driver Download on page 16](#)



NOTE: Throughout this guide the term Onload refers to both OpenOnload® and EnterpriseOnload® unless otherwise stated. Users of Onload should refer to the *Onload User Guide*, SF-104474-CD, which describes procedures for download and installation of the Onload distribution, accelerating and tuning the application using Onload to achieve minimum latency and maximum throughput.

1.1 Virtual NIC Interface

Solarflare's VNIC architecture provides the key to efficient server I/O and is flexible enough to be applied to multiple server deployment scenarios. These deployment scenarios include:

- **Kernel Driver** – This deployment uses an instance of a VNIC per CPU core for standard operating system drivers. This allows network processing to continue over multiple CPU cores in parallel. The virtual interface provides a performance-optimized path for the kernel TCP/IP stack and contention-free access from the driver, resulting in extremely low latency and reduced CPU utilization.
- **Accelerated Virtual I/O** – The second deployment scenario greatly improves I/O for virtualized platforms. The VNIC architecture can provide a VNIC per Virtual Machine, giving over a thousand protected interfaces to the host system, granting any virtualized (guest) operating system direct access to the network hardware. Solarflare's hybrid SR-IOV technology, unique to Solarflare Ethernet controllers, is the only way to provide bare-metal I/O performance to virtualized guest operating systems whilst retaining the ability to live migrate virtual machines.

- **OpenOnload™** – The third deployment scenario aims to leverage the host CPU(s) to full capacity, minimizing software overheads by using a VNIC per application to provide a kernel bypass solution. Solarflare has created both an open-source and Enterprise class high-performance application accelerator that delivers lower and more predictable latency and higher message rates for TCP and UDP-based applications, all with no need to modify applications or change the network infrastructure. To learn more about the open source OpenOnload project or EnterpriseOnload, download the Onload user guide (SF-104474-CD) or contact your reseller.

Advanced Features and Benefits

Virtual NIC support	The core of Solarflare technology. Protected VNIC interfaces can be instantiated for each running guest operating system or application, giving it a direct pipeline to the Ethernet network. This architecture provides the most efficient way to maximize network and CPU efficiency. The Solarflare Ethernet controller supports up to 1024 vNIC interfaces per port. On IBM System p servers equipped with Solarflare adapters, each adapter is assigned to a single Logical Partition (LPAR) where all VNICS are available to the LPAR.
PCI Express	Implements PCI Express 3.1.
High Performance	Support for 40G Ethernet interfaces and a new internal datapath micro architecture.
Hardware Switch Fabric	Full hardware switch fabric in silicon capable of steering any flow based on Layer 2, Layer 3 or application level protocols between physical and virtual interfaces. Supporting an open software defined network control plane with full PCI-IOV virtualization acceleration for high performance guest operating systems and virtual applications.
Improved flow processing	The addition of dedicated parsing, filtering, traffic shaping and flow steering engines which are capable of operating flexibly and with an optimal combination of a full hardware data plane with software based control plane.
TX PIO	Transmit Programmed input/output is the direct transfer of data to the adapter without CPU involvement. As an alternative to the usual bus master DMA method, TX PIO improves latency and is especially useful for smaller packets.

CTPIO	Cut Through PIO - TX packets are streamed directly from the PCIe interface to the adapter port bypassing the main TX datapath to deliver lowest TX latency.
Multicast Replication	Received multicast packets are replicated in hardware and delivered to multiple receive queues.
Sideband management	NCSI RMII interface for base board management integration. SMBus interface for legacy base board management integration.
PCI Single-Root-IOV, SR-IOV, capable	16 Physical functions and up to 240 Virtual functions per adapter. Flexible deployment of 1024 channels between Virtual and Physical Functions. Support Alternate Routing ID (ARI). SR-IOV is not supported for Solarflare adapters on IBM System p servers.
10 Gigabit Ethernet	Supports the ability to design a cost effective, high performance 10 Gigabit Ethernet solution.
25 Gigabit Ethernet	Supported on X2 series adapters. See 25G Link Speed on page 35
Receive Side Scaling (RSS)	IPv4 and IPv6 RSS raises the utilization levels of multi-core servers dramatically by distributing I/O load across all CPUs and cores.
Stateless offloads	Through the addition of hardware based TCP segmentation and reassembly offloads, VLAN, VxLAN, NVGRE and GENEVE offloads.
Jumbo frame support	Support for up to 9216 byte jumbo frames.
MSI-X support	2048 MSI-X interrupt support enables higher levels of performance. Can also work with MSI or legacy line based interrupts.
Ultra low latency	Cut through architecture. < 7µs end to end latency with standard kernel drivers, < 1µs with Onload drivers.

Remote boot	Support for PXE boot 2.1 and UEFI Boot provides flexibility in cluster design and diskless servers (see Solarflare Boot Manager on page 285). Network boot is not supported for Solarflare adapters on IBM System p servers.
MAC address filtering	Enables the hardware to steer packets based on the MAC address to a VNIC.
Hardware timestamps	The Solarflare Flareon™ and XtremeScale™ series adapters can support hardware timestamping for all packets, sent and received - including PTP. The adapters incorporate a highly accurate stratum 3 compliant oscillator with drift of 0.37 PPM per day (c. 32ms/day).
FEC	Supported on X2 series 25GbE adapters. 25G link Forward Error Correction employs redundancy in channel coding as a technique used to reduce bit errors (BER) in noisy or unreliable communications channels. See Forward Error Correction on page 35
AN/LT	Supported on X2 series 25GbE adapters. Auto-negotiation/Link Training See Auto-negotiation/Link Training on page 35

1.2 Product Specifications

Solarflare XtremeScale™ X2 Series SFP28 Series Network Adapters

Solarflare XtremeScale™ X2522 Dual-Port 25GbE SFP28 PCIe 3.1 Server Adapter

Part numbers	X2522-25G or X2522-25G-Plus
Controller silicon	SFC9250
Power	13.2W typical
PCI Express	8/16 lanes Gen 3.1 (8.0GT/s)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes
PTP and hardware timestamps	Yes
1PPS	Yes
SR-IOV	Yes
CTPIO	Yes
FEC - Forward Error Correction	Yes
Network ports	2 x SFP28 (1G/10G/25G)

Solarflare XtremeScale™ X2522 Dual-Port 10GbE SFP28 PCIe 3.1 Server Adapter

Part numbers	X2522 or X2522-Plus
Controller silicon	SFC9250
Power	13.2W typical
PCI Express	8/16 lanes Gen 3.1 (8.0GT/s)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes
PTP and hardware timestamps	Yes
1PPS	Yes
SR-IOV	Yes
CTPIO	Yes
Network ports	2 x SFP28 (1G/10G)

Solarflare XtremeScale™ X2 Series QSFP28 Network Adapters

Solarflare XtremeScale™ X2542 Dual-Port 100GbE QSFP28 PCIe 3.1 Server Adapter

Part numbers	X2542 or X2542-Plus
Controller silicon	SFC9250
Power	13.2W typical
PCI Express	8/16 lanes Gen 3.1 (8.0GT/s)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes
PTP and hardware timestamps	Yes
1PPS	Yes
SR-IOV	Yes
CTPIO	Yes
FEC - Forward Error Correction	Yes
Network ports	2 x QSFP28 (1G/10G/25G/40G/50G/100G)

Solarflare XtremeScale™ X2541 Single-Port 100GbE QSFP28 PCIe 3.1 Server Adapter

Part numbers	X2541 or X2541-Plus
Controller silicon	SFC9250
Power	13.2W typical
PCI Express	8/16 lanes Gen 3.1 (8.0GT/s)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes
PTP and hardware timestamps	Yes
1PPS	Yes
SR-IOV	Yes
CTPIO	Yes
FEC - Forward Error Correction	Yes
Network ports	1 x QSFP28 (1G/10G/25G/50G/100G)



Solarflare XtremeScale™ 8000 Series Network Adapters

Solarflare XtremeScale™ SFN8722 Dual-Port 10GbE SFP+ PCIe 3.1 OCP Server Adapter

Part numbers	SFN8722
Controller silicon	SFC9240
Power	10.5W typical
PCI Express	8 lanes Gen 3.1 (8.0GT/s)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes
PTP and hardware timestamps	Yes
1PPS	No
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare XtremeScale™ SFN8542 Dual-Port 40GbE QSFP+ PCIe 3.1 Server I/O Adapter

Part numbers	SFN8542 or SFN8542-Plus
Controller silicon	SFC9240
Power	12.5W typical
PCI Express	16 lanes Gen 3.1 (8.0GT/s), x16 edge connector
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled for the Plus version)
PTP and hardware timestamps	Yes (factory enabled for the Plus version)
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x QSFP+ (40G/10G)

**Solarflare XtremeScale™ SFN8522M Dual-Port 10GbE SFP+ PCIe 3.1 Server I/O Adapter**

Part numbers	SFN8522M, SFN8522M-Onload, or SFN8522M-Plus
Controller silicon	SFC9240
Power	10.5W typical
PCI Express	8 lanes Gen 3.1 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled for the Onload and Plus versions)
PTP and hardware timestamps	Yes (factory enabled for the Plus version)
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare XtremeScale™ SFN8522 Dual-Port 10GbE SFP+ PCIe 3.1 Server I/O Adapter

Part numbers	SFN8522, SFN8522-Onload, or SFN8522-Plus
Controller silicon	SFC9240
Power	10.5W typical
PCI Express	8 lanes Gen 3.1 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled for the Onload and Plus versions)
PTP and hardware timestamps	Yes (factory enabled for the Plus version)
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)



Solarflare XtremeScale™ SFN8042 Dual-Port 40GbE QSFP+ PCIe 3.1 Server I/O Adapter

Part numbers	SFN8042
Controller silicon	SFC9240
Power	12.5W typical
PCI Express	8 lanes Gen 3.1 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes
PTP and hardware timestamps	Yes
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x QSFP+ (40G/10G)

Solarflare Flareon™ Network Adapters

Solarflare Flareon™ Ultra SFN7322F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Part number	SFN7322F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	Yes (factory enabled)
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

**Solarflare Flareon™ Ultra SFN7142Q Dual-Port 40GbE QSFP+ PCIe 3.0 Server I/O Adapter**

Part number	SFN7142Q
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x QSFP+ (40G/10G)

Solarflare Flareon™ Ultra SFN7124F Quad-Port 10GbE SFP+ PCIe 3.0 Server I/O Adapter

Part number	SFN7124F
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	4 x SFP+ (10G/1G)

**Solarflare Flareon™ Ultra SFN7122F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter**

Part number	SFN7122F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	1Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts.
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	AppFlex™ activation key required
1PPS	Optional bracket and cable assembly – not factory installed.
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare Flareon™ SFN7042Q Dual-Port 40GbE QSFP+ PCIe 3.0 Server I/O Adapter

Part number	SFN7042Q
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Enabled by installing AppFlex activation key
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	Optional bracket and cable assembly – not factory installed
SR-IOV	Yes
Network ports	2 x QSFP+ (40G/10G)

**Solarflare Flareon™ Ultra SFN7024F Quad-Port 10GbE SFP+ PCIe 3.0 Server I/O Adapter**

Part number	SFN7024F
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Enabled by installing AppFlex activation key
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	No
SR-IOV	Yes
Network ports	4 x SFP+ (10G/1G)

Solarflare Flareon™ Ultra SFN7022F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Part number	SFN7022F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts.
Supports OpenOnload	Enabled by installing AppFlex activation key
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	Optional bracket and cable assembly – not factory installed.
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

**Solarflare Flareon™ SFN7004F Quad-Port 10GbE SFP+ PCIe 3.0 Server I/O Adapter**

Part number	SFN7004F
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Enabled by installing AppFlex activation key
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	No
SR-IOV	Yes
Network ports	4 x SFP+ (10G/1G)

Solarflare Flareon™ SFN7002F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Part number	SFN7002F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts.
Supports OpenOnload	Enabled by installing AppFlex activation key
PTP and hardware timestamps	Enabled by installing AppFlex activation key
1PPS	Optional bracket and cable assembly – not factory installed.
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

1.3 Software Driver Support

The software driver is currently supported on the following distributions:

- Windows® Server 2008 R2
- Windows® Server 2012
- Windows® Server 2012 R2
- Windows® Server 2016
- Red Hat Enterprise Linux 6 (6.5 or later)
- Red Hat Messaging Realtime and Grid 2 update 5
- Red Hat Enterprise Linux 7.x
- Red Hat Enterprise Linux for Realtime 7.x
- SUSE Linux Enterprise Server 11 (SP3 or later), and 12 (base release)
- SUSE Linux Enterprise Real Time 11 (SP3 or later)
- Ubuntu 14.04 LTS, 14.10, 15.04, 15.10, 16.04 and 18.04 LTS
- Debian 7.x, 8.x and 9.x
- FreeBSD 10.x
- VMware® ESXi™ 6.0-u3e (and higher 6.0-uX versions) ESXi™ 6.5, ESXi™ 6.7
- Linux® KVM.
- Kernel.org Linux kernels 2.6.18 to 4.18

Support includes all minor updates/releases/service packs of the above major releases, for which the distributor has not yet declared end of life/support.

The Solarflare accelerated network middleware, OpenOnload and EnterpriseOnload, is supported on all Linux, Ubuntu, and Debian variants listed above, and is available for all Solarflare Onload network adapters. Solarflare are not aware of any issues preventing OpenOnload installation on other Linux variants such as Centos and Fedora.

1.4 Solarflare AppFlex™ Technology.

Solarflare AppFlex technology allows Solarflare server adapters to be selectively configured to enable on-board applications. AppFlex activation keys are required to enable selected functionality on the Solarflare XtremeScale™ and Flareon™ adapters and on the AOE ApplicationOnload™ Engine.

Customers can obtain access to AppFlex applications via their Solarflare sales channel by obtaining the corresponding AppFlex authorization code. The authorization code allows the customer to generate activation keys at the [MyAppFlex](https://support.solarflare.com/myappflex) page at <https://support.solarflare.com/myappflex>.

The sfkey utility application is used to install the generated activation key file on selected adapters. For detailed instructions for sfkey and key file installation refer to [Activation key install with sfkey on page 87](#).

1.5 Open Source Licenses

Solarflare Boot Manager

The Solarflare Boot Manager is installed in the adapter's flash memory. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Controller Firmware

The firmware running on the SFC9xxx controller includes a modified version of libcoroutine. This software is free software published under a BSD license reproduced below:

Copyright (c) 2002, 2003 Steve Dekorte

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.6 Support and Driver Download

Solarflare network drivers, RPM packages and documentation are available for download from <https://support.solarflare.com/>.

Software and documentation for OpenOnload and EnterpriseOnload is available from www.openonload.org.

1.7 Regulatory Information and Approvals

Refer to www.solarflare.com/quickstart.

2

Installation

This chapter covers the following topics:

- [Solarflare Network Adapter Products on page 18](#)
- [Fitting a Full Height Bracket \(optional\) on page 19](#)
- [Inserting the Adapter in a PCI Express \(PCIe\) Slot on page 19](#)
- [Attaching a Cable \(RJ-45\) on page 20](#)
- [Attaching a Cable \(SFP+, SFP28\) on page 21](#)
- [Cables and Transceivers on page 24](#)
- [Supported Speed and Mode on page 34](#)
- [Forward Error Correction on page 35](#)
- [LED States on page 37](#)
- [Port Modes on page 38](#)
- [Single Optical Fiber - RX Configuration on page 44](#)
- [Solarflare Precision Time Synchronization Adapters on page 45](#)
- [Solarflare ApplicationOnload™ Engine on page 45](#)



CAUTION: Servers contain high voltage electrical components. Before removing the server cover, disconnect the mains power supply to avoid the risk of electrocution.



CAUTION: Static electricity can damage computer components. Before handling computer components, discharge static electricity from yourself by touching a metal surface, or wear a correctly fitted anti-static wrist band.

2.1 Solarflare Network Adapter Products

Solarflare XtremeScale™ adapters

- Solarflare XtremeScale X2542 Dual-Port 100GbE QSFP28 PCIe 3.1 Server Adapter
- Solarflare XtremeScale X2541 Single-Port 100GbE QSFP28 PCIe 3.1 Server Adapter
- Solarflare XtremeScale X2522-25G Dual-Port 10GbE/25GbE SFP28 PCIe 3.1 Server Adapter
- Solarflare XtremeScale X2522 Dual-Port 10GbE SFP28 PCIe 3.1 Server Adapter
- Solarflare XtremeScale SFN8722 Dual-Port 10GbE SFP+ PCIe 3.1 OCP Server Adapter
- Solarflare XtremeScale SFN8542 Dual-Port 40GbE PCIe 3.1 QSFP+ Server Adapter
- Solarflare XtremeScale SFN8522M Dual-Port 10GbE PCIe 3.1 SFP+ Server Adapter
- Solarflare XtremeScale SFN8522 Dual-Port 10GbE PCIe 3.1 SFP+ Server Adapter
- Solarflare XtremeScale SFN8042 Dual-Port 40GbE PCIe 3.1 QSFP+ Server Adapter.

Solarflare Flareon™ adapters

- Solarflare Flareon Ultra SFN7322F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter
- Solarflare Flareon Ultra SFN7142Q Dual-Port 40GbE PCIe 3.0 QSFP+ Server Adapter
- Solarflare Flareon Ultra SFN7124F Quad-Port 10GbE PCIe 3.0 SFP+ Server Adapter
- Solarflare Flareon Ultra SFN7122F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter
- Solarflare Flareon SFN7042Q Dual-Port 40GbE PCIe 3.0 QSFP+ Server Adapter
- Solarflare Flareon Ultra SFN7024F Quad-Port 10GbE PCIe 3.0 SFP+ Server Adapter
- Solarflare Flareon Ultra SFN7022F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter
- Solarflare Flareon SFN7004F Quad-Port 10GbE PCIe 3.0 SFP+ Server Adapter
- Solarflare Flareon SFN7002F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter.

CPU/PCIe Requirements

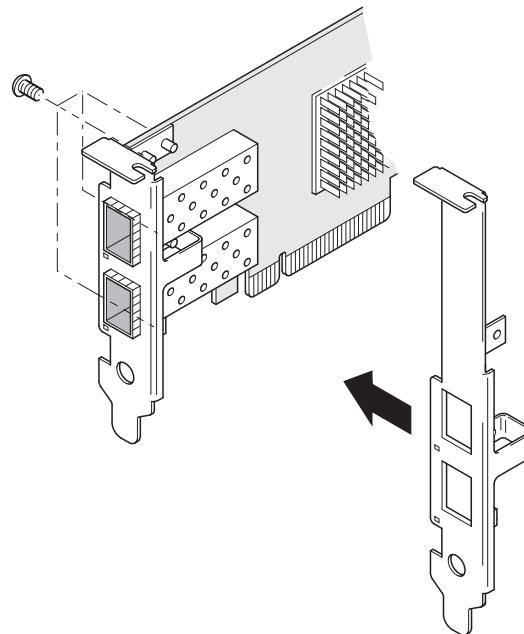
Solarflare network adapters can be installed on Intel/AMD x86 based 32 bit or 64 bit servers. The network adapter must be inserted into a PCIe x8 OR PCIe x 16 slot for maximum performance.

2.2 Fitting a Full Height Bracket (optional)

Solarflare adapters are supplied with a low-profile bracket fitted to the adapter. A full height bracket has also been supplied for PCIe slots that require this type of bracket.

To fit a full height bracket to the Solarflare adapter:

- 1 From the back of the adapter, remove the screws securing the bracket.
- 2 Slide the bracket away from the adapter.
- 3 Taking care not to overtighten the screws, attach the full height bracket to the adapter.

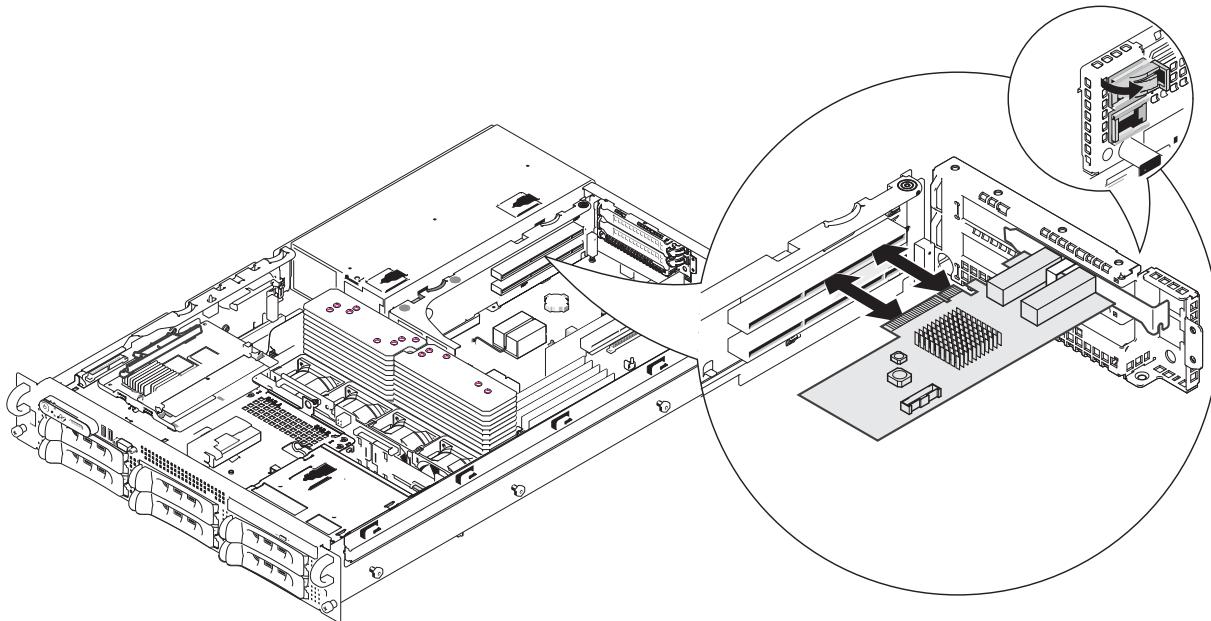


2.3 Inserting the Adapter in a PCI Express (PCIe) Slot

To insert the adapter in a PCI Express (PCIe) slot:

- 1 Shut down the server and unplug it from the mains. Remove the server cover to access the PCIe slots in the server.
- 2 Locate an 8-lane or 16-lane PCIe slot (refer to the server manual if necessary) and insert the Solarflare card.

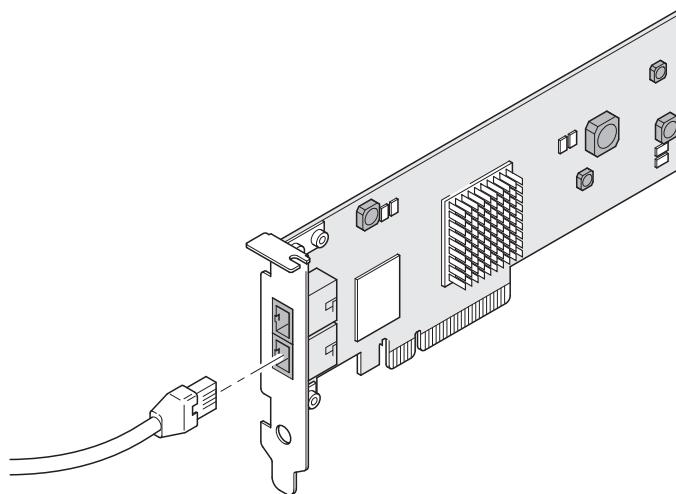
- 3 Secure the adapter bracket in the slot.
- 4 Replace the cover and restart the server.



- 5 After restarting the server, the host operating system may prompt you to install drivers for the new hardware. Click Cancel or abort the installation and refer to the relevant chapter in this manual for how to install the Solarflare adapter drivers for your operating system.

2.4 Attaching a Cable (RJ-45)

Solarflare 10GBASE-T Server Adapters connect to the Ethernet network using a copper cable fitted with an RJ-45 connector (shown below).



RJ-45 Cable Specifications

Table 1 below lists the recommended cable specifications for various Ethernet port types. Depending on the intended use, attach a suitable cable. For example, to achieve 10 Gb/s performance, use a Category 6 cable. To achieve the desired performance, the adapter must be connected to a compliant link partner, such as an IEEE 802.3an-compliant gigabit switch.

Table 1: RJ-45 Cable Specification

Port type	Connector	Media Type	Maximum Distance
10GBASE-T	RJ-45	Category 6A	100m (328 ft.)
		Category 6 unshielded twisted pairs (UTP)	55m (180 ft.)
		Category 5E	55m (180 ft.)
1000BASE-T	RJ-45	Category 5E, 6, 6A UTP	100m (328 ft.)
100BASE-TX	RJ-45	Category 5E, 6, 6A UTP	100m (328 ft.)

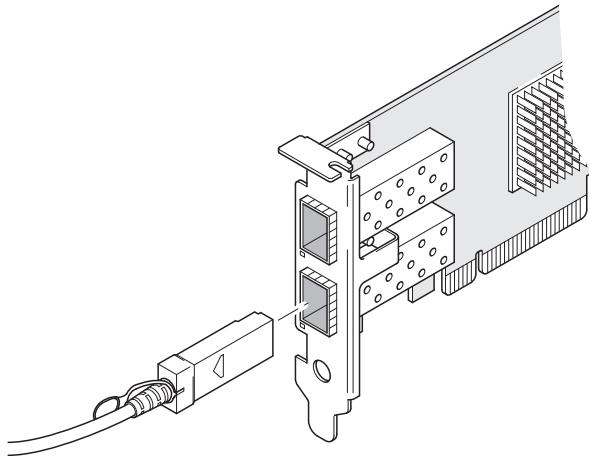
2.5 Attaching a Cable (SFP+, SFP28)

Solarflare SFP+, SFP28 Server Adapters can be connected to the network using either a Direct Attach cable or a fiber optic cable.

Attaching a Direct Attach Cable

To attach a Direct Attach cable:

- 1 Turn the cable so that the connector retention tab and gold fingers are on the same side as the network adapter retention clip.
- 2 Push the cable connector straight in to the adapter socket until it clicks into place.



Removing a Direct Attach Cable

To remove a Direct Attach cable:

- 1 Pull straight back on the release ring to release the cable retention tab. Alternatively, you can lift the retention clip on the adapter to free the cable if necessary.
- 2 Slide the cable free from the adapter socket.

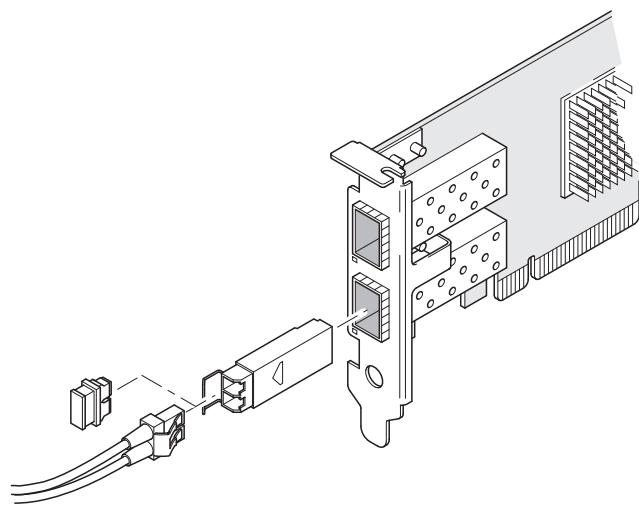
Attaching a fiber optic cable



WARNING: Do not look directly into the fiber transceiver or cables as the laser beams can damage your eyesight.

To attach a fiber optic cable:

- 1 Remove and save the fiber optic connector cover.
- 2 Insert a fiber optic cable into the ports on the network adapter bracket as shown. Most connectors and ports are keyed for proper orientation. If the cable you are using is not keyed, check to be sure the connector is oriented properly (transmit port connected to receive port on the link partner, and vice versa).



Removing a fiber optic cable



WARNING: *Do not look directly into the fiber transceiver or cables as the laser beams can damage your eyesight.*

To remove a fiber optic cable:

- 1 Remove the cable from the adapter bracket and replace the fiber optic connector cover.
- 2 Pull the plastic or wire tab to release the adapter bracket.
- 3 Hold the main body of the adapter bracket and remove it from the adapter.

2.6 Cables and Transceivers

The following tables identify adapter cables and transceiver modules that have been tested by Solarflare Communications.

Solarflare are not aware of any issues preventing the use of other brands with Solarflare adapters.

- SFP28 Direct Attach Cables [Table 2 on page 24](#)
- QSFP28 to SFP28 Splitter DAC Cables [Table 3 on page 25](#)
- QSFP28 to QSFP28 Cables [Table 4 on page 26](#)
- QSFP Optical Transceivers [Table 5 on page 26](#)
- SFP28 SR Optical Transceivers [Table 6 on page 27](#)
- SFP+ Direct Attach Cables [Table 7 on page 27](#)
- SFP+ 10G SR Optical Transceivers [Table 8 on page 28](#)
- SFP+ 10G LR Optical Transceivers [Table 9 on page 29](#)
- SFP 1000BASE-T Transceivers [Table 10 on page 30](#)
- 1G Optical Transceivers [Table 11 on page 30](#)

Table 2: SFP28 Direct Attach Cables

Manufacturer	Product Code	Notes	X2 10G	X2 25G	X2541	X2542
Amphenol	SF-NDCCGF28GB-000.5M	30AWG (CA-N)	✓	✓	✓	
Amphenol	SF-NDCCGF28GB-001M	30AWG (CA-N)	✓	✓	✓	
Amphenol	SF-NDCCGJ28GB-002M	26AWG (CA-N)	✓	✓	✓	
Amphenol	SF-NDCCGJ28GB-003M	26AWG (CA-N)	✓	✓	✓	
Amphenol	SF-NDCCGF28GB-002M	30AWG (CA-S)	✓	✓	✓	
Amphenol	SF-NDCCGF28GB-003M	30AWG (CA-L)	✓	✓	✓	
Amphenol	SF-NDCCGJ28GB-005M	26AWG (CA-L)	✓	✓	✓	
Arista	CAB-S-S-25G-1M	1M	✓	✓		
Arista	CAB-S-S-25G-3M	3M	✓	✓		
Arista	CAB-S-S-25G-5M	5M	✓	✓		
Fiberstore	S28-PC01	30AWG	✓	✓	✓	
Fiberstore	S28-PC05	26AWG	✓	✓	✓	

Table 2: SFP28 Direct Attach Cables

Manufacturer	Product Code	Notes	X2 10G	X2 25G	X2541	X2542
Siemon	S1S28P301.0-01P	1M (CA-N) 30AWG	✓	✓	✓	
Siemon	S1S28P302.0-01P	2M (CA-N) 30AWG	✓	✓	✓	
Siemon	S1S28P265.0-01P	5M (CA-L) 26AWG	✓	✓	✓	

Table 3: QSFP28 to SFP28 Splitter DAC Cables

Manufacturer	Product Code	Notes	X2 10G	X2 25G	X2541	X2542
Amphenol	SF-NDAQGF-C101	1M 30AWG (CA-N)	✓	✓	✓	✓
Amphenol	SF-NDAQGF-C102	2M 30AWG (CA-N)	✓	✓	✓	✓
Amphenol	SF-NDAQGF-C103	2M 26AWG (CA-S)	✓	✓	✓	✓
Amphenol	SF-NDAQGF-C103	3M 30AWG (CA-L)	✓	✓	✓	✓
Amphenol	SF-NDAQGJ-C102	5M 26AWG (CA-L)	✓	✓	✓	✓
Fiberstore	Q-4S28PC01 Cisco QSFP-4SFP25G CU1M	1M 30AWG	✓	✓	✓	✓
Fiberstore	A-4S2803	3M	✓	✓	✓	✓
Legrand	100G4SFPPDAC3M-LEG	28AWG		✓	✓	
Legrand	100G2X25GPD2M-LEG	30AWG		✓	✓	
ProLabs	QSFP28-2SFP28-PDAC3M-13-C	30AWG		✓	✓	
Siemon 0.5M	Q4S28P300.5-01P	30AWG (CA-N)	✓	✓	✓	✓
Siemon 1M	Q4S28P301.0-01P	30AWG (CA-N)	✓	✓	✓	✓
Siemon 1.5M	Q4S28P301.5-01P	30AWG (CA-N)	✓	✓	✓	✓
Siemon 2M	Q4S28P302.0-01P	30AWG (CA-N)	✓	✓	✓	✓
Siemon 3M	Q4S28P263.0-01P	26AWG (CA-N)	✓	✓	✓	✓
Siemon 5M	Q4S28P265.0-01P	26AWG (CA-I)	✓	✓	✓	✓

Table 4: QSFP28 to QSFP28 Cables

Manufacturer	Product Code	Notes	X2 10G	X2 25G	X2541	X2542
Amphenol	NDAAFF-C101	1M 30AWG				✓
Amphenol	NDAAFF-C102	2M 30AWG				✓
Amphenol	NDAAFJ-C103	2M 26AWG				✓
Amphenol	NDAAFF-C103	2M 30AWG				✓
Amphenol	NDAAFJ-C104	3M 26AWG				✓
Amphenol	NDAAFJ-C102	5M 26AWG				✓
Arista	CAB-Q-Q-100G-1M, NDAAFF-0001	1M 30AWG				✓
Arista	CAB-Q-Q-100G-3M	3M 30AWG				✓
Arista	CAB-Q-Q-100G-5M, NDAAFJ-0004	5M 26AWG				✓
Fiberstore	Q28-PC01	1M 26AWG				✓
Fiberstore	Q28-PC02	2M 26AWG				✓
Fiberstore	Q28-PC03	3M 26AWG				✓
Legrand	QSFP100GPDAC3M-LEG	3M 26AWG				✓
Legrand	100G2X50GPD3M-LEG	2M 28AWG				✓
ProLabs	QSFP-100G-CU3M-C	3M 28AWG				✓
ProLabs	QSFP28-2QSFP28- PDAC3M-12-34-C	3M 30AWG				✓
ProLabs	QSFP28-1QSFP28- PDAC3M-12-C	3M 30AWG				✓

Table 5: QSFP Optical Transceivers

Manufacturer	Product Code	Notes	X2 10G	X2 25G	X2541	X2542
Arista	QSFP-100G-SR4	QSFP OPT-SR				✓
Avago	AFBR-89CDDZ	QSFP OPT-SR				✓
Finisar	FTLC9551REPM	QSFP OPT-SR				✓
Juniper	JNP-QSFP-100G-SR4	QSFP OPT-SR				✓

Table 6: SFP28 SR Optical Transceivers

Manufacturer	Product Code	Notes	X2 10G	X2 25G	X2541	X2542
Cisco	SFP-25G-SR-S	S25 Gigabit LAN	✓	✓		
Fiberstore	SFP28-25GSR-85		✓	✓		
Finisar	FTLF8536P4BCL		✓	✓		
Finisar	FTLF8538P4BCL		✓	✓		
Finisar	FTLF8536P4BCV		✓	✓		

Table 7: SFP+ Direct Attach Cables

	Product Code	Cable	Notes	SFN7XXX	SFN8XXX	X2 10G
Arista	CAB-SFP-SFP-1M	1m		✓	✓	✓
Arista	CAB-SFP-SFP-3M	3m		✓	✓	✓
Arista	CBL-00006-02	5m			✓	
Cisco	SFP-H10GB-CU1M	1m		✓	✓	
Cisco	SFP-H10GB-CU3M	3m		✓	✓	
Cisco	SFP-H10GB-CU5M	5m		✓	✓	
Cisco	CISCO-TYCO 1-2053783-1	1m			✓	
Cisco	CISCO-LOROM LRHSPB54D030	3m			✓	
Cisco	CISCO-LOROM LRHSPB54A050	5m			✓	
HP	J9283A/B Procurve	3m		✓	✓	
HPE	MergeOptics GmbH 10119467-3030LF	3m			✓	
Juniper	EX-SFP-10GE-DAC-1m	1m		✓	✓	
Juniper	EX-SFP-10GE-DAC-3m	3m		✓	✓	
Juniper	Amphenol 584990001	1m			✓	
Juniper	Amphenol 584990002	3m			✓	
Molex	74752-1101	1m		✓	✓	✓
Molex	74752-2301	3m		✓	✓	✓

Table 7: SFP+ Direct Attach Cables

	Product Code	Cable	Notes	SFN7XXX	SFN8XXX	X2 10G
Molex	74752-3501	5m		✓	✓	✓
Molex	74752-9093	1m	37-0960-01 / 0K585N	✓		
Molex	74752-9094	3m	37-0961-01 / 0J564N	✓		
Molex	74752-9096	5m	37-0962-01 / 0H603N	✓		
Panduit	PSF1PXA1M	1m		✓	✓	
Panduit	PSF1PXA3M	3m		✓	✓	
Panduit	PSF1PXD5MBU	5m		✓	✓	
Panduit	PSF1PXA1MBL	1m				✓
Panduit	PSF1PXA3MBL	3m				✓
Panduit	PSF1PXA5MBL	5m				✓
Siemon	SFPP30-01	1m		✓	✓	✓
Siemon	SFPP30-02	2m		✓	✓	✓
Siemon	SFPP30-03	3m		✓	✓	✓
Siemon	SFPP24-05	5m		✓	✓	✓
Tyco	2032237-2 D	1m		✓	✓	
Tyco	2032237-4	3m		✓	✓	
Tyco	2053125-2	1m				✓
Tyco	2127934-4	3m				✓

Table 8: SFP+ 10G SR Optical Transceivers

	Product Code	Notes	SFN7XXX	SFN8XXX	X2 -10G	X2 - 25G
Arista	SFP-10G-SR	10G	✓			
Arista	XVR-00002-02	10G		✓	✓	
Arista	XVR-10002-20	10G			✓	
Avago	AFBR-703SDZ	10G	✓	✓	✓	

Table 8: SFP+ 10G SR Optical Transceivers

	Product Code	Notes	SFN7XXX	SFN8XXX	X2-10G	X2 - 25G
Avago	AFBR-703SDDZ	Dual speed 1G/ 10G optic.	✓			
Avago	AFBR-703SMZ	10G	✓			
Avago	AFBR-709SMZ-SF1	10G		✓		
DELL	PLRXPL-SC-S43-811	10G			✓	
Finisar	FTLX1471D3BCL	10G			✓	
Finisar	FTLX8571D3BCL	10G	✓	✓	✓	✓
Finisar	FTLX8571D3BCL-SL	10G		✓		
Finisar	FTLX8571D3BCV	Dual speed 1G/ 10G optic.	✓	✓		
Finisar	FTLX8574D3BCL	10G	✓	✓	✓	
HP	456096-001	Also labeled as 455883-B21 and 455885-001	✓	✓	✓	
Intel	AFBR-703SDZ	10G	✓	✓	✓	
JDSU	PLRXPL-SC-S43-22-N	10G	✓		✓	
JDSU	PLRXPL-SC-S43-SF	10G		✓	✓	✓
Lumentum						
Juniper	AFBR-700SDZ-JU1	10G	✓			
MergeOptics	TRX10GVP2010	10G	✓	✓	✓	
Solarflare	SFM-10G-SR	10G	✓	✓	✓	✓
Vorboss	VBO-PXG-SR-300	10G	✓		✓	

Table 9: SFP+ 10G LR Optical Transceivers

Manufacturer	Product Code	Notes	SFN7xx	SFN8xx
Avago	AFCT-701SDZ	10G single mode fiber	✓	
Finisar	FTLX1471D3BCL	10G single mode fiber	✓	

Table 10: SFP 1000BASE-T Transceivers

Manufacturer	Product Code	Notes	SFN7xx	SFN8xx
Arista	SFP-1G-BT		✓	
Avago	ABCU-5710RZ		✓	✓
Cisco	30-1410-03		✓	
Dell	FCMJ-8521-3-(DL)		✓	✓
Finisar	FCLF-8521-3		✓	✓
Finisar	FCMJ-8521-3		✓	
Finisar	FCLF8522P2BTL		✓	
HP	453156-001		✓	✓
	453154-B21			
3COM	3CSFP93		✓	✓

Table 11: 1G Optical Transceivers

Manufacturer	Product Code	Type	SFN7xx	SFN8xx
Avago	AFBR-5710PZ	1000Base-SX	✓	
Cisco	GLC-LH-SM	1000Base-LX/LH	✓	
Cisco	30-1299-01	1000Base-LX	✓	
Finisar	FTLF8519P2BCL	1000Base-SX	✓	✓
Finisar	FTLF8519P3BNL	1000Base-SX	✓	
Finisar	FTLF1318P2BCL	1000Base-LX	✓	
Finisar	FTLF1318P3BTL	1000Base-LX	✓	✓
HP	453153-001	1000Base-SX	✓	
	453151-B21			

SFP 10GBASE-T Transceivers

Solarflare adapters do not support 10GBASE-T transceiver modules.

2.7 QSFP+ Transceivers and Cables

The following tables identify QSFP+ transceiver modules and cables tested by Solarflare with the SFN7000 and SFN8000 series QSP+ adapters.

Solarflare are not aware of any issues preventing the use of other brands of QSFP+ 40G transceivers and cables with Solarflare SFN7000 and SFN8000 series QSFP+ adapters.



NOTE: QSFP Cables may not work with all switches.

- QSFP+ 40GBASE-SR4 Transceivers [Table 12 on page 31](#)
- QSFP+ 40G Active Optical Cables (AOC) [Table 13 on page 31](#)
- QSFP+ 40G Direct Attach Cables [Table 14 on page 32](#)
- QSFP+ to SFP+ Breakout Cables [Table 15 on page 33](#)

Table 12: QSFP+ 40GBASE-SR4 Transceivers

Manufacturer	Product Code	Notes	SFN7xx	SFN8x42
Arista	AFBR-79E4Z	Standard 100m (OM3 Multimode fiber) range.	✓	✓
Avago	AFBR-79EADZ		✓	
Avago	AFBR-79EIDZ		✓	✓
Avago	AFBR-79EQDZ		✓	✓
Avago	AFBR-79EQPZ		✓	
Finisar	FTL410QE2C		✓	
JDSU	JQP-04SWAA1		✓	
JDSU	JDSU-04SRAB1		✓	
Solarflare	SFM-40G-SR4		✓	

Table 13: QSFP+ 40G Active Optical Cables (AOC)

Manufacturer	Product Code	Notes	SFN7xx	SFN8x42
Avago	AFBR-7QER05Z	3m	✓	
Finisar	FCBG410QB1C03	3m	✓	✓
Finisar	FCBN410QB1C05	5m	✓	

Table 14: QSFP+ 40G Direct Attach Cables

Manufacturer	Product Code	Notes	SFN7xx	SFN8x42
Arista	CAB-Q-Q-3M	3m	✓	✓
Arista	CAB-Q-Q-5M	5m	✓	
Cisco	QSFP-H40G-CU3M	3m	✓	✓
FCI	10093084-2010LF	1m	✓	✓
FCI	10093084-3030LF	3m	✓	✓
Molex	74757-1101	1m	✓	
Molex	74757-2101	1m		✓
Molex	74757-2301	3m	✓	✓
Panduit	40GBASE-CR4-PQSFPXA3MBU	3m		✓
Siemon	QSFP30-01	1m	✓	
Siemon	QSFP30-03	3m	✓	✓
Siemon	QSFP26-05	5m	✓	✓

QSFP+ to SFP+ Breakout Cables

Solarflare QSFP+ to SFP+ breakout cables enable users to connect Solarflare dual-port QSFP+ server I/O adapters to work as a quad-port SFP+ server I/O adapters. The breakout cables offer a cost-effective option to support connectivity flexibility in high-speed data center applications.

These high performance direct-attach assemblies support 2 lanes of 10 Gb/s per QSFP+ port and are available in lengths of 1 meters and 3 meters. The SOLR-QSFP2SFP-1M, -3M copper DAC cables are fully tested and compatible with the Solarflare SFN7142Q and SFN7042Q server I/O adapters. These cables are compliant with the SFF-8431, SFF-8432, SFF-8436, SFF-8472 and IBTA Volume 2 Revision 1.3 specifications.

Table 15: QSFP+ to SFP+ Breakout Cables

Manufacturer	Product Code	Notes	SFN7xx	SFN8x42
Solarflare	SOLR-QSFP2SFP-1M	1m		✓
Solarflare	SOLR-QSFP2SFP-3M	3m		✓
Arista	QSFP-4SFP CAB-Q-S-3M	3m		✓
Arista	QSFP-4SFP CAB-Q-S-5M	5m		✓
Mellanox	MC2609130-003	3m		✓
Panduit	PHQ4SFPXA1MBL	1m		✓
Prolabs	CU1.0M-QSFP-2SFP-NS-13-C	1m		✓
Prolabs	CU1.5M-QSFP-2SFP-NS-13-C	1.5m		✓
Siemon	SFPPQSFP30-01	1m		✓
Siemon	SFPPQSFP28-03	3m		✓
Siemon	SFPPQSFP28-05	5m		✓
10GTek	CAB-QSFP.4SFP-P1M	1m		✓
10GTek	CAB-QSFP.4SFP-P3M	3m		✓
10GTek	CAB-QSFP.4SFP-P5M	5m		✓

Breakout cables have been tested with the SFN8x42 family of adapters. Testing is not complete on other 8000 series adapters

2.8 Supported Speed and Mode

Solarflare network adapters support either QSFP+, SFP, SFP+, SFP28, QSFP28 or Base-T standards.

On Base-T adapters speeds supported are 1Gbps and 10Gbps. The adapters use auto negotiation to automatically select the highest speed supported in common with the link partner.

On SFP+ adapters the SFP module (transceiver) determines the supported speeds, typically SFP modules only support a single speed. Some Solarflare SFP+ adapters support dual speed optical modules that can operate at either 1Gbps or 10Gbps. However, these modules do not auto-negotiate link speed and operate at the maximum (10G) link speed unless explicitly configured to operate at a lower speed (1G).

On XtremeScale X2 series adapters, the SFP28 transceiver module determines supported speeds. These adapters support 1Gbps, 10Gbps or 25Gbps. QSFP28 modules support 1Gbps-100Gbps.

The tables below summarizes the speeds supported by Solarflare network adapters.

Supported Modes	Auto neg speed	Speed	Comment
QSFP+ direct attach or optical cables	No	10G or 40G	SFN8542, SFN8042, SFN7142Q, SFN7042Q
QSFP28	Yes	10G, 25G, 40G, 50G, 100G	X2541, X2542
SFP28 direct attach or optical	Yes	25G, 10G or 1G	X2 adapters
SFP+ direct attach or optical cables	No	10G	
SFP optical module (1G)	No	1G	
SFP+ optical module (10G/1G)	No	10G or 1G	Dual speed modules run at the maximum speed (10G) unless explicitly configured to the lower speed (1G)
SFP 1000BASE-T module	No	1G	These modules support only 1G and will not link up at 100Mbps

2.9 25G Link Speed

Solarflare X2 series adapters automatically detect the link speed and configuration of the link partner (switch) and no configuration is necessary on 25G links.

If the link partner/switch has auto-negotiation (AN) enabled, the adapter will determine this from the AN protocol. If auto-negotiation is disabled, the adapter will instead auto-configure from analyzing the received signal.

Solarflare recommend, in the majority of cases, to configure the 25G link properties on the connected switch port and leave the adapter in its default state. However, the adapter can also be configured manually if this is required.



NOTE: 25G links are only attempted when the DAC cable or transceiver module is rated for 25G operation. 25G links speed will not be attempted when using a standard 10G DAC cable/transceiver.

Forward Error Correction

On 25G adapters, Forward Error Correction (FEC) employs redundancy in the channel coding as a technique used to reduce bit errors (BER) in noisy or unreliable communications channels and over long cables. The receiver is able to detect and correct errors without the need for a reverse channel or data re-transmission.



NOTE: FEC can potentially impact latency with an additional error correction overhead of a few hundred nanoseconds.

Adapter 25G links can auto-negotiate whether to use FEC and what type of FEC to apply on a link.

25G Direct Attach Cables

DAC Cable	FEC Requirement
CA-25G-L up to 5m	requires RS-FEC
CA-25G-S up to 3m	requires either RS-FEC or BASE-R FEC (default is BASE-R)
CA-25G-N up to 3m	can work with RS-FEC, BASE-R FEC or (the default) no FEC.

25G Optical Cables

No support for auto-negotiation. By default RS-FEC is used.

Auto-negotiation/Link Training

On Solarflare 25G adapters, AN/LT is enabled by default and will negotiate the link speed and the type of FEC to be used.

FEC - Auto Configuration

When a 25G DAC is inserted, using auto-detect, the adapter will attempt to establish a link at the highest speed and detect the type of FEC being used.

If auto-detect fails, the adapter will fallback to use ‘parallel detect’ at 25G, then 10G then 1G. Parallel detect is required on 10G transceivers and for some switches that do not support AN/LT.

FEC - Manual Configuration

Ethtool

FEC settings can be configured with ethtool from version 4.8 on RHEL7.5 (kernel 3.10.0-862.14.4.el7.x86_64) or later RHEL versions or using generic kernel 4.14+.

- Identify FEC Settings

```
# ethtool --show-fec <interface>
```

- Set FEC

```
# ethtool --set-fec <interface> [encoding auto|off|rs|baser]
```

The auto setting means adapter firmware will attempt to use the FEC type required by the DAC/Optical cable.

sfctool

Solarflare sfctool is a network driver utility that adopts some of the more advanced Ethtool features, making these available on older Linux kernels. sfctool is available from the Solarflare Linux Utilities package:

SF-107601-LS Issue 56 which includes sfctool v7.3.1.1001.

Users should ensure they have a recent Solarflare adapter driver. sfctool is not supported with the OS ‘in-tree’ driver (sfc driver versions 4.0 or 4.1).

sfctool uses the same command format as ethtool - replace ‘ethtool’ with ‘sfctool’.

2.10 LED States

There are two LEDs on the Solarflare network adapter transceiver module. LED states are as follows

Table 16: LED States

Adapter Type	LED Description	State
QSFP+, SFP/ SFP+SFP28	Speed	Green (solid) at all speeds
	Activity	Flashing green when network traffic is present LEDs are OFF when there is no link present
BASE-T	Speed	Green (solid) 10Gbps Yellow (solid) 100/1000Mbps
	Activity	Flashing green when network traffic is present LEDs are OFF when there is no link present

2.11 Port Modes

Port modes are configured using the sfboot utility available for Linux, FreeBSD or ESXi systems and the SfConfig utility on Windows systems.

The port-mode is a GLOBAL option and applies to all ports on the adapter.

A server reboot (power off/on) is required following changes to port modes.

Adapter Model	port-mode
<i>SFN7000 series</i>	
7x22	[1x10G], [1x10G][1x10G]*
7x24	[1x10G][1x10G], [4x10G]*
7x42	[1x10G][1x10G], [4x10G], [1x40G][1x40G]*
<i>SFN8000 series</i>	
8x22	[1x10G][1x10G]*
8x41	[1x40G]*
8x42	[4x10G], [2x10G][2x10G], [1x40G][1x40G]*
<i>X2 series</i>	
X2522	[1x10/25G][1x10/25G]*
X2541	[4x10/25G], [2x50G], [1x100G]*
X2542	[4x10/25G], [2x10/25G][2x10/25G], [2x50G], [1x50G][1x50G]*, [1x100G]
The mode annotated with * is the Default port mode for that model	

Select a link below:

- [SFN7x42Q QSFP+ Adapter on page 39](#)
- [SFN8x42 QSFP+ Adapter on page 40](#)
- [QSFP28 X2542 and X2541 on page 41](#)
- [QSFP28 X2542, X2541 - 100G on page 41](#)
- [QSFP28 X2542, X2541 - 10G/25G on page 41](#)
- [X2542, X2541 - \[2x50G\] on page 42](#)
- [X2542 - \[1x50G\]\[1x50G\] on page 42](#)
- [X2542 - \[2x10/25G\]\[2x10/25G\] on page 43](#)

SFN7x42Q QSFP+ Adapter

SFN7x42Q adapters can operate as

- 2 x 10Gbps per QSFP+ port
- 1 x 40Gbps per QSFP+ port
- 1 x 10Gbps per QSFP+ port
- A configuration of 1 x 40G and 2 x 10G ports is not supported.

```
sfboot port-mode=[1x40G][1x40G]
```

```
sfboot port-mode=[1x10G][1x10G]
```

```
sfboot port-mode=[4x10G]
```

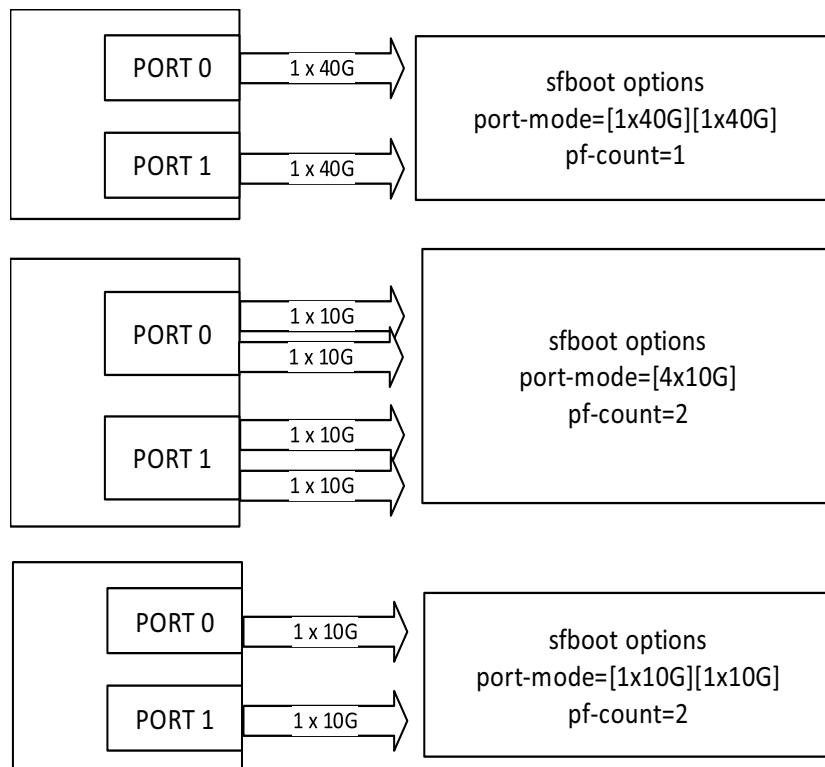


Figure 1: Port Configuration: SFN7x42Q

BreakOut Cables

The Solarflare 40G breakout cable has only 2 physical cables. Cables from other suppliers may have 4 physical cables. When connecting a third party breakout cable to the SFN7x42Q 40G QSFP+ cage (in 10G mode), **only cables 1 and 3 are active**.

SFN8x42 QSFP+ Adapter

SFN8x42 adapters can operate as

- 1 x 40G per QSFP+ port - 2nd cage is disabled
- 4 x 10G on one of the QSFP+ port
- 2 x 10G per QSFP+ port
- A configuration of 1 x 40G and 4 x 10G ports is not supported.

```
sfboot port-mode=[1x40G][1x40G]
```

```
sfboot port-mode=[2x10G][2x10G]
```

```
sfboot port-mode=[4x10G]
```

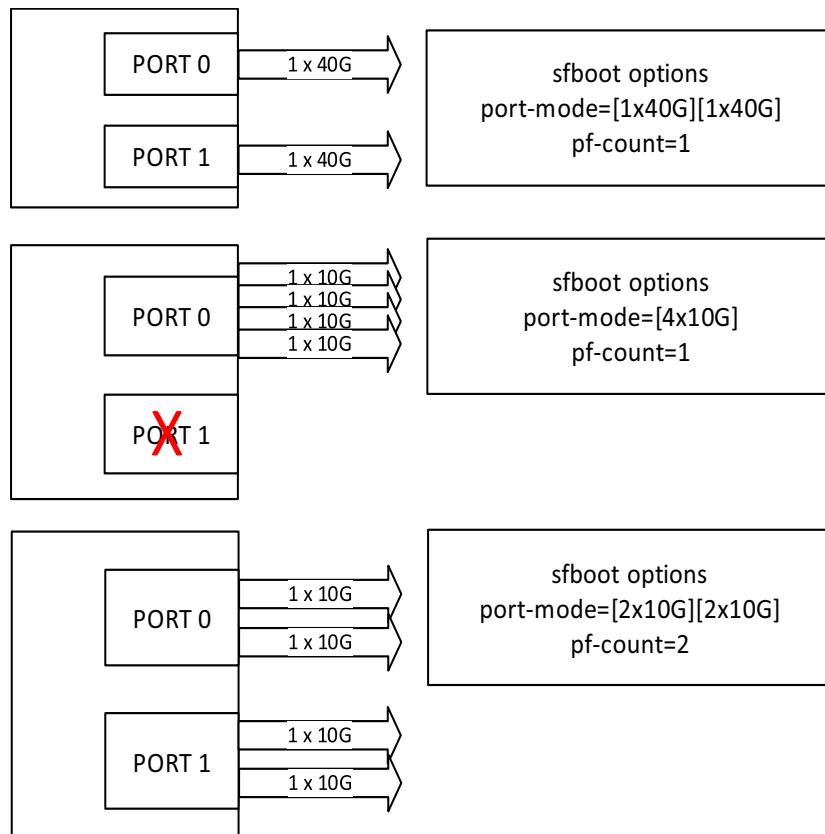


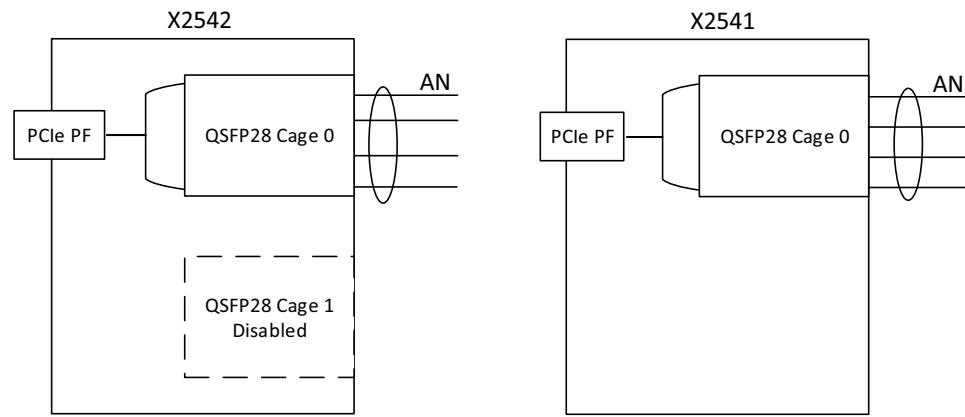
Figure 2: Port Configuration: SFN8x42

QSFP28 X2542 and X2541

The Solarflare X2541 and X2542 adapters have 4 x 10G/25G network ports, 2 x 50G ports or a single 100G port and can operate at 1G, 10G, 25G, 40G, 50G and 100G.

QSFP28 X2542, X2541 - 100G

- 1 x 100G on the first QSFP+ port. On the X2542 adapter, the 2nd cage is disabled
 - Exposes a single PCIe PF to the OS
- ```
sfboot port-mode=[1x100G] pf-count=1
```



**Figure 3: X254x - 100G**

### QSFP28 X2542, X2541 - 10G/25G

- Uses one QSFP28 port as 4 separate SFP28 ports. On the X2542 adapter, the 2nd cage is disabled.
  - Four PCIe PFs exposed to the OS - each port can operate at 1G, 10G or 25G
- ```
sfboot-mode=[4x10/25G] pf-count=4
```

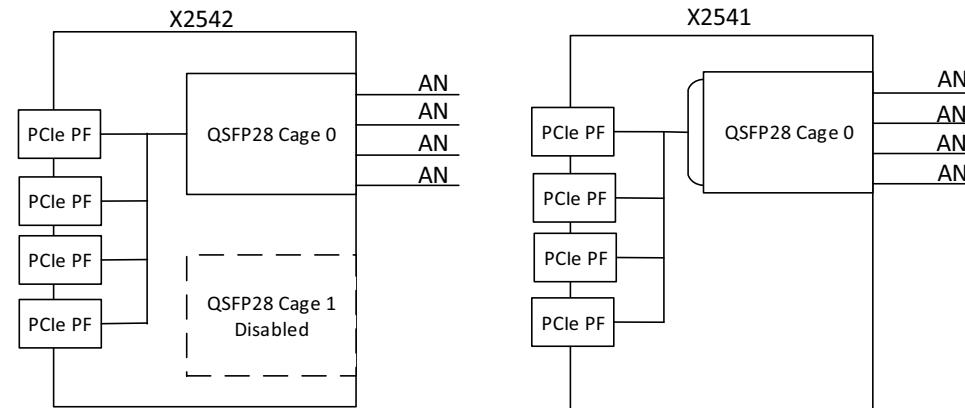


Figure 4: X254x - 1G, 10G, 25G

X2542, X2541 - [2x50G]

- X2 supports 2 x 50G MACs - each MAC needs 2 x 25G lanes
- One physical QSFP28 port used as 2 x 50G ports
- Two PCIe PFs exposed to the OS - each port can operate at 1G, 10G, 25G or 50G
- On the X2542 adapter, the 2nd cage is disabled

```
sfboot port-mode=[2x50G] pf-count=2
```

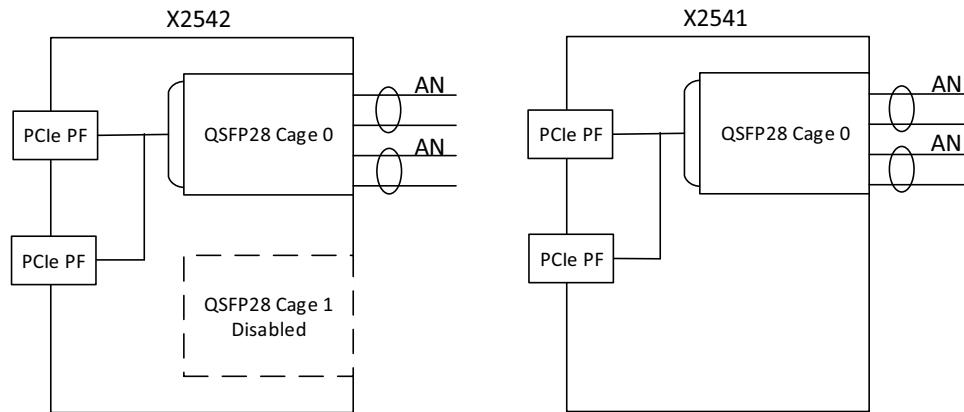


Figure 5: X254x - 2x50G

X2542 - [1x50G][1x50G]

- Uses both QSFP28 ports
- Two PCIe PFs exposed to the OS - each port can operate at 1G, 10G, 25G or 50G
- NIC does not advertise 40G or 100G on either port

```
sfboot port-mode=[1x50G][1x50G] pf-count=2
```

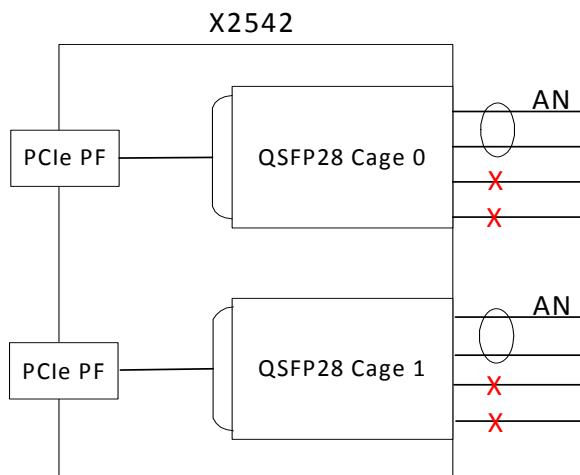


Figure 6: X2542 - 1x50G

X2542 - [2x10/25G][2x10/25G]

- Uses both QSFP28 ports
- Four PCIe PFs exposed to the OS - each port can operate at 1G, 10G or 25G

```
sfboot port-mode=[2x10/25G][2x10/25G] pf-count=2
```

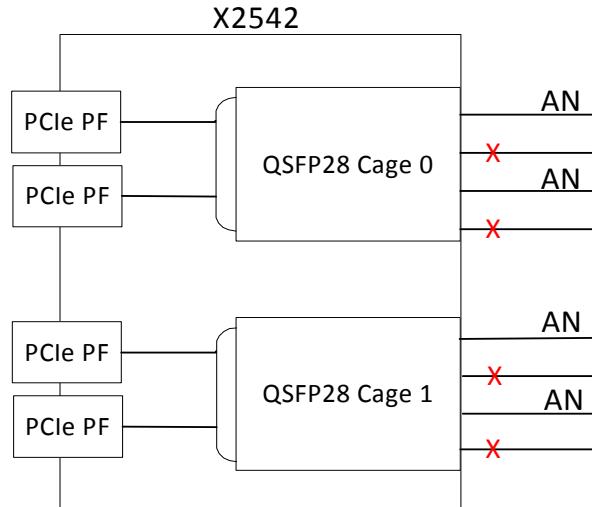


Figure 7: X2542 - 2X10/25G

2.12 Single Optical Fiber - RX Configuration

The Solarflare adapter will support a receive (RX) only fiber cable configuration when the adapter is required only to receive traffic, but have no transmit link. This can be used, for example, when the adapter is to receive traffic from a fiber tap device.

Solarflare have successfully tested this configuration on a 10G link on Flareon series and XtremeScale series adapters when the link partner is configured to be TX only (this will always be the case with a fiber tap). Some experimentation might be required when splitting the light signal to achieve a ratio that will deliver sufficient signal strength to all endpoints.



NOTE: Solarflare adapters do not support a receive only configuration on 1G links.

2.13 Solarflare Mezzanine Adapter: SFN8722 OCP

The Solarflare XtremeScale SFN8722 Dual-Port 10GbE SFP+ PCIe 3.1 OCP Server Adapter is an Open Compute Project mezzanine adapter for Ethernet connectivity.

The adapter meets the design requirements of the OCP Mezzanine Card 2.0 Design Specification.

- 1 Shut down the server and unplug from the power source before removing the server cover.
- 2 Locate the mezzanine slot and the SFP+ port slots - refer to the server manual if necessary.
- 3 Align the SFP+ cages with the port slots and seat the adapter in the mezzanine slot. Secure the adapter to the standoffs.

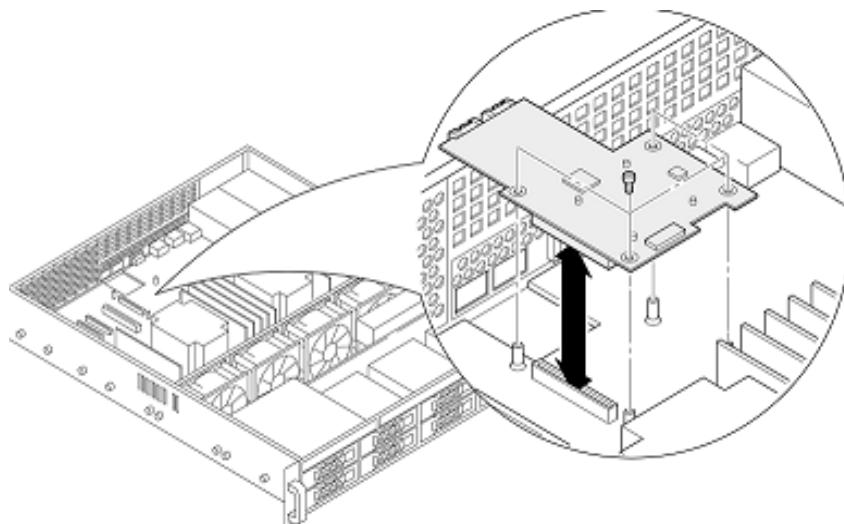


Figure 8: Installing the OCP Mezzanine Adapter



2.14 Solarflare Precision Time Synchronization Adapters

Solarflare adapters can generate hardware timestamps for PTP packets in support of a network precision time protocol deployment compliant with the IEEE 1588-2008 specification.

Some adapters require an additional AppFlex activation key to enable PTP/HW timestamping.

Customers requiring configuration instructions for these adapters and Solarflare PTP in a PTP deployment should refer to the *Solarflare Enhanced PTP User Guide* (SF-109110-CD).

2.15 Solarflare ApplicationOnload™ Engine

The ApplicationOnload™ Engine (AOE) SFA7942Q is a half-length, full-height PCIe form factor adapter combining the ultra-low latency dual-port 40GbE adapter with an Altera Stratix V FPGA.

For details of the SFA7942Q adapter refer to the Solarflare ApplicationOnload Users Guide (SF-115020-CD).

3

Solarflare Adapters on Linux

This chapter covers the following topics on the Linux® platform:

- [System Requirements on page 46](#)
- [Linux Platform Driver Feature Set on page 47](#)
- [Installing the Adapter Driver on page 49](#)
- [SUSE Linux Enterprise Server Distributions on page 52](#)
- [Installing DKMS Driver and Utilities on Ubuntu/Debian Servers on page 53](#)
- [Remove ‘in-tree’ Driver on page 54](#)
- [Configure the Solarflare Adapter on page 55](#)
- [Setting Up VLANs on page 57](#)
- [Setting Up Teams on page 58](#)
- [NIC Partitioning on page 59](#)
- [NIC Partitioning with SR-IOV on page 63](#)
- [Receive Side Scaling \(RSS\) on page 66](#)
- [Receive Flow Steering \(RFS\) on page 68](#)
- [Solarflare Accelerated RFS \(SARFS\) on page 70](#)
- [Transmit Packet Steering \(XPS\) on page 71](#)
- [Linux Utilities RPM on page 73](#)
- [Configuring the Boot Manager with sfboot on page 74](#)
- [Upgrading Adapter Firmware with sfupdate on page 82](#)
- [Activation key install with sfkey on page 87](#)
- [Performance Tuning on Linux on page 90](#)
- [Web Server - Driver Optimization on page 97](#)
- [Interrupt Affinity on page 99](#)
- [Module Parameters on page 109](#)
- [Linux ethtool Statistics on page 111](#)

3.1 System Requirements

Refer to [Software Driver Support on page 14](#) for supported Linux Distributions.

3.2 Linux Platform Driver Feature Set

Table 17: Linux Feature Set

Fault diagnostics	Support for comprehensive adapter and cable fault diagnostics and system reports. <ul style="list-style-type: none">• See Linux Utilities RPM on page 73
Firmware updates	Support for Boot ROM, Phy transceiver and adapter firmware upgrades. <ul style="list-style-type: none">• See Upgrading Adapter Firmware with sfupdate on page 82
Hardware Timestamps	Solarflare XtremeScale X2522, SFN8542-Plus, SFN8522-Plus and SFN8042 ¹ adapters, and Solarflare Flareon SFN7322F, SFN7142Q ¹ , SFN7124F ¹ , SFN7122F ¹ , SFN7042Q ¹ , SFN7024F ¹ , SFN7022F ¹ adapters support the hardware timestamping of all received packets - including PTP packets. The Linux kernel must support the SO_TIMESTAMPING socket option (2.6.30+) to allow the driver to support hardware packet timestamping. Therefore hardware packet timestamping is not available in RHEL 5.
Jumbo frames	Support for MTUs (Maximum Transmission Units) from 1500 bytes to 9216 bytes. <ul style="list-style-type: none">• See Configuring Jumbo Frames on page 57
PXE and UEFI booting	Support for diskless booting to a target operating system via PXE or UEFI boot. <ul style="list-style-type: none">• See Configuring the Boot Manager with sfboot on page 74• See Solarflare Boot Manager on page 285 PXE or UEFI boot are not supported for Solarflare adapters on IBM System p servers.
Receive Side Scaling (RSS)	Support for RSS multi-core load distribution technology. <ul style="list-style-type: none">• See Receive Side Scaling (RSS) on page 66.
ARFS	Linux Accelerated Receive Flow Steering. Improve latency and reduce jitter by steering packets to the core where a receiving application is running. See Receive Flow Steering (RFS) on page 68.

Table 17: Linux Feature Set

SARFS	Solarflare Accelerated RFS. See Solarflare Accelerated RFS (SARFS) on page 70 .
Transmit Packet Steering (XPS)	Supported on Linux 2.6.38 and later kernels. Selects the transmit queue when transmitting on multi-queue devices. See Transmit Packet Steering (XPS) on page 71 .
NIC Partitioning	Each physical port on the adapter can be exposed as up to 8 PCIe Physical Functions (PF). See NIC Partitioning on page 59 .
SR-IOV	Support for Linux KVM SR-IOV. <ul style="list-style-type: none"> • See SR-IOV Virtualization Using KVM on page 237 SR-IOV is not supported for Solarflare adapters on IBM System p servers.
Task offloads	Support for TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. <ul style="list-style-type: none"> • See Configuring Task Offloading on page 56
TX PIO	Use of programmed IO buffers in order to reduce latency for small packet transmission. <ul style="list-style-type: none"> • See TX PIO on page 96.
CTPIO	Cut Through PIO - TX packets are streamed directly from the PCIe interface to the adapter port bypassing the main TX datapath to deliver lowest TX latency. For details refer to the Onload User Guide (SF-104474-CD).
Teaming	Improve server reliability and bandwidth by combining physical ports, from one or more Solarflare adapters, into a team, having a single MAC address and which function as a single port providing redundancy against a single point of failure. <ul style="list-style-type: none"> • See Setting Up Teams on page 58
Virtual LANs (VLANS)	Support for multiple VLANs per adapter. <ul style="list-style-type: none"> • See Setting Up VLANs on page 57

1. Requires an AppFlex activation key - for details refer to [Solarflare AppFlex™ Technology](#) on page 15.

3.3 Installing the Adapter Driver

The Solarflare adapter driver can be installed from the following packages:

- Source DKMS (SF-104979-LS)
- Source RPM (SF-103848-LS)
- OpenOnload or EnterpriseOnload distribution.
- Drivers are available from: support.solarflare.com



NOTE: Onload users only need to install the Onload package - there is no need to install the driver again from RPM or DKMS.

The in-tree driver



CAUTION: Linux (and Linux based OS distributions) already include a version of the Solarflare adapter driver. This is known as the OS ‘in-tree’ driver.

The in-tree driver is good for normal networking operation, but does not support the following operations:

- Adapter Hardware timestamping
- sfptpd (Solarflare PTP daemon)
- sfkey (feature activation key utility)
- sfctool
- will not include more advanced/recent features available with later drivers

To identify if the in-tree driver is being used by the adapter:

```
# ethtool -i <interface>
driver: sfc
version: 4.0
```

The in-tree driver will be version 4.0 or 4.1.



CAUTION: A later version of the driver can be installed without removing the in-tree driver, but the in-tree driver will automatically reload unless removed from the OS initramfs.

Refer to [SUSE Linux Enterprise Server Distributions on page 52](#) below to remove the in-tree driver and rebuild the initramfs.

DKMS RPM

Dynamic Kernel Module Support is a framework where device driver source can reside outside the kernel source tree. This supports an easy method to rebuild modules when kernels are upgraded.

Requirements

DKMS must be installed on the server. If the following command returns nothing, then DKMS is not installed. Refer to Linux online documentation to install DKMS.

```
# dkms --version
```

Install

To install the Solarflare driver DKMS package execute the following command:

```
# rpm -i sfc-dkms-<version>.noarch.rpm
```

Load the driver

```
# modprobe sfc
```

Confirm

To check the adapter is using the newly installed driver (check version):

```
# ethtool -i <interface>
```

Source RPM

To install the driver from the source RPM package, the binary driver must be built from the source RPM, then installed and then loaded.

Build the Binary driver from the src RPM

Requirements

Kernel headers for the running kernel must be installed at /lib/modules/<kernel-version>/build.

- On Red Hat systems, install the appropriate kernel-smp-devel or kernel-devel package
- On SUSE systems install the kernel-source package

Build the Binary RPM (*example - version numbers may be different*)

- 1 Copy the driver distribution package to the server and unzip to reveal the source RPM file:

```
# unzip SF-103848-LS-46_Solarflare_NET_driver_source_RPM.zip  
Archive: SF-103848-LS-46_Solarflare_NET_driver_source_RPM.zip  
inflating: sfc-4.13.1.1034-1.src.rpm
```

2 Build the Binary

```
# rpmbuild --rebuild sfc-4.13.1.1034-1.src.rpm
```

3 The build procedure will generate a lot of console output. Towards the end of the build a '**Wrote**' line identifies the location of the built binary driver file:

```
Wrote: /root/rpmbuild/RPMS/x86_64/kernel-module-sfc-RHEL7-3.10.0-  
514.26.2.el7.x86_64-4.13.1.1034-1.x86_64.rpm
```

Install the Binary RPM

Copy the location from the previous build step to install the binary driver:

```
# rpm -ivh /root/rpmbuild/RPMS/x86_64/kernel-module-sfc-RHEL7-3.10.0-  
514.26.2.el7.x86_64-4.13.1.1034-1.x86_64.rpm
```

Load the Driver

```
# modprobe sfc
```

Confirm

To check the adapter is using the newly installed driver (check version):

```
# ethtool -i <interface>
```

Building for a different kernel

To build for a different kernel to the running system, enter the following command to identify the target kernel.

```
# rpmbuild --define 'kernel <kernel version>' --rebuild <package_name>
```

3.4 SUSE Linux Enterprise Server Distributions

Refer to [Build the Binary driver from the src RPM on page 50](#) to create the binary RPM.

Requirements

The Solarflare drivers are currently classified as 'unsupported' by SUSE Enterprise Linux 10 and 11. To allow unsupported drivers to load in SLES10, edit the following file:

```
/etc/sysconfig/hardware/config
```

Find the line:

```
LOAD_UNSUPPORTED_MODULES_AUTOMATICALLY=no
```

and change no to yes.

For SLES 11, edit the unsupported modules file in:

```
/etc/modprobe.d/unsupported-modules  
allow_unsupported_modules 1
```

Install the RPMs:

```
# rpm -ivh kernel-module-sfc-2.6.5-7.244-smp-2.1.0111-0.sf.1.SLES9.i586.rpm
```

Run YaST to configure the Solarflare Network Adapter.

When the Ethernet Controller is selected, the **Configuration Name** will take one of the following forms:

- eth-bus-pci-dddd:dd:dd.N where N is either 0 or 1.
- eth-id-00:0F:53:XX:XX:XX

Once configured, the **Configuration Name** for the correct Ethernet Controller will change to the second form, and an ethX interface will appear on the host.

If the incorrect Ethernet Controller is chosen and configured, then the **Configuration Name** will remain as eth-bus-pci-dddd:dd:dd.1 after configuration by YaST, and an ethX interface will not appear on the system. If this happens, should remove the configuration for this Ethernet Controller, and configure the other Ethernet Controller of the pair.

3.5 Installing DKMS Driver and Utilities on Ubuntu/Debian Servers

Solarflare recommend that the DKMS driver package is installed on the Ubuntu server and NOT the source RPM package. Onload users only need to install the Onload distribution which includes the adapter driver.

Net Driver DKMS

Requirements

dkms must be installed on the server. If the following command returns nothing, dkms is not installed - refer to OS online documentation to install dkms:

```
# dkms -version
```

The Solarflare net driver DKMS package (SF-104979-LS) is available from:

<https://support.solarflare.com/>

Create .deb file

- 1 Download the Solarflare net driver DKMS source package and unzip on the target server.
- 2 Create the .deb file:

```
sudo alien -c sfc-dkms-<version>.sf.1.noarch.rpm
```

This command generates the sfc-dkms_<version>_all.deb file.



NOTE: The -c option is required to convert source scripts and build the driver.

Install the deb file:

```
sudo dpkg -i -dkms_<version>_all.deb
```

Reload the sfc driver:

```
modprobe -r sfc  
modprobe sfc
```

3.6 Remove ‘in-tree’ Driver

The ‘in-tree’ driver is the Solarflare adapter driver included with the OS distribution. This driver will be built into the system initramfs and will automatically reload if the server is rebooted.

If the OS ‘in-tree’ driver is installed on the system. This can be removed, and the initramfs can be rebuilt before installing a newer DKMS driver.

- 1 To identify if the ‘in-tree’ driver is being used:

```
# ethtool -i <solarflare interface>
```

```
driver: sfc
```

```
version: 4.0 (this might also be the 4.1 driver)
```

- 2 To remove the ‘in-tree’ driver and rebuild the initramfs - so that the ‘in-tree’ driver does not automatically reload following reboot:

```
# find /lib/modules/$(uname -r) -name 'sfc*.ko' | xargs rm -rf
```

```
# rmmod sfc
```

```
# dracut -f
```

Utilities

The Solarflare Linux Utilities package (SF-107601-LS) is available from:

<https://support.solarflare.com/>

- 1 Download and unzip the package on the target server.

- 2 Create the .deb file:

```
sudo alien sfutils-<version>.x86_64.rpm
```

This command generates the sfutils_<version>_amd64.deb file.

- 3 Install the deb file:

```
sudo dpkg -i sfutils_<version>_amd64.deb
```

- 4 Utilities sfupdate, sfkey, sfctool and sfboot are available on the server.

3.7 Configure the Solarflare Adapter

Ethtool is a standard Linux tool to set, view and change Ethernet adapter settings.

```
ethtool <option> <interface>
```

Root permissions are required to configure the adapter.

Hardware Timestamps

The Solarflare Flareon series and XtremeScale series adapters can support hardware timestamping for all received network packets.

The Linux kernel must support the SO_TIMESTAMPING socket option (2.6.30+) therefore hardware packet timestamping is not supported on RHEL 5.

For more information about using the kernel timestamping API, users should refer to the Linux documentation: <http://lxr.linux.no/linux/Documentation/networking/timestamping.txt>

Configuring Speed and Modes

Solarflare adapters by default automatically negotiate the connection speed to the maximum supported by the link partner.

- On the 10GBASE-T adapters “auto” instructs the adapter to negotiate the highest speed supported in common with its link partner.
- On SFP28, SFP+ adapters, “auto” instructs the adapter to use the highest link speed supported by the inserted SFP+ module.

On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. Dual speed SFP+ modules operate at their maximum (10G) link speed unless explicitly configured to operate at a lower speed (1G).

The following commands demonstrate ethtool to configure the network adapter Ethernet settings.

- Identify interface configuration settings:
`ethtool ethX`
- Set link speed:
`ethtool -s ethX speed 1000|100`
- To return the connection speed to the default auto-negotiate, enter:
`ethtool -s <ethX> autoneg on`
- Configure auto negotiation:
`ethtool -s ethX autoneg [on|off]`
- Set auto negotiation advertised speed 1G:
`ethtool -s ethX advertise 0x20`
- Set autonegotiation advertised speed 10G:
`ethtool -s ethX advertise 0x2000`

- ```
ethtool -s ethX advertise 0x1000
```
- Set autonegotiation advertised speeds 1G and 10G:  

```
ethtool -s ethX advertise 0x1020
```
- Identify interface auto negotiation pause frame setting:  

```
ethtool -a ethX
```
- Configure auto negotiation of pause frames:  

```
ethtool -A ethX autoneg on [rx on|off] [tx on|off]
```



**NOTE:** Due to a limitation in ethtool, when auto-negotiation is enabled, the user must specify both speed and duplex mode or speed and set an advertise mask otherwise speed configuration will not function.

## Configuring Task Offloading

Solarflare adapters support transmit (Tx) and receive (Rx) checksum offload, as well as TCP segmentation offload. To ensure maximum performance from the adapter, all task offloads should be enabled, which is the default setting on the adapter. For more information, see [Performance Tuning on Linux on page 90](#).

To change offload settings for Tx and Rx, use the ethtool command:

```
ethtool --offload <ethX> [rx on|off] [tx on|off]
```

## Configuring Receive/Transmit Ring Buffer Size

By default receive and transmit ring buffers on the Solarflare adapter support 1024 descriptors. The user can identify and reconfigure ring buffer sizes using the ethtool command.

To identify the current ring size:

```
ethtool -g ethX
```

To set the new transmit or receive ring size to value N

```
ethtool -G ethX [rx N] [tx N]
```

The ring buffer size must be a value between 128 and 4096. On the SFN7000, SFN8000 and X2 series adapters the maximum TX buffer size is restricted to 2048. Buffer size can also be set directly in the modprobe.conf file or add the options line to a file under the /etc/modprobe.d directory e.g.

```
options sfc rx_ring=4096
```

Using the modprobe method sets the value for all Solarflare interfaces. Then reload the driver for the option to become effective:

```
modprobe -r sfc
modprobe sfc
```

## Configuring Jumbo Frames

Solarflare adapters support frame sizes from 1500 bytes to 9216 bytes. For example, to set a new frame size (MTU) of 9000 bytes, enter the following command:

```
ifconfig <ethX> mtu 9000
```

To make the changes permanent, edit the network configuration file for <ethX>; for example, /etc/sysconfig/network-scripts/ifcfg-eth1 and append the following configuration directive, which specifies the size of the frame in bytes:

```
MTU=9000
```

## 3.8 Setting Up VLANs

VLANs offer a method of dividing one physical network into multiple broadcast domains. In enterprise networks, these broadcast domains usually match with IP subnet boundaries, so that each subnet has its own VLAN. The advantages of VLANs include:

- Performance
- Ease of management
- Security
- Trunks
- You don't have to configure any hardware device, when physically moving your server to another location.

To set up VLANs, consult the following documentation:

- To configure VLANs on SUSE Linux Enterprise Server, see:  
<http://www.novell.com/support/viewContent.do?externalId=3864609>
- To configure tagged VLAN traffic only on Red Hat Enterprise Linux, see:  
<http://kbase.redhat.com/faq/docs/DOC-8062>
- To configure mixed VLAN tagged and untagged traffic on Red Hat Enterprise Linux, see:  
<http://kbase.redhat.com/faq/docs/DOC-8064>

## 3.9 Setting Up Teams

Teaming network adapters (network bonding) allows a number of physical adapters to act as one, virtual adapter. Teaming network interfaces, from the same adapter or from multiple adapters, creates a single virtual interface with a single MAC address.

The virtual adapter or virtual interface can assist in load balancing and providing failover in the event of physical adapter or port failure.

Teaming configuration support provided by the Linux bonding driver includes:

- 802.3ad Dynamic link aggregation
- Static link aggregation
- Fault Tolerant

To set up an adapter team, consult the following documentation:

- General:  
<http://www.kernel.org/doc/Documentation/networking/bonding.txt>
- RHEL 5:  
[http://www.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5.4/html/Deployment\\_Guide/s2-modules-bonding.html](http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Deployment_Guide/s2-modules-bonding.html)
- RHEL6:  
[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Deployment\\_Guide/s2-networkscripts-interfaces-chan.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-networkscripts-interfaces-chan.html)
- SLES:  
[http://www.novell.com/documentation/sles11/book\\_sle\\_admin/data/sec\\_basicnet\\_yast.html#sec\\_basicnet\\_yast\\_netcard\\_man](http://www.novell.com/documentation/sles11/book_sle_admin/data/sec_basicnet_yast.html#sec_basicnet_yast_netcard_man)

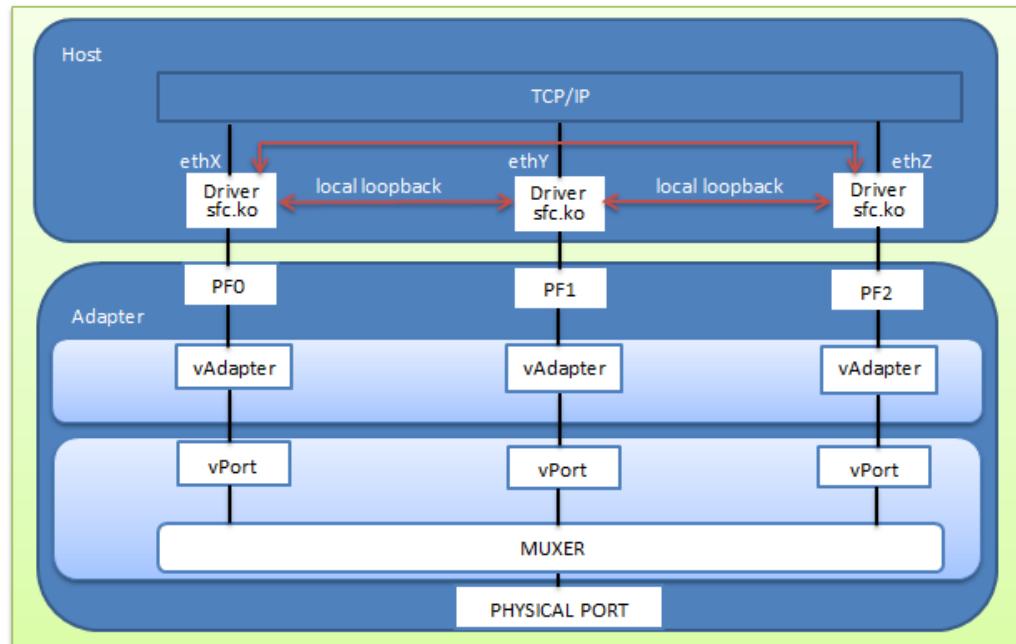
## 3.10 NIC Partitioning

NIC Partitioning is a feature supported on Solarflare adapters starting with the SFN7000 series. By partitioning the NIC, each physical network port can be exposed to the host as multiple PCIe Physical Functions (PF) with each having a unique interface name and unique MAC address.

When the Solarflare NET driver (sfc.ko) is loaded in the host, each PF is backed by a virtual adapter connected to a virtual port. A switching function supports the transport of network traffic between virtual ports (vport) and the physical port. Partitioning is particularly useful when, for example, splitting a single 40GbE interface into multiple PFs.

- Up to 16 PFs and 16 MAC addresses are support PER ADAPTER.
- On a 10GbE dual-port adapter each physical port can be exposed as a maximum 8 PFs.
- On a 40GbE dual-port adapter (in 2\*40G mode) each physical port can be exposed as a maximum 8 PFs.
- On a 40GbE dual-port adapter (in 4\*10G mode) each physical port can be exposed as a maximum 4 PFs.

### NIC Partitioning Without VLANs



**Figure 9: NIC Partitioning - without VLANs**

- Configured without VLANs, all PFs are in the same Ethernet layer 2 broadcast domain i.e. a packet broadcast from any one PF would be received by all other PFs.

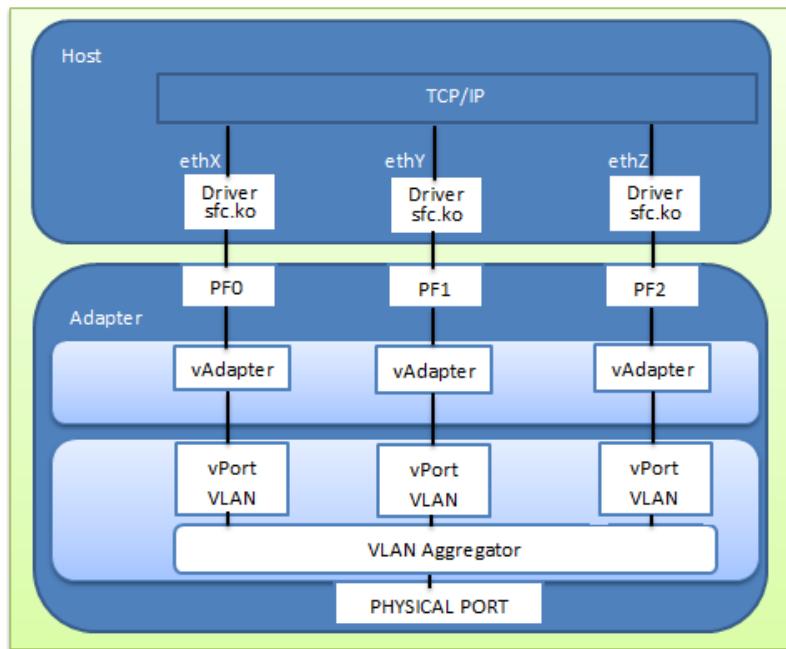
- Transmitted packets go directly to the wire. Packets sent between PFs are routed through the local TCP/IP stack loopback interface without touching the sfc driver.
- Received broadcast packets are replicated to all PFs.
- Received multicast packets are delivered to each subscriber.
- Received unicast packets are delivered to the PF with a matching MAC address. Because the TCP/IP stack has multiple network interfaces on the same broadcast domain, there is always the possibility that any interface could respond to an ARP request. To avoid this the user should use arp\_ignore=2 to avoid ARP cache pollution ensuring that ARP responses are only sent from an interface if the target IP address in the ARP request matches the interface address with both sender/receiver IP addresses in the same subnet.
- To set arp\_ignore for the current session:  
`echo 2 >/proc/sys/net/ipv4/conf/all/arp_ignore`
- To set arp\_ignore permanently (does not affect the current session), add the following line to the /etc/sysctl.conf file:  
`net.ipv4.conf.all.arp_ignore = 2`
- The MUXER function is a layer2 switching function for received traffic enabled in adapter firmware. When the OS delivers traffic to local interfaces via the loopback interface, the MUXER acts as a layer2 switch for both transmit and receive.

## VLAN Support

When PFs are configured with VLAN tags each PF must be in a different VLAN. The MUXER function acts as a VLAN aggregator such that transmitted packets are sent to the wire and received packets are demultiplexed based on the VLAN tags. VLAN tags are added/stripped by the adapter firmware transparent to the OS and driver. VLAN tags can be assigned when PFs are enabled using the sfboot command. A single PF can be assigned VLAN tag 0 allowing it to receive untagged traffic.

```
sfboot switch-mode=partitioning pf-count=3 pf-vlans=0,200,300
```

The first VLAN ID in the pf-vlans comma separated list is assigned to the first PF of the physical port and thereafter tags are assigned to PFs in lowest MAC address order.



**Figure 10: NIC Partitioning - VLAN Support**

## NIC Partitioning Configuration

Up to 16 PFs and 16 MAC addresses are supported per adapter. The PF count value applies to all physical ports. Ports cannot be configured individually.

- 1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.
- 2 The `sfboot` utility (`pf-count`) from the Solarflare Linux Utilities package (SF-107601-LS) is used to partition physical interfaces to the required number of PFs.
- 3 To partition all ports (example configures 4 PFs per port):

```
sfboot switch-mode=partitioning pf-count=4
Solarflare boot configuration utility [v4.5.0]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
```

|                             |                               |
|-----------------------------|-------------------------------|
| eth2:                       |                               |
| Boot image                  | Option ROM only               |
| Link speed                  | Negotiated automatically      |
| Link-up delay time          | 5 seconds                     |
| Banner delay time           | 2 seconds                     |
| Boot skip delay time        | 5 seconds                     |
| Boot type                   | Disabled                      |
| Physical Functions per port | 4                             |
| MSI-X interrupt limit       | 32                            |
| Number of Virtual Functions | 0                             |
| VF MSI-X interrupt limit    | 8                             |
| Firmware variant            | full feature / virtualization |
| Insecure filters            | Disabled                      |

|              |              |
|--------------|--------------|
| MAC spoofing | Disabled     |
| VLAN tags    | None         |
| Switch mode  | Partitioning |

*A cold reboot of the server is required for sfboot changes to be effective.*

- 4 Following reboot each PF will be visible using the `lspci` command:

```
lspci -d 1924:
```

```
07:00.0 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.1 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.2 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.3 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.4 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.5 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.6 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.7 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
```

- If more than 8 functions are required the server must support ARI - see [Alternative Routing-ID Interpretation \(ARI\) on page 239](#).
- Solarflare also recommend setting `pci=realloc` in the kernel configuration grub file - refer to [Kernel Configuration on page 239](#) for details.

- 5 To identify which physical port a given network interface is using:

```
cat /sys/class/net/eth<N>/device/physical_port
```

- 6 If the Solarflare driver is loaded, PFs will also be visible using the `ifconfig` command where each PF is listed with a unique MAC address.

## Software Requirements

The server must have the following (minimum) net driver and firmware versions to enable NIC Partitioning:

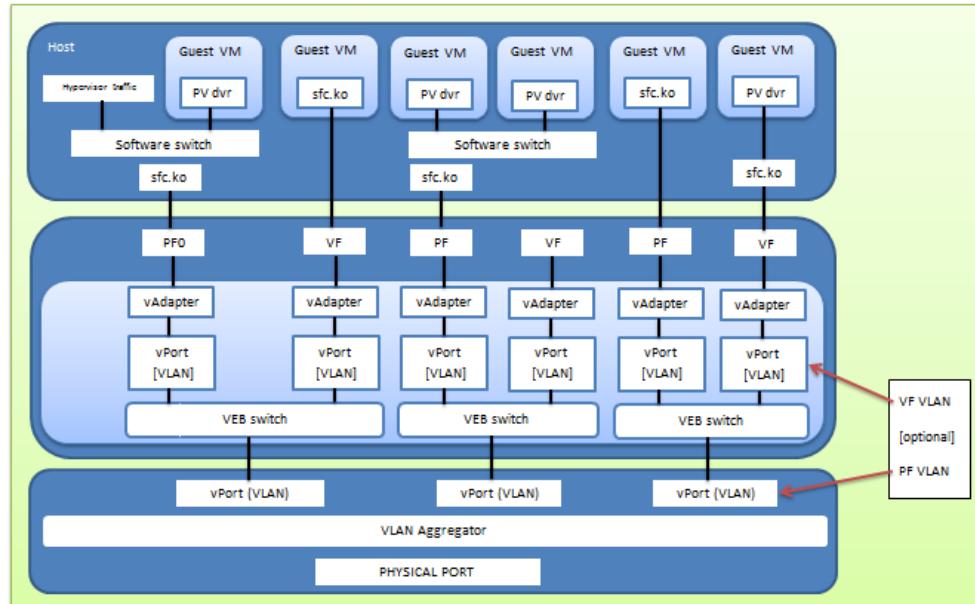
```
ethtool -i eth<N>
driver: sfc
version: 4.4.1.1017
firmware-version: 4.4.2.1011 rx0 tx0
```

The adapter must be using the *full-feature* firmware variant which can be selected using the `sfboot` utility and confirmed with `rx0 tx0` appearing after the version number in the output from `ethtool` as shown above.

The firmware update utility (`sfupdate`) and boot ROM configuration tool (`sfboot`) are available in the Solarflare Linux Utilities package (SF-107601-LS issue 28 or later).

## 3.11 NIC Partitioning with SR-IOV

When combining NIC partitioning with SR-IOV, every partition (PF) must be in a separate VLAN. The user is able to create a number of PFs per physical port and associate a number of VFs with each PF. Within this layer2 broadcast domain there is switching between a PF and its associated VFs.



**Figure 11: NIC Partitioning with SR-IOV**

### Configuration

- 1 Use the sfboot utility to set the firmware switch-mode, create PFs, assign unique VLAN ID to each PF and assign a number of VFs for each PF.

In the following example 4 PFs are configured per physical port and 2 VFs per PF:

```
sfboot switch-mode=partitioning-with-sriov pf-count=4 /
pf-vlans=0,100,110,120 vf-count=2

eth10:
Interface-specific boot options are not available. Adapter-wide
options are available via eth4 (00-0F-53-21-00-60).

eth11:
Interface-specific boot options are not available. Adapter-wide
options are available via eth4 (00-0F-53-21-00-60).

eth12:
Interface-specific boot options are not available. Adapter-wide
options are available via eth4 (00-0F-53-21-00-60).

eth13:
Interface-specific boot options are not available. Adapter-wide
```

```

options are available via eth4 (00-0F-53-21-00-60).

eth14:
Interface-specific boot options are not available. Adapter-wide
options are available via eth4 (00-0F-53-21-00-60).

eth15:
Interface-specific boot options are not available. Adapter-wide
options are available via eth4 (00-0F-53-21-00-60).

eth4:
Boot image Option ROM only
Link speed Negotiated automatically
Link-up delay time 5 seconds
Banner delay time 2 seconds
Boot skip delay time 5 seconds
Boot type Disabled
Physical Functions per port 4
MSI-X interrupt limit 32
Number of Virtual Functions 2
VF MSI-X interrupt limit 8
Firmware variant full feature / virtualization
Insecure filters Disabled
MAC spoofing Disabled
VLAN tags 0,100,110,120
Switch mode Partitioning with SRIOV

eth5:
Boot image Option ROM only
Link speed Negotiated automatically
Link-up delay time 5 seconds
Banner delay time 2 seconds
Boot skip delay time 5 seconds
Boot type Disabled
Physical Functions per port 4
MSI-X interrupt limit 32
Number of Virtual Functions 2
VF MSI-X interrupt limit 8
Firmware variant full feature / virtualization
Insecure filters Disabled
MAC spoofing Disabled
VLAN tags 0,100,110,120
Switch mode Partitioning with SRIOV

```

- 2** PF interfaces are visible in the host using the ifconfig command:

```

eth4 Link encap:Ethernet HWaddr 00:0F:53:21:00:60
eth5 Link encap:Ethernet HWaddr 00:0F:53:21:00:61
eth10 Link encap:Ethernet HWaddr 00:0F:53:21:00:64
eth11 Link encap:Ethernet HWaddr 00:0F:53:21:00:65
eth12 Link encap:Ethernet HWaddr 00:0F:53:21:00:66
eth13 Link encap:Ethernet HWaddr 00:0F:53:21:00:63
eth14 Link encap:Ethernet HWaddr 00:0F:53:21:00:62
eth15 Link encap:Ethernet HWaddr 00:0F:53:21:00:67

```

- 3 The output from steps 1 and 2 above identifies a server with 2 physical interfaces (eth4/eth5), 4 PFs per physical port and identifies the following PF-VLAN configuration:

**Table 18: PF-VLAN Configuration**

| Interface | MAC Address       | PF  | VLAN ID |
|-----------|-------------------|-----|---------|
| eth4      | 00:0F:53:21:00:60 | PF0 | 0       |
| eth10     | 00:0F:53:21:00:64 | PF4 | 110     |
| eth12     | 00:0F:53:21:00:66 | PF6 | 120     |
| eth14     | 00:0F:53:21:00:62 | PF2 | 100     |
| eth5      | 00:0F:53:21:00:61 | PF1 | 0       |
| eth11     | 00:0F:53:21:00:65 | PF5 | 110     |
| eth13     | 00:0F:53:21:00:63 | PF3 | 100     |
| eth15     | 00:0F:53:21:00:67 | PF7 | 120     |

- 4 Refer to [SR-IOV Configuration on page 243](#) for procedures to create VMs and VFs.

## VLAN Configuration

When using partitioning with SR-IOV, all PFs must have a unique VLAN tag. A single PF from each physical port can use tag 0 (zero) to receive untagged traffic. VLAN tags are transparently inserted/stripped by the adapter firmware.

## LACP Bonding

LACP Bonding is not currently supported using the NIC Partitioning configuration mode as the LACP partner i.e. the switch will be unaware of the configured partitions.

Users are advised to refer to the sfc driver release notes for current limitations when using the NIC partitioning features.

## 3.12 Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts. RSS is enabled by default.

When RSS is enabled the controller uses multiple receive queues to deliver incoming packets. The receive queue selected for an incoming packet is chosen to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained. Each receive queue has its own dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores, providing a considerable performance advantage over a conventional adapter architecture in which all received packets for a given interface are processed by just one CPU core. RSS can be restricted to only process receive queues on the NUMA node local to the Solarflare adapter. To configure this the driver module option `rss numa local` should be set to 1.

By default the driver enables RSS and configures one RSS Receive queue per CPU core. The number of RSS Receive queues can be controlled via the driver module parameter `rss_cpus`. The following table identifies `rss_cpus` options.

**Table 19: `rss_cpus` Options**

| Option                        | Description                                                                                                                                      | Interrupt Affinity (MSI-X)                                                                             |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <code>&lt;num_cpus&gt;</code> | Indicates the number of RSS queues to create.                                                                                                    | A separate MSI-X interrupt for a receive queue is affinitized to each CPU.                             |
| packages                      | An RSS queue will be created for each multi-core CPU package. The first CPU in the package will be chosen.                                       | A separate MSI-X interrupt for a receive queue, is affinitized to each of the designated package CPUs. |
| cores                         | An RSS queue will be created for each CPU. The first hyperthread instance (If CPU has hyperthreading) will be chosen.<br><br>The default option. | A separate MSI-X interrupt for a receive queue, is affinitized to each of the CPUs.                    |
| hyperthreads                  | An RSS queue will be created for each CPU hyperthread (hyperthreading must be enabled).                                                          | A separate MSI-X interrupt for a receive queue, is affinitized to each of the hyperthreads.            |

Add the following line to `/etc/modprobe.conf` file or add the options line to a user created file under the `/etc/modprobe.d` directory. The file should have a `.conf` extension:

```
options sfc rss_cpus=<option>
```

To set `rss_cpus` equal to the number of CPU cores:

```
options sfc rss_cpus=cores
```

Sometimes, it can be desirable to disable RSS when running single stream applications, since all interface processing may benefit from taking place on a single CPU:

```
options sfc rss_cpus=1
```

The driver must be reloaded to enable option changes:



**NOTE:** The association of RSS receive queues to a CPU is governed by the receive queue's MSI-X interrupt affinity. See [Interrupt Affinity on page 99](#) for more details.

```
rmmod sfc
modprobe sfc
```



**NOTE:** RSS also works for UDP packets. For UDP traffic the Solarflare adapter will select the Receive CPU based on IP source and destination addresses. Solarflare adapters support IPv4 and IPv6 RSS.

## 3.13 Receive Flow Steering (RFS)

RFS will attempt to steer packets to the core where a receiving application is running. This reduces the need to move data between processor caches and can significantly reduce latency and jitter. Modern NUMA systems, in particular, can benefit substantially from RFS where packets are delivered into memory local to the receiving thread.

Unlike RSS which selects a CPU from a CPU affinity mask set by an administrator or user, RFS will store the application's CPU core identifier when the application process calls `recvmsg()` or `sendmsg()`.

- A hash is calculated from a packet's addresses or ports (2-tuple or 4-tuple) and serves as the consistent hash for the flow associated with the packet.
- Each receive queue has an associated list of CPUs to which RFS may enqueue the received packets for processing.
- For each received packet, an index into the CPU list is computed from the flow hash modulo the size of the CPU list.

There are two types of RFS implementation; Soft RFS and Hardware (or Accelerated) RFS.

Soft RFS is a software feature supported since Linux 2.6.35 that attempts to schedule protocol processing of incoming packets on the same processor as the user thread that will consume the packets.

Accelerated RFS requires Linux kernel version 2.6.39 or later, with the Linux sfc driver or Solarflare v3.2 network adapter driver.

RFS can dynamically change the allowed CPUs that can be assigned to a packet or packet stream and this introduces the possibility of out of order packets. To prevent out of order data, two tables are created that hold state information used in the CPU selection.

- **Global\_flow\_table:** Identifies the number of simultaneous flows that are managed by RFS.
- **Per\_queue\_table:** Identifies the number of flows that can be steered to a queue. This holds state as to when a packet was last received.

The tables support the steering of incoming packets from the network adapter to a receive queue affinitized to a CPU where the application is waiting to receive them. The Solarflare accelerated RFS implementation requires configuration through the two tables and the ethtool -K command.

The following sub-sections identify the RFS configuration procedures:

### Kernel Configuration

Before using RFS the kernel must be compiled with the kconfig symbol `CONFIG_RPS` enabled. Accelerated RFS is only available if the kernel is compiled with the kconfig symbol `CONFIG_RFS_ACCEL` enabled.

## Global Flow Count

Configure the number of simultaneous flows that will be managed by RFS. The suggested flow count will depend on the expected number of active connections at any given time and may be less than the number of open connections. The value is rounded up to the nearest power of two.

```
echo 32768 > /proc/sys/net/core/rps_sock_flow_entries
```

## Per Queue Flow Count

For each adapter interface there will exist a ‘queue’ directory containing one ‘rx’ or ‘tx’ subdirectory for each queue associated with the interface. For RFS only the receive queues are relevant.

```
cd /sys/class/net/eth3/queue
```

Within each ‘rx’ subdirectory, the `rps_flow_cnt` file holds the number of entries in the per-queue flow table. If only a single queue is used then `rps_flow_cnt` will be the same as `rps_sock_flow_entries`. When multiple queues are configured the count will be equal to `rps_sock_flow_entries/N` where  $N$  is the number of queues, for example:

`rps_sock_flow_entries = 32768` and there are 16 queues then `rps_flow_cnt` for each queue will be configured as 2048.

```
echo 2048 > /sys/class/net/eth3/queues/rx-0/rps_flow_cnt
echo 2048 > /sys/class/net/eth3/queues/rx-1/rps_flow_cnt
```

## Disable RFS

To turn off RFS using the following command:

```
ethtool -K <devname> ntuple off
```

## 3.14 Solarflare Accelerated RFS (SARFS)

The Solarflare Accelerated RFS feature directs TCP flows to queues processed on the same CPU core as the user process which is consuming the flow. By querying the CPU when a TCP packet is sent, the transmit queue can be selected from the interrupt associated with the correct CPU core. A hardware filter directs the receive flow to the same queue.

SARFS is provided for servers that do not support standard Linux ARFS. For details of Linux ARFS, refer to the previous section. Additional information can be found at the following link:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Performance\\_Tuning\\_Guide/network-acc-rfs.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/network-acc-rfs.html)

Overall SARFS can improve bandwidth, especially for smaller packets and because core assignment is not subject to the semi-random selection of transmit and receive queues, both bandwidth and latency become more consistent.

The SARFS feature is disabled by default and can be enabled using net driver module parameters. Driver module parameters can be specified in a user created file (e.g. sfc.conf) in the /etc/modprobe.d directory:

```
sxps_enabled
sarfs_table_size
sarfs_global_holdoff_ms
sarfs_sample_rate
```

If the kernel supports XPS, this should be enabled when using the SARFS feature. When the kernel does not supports XPS, the sxps\_enabled parameter should be enabled when using SARFS.



**NOTE:** sxps\_enabled is known to work on RHEL version up to and including RHEL6.5, but does not function on RHEL7 due to changes in the interrupt hint policy.

Refer to [Module Parameters on page 109](#) for a description of the SARFS driver module parameters.

## 3.15 Transmit Packet Steering (XPS)

Transmit Packet Steering (XPS) is supported in Linux 2.6.38 and later. XPS is a mechanism for selecting which transmit queue to use when transmitting a packet on a multi-queue device.

XPS is configured on a per transmit queue basis where a bitmap of CPUs identifies the CPUs that may use the queue to transmit.

### Kernel Configuration

Before using XPS the kernel must be compiled with the kconfig symbol CONFIG\_XPS enabled.

### Configure CPU/Hyperthreads

Within in each /sys/class/net/<interface>/queues/tx-N directory there exists an xps\_cpus file which contains a bitmap of CPUs that can use the queue to transmit. In the following example transmit queue 0 can be used by the first two CPUs and transmit queue 1 can be used by the following two CPUs:

```
echo 3 > /sys/class/net/eth3/queues/tx-0/xps_cpus
echo c > /sys/class/net/eth3/queues/tx-0/xps_cpus
```

If hyperthreading is enabled, each hyperthread is identified as a separate CPU, for example if the system has 16 cores but 32 hyperthreads then the transmit queues should be paired with the hyperthreaded cores:

```
echo 30003 > /sys/class/net/eth3/queues/tx-0/xps_cpus
echo c000c > /sys/class/net/eth3/queues/tx-0/xps_cpus
```

### XPS - Example Configuration

#### System Configuration:

- Single Solarflare adapter
- 2 x 8 core processors with hyperthreading enabled to give a total of 32 cores
- rss\_cpus=8
- Only 1 interface on the adapter is configured
- The IRQ Balance service is disabled

#### Identify interrupts for the configured interface:

```
cat /proc/interrupts | grep 'eth3\|CPU'
> cat /proc/irq/132/smp_affinity
00000000,00000000,00000000,00000001
> cat /proc/irq/133/smp_affinity
00000000,00000000,00000000,00000100
> cat /proc/irq/134/smp_affinity
00000000,00000000,00000000,00000002
```

```
[...snip...]
> cat /proc/irq/139/smp_affinity
00000000,00000000,00000000,00000800
```

The output identifies that IRQ-132 is the first queue and is routed to CPU0. IRQ-133 is the second queue routed to CPU8, IRQ-134 to CPU2 and so on.

### Map TX queue to CPU

Hyperthreaded cores are included with the associated physical core:

```
> echo 110011 > /sys/class/net/eth3/queues/tx-0/xps_cpus
> echo 11001100 > /sys/class/net/eth3/queues/tx-1/xps_cpus
> echo 220022 > /sys/class/net/eth3/queues/tx-2/xps_cpus
> echo 22002200 > /sys/class/net/eth3/queues/tx-3/xps_cpus
> echo 440044 > /sys/class/net/eth3/queues/tx-4/xps_cpus
> echo 44004400 > /sys/class/net/eth3/queues/tx-5/xps_cpus
> echo 880088 > /sys/class/net/eth3/queues/tx-6/xps_cpus
> echo 88008800 > /sys/class/net/eth3/queues/tx-7/xps_cpus
```

### Configure Global and Per Queue Tables

- The flow count (number of active connections at any one time) = 32768
- Number of queues = 8 (rss\_cpus)
- So the flow count for each queue will be 32768/8

```
> echo 32768 > /proc/sys/net/core/rps_sock_flow_entries
> echo 4096 > /sys/class/net/eth3/queues/rx-0/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-1/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-2/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-3/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-4/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-5/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-6/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-7/rps_flow_cnt
```

## 3.16 Linux Utilities RPM

The Solarflare Linux Utilities RPM contains:

- A boot ROM utility.  
See [Configuring the Boot Manager with sfboot on page 74](#).
- A flash firmware update utility.  
See [Upgrading Adapter Firmware with sfupdate on page 82](#).
- A firmware feature activation key install utility.  
See [Activation key install with sfkey on page 87](#).

The RPM package, is supplied as 64bit and 32bit binaries compiled to be compatible with GLIBC versions for all supported distributions. The Solarflare utilities RPM file can be downloaded from the following location:

<https://support.solarflare.com/>

- SF-104451-LS is a 32bit binary RPM package.
- SF-107601-LS is a 64bit binary RPM package.

### Uninstall a previous Utilities RPM

The rpm install command will warn if another version of the Utilities RPM is already installed on the system.

- To identify this RPM:

```
rpm -qa | grep sfutils
sfutils-<version>.x86_64
```

- To remove the sfutils RPM:

```
rpm -e sfutils-<version>.x86_64
```

### Install Utilities

**1** Download and copy the zipped binary RPM package to the required directory.

**2** Unzip the package:

```
unzip SF-107601-LS-<version>_Solarflare_Linux_Utils_RPM_64bit.zip
```

**3** Install the binary RPM:

```
rpm -Uvh sfutils-<version>.x86_64.rpm
Preparing... ###### [100%]
1:sfutils ###### [100%]
```

**4** Check that the RPM installed correctly:

```
rpm -q sfutils
sfutils-<version>.x86_64
```

## 3.17 Configuring the Boot Manager with sfboot

- [Sfboot: Command Usage on page 74](#).
- [Sfboot: Command Line Options on page 75](#).
- [Sfboot: Examples on page 80](#).

Sfboot is a command line utility for configuring Solarflare adapter Boot Manager options, including PXE and UEFI booting. Using sfboot is an alternative to using **Ctrl + B** to access the Boot ROM agent during server startup.

See [Solarflare Boot Manager on page 285](#) for more information on the Boot Rom agent.

PXE and UEFI network boot is not supported for Solarflare adapters on IBM System p servers.

### Sfboot: SLES 11 Limitation

Due to limitations in SLES 11 using kernel versions prior to 2.6.27.54 it is necessary to reboot the server after running the sfboot utility.

### Sfboot: Command Usage

The general usage for sfboot is as follows (as root):

```
sfboot [--adapter=eth<N>] [options] [parameters]
```

When the --adapter option is not specified, the sfboot command applies to all adapters present in the target host.

The format for the parameters are:

```
<parameter>=<value>
```

## Sfboot: Command Line Options

[Table 20](#) lists the options for sfboot, [Table 21](#) lists the available global parameters, and [Table 22](#) lists the available per-adapter parameters. Note that command line options are case insensitive and may be abbreviated.



**NOTE:** Abbreviations in scripts should be avoided, since future updates to the application may render abbreviated scripts invalid.

**Table 20: Sfboot Options**

| Option                | Description                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -h, --help            | Displays command line syntax and provides a description of each sfboot option.                                                                                                                                                                                                                                                                                   |
| -V, --version         | Shows detailed version information and exits.                                                                                                                                                                                                                                                                                                                    |
| -v, --verbose         | Shows extended output information for the command entered.                                                                                                                                                                                                                                                                                                       |
| -y, --yes             | Update without prompting.                                                                                                                                                                                                                                                                                                                                        |
| -s, --quiet           | Suppresses all output, except errors; no user interaction. The user should query the completion code to determine the outcome of commands when operating silently.                                                                                                                                                                                               |
| Aliases: --silent     |                                                                                                                                                                                                                                                                                                                                                                  |
| -l, --list            | <p>Lists all available Solarflare adapters. This option shows the ifname and MAC address.</p> <p>Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.</p>                                                                                    |
| -i, --adapter =<ethX> | Performs the action on the identified Solarflare network adapter. The adapter identifier ethX can be the ifname or MAC address, as output by the --list option. If --adapter is not included, the action will apply to all installed Solarflare adapters.                                                                                                        |
| -c, --clear           | <p>Resets all adapter configuration options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in <a href="#">Table 21</a>) are not reset.</p> <p>Note that --clear can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.</p> |
| -r, --repair          | Restore firmware configuration settings to default values. The sfboot option should only be used if a firmware upgrade/downgrade using sfboot has failed.                                                                                                                                                                                                        |

The following global parameters are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

**Table 21: Sfboot Global Parameters**

| Parameter                                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>boot-image=all optionrom uefi disabled</code>                                     | Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the --adapter option has been specified. This is a global option and applies to all ports on the NIC.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>port-mode=refer to <a href="#">Port Modes on page 38</a></code>                   | Configure the port mode to use. This is for SFN7000, SFN8000 and X2 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent.<br><br>For details of port-modes refer to <a href="#">Port Modes on page 38</a><br><br>Changes to this setting with sfboot require a cold reboot to become effective. MAC address assignments may change after altering this setting.                                                                                                                                                                                                                                                                                                                                     |
| <code>firmware-variant=full-feature ultra-low-latency capture-packed-stream auto</code> | Configure the firmware variant to use. This is for SFN7000, SFN8000 and X2 series adapters only: <ul style="list-style-type: none"> <li>• the SFN7002F adapter is factory set to full-feature</li> <li>• all other adapters are factory set to auto.</li> </ul> Default value = auto - means the driver will select a variant that meets its needs: <ul style="list-style-type: none"> <li>• the VMware driver always uses full-feature</li> <li>• otherwise, ultra-low-latency is used.</li> </ul> The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features). It is recommended that Onload customers use the ultra-low-latency variant. This is a global option and applies to all ports on the NIC. |
| <code>insecure-filters=enabled disabled</code>                                          | If enabled bypass filter security on non-privileged functions. This is for SFN7000 and SFN8000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement and should be enabled when using Onload or when using bonded interfaces. This is a global option and applies to all ports on the NIC.                                                                                                                                                                                                                                                                                                                      |

**Table 21: Sfboot Global Parameters**

| Parameter                                                       | Description                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mac-spoofing=default enabled disabled</code>              | If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000, SFN8000 and X2 series adapters only. The default is disabled.<br><br>Changes to this setting with sfboot require a cold reboot to become effective. This is a global option and applies to all ports on the NIC. |
| <code>rx-dc-size=8 16 32 64</code>                              | Specifies the size of the descriptor cache for each receive queue. This is for SFN7000, SFN8000 and X2 series adapters only. The default is: <ul style="list-style-type: none"><li>• 16 if the port-mode supports the maximum number of connectors for the adapter</li><li>• 32 if the port-mode supports a reduced number of connectors.</li></ul>        |
| <code>change-mac-default enabled disabled</code>                | This is for SFN7000, SFN8000 and X2 series adapters only. Change the unicast MAC address for a non-privileged function on this port. This is a global option and applies to all physical ports on the NIC.                                                                                                                                                 |
| <code>tx-dc-size=8 16 32 64</code>                              | Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000, SFN8000 and X2 series adapters only. The default is: <ul style="list-style-type: none"><li>• 32 if the port-mode supports the maximum number of connectors for the adapter</li><li>• 64 if the port-mode supports a reduced number of connectors.</li></ul>       |
| <code>vi-count=&lt;vi count&gt;</code>                          | Sets the total number of virtual interfaces that will be available on the NIC.                                                                                                                                                                                                                                                                             |
| <code>event-merge-timeout=&lt;timeout in nanoseconds&gt;</code> | Specifies the timeout in nanoseconds for RX event merging. A timeout of 0 means that event merging is disabled.                                                                                                                                                                                                                                            |

The following per-adapter parameters are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

**Table 22: Sfboot Per-adapter Parameters**

| Parameter                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| link-speed=auto 10g 1g 100m            | <p>Specifies the network link speed of the adapter used by the Boot ROM. The default is auto. On the 10GBASE-T adapters, auto instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, auto instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link.</p> <p>auto Auto-negotiate link speed (default)</p> <p>10G 10G bit/sec</p> <p>1G 1G bit/sec</p> <p>100M 100M bit/sec</p> |
| linkup-delay=<delay time in seconds>   | <p>Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds.</p>                                                                                                                                                                                                                                                                                                      |
| banner-delay=<delay time in seconds>   | <p>Specifies the wait period for Ctrl-B to be pressed to enter adapter configuration tool.</p> <p>&lt;delay time in seconds&gt; = 0-256</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| bootskip-delay=<delay time in seconds> | <p>Specifies the time allowed for Esc to be pressed to skip adapter booting.</p> <p>&lt;delay time in seconds&gt; = 0-256</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| boot-type=pxe disabled                 | <p>Sets the adapter boot type – effective on next boot.</p> <p>pxe – PXE (Preboot eXecution Environment) booting</p> <p>disabled – Disable adapter booting</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**Table 22: Sfboot Per-adapter Parameters**

| Parameter                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>pf-count=&lt;pf count&gt;</code>              | <p>This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter.</p> <p>Changes to this setting with sfboot require a cold reboot to become effective. MAC address assignments may change after altering this setting.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>msix-limit=8 16 32 64 128 256 512 1024</code> | <p>Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32.</p> <p>Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>vf-count=&lt;vf count&gt;</code>              | <p>The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port.</p> <p>Adapters support 2048 interrupts</p> <p>Adapters support a total limit of 127 virtual functions per port.</p> <p>Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured.</p> <p>Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count.</p> <p>The sriov parameter is implied if vf-count is greater than zero.</p> <p>Changes to this setting with sfboot require a cold reboot to become effective.</p> |
| <code>vf-msix-limit=1 2 4 8 16 32 64 128 256</code> | <p>The maximum number of interrupts a virtual function may use.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

**Table 22: Sfboot Per-adapter Parameters**

| Parameter                                                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>pf-vlans=&lt;tag&gt;[,&lt;tag&gt;[,...]] none</code>                        | Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.<br><br>Setting pf-vlans=none will clear all VLAN tags on the port. pf-vlans should be included after the pf-count option on the sfboot command line.<br><br>If the number of PFs is changed then the VLAN tags will be cleared.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>switch-mode=default sriov partitioning partitioning-with-sriov pfiov</code> | Specifies the mode of operation that the port will be used in:<br><br><code>default</code> - single PF created, zero VFs created.<br><br><code>sriov</code> - SR-IOV enabled, single PF created, VFs configured with vf-count.<br><br><code>partitioning</code> - PFs configured with pf-count, VFs configured with vf-count. See <a href="#">NIC Partitioning on page 59</a> for details.<br><br><code>partitioning-with-sriov</code> - SR-IOV enabled, PFs configured with pf-count, VFs configured with vf-count. See <a href="#">NIC Partitioning on page 59</a> for details.<br><br><code>pfiov</code> - PFIOV enabled, PFs configured with pf-count, VFs not supported.<br><br>Changes to this setting with sfboot require a cold reboot to become effective. |

## Sfboot: Examples

- Show the current boot configuration for all adapters:
- ```
sfboot
```

```
# ./sfboot
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth4:
  Boot image          Option ROM only
  Link speed         Negotiated automatically
  Link-up delay time 5 seconds
  Banner delay time 2 seconds
  Boot skip delay time 5 seconds
  Boot type          Disabled
  Physical Functions per port 1
  MSI-X interrupt limit 32
  Number of Virtual Functions 0
  VF MSI-X interrupt limit 8
```

Firmware variant	full feature / virtualization
Insecure filters	Disabled
VLAN tags	None
Switch mode	Default

- List all Solarflare adapters installed on the localhost:

```
sfboot --list
```

```
./sfboot -l
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
Adapter list:
eth4
eth5
```

- Enable Firmware Variant

```
sfboot firmware-variant=full-feature
```

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
```

eth4:	
Boot image	Option ROM only
Link speed	Negotiated automatically
Link-up delay time	7 seconds
Banner delay time	3 seconds
Boot skip delay time	6 seconds
Boot type	PXE
MSI-X interrupt limit	32
Number of Virtual Functions	0
VF MSI-X interrupt limit	1
Firmware variant	full feature / virtualization

- SR-IOV enabled and using Virtual Functions

```
sfboot switch-mode=sriov vf-count=4
```

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
```

eth4:	
Boot image	Option ROM only
Link speed	Negotiated automatically
Link-up delay time	5 seconds
Banner delay time	2 seconds
Boot skip delay time	5 seconds
Boot type	Disabled
Physical Functions per port	1
MSI-X interrupt limit	32
Number of Virtual Functions	4
VF MSI-X interrupt limit	8
Firmware variant	full feature / virtualization
Insecure filters	Disabled
VLAN tags	None
Switch mode	SRIOV

3.18 Upgrading Adapter Firmware with sfupdate

- [Sfupdate: Command Usage on page 82.](#)
- [Sfupdate: Command Line Options on page 85.](#)
- [Sfupdate: Examples on page 86.](#)

Sfupdate is a command line utility to manage and upgrade the Solarflare adapter Boot ROM, Phy and adapter firmware. Embedded within the sfupdate executable are firmware images for the Solarflare adapter - the exact updates available via sfupdate depend on the specific adapter type.

See [Solarflare Boot Manager on page 285](#) for more information on the Boot Rom agent.



CAUTION: All Applications accelerated with OpenOnload should be terminated before updating the firmware with sfupdate.



CAUTION: Solarflare PTP (sfptpd) should be terminated before updating firmware.

Sfupdate: Command Usage

The general usage for sfupdate is as follows (as root):

```
# sfupdate [--adapter=eth<N>] [options]
```

where:

- ethN is the interface name (ifname) of the Solarflare adapter to be upgraded.
- option is one of the command options listed in [Table 23](#).

The format for the options are:

```
<option>=<parameter>
```

Running the command sfupdate with no additional parameters will show the current firmware version for all Solarflare adapters and identifies whether the firmware version within sfupdate is more up to date. To update the firmware for all Solarflare adapters run the command sfupdate --write

Solarflare recommend the following procedure:

- 1 Run sfupdate to check that the firmware on all adapters is up to date.
- 2 Run sfupdate --write to update the firmware on all adapters.

```
# sfupdate --adapter=<interface> --write [--backup|--force]
```

Sfupdate: Linux MTD Limitations

The driver supplied “inbox” within RedHat and Novell distributions has a limitation on the number of adapters that sfupdate can support. This limitation is removed from RHEL 6.5 onwards. The Solarflare supplied driver is no longer subject to this limitation on any distro/kernel.

Linux kernel versions prior to 2.6.20 support up to 16 MTD (flash) devices. Solarflare adapters are equipped with 6 flash partitions. If more than two adapters are deployed within a system a number of flash partitions will be inaccessible during upgrade.

The limit was raised to 32 in Linux kernel version 2.6.20 and removed altogether in 2.6.35.

If issues are encountered during sfupdate, the user should consider one of the following options when upgrading firmware on systems equipped with more than two Solarflare adapters:

- Upgrade two adapters at a time with the other adapters removed.
- Upgrade the kernel.
- Rebuild the kernel, raising the value of MAX_MTD_DEVICES in include/linux/mtd/mtd.h.
- Download an *Sfutils bootable image* from:
https://support.solarflare.com/index.php?id=1960&option=com_cognidox

Overcome Linux MTD Limitations

An alternative method is available to upgrade the firmware without removing the adapters.

- 1 Unbind all interfaces from the drivers:

```
# for bdf in $(lspci -D -d 1924: | awk '{ print $1 }'); do \
    echo -n ${bdf}\ > /sys/bus/pci/devices/${bdf}/driver/unbind; done
```

- 2 Identify the bus/device/function for all Solarflare interfaces.

Using ifconfig -a will not discover any Solarflare interfaces. Use lspci:

```
# lspci -D -d 1924:
```

Output similar to the following will be produced (5 NICs installed in this example):

```
# lspci -D -d 1924:  
0000:02:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:02:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:03:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:03:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:04:00.0 Ethernet controller: Solarflare Communications SFL9021 [Solarstorm]  
0000:04:00.1 Ethernet controller: Solarflare Communications SFL9021 [Solarstorm]  
0000:83:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:83:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:84:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]  
0000:84:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
```

- 3 There are enough resources to upgrade two NICs at a time, so re-bind interfaces in groups of four (2x2NICs):

```
# echo -n "0000:02:00.0" > /sys/bus/pci/drivers/sfc/bind  
# echo -n "0000:02:00.1" > /sys/bus/pci/drivers/sfc/bind  
# echo -n "0000:03:00.0" > /sys/bus/pci/drivers/sfc/bind  
# echo -n "0000:03:00.1" > /sys/bus/pci/drivers/sfc/bind
```

- 4 Run sfupdate to update these NICs (command options may vary):

```
# sfupdate --write --yes --force
```

- 5 Run the command to unbind the interfaces again. There will be failures reported because some of the interfaces are not bound:

```
# for bdf in $(lspci -D -d 1924: | awk '{ print $1 }'); do \  
echo -n ${bdf}\ > /sys/bus/pci/devices/${bdf}/driver/unbind; done
```

- 6 Repeat the process for the other interfaces (0000:04:00.x; 0000:83:00.x and 0000:84:00.x) doing so in pairs until all the NICs have been upgraded.

- 7 Rebind all interfaces, doing so en-mass and ignoring errors from those already bound:

```
# for bdf in $(lspci -D -d 1924: | awk '{ print $1 }'); do \  
echo -n ${bdf}\ > /sys/bus/pci/drivers/sfc/bind; done
```

Alternatively reload the sfc driver:

```
# onload_tool reload
```

or:

```
# modprobe -r sfc  
# modprobe sfc
```

- 8 Run ifconfig -a again to find that all the interfaces are reported and all have been firmware upgraded without having to physically touch the server or change the kernel.

Sfupdate: SLES 11 Limitation

Due to limitations in SLES 11 using kernel versions prior to 2.6.27.54 it is necessary to reboot the server after running the sfupdate utility to upgrade server firmware.

Sfupdate: Command Line Options

Table 23 lists the options for sfupdate.

Table 23: Sfupdate Options

Option	Description
-h, --help	Shows help for the available options and command line syntax.
-i, --adapter=ethX	Specifies the target adapter when more than one adapter is installed in the localhost. ethX = Adapter ifname or MAC address (as obtained with --list).
--list	Shows the adapter ID, adapter name and MAC address of each adapter installed in the localhost.
--write	Re-writes the firmware from the images embedded in the sfupdate tool. To re-write using an external image, specify --image=<filename> in the command. --write fails if the embedded image is the same or a previous version. To force a write in this case, specify --force in the command.
--force	Force the update of all firmware, even if the installed firmware version is the same as, or more recent than, the firmware embedded in sfupdate.
--backup	Backup existing firmware image before updating. This option may be used with --write and --force.
--image=(filename)	Update the firmware using the binary image from the given file rather than from those embedded in the utility.
--ipxe-image=(filename)	Install an iPXE image from the given file, replacing the Solarflare boot ROM image. sfupdate will not automatically replace the iPXE image in subsequent flash updates unless the --restore-bootrom option is used.
--restore-bootrom	Replace an iPXE image in flash with the standard Solarflare Boot Manager PXE image included in sfupdate.
-y, --yes	Update without prompting. This option can be used with the --write and --force options.
-v, --verbose	Verbose mode.

Table 23: Sfupdate Options

Option	Description
<code>-s, --silent</code>	Suppress output while the utility is running; useful when the utility is used in a script.
<code>-V, --version</code>	Display version information and exit.

Sfupdate: Examples

- Display firmware versions for all adapters:

```
sfupdate
```

```
Solarstorm firmware update utility [v4.3.1]
Copyright Solarflare Communications 2006-2013, Level 5 Networks 2002-2005
```

```
eth4 - MAC: 00-0F-53-21-00-61
    Controller type: Solarflare SFC9100-family
    Controller versoin: unknown
    Boot ROM version: unknown
```

```
This utility contains more recent Boot ROM firmware [v4.2.1.1000]
    - run "sfupdate --write" to perform an update
```

```
This utility contains more recent controller firmware [v4.2.1.1010]
    - run "sfupdate --write" to perform an update
```

```
eth5 - MAC: 00-0F-53-21-00-60
    Controller type: Solarflare SFC9100-family
    Controller version: unknown
    Boot ROM version: unknown
```

```
This utility contains more recent Boot ROM firmware [v4.2.1.1000]
    - run "sfupdate --write" to perform an update
```

```
This utility contains more recent controller firmware [v4.2.1.1010]
    - run "sfupdate --write" to perform an update
```

- Update adapter firmware:

```
# sfupdate --adapter=<interface> --write
```

- Update adapter firmware + create a backup firmware image:

```
# sfupdate --adapter=<interface> --write --backup
```

- Update firmware to an earlier or the same version:

```
# sfupdate --adapter=<interface> --write --force
```

A backup firmware image file can be restored to the adapter using the '--image' option.

3.19 Activation key install with sfkey

The sfkey utility is distributed with the Linux Utilities RPM package. This utility is used to install Solarflare AppFlex™ activation keys and enable selected on-board services for Solarflare adapters. For more information about activation key requirements see [Solarflare AppFlex™ Technology. on page 15](#).

sfkey: Command Usage

```
# sfkey [--adapter=eth<N>] [options]
```

If the adapter option is not specified, operations will be applied to all installed adapters.

- To view all sfkey options:

```
# sfkey --help
```

- To list (by key ID) all adapters that support activation keys:

```
# sfkey --inventory --all
```

```
eth2: 714100101282140148200014
```

- To display an adapter's activation keys:

```
# sfkey --adapter=eth2 --report
```

```
eth2: 714100101282140148200014 (Flareon)
```

```
Product name          Solarflare SFN7141Q QSFP+ Flareon Ultra Server Adapter
Installed keys    Onload
```

- To install a activation key:

Copy the activation key to a .txt file on the target server. All keys can be in the same key file and the file applied on multiple servers. The following example uses an activation key file called keys.txt created on the local server.

```
# sfkey --adapter=eth2 --install keys.txt
```

```
Reading keys...
```

```
Writing all keys to eth2...
```

```
eth2: 714100101282140148200014 (Flareon)
```

```
Product name          Solarflare SFN7141Q QSFP+ Flareon Ultra Server Adapter
Installed keys    Onload, SolarCapture Pro, Capture SolarSystem
```

Activation Keys Inventory

Use the combined --inventory and --keys options to identify the activation keys installed on an adapter.

```
# sfkey --adapter=eth2 --inventory --keys
```

```
eth2: 714100101282140148200014 (Flareon), $ONL, !PTP, !SCL, SCP, CSS, !SSFE, !PM, !NAC
```

Activation key information is displayed in [Prefix] [AppID] [Suffix] format.

Prefix:	<none>	Feature is active
	\$	Factory-fitted
	!	Not present
AppID:	An	Application ID number
	<name>	Application acronym
Suffix:	<none>	Feature is active
	+	Site activated
	~	Evaluation key
	*	Inactive key
	@	Inactive site key
	-	No state available

sfkey Options

[Table 24](#) describes all sfkey options.

Table 24: sfkey options

Option	Description
--backup	Output a report of the installed keys in all adapters. The report can be saved to file and later used with the --install option.
--install <filename>	Install activation keys from the given file and report the result. To read from stdin use “-” in place of filename. Keys are installed to an adapter, so if an adapter’s ports are eth4 and eth5, both ports will be affected by the keys installed. <i>sfc driver reload is required after sfkey installs certain feature (e.g. a PTP key).</i> To reload the sfc driver: <pre># modprobe -r sfc; modprobe sfc</pre> or when Onload is installed: <pre># onload_tool reload</pre>
--inventory	List the adapters that support activation keys. To list all adapters use the --all option. To list keys use the --keys option.

Table 24: sfkey options

Option	Description
<code>--keys</code>	Include keys in --inventory output - see Inventory above.
<code>--noevaluationupdate</code>	Do not update any evaluation keys.
<code>-a, --all</code>	Apply sfkey operation to all adapters that support licensing.
<code>-c, --clear</code>	Delete all existing activation keys from an adapter - except factory installed keys.
<code>-h, --help</code>	Display all sfkey options.
<code>-i, --adapter</code>	identify specific adapter to apply sfkey operation to.
<code>-r, --report</code>	Display an adapter serial number and current activation key status (see example above). Use with --all or with --adapter. If an installed or active key is reported as 'An' (where n is a number), it indicates a key unknown to this version of sfkey - use an updated sfkey version.
<code>-s, --silent</code>	Silent mode, output errors only.
<code>-v, --verbose</code>	Verbose mode.
<code>-V, --version</code>	Display sfkey version and exit.
<code>-x, --xml</code>	Report formated as XML.

3.20 Performance Tuning on Linux

- [Introduction on page 90](#)
- [Tuning settings on page 90](#)
- [Other Considerations on page 102](#)

Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to give good performance across a broad class of applications. Occasionally, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this section is to highlight adapter driver settings that affect the performance metrics described. This section covers the tuning of all Solarflare adapters.

Latency will be affected by the type of physical medium used: 10GBase-T, twinaxial (direct-attach), fiber or KX4. This is because the physical media interface chip (PHY) used on the adapter can introduce additional latency. Likewise, latency can also be affected by the type of SFP/SFP+/QSFP module fitted.

In addition, you may need to consider other issues influencing performance, such as application settings, server motherboard chipset, CPU speed, cache size, RAM size, additional software installed on the system, such as a firewall, and the specification and configuration of the LAN. Consideration of such issues is not within the scope of this guide.

Tuning settings

Port mode

The selected port mode for SFN7000, SFN8000 and X2 series adapters should correspond to the speed and number of connectors in use, after using any splitter cables. If a restricted set of connectors is configured, the driver can then transfer resources from the unused connectors to those configured, potentially improving performance.

Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved, because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support an MTU of up to 9216 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

The MTU is changed dynamically using ifconfig, where ethX is the interface name and <size> is the MTU size in bytes:

```
# /sbin/ifconfig <ethX> mtu <size>
```

Verification of the MTU setting may be performed by running ifconfig with no options and checking the MTU value associated with the interface. The change in MTU size can be made to persist across reboots by editing the file /etc/sysconfig/network-scripts/ifcfg-ethX and adding MTU=<mtu> on a new line.

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation reduces the number of interrupts generated by the adapter by coalescing multiple received packet events and/or transmit completion events together into a single interrupt.

The *interrupt moderation interval* sets the minimum time (in microseconds) between two consecutive interrupts. Coalescing occurs only during this interval:

- When the driver generates an interrupt, it starts timing the moderation interval.
- Any events that occur before the moderation interval expires are coalesced together into a single interrupt, that is raised only when the interval expires. A new moderation interval then starts, during which no interrupt is raised.
- An event that occurs after the moderation interval has expired gets its own dedicated interrupt, that is raised immediately. A new moderation interval then starts, during which no interrupt is raised.

Solarflare adapters, by default, use an *adaptive algorithm* where the interrupt moderation delay is automatically adjusted between zero (no interrupt moderation) and 60 microseconds. The adaptive algorithm detects latency sensitive traffic patterns and adjusts the interrupt moderation interval accordingly.

Interrupt moderation settings are **critical for tuning adapter latency**:

- Disabling the adaptive algorithm will:
 - reduce jitter

- allow setting the moderation interval as required to suit conditions.
- Increasing the interrupt moderation interval will:
 - generate less interrupts
 - reduce CPU utilization (because there are less interrupts to process)
 - increase latency
 - improve peak throughput.
- Decreasing the interrupt moderation interval will:
 - generate more interrupts
 - increase CPU utilization (because there are more interrupts to process)
 - decrease latency
 - reduce peak throughput.
- Turning off interrupt moderation will:
 - generate the most interrupts
 - give the highest CPU utilization
 - give the lowest latency
 - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits typically outweigh the cost of increased CPU utilization. It is recommended that:

- Interrupt moderation is disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation is enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.

Interrupt moderation can be changed using ethtool, where ethX is the interface name. Before adjusting the interrupt moderation interval, it is recommended to disable adaptive moderation:

```
ethtool -C <ethX> adaptive-rx off
```

To set the RX interrupt moderation interval in microseconds (μ s):

```
ethtool -C <ethX> rx-usecs <interval>
```

To turn off interrupt moderation, set an interval of zero (0):

```
ethtool -C <ethX> rx-usecs 0
```

The above example also sets the transmit interrupt moderation interval, unless the driver module parameter `separate_tx_channels` is enabled. (Normally packet RX and TX completions will share interrupts, so RX and TX interrupt moderation intervals must be equal, and the adapter driver automatically adjusts tx-usecs to match rx-usecs.) Refer to [Table 29 on page 109](#).

To set the TX interrupt moderation interval, if `separate_tx_channels` is enabled:

```
ethtool -C <ethX> tx-usecs <interval>
```

Interrupt moderation settings can be checked using `ethtool -c`.



NOTE: The performance benefits of TCP Large Receive Offload are limited if interrupt moderation is disabled. See [TCP Large Receive Offload \(LRO\) on page 93](#).

TCP/IP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver has all checksum offload features enabled by default. Therefore, there is no opportunity to improve performance from the default.

Checksum offload is controlled using `ethtool`:

- Receive Checksum:

```
# /sbin/ethtool -K <ethX> rx <on|off>
```

- Transmit Checksum:

```
# /sbin/ethtool -K <ethX> tx <on|off>
```

Verification of the checksum settings may be performed by running `ethtool` with the `-k` option.



NOTE: Solarflare recommend you do not disable checksum offload.

TCP Segmentation Offload (TSO)

TCP Segmentation Offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TSO benefits applications using TCP. Applications using protocols other than TCP will not be affected by TSO.

Enabling TSO will reduce CPU utilization on the transmit side of a TCP connection and improve peak throughput, if the CPU is fully utilized. Since TSO has no effect on latency, it can be enabled at all times. The driver has TSO enabled by default. Therefore, there is no opportunity to improve performance from the default.

TSO is controlled using `ethtool`:

```
# /sbin/ethtool -K <ethX> tso <on|off>
```

Verification of the TSO settings may be performed by running `ethtool` with the `-k` option.

TCP and IP checksum offloads must be enabled for TSO to work.



NOTE: Solarflare recommend that you do not disable this setting.

TCP Large Receive Offload (LRO)

TCP Large Receive Offload (LRO) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single larger packet before passing this onto the network stack for receive processing. This reduces CPU utilization and improves peak throughput when the CPU is fully utilized. The effectiveness of LRO

is bounded by the interrupt moderation delay, and is limited if interrupt moderation is disabled (see [Interrupt Moderation \(Interrupt Coalescing\) on page 91](#)). Enabling LRO does not itself negatively impact latency.



NOTE: The Solarflare network adapter driver enables LRO by default. By its design, LRO is of greater benefit when working with smaller packets. For Solarflare adapter, LRO will become disabled if the MTU is set larger than 3979. When the MTU is set larger than 3978, LRO cannot be enabled and will be reported as 'fixed disabled' by ethtool.



NOTE: LRO should **NOT** be enabled when using the host to forward packets from one interface to another. For example, if the host is performing IP routing.



NOTE: It has been observed that as RHEL6 boots the libvirtd daemon changes the default forwarding setting such that LRO is disabled on all network interfaces. This behavior is undesirable as it will potentially lower bandwidth and increase CPU utilization - especially for high bandwidth streaming applications.

To determine if LRO is enabled on an interface:

```
ethtool -k ethX
```

If IP forwarding is not required on the server, Solarflare recommends either:

- Disabling the libvirtd service (if this is not being used),
- Or, as root before loading the Solarflare driver:

```
sysctl -w net.ipv4.conf.default.forwarding=0
```

(This command can be loaded into /etc/rc.local),
- Or, after loading the Solarflare driver, turn off forwarding for only the Solarflare interfaces and re-enable LRO:

```
sysctl -w net.ipv4.conf.ethX.forwarding=0
```



```
ethtool -K ethX lro on
```

(where X is the id of the Solarflare interface).

Disabling the libvirtd service is a permanent solution, whereas the other recommendations are temporary and will not persist over reboot.

LRO should not be enabled if IP forwarding is being used on the same interface as this could result in incorrect IP and TCP operation.

LRO can be controlled using the module parameter `lro`. Add the following line to `/etc/modprobe.conf` or add the options line to a file under the `/etc/modprobe.d` directory to disable LRO:

```
options sfc lro=0
```

Then reload the driver so it picks up this option:

```
rmmod sfc  
modprobe sfc
```

The current value of this parameter can be found by running:

```
cat /sys/module/sfc/parameters/lro
```

LRO can also be controlled on a per-adapter basis by writing to this file in sysfs:

```
/sys/class/net/ethX/device/lro
```

- To disable LRO:
`echo 0 > /sys/class/net/ethX/device/lro`
- To enable LRO:
`echo 1 > /sys/class/net/ethX/device/lro`
- To show the current value of the per-adapter LRO state:
`cat /sys/class/net/ethX/device/lro`

Modifying this file instantly enables or disables LRO, no reboot or driver reload is required. This setting takes precedence over the `lro` module parameter

Current LRO settings can be identified with Linux ethtool e.g.

```
ethtool -k ethX
```

TCP and IP checksum offloads must be enabled for LRO to work.

TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

For Linux kernel versions, including 2.6.16 and later, initial buffering settings should provide good performance. However for earlier kernel versions, and for certain applications even on later kernels, tuning buffer settings can significantly benefit throughput. To change buffer settings, adjust the `tcp_rmem` and `tcp_wmem` using the `sysctl` command:

- Receive buffering:
`sysctl net.ipv4.tcp_rmem=<min> <default> <max>"`
- Transmit buffering:
`sysctl net.ipv4.tcp_wmem=<min> <default> <max>"`

(`tcp_rmem` and `tcp_wmem` can also be adjusted for IPV6 and globally with the `net.ipv6` and `net.core` variable prefixes respectively).

Typically it is sufficient to tune just the max buffer value. It defines the largest size the buffer can grow to. Suggested alternate values are `max=500000` (1/2 Mbyte). Factors such as link latency, packet loss and CPU cache size all influence the affect of the max buffer size values. The minimum and default values can be left at their defaults `minimum=4096` and `default=87380`.

Buffer Allocation Method

The Solarflare driver has a single optimized buffer allocation strategy. This replaces the two different methods controlled with the `rx_alloc_method` driver module parameter which were available using 3.3 and previous drivers.

The net driver continues to expose the `rx_alloc_method` module option, but the value is ignored and it only exists to not break existing customer configurations.

TX PIO

PIO (programmed input/output) describes the process where data is directly transferred by the CPU to or from an I/O device. It is an alternative technique to the I/O device using bus master DMA to transfer data without CPU involvement.

Solarflare SFN7000, SFN8000 and X2 series adapters support TX PIO, where packets on the transmit path can be “pushed” to the adapter directly by the CPU. This improves the latency of transmitted packets but can cause a very small increase in CPU utilization. TX PIO is therefore especially useful for smaller packets.

The TX PIO feature is enabled by default for packets up to 256 bytes. The maximum packet size that can use PIO can be configured with the driver module option `piobuf_size`.

CTPIO

Supported on the XtremeScale X2 series adapters, cut-through PIO delivers the lowest transmit latency when packets are transmitted on the wire while still being streamed over the PCI interface from the host.

For further details, refer to the Onload User Guide (SF-104474-CD).

3.21 Web Server - Driver Optimization

Introduction

The Solarflare net driver from version 4.4.1.1017 includes optimizations aimed specifically at web service providers and cloud based applications.

Tuning recommendations are documented in [Table 25](#) for users concerned with Content Delivery Networks (CDN), HTTP web hosting application technologies such as HA Proxy, nginx and HTTP web servers.

Performance improvements have been observed in the following areas:

- increased the rate at which servers can process new HTTP connections
- increased the rate at which servers can process HTTP requests
- increased sustained throughput when processing large files via HTTP
- improved kernel throughput performance

Customers requiring further details or to access test data should send an email to support@solarflare.com.

Driver Tuning

Whilst most driver enhancements are internal changes, transparent and non-configurable by the user, the following driver module options can be used to tune the driver for particular user applications.

- rss numa local

Using the 4.4.1.1017 driver this option is enabled by default. This will restrict RSS to use CPU cores only on the NUMA node closest to the adapter. This is particularly important for processors supporting DDIO.

RSS channels not on the local NUMA node can still be accessed using the ethtool -U commands to identify a core (action) on which to process the specified ethtool ntuple filter traffic. For example if rss_cpus=cores, then an RSS receive channel and associated MSI-X interrupt is created for every core.

- rx_recycle_ring_size

The default value for the maximum number of receive buffers to recycle pages for has been changed to 512, and in newer drivers will be further increased to 1024.

- rx_copybreak

A default value of 192 bytes has been selected as the maximum size of packet (bytes) that will be copied directly to the network stack.

Driver module options can be enabled in a user-created file (e.g sfc.conf) in the /etc/modprobe.d directory, for example:

```
options sfc rss numa local=Y  
options sfc rx_recycle_ring_size=512
```

For further descriptions and to list all sfc driver module options:

```
# modinfo sfc
```

nginx Tuning

Table 25: nginx Server Tuning

Tuning	Notes
SO_REUSEPORT	Solarflare testing involving nginx used version v1.7.9 with applied patch to support so_reuseport. See the following link for details: http://forum.nginx.org/read.php?29,241283,241283 .
rss_cpus=N	Create N receive queues where N=(number of logical cores)/2. See Receive Side Scaling (RSS) on page 66 for options.
rss numa local=1	On SMP systems it is recommended to have all interrupts on the NUMA node local to the Solarflare adapter: rss numa-local=1, and pin nginx threads to the free CPUs even when these are on the non-local node. When this is not possible, CPU cores can be divided equally between interrupts and nginx threads. rss numa_local=1 is the default setting.
Pinning threads	Application threads and interrupts should not be pinned to the same CPU cores.
ethtool -C adaptive-rx off	Disable the irq-balance service to prevent re-distribution of interrupts by the kernel. Disable adaptive interrupt moderation before setting the interrupt moderation interval.
ethtool -C rx-usecs 60	Set the interrupt moderation interval. When processing smaller packets it is generally better to set a higher interval i.e. 60µsecs and for larger packets a lower interval or even zero to disable interrupt moderation. See Interrupt Moderation (Interrupt Coalescing) on page 91 .

Adapters - Software Support

To benefit from recent driver optimizations, the following (minimum) net driver and firmware versions should be used:

```
# ethtool -i eth<N>
driver: sfc
version: 4.4.1.1017
firmware-version: 4.4.2.1011 rx1 tx1
```

For latency sensitive applications, the adapter firmware variant should be set with the sfboot utility to ultra-low-latency:

```
# sfboot --adapter=eth<N> firmware-variant=ultra-low-latency
```

The ultra-low-latency firmware variant is being used when the output from ethtool (above) shows the *rx1* and *tx1* values.

A reboot of the server is required after changes using sfboot.

3.22 Interrupt Affinity

Interrupt affinity describes the set of host CPUs that may service a particular interrupt.

This affinity therefore dictates the CPU context where received packets will be processed and where transmit packets will be freed once sent. If the application can process the received packets in the same CPU context by being affinitized to the relevant CPU, then latency and CPU utilization can be improved. This improvement is achieved because well tuned affinities reduce inter-CPU communication.

Tuning interrupt affinity is most relevant when MSI-X interrupts and RSS are being used. The irqbalance service, which typically runs by default in most Linux distributions, is a service that automatically changes interrupt affinities based on CPU workload.

In many cases the irqbalance service hinders rather than enhances network performance. It is therefore necessary to disable it and then set interrupt affinities.

- To disable irqbalance permanently, run:
`/sbin/chkconfig -level 12345 irqbalance off`
- To see whether irqbalance is currently running, run:
`/sbin/service irqbalance status`
- To disable irqbalance temporarily, run:
`/sbin/service irqbalance stop`

Once the irqbalance service has been stopped, the Interrupt affinities can be configured manually.



NOTE: The Solarflare driver will evenly distribute interrupts across the available host CPUs (based on the `rss_cpus` module parameter).

To use the Solarflare driver default affinities (recommended), the irqbalance service must be disabled before the Solarflare driver is loaded (otherwise it will immediately overwrite the affinity configuration values set by the Solarflare driver).

Example 1:

How affinities should be manually set will depend on the application. For a single streamed application such as Netperf, one recommendation would be to affinize all the Rx queues and the application on the same CPU. This can be achieved with the following steps:

- 1 Determine which interrupt line numbers the network interface uses. Assuming the interface is eth0, this can be done with:

```
# cat /proc/interrupts | grep eth0-
123:      13302      0      0      0      PCI-MSI-X  eth0-0
131:        0      24      0      0      PCI-MSI-X  eth0-1
139:        0      0      32      0      PCI-MSI-X  eth0-2
147:        0      0      0     21      PCI-MSI-X  eth0-3
```

This output shows that there are four channels (rows) set up between four CPUs (columns).

- 2 Determine the CPUs to which these interrupts are assigned to:

```
# cat /proc/irq/123/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000001
# cat /proc/irq/131/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000002
# cat /proc/irq/139/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000004
# cat /proc/irq/147/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000008
```

This shows that RXQ[0] is affinized to CPU[0], RXQ[1] is affinized to CPU[1], and so on. With this configuration, the latency and CPU utilization for a particular TCP flow will be Dependant on that flow's RSS hash, and which CPU that hash resolves onto.



NOTE: Interrupt line numbers and their initial CPU affinity are not guaranteed to be the same across reboots and driver reloads. Typically, it is therefore necessary to write a script to query these values and apply the affinity accordingly.

- 3 Set all network interface interrupts to a single CPU (in this case CPU[0]):

```
# echo 1 > /proc/irq/123/smp_affinity
# echo 1 > /proc/irq/131/smp_affinity
# echo 1 > /proc/irq/139/smp_affinity
# echo 1 > /proc/irq/147/smp_affinity
```



NOTE: The read-back of /proc/irq/N/smp_affinity will return the old value until a new interrupt arrives.

- 4 Set the application to run on the same CPU (in this case CPU[0]) as the network interface's interrupts:

```
# taskset 1 netperf
# taskset 1 netperf -H <host>
```



NOTE: The use of taskset is typically only suitable for affinity tuning single threaded, single traffic flow applications. For a multi threaded application, whose threads for example process a subset of receive traffic, taskset is not suitable. In such applications, it is desirable to use RSS and Interrupt affinity to spread receive traffic over more than one CPU and then have each receive thread bind to each of the respective CPUs. Thread affinities can be set inside the application with the `shed_setaffinity()` function (see Linux man pages). Use of this call and how a particular application can be tuned is beyond the scope of this guide.

If the settings have been correctly applied, all interrupts from eth0 are being handled on CPU[0]. This can be checked:

```
# cat /proc/interrupts | grep eth0-
123:      13302      0      0      0      PCI-MSI-X  eth0-0
131:          0      24      0      0      PCI-MSI-X  eth0-1
139:          0      0      32      0      PCI-MSI-X  eth0-2
147:          0      0      0      21      PCI-MSI-X  eth0-3
```

Example 2:

An example of affinitizing each interface to a CPU on the same package:

First identify which interrupt lines are servicing which CPU and IO device:

```
# cat /proc/interrupts | grep eth0-
123:      13302      0  1278131      0      PCI-MSI-X  eth0-0
# cat /proc/interrupts | grep eth1-
131:          0      24      0      0      PCI-MSI-X  eth1-0
```

Find CPUs on same package (have same ‘package-id’):

```
# more /sys/devices/system/cpu/cpu*/topology/physical_package_id
::::::::::
/sys/devices/system/cpu/cpu0/topology/physical_package_id
::::::::::
1
::::::::::
/sys/devices/system/cpu/cpu10/topology/physical_package_id
::::::::::
1
::::::::::
/sys/devices/system/cpu/cpu11/topology/physical_package_id
::::::::::
0
...  
...
```

Having determined that cpu0 and cpu10 are on package 1, we can assign each ethX interface’s MSI-X interrupt to its own CPU on the same package. In this case we choose package 1:

```
# echo 1 > /proc/irq/123/smp_affinity      # 1hex is bit 0 = CPU0
# echo 400 > /proc/irq/131/smp_affinity      # 400hex is bit 10 = CPU10
```

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different speeds and widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen 1, 5.0 Gbps for PCIe Gen 2 and 8.0Gbps for PCIe Gen 3) in each direction. *Solarflare adapters are designed for x8 or x16 lane operation.*

On some server motherboards, choice of PCIe slot is important. This is because some slots (including those that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will warn if it detects that the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

Solarflare adapters require a PCIe Gen 3 x8 or x16 slot for optimal performance. The Solarflare driver will warn if it detects that the adapter is placed in a sub-optimal slot.

Warning messages can be viewed in dmesg from /var/log/messages.

The `lspci` command can be used to discover the currently negotiated PCIe lane width and speed:

```
lspci -d 1924: -vv
02:00.1 Class 0200: Unknown device 1924:0710 (rev 01)
...
Link: Supported Speed 2.5Gb/s, Width x8, ASPM L0s, Port 1
Link: Speed 2.5Gb/s, Width x8
```



NOTE: The Supported speed may be returned as 'unknown', due to older `lspci` utilities not knowing how to determine that a slot supports PCIe Gen. 2.0/5.0 Gb/s or PCIe Gen 3.0/8,0 Gb/s.

In addition, the latency of communications between the host CPUs, system memory and the Solarflare PCIe adapter may be PCIe slot dependent. Some slots may be "closer" to the CPU, and therefore have lower latency and higher throughput. If possible, install the adapter in a slot which is local to the desired NUMA node

Please consult your server user guide for more information.

CPU Speed Service

Most Linux distributions will have the `cpuspeed` service running by default. This service controls the CPU clock speed dynamically according to current processing demand. For latency sensitive applications, where the application switches between having packets to process and having periods of idle time waiting to receive a packet, dynamic clock speed control may increase packet latency. Solarflare recommend disabling the `cpuspeed` service if minimum latency is the main consideration.

The service can be disabled temporarily:

```
/sbin/service cpuspeed stop
```

The service can be disabled across reboots:

```
/sbin/chkconfig -level 12345 cpuspeed off
```

CPU Power Service

On RHEL7 systems, cpuspeed is replaced with cpupower. Solarflare recommend disabling the cpupower service if minimum latency is the main consideration. The service is controlled via systemctl:

```
systemctl stop cpupower  
systemctl disable cpupower
```

Tuned Service

On RHEL7 systems, it may be beneficial to disable the tuned service if minimum latency is the main consideration. Users are advised to experiment. The service is controlled via systemctl:

```
systemctl stop tuned  
systemctl disable tuned
```

Busy poll

If the kernel supports the *busy poll* features (Linux 3.11 or later), and minimum latency is the main consideration, Solarflare recommend that the busy_poll socket options should be enabled with a value of 50 microseconds as follows:

```
sysctl net.core.busy_poll=50 && sysctl net.core.busy_read=50
```

Only sockets having a non-zero value for SO_BUSY_POLL will be polled, so the user should do one of the following:

- set the poll timeout with the global busy_read option, as shown above,
- set the per-socket SO_BUSY_POLL socket option on selected sockets.

Setting busy_read also sets the default value for the SO_BUSY_POLL option.

Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of memory modules (DIMMs) to populate in the server. For optimal memory bandwidth in the system, it is likely that:

- all DIMM slots should be populated
- all NUMA nodes should have memory installed.

Please consult the motherboard documentation for details.

Intel® QuickData / NetDMA

On systems that support Intel I/OAT (I/O Acceleration Technology) features such as QuickData (a.k.a NetDMA), Solarflare recommend that these are enabled as they are rarely detrimental to performance.

Using Intel® QuickData Technology allows data copies to be performed by the system and not the operating system. This enables data to move more efficiently through the server and provide fast, scalable, and reliable throughput.

Enabling QuickData

- On some systems the hardware associated with QuickData must first be enabled (once only) in the BIOS
- Load the QuickData drivers with `modprobe ioatdma`

Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

Tuning Recommendations

The following tables provide recommendations for tuning settings for different applications.

- Throughput - [Table 26 on page 104](#)
- Latency - [Table 27 on page 106](#)
- Forwarding - [Table 28 on page 107](#)

Recommended Throughput Tuning

Table 26: Throughput Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: <code>/sbin/ifconfig <ethX> mtu <size></code>
Interrupt moderation	Leave at default (Enabled).
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Leave at default (Enabled).

Table 26: Throughput Tuning Settings

Tuning Parameter	How?
TCP Protocol Tuning	Leave at default for 2.6.16 and later kernels. For earlier kernels: <code>sysctl net.core.tcp_rmem 4096 87380 524288</code> <code>sysctl net.core.tcp_wmem 4096 87380 524288</code>
Receive Side Scaling (RSS)	Application dependent
Interrupt affinity & irqbalance service	Interrupt affinity settings are application dependent Stop irq balance service: <code>/sbin/service irqbalance stop</code> Reload the drivers to use the driver default interrupt affinity.
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Speed Service (cpuspeed)	Leave enabled.
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Intel QuickData (Intel chipsets only)	Enable in BIOS and install driver: <code>modprobe ioatdma</code>

Recommended Latency Tuning

[Table 27](#) shows recommended tuning settings for latency:

Table 27: Latency Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: <code>/sbin/ifconfig <ethX> mtu <size></code>
Interrupt moderation	Disable with: <code>ethtool -C <ethX> rx-usecs-irq 0</code>
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Disable using sysfs: <code>echo 0 > /sys/class/net/ethX/device/lro</code>
TCP Protocol Tuning	Leave at default, but changing does not impact latency.
Receive Side Scaling	Application dependent.
Interrupt affinity & irqbalance service	Interrupt affinity settings are application dependent Stop irq balance service: <code>/sbin/service irqbalance stop</code> Reload the drivers to use the driver default interrupt affinity.
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Speed Service (cpuspeed)	Disable with: <code>/sbin/service cpuspeed stop</code>
CPU Power Service (cpupower)	Disable with: <code>systemctl stop cpupower</code> <code>systemctl disable cpupower</code>

Table 27: Latency Tuning Settings

Tuning Parameter	How?
Tuned Service	Experiment disabling this with: <code>systemctl stop tuned</code> <code>systemctl disable tuned</code>
Busy poll (Linux 3.11 and later)	Enable with a value of 50µs: <code>sysctl net.core.busy_poll=50 \</code> <code>&& sysctl net.core.busy_read=50</code>
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Intel QuickData (Intel chipsets only)	Enable in BIOS and install driver: <code>modprobe ioatdma</code>

Recommended Forwarding Tuning

Table 28 shows recommended tuning settings for forwarding

Table 28: Forwarding Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: <code>/sbin/ifconfig <ethX> mtu <size></code>
Interrupt moderation	Configure an explicit interrupt moderation interval by setting the following driver options (see Driver Tuning on page 97): <code>irq_adapt_enable=0</code> <code>tx_irq_mod_usec=150</code>
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Disable using sysfs: <code>echo 0 > /sys/class/net/ethX/device/lro</code>
TCP Protocol Tuning	Leave at default for 2.6.16 and later kernels. For earlier kernels: <code>sysctl net.core.tcp_rmem 4096 87380 524288</code> <code>sysctl net.core.tcp_wmem 4096 87380 524288</code>
Receive Side Scaling (RSS)	Leave the <code>rss_cpus</code> option at the default, to use all CPUs for RSS. Ensure the <code>rss numa local</code> driver option is set to its default value of 1 (see Driver Tuning on page 97).

Table 28: Forwarding Tuning Settings

Tuning Parameter	How?
Interrupt affinity & irqbalance service	<p>Interrupt affinity. Affinitize each ethX interface to its own CPU (if possible select CPU's on the same Package). Refer to Interrupt Affinity on page 99.</p> <p>Stop irqbalance service:</p> <pre>/sbin/service irqbalance stop</pre>
Buffer Allocation Method	<p>Leave at default. Some applications may benefit from specific setting.</p> <p>The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored.</p>
Buffer Recycling	<p>Make receive buffer recycling more aggressive by setting the following driver option (see Driver Tuning on page 97):</p> <pre>rx_recycle_ring_size=256</pre>
PIO	<p>Disable PIO by setting the following driver option (see Driver Tuning on page 97):</p> <pre>piobuf_size=0</pre>
Transmit push	<p>Disable transmit push by setting the following driver option (see Driver Tuning on page 97):</p> <pre>tx_push_max_fill=0</pre>
Direct copying	<p>Disable copying directly from the network stack for transmits by setting the following driver option (see Driver Tuning on page 97):</p> <pre>tx_copybreak=0</pre>
Ring sizes	<p>Change the number of descriptor slots on each ring by setting the following driver options (see Driver Tuning on page 97):</p> <pre>tx_ring=512 rx_ring=512</pre> <p>Note that as the tx_irq_mod_usec interrupt moderation interval increases, the number of required tx_ring and rx_ring descriptor slots also increases. Insufficient descriptor slots will cause dropped packets.</p>

3.23 Module Parameters

[Table 29](#) lists the available parameters in the Solarflare Linux driver module (`modinfo sfc`):

Table 29: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
sxps_enabled	Enable or disable the Solarflare net driver to perform transmit flow steering. If the kernel does support XPS, this should be enabled in the kernel before using the SARFS feature.	0 1	0
sarfs_table_size	The size of the table used to maintain SARFS filters.	uint	256
sarfs_global_holdoff_ms	The maximum rate at which SARFS will insert or remove filters. This can be increased on heavily loaded servers or decreased to increase responsiveness.	uint	10ms
sarfs_sample_rate	The frequency at which TCP packets are inspected by the SARFS feature. This can be increased on heavily loaded servers to reduce the CPU usage by ARFS. Setting the sample rate to a non-zero value enables the SARFS feature. See also <code>sxps_enabled</code> above. The recommended sample rate is 20.	uint	0 packets
piobuf_size	Identify the largest packet size that can use PIO. Setting this to zero effectively disables PIO	uint	256 bytes
rx_alloc_method	Allocation method used for RX buffers. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the <code>rx_alloc_method</code> parameter is ignored. See Buffer Allocation Method on page 96 .	uint	AVN(0) new kernels. PAGE(2) old kernels
rx_refill_threshold	RX descriptor ring fast/slow fill threshold (%).	uint	90
lro_table_size ¹	Size of the LRO hash table. Must be a power of 2.	uint	128

Table 29: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
lro_chain_max ¹	Maximum length of chains in the LRO hash table.	uint	20
lro_idle_jiffies ¹	Time (in jiffies) after which an idle connection's LRO state is discarded.	uint	101
lro_slow_start_packets ¹	Number of packets that must pass in-order before starting LRO.	uint	20000
lro_loss_packets ¹	Number of packets that must pass in-order following loss before restarting LRO.	uint	20
rx_desc_cache_size	Set RX descriptor cache size.	int	64
tx_desc_cache_size	Set TX descriptor cache size.	int	16
rx_xoff_thresh_bytes	RX fifo XOFF threshold.	int	-1 (auto)
rx_xon_thresh_bytes	RX fifo XON threshold.	int	-1 (auto)
lro	Large receive offload acceleration	int	1
separate_tx_channels	Use separate channels for TX and RX	uint	0
rss_cpus	Number of CPUs to use for Receive-Side Scaling, or 'packages', 'cores' or 'hyperthreads'	uint or string	<empty>
irq_adapt_enable	Enable adaptive interrupt moderation	uint	1
irq_adapt_low_thresh	Threshold score for reducing IRQ moderation	uint	10000
irq_adapt_high_thresh	Threshold score for increasing IRQ moderation	uint	20000
irq_adapt_irqs	Number of IRQs per IRQ moderation adaptation	uint	1000
napi_weight	NAPI weighting	uint	64
rx_irq_mod_usec	Receive interrupt moderation (microseconds)	uint	60
tx_irq_mod_usec	Transmit interrupt moderation (microseconds)	uint	150
allow_load_on_failure	If set then allow driver load when online self-tests fail	uint	0
onload_offline_selftest	Perform offline self-test on load	uint	1
interrupt_mode	Interrupt mode (0=MSIX, 1=MSI, 2=legacy)	uint	0
falcon_force_internal_sram	Force internal SRAM to be used	int	0

Table 29: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
rss numa local	Constrain RSS to use CPU cores on the NUMA node local the Solarflare adapter. Set to 1 to restrict, 0 otherwise.	0 1	1
max_vfs	Enable VFs in the net driver. When specified as a single integer the VF count will be applied to all PFs. When specified as a comma separated list, the first VF count is assigned to the PF with the lowest index i.e. the lowest MAC address, then the PF with the next highest MAC address etc.	uint	0

1. Check OS documentation for availability on SUSE and RHEL versions.

3.24 Linux ethtool Statistics

The Linux command `ethtool` will display an extensive range of statistics originated from the MAC on the Solarflare network adapter. To display statistics use the following command:

```
ethtool -S ethX
```

(where X is the ID of the Solarflare interface)

Using a Solarflare net driver earlier than version 4.4.1.1017, the `ethtool` statistics counters can be reset by reloading the `sfc` driver:

```
# modprobe -r sfc
# modprobe sfc
```

Drivers from version 4.4.1.1017 (included in `onload-201502`) have to manage multi-PF configurations and for this reason statistics are not reset by reloading the driver. The only methods currently available to reset stats is to cold-reboot (power OFF/ON) the server or reload the firmware image.

Per port statistics (`port_`) are from the physical adapter port. Other statistics are from the specified PCIe function.

Table 30 below lists the complete output from the `ethtool -S` command.



NOTE: `ethtool -S` output depends on the features supported by the adapter type

Table 30: Ethtool -S output

Field	Description
port_tx_bytes	Number of bytes transmitted.
port_tx_packets	Number of packets transmitted.
port_tx_pause	Number of pause frames transmitted with valid pause op_code.
port_tx_control	Number of control frames transmitted. Does not include pause frames.
port_tx_unicast	Number of unicast packets transmitted. Includes flow control packets.
port_tx_multicast	Number of multicast packets transmitted.
port_tx_broadcast	Number of broadcast packets transmitted.
port_tx_lt64	Number of frames transmitted where the length is less than 64 bytes.
port_tx_64	Number of frames transmitted where the length is exactly 64 bytes.
port_tx_65_to_127	Number of frames transmitted where the length is between 65 and 127 bytes
port_tx_128_to_255	Number of frames transmitted where the length is between 128 and 255 bytes
port_tx_256_to_511	Number of frames transmitted where the length is between 256 and 511 bytes
port_tx_512_to_1023	Number of frames transmitted where length is between 512 and 1023 bytes
port_tx_1024_to_15xx	Number of frames transmitted where the length is between 1024 and 1518 bytes (1522 with VLAN tag).
port_tx_15xx_to_jumbo	Number of frames transmitted where length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.
port_rx_bytes	Number of bytes received. Not include collided bytes.
port_rx_good_bytes	Number of bytes received without errors. Excludes bytes from flow control packets.
port_rx_bad_bytes	Number of bytes with invalid FCS. Includes bytes from packets that exceed the maximum frame length.
port_rx_packets	Number of packets received.
port_rx_good	Number of packets received with correct CRC value and no error codes.

Table 30: Ethtool -S output

Field	Description
port_rx_bad	Number of packets received with incorrect CRC value.
port_rx_pause	Number of pause frames received with valid pause op_code.
port_rx_control	Number of control frames received. Does not include pause frames.
port_rx_unicast	Number of unicast packets received.
port_rx_multicast	Number of multicast packets received.
port_rx_broadcast	Number of broadcasted packets received.
port_rx_lt64	Number of packets received where the length is less than 64 bytes.
port_rx_64	Number of packets received where the length is exactly 64 bytes.
port_rx_65_to_127	Number of packets received where the length is between 65 and 127 bytes.
port_rx_128_to_255	Number of packets received where the length is between 128 and 255 bytes.
port_rx_256_to_511	Number of packets received where the length is between 256 and 511 bytes.
port_rx_512_to_1023	Number of packets received where the length is between 512 and 1023 bytes.
port_rx_1024_to_15xx	Number of packets received where the length is between 1024 and 1518 bytes (1522 with VLAN tag).
port_rx_15xx_to_jumbo	Number of packets received where the length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.
port_rx_gtjumbo	Number of packets received with a length greater than 9000 bytes.
port_rx_bad_gtjumbo	Number of packets received with a length greater than 9000 bytes, but with incorrect CRC value.
port_rx_overflow	Number of packets dropped by receiver because of FIFO overrun.

Table 30: Ethtool -S output

Field	Description
port_rx_nodesc_drop_cnt	Number of packets dropped by the network adapter because of a lack of RX descriptors in the RX queue.
port_rx_nodesc_drops	Packets can be dropped by the NIC when there are insufficient RX descriptors in the RX queue to allocate to the packet. This problem occurs if the receive rate is very high and the network adapter receive cycle process has insufficient time between processing to refill the queue with new descriptors. A number of different steps can be tried to resolve this issue: <ul style="list-style-type: none"> • Disable the irqbalance daemon in the OS • Distribute the traffic load across the available CPU/cores by setting rss_cpus=cores. Refer to Receive Side Scaling section • Increase receive queue size using ethtool.
port_rx_pm_trunc_bb_overflow	Overflow of the packet memory burst buffer - should not occur.
port_rx_pm_discard_bb_overflow	Number of packets discarded due to packet memory buffer overflow.
port_rx_pm_trunc_vfifo_full	Number of packets truncated or discarded because there was not enough packet memory available to receive them. Happens when packets cannot be delivered as quickly as they arrive due to: <ul style="list-style-type: none"> • packet rate exceeds maximum supported by the adapter. • adapter is inserted into a low speed or low width PCI slot – so the PCIe bus cannot support the required bandwidth. • packets are being replicated by the adapter and the resulting bandwidth cannot be handled by the PCIe bus. • host memory bandwidth is being used by other devices resulting in poor performance for the adapter.
port_rx_pm_discard_vfifo_full	Count of the number of packets dropped because of a lack of main packet memory on the adapter to receive the packet into.
port_rx_pm_trunc_qbb	Not currently supported.
port_rx_pm_discard_qbb	Not currently supported.
port_rx_pm_discard_mapping	Number of packets dropped because they have an 802.1p priority level configured to be dropped

Table 30: Ethtool -S output

Field	Description
port_rx_dp_q_disabled_packets	Increments when the filter indicates the packet should be delivered to a specific rx queue which is currently disabled due to configuration error or error condition.
port_rx_dp_di_dropped_packets	Number of packets dropped because the filters indicate the packet should be dropped. Can happen because: <ul style="list-style-type: none"> • the packet does not match any filter. • the matched filter indicates the packet should be dropped.
port_rx_dp_streaming_packets	Number of packets directed to RXDP streaming bus which is used if the packet matches a filter which directs it to the MCPU. Not currently used.
port_rx_dp_hlb_fetch	Count the number of times the adapter descriptor cache is empty and a fetch operation is triggered to refill with more descriptors.
port_rx_dp_hlb_wait	Packet arrives while adapter descriptor cache is empty, refill is in progress, but not yet complete.
rx_unicast	Number of unicast packets received.
rx_unicast_bytes	Number of unicast bytes received.
rx_multicast	Number of multicast packets received.
rx_multicast_bytes	Number of multicast bytes received.
rx_broadcast	Number of broadcast packets received.
rx_broadcast_bytes	Number of broadcast bytes received.
rx_bad	Number of packets received with incorrect CRC value.
rx_bad_bytes	Number of bytes received from packets with incorrect CRC value.
rx_overflow	Number of packets dropped by receiver because of FIFO overrun.
tx_unicast	Number of unicast packets transmitted.
tx_unicast_bytes	Number of unicast bytes transmitted.
tx_multicast	Number of multicast packets transmitted.
tx_multicast_bytes	Number of multicast bytes transmitted.
tx_broadcast	Number of broadcast packets transmitted.
tx_broadcast_bytes	Number of broadcast bytes transmitted.
tx_bad	0.

Table 30: Ethtool -S output

Field	Description
tx_bad_bytes	0.
tx_overflow	Number of packets dropped by transmitter because of FIFO overrun.
fec_uncorrected_errors	Number of uncorrected errors (RS-FEC)
fec_corrected_errors	Number of corrected errors (RS-FEC)
fec_corrected_symbols_lane0	
fec_corrected_symbols_lane1	
fec_corrected_symbols_lane2	per 25G lane corrected symbols
fec_corrected_symbols_lane3	
ctpio_vi_busyFallback	When a CTPIO push occurs from a VI, but the VI DMA datapath is still busy with packets in flight or waiting to be sent. The packet is sent over the DMA datapath.
ctpio_long_write_success	Host wrote excess data beyond 32-byte boundary after frame end, but the CTPIO send was successful.
ctpio_missing_dbell_fail	When CTPIO push is not accompanied by a TX doorbell.
ctpio_overflow_fail	When the host pushes packet bytes too fast and overflows the CTPIO buffer.
ctpio_underflow_fail	When the host fails to push packet bytes fast enough to match the adapter port speed. The packet is truncated and data transmitted as a poisoned packet.
ctpio_timeout_fail	When host fails to send all bytes to complete the packet to be sent by CTPIO before the VI inactivity timer expires. The packet is truncated and data transmitted as a poisoned packet.
ctpio_noncontig_wr_fail	A non-sequential address (for packet data) is encountered during CTPIO, caused when packet data is sent over PCIe interface as out-of-order or with gaps. Packet is truncated and transmitted as a poisoned packet.
ctpio_frm_clobber_fail	When a CTPIO push from one VI would have ‘clobbered’ a push already in progress by the same VI or another VI. One or both packets are sent over the DMA datapath - no packets are dropped.

Table 30: Ethtool -S output

Field	Description
ctpio_invalid_wr_fail	If packet length is less than length advertised in the CTPIO header the CTPIO fails. Or packet write is not aligned to (or multiple of) 32-bytes, Packet maybe transmitted as a poisoned packet if sending has already started. Or erased if send has not already started.
ctpio_vi_clobberFallback	When a ctpio collided with another already in progress. The in-progress packet succeeds, other packet is sent via DMA.
ctpio_unqualifiedFallback	When the VI is not enabled to send using ctpio or first write is not the packet header. The packet is sent using DMA datapath.
ctpio_runtFallback	Length in header < 29 bytes. The packet is sent using DMA datapath.
ctpio_success	Number of successful ctpio tx events
ctpioFallback	Number of instances when CTPIO push was rejected. This can occur because: <ul style="list-style-type: none"> • the VI legacy datapath is still busy • another CTPIO is in progress • VI is not enabled to use CTPIO • push request for illegal sized frame Fallback events do not result in poison packets. Rejected packets will use the DMA datapath path.
ctpioPoison	When the packet send has started, if CTPIO has to abort this packet, a corrupt CRC is attached to the packet. A poisoned packet may be sent over the wire - depending on the mode. The packet is sent using DMA datapath.
ctpioErase	Before a packet send has started. Corrupt, undersized or poisoned packets are erased from the CTPIO datapath. The packet is sent using DMA datapath.
tx_merge_events	The number of TX completion events where more than one TX descriptor was completed.
tx_tso_bursts	Number of times when outgoing TCP data is split into packets by the adapter driver. Refer to TCP Segmentation Offload (TSO) on page 93 .
tx_tso_long_headers	Number of times TSO is applied to packets with long headers.

Table 30: Ethtool -S output

Field	Description
tx_tso_packets	Number of physical packets produced by TSO.
tx_tso_fallbacks	0
tx_pushes	Number of times a packet descriptor is ‘pushed’ to the adapter from the network adapter driver.
tx_pio_packets	Number of packets sent using PIO.
tx_cb_packets	0
rx_reset	0
rx_tobe_disc	<p>Number of packets marked by the adapter to be discarded because of one of the following:</p> <ul style="list-style-type: none"> • Mismatch unicast address and unicast promiscuous mode is not enabled. • Packet is a pause frame. • Packet has length discrepancy. • Due to internal FIFO overflow condition. • Length < 60 bytes.
rx_[inner outer]ip_hdr_chksum_err	Number of packets received with IP header Checksum error.
rx_[inner outer]tcp_udp_chksum_err	Number of packets received with TCP/UDP checksum error.
rx_eth_crc_err	Number of packets received where the CRC did not match the internally generated CRC value. This is the total of all receive channels receiving CRC errors.
rx_mcast_mismatch	Number of unsolicited multicast packets received. Unwanted multicast packets can be received because a connected switch simply broadcasts all packets to all endpoints or because the connected switch is not able or not configured for IGMP snooping - a process from which it learns which endpoints are interested in which multicast streams.
rx_frm_trunc	Number of frames truncated because an internal FIFO is full. As a packet is received it is fed by the MAC into a 128K FIFO. If for any reason the PCI interface cannot keep pace and is unable to empty the FIFO at a sufficient rate, the MAC will be unable to feed more of the packet to the FIFO. In this event the MAC will truncate the frame - marking it as such and discard the remainder. The driver on seeing a 'partial' packet which has been truncated will discard it.
rx_merge_events	Number of RX completion events where more than one RX descriptor was completed.
rx_merge_packets	Number of packets delivered to the host through merge events.

Table 30: Ethtool -S output

Field	Description
tx-N.tx_packets	Per TX queue transmitted packets.
rx_N.rx_packets	Per RX queue received packets.
rx_no_skb_drops	Number of packets dropped by the adapter when there are insufficient socket buffers available to receive packets into. See also port_rx_nodesc_drop_cnt and port_rx_nodesc_drops above.
rx_nodesc_trunc	Number of frames truncated when there are insufficient descriptors to receive data into. Truncated packets will be discarded by the adapter driver.
ptp_good_syncs	These PTP stats counters relate to the mechanism used by sftpd to synchronize the system clock and adapter clock(s) in a server.
ptp_fast_syncs	
ptp_bad_syncs	For each synchronization event sftpd will select a number of system clock times to be compared to the adapter clock time. If the times can be synchronized, the good_syncs counter is incremented, otherwise the bad_syncs counter is incremented. If sftpd is unable to synchronize the clocks at this event, the sync_timeout counter is incremented.
ptp_sync_timeouts	
ptp_no_time_syncs	
ptp_invalid_sync_windows	
ptp_undersize_sync_windows	
ptp_oversize_sync_windows	sftpd will synchronize clocks 16 times per second - so incrementing counters does not necessarily indicate bad synchronization between local server clocks and an external PTP master clock.
ptp_rx_no_timestamp	Number of PTP packets received for which a hardware timestamp was not recovered from the adapter.
ptp_tx_timestamp_packets	Number of PTP packets transmitted for which the adapter generated a hardware timestamp.
ptp_rx_timestamp_packets	Number of PTP packets received for which the adapter generated a hardware timestamp.
ptp_timestamp_packets	Total number of PTP packets for which the adapter generated a hardware timestamp.
ptp_filter_matches	Number of PTP packets hitting the PTP filter.
ptp_non_filter_matches	Number of PTP packets which did not match the PTP filter.



NOTE: The adapter will double count packets less than 64bytes (port_rx_lt64) as also being a CRC error. This can result in port_rx_bad => rx_eth_crc_err counter. The difference should be equal to the port_rx_lt64 counter.

3.25 Driver Logging Levels

For the Solarflare net driver, two settings affect the verbosity of log messages appearing in dmesg output and /var/log/messages:

- The kernel console log level
- The netif message per network log level

The kernel console log level controls the overall log message verbosity and can be set with the command dmesg -n or through the /proc/sys/kernel/printk file:

```
echo 6 > /proc/sys/kernel/printk
```

Refer to ‘man 2 syslog’ for log levels and Documentation/sysctl/kernel.txt for a description of the values in /proc/sys/kernel/printk.

The netif message level provides additional logging control for a specified interface. These message levels are documented in Documentation/networking/netif-msg.txt. A message will only appear on the terminal console if both the kernel console log level and netif message level requirements are met.

The current netif message level can be viewed using the following command:

```
ethtool <iface> | grep -A 1 'message level:'  
Current message level: 0x000020f7 (8439)  
drv probe link ifdown ifup rx_err tx_err hw
```

Changes to the netif message level can be made with ethtool. Either by name:

```
ethtool -s <iface> msglvl rx_status on
```

or by bit mask:

```
ethtool -s <iface> msglvl 0x7fff
```

The initial setting of the netif msg level for all interfaces is configured using the debug module parameter e.g.

```
modprobe sfc debug=0x7fff  
ethtool <iface> | grep -A 1 'message level:'  
Current message level: 0x00007fff (32767)  
drv probe link timer ifdown ifup rx_err  
tx_err tx_queued intr tx_done rx_status pktdata hw wol
```

3.26 Running Adapter Diagnostics

You can use ethtool to run adapter diagnostic tests. Tests can be run offline (default) or online. Offline runs the full set of tests, which can interrupt normal operation during testing. Online performs a limited set of tests without affecting normal adapter operation.



CAUTION: Offline tests should not be run while sfptpd is running. The PTP daemon should be terminated before running the offline test.

As root user, enter the following command:

```
ethtool --test ethX offline|online
```

The tests run by the command are as follows:

Table 31: Adapter Diagnostic Tests

Diagnostic Test	Purpose
core.nvram	Verifies the flash memory ‘board configuration’ area by parsing and examining checksums.
core.register	Verifies the adapter registers by attempting to modify the writable bits in a selection of registers.
core.interrupt	Examines the available hardware interrupts by forcing the controller to generate an interrupt and verifying that the interrupt has been processed by the network driver.
tx/rx.loopback	Verifies that the network driver is able to pass packets to and from the network adapter using the MAC and Phy loopback layers.
core.memory	Verifies SRAM memory by writing various data patterns (incrementing bytes, all bit on and off, alternating bits on and off) to each memory location, reading back the data and comparing it to the written value.
core.mdio	Verifies the MII registers by reading from PHY ID registers and checking the data is valid (not all zeros or all ones). Verifies the MMD response bits by checking each of the MMDs in the Phy is present and responding.
chanX eventq.poll	Verifies the adapter’s event handling capabilities by posting a software event on each event queue created by the driver and checking it is delivered correctly. The driver utilizes multiple event queues to spread the load over multiple CPU cores (RSS).
phy.bist	Examines the PHY by initializing it and causing any available built-in self tests to run.

3.27 Running Cable Diagnostics

Cable diagnostic data can be gathered from the Solarflare 10GBASE-T adapters physical interface using the `ethtool -t` command which runs a comprehensive set of diagnostic tests on the controller, PHY, and attached cables. To run the cable tests enter the following command:

```
ethtool -t ethX [online | offline]
```

Online tests are non-intrusive and will not disturb live traffic.



CAUTION: Offline tests should not be run while sfptpd is running. The PTP daemon should be terminated before running the offline test.

The following is an extract from the output of the ethtool diagnostic offline tests:

```
phy    cable.pairA.length      9
phy    cable.pairB.length      9
phy    cable.pairC.length      9
phy    cable.pairD.length      9
phy    cable.pairA.status      1
phy    cable.pairB.status      1
phy    cable.pairC.status      1
phy    cable.pairD.status      1
```

Cable length is the estimated length in metres. A length value of 65535 indicates length not estimated due to pair busy or cable diagnostic routine not completed successfully.

The cable status can be one of the following values:

- 0 - invalid, or cable diagnostic routine did not complete successfully
- 1 - pair ok, no fault detected
- 2 - pair open or $R_t > 115$ ohms
- 3 - intra pair short or $R_t < 85$ ohms
- 4 - inter pair short or $R_t < 85$ ohms
- 9 - pair busy or link partner forces 100Base-Tx or 1000Base-T test mode.

4

Solarflare Adapters on Windows

This chapter documents procedures for the configuration and management of Solarflare adapters on **Windows Server 2016** and later Windows Server versions.

- [Windows 2016 Driver](#) on page 124
- [Legacy Driver](#) on page 124
- [Minimum Driver and Firmware Packages](#) on page 124
- [Firmware Variants](#) on page 125
- [Windows Feature Set](#) on page 126
- [Installing Solarflare Driver Package](#) on page 127
- [Adapter Configuration](#) on page 130
- [Flow Control](#) on page 131
- [Jumbo Frames](#) on page 132
- [Checksum Offload](#) on page 132
- [Large Send Offload \(LSO\)](#) on page 133
- [Interrupt Moderation \(Interrupt Coalescing\)](#) on page 133
- [NUMA Node](#) on page 134
- [Receive Side Scaling \(RSS\)](#) on page 135
- [Ethernet Frame Length](#) on page 137
- [Ethernet Link Speed](#) on page 137
- [Teaming and VLANs](#) on page 139
- [Adapter Statistics](#) on page 142
- [Performance Tuning on Windows](#) on page 143

4.1 Windows 2016 Driver

The Solarflare adapter driver package includes a Windows driver based on the NDIS 6.60 specification for **Windows Server 2016** and later versions of Windows Server.

- The driver is not supported on Windows Server versions earlier than 2016.
- The driver is not supported on Windows Client OS versions, this includes Windows 10 clients.

The Solarflare driver currently supports the following Solarflare adapters:

- X2522 (10G) and X2522-25G adapter models.

The driver is distributed as a single .zip package containing the .inf installer and uses standard Windows tools for installation, adapter configuration and management.

4.2 Legacy Driver

Earlier generations of Solarflare adapters; SFN5000, SFN6000, SFN7000 and SFN8000 series, are supported on Windows Server platforms, including Windows Server 2016, using the Solarflare legacy Bus/NDIS based driver.

A previous issue of this user manual (Issue 22 or earlier), documenting the configuration/management of Solarflare adapters using the Bus/NDIS driver, is available from support@solarflare.com.



NOTE: The legacy driver is no longer under active development for new features, but continues to be maintained for security updates and bug fixes.

4.3 System Requirements

Refer to [Software Driver Support on page 14](#) for details of supported Windows versions.

4.4 Driver Certification

The Solarflare Windows 2016 driver and Solarflare X2 series adapters are certified compatible with the Windows Hardware Compatibility Program (WHCP).

Drivers are available via Windows update and from: support@solarflare.com.

4.5 Minimum Driver and Firmware Packages

- Driver: SF-121281-LS issue 1 containing driver 1.0.0.1012
- Firmware: 7.4.0.1021

4.6 Firmware Variants

The Solarflare adapter firmware can be configured into different variants of the adapter firmware.

Firmware variant	Limitations
Full-feature	None.
Ultra-low-latency	Overlay acceleration, VLAN strip/insert, SRIOV are disabled.
Auto	Selects Full-feature.

4.7 Windows Feature Set

The following table lists the features supported by Solarflare adapters on Windows.

Table 32: Solarflare Windows Features

Flow Control	Refer to Flow Control on page 131
Jumbo frames	MTUs (Maximum Transmission Units) from 1500 bytes to 9216 bytes. <ul style="list-style-type: none">• Refer to Jumbo Frames on page 132
Task offloads	Large Segmentation Offload (LSO) and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. <ul style="list-style-type: none">• Checksum Offload on page 132 and Large Send Offload (LSO) on page 133
Receive Side Scaling (RSS)	RSS multi-core load distribution technology. <ul style="list-style-type: none">• Receive Side Scaling (RSS) on page 135
Interrupt Moderation	Interrupt Moderation to reduce the number of interrupts on the host processor from packet events. <ul style="list-style-type: none">• Interrupt Moderation (Interrupt Coalescing) on page 133
Teaming and/or Link Aggregation	Improve server reliability and bandwidth by bonding physical ports, from one or more Solarflare adapters, into a team, having a single MAC address and which function as a single port providing redundancy against a single point of failure. <ul style="list-style-type: none">• Teaming and VLANs on page 139
Virtual LANs (VLANs)	Support for multiple VLANs per adapter: <ul style="list-style-type: none">• Teaming and VLANs on page 139
State and statistics analysis	<ul style="list-style-type: none">• Adapter Statistics on page 142

4.8 Installing Solarflare Driver Package

The adapter drivers are supplied as a single .zip package **SF-121281-LS** available from: support@solarflare.com.

A device driver package must be added (staged) to the Windows driver store before it can be installed on any device. Use the **pnputil** utility from command prompt or Powershell prompt to add (stage) or remove the Solarflare driver package.

Identify installed driver

```
PS> pnputil /enum-drivers

Published Name:          oem5.inf
Original Name:           sfn.inf
Provider Name:           Solarflare
Class Name:              Network adapters
Class GUID:              {4d36e972-e325-11ce-bfc1-08002be10318}
Driver Version:          10/05/2018 1.0.0.1012
Signer Name:              Microsoft Windows Hardware Compatibility Publisher
```

This will enumerate/list all third-party drivers in the driver store. This command does not list drivers packaged as part of the Windows OS distribution.

Remove and uninstall driver

```
PS> pnputil /delete-driver oem#.inf [/force]
# is the enumerated driver .inf file as identified from the /enum-drivers option.
```

Add and install driver

1 Copy the .zip file driver package to a directory on the Windows Server - **in this install example we create and use the directory C:\sfdriver**

2 Set location to the directory:

```
PS> Set-Location C:\sfdriver
```

3 Expand archive:

```
PS C:\sfdriver> Expand-Archive "SF-121281-LS-Solarflare XtremeScale X2 Series Windows Driver Package.zip" -DestinationPath .
```

The following files should be present in the directory:

```
license.txt      readme.txt      sfn.cat      sfn.inf      sfn.man      sfn.sys      sfn.wprp
SF-121281-LS-A-Solarflare XtremeScale X2 Series Windows Driver Package.zip
```

4 Install Driver:

```
PS C:\sfdriver> pnutil /add-driver sfn.inf /install [/subdirs]
```

Microsoft PnP Utility

```
Adding driver package: sfn.inf
Driver package added successfully.
Published Name:          oem5.inf
Driver package installed on matching devices.
Total driver packages: 1
Added driver packages: 1
```

This command will add the driver to the driver store and, using the /install directive means the driver will also be installed on adapters.

5 Install the manifest to allow for management event logging:

```
PS C:\sfdriver> wevtutil.exe install-manifest sfn.man /
resourceFilePath:"${Env:SystemRoot}\system32\drivers\sfn.sys" /
messageFilePath:"${Env:SystemRoot}\system32\drivers\sfn.sys"
```

6 Identify installed driver version:

```
PS C:\sfdriver> Get-NetAdapter | ? DriverProvider -eq Solarflare | Format-Table -View Driver
```

Name	InterfaceDescription	DriverFileName
Ethernet 4	Solarflare XtremeScale X2522 (10G) ...#2	SFN.sys

DriverDate	DriverVersion	NdisVersion
2018-10-05	1.0.0.1012	6.60

4.9 Configuration & Management

Solarflare adapters can be configured and managed using Windows Powershell cmdlets or the Windows standard GUI interface.

Refer to Microsoft documentation for information on standard adapter cmdlets:

<https://docs.microsoft.com/powershell/module/netadapter>

Powershell Cmdlets

For description and usage help on any cmdlet:

```
Get-Help <cmdlet-name>
```

For detailed description/usage information

```
Get-Help <cmdlet-name> [-Detailed|-Examples|-Full|-Online]
```

GUI:

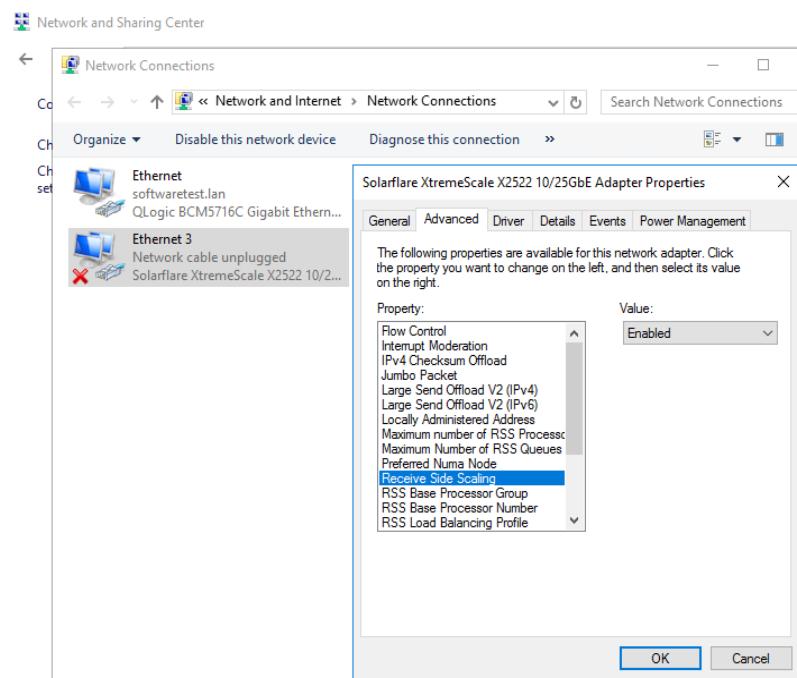
Launch the Network Connections window from command line:

```
PS > ncpa.cpl
```

or from control panel:

Control Panel > Network and Internet > Network and Sharing Center > Change adapter settings

Select an adapter (right click) to open the Properties window and use the Configure button to open the Advanced Properties window.



4.10 Adapter Configuration

Identify installed adapters

```
PS > Get-NetAdapter
```

Name	InterfaceDescription	ifIndex	Status
Ethernet 4	Solarflare XtremeScale X2522 10/25...#2	25	Up
MacAddress	LinkSpeed		
00-0F-53-64-4F-11	10 Gbps		

Adapter PCIe Information

```
PS > Get-NetAdapterHardwareInfo -Name "<interface>"
```

Name	Segment	Bus	Device	Function	Slot	NumaNode	PcieLinkSpeed	Width
Eth 4	0	1	0	1	1		8.0 GT/s	8

Adapter hardware properties

For an extensive list of adapter, adapter port and server hardware properties for the specified Solarflare interface.

```
PS > Get-NetAdapter -Name "<interface>" | Format-List -Property *
```

Adapter advanced (configuration) properties

For adapter configurable properties.

```
PS > Get-NetAdapterAdvancedProperty -Name "<interface>" [-AllProperties]
```

DisplayName	DisplayValue	RegKeyword	RegValue
Flow Control	Auto Negotiation	*FlowControl	{4}
Interrupt Moderation	Enabled	*InterruptMo...	{1}
IPv4 Checksum Offload	Rx & Tx Enabled	*IPChecksumO...	{3}
Jumbo Packet	1518	*JumboPacket	{1518}
Large Send Offload V2 (IPv4)	Disabled	*LS0v2IPv4	{0}
Large Send Offload V2 (IPv6)	Disabled	*LS0v2IPv6	{0}
Maximum number of RSS Proce...	16	*MaxRssProce...	{16}
Preferred Numa Node	1	*NumaNodeId	{1}
Maximum Number of RSS Queues	8	*NumRSSQueues	{8}
Receive Side Scaling	Enabled	*Rss	{1}
RSS Base Processor Group	0	*RssBaseProc...	{0}
RSS Base Processor Number	0	*RssBaseProc...	{0}
RSS Max Processor Group	9	*RssMaxProcG...	{9}
RSS Max Processor Number	1	*RssMaxProcN...	{1}
RSS Load Balancing Profile	NUMA Scaling Static	*RssProfile	{4}
Speed & Duplex	Auto Negotiation	*SpeedDuplex	{0}
TCP Checksum Offload (IPv4)	Rx & Tx Enabled	*TCPChecksum...	{3}
TCP Checksum Offload (IPv6)	Rx & Tx Enabled	*TCPChecksum...	{3}
UDP Checksum Offload (IPv4)	Rx & Tx Enabled	*UDPChecksum...	{3}
UDP Checksum Offload (IPv6)	Rx & Tx Enabled	*UDPChecksum...	{3}

4.11 Flow Control

Ethernet flow control allows two communicating devices to inform each other when they are being overloaded by received data. This prevents one device from overwhelming the other device with network packets. For instance, when a switch is unable to keep up with forwarding packets between ports. Solarflare adapters can auto-negotiate flow control settings with the link partner.

Flow control can be configured using adapter advanced properties.

Identify current settings

```
PS > Get-NetAdapterAdvancedProperty -Name "<interface>"
```

DisplayName	DisplayValue	RegKeyword	RegValue
Flow Control	Auto Negotiation	*FlowControl	{4}

Change settings

```
PS > Set-NetAdapterAdvancedProperty -Name "<interface>" -Displayname "Flow Control" -Displayvalue "Rx Enabled"
```

Option	Description
Auto-negotiation	Flow control is auto-negotiated between the devices. This is the default setting, preferring Rx & Tx Enabled if the link partner is capable.
Rx & Tx Enabled	Adapter generates and responds to flow control messages.
Tx Enabled	Adapter responds to flow control messages but is unable to generate messages if it becomes overwhelmed.
Rx Enabled	Adapter generates flow control messages but is unable to respond to incoming messages and will keep sending data to the link partner.
Disabled	Ethernet flow control is disabled on the adapter. Data will continue to flow even if the adapter or link partner is overwhelmed.

4.12 Jumbo Frames

Solarflare adapters support Jumbo frames up to 9216 bytes. Jumbo frames can be configured using adapter advanced properties.

Identify current settings

```
PS > Get-NetAdapterAdvancedProperty -Name "<interface>"
```

DisplayName	DisplayValue	RegKeyword	RegValue
Jumbo Packet	1518	*JumboPacket	{1518}

Change settings

```
PS > Set-NetAdapterAdvancedProperty -Name "<interface>" -DisplayName "Jumbo Packet" -DisplayValue "9216"
```

4.13 Checksum Offload

Checksum offloading is supported for IP, TCP and UDP packets. Before transmitting a packet, a checksum is generated by the adapter and appended to the packet. At the receiving end, the calculation is performed by the adapter against the received packet. Offloading checksum calculation to the network adapter decreases the work load on server CPUs.

Identify current settings

```
PS > Get-NetAdapterChecksumOffload -Name "<interface>"
```

Name	IpIPv4Enabled	TcpIPv4Enabled	TcpIPv6Enabled
Ethernet 4	RxTxEnabled	RxTxEnabled	RxTxEnabled

UdpIPv4Enabled	UdpIPv6Enabled
RxTxEnabled	RxTxEnabled

Checksums can be enabled/disabled in both transmit and receive directions:

TxEnabled, RxEnabled, RxTxEnabled, Disabled

Set values

```
PS > Set-NetAdapterChecksumOffload -Name "<interface>" -IpIPv4Enabled  
RxTxEnabled
```

Disable All

```
PS > Disable-NetAdapterChecksumOffload -Name "<interface>"
```

Returns all checksums to Disabled.

Enable All

```
PS > Enable-NetAdapterChecksumOffload -Name "<interface>"
```

Enabling returns all checksum to RxTxEnabled.



NOTE: Changing the Checksum Offload settings can impact the performance of the adapter. Solarflare recommend that these remain at the default values. Disabling Checksum Offload disables Large Send Offload.

Large Send Offload (LSO)

LSO offloads to the adapter the task of splitting large outgoing TCP data into smaller packets. This improves throughput performance and has no effect on latency.

Identify current settings

```
PS > Get-NetAdapterLso -Name "<interface>"
```

Name	Version	V1IPv4Enabled	IPv4Enabled	IPv6Enabled
Eth 4	LSO Version 2	False	True	True

Enable LSO

```
PS > Enable-NetAdapterLso -Name "<interface>"
```

Disable LSO

```
PS > Disable-NetAdapterLso -Name "<interface>"
```



NOTE: LSO is enabled by default and there is generally no reason to disable this feature.

4.14 Interrupt Moderation (Interrupt Coalescing)

Reduces the number of interrupts generated by the adapter by combining multiple received packet events and/or transmit completion events into a single interrupt thereby reducing the number of interrupts sent to the CPU and reducing the CPU workload.

Identify current settings

```
PS > Get-NetAdapterAdvancedProperty -Name "<interface>"
```

Name	DisplayName	DisplayValue	RegistryKeyword	RegistryValue
Eth 4	Interrupt Moderation	Enabled	*InterruptMo...	{1}

Enable

```
PS > Set-NetAdapterAdvancedProperty -Name "<interface>" -DisplayName "Interrupt Moderation Packet" -Displayvalue "Enabled"
```

Disable

```
PS > Set-NetAdapterAdvancedProperty -Name "<interface>" -Displayname  
"Interrupt Moderation Packet" -Displayvalue "Disabled"
```

4.15 NUMA Node

The adapter driver can select a subset of available CPU cores to handle transmit and receive processing. The preferred NUMA node setting can be used to constrain the set of CPU cores used to those on a specific NUMA Node.

To force processing onto a particular NUMA Node, set the preferred NUMA node value in the adapter advanced properties.

The NUMA distance of the cores used for the RSS queue and for the network application will influence performance.

NUMA Distance

To check the NUMA distance of each core from the interface.

```
PS > Get-NetAdapterRss -Name "<interface>"  
  
Name : Ethernet 4  
InterfaceDescription : Solarflare XtremeScale X2522 10/25GbE Adapter #2  
Enabled : True  
NumberOfReceiveQueues : 8  
Profile : NUMAStatic  
BaseProcessor: [Group:Number] : 0:0  
MaxProcessor: [Group:Number] : 0:3  
MaxProcessors : 4  
RssProcessorArray: [Group:Number/NUMA Distance] : 0:0/0 0:1/0 0:2/0  
0:3/0  
IndirectionTable: [Group:Number] :
```

RSS Queue - NUMA Node

For low latency low jitter applications, RSS queues should be mapped to NUMA nodes that are local to the interface. The local NUMA node is always selected automatically when either of the following RSS profiles is selected:

- ClosestProcessor
- ClosestProcessorStatic

Application - NUMA Node

For low latency low jitter, run the network applications on a NUMA node local to the interface.

```
> start /affinity <hexmask> <command>  
or  
> start /node <num> <command>
```

or

```
> start /node <num> /affinity <hexmask> <command>
```

Preferred NUMA Node

Assign the adapter to a specific NUMA node where the registry value is a number between 0-15 or use 65535 to use ALL.

```
PS > Set-NetAdapterAdvancedProperty -Name "<interface>" -RegistryKeyword '*numanodeid' -RegistryValue '0'
```

RSS - NICs sharing a NUMA node

When adapters share a NUMA node, RSS, for each adapter, can be limited to a subset of processors within the node.

```
PS > Set-NetAdapterRss -Name "<interface>" -NumaNode 0 -  
BaseProcessorNumber 1 -MaxProcessorNumber 4
```

Allows the adapter to use processors 1,2,3,4 only on the NUMA node 0.

4.16 Receive Side Scaling (RSS)

RSS attempts to dynamically distribute data processing across the available host CPUs in order to spread the workload. RSS is enabled by default and can significantly improve the performance of the host CPU when handling large amounts of network data.

RSS cmdlets allow per-adapter RSS configuration, so different adapters can have different RSS configurations.

Identify current settings

```
PS > Get-NetAdapterRss -Name "<interface>"  
  
Name : Ethernet 4  
InterfaceDescription : Solarflare XtremeScale X2522 10/25GbE Adapter #2  
Enabled : True  
NumberOfReceiveQueues : 8  
Profile : NUMAStatic  
BaseProcessor: [Group:Number] : 0:0  
MaxProcessor: [Group:Number] : 0:3  
MaxProcessors : 4  
RssProcessorArray: [Group:Number/NUMA Distance] : 0:0/0 0:1/0 0:2/0  
0:3/0  
IndirectionTable: [Group:Number] :
```

Enable RSS

```
PS > Set-NetAdapterRss -Name "<interface>" -Enabled 1
```

Disable RSS

```
PS > Set-NetAdapterRss -Name "<interface>" -Enabled 0
```

Number of Receive Queues

```
PS > St-NetAdapterRss -Name "<interface>" -NumberOfReceiveQueues 4
```

RSS Profile

Determines the logical processors that can be used by a network adapter for RSS.

```
PS > Set-NetAdapterRss -Name "<interface>" -Profile closest
```

RSS Profile	Description
Closest	Processors near the network adapter's base RSS processor are preferred. Windows may rebalance processors dynamically based on load.
ClosestStatic	Processors near the network adapter's base RSS processor are preferred. Windows will NOT rebalance processors dynamically based on load.
NUMA	Will select processors on different NUMA nodes. Windows may rebalance processors dynamically based on load.
NUMAStatic	Default. Will select processors on different NUMA nodes. Windows will NOT rebalance processors dynamically based on load.
Conservative	Use as few processors as possible to sustain the load. This option can help to reduce the number of interrupts.

Refer to [NUMA Node on page 134](#) for further NUMA node considerations.



NOTE: Changing the RSS profile requires a restart of the adapter.

RSS Base Processor

For a Solarflare network adapter, the RSS base processor is 0 (zero), which means it starts processing on CPU core 0 which is also the default processor for all other general Windows processes and will likely be the default for all other network adapters in the server.

To avoid this unnecessary contention, set the adapter RSS base processor to another processor on the NUMA node the Solarflare adapter is assigned to.

cmdlet:

```
PS > Set-NetAdapterRss -Name "<interface>" -BaseProcessorNumber 1 -  
      Numanode 0
```

RSS Max Processor

Used with the BaseProcessorNumber, this identifies the range of processors that can be used by RSS.

cmdlet:

```
PS > Set-NetAdapterRss -Name "<interface>" -Numanode 0 -  
BaseProcessorNumber 4 -MaxProcessorNumber 7
```

The above example means that lowest processor that can be used for RSS is processor 4, and it can use processors 4,5,6,7.

RSS Base Processor Group

On systems with more than 64 logical processors - identify the processor group. Systems with <=64 processors have only the single group 0.

cmdlet:

```
PS > Set-NetAdapterRss -Name "<interface>" -BaseProcessorGroup <value>
```

NOTE: Setting the 'numanode' parameter will automatically set the correct base processor group.

RSS Max Processors

Set the number of processors to be used by RSS.

cmdlet:

```
PS > Set-NetAdapterRss -Name "<interface>" -MaxProcessors 4
```

Ethernet Frame Length

The maximum Ethernet frame length used by the adapter to transmit data is (or should be) closely related to the MTU (maximum transmission unit) of your network. The network MTU determines the maximum frame size that your network is able to transmit across all devices in the network.

NOTE: For optimum performance set the Ethernet frame length to your network MTU.

Ethernet Link Speed

Generally, it is not necessary to configure the link speed of the adapter. The adapter by default will negotiate the link speed dynamically, connecting at the maximum, supported speed. However, if the adapter is unable to connect to the link partner, set a fixed link speed through the advanced properties.

```
PS > Get-NetAdapterAdvancedProperty -Name "<interface>" | Format-List -  
Property "*"
```

ValueName	:	*SpeedDuplex
ValueData	:	{0}
ifAlias	:	Ethernet 4
InterfaceAlias	:	Ethernet 4
ifDesc	:	Solarflare XtremeScale X2522 10/25GbE Adapter
#2		

```
Caption : MSFT_NetAdapterAdvancedPropertySettingData
'Solarflare XtremeScale X2522 10/25GbE Adapter #2'
Description : Speed & Duplex
ElementName : Speed & Duplex
InstanceID : {F6B41E13-80D5-4BE0-B45C-
CC314CCADB6C}::*SpeedDuplex
InterfaceDescription : Solarflare XtremeScale X2522 10/25GbE Adapter
#2
Name : Ethernet 4
Source : 3
SystemName : <server>.<domain>.lan
DefaultDisplayValue : Auto Negotiation
DefaultRegistryValue : 0
DisplayName : Speed & Duplex
DisplayParameterType : 5
DisplayValue : Auto Negotiation
NumericParameterBaseValue :
NumericParameterMaxValue :
NumericParameterMinValue :
NumericParameterStepValue :
Optional : False
RegistryDataType : 1
RegistryKeyword : *SpeedDuplex
RegistryValue : {0}
ValidDisplayValues : {Auto Negotiation, 1.0 Gbps Full Duplex, 10
Gbps Full Duplex}
ValidRegistryValues : {0, 6, 7}
PSComputerName :
CimClass : ROOT/
StandardCimv2:MSFT_NetAdapterAdvancedPropertySettingData
CimInstanceProperties : {Caption, Description, ElementName,
InstanceID...}
CimSystemProperties :
Microsoft.Management.Infrastructure.CimSystemProperties
```

```
PS > Set-NetAdapterAdvancedProperty -Name "<interface>" -Displayname
"Speed & Duplex" -Displayvalue "10 Gbps Full Duplex"
```

4.17 Teaming and VLANs

About Teaming

Solarflare adapters use the native Windows **NetLbfo** module for teaming configuration and management. The following teaming configurations are supported:

- IEEE 802.1AX (802.3ad) Dynamic link aggregation.
- Static link aggregation.
- Fault tolerant teams.

Team Configurations

- All Solarflare adapter ports on all installed Solarflare adapters.
- Selected ports e.g. from a dual port Solarflare adapter, the first port could be a member of team A and the second port a member of team B or both ports members of the same team.
- Mixed Solarflare and non-Solarflare adapters.
- A port can be a member of more than one team.
- A port can be assigned more than one VLAN.

Link Aggregation

A mechanism supporting load balancing and fault tolerance across a team of network adapters and intermediate switch.

- Requires configuration at both ends of the link.
- All links in the team are bonded into a single virtual link with a single MAC address.

Two or more physical links can increase the potential throughput available between the link partners, and improve resilience against link failures.

- All links in the team must be between the same two link partners.
- Links must be full-duplex.
- Traffic is distributed evenly to all links connected to the same switch.
- In case of link failover, traffic on the failed link will be re-distributed to the remaining links.

Link aggregation offers the following functionality:

- Teams can be built from mixed media (i.e. UTP and Fiber).
- All protocols can be load balanced without transmit or receive modifications to frames.

- Multicast and broadcast traffic can be load balanced.
- Short recovery time in case of failover.
- Solarflare supports up to 64 link aggregation port groups per system.
- Solarflare supports up to 64 ports and VLANs in a link aggregation port group.

Dynamic Link Aggregation

Uses the Link Aggregation Control Protocol (LACP) (IEEE 802.1AX - previously called 802.3ad) to negotiate the ports that will make up the team.

- LACP must be enabled at both ends of the link for a team to be operational.
- LACP will automatically determine which physical links can be aggregated.
- Provides fault tolerance and load balancing.
- Standby links are supported, but are not considered part of a link aggregation until a link within the aggregation fails.
- VLANs are supported within 802.1AX teams.
- In the event of failover, the load on the failed link is redistributed over the remaining links.



NOTE: A switch must support 802.1AX (802.3ad) dynamic link aggregation to use this method of teaming.

Fault-Tolerant Teams

Fault tolerant teaming can be implemented on any switch. It can also be used with each team member network link connected to separate switches.

A fault-tolerant team is a set of one or more network adapters bound together by the teaming driver. The team improves network availability by providing standby adapters. At any one moment no more than one of the adapters will be active with the remainder either in standby or in a fault state.



NOTE: All adapters in a fault-tolerant team must be part of the same broadcast domain.

Failover

The teaming driver monitors the state of the active adapter and, in the event that its physical link is lost (down) or that it fails in service, swaps to one of the standby adapters. A link in a failed state will not be available as a standby while the failed state persists.

VLANs and Teaming

VLANs are used to divide a physical network into multiple broadcast domains and are supported on all Solarflare adapter teaming configurations.

Teaming - Configuration

Teams can be configured using NetLbfo specific powershell Cmdlets or via the NIC Teaming dialog (GUI).

Teaming - Cmdlets

The Windows teaming module, NetLbfo, can be configured using Powershell cmdlets.

```
PS > Get-Command -Module NetLbfo
```

Name	Version	Source
Add-NetLbfoTeamMember	2.0.0.0	NetLBFO
Add-NetLbfoTeamNic	2.0.0.0	NetLBFO
Get-NetLbfoTeam	2.0.0.0	NetLBFO
Get-NetLbfoTeamMember	2.0.0.0	NetLBFO
Get-NetLbfoTeamNic	2.0.0.0	NetLBFO
New-NetLbfoTeam	2.0.0.0	NetLBFO
Remove-NetLbfoTeam	2.0.0.0	NetLBFO
Remove-NetLbfoTeamMember	2.0.0.0	NetLBFO
Remove-NetLbfoTeamNic	2.0.0.0	NetLBFO
Rename-NetLbfoTeam	2.0.0.0	NetLBFO
Set-NetLbfoTeam	2.0.0.0	NetLBFO
Set-NetLbfoTeamMember	2.0.0.0	NetLBFO
Set-NetLbfoTeamNic	2.0.0.0	NetLBFO

For further information - use the Powershell cmdlet help:

```
PS > Get-Help [Add|Remove|Rename|Get|Set]-NetLbfo
```

A complete list describing NetLbfo (teaming) cmdlets can also be found with the following links:

[Windows-teaming-documentation](#)

<https://docs.microsoft.com/en-us/powershell/module/netlbfo/>

Teaming - GUI

Enter the following at a command prompt or powershell prompt:

```
PS > LbfoAdmin
```

Select the adapter(s) and then use the TASKS tab to create/configure/add/remove teams.

4.18 Adapter Statistics

Networking statistics for an adapter can be viewed with the following cmdlet.

```
PS > Get-NetAdapterStatistics -Name "<interface>"
```

Name	RxBytes	RxUnicastPackets	TxBytes	TxUnicastPackets
Ethernet 4	0	0	0	0

```
PS > Get-NetAdapterStatistics -Name "<interface>" | Format-List -Property  
**"
```

```
ifAlias : Ethernet 4
InterfaceAlias : Ethernet 4
ifDesc : Solarflare XtremeScale X2522 10/25GbE Adapter #2
Caption : MSFT_NetAdapterStatisticsSettingData
'Solarflare XtremeScale X2522 10/25GbE Adapter #2'
Description : Solarflare XtremeScale X2522 10/25GbE Adapter #2
ElementName : Solarflare XtremeScale X2522 10/25GbE Adapter #2
InstanceId : {F6B41E13-80D5-4BE0-B45C-CC314CCADB6C}
InterfaceDescription : Solarflare XtremeScale X2522 10/25GbE Adapter #2
Name : Ethernet 4
Source : 2
SystemName : <server>.<domain>.lan
OutboundDiscardedPackets : 0
OutboundPacketErrors : 0
RdmaStatistics :
ReceivedBroadcastBytes : 0
ReceivedBroadcastPackets : 0
ReceivedBytes : 0
ReceivedDiscardedPackets : 0
ReceivedMulticastBytes : 0
ReceivedMulticastPackets : 0
ReceivedPacketErrors : 0
ReceivedUnicastBytes : 0
ReceivedUnicastPackets : 0
RscStatistics : MSFT_NetAdapter_RscStatistics
SentBroadcastBytes : 0
SentBroadcastPackets : 0
SentBytes : 0
SentMulticastBytes : 0
SentMulticastPackets : 0
SentUnicastBytes : 0
SentUnicastPackets : 0
SupportedStatistics : 4163583
```

4.19 Performance Tuning on Windows

Introduction

Solarflare network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings designed for optimum performance across a broad class of applications.

Occasionally, application performance can be improved by additional tuning to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are more dependent on throughput.

The tuning recommendations should be considered in conjunction the following Microsoft performance tuning guides:

- [Performance Tuning Guide Windows Server 2016](#)

Max Frame Size

A larger maximum frame size will improve adapter throughput and CPU utilization. CPU utilization is improved, because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support maximum frame sizes up to 9216 bytes (this does not include the Ethernet preamble or frame check sequence).



NOTE: The maximum frame size setting should include the Ethernet frame header. The Solarflare drivers support 802.1p. This allows Solarflare adapters on Windows to optionally transmit packets with 802.1Q tags for QoS applications. It requires an Ethernet frame header size of 18 bytes (6 bytes source MAC address, 6 bytes destination MAC address, 2 bytes 802.1Q tag protocol identifier, 2 bytes 802.1Q tag control information, and 2 bytes EtherType). The default maximum frame size is therefore 1518 bytes.

The maximum frame size is changed by changing the Max Frame Size setting in the Network Adapter's Advanced Properties Page.

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation reduces the number of interrupts generated by the adapter by coalescing multiple received packet events and/or transmit completion events together into a single interrupt.

The *interrupt moderation interval* sets the minimum time (in microseconds) between two consecutive interrupts. Coalescing occurs only during this interval:

- When the driver generates an interrupt, it starts timing the moderation interval.
- Any events that occur before the moderation interval expires are coalesced together into a single interrupt, that is raised only when the interval expires. A new moderation interval then starts, during which no interrupt is raised.
- An event that occurs after the moderation interval has expired gets its own dedicated interrupt, that is raised immediately. A new moderation interval then starts, during which no interrupt is raised.

Interrupt moderation settings are **critical for tuning adapter latency**:

- Disabling the adaptive algorithm will:
 - reduce jitter
 - allow setting the moderation interval as required to suit conditions.
- Increasing the interrupt moderation interval will:
 - generate less interrupts
 - reduce CPU utilization (because there are less interrupts to process)
 - increase latency
 - improve peak throughput.
- Decreasing the interrupt moderation interval will:
 - generate more interrupts
 - increase CPU utilization (because there are more interrupts to process)
 - decrease latency
 - reduce peak throughput.
- Turning off interrupt moderation will:
 - generate the most interrupts
 - give the highest CPU utilization
 - give the lowest latency
 - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits typically outweigh the cost of increased CPU utilization. It is recommended that:

- Interrupt moderation is disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation is enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.

Interrupt moderation can be disabled or enabled using the Interrupt Moderation setting in the Network Adapter's Advanced Properties Page. The interrupt moderation time value can also be configured from the Network Adapter's Advanced Properties Page.

Receive Side Scaling (RSS)

RSS is enabled by default for best networking performance.

The number of RSS queues can be adjusted to suit the workload:

- The number of RSS CPUs is limited by the number of RSS queues. The driver does not target multiple RSS queues to the same CPU. Therefore:
 - It is best to set the maximum number of RSS queues to be equal to the maximum number of RSS CPUs (or the next higher setting if the equal option is unavailable).
 - The number of queues can be reduced in order to isolate CPU cores for application processing.
 - The number of queues can be increased to spread the load over more cores. This will also increase the amount of receive buffering due to a larger number of RX queues.



NOTE: If hyper-threading is enabled, RSS will only select one thread from each CPU core.

- For low latency low jitter applications select the NUMA scaling static RSS profile. Set both the maximum number of RSS processors and the number of RSS queues to be equal to the number of CPU cores
In multi-port scenarios, restrict each port to a subset of RSS processors using the base and max processor settings.
- For other applications use as few RSS processors as required to cope with the traffic load, leaving other CPUs free for other tasks

Other Considerations

PCI Express Lane Configurations

Solarflare adapters require a PCIe Gen 3 x16 slot for optimal performance. The Solarflare driver will insert a warning in the Windows Event Log if it detects that the adapter is placed in a sub-optimal slot.

In addition, the latency of communications between the host CPUs, system memory and the Solarflare PCIe adapter may be PCIe slot dependent. Some slots may be "closer" to the CPU, and therefore have lower latency and higher throughput:

- If possible, install the adapter in a slot which is local to the desired NUMA node

Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of memory modules (DIMMs) to populate in the server. For optimal memory bandwidth in the system, it is likely that:

- all DIMM slots should be populated
- all NUMA nodes should have memory installed.

Please consult the motherboard documentation for details.

Intel Hyper-Threading Technology

On systems that support Intel Hyper-Threading Technology users should consider benchmarking or application performance data when deciding whether to adopt hyper-threading on a particular system and for a particular application. Solarflare have identified that hyper-threading is generally beneficial on systems fitted with Core i5, Core i7 and Xeon (Nehalem or later) CPUs.

TCP/IP Options

TCP timestamps, window scaling and selective acknowledgments are enabled by default on supported platforms, and include receive window tuning and congestion control algorithms that automatically adapt to 10 gigabit connections. There is therefore no need to change these settings.

Power Saving Mode

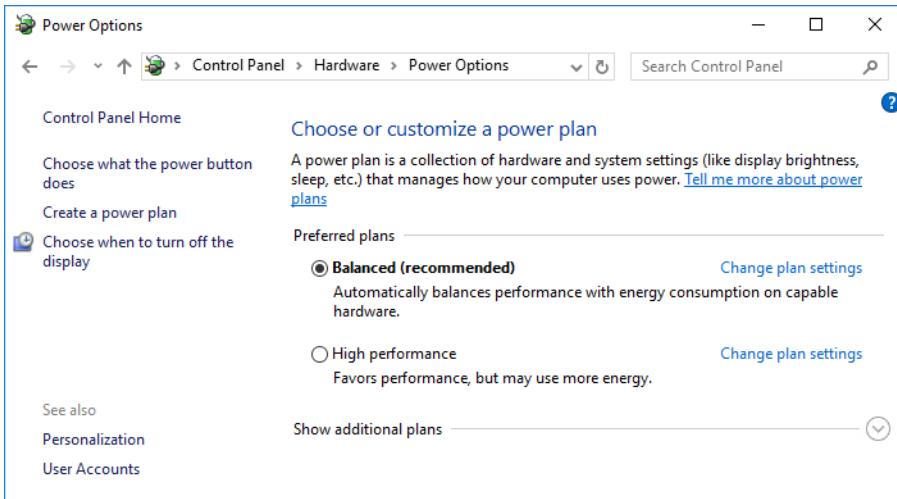
Modern processors utilize design features that enable a CPU core to drop into low power states when instructed by the operating system that the CPU core is idle. When the OS schedules work on the idle CPU core (or when other CPU cores or devices need to access data currently in the idle CPU core's data cache) the CPU core is signaled to return to the fully on power state. These changes in CPU core power states create additional network latency and jitter. Solarflare recommend to achieve the lowest latency and lowest jitter that the "C1E power state" or "CPU power saving mode" is disabled within the system BIOS.

In general the user should examine the system BIOS settings and identify settings that favor performance over power saving. In particular look for settings to disable:

- C states / Processor sleep/idle states
- Enhanced C1 CPU sleep state (C1E)
- Any deeper C states (C3 through to C6)
- P states / Processor throttling
- Processor Turbo mode
- Ultra Low Power State
- PCIe Active State Power Management (ASPM)

- Unnecessary SMM/SMI features

The latency can be improved by selecting the highest performance power plan from the Control Panel > Hardware > Power Options:



The `powercfg.exe` utility that is installed with Windows can also be used to select the power scheme.

- List all power schemes in the current user's environment:

```
PS > PowerCfg /LIST
```

```
Existing Power Schemes (* Active)
```

```
Power Scheme GUID: 381b4222-f694-41f0-9685-ff5bb260df2e (Balanced) *
Power Scheme GUID: 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c (High
performance)
Power Scheme GUID: a1841308-3541-4fab-bc81-f71556f20b4a (Power saver)
```

```
PS > PowerCfg /SETACTIVE <GUID>
```

Firewalls and anti-virus software

Depending on the system configuration, the following software may have a significant impact on throughput and CPU utilization, in particular when receiving multicast UDP traffic:

- the built-in Windows Firewall and Base Filtering Engine
- other third-party firewall or network security products
- anti-virus checkers.

This is the case even if the software has no rules configured but is still active.

Where high throughput is required on a particular port, the performance will be improved by disabling the software on that port:

NOTE: The Windows (or any third party) Firewall should be disabled with caution. The network administrator should be consulted before making any changes.



Tuning Recommendations

The following tables provide recommendations for tuning settings for different application characteristics.

Table 33: Throughput Tuning Settings

Tuning Parameter	How?
Firewall	Disable the Base Filter Engine.
Interrupt Moderation	Leave at default (Enabled).
Large Send Offloads	Leave at default (Enabled).
Max Frame Size	Configure to maximum supported by network in Network Adapter's Advanced Properties.
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Offload Checksums	Leave at default.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (PCIe Gen 2.0 or PCIe Gen 3.x) or x16 slot.
Power Saving Mode	Leave at default.
Receive Side Scaling (RSS)	Leave at default.
RSS NUMA Node	Leave at default (All).
TCP Protocol Tuning	Leave at default (install with “Optimize Windows TCP/IP protocol settings for 10G networking” option selected).

Table 34: Latency Tuning Settings

Tuning Parameter	How?
Firewall	Disable the Base Filter Engine.
Interrupt Moderation	Disable in Network Adapter's Advanced Properties.
Interrupt Moderation Time	Leave at default (60µs). This setting is ignored when interrupt moderation is disabled.
Large Receive Offloads	Disable in Network Adapter's Advanced Properties.
Large Send Offloads	Leave at default (Enabled).

Table 34: Latency Tuning Settings

Tuning Parameter	How?
Max Frame Size	Configure to maximum supported by network in Network Adapter's Advanced Properties.
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Offload Checksums	Leave at default (Enabled).
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (PCIe Gen 2.0 or PCIe Gen 3.x) or x16 slot.
Power Saving Mode	Disable C1E and other CPU sleep modes to prevent OS from putting CPUs into lowering power modes when idle.
Receive Side Scaling	Application dependent
RSS NUMA Node	Leave at default (All).
TCP Protocol Tuning	Leave at default (install with “Optimize Windows TCP/IP protocol settings for 10G networking” option selected).
TCP/IP Checksum Offload	Leave at default

5

Solarflare Adapters on VMware

This chapter documents procedures for installation and configuration of Solarflare adapters on VMware® vSphere 2016 ESXi 6.0u3e™ (and later *ux* versions), ESXi 6.5™ and later versions.

- [Native ESXi Driver \(VMkernel API\) on page 151](#)
- [Legacy Driver \(vmklinux API\) on page 151](#)
- [System Requirements on page 151](#)
- [VMware Feature Set on page 153](#)
- [Install Solarflare Drivers on page 155](#)
- [Driver Configuration on page 157](#)
- [Granting access to the NIC from the Virtual Machine on page 158](#)
- [NIC Teaming on page 158](#)
- [Configuring VLANs on page 159](#)
- [Performance Tuning on VMware on page 160](#)
- [VMware ESXi NetQueue on page 160](#)
- [Adapter MTU \(Maximum Transmission Unit\) on page 163](#)
- [Interrupt Moderation \(Interrupt Coalescing\) on page 164](#)
- [TCP/UDP Checksum Offload on page 166](#)
- [TCP Segmentation Offload \(TSO\) on page 166](#)
- [TCP Protocol Tuning on page 167](#)
- [Receive Side Scaling \(RSS\) on page 168](#)
- [Interface Statistics on page 170](#)
- [vSwitch/VM Network Statistics on page 170](#)
- [CIM Provider on page 173](#)
- [Adapter Firmware Upgrade - sfupdate_esxi on page 174](#)
- [Adapter Configuration - sfboot_esxi on page 177](#)
- [ESXCLI Extension on page 179](#)
- [vSphere Client Plugin on page 188](#)
- [Adapter Diagnostic Selftest on page 199](#)

5.1 Native ESXi Driver (VMkernel API)

The Solarflare VMware driver package includes a *native* ESXi driver using the VMware VMkernel API.

- VMkernel is VMWare's new recommended API for network drivers.
- All future Solarflare development will focus on this native driver.

The Solarflare native driver supports the following Solarflare adapters:

- SFN8042M
- SFN8522M, SFN8522 (SFP+)
- SFN8522-Onload (SFP+), SFN8522-Plus (SFP+)
- SFN8542 (QSFP+), SFN8542-Plus (QSFP+)
- SFN8722 (OCP mezzanine adapter)
- X2522, X2522-Plus (10G)
- X2522-25G, X2522-25G-Plus
- X2541, X2542 models.

The Solarflare native driver requires ESXi 6.0u3e (and higher ux versions), 6.5 or 6.7



NOTE: This release of the native driver does not support SR-IOV. Customers wanting to use this feature must use the legacy driver (see [Legacy Driver \(vmklinux API\) on page 151](#)). Support for SR-IOV will be added to the native driver in a future release.

5.2 Legacy Driver (vmklinux API)

There is also a *legacy* driver package, that uses the deprecated *vmklinux* API.

- The older vmklinux API will be removed in future ESXi versions.
- This legacy driver is no longer under active development.

The Solarflare legacy driver supports the following Solarflare adapters:

- SFN5000 series
- SFN6000 series
- SFN7000 series
- SFN8000 series.

The Solarflare legacy driver supports ESXi versions up to and including 6.5.



NOTE: The Solarflare legacy driver supports SR-IOV.

5.3 System Requirements

Refer to [Software Driver Support on page 14](#) for supported VMware host platforms.

5.4 Distribution Packages

VMware drivers, firmware and utilities are available from support@solarflare.com.

Packages for ESXi 6.5 (*and later versions*)

Part Number	Description
SF-118824-LS	<i>Solarflare VMware ESXi 6.5 / 6.7 Native Driver (VIB)</i>
SF-118825-LS	<i>Solarflare VMware ESXi 6.5 / 6.7 Native Driver (Offline Bundle)</i>
SF-120055-LS	<i>Solarflare VMware Utilities CIM Provider for Native Driver</i>
SF-120056-LS	<i>Solarflare vSphere client plugin Windows Installer</i> <i>Available for ESXi 6.5 or later versions.</i>
SF-120054-LS	<i>Solarflare Linux Utilities (32-bit) for use with the ESXi CIM Provider for Native Driver</i>
SF-120773-LS	<i>ESXCLI extensions (VIB)</i>
SF-121528-LS	<i>Solarflare Firmware VIB</i>

Packages for ESXi 6.0-u3e (*and later ux versions*)

Part Number	Description
SF-120732-LS	<i>Solarflare VMware ESXi 6.0 Native Driver (VIB)</i>
SF-120733-LS	<i>Solarflare VMware ESXi 6.0 Native Driver (Offline Bundle)</i>
SF-120055-LS	<i>Solarflare VMware Utilities CIM Provider for Native Driver</i>
SF-120054-LS	<i>Solarflare Linux Utilities (32-bit) for use with the ESXi CIM Provider for Native Driver</i>
SF-120773-LS	<i>ESXCLI extensions (VIB)</i>
SF-121528-LS	<i>Solarflare Firmware VIB</i>



NOTE: To prevent file deletion when the ESXi host is rebooted, files can be copied to a directory created by the user in any host datastore under /vmfs, for example:
`/vmfs/volumes/datastore1/solarflare`

5.5 VMware Feature Set

The table below lists the features available from the VMware host.

Table 35: VMware Host Feature Set

Basic Driver Features	
Jumbo frames	Support for MTUs (Maximum Transmission Units) from 1500 bytes to 9000 bytes. • See Adapter MTU (Maximum Transmission Unit) on page 163
Teaming	Improve server reliability by creating teams on either the host vSwitch, Guest OS or physical switch to act as a single adapter, providing redundancy against single adapter failure. • See NIC Teaming on page 158
Virtual LANs (VLANs)	Support for VLANs on the host, guest OS and virtual switch. • See Configuring VLANs on page 159
VLAN tag insertion	Support offload of vlan tag insertion to hardware (firmware). The NIC must use the full-feature or auto firmware variants. If the firmware variant is not full-feature or auto, vlan tag insertion offload is not available.
FEC	Forward Error Correction employs redundancy in the channel coding as a technique used to reduce bit errors (BER) in noisy or unreliable communications channels. • See ESXCLI Extension on page 179
Fault diagnostics	Support for comprehensive adapter and cable fault diagnostics and system reports. • See CIM Provider on page 173
Interrupt moderation	Coalesce multiple received packets events or transmit completion events into a single interrupt. • See Interrupt Moderation (Interrupt Coalescing) on page 164
Pause frames	Separate control for receive and transmit.
RX/TX ring buffers	Set the adapter RX/TX buffer sizes. • See Adapter RX/TX ring buffer size on page 158
Network core dumping	Transfer core dump file to vCenter Server Appliance after host panic. • See Network Core Dump on page 199 for configuration detail.
Firmware Features	
Port level stats	• See ESXCLI Extension on page 179
Cable Type	• See Adapter Diagnostic Selftest on page 199
PHY Address	• See Adapter Diagnostic Selftest on page 199
Transceiver Type	• See Adapter Diagnostic Selftest on page 199

Table 35: VMware Host Feature Set

Offload Features	
Checksum offload	TCP/UDP over IPv4/IPv6 checksum. <ul style="list-style-type: none"> See TCP/UDP Checksum Offload on page 166
TSO	Support for TCP Segmentation Offload (TSO). <ul style="list-style-type: none"> See TCP Segmentation Offload (TSO) on page 166
Overlay support VXLAN	Support for VXLAN, checksum offload against these packets and RSS support for encapsulated inner layer 3/4 headers.
Overlay support GENEVE	Support for Geneve, checksum offload against these packets and RSS support for encapsulated inner layer 3/4 headers. Not supported on ESXi 6.0.
Performance Features	
NetQueue	Configurable number of NetQueues <ul style="list-style-type: none"> See Driver Configuration on page 157 See VMware ESXi NetQueue on page 160
RSS	Configurable number of RSS queues <ul style="list-style-type: none"> See Driver Configuration on page 157 See Receive Side Scaling (RSS) on page 168
Management Features	
vSphere Client Plugin	Registered with a vCenter Server Appliance, for configuration and management of Solarflare adapters. Not supported on ESXi 6.0. <ul style="list-style-type: none"> See vSphere Client Plugin on page 188
ESXCLI extension	Solarflare extensions to the command line interface. <ul style="list-style-type: none"> See ESXCLI Extension on page 179
Firmware update	Support for Boot ROM and Phy transceiver firmware upgrades for in-field upgradable adapters. The firmware update utility, sfupdate_esxi, is provided through the supplied CIM provider package. See Adapter Firmware Upgrade - sfupdate_esxi on page 174 . Firmware can also be upgraded through the vSphere Client Plugin. See vSphere Client Plugin on page 188 . Firmware can also be upgraded through the extensions command line. See ESXCLI Extension on page 179 .
Adapter hardware and bootROM configuration	Adapter configuration with sfboot. <ul style="list-style-type: none"> See Adapter Configuration - sfboot_esxi on page 177.
Sensors	Read adapter voltage and temperature sensors. <ul style="list-style-type: none"> See Sensors on page 185.

5.6 Install Solarflare Drivers



CAUTION: The Solarflare native ESXi driver is a host driver only. This driver should NOT be installed on a Virtual Machine.

The Solarflare adapter driver on the ESXi host is named **sfvmk**.

Identify an installed driver vib version

```
esxcli software vib list | grep [sfvmk|sfc]  
sfvmk    2.2.0.1000-10EM.650.0.0.4598673  SFC  VMwareCertified   2019-02-05  
Remove an installed driver vib  
esxcli software vib remove --vibname=sfvmk  
To remove the earlier versions of the Solarflare driver:  
esxcli software vib remove --vibname=net-sfc
```



NOTE: When a driver has been removed the ESXi host server must be rebooted.

Install the vib through the host CLI

```
esxcli software vib install -v <absolute PATH to the .vib>  
Installation Result  
Message: The update completed successfully, but the system needs to be  
rebooted for the changes to be effective.  
Reboot Required: true  
VIBs Installed: SFC_bootbank_sfvmk_2.2.0.1000-10EM.650.0.0.4598673  
VIBs Removed:  
VIBs Skipped:
```



NOTE: When a driver has been installed the ESXi host server must be rebooted.

Install the offline bundle

```
esxcli software vib install -d <absolute PATH to the .zip>
```

Identify installed/loaded driver module

```
esxcli system module get -m=sfvmk  
  
Module: sfvmk  
Module File: /usr/lib/vmware/vmkmod/sfvmk  
License: BSD  
Version: 2.2.0.1000-10EM.650.0.0.4598673  
Build Type: release  
Provided Namespaces:  
Required Namespaces: com.vmware.vmkapi@v2_4_0_0  
Containing VIB: sfvmk    VIB Acceptance Level: certified
```

To identify adapter driver and firmware versions

- 1 List adapter interfaces:

```
esxcli network nic list

Name      PCI Device      Driver Admin Status Link Status Speed Duplex
vmnic4   0000:04:00.0  sfvmk    Up        Up       10000  Full
MAC Address          MTU  Description
00:0f:53:29:eb:60  1500  Solarflare SFC9220 10/40G Ethernet Controller
```

- 2 Identify adapter driver/firmware/link status:

```
esxcli network nic get -n vmnic4
```

```
Advertised Auto Negotiation: false
Advertised Link Modes: 1000BaseT/Full
Auto Negotiation: false
Cable Type: DA
Current Message Level: 1
Driver Info:
  Bus Info: 0000:04:00:0
  Driver: sfvmk
  Firmware Version: 7.5.0.1016 rx0 tx0
  Version: 2.2.0.1000
Link Detected: true
Link Status: Up
Name: vmnic4
PHYAddress: 0
Pause Autonegotiate: true
Pause RX: false
Pause TX: false
Supported Ports: FIBRE, DA
Supports Auto Negotiation: false
Supports Pause: true
Supports Wakeon: false
Transceiver:
Virtual Address: 00:50:56:5b:94:f1
Wakeon: None
```

5.7 Driver Configuration

List Adapter Driver Parameters

Run the following command to identify driver module parameters and current values of parameters configurable on the ESXi host:

```
esxcli system module parameters list -m sfvmk
```

Table 36: Driver Module Parameters

Diagnostic Test	Purpose
debugMask	Debug logging bit masks
evqType	Optimize driver for low-latency or throughput performance. EVQ type [0:Auto, 1: Throughput (default), 2: Low latency] Invalid value sets evqtype to the default value. Auto will select a setting based on firmware variant.
vxlanOffload	VXLAN offload [0: Disable, 1: Enable (default)]
geneveOffload	GENEVE offload [0: Disable, 1: Enable (default)]
netQCount	NetQ count (includes defQ) [Min:1 Max:15 Default:8] Invalid value sets netQCount to default value
rssQCount	RSSQ count [Min:1 (RSS disabled) Max:4 Default: RSS disabled] Invalid value sets rssQCount disables RSS

Set Driver Parameters

```
esxcli system module parameters set -m sfvmk --parameter-string="rssQCount=4"
```

The parameter string can also set multiple values in one command:

```
--parameter-string="rssQCount=4 netQCount=8"
```

Use the list command to view current settings.

5.8 Adapter Configuration

Uplink link state

To change/get uplink state:

```
esxcli network nic [up|down] -n <uplink-interface>  
esxcli network nic [up|down] -n vmnic4
```

Adapter RX/TX ring buffer size

Use the following command to get/set adapter RX/TX ring buffers sizes:

```
esxcli network nic ring current set -n vmnic4 -r 2048 -t 1024  
esxcli network nic ring current set -n vmnic4 -r 4096  
esxcli network nic ring current get -n vmnic4  
    RX: 4096  RX Mini: 0  RX Jumbo: 0  TX: 1024
```



NOTE: Changes are not preserved over reboot.

5.9 Granting access to the NIC from the Virtual Machine

Before a guest operating system has access to the Solarflare adapter, the device should be connected to a vSwitch to which the guest also has a connection.

5.10 NIC Teaming

A team allows two or more network adapters to be connected to a virtual switch (vSwitch). The main benefits of creating a team are:

- Increased network capacity for the virtual switch hosting the team.
- Passive failover in the event one of the adapters in the team fails.



NOTE: The VMware host only supports NIC teaming on a single physical switch or stacked switches.

To create a team

- 1 From the host web client, select the **Networking** folder.
- 2 Select the required **vSwitch** under **Networking**.
- 3 **Edit** Settings on the vSwitch.
- 4 Select **NIC Teaming** from the Edit Standard virtual switch settings dialog.

The following options are configurable:

- Load Balancing
- Network Failover Detection
- Notify Switches
- Failover
- Failover Order

Teaming - further reading

Refer to VMware documentation for additional teaming configuration information.

5.11 Configuring VLANs

There are three methods for creating VLANs on VMware ESXi:

- 1** Virtual Switch Tagging (VST)
- 2** External Switch Tagging (EST)
- 3** Virtual Guest Tagging (VGT)

For EST and VGT tagging, consult the documentation for the switch or for the guest OS.

To Configure Virtual Switch Tagging (VST)

With vSwitch tagging:

- All VLAN tagging of packets is performed by the virtual switch, before leaving the VMware ESXi host.
 - The host network adapters must be connected to trunk ports on the physical switch.
 - The port groups connected to the virtual switch must have an appropriate VLAN ID specified.
- 1** From the host web client, select the **Networking** folder.
 - 2** Select **Port Groups** tab to list port groups.
 - 3** Select a **Port Group** and click **Edit Settings**.
 - 4** Enter a valid VLAN ID.
 - a)** VLAN ID 0 (zero) disables VLAN tagging on the port group (EST mode).
 - b)** VLAN ID 4095 enables trunking on the port group (VGT mode).

5.12 Performance Tuning on VMware

Introduction

The Solarflare network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to provide good performance across a broad class of applications.

Install VMware Tools in the Guest Platform

Installing VMware tools will deliver greatly improved networking performance in the guest. When VMware Tools are installed, the guest will see virtual adapters of type **vmxnet3** which is a virtual adapter designed to deliver high performance with minimal I/O overheads in VMs.

To check that VMware Tools are installed:

- 1 From the host web client, select the virtual machine.
- 2 **Edit Settings > VM options > VMware Tools.**

If VMware Tools are not installed/enabled, refer to VMware documentation:

<https://kb.vmware.com/s/article/2004754>

VMware ESXi NetQueue

Solarflare adapters support VMware's NetQueue technology, accelerating network performance in Ethernet virtualized environments. NetQueue is enabled by default in VMKernel releases.

There is usually no reason not to enable NetQueue.



NOTE: VMware NetQueue accelerates receive and transmit traffic.

NetQueue Filtering

NetQueue distributes traffic among physical queues, with each queue having its own ESX thread for packet processing and each thread represents a CPU core.

Traffic is filtered on MAC address and NetQueue will use the adapter outer MAC address meaning that all traffic having the same MAC address is directed to the same queue.

When using VXLAN, packets addressed to multiple VMs will have the same destination MAC. VXLAN also filters on the inner MAC address.

Is NetQueue Enabled

```
esxcli system settings kernel list -o netNetqueueEnabled
```

Name	Type	Description
netNetqueueEnabled	Bool	Enable/Disable NetQueue support.
Configured	Runtime	Default
TRUE	TRUE	TRUE

Configure Number of NetQueues

To configure the Solarflare adapter driver to use NetQueue - specify the number of queues required.

Refer to [List Adapter Driver Parameters on page 157](#) above.

Check the current NetQueue configuration

```
esxcli network nic queue count get
NIC      Tx netqueue count  Rx netqueue count
-----
vmnic4      8          8
vmnic5      8          8
vmnic6      8          8
vmnic7      8          8
```

Binding NetQueue queues and Virtual Machines to CPUs

NetQueue can deliver improved performance when each queue's associated interrupt and the virtual machine are pinned to the same CPU. This is particularly true when workloads with sustained high bandwidth are evenly distributed across multiple virtual machines.

To pin a Virtual Machine to one or more CPUs:

- 1 From the host web client, select the virtual machine.
- 2 **Edit Settings > Virtual Hardware > CPU.**
- 3 In the **Scheduling Affinity** box, enter a comma separated list (or hyphenated range) of CPU(s) to which the virtual machine is to be bound.

Identify NetQueue Interrupts

Use `esxtop` to identify interrupts assigned to the Solarflare adapter. Interrupts are listed in order: the first interrupt will be for the **default** queue, the second interrupt for the queue dedicated to the first virtual machine to have been started, the third interrupt for the queue dedicated to the second virtual machine to have been started, and so on.

```
esxtop
press i
press f
toggle fields B C D (press B, then C then D then Enter key)
```

The following example lists the NetQueues and associated IRQs from vnmic4 when four NetQueues are configured.

```
0x14  VMK vnmic4-intr0
0x15  VMK vnmic4-intr1
0x16  VMK vnmic4-intr2
0x17  VMK vnmic4-intr3
```

Toggle the esxtop fields ABCDEF as required for different views of interrupts and CPU usage.

Number of VMs > number CPU

If there are more virtual machine's than CPUs on the host, optimal performance is obtained by pinning each virtual machine and its associated interrupt to the same CPU.

Number of VMs < number CPU

If there are fewer virtual machines than CPUs, optimal results are obtained by pinning the virtual machine and associated interrupt to two different cores which share an L2 cache.

Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved because it takes fewer packets to send and receive the same amount of data.

Solarflare adapters support frame sizes up to 9000 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

Commands

```
esxcli network vswitch
```

```
Usage: esxcli network vswitch {cmd} [cmd options]
```

Available Namespaces:

```
dvs      Commands to retrieve Distributed Virtual Switch information  
standard  Commands to list and manipulate Legacy Virtual Switches on an  
ESX host.
```

Set/Change MTU

Changing the MTU on the vSwitch will also change the value on the Solarflare uplink interface(s).

```
esxcli network vswitch standard set -m <MTU size> -v <vSwitch name>
```

Verify MTU

```
esxcli network vswitch standard list
```

```
esxcli network vswitch dvs vmware list
```

A change in MTU size on a vSwitch will persist across reboots of the VMware ESXi host.

Check Adapter MTU

To check the MTU size on the adapter uplink:

```
esxcli network nic list
```

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation reduces the number of interrupts generated by the adapter by combining multiple received packet events and/or transmit completion events into a single interrupt.

The interrupt moderation interval is the minimum time (microseconds) between two consecutive interrupts. Coalescing occurs only during the interval. Setting a moderation interval of zero (0) will disable interrupt moderation.

Interrupt moderation settings are **crucial for tuning adapter latency**:

- Increasing the interrupt moderation interval will:
 - generate less interrupts
 - reduce CPU utilization (because there are less interrupts to process)
 - increase latency
 - improve peak throughput.
- Decreasing the interrupt moderation interval will:
 - generate more interrupts
 - increase CPU utilization (because there are more interrupts to process)
 - decrease latency
 - reduce peak throughput.
- Turning off interrupt moderation will:
 - generate the most interrupts
 - give the highest CPU utilization
 - give the lowest latency
 - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits may outweigh the cost of increased CPU utilization.

- Interrupt moderation should be disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation should be enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.



NOTE: The interrupt moderation interval dictates the minimum gap between two consecutive interrupts. It does not mandate a delay on the triggering of an interrupt on the reception of every packet. For example, an interval of 30µs will not delay the reception of the first packet received, but the interrupt for any following packets will be delayed until 30µs after the reception of that first packet.

Commands

```
esxcli network nic coalesce  
Usage: esxcli network nic coalesce {cmd} [cmd options]
```

Available Namespaces:

high Commands to access coalesce parameters for a NIC at high packet rate
low Commands to access coalesce parameters for a NIC at low packet rate

Available Commands:

get	Get coalesce parameters
set	Set coalesce parameters on a nic

Set interrupt moderation

```
esxcli network nic coalesce set -t 30 -n vmnicX
```

CAUTION: Settings do not persist over host reboots.

Get interrupt moderation

```
esxcli network nic coalesce get -n vmnicX
```

Identify interrupt activity with esxtop

Refer to [Identify NetQueue Interrupts on page 161](#) above.

Adaptive Moderation

The adaptive interrupt moderation feature is not currently supported.



TCP/UDP Checksum Offload

Checksum offload moves calculation and verification of TCP and UDP packet checksums to the adapter. The driver by default has checksum offload features enabled.

Commands

```
esxcli network nic cso  
Usage: esxcli network nic cso {cmd} [cmd options]
```

Available Commands:

get	Get checksum offload settings
set	Set checksum offload settings on a nic

Enable Checksum Offload

```
esxcli network nic cso set -e=1 -n=<uplink interface>
```

Disable Checksum Offload

```
esxcli network nic cso set -e=0 -n=<uplink interface>
```

Verify Checksum Offload

```
esxcli network nic cso get -n <uplink interface e.g. vmnic4>  
NIC      RX Checksum Offload  TX Checksum Offload  
vmnic4  on                      on
```

When configuring checksum offload in the guest, consult the relevant Solarflare section for the guest OS, or documentation for the guest OS.

TCP Segmentation Offload (TSO)

TCP Segmentation offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TCP segmentation offload benefits applications using TCP. Enabling TCP segmentation offload will reduce CPU utilization on the transmit side of a TCP connection, and so improve peak throughput, if the CPU is fully utilized.

Since TSO has no effect on latency, it can be enabled at all times. The driver has TSO enabled by default.

Commands

```
esxcli network nic tso  
Usage: esxcli network nic tso {cmd} [cmd options]
```

Available Commands:

get	Get TCP segmentation offload settings
set	Set TCP segmentation offload settings on a nic

Enable TSO

```
esxcli network nic tso set -e=1 -n vmnic4
```

Disable TSO

```
esxcli network nic tso set -e=0 -n vmnic4
```

Verify TSO

```
esxcli network nic tso get -n <uplink interface e.g. vmnic4>
```

NIC	Value
vmnic4	on



NOTE: Non TCP protocol applications will not benefit (but will not suffer) when TSO is enabled.

TCP Large Receive Offload (LRO)

Solarflare sfvmk does not support LRO offload.

TSO and LRO Further Reading

Users should refer to [Understanding TCP Segmentation Offload \(TSO\) and Large Receive Offload \(LRO\) in a VMware environment](#).

TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

Typically it is sufficient to tune just the max buffer value. It defines the largest size the buffer can grow to. Suggested alternate values are max=500000 (1/2 Mbyte). Factors such as link latency, packet loss and CPU cache size all influence the affect of the max buffer size values. The minimum and default values can be left at their defaults minimum=4096 and default=87380.

When tuning the guest TCP stack consult the documentation for the guest operating system.

Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts. RSS is disabled by default.

When RSS is enabled the controller uses multiple receive queues into which to deliver incoming packets. The receive queue selected for an incoming packet is chosen in such a way as to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained.

Each receive queue has a dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores.

When VXLAN or GENEVE overlay encapsulation is enabled, RSS will distribute traffic based on inner layer 3/4 headers.

RSS can be enabled independently of NetQueue i.e. both can be enabled or either can be enabled.

Disable RSS

- On the ESXi host (requires host reboot):

```
esxcli system module parameters set -p rssQCount=1 -m sfvmk
```

Enable RSS

- On the ESXi host (requires host reboot):

```
esxcli system module parameters set -p rssQCount=4 -m sfvmk
```

Identify RSS Queue Count

```
esxcli system module parameters list -m sfvmk | grep rssQCount
```

Also refer to [List Adapter Driver Parameters on page 157](#) above.

Interrupt Balancing

Interrupt (IRQ) balancing in the hypervisor aims to distribute interrupts over available CPU cores based on CPU workload. When setting interrupt affinity to specific CPU cores it is best to disable IRQ balancing.

Commands

```
esxcli system settings kernel  
Usage: esxcli system settings kernel {cmd} [cmd options]
```

Available Commands:

list	List VMkernel kernel settings.
set	Set a VMKernel setting.

Enable Balance

```
esxcli system settings kernel set --setting="intrBalancingEnabled" --  
value="TRUE"
```

Disable Balance

```
esxcli system settings kernel set --setting="intrBalancingEnabled" --  
value="FALSE"
```

Verify Balance

```
esxcli system settings kernel list | grep intrBalancingEnabled
```

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different widths. This is independent of the physical slot size used to connect the adapter. Widths are multiples x1, x2, x4, x8 and x16 lanes:

- PCIe 1.0 (2.5 GT/s - in each direction)
- PCIe 2.0 (5.0 GT/s - in each direction)
- PCIe 3.x (8.0 GT/s - in each direction)

Solarflare Adapters are designed for x8 and x16 lane operation. When PCIe slots are only configured electrically to support x4 lanes, adapters will continue to operate, but at reduced speed.

Memory bandwidth

Many chipsets/CPUs use multiple channels to access main system memory. Maximum memory performance is only achieved when the server can make use of all channels simultaneously. This should be taken into account when selecting the number of DIMMs to populate in the server. Consult server/motherboard documentation for details.

Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

5.13 Interface Statistics

Use the following VMkernel Sys Info Shell command to list adapter statistics:

```
vsish -e cat /net/pNics/<uplink-interface>/stats
```

e.g.

```
vsish -e cat /net/pNics/vmnic4/stats
```

This command generates an extensive list of counters for RX/TX packets, dropped packet counters and per-queue counters.

5.14 vSwitch/VM Network Statistics

- 1 Identify the network port for the VM:

```
net-stats -l
```

PortNum	Type	SubType	SwitchName	MACAddress	ClientName
33554434	4	0	vSwitch0	b0:83:fe:e3:88:56	vmnic0
33554436	3	0	vSwitch0	b0:83:fe:e3:88:56	vmk0
33554437	5	9	vSwitch0	00:0c:29:e7:61:11	vmrhe173

- 2 List vSwitch Port Network Statistics

```
esxcli network port stats get -p 33554437
```

```
Packet statistics for port 33554437
  Packets received: 3955
  Packets sent: 153
  Bytes received: 414779
  Bytes sent: 14294
  Broadcast packets received: 3829
  Broadcast packets sent: 21
  Multicast packets received: 4
  Multicast packets sent: 8
  Unicast packets received: 122
  Unicast packets sent: 124
  Receive packets dropped: 0
  Transmit packets dropped: 0
```

3 Identify the PortNum being used by the Solarflare uplink-interface:

```
net-stats -1
PortNum          Type SubType SwitchName      MACAddress
ClientName
33554434        4     0 vSwitch0          b0:83:fe:e3:88:56  vmnic0
33554436        3     0 vSwitch0          b0:83:fe:e3:88:56  vmk0
33554437        5     9 vSwitch0          00:0c:29:e7:61:11  vmrhe173
50331652        3     0 vdataSW         00:50:56:61:65:f3  vmk1
67108868        4     0 DvsPortset-1   00:0f:53:43:23:f1 vmnic5
```

Solarflare adapter MAC addresses begin **00:0f:53**

4 Port ClientStats:

```
vsish -e get /net/portsets/DvsPortset-1/ports/67108868/clientStats
port client stats {
    pktsTxOK:56363495
    bytesTxOK:3720155568
    droppedTx:0
    pktsTsoTxOK:0
    bytesTsoTxOK:0
    droppedTsoTx:0
    pktsSwTsoTx:0
    droppedSwTsoTx:0
    pktsZeroCopyTxOK:16
    droppedTxExceedMTU:0
    pktsRxOK:275131192
    bytesRxOK:3999992213561
    droppedRx:4676
    pktsSwTsoRx:313
    droppedSwTsoRx:0
    actions:0
    uplinkRxPkts:2740376953
    clonedRxPkts:0
    pksBilled:0
    droppedRxDueToPageAbsent:0
    droppedTxDueToPageAbsent:0
```

5 Port Stats Summary:

```
vsish -e get /net/portsets/DvsPortset-1/ports/67108868/vmxnet3/rxSummary
stats of a vmxnet3 vNIC rx queue {
    LRO pkts rx ok:272133667
    LRO bytes rx ok:3995454566846
    pkts rx ok:275131192
    bytes rx ok:3999992213561
    unicast pkts rx ok:275131186
    unicast bytes rx ok:3999992213201
    multicast pkts rx ok:0
    multicast bytes rx ok:0
    broadcast pkts rx ok:6
    broadcast bytes rx ok:360
    running out of buffers:711
    pkts receive error:0
    1st ring size:256
    2nd ring size:128
```

```
# of times the 1st ring is full:398
# of times the 2nd ring is full:313
fail to map a rx buffer:0
request to page in a buffer:0
# of times rx queue is stopped:0
failed when copying into the guest buffer:0
# of pkts dropped due to large hdrs:0
# of pkts dropped due to max number of SG limits:0
pkts rx via data ring ok:0
bytes rx via data ring ok:0
Whether rx burst queuing is enabled:0
current backend burst queue length:0
maximum backend burst queue length so far:0
aggregate number of times packets are requeued:0
aggregate number of times packets are dropped by PktAgingList:0
```

6 VM Receive Queues

```
vsish -e ls /net/portsets/DvsPortset-1/ports/67108868/vmxnet3/rxqueues/
0/
1/
2/
3/
4/
5/
6/
7/
= 8 Linux VM queues being used by the guest and each has its own RX queue:
```

7 Per Receive-Queue Statistics

Per-receive queue stats are available for each VM receive queue:

```
vsish -e get /net/portsets/DvsPortset-1/ports/67108868/vmxnet3/rxqueues/
<rxqueue-number>/stats
```

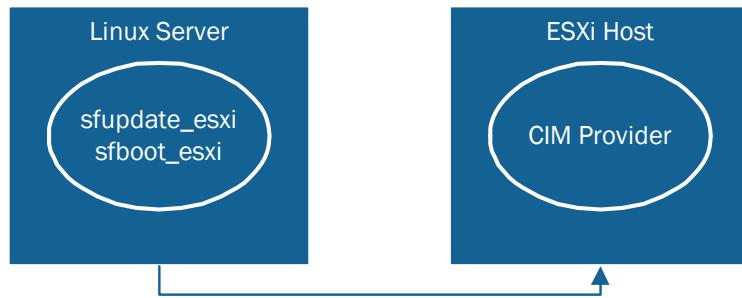
Identify the PortNum, in this example it is 67108868, being used by the Solarflare uplink-interface using the following command:

```
net-stats -l
```

5.15 CIM Provider

The Solarflare Common Information Model (CIM) Provider package is available as a VIB for installation on the ESXi host.

The CIM Provider allows remote access to the ESXi host using CIM transport.



Install CIM Provider

- 1 Remove any existing installed Solarflare CIM .vib

```
esxcli software vib list | grep solarflare
solarflare-cim-provider 2.1-0.19 SLF VMwareAccepted 2019-02-05
```

```
esxcli software vib remove --vibname=solarflare-cim-provider
```

- 2 Copy the CIM VIB package, SF-120055-LS, to a directory on the ESXi host.

- 3 Install the .vib

```
esxcli software vib install -v /vmfs/volumes/datastore1/solarflare/SFC-ESX-solarflare-cim-provider-2.1-0.19.vib
```

Installation Result

Message: The update completed successfully, but the system needs to be rebooted for the changes to be effective.

Reboot Required: true

VIBs Installed: SFC_bootbank_solarflare-cim-provider_2.1-0.19

VIBs Removed:

VIBs Skipped:

NOTE: When a vib has been installed the ESXi host server must be rebooted.



Verify CIM Provider

```
esxcli system wbem provider list
Name           Enabled   Loaded
vmw_solarflare-cim-provider   true     true
sfcb_base      true     true
```

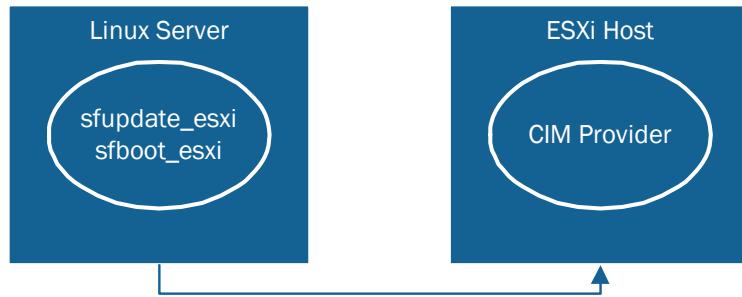
5.16 Adapter Firmware Upgrade - sfupdate_esxi

Adapter firmware can be upgraded using any of the following methods:

- Using the *VCP web plugin* from a VCSA - Refer to [vSphere Client Plugin on page 188](#).
- Locally using the esxcli extension *firmware* command - Refer to [Firmware Images VIB on page 181](#).
- Remotely using *sfupdate_esxi* connected to the CIM Provider.

sfupdate_esxi

The CIM Provider must be installed on the ESXi host. The *sfupdate_esxi* utility connects to the CIM Provider from a remote Linux server.



NOTE: This is the final release version of *sfupdate_esxi*. In future all firmware upgrade will be done with the Solarflare esxcli extension commands or by using the *VCP web plugin*.

See [Firmware Upgrade Examples on page 176](#).

sfupdate_esxi Options

Enter the following command to display available options:

```
./sfupdate_esxi_<version> -?
```

Table 37: sfupdate_esxi Options

Option	Description
<code>-h, --help</code>	Display options and usage
<code>-a, --cim-address=STRING</code>	Address of CIM Server e.g. “https://hostname:5989” “hostname” “hostname:5988” Default protocol is HTTP, default port for HTTP is 5988, default port for HTTPS is 5989
<code>-s, --https</code>	Use HTTPS to access CIM Server (has no effect if you specified protocol in --cim-address parameter)
<code>-u, --cim-user=STRING</code>	CIM Server user name
<code>-p, --cim-password=STRING</code>	CIM Server user password
<code>-n, --cim-namespace=STRING</code>	CIM Provider namespace (solarflare/cimv2 by default)
<code>-i, --interface-name=STRING</code> <code>--controller</code> <code>--bootrom</code> <code>--uefirom</code> <code>--sucfw</code>	Interface name (if not specified, firmware for all interfaces would be processed) Process Controller firmware Process BootROM firmware Process UEFIROM firmware Process support microprocessor (SUC) adapter firmware
<code>-y, --yes</code>	Do not ask for confirmation before updating firmware
<code>-w, --write</code>	Perform firmware update
<code>--force</code>	Force update of the firmware even if the version of the image is lower than or the same as they image already installed - see examples below. If this option is required, but not on the sfupdate command line, the following is displayed: “won’t be applied without --force”
<code>--firmware-url=STRING</code>	URL of firmware image(s) to be used instead of the image included with this version of sfupdate_esxi. Currently support FTP and TFTP - see examples below.
<code>--firmware-path=STRING</code>	Path to firmware image(s) to be used instead of the image included with this version of sfupdate_esxi
<code>--firmware-url-no-local-</code>	Do not try to access firmware images from an URL specified from this tool, just pass URL to CIM provider. Version checks will be disabled; --fw-url-use-cim-transfer cannot be used together with this option
<code>--firmware-url-cim-transfer</code>	Do not pass firmware URL to CIM provider but transfer downloaded firmware images via private CIM methods. Useful when there are issues with ESXi firewall or if URL specified is not available on the ESXi target host

Firmware Upgrade Examples

```
./sfupdate_esxi_v2.1.0.17 --cim-address="https://servername:5989" --write
```

The user will be prompted for the user password.

```
vmnic4 - MAC: 000f53644f10
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter
SUCFW version: 2.1.1.1003
    Available update: 2.1.1.1001 (won't be applied without --force)
UEFIROM version: 2.7.8.5
    Available update: 2.7.5.0 (won't be applied without --force)
BootROM version: 5.2.1.1000
    Available update: 5.2.0.1004 (won't be applied without --force)
Controller version: 7.5.0.1016 rx0 tx0
    Available update: 7.5.0.1009 (won't be applied without --force)
```

*

```
./sfupdate_esxi_v2.1.0.17
--cim-address="servername" --https --cim-user=<user>
--cim-password=<password> -i vmnic5 --write

vmnic5 - MAC: 000f534323f1
NIC model: Solarflare Flareon Ultra 8000 Series 10G Adapter
SUCFW Not Applicable
UEFIROM version: 2.4.4.8
    Available update: 2.4.4.8 (won't be applied without --force)
BootROM version: 5.0.5.1002
    Available update: 5.0.5.1002 (won't be applied without --force)
Controller version: 6.5.1.1023 rx0 tx0
    Available update: 6.5.2.1000
```

Do you want to update Controller firmware on vmnic5? [yes/no]

*

Upgrade firmware using FTP (example)

```
./sfupdate_esxi_2.1.0.17
--cim-address="https://root@10.40.128.17:5989"
--interface-name=vmnic2 --controller --write --force
--firmware-url="ftp://Guest:guest123@10.40.30.20/mcfw.dat"
```

Upgrade firmware using TFTP (example)

```
./sfupdate_esxi_2.1.0.17
--cim-address="https://root@10.40.128.17:5989"
--interface-name=vmnic2 --controller --write --force --yes
--firmware-url=tftp://10.40.128.230/mcfw.dat
```

5.17 Adapter Configuration - sfboot_esxi

sfboot_esxi is a command line utility for configuring Solarflare adapter Boot Manager options including PXE and UEFI booting. sfboot_esxi is an alternative to using the Ctrl+B to access the bootROM agent during server restart.

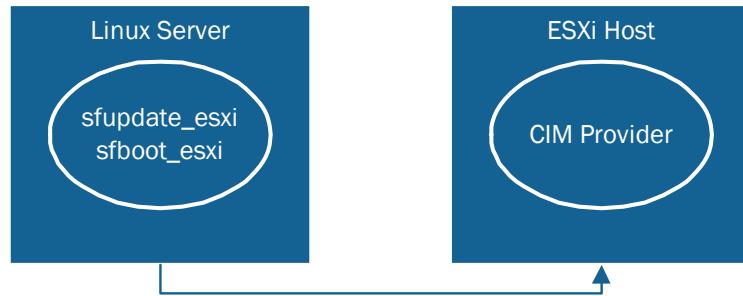
Adapters on the ESXi host can be configured remotely using sfboot_esxi connected to the CIM Provider.

sfboot_esxi

The CIM Provider must be installed on the ESXi host. The sfboot_esxi utility connects to the CIM Provider from a remote Linux server.

This feature requires:

- Solarflare boot configuration utility [v7.6.0] or later
- Solarflare-CIM-provider [2.1.0.19] or later
- Solarflare sfvmk driver [2.2.0.1000] or later



sfboot_esxi Options

```
# ./sfboot_esxi -h
```

For more information about sfboot_esxi options, refer to [Sfboot: Command Usage on page 74](#).

Using sfboot_esxi

Usage:

```
# ./sfboot_esxi -i <interface> -a "https://fully qualified server domain  
name:5989" -u root -p <root password>
```

Example:

```
# ./sfboot_esxi -i vmnic6 -a "https://mserv1.companydomaincom.com:5989"  
-u root -p tester
```

```
Solarflare boot configuration utility [v7.6.0]  
Copyright Solarflare Communications 2006-2018, Level 5 Networks 2002-2005
```

vmnic6:	
Boot image	Option ROM and UEFI
Link speed	Negotiated automatically
Link-up delay time	5 seconds
Banner delay time	2 seconds
Boot skip delay time	5 seconds
Boot type	PXE
Physical Functions on this port	1
PF MSI-X interrupt limit	32
Virtual Functions on each PF	0
VF MSI-X interrupt limit	8
Port mode	Default
Firmware variant	Auto
Insecure filters	Default
MAC spoofing	Default
Change MAC	Default
VLAN tags	None
Switch mode	Default
RX descriptor cache size	32
TX descriptor cache size	16
Total number of VIs	2048
Event merge timeout	1500 nanoseconds



NOTE: A ESXi host server cold reboot is required after changes with sfboot_esxi.

5.18 ESXCLI Extension

Solarflare provide extensions to the VMware esxcli command line interface.

Install

Solarflare esxcli extensions are supplied as a VIB package - see [Distribution Packages on page 152](#) above.

```
esxcli software vib install -v <absolute PATH to the .vib>
```

Identify installed package

```
esxcli software vib list | grep sfvmk  
sfvmkcli      2.2.0.1000-05    SFC    PartnerSupported   2019-02-05
```

The esxcli command will confirm that the Solarflare extensions commands are present:

```
esxcli | grep sfvmk  
  
sfvmk           SFVMK esxcli functionality
```

List extensions commands:

```
esxcli sfvmk
```

Usage: esxcli sfvmk {cmd} [cmd options]

Available Namespaces:

fec	esxcli extension to get/ set FEC mode settings
firmware	esxcli extension to get firmware version and update firmware image
mclog	esxcli extension to get/ set the MC logging enable state
sensor	esxcli extension to get hardware sensor information
stats	esxcli extension to get hardware queue statistics
vfd	esxcli extension to get VFD information

25G Link Speed

Solarflare X2 series adapters automatically detect the link speed and configuration of the link partner (switch) and no configuration is necessary on 25G links.

If the link partner/switch has auto-negotiation (AN) enabled, the adapter will determine this from the AN protocol. If auto-negotiation is disabled, the adapter will instead auto-configure from analyzing the received signal.

Solarflare recommend, in the majority of cases, to configure the 25G link properties on the connected switch port and leave the adapter in its default state. However, the adapter can also be configured manually if this is required.



NOTE: 25G links are only attempted when the DAC cable or transceiver module is rated for 25G operation. 25G links speed will not be attempted when using a standard 10G DAC cable/transceiver.

FEC Configuration

Forward Error Correction employs redundancy in the channel coding as a technique used to reduce bit errors (BER) in noisy or unreliable communications channels. The receiver is able to detect and correct errors without the need for a reverse channel or data re-transmission.



NOTE: FEC can potentially impact latency with an additional error correction overhead of a few hundred nanoseconds.

25G links will auto-negotiate whether to use FEC and what type of FEC to apply on a link.

25G DAC Cables

DAC Cable	FEC Requirement
CA-25G-L up to 5m	requires RS-FEC
CA-25G-S up to 3m	requires either RS-FEC or BASE-R FEC (default is BASE-R)
CA-25G-N up to 3m	can work with RS-FEC, BASE-R FEC or (the default) no FEC.
25G Optical cables do not auto-negotiate FEC and use RS-FEC by default.	

Identify current FEC setting:

```
esxcli sfvmk fec get -n vmnic4
```

```
FEC parameters for vmnic4:  
Configured FEC encodings: None  
Active FEC encoding: None
```

Set/Change FEC:

```
esxcli sfvmk fec set -n vmnic4
```

```
Usage: esxcli sfvmk fec set [cmd options]
```

Description:	
set	Sets FEC mode settings

Cmd options:	
-m --mode=<str>	FEC mode (auto off rs baser [, ...]). (required)
-n --nic-name=<str>	The name of the NIC to configured. (required)

```
esxcli sfvmk fec set -n vmnic4 -m auto
```

FEC parameters for vmnic4 applied

```
esxcli sfvmk fec set -n vmnic4 -m rs,baser
```

Using the above example FEC will try to use RS, and if this fails it will fallback to use BASER.



NOTE: FEC configuration is non-persistent.

Firmware Images VIB

With the firmware images VIB installed on the ESXi host, adapter firmware can be updated using the esxcli firmware command **-d|--default** option.

Firmware Images

A firmware images VIB contains all adapter firmware components for all Solarflare adapters:

- Controller (MAC controller) firmware
- BootROM firmware
- UEFI firmware
- SUC (support microprocessor controller) firmware

Installing the images VIB installs the firmware components on the ESXI host.

Identify installed firmware images VIB

```
esxcli software vib list | grep fw
sfc-fw-images 7.5.0-1019 Solarflare PartnerSupported 2019-02-06
```

Install the firmware images VIB

```
esxcli software vib install -v /<absolute path to the vib>/fw_images.vib
[--no-sig-check]
```

Installation Result

Message: Operation finished successfully.

Reboot Required: false

VIBs Installed: Solarflare_bootbank_sfc-fw-images_7.5.0-1019

VIBs Removed:

VIBs Skipped:

A firmware image file (.dat) can also be copied to the ESXi host and firmware installed from this file using the esxcli firmware extension with **-f|--file-name** option.

Firmware Update Options

set

Sets new firmware image. Either the **-d** or **-f** option must be specified.



NOTE: esxcli does not allow interactive cli. The progress of firmware update is not visible to the user and output is only displayed when all images are successfully updated or there is a failure. It may take a few minutes (or more when there are multiple adapters in the server) before the operation completes.

get

Display current firmware versions.

Option	Description
-d	This option assumes a firmware images VIB is installed.
--default	Used without other options, this will update all firmware, on all Solarflare NICs from the firmware VIB. USE WITH: -n to update all firmware on a particular NIC. -t to update a specific firmware type. -w to overwrite the existing firmware image even if the firmware image in the VIB is the same as the firmware image on the adapter.
-f	Update a specific firmware image from a firmware image file. USE WITH: -t will compare the specified firmware image type [controller suc bootrom uefirom] against the file image type and will FAIL if types do not match . -n is mandatory.
-n --nic-name=<str>	The name of the NIC to configure. NIC name is mandatory with -f --file-name option.
-w --overwrite	Overwrites firmware image even if firmware image version being updated is same as the firmware image on the NIC. This is applicable only with -d --default option.
-t --type=<str>	Firmware image type [controller bootrom uefirom suc]. USE WITH: -d it specifies the firmware image type to be updated. -f it will compare the specified firmware image type against the file image type and will FAIL if types do not match .

Identify current adapter firmware:

```
esxcli sfvmk firmware get [-n <interface e.g. vmnic6>]  
esxcli sfvmk firmware get -n vmnic6  
  
vmnic6 - MAC: 00:0f:53:64:4f:10  
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter  
Controller version: 7.5.0.1016 rx0 tx0  
BOOTROM version: 5.2.1.1000  
UEFI version: 2.7.8.5  
SUC version: 2.1.1.1003
```

Update firmware -t -d

Specify the type of firmware [controller|suc|bootrom|uefirom]. The image is taken from the default firmware bundle VIB on the host.

```
esxcli sfvmk firmware set -n vmnic6 -t=suc -d  
  
vmnic6 - MAC: 00:0f:53:64:4f:10  
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter  
Previous firmware version:  
    SUC version: 2.1.0.0001  
Updated firmware successfully for vmnic6 vmnic7...  
Current firmware version:  
    SUC version: 2.1.1.1003
```

Update firmware -t -f

Specify the type of firmware ([controller|suc|bootrom|uefirom]). The image is taken from the specified firmware image file which must be present on the host.

This will compare the specified type with the image file type and **will FAIL if types do not match**.

```
esxcli sfvmk firmware set -n vmnic4 -t=bootrom -f=/<absolute path to the  
image file>/BOOTROM_2_6_v5.1.0.1005.dat  
  
vmnic4 - MAC: 00:0f:53:43:25:40  
Solarflare Flareon Ultra 8000 Series 10G Adapter  
NIC model: Solarflare Flareon Ultra 8000 Series 10G Adapter  
BOOTROM version: 5.0.7.1000  
Updating firmware...  
Updating bootrom firmware for vmnic4...  
Firmware was successfully updated!
```

Update firmware -d

Updates all firmware components on all adapters from the default firmware bundle on the host. Can also be used with -n to specify a single adapter.

```
esxcli sfvmk firmware set -n vmnic6 -d

vmnic6 - MAC: 00:0f:53:64:4f:10
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter
Previous firmware version:
  SUC version: 2.1.0.1007
Updated firmware successfully for vmnic6 vmnic7...
Current firmware version:
  SUC version: 2.1.1.1003

vmnic6 - MAC: 00:0f:53:64:4f:10
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter
Previous firmware version:
  UEFI version: 2.7.2.10
Updated firmware successfully for vmnic6 vmnic7...
Current firmware version:
  UEFI version: 2.7.8.5

vmnic6 - MAC: 00:0f:53:64:4f:10
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter
Previous firmware version:
  BOOTROM version: 5.2.0.1004
Updated firmware successfully for vmnic6 vmnic7...
Current firmware version:
  BOOTROM version: 5.2.1.1000

vmnic6 - MAC: 00:0f:53:64:4f:10
NIC model: Solarflare XtremeScale X2522 10/25GbE Adapter
Previous firmware version:
  Controller version: 7.4.0.1021 rx0 tx0
Updated firmware successfully for vmnic6 vmnic7...
Current firmware version:
  Controller version: 7.5.0.1016 rx0 tx0
```



NOTE: esxcli does not allow interactive cli. The progress of firmware update is not visible to the user and output is only displayed when all images are successfully updated or there is a failure. It may take a few minutes (or more when there are multiple adapters in the server) before the operation completes.

STATS

The sfvmk extensions **stats** option will generate an extensive list of stats for all network packets types sent/received via the adapter. The generated stats list includes:

- per network packet type counters
- per network packet length counters
- categorized errors packets counters
- categorized dropped packet counters
- virtual adapter packet counters
- FEC corrections
- per NetQueue transmitted/received packet counters

To retrieve NIC stats:

```
esxcli sfvmk stats get -n <interface e.g.vmnic4>
```

Sensors

The **sensors** option, supported from sfvmk extensions version 2.2.0.0014, displays power and temperature readings from Solarflare adapter sensors.

```
esxcli sfvmk sensor get -n vmnic4
```

Sensor Name	Warn Min.	Warn Max.	Fatal Min.	Fatal Max.	Read Value	Sensor State
0.9v power current: mA	0	8500	0	9500	4264	OK
1.2v power current: mA	0	3500	0	5000	1914	OK
0.9v power voltage (at ADC): mV	500	1100	400	1150	1060	OK
ambient temperature: degC	0	75	0	85	35	OK
Port 0 PHY power switch over-current: bool	--	--	--	--	--	OK
Controller die temperature (TDIODE): degC	0	90	0	100	43	OK
Board temperature (back): degC	0	75	0	85	31	OK

MCLOG (MCDI logging)



CAUTION: MCDI logging should be used for debugging purposes only and should be enabled only when advised by Solarflare customer support.

The **mclog** option enables logging of MCDI messages between adapter driver and adapter firmware.

Usage: esxcli sfvmk mclog {cmd} [cmd options]

Available Commands:

get	Gets MC logging state
set	Sets MC logging state to enable/ disable

Description:

set	Sets MC logging state to enable/ disable
-----	--

Cmd options:

-e --enable	Enable/ Disable MC logging (y[es], n[o]) (required)
-n --nic-name=<str>	The name of the NIC to configured. (required)

Enable MCDI logging:

```
esxcli sfvmk mclog set -e Y -n <interface>
Enabled
```

Disable MCDI logging:

```
esxcli sfvmk mclog set -e N -n <interface>
Disabled
```

VPD

The Vital Product Data option identifies the adapter product range, model and serial number data.

```
esxcli sfvmk vpd get -n <interface>
Product Name: Solarflare Flareon Ultra 8000 Series 10G Adapter
[PN] Part number: SFN8522
[SN] Serial number: 852200201000161724100387
[EC] Engineering changes: PCBR2:CCSA2
[VD] Version: [missing]
```

ESXCLI extensions via SSH

As with all esxcli commands, ESXCLI extensions commands can be invoked remotely using SSH from a remote Linux server when SSH is enabled on the ESXi host. The following are example command formats:

```
# ssh <esxi-host-server> esxcli sfvmk fec get -n vmnic4  
# ssh <esxi-host-server> esxcli sfvmk firmware set -d -n vmnic6  
# ssh <esxi-host-server> esxcli sfvmk sensor get -n vmnic4
```

When using SSH to upgrade firmware using the **-f** option, the path to the file is the path on the esxi host server.

ESXCLI extensions via vCLI

With the VMware vCLI package installed on a remote Linux or Windows platform, esxcli commands can be run from the remote machine. Refer to VMware documentation for further vCLI information.

5.19 vSphere Client Plugin

The Solarflare VCP is a HTML5 vSphere client plugin for ESXi 6.5 (and later) allowing the user to manage Solarflare adapters via a web browser connecting to a vCenter Server Appliance hosted on an ESXi platform.

The VCP provides a graphical frontend, via vSphere, to manage Solarflare CIM objects i.e. adapter and driver. The plugin can be installed on a 64bit Windows Server 2012 R2 or Windows Server 2016.

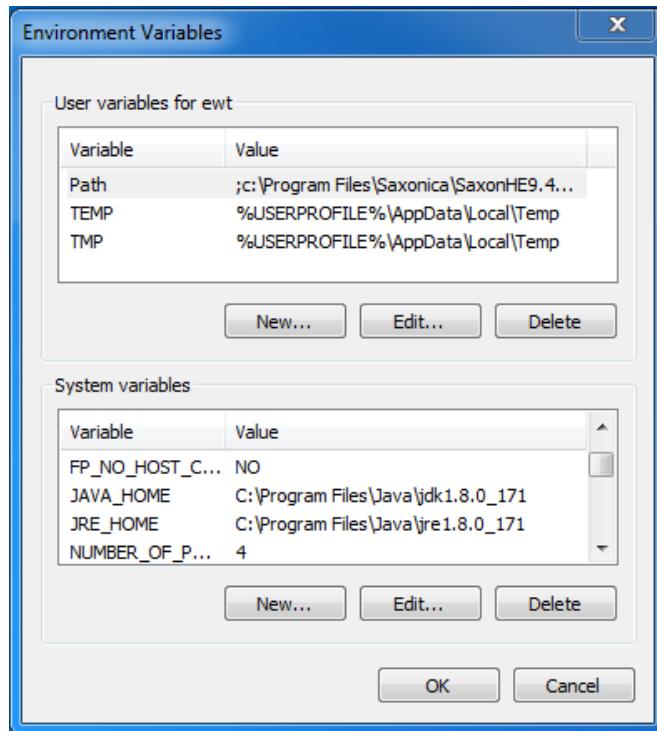
Requirements



NOTE: The Solarflare CIM Provider vib must be installed on all ESXi hosts that will be managed via the vSphere Client Plugin.

The machine from which the plugin installer.msi will run must have the correct value for the **JRE_HOME** environment variable. The Java version must be 1.8 or later.

Check settings from the *Windows Control Panel -> System Properties -> Environment variables* tab:

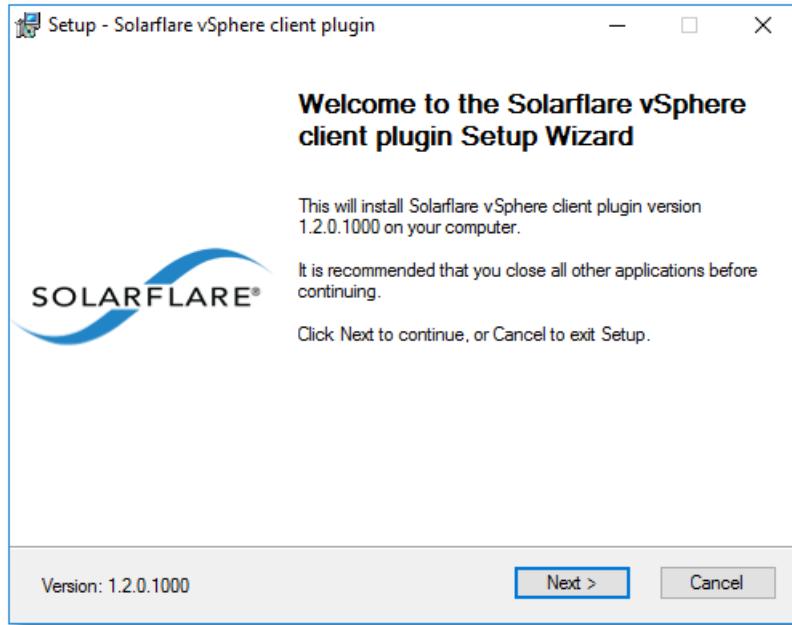


The installer includes all components including Apache Tomcat which is installed on the local server. Changes will be made to the local machine system registry.

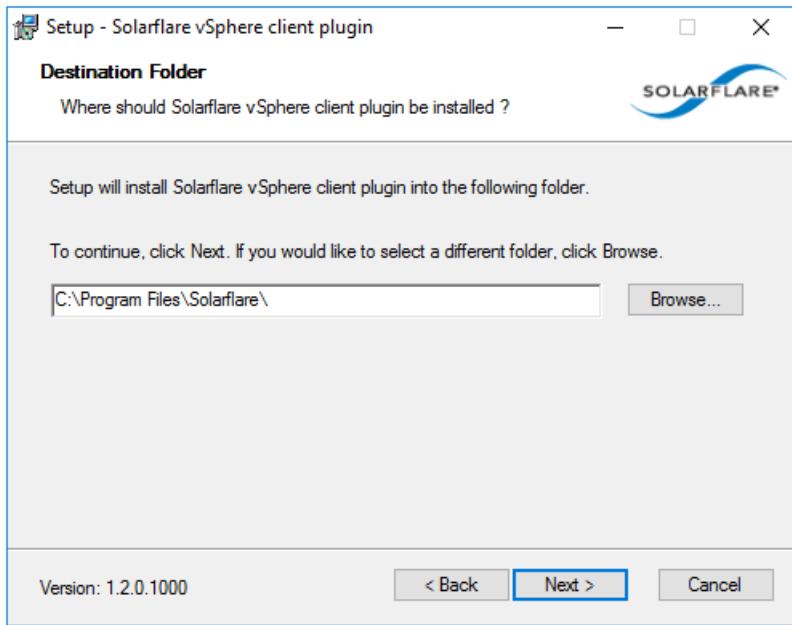
Plugin Installation

Copy the Solarflare vSphere client plugin Windows Installer package SF-120056-LS to the Windows machine where it will be installed.

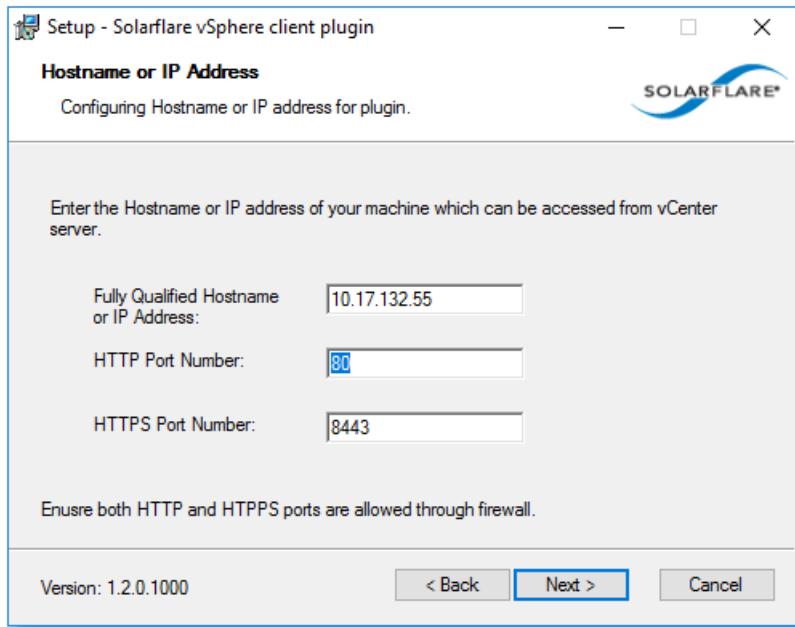
In the *Solarflare VCP-Windows Installer* directory, right click the .msi installer.



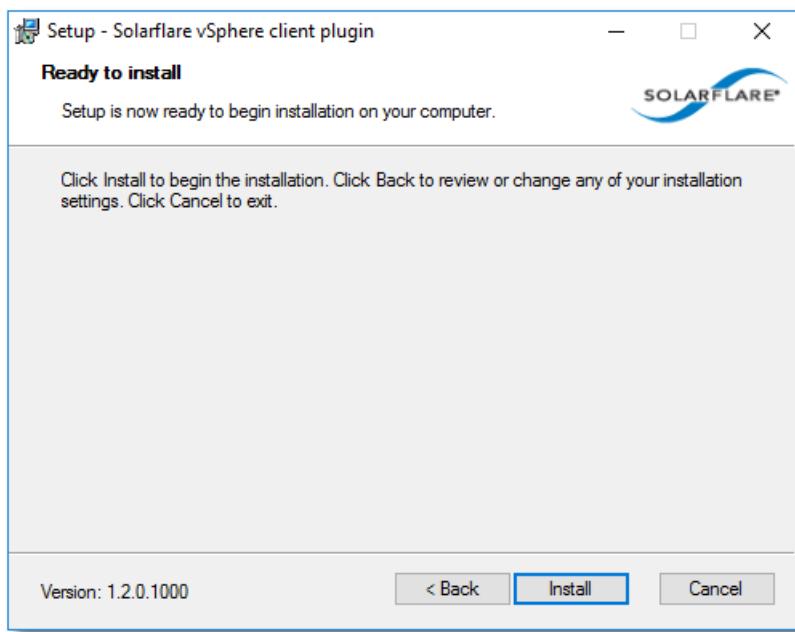
Click **Next**. The Solarflare directory will be created on the local machine. Change the install directory if required and select **Next** to continue.



Enter the hostname or IP address of the local Windows server.

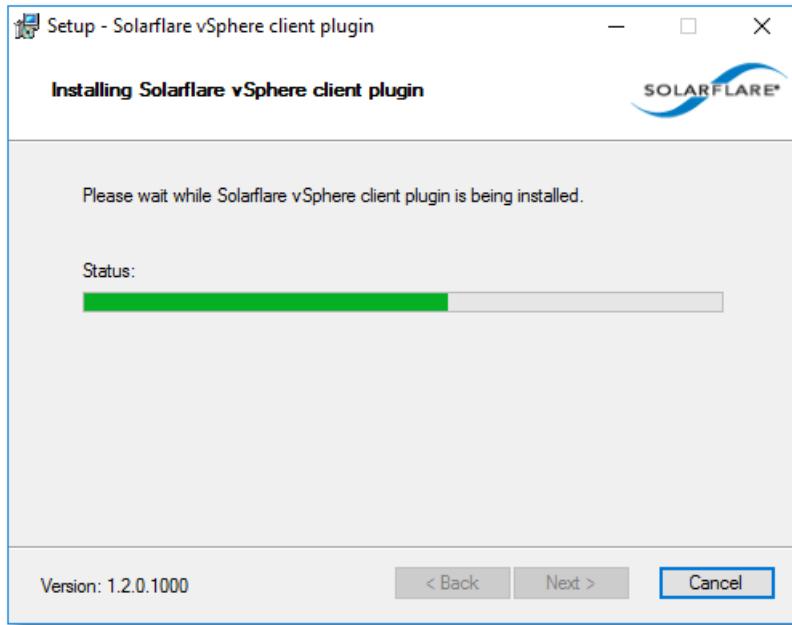


Click **Next** to launch the installation.



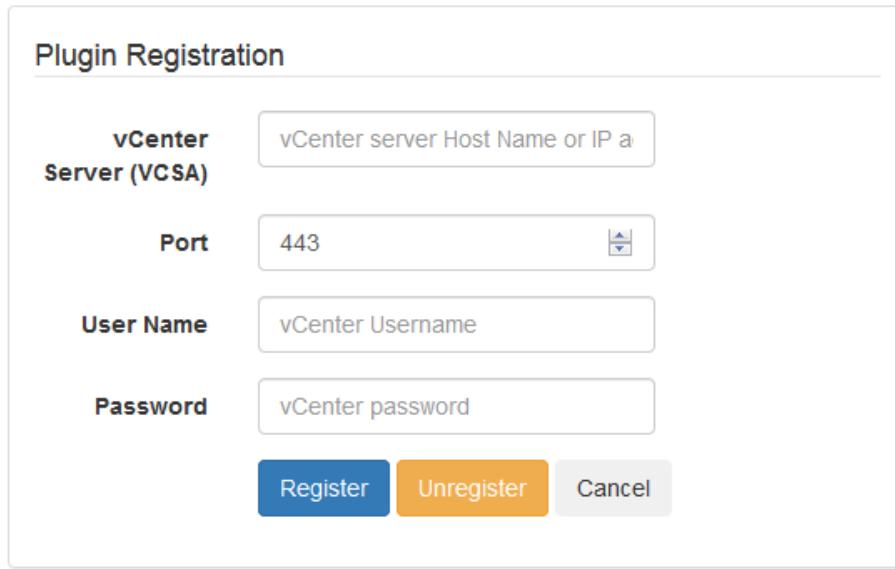
Click **Install** to begin installation.

If installation fails - refer to [Plugin Install Troubleshoot on page 195](#).



When installation is complete, the installer will launch the *Plugin Registration* window to register the plugin with the VCSA.

If a plugin is already installed, registration will first prompt the user to unregister the existing plugin.

A screenshot of a "Plugin Registration" dialog box. It has fields for "vCenter Server (VCSA)", "Port" (set to 443), "User Name", and "Password". At the bottom are "Register", "Unregister", and "Cancel" buttons.

Enter the name of the VCSA and the VCSA user name and password used when the VCSA was created. Select the **Register** button to complete the installation.

A banner message will display the results of the registration process.

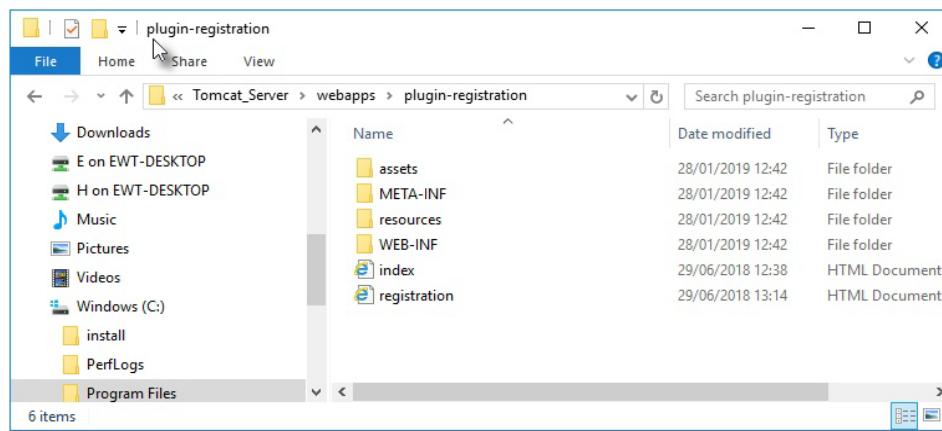
Verify Installation on the local machine.

When the install is complete there will be a Solarflare directory created in the specified location on the local machine, (*C:\Program Files\Solarflare* by default).

Register Later

The VCP plugin can be registered with the VCSA at anytime after the plugin has been installed on the local Windows server. Select the *registration.html* from:

Windows > Program Files > Solarflare > Tomcat_Server > webapps > plugin-registration



Unregister VCP

To unregister the VCP plugin from the VCSA, launch the Plugin Registration dialog window by selecting the '*registration.html*' file (see Register Later) and select the 'Unregister' control.

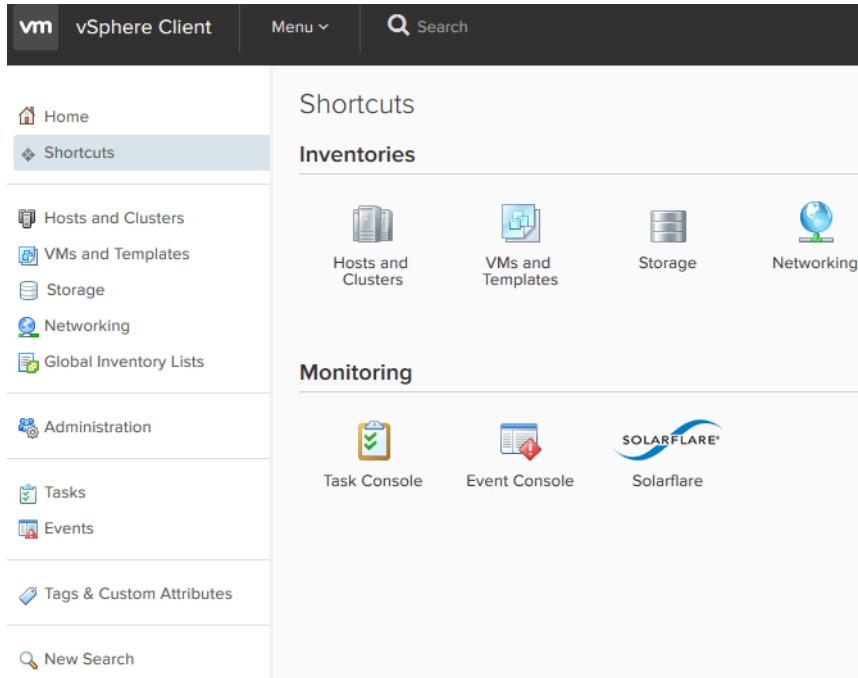
Uninstall VCP

Unregister the VCP plugin from the VCSA before uninstalling the *Solarflare vSphere client plugin installer* via the *Control Panel > Programs > Uninstall a program*.

When uninstalled the VCP components will not be present under the *Windows > Program Files > Solarflare > Tomcat_Server > webapps* directory.

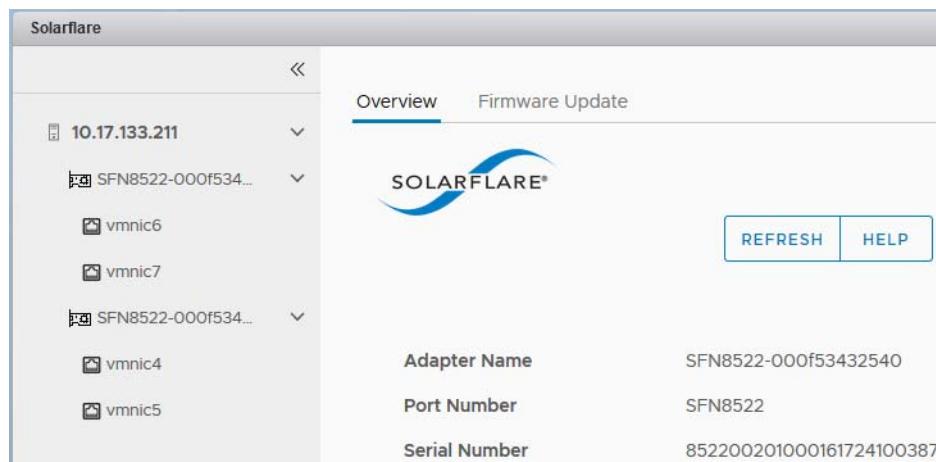
Verify Plugin on the VCSA

When the plugin has been registered with the VCSA, the Solarflare plugin icon will be visible on the *vSphere Client, Menu > Shortcuts* page.



Click the Solarflare icon to load and display the Solarflare plugin.

Select a Solarflare adapter from the adapter list to display configuration menus.



Before hosts or Solarflare adapters are visible in the left pane, a host(s) must be added to the VCSA under a new or existing datacenter. When a datacenter is created, host(s) can be added under this datacenter. Once the host is added, the VMs and Solarflare adapters on the host should be visible to the VCP.

Plugin - Configuration Menus

Host view

- *Overview*
 - Identify the number of adapters present
 - Identify adapter driver version
 - Identify CIM Provider version
 - Identify Solarflare esxcli extensions version
- *Firmware Update*
 - Identify current firmware versions [controller|boot ROM|uefi ROM].
 - Supports per-adapter firmware update
- *Configuration*
 - NetQueue Count
 - RSS Queue Count
 - Driver debug mask
 - Enable disable VXLAN/GENEVE overlay offload

Adapter view

- *Overview*
 - Identify adapter model
 - Identify adapter serial number
- *Firmware Update*
 - Identify current firmware version and firmware version available for update [controller|boot ROM|uefi ROM].

Interface view (vmnic)

- *Overview*
 - Port driver/link status/port speed + hardware vendor information + PCI address
- *Statistics*
 - Timed period stats for packets/bytes sent/received
 - Drop packet count
 - Total multicast/broadcast packets
 - Total receive/transmit error packets

Plugin Install Troubleshoot

When plugin install fails before registration.

If installation fails to complete, generate the installer log file by running the following command from Windows PowerShell or command line, in the directory where the installer.msi resides.

```
msiexec /i Solarflare_VCP_<version>_Installer.msi /l*v  
installer_<version>_log.txt
```

e.g

```
msiexec /i Solarflare_VCP_1.0.5.0_Installer.msi /l*v  
installer_<version>_log.txt
```

(the command should be presented on a single line).

Return [1] the install log file and [2] the failure message/screenshot from the installer by email to support@solarflare.com.

When plugin install fails during registration.

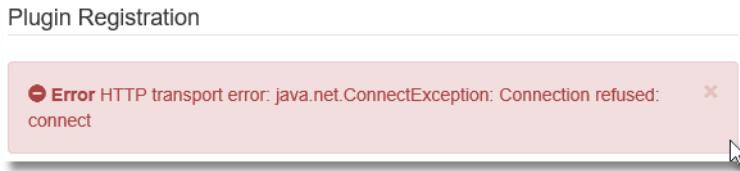
Plugin Information	
Plugin Name	Solarflare
Plugin Key	com.solarflare.vcp
Version	1.147.0
Summary	Solarflare vSphere Client plugin
Plugin URL	https://10.101.10.132:443/solarflare-vcp/solarflare-1.147.0.zip
SSL Thumbprint	D3:4E:92:70:9B:B3:71:FA:C2:3B:DD:E5:91:06:2B:3A:C7:2B:49:2
	7

Check the Plugin URL is valid and accessible from vCenter Server Appliance.

When plugin registration page does not open in the browser.

- This can occur if the Apache Tomcat server is not properly installed or encountered a problem with a previously running service on the same port.
- Check the default browser is working correctly.
- Ensure ports are free and open in the firewall.

When registration fails due to URL - HTTPS connect



If registration cannot complete because there is a problem connecting from the local machine URL, reinstall the client plugin on the local machine and specify the local Windows machine by IP address and not by hostname@domain.

Also confirm that the local Windows server can ping the VCSA IP address and ESXi host machine IP address.

When registration succeeds

Check the correct plugin key (com.solarflare.vcp) and version are present in the vSphere MOB.

Login to the MOB browser:

```
https://<vCenter hostname or IP>/mob
```

The **com.solarflare.vcp** key should be present in the extensionList:

```
extensionList[“com.solarflare.vcp”]
```

Select the Solarflare extension list entry to display the Properties window showing the plugin key version.

Cannot see Shortcuts menu

If the shortcuts menu is not visible after logging in via the ‘vSphere Client (HTML5) - partial functionality’ link, try opening with the ‘vSphere Web Client (Flash)’ link.

Cannot see Solarflare plugin icon.

Restart UI

If the Solarflare plugin icon is not visible after installing and completing a successful install and plugin registration, try the following procedure to stop/restart the vsphere user interface.

- 1 Login to the server hosting the VCSA.
- 2 From the Navigator plane, select the VCSA - make sure the VCSA is powered on.
- 3 From the VCSA Actions menu - open a browser console.
- 4 Login at the console with root access and password.
- 5 Enter the following commands at the Command prompt:
`service-control --status vsphere-ui`

```
service-control --stop vsphere-ui  
service-control --start vsphere-ui
```

Logout and login to VCSA vSphere Client from a web browser. Check if the Solarflare plugin icon is visible under *Menu -> Shortcuts* page.

i.e. <https://<vcse name>:5480>

VCP Installer log

If a UI restart does not display the plugin icon, collect the vsphere-ui log from the VCSA command console:

At the console Command prompt enter 'shell' to access the VCSA file system.

Command> shell

Navigate to the following log:

```
/var/log/vmware/vsphere-ui/logs/vsphere_client_virgo.log
```

Return the log to solarflare support.

Unable to load left menu pane after Refresh in PluginLandingPage

This is an known VMware issue. Return to the *Menu > Shortcuts* drop-down menu and reselect the Solarflare icon.

Error when fetching data after browser inactivity

Following a long period of inactivity it maybe necessary to refresh the browser or logout/login to vCenter to update adapter information.

5.20 Fault Reporting - Diagnostics

sfreport is a command line utility generating a diagnostic log file identifying configuration and statistical data from the VMware host server and installed Solarflare adapters. The Solarflare VMkernel driver (sfvmk) must be installed on the host server.



NOTE: It is advisable to include the sfreport log when reporting issues to Solarflare support.

Run sfreport on local host

Download the document package SF-120088-LS from: support@solarflare.com.

Copy the *sfreport.py* file to a directory on the host server.

To prevent file deletion when the host is rebooted, the file should be copied to a directory created by the user in any host datastore under /vmfs e.g

/vmfs/volumes/datastore1/solarflare

Run the sfreport from the esxcli and return the generated HTML file to support@solarflare.com.

```
python sfreport.py
```

```
sfreport version: v0.1.0
Solarflare Adapters detected..
Please be patient.
SolarFlare system report generation is in progress....
Generated output file: sfreport-2018-03-13-14-15-51.html
```

Run sfreport from remote host

sfreport can also be run from a remote server meeting the following requirements:

- A server running the vSphere Management Assistant (VMA) which has vCLI and is compatible with ESXi6.5. The target host must be reachable from the VMA host.
- A server with vCLI installed and able to reach the target server.

5.21 Network Core Dump

The native driver network core dump feature allows a core dump file to be transferred to a vCenter Server Appliance following a panic of the host.

Configure in the host for each interface:

```
esxcli system coredump network set --interface-name <vmnicN> --server-ip  
<vcSA-server-ip-address>  
esxcli system coredump network set -e 1
```

5.22 Adapter Diagnostic Selftest

1 Identify the Solarflare Adapter uplink(s):

```
esxcli network nic list  
  
vmnic4 0000:82:00.0 sfvmk Up Up 10000 Full  
00:0f:53:43:23:f0 1500 Solarflare SFC 9220 Ethernet Controller  
  
vmnic5 0000:82:00.1 sfvmk Up Up 10000 Full  
00:0f:53:43:23:f1 1500 Solarflare SFC 9220 Ethernet Controller
```

2 Run the adapter diagnostics:

```
esxcli network nic selftest run -n vmnic4
```

Item	Value
Result	Failed
Info	Phy Test : PASSED
Info	Register Test : PASSED
Info	Memory Test : PASSED

The erroneous ‘Failed’ result is a known VMware esxcli issue resolved in ESXi 6.7.

6

Solarflare Adapters on FreeBSD

This chapter covers the following topics on the FreeBSD platform:

- [System Requirements on page 200](#)
- [FreeBSD Platform Feature Set on page 201](#)
- [Installing Solarflare Drivers on page 201](#)
- [Unattended Installation on page 203](#)
- [Configuring the Solarflare Adapter on page 205](#)
- [Setting Up VLANs on page 206](#)
- [FreeBSD Utilities Package on page 207](#)
- [Configuring the Boot ROM with sfboot on page 208](#)
- [Upgrading Adapter Firmware with sfupdate on page 214](#)
- [Performance Tuning on FreeBSD on page 216](#)
- [Module Parameters on page 226](#)
- [Kernel and Network Adapter Statistics on page 228](#)

6.1 System Requirements

Refer to [Software Driver Support on page 14](#) for details of supported FreeBSD distributions.



NOTE: FreeBSD includes a previous version of the Solarflare adapter driver that does not support all features of this version. To update the supplied driver, see [Installing Solarflare Drivers on page 201](#).

6.2 FreeBSD Platform Feature Set

Table 38 lists the features supported by Solarflare adapters on FreeBSD.

Table 38: FreeBSD Feature Set

Jumbo frames	Support for MTUs (Maximum Transmission Units) to 9000 bytes. <ul style="list-style-type: none">• See Configuring Jumbo Frames on page 206
Task offloads	Support for TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. <ul style="list-style-type: none">• See Configuring Task Offloading on page 206
Receive Side Scaling (RSS)	Support for RSS multi-core load distribution technology. <ul style="list-style-type: none">• See Receive Side Scaling (RSS) on page 221
Virtual LANs (VLANs)	Support for multiple VLANs per adapter. <ul style="list-style-type: none">• See Setting Up VLANs on page 206
PXE and booting	Support for diskless booting to a target operating system via PXE or UEFI boot. <ul style="list-style-type: none">• See Configuring the Boot ROM with sfboot on page 208• See Solarflare Boot Manager on page 285
Firmware updates	Support for Boot ROM, PHY transceiver and adapter firmware upgrades. <ul style="list-style-type: none">• See Upgrading Adapter Firmware with sfupdate on page 214

6.3 Installing Solarflare Drivers

The FreeBSD drivers for Solarflare are available in a source package.

- A package is available for FreeBSD 10.0 and 10.1:
 - this package might perform correctly with other FreeBSD kernels, but has not been tested with them
 - for further details see the *Release Notes*.



NOTE: The Solarflare adapter should be physically installed in the host computer before you attempt to install drivers. You must have root permissions to install the adapter drivers.

This source can be used:

- To compile and install a driver on a development machine.
- The development machine must have the following installed:
- development tools
 - the *ports* system, and its Makefiles
 - the kernel source.
- To create a binary driver package, for installing on other target machines.

A target machine:

- must have the same kernel and architecture as the development machine that built the package
- does not require any of the development tools or source.

To install the driver, use `pkg add` or `pkg_add`.

The following instructions assume that the source package has been downloaded to the `/tmp` directory.

- 1 Ensure you are the root user. If not:

```
su -
```

- 2 To avoid using the previous driver that is distributed with the OS, rename it:

```
mv /boot/kernel/sfxge.ko /boot/kernel/sfxge.ko_default
```

- If desired, it can instead be removed:

```
rm /boot/kernel/sfxge.ko
```

- 3 Unpack the downloaded source:

```
cd /tmp  
tar xvf sfxge-freebsd-<version_no>.txz
```

For example:

```
cd /tmp  
tar xvf sfxge-freebsd-4.5.3.1002.txz
```

- 4 Change directory into the source:

```
cd sfxge-freebsd-<version_no>
```

For example:

```
cd sfxge-freebsd-4.5.3.1002
```

- 5 Build the package:

```
make package
```

- If you do not have the ports system installed you will see this error:

The ports system must be installed first.

- You can install the ports system by running:

```
portsnap fetch extract
```

6 Install the package:

- ```
make install
```
- the driver (for use on this machine) is installed in /boot/modules/sfxge.ko
  - the binary driver package (for use on other machines) is installed in the build directory in work/package/sfxge-kmod-<version\_no>.txz, and in /usr/ports/packages/All/sfxge-kmod-<version\_no>.txz

**7** Load the driver:

```
kldload /boot/modules/sfxge.ko
```

For information on configuring this network interface see [Configuring the Solarflare Adapter on page 205](#).

## 6.4 Unattended Installation

Unattended installations of FreeBSD can be performed by PXE booting over the network, and using the bsdinstall command. Set this up as follows:

- Ensure that DHCP is available, with PXE boot options.
- Ensure that a TFTP server is available.
- Ensure that a FreeBSD server is available.

This is required only to generate the FreeBSD PXE boot image:

- Download the mfsbsd utility (available from <http://mfsbsd.vx.sk/>).  
Install it on the FreeBSD server.<sup>1</sup>
- Download the ISO image for the required FreeBSD release.  
Mount this boot image on the FreeBSD server.
- Configure and customize the boot image.  
Change any PXE boot settings as necessary.  
Modify the /etc/installerconfig file in the boot image to add any post-install tasks.
- Use the mfsbsd utility to build the PXE boot image, using the modified boot image as source.
- Copy the PXE boot image to the TFTP server  
Add a FreeBSD option to its pxelinux boot menu.
- PXE boot the target server and select the FreeBSD image.
- The FreeBSD server that was used to generate the FreeBSD PXE boot image can now be re-used.

---

1. The mfsbsd utility runs only under FreeBSD.

FreeBSD install and booting is documented as follows:

- For information on booting a FreeBSD system over the network, see:  
<https://www.freebsd.org/cgi/man.cgi?query=diskless&sektion=8>
- For general information on using bsdinstall, see:  
<https://www.freebsd.org/doc/handbook/bsdinstall.html>
- For a reference description of the bsdinstall command, see:  
<https://www.freebsd.org/cgi/man.cgi?query=bsdinstall&sektion=8>  
 especially the *SCRIPTING* section.

**Table 39** shows an example time line for an unattended installation.

**Table 39: Installation Stages**

| In Control               | Stages of Boot                                            | Setup needed                                                                                |
|--------------------------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------|
| BIOS                     | PXE code on the adapter runs.                             | Adapter must be in PXE boot mode. See <a href="#">Solarflare Boot Manager on page 285</a> . |
| SF Boot ROM (PXE)        | DHCP request from PXE (SF Boot ROM).                      | DHCP server filename and next-server options.                                               |
| SF Boot ROM (PXE)        | TFTP request for filename to next-server, e.g. pxelinux.0 | TFTP server.                                                                                |
| pxelinux                 | TFTP retrieval of pxelinux configuration.                 | pxelinux configuration on TFTP server.                                                      |
| pxelinux                 | TFTP menu retrieval of FreeBSD kernel image.              | pxelinux configuration<br>Kernel, kernel command                                            |
| FreeBSD kernel/installer | Installer retrieves configuration.                        | FreeBSD image                                                                               |
| Installation occurs      | Machine reboots                                           | /etc/installerconfig file                                                                   |
| Target FreeBSD kernel    | kernel reconfigures network adapters.                     | DHCP server.                                                                                |

## 6.5 Configuring the Solarflare Adapter

The drivers will be loaded as part of the as part of the installation. However the adapter will not be configured (adding IP address and netmask).

Each Solarflare network adapter interface will be named `sfxge<n>` where `<n>` is a unique identifier. There will be one interface per physical port on the Solarflare adapter.

To configure the interface and bring it up to allow data to pass, enter the following:

```
ifconfig sfxge<n> inet <IPv4 address> netmask <netmask> up
```

This configures the interface and initializes it with the `up` command.



**NOTE:** This method of configuring is temporary. If you reboot your computer the settings will be lost. To make these settings permanent, create entries in the configuration file as described below.

### Using IPv6

To configure using IPv6, create an IPv6 interface `sfxge<n>` interface with a link local IPv6 address by entering:

```
ifconfig sfxge<n> inet6 <IPv6 address> prefixlen <IPv6 prefix length>
```

This uses automatic link-local address configuration, which is enabled by default in FreeBSD. It will give an IPv6 interface name of `sfxge<n>:1`

### Using a Configuration File with IPv4

Configuration is set in the `/etc/rc.conf` file. There are three options with IPv4:

- Using a static IPv4 address. To use this option, add:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask>"
```

- Using a hostname. To use this option, add:

```
ifconfig_sfxge<n>="inet <hostname>"
```

and modify `/etc/hosts` and `/etc/netmasks`

- Using DHCP. To use this option, add:

```
ifconfig_sfxge<n>="DHCP"
```

### Using Configuration files with IPv6

Configuration is set in the `/etc/rc.conf` file:

- For automatic configuration by StateLess Address AutoConfiguration (SLAAC), add:

```
ifconfig_sfxge<n>_ipv6="inet6 accept_rtadv"
```

## Configuring Task Offloading

Solarflare adapters support transmit (Tx) and receive (Rx) checksum offload, as well as TCP segmentation offload. To ensure maximum performance from the adapter, all task offloads should be enabled, which is the default setting on the adapter. For more information, see [Performance Tuning on FreeBSD on page 216](#).

## Configuring Jumbo Frames

Solarflare adapters support a frame size (MTU) from 1500 bytes to 9000 bytes.

The default maximum driver MTU size is 1500 bytes. For example, to set a new frame size (MTU) of 9000 bytes, enter the following command:

```
ifconfig sfxge<n> inet mtu 9000
```

To view the current MTU, enter:

```
ifconfig sfxge<n>
sfxge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
...
```

If you want to have an MTU configured when the interface is brought up, add an `mtu` parameter to the single line of interface configuration data in the `/etc/rc.conf` file. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> mtu <MTU size>"
```

## 6.6 Setting Up VLANs

VLANs offer a method of dividing one physical network into multiple broadcast domains. In enterprise networks, these broadcast domains usually match with IP subnet boundaries, so that each subnet has its own VLAN. The advantages of VLANs include:

- Performance
- Ease of management
- Security
- Trunks
- You don't have to configure any hardware device, when physically moving your server to another location.

To have a single interface exist on multiple VLANs (if the port on the connected switch is set to “trunked” mode) see the following documentation:

[http://people.freebsd.org/~arved/vlan/vlan\\_en.html](http://people.freebsd.org/~arved/vlan/vlan_en.html)

## 6.7 FreeBSD Utilities Package

The Solarflare FreeBSD Utilities package is supplied as a source package or a 64 bit binary package, and is available from <https://support.solarflare.com/>. It contains the following utilities:

**Table 40: Utilities Package**

| Utility File | Description                                                                                                                     |
|--------------|---------------------------------------------------------------------------------------------------------------------------------|
| sfupdate     | A command line utility that contains an adapter firmware version which can update Solarflare adapter firmware.                  |
| sfboot       | A command line utility for configuring Solarflare adapter Boot ROM options, including PXE and UEFI booting.                     |
| sfreport     | A command line utility that generates a diagnostic log file providing diagnostic data about the server and Solarflare adapters. |

By default, sfboot and sfupdate are installed to /usr/local/sbin, and sfreport is installed to /usr/local/bin.

### Building and installing the source package

To build and install the source package:

- 1 Unpack the source package:

```
tar -xf <source package name>
```

- 2 Go to its directory:

```
cd <source package dir>
```

- 3 Build and install a binary package from the source:

```
make install
```

Alternatively, to build and install in separate steps:

- a) Build a binary package from the source:

```
make package
```

- b) Install the resulting binary package:

```
pkg install ./work/pkg/sfutils-<version>.txz
```

### Installing the 64 bit binary package

- 1 Install the binary package:

```
pkg install <path to package file>
```

## 6.8 Configuring the Boot ROM with sfboot

- [Sfboot: Command Usage on page 208](#)
- [Sfboot: Command Line Options on page 208](#)
- [Sfboot: Examples on page 214](#)

Sfboot is a command line utility for configuring Solarflare adapter Boot ROM options, including PXE and UEFI booting. Using sfboot is an alternative to using **Ctrl + B** to access the Boot Rom agent during server startup.

See [Solarflare Boot Manager on page 285](#) for more information on the Boot Rom agent.

### Sfboot: Command Usage

The general usage for sfboot is as follows (as root):

```
sfboot [---adapter=sfxge<n>] [options] [parameters]
```

Note that without --adapter, the sfboot command applies to all adapters that are present in the target host.

The format for the parameters are:

```
<parameter>=<value>
```

### Sfboot: Command Line Options

[Table 41](#) lists the options for sfboot, [Table 42](#) lists the available global parameters, and [Table 43](#) lists the available per-adapter parameters.

**Table 41: Sfboot Options**

| Option                     | Description                                                                                                                                                                   |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -h, --help                 | Displays command line syntax and provides a description of each sfboot option.                                                                                                |
| -V, --version              | Shows detailed version information and exits.                                                                                                                                 |
| -v, --verbose              | Shows extended output information for the command entered.                                                                                                                    |
| -s, --silent               | Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently. |
| --log <filename>           | Logs output to the specified file in the current folder or an existing folder. Specify --silent to suppress simultaneous output to screen, if required.                       |
| --computer <computer_name> | Performs the operation on a specified remote computer. Administrator rights on the remote computer is required.                                                               |

**Table 41: Sfboot Options**

| Option                    | Description                                                                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --list                    | <p>Lists all available Solarflare adapters. This option shows the adapter's ID number, ifname and MAC address.</p> <p>Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.</p>                                  |
| -d, --adapter =<sfxge<n>> | <p>Performs the action on the identified Solarflare network adapter. The adapter identifier sfxge can be the adapter ID number, ifname or MAC address, as output by the --list option. If --adapter is not included, the action will apply to all installed Solarflare adapters.</p>                                                |
| --clear                   | <p>Resets all options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in <a href="#">Table 42</a>) are not reset. Note that --clear can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.</p> |

The following global parameters in [Table 42](#) are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

**Table 42: Sfboot Global Parameters**

| Parameter                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| boot-image=<br>all optionrom uefi disabled               | <p>Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the --adapter option has been specified.</p>                                                                                                                                                                                                                                                              |
| port-mode=refer to <a href="#">Port Modes on page 38</a> | <p>Configure the port mode to use. This is for SFN7000 and SFN8000 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent.</p> <p>Refer to <a href="#">Port Modes on page 38</a></p> <p>Changes to this setting with sfboot require a cold reboot to become effective. MAC address assignments may change after altering this setting.</p> |

**Table 42: Sfboot Global Parameters**

| Parameter                                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>firmware-variant=full-feature ultra-low-latency capture-packed-stream auto</code> | <p>Configure the firmware variant to use. This is for SFN7000 and SFN8000 series adapters only:</p> <ul style="list-style-type: none"> <li>the SFN7002F adapter is factory set to full-feature</li> <li>all other adapters are factory set to auto.</li> </ul> <p>Default value = auto - means the driver will select a variant that meets its needs:</p> <ul style="list-style-type: none"> <li>the VMware driver always uses full-feature</li> <li>otherwise, ultra-low-latency is used.</li> </ul> <p>The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features). It is recommended that Onload customers use the ultra-low-latency variant.</p> |
| <code>insecure-filters=enabled disabled</code>                                          | If enabled bypass filter security on non-privileged functions. This is for SFN7000 and SFN8000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement when using Onload or when using bonded interfaces.                                                                                                                                                                                                                                                                                                                                                     |
| <code>mac-spoofing=enabled disabled</code>                                              | <p>If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000 and SFN8000 series adapters only.</p> <p>The default is disabled.</p> <p>Changes to this setting with sfboot require a cold reboot to become effective.</p>                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>rx-dc-size=8 16 32 64</code>                                                      | <p>Specifies the size of the descriptor cache for each receive queue. This is for SFN7000 and SFN8000 series adapters only. The default is:</p> <ul style="list-style-type: none"> <li>16 if the port-mode supports the maximum number of connectors for the adapter.</li> <li>32 if the port-mode supports a reduced number of connectors.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                |

**Table 42: Sfboot Global Parameters**

| Parameter                                    | Description                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tx-dc-size=8 16 32 64                        | <p>Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000 and SFN8000 series adapters only. The default is:</p> <ul style="list-style-type: none"> <li>• 32 if the port-mode supports the maximum number of connectors for the adapter.</li> <li>• 64 if the port-mode supports a reduced number of connectors.</li> </ul> |
| vi-count=<vi count>                          | <p>Sets the total number of virtual interfaces that will be available on the NIC.</p>                                                                                                                                                                                                                                                                       |
| event-merge-timeout=<timeout in nanoseconds> | <p>Specifies the timeout in nanoseconds for RX event merging. A timeout of 0 means that event merging is disabled.</p>                                                                                                                                                                                                                                      |

The following per-adapter parameters in [Table 43](#) are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

**Table 43: Sfboot Per-adapter Parameters**

| Parameter                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| link-speed=auto 10g 1g 100m | <p>Specifies the network link speed of the adapter used by the Boot ROM. The default is auto. On the 10GBASE-T adapters, auto instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, auto instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link.</p> <p>auto Auto-negotiate link speed (default)<br/>         10G 10G bit/sec<br/>         1G 1G bit/sec<br/>         100M 100M bit/sec</p> |

**Table 43: Sfboot Per-adapter Parameters**

| Parameter                                                 | Description                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>linkup-delay=&lt;delay time in seconds&gt;</code>   | Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds. |
| <code>banner-delay=&lt;delay time in seconds&gt;</code>   | Specifies the wait period for Ctrl-B to be pressed to enter adapter configuration tool.<br><br><code>&lt;delay time in seconds&gt; = 0-256</code>                                                                                                                                                                                                                   |
| <code>bootskip-delay=&lt;delay time in seconds&gt;</code> | Specifies the time allowed for Esc to be pressed to skip adapter booting.<br><br><code>&lt;delay time in seconds&gt; = 0-256</code>                                                                                                                                                                                                                                 |
| <code>boot-type=pxe uefi disabled</code>                  | Sets the adapter boot type – effective on next boot.<br><br>pxe – PXE (Preboot eXecution Environment) booting<br>disabled – Disable adapter booting                                                                                                                                                                                                                 |
| <code>pf-count=&lt;pf count&gt;</code>                    | This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter.<br><br>Changes to this setting with sfboot require a cold reboot to become effective. MAC address assignments may change after altering this setting.                                                                                      |
| <code>msix-limit=8 16 32 64 128 256 512 1024</code>       | Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32.<br><br>Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.                                                                                                                   |
| <code>sriov=enabled disabled</code>                       | Enable SR-IOV support for operating systems that support this. Not required on SFN7000 or SFN8000 series adapters.                                                                                                                                                                                                                                                  |

**Table 43: Sfboot Per-adapter Parameters**

| Parameter                                                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>vf-count=&lt;vf count&gt;</code>                                            | <p>The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port. SFN7000 and SFN8000 series adapters have a total limit of 2048 interrupts. Earlier adapters support a total limit of 127 virtual functions per port and a total of 1024 interrupts.</p> <p>Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured.</p> <p>Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count.</p> <p>The sriov parameter is implied if vf-count is greater than zero.</p> <p>Changes to this setting with sfboot require a cold reboot to become effective.</p> |
| <code>vf-msix-limit=1 2 4 8 16 32 64 128 256</code>                               | The maximum number of interrupts a virtual function may use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>pf-vlans=&lt;tag&gt;[,&lt;tag&gt;[,...]] none</code>                        | Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>switch-mode=default sriov partitioning partitioning-with-sriov pfiov</code> | <p>Specifies the mode of operation that the port will be used in:</p> <p><code>default</code> - single PF created, zero VFs created.</p> <p><code>sriov</code> - SR-IOV enabled, single PF created, VFs configured with <code>vf-count</code>.</p> <p><code>partitioning</code> - PFs configured with <code>pf-count</code>, VFs configured with <code>vf-count</code>. See <a href="#">NIC Partitioning on page 59</a> for details.</p> <p><code>partitioning-with-sriov</code> - SR-IOV enabled, PFs configured with <code>pf-count</code>, VFs configured with <code>vf-count</code>. See <a href="#">NIC Partitioning on page 59</a> for details.</p> <p><code>pfiov</code> - PFIOV enabled, PFs configured with <code>pf-count</code>, VFs not supported.</p> <p>Changes to this setting with sfboot require a cold reboot to become effective.</p>    |

## Sfboot: Examples

- Show the current boot configuration for all adapters:

```
sfboot
```

```
Solarflare boot configuration utility [v3.0.5]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
```

```
sfxge0:
```

|                       |          |
|-----------------------|----------|
| Boot image            | Disabled |
| MSI-X interrupt limit | 32       |

```
sfxge1:
```

|                       |          |
|-----------------------|----------|
| Boot image            | Disabled |
| MSI-X interrupt limit | 32       |

- List all Solarflare adapters installed on the localhost:

```
sfboot --list
```

```
Solarflare boot configuration utility [v3.0.5]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
sfxge0 - 00-0F-53-01-38-40
sfxge1 - 00-0F-53-01-38-41
```

## 6.9 Upgrading Adapter Firmware with sfupdate

### To Update Adapter Firmware

Reinstall the `sfutils` package, as described in [FreeBSD Utilities Package on page 207](#).

### Sfupdate: Command Usage

The general usage for `sfupdate` is as follows (as root):

```
sfupdate [--adapter=sfxge<n>] [options]
```

where:

- `sfxge<n>` is the interface name of the Solarflare adapter you want to upgrade.
- option is one of the command options listed in [Table 44](#).

The format for the options are:

```
--<option>=<parameter>
```

Running the command `sfupdate` with no additional parameters will show the current firmware version for all Solarflare adapters and whether the firmware within `sfupdate` is more up to date. To update the firmware for all Solarflare adapters run the command `sfupdate --write`

Solarflare recommend that you use `sfupdate` in the following way:

- 1 Run `sfupdate` to check that the firmware on all your adapters are up to date.

**2** Run sfupdate --write to update the firmware on all adapters.

## Sfupdate: Command Line Options

[Table 44](#) lists the options for sfupdate.

**Table 44: Sfupdate Options**

| Option                 | Description                                                                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -h, --help             | Shows help for the available options and command line syntax.                                                                                                                                                                                                                                    |
| -v, --verbose          | Enable verbose output mode.                                                                                                                                                                                                                                                                      |
| -s, --silent           | Suppress all output except for errors. Useful for scripts.                                                                                                                                                                                                                                       |
| -V, --version          | Display version information and exit.                                                                                                                                                                                                                                                            |
| -i, --adapter=sfxge<n> | Specifies the target adapter when more than one adapter is installed in the machine.<br>sfxge<n> = Adapter ifname or MAC address (as obtained with --list).                                                                                                                                      |
| --list                 | Shows the adapter ID, adapter name and MAC address of each adapter installed in the machine.                                                                                                                                                                                                     |
| --write                | Re-writes the firmware from the images embedded in the sfupdate tool.<br>To re-write using an external image, specify<br>--image=<filename> in the command.<br>--write fails if the embedded image is the same or a previous version. To force a write in this case, specify the option --force. |
| --force                | Force update of all firmware, even if the installed firmware version is the same or more recent than the images embedded in the utility.                                                                                                                                                         |
| --image=(filename)     | Update the firmware using the image contained in the specified file, rather than the image embedded in the utility. Use with the --write and, if needed, --force options.                                                                                                                        |
| -y, --yes              | Prompts for user confirmation before re-writing the firmware.                                                                                                                                                                                                                                    |

## Sfupdate: Examples

- Display firmware versions for all adapters:

```
sfupdate
```

```
sfupdate: Solarflare Firmware Update Utility [v3.0.5.2164]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
Network adapter driver version: v3.0.5.2163
```

```
sfxge0 - MAC: 00:0F:53:01:38:90
Firmware version: v3.0.5
Boot ROM version: v3.0.5.2163
```

```
PHY version: v2.0.2.5
Controller version: v3.0.5.2161

The Boot ROM firmware is up to date
The PHY firmware is up to date
The image contains a more recent version of the Controller [v3.0.5.2163]
vs [v3.0.5.2161]
Use the -w|--write option to perform an update

sfxge1 - MAC: 00:0F:53:01:38:91
 Firmware version: v3.0.5
 Boot ROM version: v3.0.5.2163
 PHY version: v2.0.2.5
 Controller version: v3.0.5.2161

The Boot ROM firmware is up to date
The PHY firmware is up to date
The image contains a more recent version of the Controller [v3.0.5.2163]
vs [v3.0.5.2161]
Use the -w|--write option to perform an update
```

## 6.10 Performance Tuning on FreeBSD

- [Introduction on page 216](#)
- [Tuning settings on page 217](#)
- [Other Considerations on page 222](#)

### Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to give good performance across a broad class of applications. Occasionally, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this section is to highlight adapter driver settings that affect the performance metrics described. This section covers the tuning of all Solarflare adapters.

Latency will be affected by the type of physical medium used: 10GBase-T, twinaxial (direct-attach), fiber or KX4. This is because the physical media interface chip (PHY) used on the adapter can introduce additional latency. Likewise, latency can also be affected by the type of SFP/SFP+/QSFP module fitted.

In addition, you may need to consider other issues influencing performance, such as application settings, server motherboard chipset, CPU speed, cache size, RAM size, additional software installed on the system, such as a firewall, and the specification and configuration of the LAN. Consideration of such issues is not within the scope of this guide.

## Tuning settings

### Port mode

The selected port mode for SFN7000 and SFN8000 series adapters should correspond to the speed and number of connectors in use, after using any splitter cables. If a restricted set of connectors is configured, the driver can then transfer resources from the unused connectors to those configured, potentially improving performance.

### Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved, because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support an MTU of up to 9216 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

The MTU is changed dynamically using `ifconfig`, where `sfxge<n>` is the interface name and `<size>` is the MTU size in bytes:

```
ifconfig sfxge<n> mtu <size>
```

Verification of the MTU setting may be performed by running `ifconfig` with no options and checking the MTU value associated with the interface. The change in MTU size can be made to persist across reboots by editing the `/etc/rc.conf` file and adding an `mtu` parameter to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> mtu <size>"
```

## Interrupt Moderation (Interrupt Coalescing)

*Interrupt moderation* reduces the number of interrupts generated by the adapter by coalescing multiple received packet events and/or transmit completion events together into a single interrupt.

The *interrupt moderation interval* sets the minimum time (in microseconds) between two consecutive interrupts. Coalescing occurs only during this interval:

- When the driver generates an interrupt, it starts timing the moderation interval.
- Any events that occur before the moderation interval expires are coalesced together into a single interrupt, that is raised only when the interval expires. A new moderation interval then starts, during which no interrupt is raised.
- An event that occurs after the moderation interval has expired gets its own dedicated interrupt, that is raised immediately. A new moderation interval then starts, during which no interrupt is raised.

Interrupt moderation settings are **critical for tuning adapter latency**:

- Increasing the interrupt moderation interval will:
  - generate less interrupts
  - reduce CPU utilization (because there are less interrupts to process)
  - increase latency
  - improve peak throughput.
- Decreasing the interrupt moderation interval will:
  - generate more interrupts
  - increase CPU utilization (because there are more interrupts to process)
  - decrease latency
  - reduce peak throughput.
- Turning off interrupt moderation will:
  - generate the most interrupts
  - give the highest CPU utilization
  - give the lowest latency
  - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits typically outweigh the cost of increased CPU utilization. It is recommended that:

- Interrupt moderation is disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation is enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.

Interrupt moderation is changed dynamically using `sysctl`.

To set the interrupt moderation, where `sfxge<n>` is the interface name, and the `<interval>` is in microseconds ( $\mu\text{s}$ ):

```
sysctl dev.sfxge.<n>.int_mod=<interval>
```

To turn off interrupt moderation, set an interval of zero (0):

```
sysctl dev.sfxge.<n>.int_mod=0
```

The change in interrupt moderation can be made to persist across reboots by editing the file `/etc/sysctl.conf` and adding `dev.sfxge.<n>.int_mod=<interval>` on a new line.



**NOTE:** The performance benefits of TCP Large Receive Offload are limited if interrupt moderation is disabled. See [TCP Large Receive Offload \(LRO\) on page 220](#).

### TCP/IP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver has all checksum offload features enabled by default. Therefore, there is no opportunity to improve performance from the default.

Checksum offload is changed dynamically using `ifconfig`, with the following parameters:

- `rxcsum, txcsu, rxcsum6, txcsu6`  
Enable Rx and Tx checksum offload for IPv4 and IPv6
- `-rxcsum, -txcsu, -rxcsum6, -txcsu6`  
Disable Rx and Tx checksum offload for IPv4 and IPv6

To enable checksum offload, where `sfxge<n>` is the interface name:

```
ifconfig sfxge<n> rxcsum txcsu rxcsum6 txcsu6
```

To disable checksum offload:

```
ifconfig sfxge<n> -rxcsum -txcsu -rxcsum6 -txcsu6
```

Verification of the checksum offload setting may be performed by running `ifconfig` with no options and checking the checksum offload value associated with the interface. The change in checksum offload can be made to persist across reboots by editing the `/etc/rc.conf` file and adding the appropriate parameters to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> rxcsum txcsu rxcsum6 txcsu6"
```



**NOTE:** Solarflare recommend you do not disable checksum offload.

### TCP Segmentation Offload (TSO)

TCP Segmentation Offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TSO benefits applications using TCP. Applications using protocols other than TCP will not be affected by TSO.

The FreeBSD TCP/IP stack provides a large TCP segment to the driver, which splits the data into MSS size, each with adjusted sequence space and a hardware calculated checksum.

Enabling TSO will reduce CPU utilization on the transmit side of a TCP connection and improve peak throughput, if the CPU is fully utilized. Since TSO has no effect on latency, it can be enabled at all times. The driver has TSO enabled by default. Therefore, there is no opportunity to improve performance from the default.

TSO is changed dynamically using `ifconfig`.

To enable TSO, where `sfxge<n>` is the interface name:

```
ifconfig sfxge<n> tso
```

To disable TSO:

```
ifconfig sfxge<n> -tso
```

Verification of the TSO setting may be performed by running `ifconfig` with no options and checking the TSO value associated with the interface. The change in TSO can be made to persist across reboots by editing the `/etc/rc.conf` file and adding the appropriate parameter to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> tso"
```

TCP and IP checksum offloads must be enabled for TSO to work.



**NOTE:** Solarflare recommend that you do not disable this setting.

### TCP Large Receive Offload (LRO)

TCP Large Receive Offload (LRO) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single larger packet before passing this onto the network stack for receive processing. This reduces CPU utilization and improves peak throughput when the CPU is fully utilized. The effectiveness of LRO is bounded by the interrupt moderation delay, and is limited if interrupt moderation is disabled (see [Interrupt Moderation \(Interrupt Coalescing\) on page 218](#)). Enabling LRO does not itself negatively impact latency.

The Solarflare network adapter driver enables LRO by default.

LRO is changed dynamically using `ifconfig`.

To enable LRO, where `sfxge<n>` is the interface name:

```
ifconfig sfxge<n> lro
```

To disable LRO:

```
ifconfig sfxge<n> -lro
```

Verification of the LRO setting may be performed by running `ifconfig` with no options and checking the LRO value associated with the interface. The change in LRO can be made to persist across reboots by editing the `/etc/rc.conf` file and adding the appropriate parameter to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> lro"
```



**NOTE:** LRO should **NOT** be enabled when using the host to forward packets from one interface to another. For example, if the host is performing IP routing.

## TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

Initial buffering settings should provide good performance. However for certain applications, tuning buffer settings can significantly benefit throughput. To change buffer settings, adjust the `tcp_rmem` and `tcp_wmem` using the `sysctl` command:

- Receive buffering:

```
sysctl net.ipv4.tcp_rmem=<min> <default> <max>"
```

- Transmit buffering:

```
sysctl net.ipv4.tcp_wmem=<min> <default> <max>"
```

(`tcp_rmem` and `tcp_wmem` can also be adjusted for IPV6 and globally with the `net.ipv6` and `net.core` variable prefixes respectively).

Typically it is sufficient to tune just the max buffer value. It defines the largest size the buffer can grow to. Suggested alternate values are `max=500000` (1/2 Mbyte). Factors such as link latency, packet loss and CPU cache size all influence the affect of the max buffer size values. The minimum and default values can be left at their defaults `minimum=4096` and `default=87380`.

See <https://wiki.freebsd.org/NetworkPerformanceTuning> for more details.

## Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts.

When RSS is enabled the controller uses multiple receive queues to deliver incoming packets. The receive queue selected for an incoming packet is chosen to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained. Each receive queue has its own dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores, providing a considerable performance advantage over a conventional adapter architecture in which all received packets for a given interface are processed by just one CPU core.

By default the driver enables RSS and configures one RSS Receive queue per CPU core. The number of RSS Receive queues is changed using `kenv` to modify the kernel environment variable `hw.sfxge.<n>.max_rss_channels`. The driver must be reloaded after the change using the `kldload` command.

To set <m> RSS Receive queues, where sfxge<n> is the interface name:

```
kenv hw.sfxge.<n>.max_rss_channels=<m>
kldload /boot/modules/sfxge.ko
```

Sometimes, it can be desirable to disable RSS when running single stream applications, since all interface processing may benefit from taking place on a single CPU. To do so, set a single RSS Receive queue:

```
kenv hw.sfxge.<n>.max_rss_channels=1
kldload /boot/modules/sfxge.ko
```

The change in RSS Receive queues can be made to persist across reboots by editing the file /boot/loader.conf and adding `hw.sfxge.<n>.max_rss_channels=<m>` on a new line.

If no MSI/MSI-X interrupts are available then the driver will fall-back to use a single legacy interrupt. RSS will be unavailable for that port.



**NOTE:** RSS also works for UDP packets. For UDP traffic the Solarflare adapter will select the Receive CPU based on IP source and destination addresses. Solarflare adapters support IPv4 and IPv6 RSS.

## Other Considerations

### PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different speeds and widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen 1, 5.0 Gbps for PCIe Gen 2 and 8.0Gbps for PCIe Gen 3) in each direction. *Solarflare adapters are designed for x8 or x16 lane operation.*

On some server motherboards, choice of PCIe slot is important. This is because some slots (including those that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will warn if it detects that the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

Solarflare adapters require a PCIe Gen 3 x8 or x16 slot for optimal performance. The Solarflare driver will warn if it detects that the adapter is placed in a sub-optimal slot.

In addition, the latency of communications between the host CPUs, system memory and the Solarflare PCIe adapter may be PCIe slot dependent. Some slots may be “closer” to the CPU, and therefore have lower latency and higher throughput. If possible, install the adapter in a slot which is local to the desired NUMA node

Please consult your server user guide for more information.

## CPU Power Management

The powerd service controls the CPU clock speed dynamically according to current processing demand. For latency sensitive applications, where the application switches between having packets to process and having periods of idle time waiting to receive a packet, dynamic clock speed control may increase packet latency. Solarflare recommend disabling the powerd service if minimum latency is the main consideration.

To stop powerd, type:

```
/etc/rc.d/powerd stop
```

To disable powerd across reboots, ensure this setting is present in /etc/rc.conf:

```
powerd_enable="NO"
```

## Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of memory modules (DIMMs) to populate in the server. For optimal memory bandwidth in the system, it is likely that:

- all DIMM slots should be populated
- all NUMA nodes should have memory installed.

Please consult the motherboard documentation for details.

## Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

## Tuning Recommendations

The following tables provide recommendations for tuning settings for different applications.

Throughput - [Table 45](#)

Latency - [Table 46](#)

Forwarding - [Table 47](#)

## Recommended Throughput Tuning

[Table 45](#) shows recommended tuning settings for throughput:

**Table 45: Throughput Tuning Settings**

| Tuning Parameter               | How?                                                                                                                                                                       |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MTU Size                       | Configure to maximum supported by network:<br><code>ifconfig sfxge&lt;n&gt; mtu &lt;size&gt;</code>                                                                        |
| Interrupt moderation           | Leave at default (Enabled).                                                                                                                                                |
| TCP/IP Checksum Offload        | Leave at default (Enabled).                                                                                                                                                |
| TCP Segmentation Offload       | Leave at default (Enabled).                                                                                                                                                |
| TCP Large Receive Offload      | Leave at default (Enabled).                                                                                                                                                |
| TCP Protocol Tuning            | Leave at default                                                                                                                                                           |
| Receive Side Scaling (RSS)     | Application dependent                                                                                                                                                      |
| Buffer Allocation Method       | Leave at default. Some applications may benefit from specific setting.                                                                                                     |
| PCI Express Lane Configuration | Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”. |
| CPU Power Management           | Leave enabled                                                                                                                                                              |
| Memory bandwidth               | Ensure memory utilizes all memory channels on system motherboard                                                                                                           |

## Recommended Latency Tuning

[Table 46](#) shows recommended tuning settings for latency:

**Table 46: Latency Tuning Settings**

| Tuning Parameter         | How?                                                                                                |
|--------------------------|-----------------------------------------------------------------------------------------------------|
| MTU Size                 | Configure to maximum supported by network:<br><code>ifconfig sfxge&lt;n&gt; mtu &lt;size&gt;</code> |
| Interrupt moderation     | Disable with:<br><code>sysctl dev.sfxge.&lt;n&gt;.int_mod=0</code>                                  |
| TCP/IP Checksum Offload  | Leave at default (Enabled).                                                                         |
| TCP Segmentation Offload | Leave at default (Enabled).                                                                         |

**Table 46: Latency Tuning Settings**

| Tuning Parameter               | How?                                                                                                                                                                       |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TCP Large Receive Offload      | Disable with:<br><code>ifconfig sfxge&lt;n&gt; -lro</code>                                                                                                                 |
| TCP Protocol Tuning            | Leave at default, but changing does not impact latency                                                                                                                     |
| Receive Side Scaling           | Application dependent                                                                                                                                                      |
| Buffer Allocation Method       | Leave at default. Some applications may benefit from specific setting.                                                                                                     |
| PCI Express Lane Configuration | Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”. |
| CPU Power Management           | Disable with:<br><code>/etc/rc.d/powerd stop</code>                                                                                                                        |
| Memory bandwidth               | Ensure memory utilizes all memory channels on system motherboard                                                                                                           |

### Recommended Forwarding Tuning

Table 47 shows recommended tuning settings for forwarding:

**Table 47: Forwarding Tuning Settings**

| Tuning Parameter           | How?                                                                                                             |
|----------------------------|------------------------------------------------------------------------------------------------------------------|
| MTU Size                   | Configure to maximum supported by network:<br><code>ifconfig sfxge&lt;n&gt; mtu &lt;size&gt;</code>              |
| Interrupt moderation       | Configure an explicit interrupt moderation interval with:<br><code>sysctl dev.sfxge.&lt;n&gt;.int_mod=150</code> |
| TCP/IP Checksum Offload    | Leave at default (Enabled).                                                                                      |
| TCP Segmentation Offload   | Leave at default (Enabled).                                                                                      |
| TCP Large Receive Offload  | Disable with:<br><code>ifconfig sfxge&lt;n&gt; -lro</code>                                                       |
| TCP Protocol Tuning        | Leave at default                                                                                                 |
| Receive Side Scaling (RSS) | Leave at default                                                                                                 |

**Table 47: Forwarding Tuning Settings**

| Tuning Parameter               | How?                                                                                                                                                                       |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Buffer Allocation Method       | Leave at default. Some applications may benefit from specific setting.                                                                                                     |
| PCI Express Lane Configuration | Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as "x8 and 5GT/s", or "x8 and 8GT/s", or "x8 and Unknown". |
| CPU Power Management           | Leave enabled                                                                                                                                                              |
| Memory bandwidth               | Ensure memory utilizes all memory channels on system motherboard                                                                                                           |

## 6.11 Module Parameters

**Table 48** lists the available parameters in the Solarflare FreeBSD driver module:

- all parameters have a `hw.sfxge.` prefix
- for example, the full name of the parameter shown as `rx_ring` is `hw.sfxge.rx_ring`:

**Table 48: Driver Module Parameters**

| Parameter                   | Description                                                                                                                                                                             | Possible Value                 | Default Value |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|---------------|
| <code>rx_ring</code>        | Size of Rx and Tx rings (maximum number of descriptors) per queue.<br><br>Values used by the driver (default or specified when module is loaded) can be obtained using the same sysctl. | 512,<br>1024,<br>2048,<br>4096 | 1024          |
| <code>tx_ring</code>        | Size of Rx and Tx rings (maximum number of descriptors) per queue.<br><br>Values used by the driver (default or specified when module is loaded) can be obtained using the same sysctl. | 512,<br>1024,<br>2048          | 1024          |
| <code>lro.table_size</code> | Size of the LRO hash table. Must be a power of 2.                                                                                                                                       | uint                           | 128           |
| <code>lro.chain_max</code>  | Maximum length of chains in the LRO hash table.                                                                                                                                         | uint                           | 20            |
| <code>lro.idle_ticks</code> | Time (in jiffies) after which an idle connection's LRO state is discarded.                                                                                                              | uint                           | 101           |

**Table 48: Driver Module Parameters**

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Possible Value | Default Value |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------------|
| lro.slow_start_packets | Number of packets that must pass in-order before starting LRO.                                                                                                                                                                                                                                                                                                                                                                                                                                                  | uint           | 20000         |
| lro.loss_packets       | Number of packets that must pass in-order following loss before restarting LRO.                                                                                                                                                                                                                                                                                                                                                                                                                                 | uint           | 20            |
| tx_dpl_get_max         | <p>Maximum number of packets queued in the software <i>get-list</i> for a transmit queue.</p> <p>The get-list is used to get packets to be put onto the Tx ring. It should be big enough to avoid drops of locally generated TCP packets when many (1000+) streams are running in parallel. Accessing this list requires the transmit queue lock.</p> <p>If a packet is dropped because this limit has been exceeded, the sender gets an ENOBUFS error, and the <code>tx_get_overflow</code> counter grows.</p> | uint           | 65536         |
| tx_dpl_get_non_tcp_max | <p>Maximum number of non-TCP packets queued in the software <i>get-list</i> for a transmit queue.</p> <p>This parameter can restrict utilizing the queue for non-TCP (e.g. UDP) packets, which can easily overflow any queue because there is no back-pressure.</p> <p>If a packet is dropped because this limit has been exceeded, the sender gets an ENOBUFS error, and the <code>tx_get_non_tcp_overflow</code> counter grows.</p>                                                                           | uint           | 1024          |
| tx_dpl_put_max         | <p>Maximum number of packets queued in the software <i>put-list</i> for a transmit queue.</p> <p>The put-list is used to put packets temporarily when the transmit queue lock cannot be obtained. The packets are moved to the get-list as soon as the transmit queue lock is acquired and the queue is served.</p> <p>If a packet is dropped because this limit has been exceeded, the sender gets an ENOBUFS error, and the <code>tx_put_overflow</code> counter grows.</p>                                   | uint           | 1024          |

**Table 48: Driver Module Parameters**

| Parameter            | Description                                                                                                                                                                                                                                                                                                                                                                         | Possible Value | Default Value |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------------|
| tso_fw_assisted      | Whether to assist TSO using the firmware.<br>Applicable to SFN7000 and SFN8000 series adapters only.                                                                                                                                                                                                                                                                                | 0 1            | 1             |
| <n>.max_rss_channels | The number of RSS Receive queues for interface sfxge<n>. See <a href="#">Receive Side Scaling (RSS) on page 221</a> . The actual number may be lower due to availability of MSI-X interrupts. There is a maximum of 32 MSI-X interrupts across all network devices.<br><br>If no value is set (the default), the number is limited only by the number of CPUs and MSI-X interrupts. | unit           | —             |

## 6.12 Kernel and Network Adapter Statistics

The Linux command `sysctl` will display an extensive range of statistics originated from the MAC on the Solarflare network adapter. To display statistics use the following command:

```
sysctl dev.sfxge.<n>.stats
```

where `sfxge<n>` is the interface name.

Tables below list the complete output from the `sysctl dev.sfxge.<n>.stats` command. See:

- [Table 49 on page 228](#)
- [Table 50 on page 231](#)
- [Table 51 on page 232](#).

Per port statistics ([Table 51 on page 232](#)) are from the physical adapter port. Other statistics are from the specified PCIe function.



**NOTE:** `sysctl dev.sfxge.<n>.stats` output depends on the features supported by the adapter type.

**Table 49: Event queue statistics**

| Field    | Description                               |
|----------|-------------------------------------------|
| ev_all   | Total number of events.                   |
| ev_rx    | Number of packets received by driver.     |
| ev_rx_ok | Number of received packets not discarded. |

**Table 49: Event queue statistics**

| Field                     | Description                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ev_rx_recovery            | Not supported.                                                                                                                                                                                                                                                                                                                                                                                                           |
| ev_rx_frm_trunc           | Number of packets truncated because an internal FIFO is full.                                                                                                                                                                                                                                                                                                                                                            |
|                           | As a packet is received it is fed by the MAC into a 128K FIFO. If for any reason the PCI interface cannot keep pace and is unable to empty the FIFO at a sufficient rate, the MAC will be unable to feed more of the packet to the FIFO. In this event the MAC will truncate the packet marking it as such, and discard the remainder. The driver on seeing a 'partial' packet which has been truncated will discard it. |
| ev_rx_tobe_disc           | Number of packets marked by the adapter to be discarded because of one of the following: <ul style="list-style-type: none"> <li>• mismatched unicast address and unicast promiscuous mode is not enabled</li> <li>• packet is a pause frame</li> <li>• packet has length discrepancy</li> <li>• internal FIFO overflow condition</li> <li>• length &lt; 60 bytes.</li> </ul>                                             |
| ev_rx_pause_frm_err       | Number of pause packets received.                                                                                                                                                                                                                                                                                                                                                                                        |
| ev_rx_buf_owner_id_err    | Event caused by internal driver error.                                                                                                                                                                                                                                                                                                                                                                                   |
| ev_rx_ipv4_hdr_chksum_err | Number of packets received with IP header checksum error.                                                                                                                                                                                                                                                                                                                                                                |
| ev_rx_tcp_udp_chksum_err  | Number of packets received with TCP/UDP checksum error.                                                                                                                                                                                                                                                                                                                                                                  |
| ev_rx_eth_crc_err         | Number of packets received whose CRC did not match the internally generated CRC value.                                                                                                                                                                                                                                                                                                                                   |
| ev_rx_ip_frag_err         | Number of IP fragments received (note this is not an error).                                                                                                                                                                                                                                                                                                                                                             |
| ev_rx_mcast_pkt           | Number of IP multicast packets received.                                                                                                                                                                                                                                                                                                                                                                                 |
| ev_rx_mcast_hash_match    | Number of IP multicast packets received which have matched the IP multicast match filter.                                                                                                                                                                                                                                                                                                                                |
| ev_rx_tcp_ipv4            | Number of TCP/IPv4 packets received.                                                                                                                                                                                                                                                                                                                                                                                     |

**Table 49: Event queue statistics**

| Field                         | Description                                                             |
|-------------------------------|-------------------------------------------------------------------------|
| ev_rx_tcp_ipv6                | Number of TCP/IPv6 packets received.                                    |
| ev_rx_udp_ipv4                | Number of UDP/IPv4 packets received.                                    |
| ev_rx_udp_ipv6                | Number of UDP/IPv6 packets received.                                    |
| ev_rx_other_ipv4              | Number of IPv4 packets received which are not TCP or UDP.               |
| ev_rx_other_ipv6              | Number of IPv6 packets received which are not TCP or UDP.               |
| ev_rx_non_ip                  | Number of packets received which are not IP.                            |
| ev_rx_overrun                 | Number of received packets dropped by receiver because of FIFO overrun. |
| ev_tx                         | Number of transmitted packets.                                          |
| ev_tx_wq_ff_full              | Number of transmitted packets dropped because of FIFO overrun.          |
| ev_tx_pkt_err                 | Number of transmitted packets dropped because of driver error.          |
| ev_tx_pkt_too_big             | Number of transmitted packets dropped because of driver error.          |
| ev_tx_unexpected              | Number of transmitted packets dropped because of driver error.          |
| ev_global                     | Internal driver event.                                                  |
| ev_global_phy                 | Internal driver event.                                                  |
| ev_global_mnt                 | Internal driver event.                                                  |
| ev_global_rx_recovery         | Internal driver event.                                                  |
| ev_driver                     | Internal driver event.                                                  |
| ev_driver_srm_upd_done        | Internal driver event.                                                  |
| ev_driver_tx_descq_fls_done   | Internal driver event.                                                  |
| ev_driver_rx_descq_fls_done   | Internal driver event.                                                  |
| ev_driver_rx_descq_fls_failed | Internal driver event.                                                  |
| ev_driver_rx_dsc_error        | Internal driver event.                                                  |
| ev_driver_tx_dsc_error        | Internal driver event.                                                  |

**Table 49: Event queue statistics**

| Field            | Description            |
|------------------|------------------------|
| ev_drv_gen       | Internal driver event. |
| ev_mcdi_response | Internal driver event. |

**Table 50: Driver statistics**

| Field            | Description                                                                                                                                                                                                                                                                            |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| lro_merges       | Number of packets absorbed by LRO.                                                                                                                                                                                                                                                     |
| lro_bursts       | Number of bursts spotted by LRO.                                                                                                                                                                                                                                                       |
| lro_slow_start   | Number of packets not merged because connection may be in slow-start.                                                                                                                                                                                                                  |
| lro_misorder     | Number of out-of-order packets seen in tracked streams.                                                                                                                                                                                                                                |
| lro_too_many     | Incremented when the driver is trying to track too many streams.                                                                                                                                                                                                                       |
| lro_new_stream   | Number of distinct streams the driver has tracked.                                                                                                                                                                                                                                     |
| lro_drop_idle    | Number of streams discarded because they went idle.                                                                                                                                                                                                                                    |
| lro_drop_closed  | Number of streams that have seen a FIN or RST.                                                                                                                                                                                                                                         |
| tso_bursts       | Number of times TSO transmit invoked by the kernel.                                                                                                                                                                                                                                    |
| tso_packets      | Number of packets sent via the TSO transmit path.                                                                                                                                                                                                                                      |
| tso_long_headers | Number of packets with headers too long for standard blocks.                                                                                                                                                                                                                           |
| tx_collapses     | Number of packets with too many fragments collapsed.                                                                                                                                                                                                                                   |
| tx_drops         | Number of packets dropped by the driver because of: <ul style="list-style-type: none"> <li>• transmit queue in inappropriate state</li> <li>• memory allocation or DMA mapping failures required to handle packet with long header by TSO</li> <li>• mbuf collapse failure.</li> </ul> |

**Table 50: Driver statistics**

| Field                   | Description                                                                                                                                                                                                             |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tx_get_overflow         | Number of packets early dropped by the driver because of software transmit queue overflow (see <code>hw.sfxge.tx_dpl_get_max</code> and <code>hw.sfxge.tx_dpl_put_max</code> in <a href="#">Table 48 on page 226</a> ). |
| tx_put_overflow         | Number of packets early dropped by the driver because of software transmit queue overflow (see <code>hw.sfxge.tx_dpl_get_max</code> and <code>hw.sfxge.tx_dpl_put_max</code> in <a href="#">Table 48 on page 226</a> ). |
| tx_get_non_tcp_overflow | Number of non-TCP packets early dropped by the driver because of software transmit queue limit for non-TCP packets (see <code>hw.sfxge.tx_dpl_get_non_tcp_max</code> in <a href="#">Table 48 on page 226</a> ).         |
| tx_netdown_drops        | Number of packets early dropped by the driver because of link is down.                                                                                                                                                  |
| tso_pdrop_too_many      | Number of TSO packets partially dropped by the driver because TSO generates too many segments (most likely because of tiny MSS).                                                                                        |
| tso_pdrop_no_rsrc       | Number of TSO packets partially dropped by the driver because the packet header is too big and requires per-segment memory allocation and DMA mapping which failed.                                                     |

**Table 51: Port statistics**

| Field            | Description                                               |
|------------------|-----------------------------------------------------------|
| rx_octets        | Number of bytes received. Not include collided bytes.     |
| rx_pkts          | Number of packets received.                               |
| rx_unicst_pkts   | Number of unicast packets received.                       |
| rx_multicst_pkts | Number of multicast packets received.                     |
| rx_brdcst_pkts   | Number of broadcasted packets received.                   |
| rx_pause_pkts    | Number of pause frames received with valid pause op_code. |

**Table 51: Port statistics**

| Field                   | Description                                                                                                                                                     |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rx_le_64_pkts           | Number of packets received where the length is less than or equal to 64 bytes.                                                                                  |
| rx_65_to_127_pkts       | Number of packets received where the length is between 65 and 127 bytes.                                                                                        |
| rx_128_to_255_pkts      | Number of packets received where the length is between 128 and 255 bytes.                                                                                       |
| rx_256_to_511_pkts      | Number of packets received where the length is between 256 and 511 bytes.                                                                                       |
| rx_512_to_1023_pkts     | Number of packets received where the length is between 512 and 1023 bytes.                                                                                      |
| rx_1024_to_15xx_pkts    | Number of packets received where the length is between 1024 and 1518 bytes (1522 with VLAN tag).                                                                |
| rx_ge_15xx_pkts         | Number of packets received where the length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.                                                          |
| rx_errors               | Number of packets received with errors.                                                                                                                         |
| rx_fcs_errors           | Number of packets received with FCS errors.                                                                                                                     |
| rx_drop_events          | Number of packets dropped by receiver.                                                                                                                          |
| rx_false_carrier_errors | Count of the instances of false carrier detected. False carrier is activity on the receive channel that does not result in a packet receive attempt being made. |
| rx_symbol_errors        | Port error condition.                                                                                                                                           |
| rx_align_errors         | Port error condition.                                                                                                                                           |
| rx_internal_errors      | Port error condition.                                                                                                                                           |
| rx_jabber_pkts          | Port error condition.                                                                                                                                           |
| rx_lane0_char_err       | Port error condition.                                                                                                                                           |
| rx_lane1_char_err       | Port error condition.                                                                                                                                           |
| rx_lane2_char_err       | Port error condition.                                                                                                                                           |
| rx_lane3_char_err       | Port error condition.                                                                                                                                           |
| rx_lane0_disp_err       | Port error condition.                                                                                                                                           |

**Table 51: Port statistics**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rx_lane1_disp_err  | Port error condition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| rx_lane2_disp_err  | Port error condition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| rx_lane3_disp_err  | Port error condition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| rx_match_fault     | Number of packets received which did not match any filter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| rx_nodesc_drop_cnt | <p>Number of packets dropped by the network adapter because of a lack of RX descriptors in the RX queue.</p> <p>Packets can be dropped by the NIC when there are insufficient RX descriptors in the RX queue to allocate to the packet. This problem occurs if the receive rate is very high and the network adapter receive cycle process has insufficient time between processing to refill the queue with new descriptors.</p> <p>A number of different steps can be tried to resolve this issue:</p> <ul style="list-style-type: none"> <li>• Disable the irqbalance daemon in the OS</li> <li>• Distribute the traffic load across the available CPU/cores by setting rss_cpus=cores. Refer to Receive Side Scaling section</li> <li>• Increase receive queue size using ethtool.</li> </ul> |
| tx_octets          | Number of bytes transmitted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| tx_pkts            | Number of packets transmitted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| tx_unicst_pkts     | Number of unicast packets transmitted.<br>Includes flow control packets.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| tx_multicst_pkts   | Number of multicast packets transmitted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| tx_brdcst_pkts     | Number of broadcast packets transmitted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| tx_pause_pkts      | Number of pause frames transmitted with valid pause op_code.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| tx_le_64_pkts      | Number of frames transmitted where the length is less than or equal to 64 bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| tx_65_to_127_pkts  | Number of frames transmitted where the length is between 65 and 127 bytes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

**Table 51: Port statistics**

| <b>Field</b>         | <b>Description</b>                                                                                   |
|----------------------|------------------------------------------------------------------------------------------------------|
| tx_128_to_255_pkts   | Number of frames transmitted where the length is between 128 and 255 bytes                           |
| tx_256_to_511_pkts   | Number of frames transmitted where the length is between 256 and 511 bytes                           |
| tx_512_to_1023_pkts  | Number of frames transmitted where length is between 512 and 1023 bytes                              |
| tx_1024_to_15xx_pkts | Number of frames transmitted where the length is between 1024 and 1518 bytes (1522 with VLAN tag).   |
| tx_ge_15xx_pkts      | Number of frames transmitted where length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes. |
| tx_errors            | Port error condition.                                                                                |
| tx_sgl_col_pkts      | Port error condition.                                                                                |
| tx_mult_col_pkts     | Port error condition.                                                                                |
| tx_ex_col_pkts       | Port error condition.                                                                                |
| tx_late_col_pkts     | Port error condition.                                                                                |
| tx_def_pkts          | Port error condition.                                                                                |
| tx_ex_def_pkts       | Port error condition.                                                                                |

## Netstat statistics

The Linux command netstat also displays some of these statistics. They are periodically updated from the port and driver statistics. See [Table 52](#):

**Table 52: Netstat statistics**

| Field  | Value                                                                                                                              |
|--------|------------------------------------------------------------------------------------------------------------------------------------|
| lpkts  | rx_pkts                                                                                                                            |
| lerrs  | rx_errors                                                                                                                          |
| ldrop  | 0                                                                                                                                  |
| lbytes | rx_octets                                                                                                                          |
| Opkts  | tx_pkts                                                                                                                            |
| Oerrs  | tx_errors + tx_drops + get_overflow + get_non_tcp_overflow + put_overflow + netdown_drops + tso_pdrop_too_many + tso_pdrop_no_rsrc |
| Obytes | tx_octets                                                                                                                          |
| Coll   | tx_sgl_col_pkts + tx_mult_col_pkts + tx_ex_col_pkts + tx_late_col_pkts                                                             |

# 7

# SR-IOV Virtualization Using KVM

## 7.1 Introduction

This chapter describes SR-IOV and virtualization using Linux KVM and Solarflare SFN7000 or SFN8000 series adapters.

SR-IOV enabled on Solarflare adapters provides accelerated cut-through performance and is fully compatible with hypervisor based services and management tools. The advanced design of Solarflare SFN7000 and SFN8000 series adapters incorporates a number of features to support SR-IOV. These features can be summarized as follows:

- PCIe Virtual Functions (VF).

A PCIe physical function, PF, can support a configurable number of PCIe virtual functions. In total 240 VFs can be allocated between the PFs. The adapter can also support a total of 2048 MSI-X interrupts.

- Layer 2 Switching Capability.

A layer 2 switch configured in firmware supports the transport of network packets between PCI physical functions (PF), Virtual functions (VF) and the external network. This allows received packets to be replicated across multiple PFs/VFs and allows packets transmitted from one PF to be received on another PF or VF.

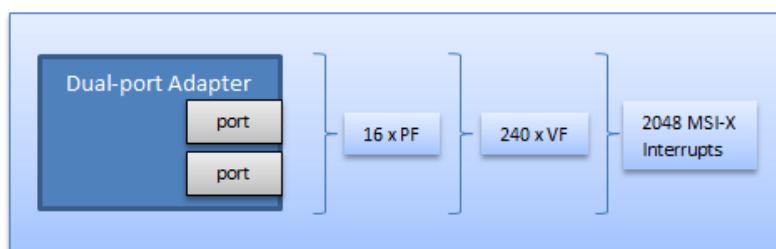


Figure 12: Per Adapter - Configuration Options

## Supported Platforms

### Host

- Red Hat Enterprise Linux 6.5 - 7.0 KVM

### Guest VM

- Red Hat Enterprise Linux 5.x, 6.x and 7.x

Acceleration of guest Virtual Machines (VM) running other (non-Linux) operating systems are not currently supported, however other schemes, for example, a KVM direct bridged configuration using the Windows virtio-net driver could be used.

## Driver/Firmware

Features described in the chapter require the following (minimum) Solarflare driver and firmware versions.

```
ethtool -i eth<N>
driver: sfc
version: 4.4.1.1017
firmware-version: 4.4.2.1011 rx0 tx0
```

The adapter must be using the *full-feature* firmware variant which can be selected using the sfboot utility and confirmed with **rx0 tx0** appearing after the version number in the output from ethtool as shown above.

The firmware update utility (**sfupdate**) and boot ROM configuration tool (**sfboot**) are available in the Solarflare Linux Utilities package (SF-107601-LS issue 28 or later).

## Platform support - SR-IOV

### BIOS

To use SR-IOV modes, SR-IOV must be enabled in the platform BIOS where the actual BIOS setting can differ between machines, but may be identified as SR-IOV, IOMMU or VT-d and VT-x on an Intel platform.

The following links identify Linux Red Hat documentation for SR-IOV BIOS settings.

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Virtualization\\_Deployment\\_and\\_Administration\\_Guide/index.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/index.html)

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Virtualization\\_Administration\\_Guide/sect-Virtualization-Troubleshooting-Enabling\\_Intel\\_VT\\_and\\_AMD\\_V\\_virtualization\\_hardware\\_extensions\\_in\\_BIOS.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/sect-Virtualization-Troubleshooting-Enabling_Intel_VT_and_AMD_V_virtualization_hardware_extensions_in_BIOS.html)

There may be other BIOS options which should be enabled to support SR-IOV, for example on DELL servers the following BIOS option must also be enabled:

Integrated Devices, SR-IOV Global Enable

*Users are advised to consult the server vendor BIOS options documentation.*

## Kernel Configuration

On an Intel platform, the IOMMU must be explicitly enabled by appending `intel_iommu=on` to the kernel line in the `/boot/grub/grub.conf` file. The equivalent setting on an AMD system is `amd_iommu=on`.

Solarflare recommends that users also enable the `pci=realloc` kernel parameter in the `/boot/grub/grub.conf` file. This allows the kernel to reassign addresses to PCIe apertures (i.e. bridges, ports) in the system when the BIOS does not allow enough PCI apertures for the maximum number of supported VFs.

## KVM - Interrupt Re-Mapping

To use PCIe VF passthrough, the server must support interrupt re-mapping. If the target server does not support interrupt re-mapping it is necessary to set the following option in a user created file e.g. `kvm_iommu_map_guest.conf` in the `/etc/modprobe.d` directory:

```
[RHEL 6] options kvm allow_unsafe_assigned_interrupts=1
[RHEL 7] options vfio_iommu_type1 allow_unsafe_assigned_interrupts=1
```

## Alternative Routing-ID Interpretation (ARI)

The ARI extension to the PCI Express Base Specification extends the capacity of a PCIe endpoint by increasing the number of accessible functions (PF+VF) from 8, up to 256. Without ARI support - which is a feature of the server hardware and BIOS, a server hosting a virtualized environment will be limited to 8 functions. Solarflare SFN7000 and SFN8000 series adapters can expose up to 16 PFs and 240 VFs per adapter.

Users should consult the appropriate server vendor documentation to ensure that the host server supports ARI.

## Supported Adapters

All Solarflare SFN7000 and SFN8000 series adapters fully support SR-IOV.

The `sfboot` utility allows the user to configure:

- The number of PFs exposed to host and/or Virtual Machine (VM).
- The number VFs exposed to host and/or Virtual Machine (VM).
- The number of MSI-X interrupts assigned to each PF or VF.

The Solarflare implementation uses a single driver (`sfc.ko`) that binds to both PFs and VFs.

## sfboot - Configuration Options

Adapter configuration options are set using the sfboot utility *v4.5.0 or later* from the Solarflare Linux Utilities package (SF-107601-LS issue 28 or later). The firmware variant must be set to full-feature / Virtualization.

```
sfboot firmware-variant=full-feature
```

To check the current adapter configuration run the sfboot command:

```
sfboot
Solarflare boot configuration utility [v4.5.0]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth5:
 Boot image Option ROM only
 Link speed Negotiated automatically
 Link-up delay time 5 seconds
 Banner delay time 2 seconds
 Boot skip delay time 5 seconds
 Boot type Disabled
 Physical Functions per port 1
 MSI-X interrupt limit 32
 Number of Virtual Functions 2
 VF MSI-X interrupt limit 8
 Firmware variant full feature / virtualization
 Insecure filters Disabled
 MAC spoofing Disabled
 VLAN tags None
 Switch mode SRIOV
```

*For some configuration option changes using sfboot, the server must be power cycled (power off/power on) before the changes are effective. sfboot will display a warning when this is required.*

Table 53 identifies sfboot SR-IOV configurable options.

**Table 53: sfboot - SR-IOV options**

| Option       | Default Value | Description                                                                                                                                                                                                |
|--------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pf-count=<n> | 1             | Number of PCIe PFs per physical port.<br><br>MAC address assignments may change, after next reboot, following changes with this option.                                                                    |
| pf-vlans     | None          | A comma separated list of VLAN tags for each PF.<br><br>sfboot pf-vlans=0,100,110,120<br><br>The first tag is assigned to the first PF, thereafter tags are assigned to PFs in (lowest) MAC address order. |

**Table 53: sfboot - SR-IOV options**

| Option             | Default Value | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mac-spoofing       | disabled      | If enabled, non-privileged functions may create unicast filters for MAC addresses that are not associated with themselves.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| msix-limit=<n>     | 32            | This should be used when using bonded interfaces where a bond slave inherits the bond master hardware address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| switch-mode=<mode> | default       | <p>Specifies the mode of operation that the port will be used in:</p> <p>default - single PF created, zero VFs created.</p> <p>sriov - SR-IOV enabled, single PF created, VFs configured with vf-count.</p> <p>partitioning - PFs configured with pf-count, VFs configured with vf-count. See <a href="#">NIC Partitioning on page 59</a> for details.</p> <p>partitioning-with-sriov - SR-IOV enabled, PFs configured with pf-count, VFs configured with vf-count. See <a href="#">NIC Partitioning on page 59</a> for details.</p> <p>pfiov - PFIOV enabled, PFs configured with pf-count, VFs not supported. Layer 2 switching between PFs.</p> |
| vf-count=<n>       | 240           | Number of virtual functions per PF.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

**Table 53: sfboot - SR-IOV options**

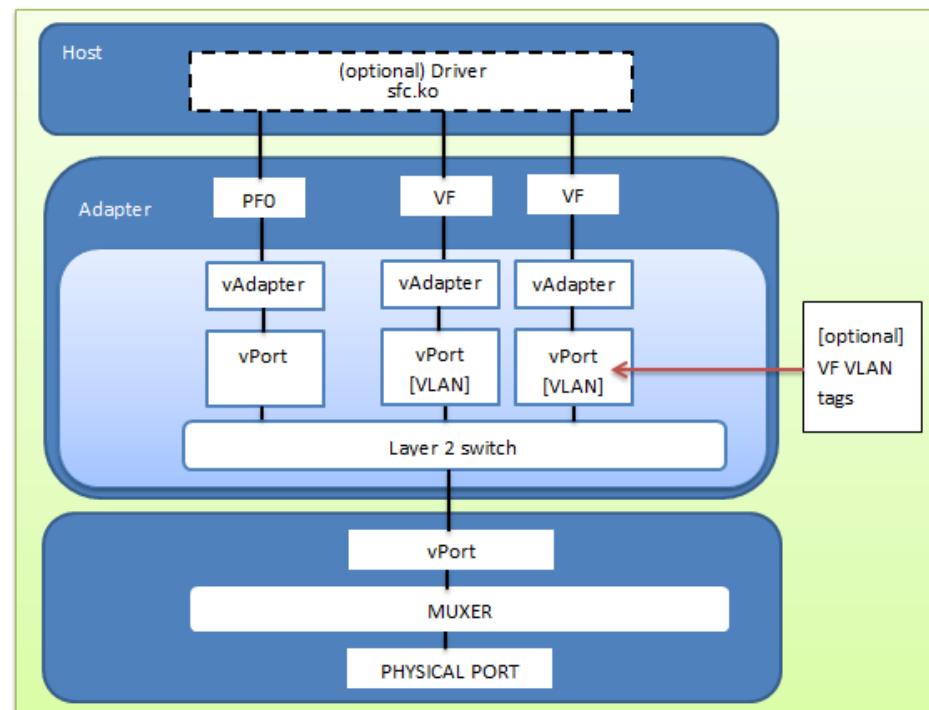
| Option                              | Default Value | Description                                                                                                                              |
|-------------------------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------|
| vf-msix-limit=<n>                   | 8             | Number of MSI-X interrupts per VF.<br>The adapter supports a maximum 2048 interrupts. The specified value for a PF must be a power of 2. |
| insecure_filters=<enabled disabled> | disabled      | When enabled, a function (PF or VF) can insert filters not qualified by its own permanent MAC address.                                   |

## 7.2 SR-IOV

In the simplest of SR-IOV supported configurations each physical port is exposed as a single PF (adapter default) and up to 240 VFs.

The Solarflare net driver (sfc.ko) will detect that PF/VFs are present from the sfboot configuration and automatically configure the virtual adapters and virtual ports as required.

Adapter firmware will also configure the firmware switching functions allowing packets to pass between PF and VFs or from VF to VF.


**Figure 13: SR-IOV - Single PF, Multiple VFs**

- With no VLAN configuration, the PFs and VFs are in the same Ethernet layer 2 broadcast domain i.e. a packet broadcast from the PF would be received by all VFs. VLAN tags can optionally be assigned to VFs using standard libvirt commands.
- The L2 switch supports replication of received/transmitted broadcast packets to all functions.
- The L2 switch will replicate received/transmitted multicast packets to all functions that have subscribed.
- The MUXER function is a firmware enabled layer2 switching function for transmit and receive traffic.

In the example above there are no virtual machines (VM) created. Network interfaces for the PF and each VF will appear in the host. An sfc NIC driver loaded in the host will identify the PF and each VF as individual network interfaces.

## SR-IOV Configuration

Ensure SR-IOV and the IOMMU are enabled on the host server kernel command line  
 - Refer to [Platform support - SR-IOV on page 238](#).

- The example configures 1 PF per port (default), 2 VFs per PF):

```
sfboot switch-mode=sriov pf-count=1 vf-count=2
Solarflare boot configuration utility [v4.5.0]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
```

|                                    |                                      |
|------------------------------------|--------------------------------------|
| <b>eth8:</b>                       |                                      |
| Boot image                         | Option ROM only                      |
| Link speed                         | Negotiated automatically             |
| Link-up delay time                 | 5 seconds                            |
| Banner delay time                  | 2 seconds                            |
| Boot skip delay time               | 5 seconds                            |
| Boot type                          | Disabled                             |
| <b>Physical Functions per port</b> | <b>1</b>                             |
| MSI-X interrupt limit              | 32                                   |
| <b>Number of Virtual Functions</b> | <b>2</b>                             |
| VF MSI-X interrupt limit           | 8                                    |
| <b>Firmware variant</b>            | <b>full feature / virtualization</b> |
| Insecure filters                   | Disabled                             |
| MAC spoofing                       | Disabled                             |
| VLAN tags                          | None                                 |
| <b>Switch mode</b>                 | <b>SRIOV</b>                         |

- Create VFs - see [Enabling Virtual Functions on page 259](#).
- The server should be cold rebooted following changes using sfboot. Following the reboot, The PF and VFs will be visible in the host using the ifconfig command and lspci (the output below is from a dual-port adapter. VFs are shown in bold text):

```
lspci -d1924:
03:00.0 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
03:00.1 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
```

```
03:00.2 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
03:00.3 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
03:00.4 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
03:00.5 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
```

- 4 To identify which physical port a given network interface is using:  

```
cat /sys/class/net/eth<N>/device/physical_port
```
- 5 To identify which PF a given VF is associated with use the following command  
(in this example there are 4 VFs assigned to PF eth4):  

```
ip link show
```

```
19: eth4: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
 link/ether 00:0f:53:21:00:61 brd ff:ff:ff:ff:ff:ff
 vf 0 MAC 76:c1:36:0a:be:2b
 vf 1 MAC 1e:b8:a8:ea:c7:fb
 vf 2 MAC 52:6e:32:3d:50:85
 vf 3 MAC b6:ad:a0:56:39:94
```

MAC addresses beginning `00:0f:53` are Solarflare designated hardware addresses. MAC addresses assigned to VFs in the above example output have been randomly generated by the host. MAC addresses visible to the host will be replaced by libvirt-generated MAC addresses in a VM.

## 7.3 KVM Network Architectures

This section identifies SR-IOV and the Linux KVM virtualization infrastructure configurations to consume adapter port Physical Functions (PF) and Virtual Functions (VF).

- [KVM libvirt Bridged on page 244](#)
- [KVM Direct Bridged on page 248](#)
- [KVM Libvirt Direct Passthrough on page 251](#)
- [KVM Libvirt Network Hostdev on page 254](#)
- [General Configuration on page 259](#)
- [Enabling Virtual Functions on page 259](#)

When migration is not a consideration, Solarflare recommends the network-hostdev configuration for highest throughput and lowest latency performance

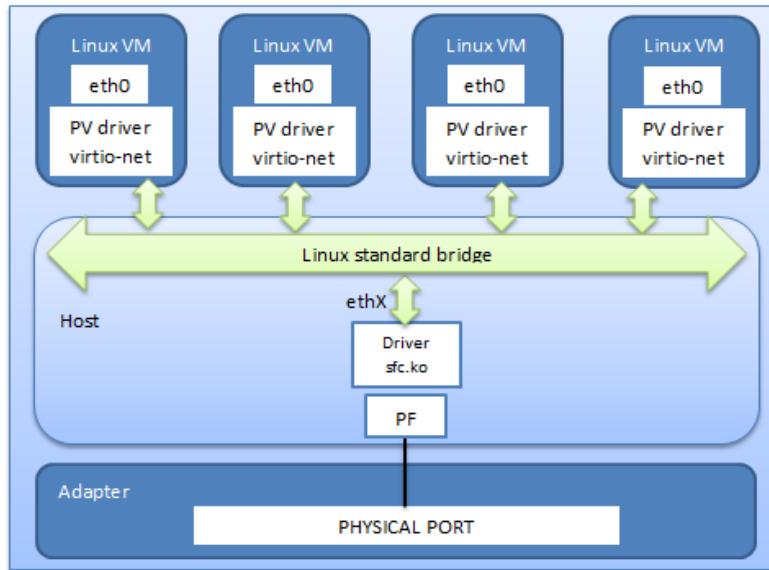
### KVM libvirt Bridged

The traditional method of configuring networking in KVM virtualized environments uses the para-virtualized (PV) driver, `virtio-net`, in the virtual machine and the standard Linux bridge in the host.

The bridge emulates a layer 2 learning switch to replicate multicast and broadcast packets in software and supports the transport of network traffic between VMs and the physical port.

This configuration uses standard Linux tools for configuration and needs only a virtualized environment and guest operating system.

Performance (latency/throughput) will not be as good as a network-hostdev configuration because network traffic must pass via the host kernel.



**Figure 14: KVM - libvirt bridged**

### KVM libvirt bridged - Configuration

- 1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.
- 2 In the host, configure the PF:
 

```
sfboot switch-mode=default pf-count=1
```

The sfboot settings shown above are the default (shipping state) settings for the SFN7000 series adapter. A cold reboot of the server is only required when changes are made using sfboot.
- 3 Create virtual machines:
 

VMs can be created from the standard Linux virt-manager GUI interface or the equivalent virsh command line tool. As root, run the command virt-manager from a terminal to start the GUI interface. A VM can also be created from an existing VM XML file.

*The following procedure assumes the VM is created.* The example procedure will create a bridge 'br1' and network 'host-network' to connect the VM to the Solarflare adapter via the bridge.
- 4 Define a bridge in /etc/sysconfig/network-scripts/ifcfg-br1
 

```
DEVICE=br1
TYPE=Bridge
BOOTPROTO=none
ONBOOT=yes
DELAY=0
NM_CONTROLLED=no
```

- 5** Associate the bridge with the required Solarflare PF (HWADDR) in a config file in /etc/sysconfig/network-scripts/ifcfg-eth4 (this example uses eth4):

```
DEVICE=eth4
TYPE=Ethernet
HWADDR=00:0F:53:21:00:60
BOOTPROTO=none
ONBOOT=yes
BRIDGE=br1
```

- 6** Bring up the bridge:

```
service network restart
```

- 7** The bridge will be visible in the host using the ifconfig command:

```
ifconfig -a
br1 Link encap:Ethernet HWaddr 00:0F:53:21:00:60
 inet6 addr: fe80::20f:53ff:fe21:60/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:170 errors:0 dropped:0 overruns:0 frame:0
 TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:55760 (54.4 KiB) TX bytes:468 (468.0 b)
```

- 8** Define a network in an XML file i.e. host-network.xml:

```
<network>
 <name>host-network</name>
 <forward mode='bridge' />
 <bridge name="br1"/>
</network>
```

- 9** Define and start the network using virsh net-<option> commands:

```
virsh net-define host-network.xml
Network host-network defined from host-network.xml

virsh net-start host-network
Network host-network started

virsh net-autostart host-network
Network host-network marked as autostarted

virsh net-list --all
Name State Autostart Persistent

default active yes yes
host-network active yes yes
```

- 10** On the host machine, edit the VM XML file:

```
virsh edit <vmname>
```

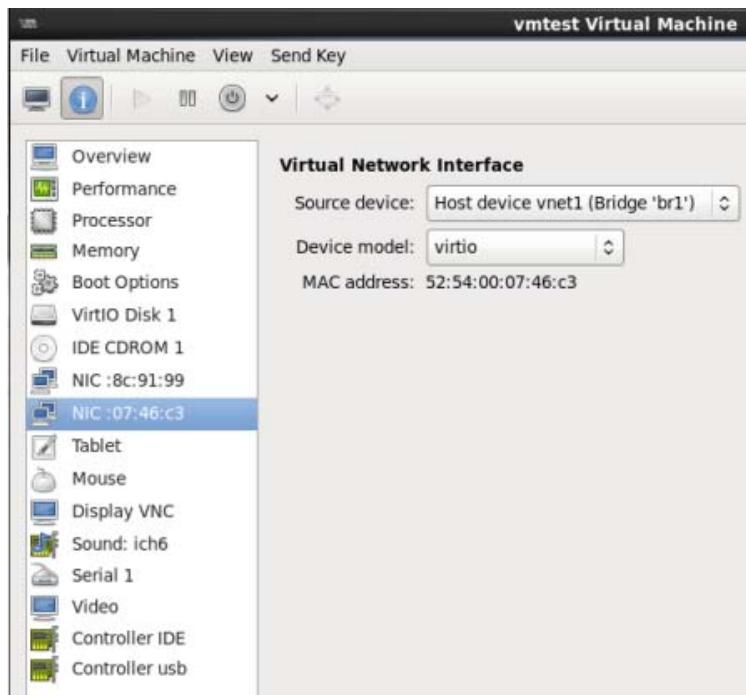
- 11** Add the network component to the VM XML file:

```
<interface type='network'>
 <source network='host-network' />
 <model type='virtio' />
</interface>
```

- 12** Restart the VM after editing the XML file.

```
virsh start <vmname>
```

- 13** The bridged interface is visible in the VM when viewed from the GUI Virtual Machine Manager:



**Figure 15: Virtual Machine Manager - Showing the network/bridged interface**

### XML Description

The following extract is from the VM XML file after the configuration procedure has been applied (line numbers have been added for ease of description):

```

1. <interface type='bridge'>
2. <mac address='52:54:00:96:0a:8a' />
3. <source bridge='br1' />
4. <model type='virtio' />
5. <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
6. </interface>
```

- 1** Interface type must be specified by the user as ‘bridge’.
- 2** The MAC address. If not specified by the user this will be automatically assigned a random MAC address by libvirt.
- 3** The source bridge as created in configuration step 4 above.
- 4** Model type must be specified by the user as ‘virtio’.
- 5** The PF PCIe address (as known by the guest) will be added automatically by libvirt.

For further information about the direct bridged configuration and XML formats, refer to the following links:

<http://libvirt.org/formatdomain.html#elementsNICSBridge>

## KVM Direct Bridged

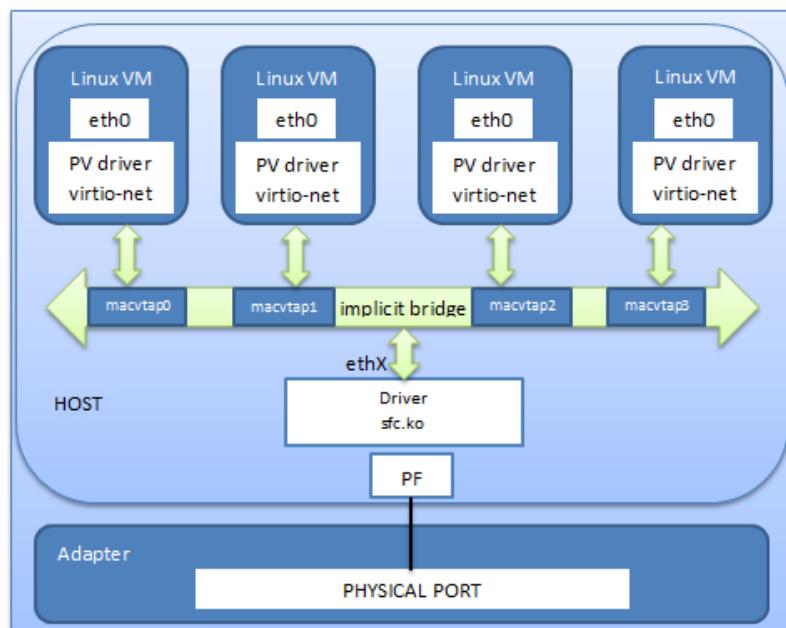
In this configuration multiple macvtap interfaces are bound over the same PF. For each VM created, libvirt will automatically instantiate a macvtap driver instance and the macvtap interfaces will be visible on the host.

Where the KVM libvirt bridged configuration uses the standard Linux bridge, a direct bridged configuration bypasses this providing an internal bridging function and increasing performance.

When using macvtap there is no link state propagation to the guest which is unable to identify if a physical link is up or down.

Macvtap does not currently forward multicast joins from the guests to the underlying network driver with the result that all multicast traffic received by the physical port is forwarded to all guests. Due to this limitation this configuration is not recommended for deployments that use a non-trivial amount of multicast traffic.

Guest migration is fully supported as there is no physical hardware state in the VM guests. A guest can be migrated to a host using a different VF or a host without an SR-IOV capable adapter.



**Figure 16: KVM - direct bridged**

## KVM direct Bridged - Configuration

**1** Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.

**2** In the host, configure the PF.

```
sfboot switch-mode=default pf-count=1
```

The sfboot settings shown above are the default (shipping state) settings for the SFN7000 series adapter. A cold reboot of the server is only required when changes are made using sfboot.

**3** Create virtual machines:

VMs can be created from the standard Linux virt-manager GUI interface or the equivalent virsh command line tool. As root, run the command virt-manager from a terminal to start the GUI interface. A VM can also be created from an existing VM XML file.

The following procedure assumes the VM is created. The example procedure will create an interface configuration file and connect the VM directly to the Solarflare adapter.

**4** Create a configuration file for the required Solarflare PF (HWADDR) in a config file in /etc/sysconfig/network-scripts/ifcfg-eth4 (this example uses eth4):

```
DEVICE=eth4
TYPE=Ethernet
HWADDR=00:0F:53:21:00:60
BOOTPROTO=none
ONBOOT=yes
```

**5** Bring up the interface:

```
service network restart
```

**6** On the host machine, edit the VM XML file:

```
virsh edit <vmname>
```

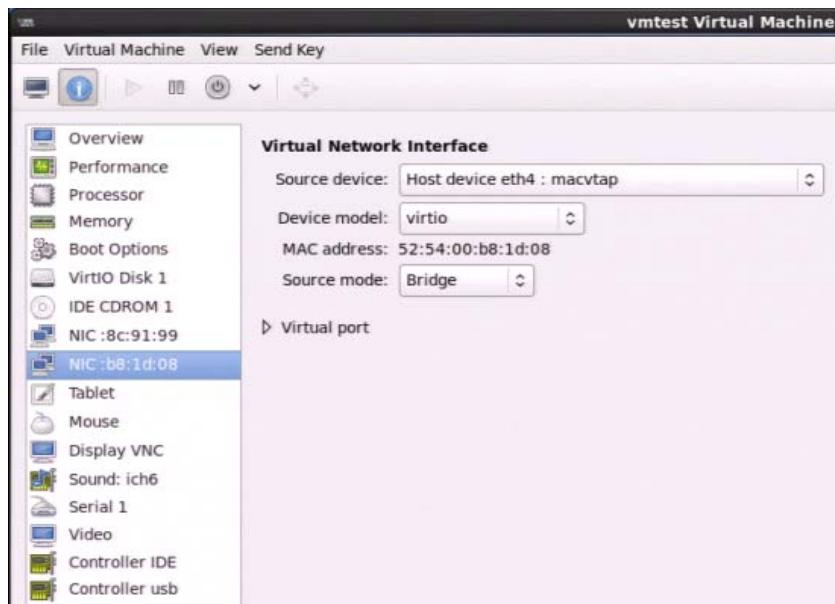
**7** Add the interface component to the VM XML file:

```
<interface type='direct'>
 <source dev='eth4' mode='bridge' />
 <model type='virtio' />
</interface>
```

**8** Restart the VM after editing the XML file.

```
virsh start <vmname>
```

**9** The bridged interface is visible when viewed from the GUI Virtual Machine Manager:



**Figure 17: Virtual Machine Manager - Showing the direct bridged interface**

### XML Description

The following extract is from the VM XML file after the configuration procedure has been applied (line numbers have been added for ease of description):

```

1. <interface type='direct'>
2. <mac address='52:54:00:db:ab:ca' />
3. <source dev='eth4' mode='bridge' />
4. <model type='virtio' />
5. <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
6. </interface>
```

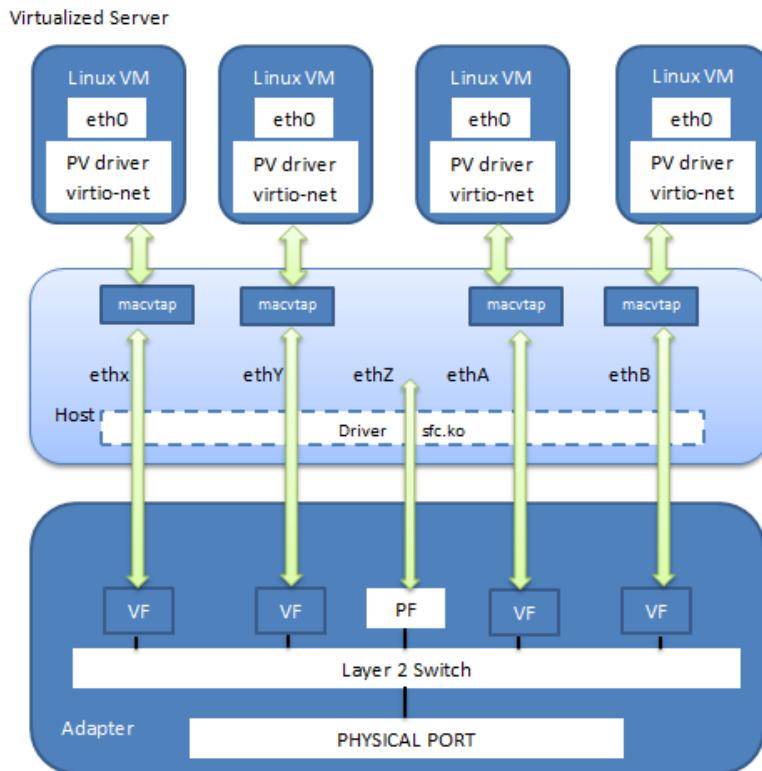
- 1 Interface type must be specified by the user as ‘direct’.
- 2 The MAC address. If not specified by the user this will be automatically assigned a random MAC address by libvirt.
- 3 The source dev is the interface identifier from the host - added by the user. The user should also specify the mode which must be ‘bridge’.
- 4 If not specified by the user, the model type will be automatically assigned by libvirt when the guest is started. Use virtio for best performance.
- 5 The PF PCIe address (as known by the guest) will be added automatically by libvirt

For further information about the direct bridged configuration and XML formats, refer to the following link:

<http://libvirt.org/formatdomain.html#elementsNICSBridge>

## KVM Libvirt Direct Passthrough

Using a libvirt direct-passthrough configuration, VFs are used in the host OS to provide network acceleration for guest VMs. The guest continues to use a paravirtualized driver and is unaware this is backed with a VF from the network adapter.



**Figure 18: SR-IOV VFs used in the host OS**

- The Solarflare net driver is bound over the top of each VF.
- Each macvtap interface is implicitly created by libvirt over a single VF network interface and is not visible to the host OS.
- Each macvtap instance builds over a different network interface - so there is no implicit macvtap bridge.
- Macvtap does not currently forward multicast joins from the guests to the underlying network driver with the result that all multicast traffic received by the physical port is forwarded to all guests. Due to this limitation this configuration is not recommended for deployments that use a non-trivial amount of multicast traffic.
- Guest migration is fully supported as there is no physical hardware state in the VM guests. A guest can be reconfigured to a host using a different VF or a host without an SR-IOV capable adapter.
- The MAC address from the VF is passed through to the para-virtualized driver.

- Because there is no VF present in a VM, Onload and other Solarflare applications such as SolarCapture cannot be used in the VM.

### KVM Libvirt Direct Passthrough - Configuration

- 1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.

- 2 In the host, configure the switch-mode, PF and VFs:

```
sfboot switch-mode=sriov pf-count=1 vf-count=4
```

A cold reboot of the server is required when changes are made using sfboot.

- 3 Create VFs in the host (example uses PF eth4):

```
echo 2 > /sys/class/net/eth4/device/sriov_numvfs
cat /sys/class/net/eth4/device/sriov_totalvfs
```

For Linux versions earlier than RHEL6.5 see [Enabling Virtual Functions on page 259](#).

- 4 PFs and VFs will be visible using the lspci command (VFs in **bold**):

```
lspci -D -d1924:
0000:03:00.0 Ethernet controller: Solarflare Communications SFC9120
0000:03:00.1 Ethernet controller: Solarflare Communications SFC9120
0000:03:00.2 Ethernet controller: Solarflare Communications Device 1903
0000:03:00.3 Ethernet controller: Solarflare Communications Device 1903
0000:03:00.4 Ethernet controller: Solarflare Communications Device 1903
0000:03:00.5 Ethernet controller: Solarflare Communications Device 1903
```

VFs will also be listed using the ifconfig command (abbreviated output below, from a dual port adapter, shows 2 x PF and 4 x VF. (pf-count=1 vf-count=2). VFs are shown in **bold**).

```
eth4 Link encap:Ethernet HWaddr 00:0F:53:21:00:60
eth5 Link encap:Ethernet HWaddr 00:0F:53:21:00:61
eth6 Link encap:Ethernet HWaddr AE:82:AB:C9:67:49
eth7 Link encap:Ethernet HWaddr 86:B4:C8:9E:27:D6
eth8 Link encap:Ethernet HWaddr 72:0B:C7:21:E1:59
eth9 Link encap:Ethernet HWaddr D2:B7:68:54:35:A5
```

- 5 Create virtual machines:

VMs can be created from the standard Linux virt-manager GUI interface or the equivalent virsh command line tool. As root, run the command virt-manager from a terminal to start the GUI interface. A VM can also be created from an existing VM XML file.

The following procedure assumes the VM is created. The example procedure will create an interface configuration file for each VF to be passed through to the VM.

- 6 For each VF to be passed through to a VM, create a configuration file in the /etc/sysconfig/network-scripts directory i.e. ifcfg-eth6:

```
DEVICE=eth6
TYPE=Ethernet
HWADDR=AE:82:AB:C9:67:49
BOOTPROTO=none
ONBOOT=yes
```

The above example is the file `ifcfg-eth6` and identifies the MAC address assigned to the VF. One file is required for each VF.

- 7 On the host machine, edit the VM XML file:

```
virsh edit <vmname>
```

- 8 Add the interface component to the VM XML file e.g:

```
<interface type='direct'>
 <source dev='eth6' mode='passthrough' />
 <model type='virtio' />
</interface>
```

One interface type component is required for each VF.

- 9 Restart the VM after editing the XML file.

```
virsh start <vmname>
```

The passed through VF interface is visible when viewed from the GUI Virtual Machine Manager

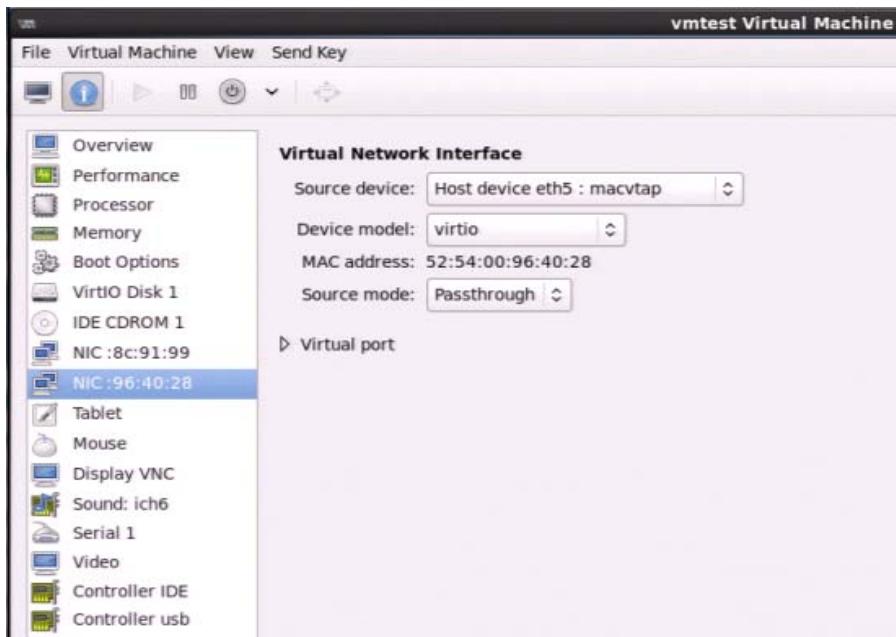


Figure 19: Virtual Machine Manager - Showing the passthrough interface

### XML Description

The following (example) extract is from the VM XML file after a VF has been passed through to the guest using the procedure above (line numbers have been added for ease of description):

```

1. <interface type='direct'>
2. <mac address='52:54:00:96:40:28' />
3. <source dev='eth6' mode='passthrough' />
4. <model type='virtio' />
5. <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
6. </interface>
```

- 1 A description of how the VF interface is managed - added by the user.
- 2 The MAC address. If not specified by the user this will be automatically assigned a random MAC address by the guest OS. The user can specify a MAC address when editing the XML file.
- 3 The source dev is the VF interface identifier - added by the user. The user should also specify the mode which must be 'passthrough'.
- 4 If not specified by the user, the model type will be automatically assigned by libvirt when the guest is started.
- 5 The VF PCIe address (as known by the guest) will be added automatically by libvirt.

For further information about the direct passthrough configuration and XML formats, refer to the following link:

<http://libvirt.org/formatdomain.html#elementsNICSDirect>

## KVM Libvirt Network Hostdev

Network Hostdev exposes VFs directly into guest VMs allowing the data path to fully bypass the host OS and therefore provides maximum acceleration for network traffic.

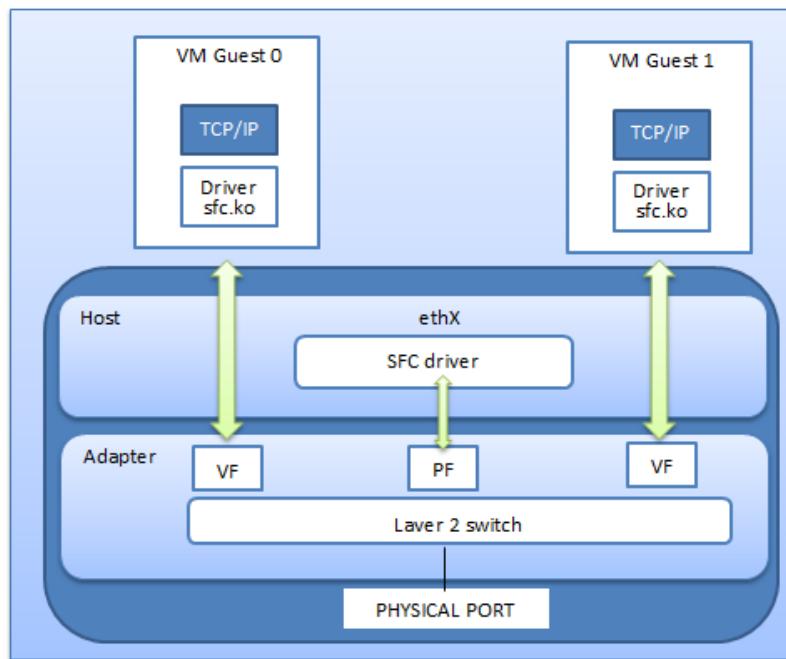


Figure 20: SR-IOV VFs passed to guests

- The hostdev configuration delivers the highest throughput and lowest latency performance. Because the guest is directly linked to the virtual function therefore directly connected to the underlying hardware.

- Migration is not supported in this configuration because the VM has knowledge of the network adapter hardware (VF) present in the server.
- The VF is visible in the guest. This allows applications using the VF interface to be accelerated using OpenOnload or to use other Solarflare applications such as SolarCapture.
- The Solarflare net driver (sfc.ko) needs to be installed in the guest.

### KVM Libvirt network hostdev - Configuration

- 1 Create the VM from the Linux virt-manager GUI interface or the virsh command line tool.
- 2 Install Solarflare network driver (sfc.ko) in the guest and host.
- 3 Create the required number of VFs:
 

```
sfboot switch-mode=sriov vf-count=4
```

A cold reboot of the server is required for this to be effective.
- 4 For the selected PF - configure the required number of VFs e.g:
 

```
echo 4 > /sys/class/net/eth8/device/sriov_numvfs
```
- 5 VFs will now be visible in the host - use ifconfig and the lscpi command to identify the Ethernet interfaces and PCIe addresses (VFs shown below in **bold** text):
 

```
lspci -D -d1924:
0000:03:00.0 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
0000:03:00.1 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
0000:03:00.2 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
0000:03:00.3 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
0000:03:00.4 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
0000:03:00.5 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
```
- 6 Using the PCIe address, unbind the VFs to be passed through to the guest from the host sfc driver e.g.:
 

```
echo 0000:03:00.5 > /sys/bus/pci/devices/0000\:03\:00.5/driver/unbind
```
- 7 Check that the required VF interface is no longer visible in the host using ifconfig.
- 8 On the host, stop the virtual machine:
 

```
virsh shutdown <vmname>
```
- 9 On the host, edit the virtual machine XML file:
 

```
virsh edit <vmname>
```
- 10 For each VF that is to be passed to the guest, add the following <interface type> section to the file identifying the VF PCIe address (use lscpi to identify PCIe address):
 

```
<interface type='hostdev' managed='yes'>
 <source>
 <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x5'/>
 </source>
</interface>
```

- 11** Restart the virtual machine in the host and VF interfaces will be visible in the guest:

```
virsh start <vmname>
```

The following (example) extract is from the VM XML file after a VF has been passed through to the guest using the procedure above (line numbers have been added for ease of description):

```
1. <interface type='hostdev' managed='yes'>
2. <mac address='52:54:00:d1:ec:85' />
3. <source>
4. <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x5' />
5. </source>
6. <alias name='hostdev0' />
7. <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
8. </interface>
```

## XML Description

- 1** A description of how the VF interface is managed - added by user.

When `managed=yes`, the VF is detached from the host before being passed to the guest and the VF will be automatically reattached to the host after the guest exits.

If `managed=no`, the user must call `virNodeDeviceDetach` (or use the command `virsh nodedev-detach`) before starting the guest or hot-plugging the device and call `virNodeDeviceReAttach` (or use command `virsh nodedev-reattach`) after hot-unplug or after stopping the guest.

- 2** The VF MAC address. If not specified by the user this will be automatically assigned a random MAC address by libvirt. The user can specify a MAC address when editing the XML file.
- 3** The VF PCIe address, this is the address of the VF interface as it is identified in the host. This should be entered by the user when editing the XML file.
- 4** If not specified by the user the alias name will be automatically assigned by libvirt. The user can supply an alias when editing the XML file.
- 5** The VF PCIe address (as known by the guest) will be added automatically by libvirt.

For further information about the hostdev configuration and XML formats, refer to the following link:

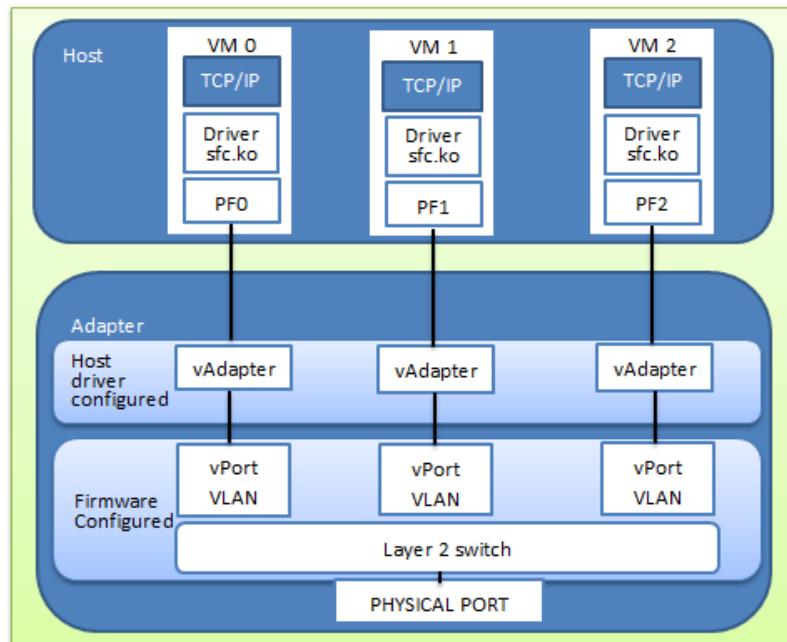
<http://libvirt.org/formatdomain.html#elementsNICSHostdev>

## 7.4 PF-IOV

Physical Function I/O Virtualization allows PFs to be passed to a VM. Although this configuration is not widely used, it is included here for completeness. This mode provides no advantage over “Network Hostdev” and therefore Solarflare recommends that customers deploy “Network hostdev instead of PF-IOV. PF-IOV does not use SR-IOV and does not require SR-IOV hardware support.

Each physical port is partitioned into a number of PFs with each PF passed to a different Virtual Machine (VM). Each VM supports a TCP/IP stack and Solarflare adapter driver (sfc.ko).

This mode allows switching between PFs via the Layer 2 switch function configured in firmware.



**Figure 21: PFIOV**

- Up to 16 PFs and 16 MAC addresses are supported *per adapter*.
- With no VLAN configuration, all PFs are in the same Ethernet layer 2 broadcast domain i.e. a packet broadcast from any one PF would be received by all other PFs.
- PF VLAN tags can optionally be assigned when creating PFs using the sfboot utility.
- The layer 2 switch supports replication of received/transmitted broadcast packets to all PFs and to the external network.
- The layer 2 switch supports replication of received/transmitted multicast packets to all subscribers.
- VFs are not supported in this mode.

## PF-IOV Configuration

The sfboot utility from the Solarflare Linux Utilities package (SF-107601-LS) is used to partition physical interfaces to the required number of PFs.

- Up to 16 PFs and 16 MAC addresses are supported per adapter.
- The PF setting applies to all physical ports. Ports cannot be configured individually.
- vf-count must be zero.

- 1 To partition all ports (example configures 4 PFs per port):

```
sfboot switch-mode=pfiov pf-count=4
```

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
```

```
eth5:
 Boot image Option ROM only
 Link speed Negotiated automatically
 Link-up delay time 5 seconds
 Banner delay time 2 seconds
 Boot skip delay time 5 seconds
 Boot type Disabled
 Physical Functions per port 4
 MSI-X interrupt limit 32
 Number of Virtual Functions 0
 VF MSI-X interrupt limit 8
 Firmware variant full feature / virtualization
 Insecure filters Disabled
 VLAN tags None
 Switch mode PFIOV
```

- 2 A reboot of the server is required for the changes to be effective.
- 3 Following reboot the PFs will be visible using the ifconfig or ip commands - each PF will have a unique MAC address. The lspci command will also identify the PFs:

```
lspci -d 1924:
```

```
07:00.0 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.1 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.2 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.3 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.4 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.5 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.6 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.7 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
```

### Identify PFs to Physical Port

The following command can be used to identify the physical port that a PF belongs to:

```
cat /sys/class/net/enp4s0f?/device/physical_port
0
1
0
1
0
1
0
```

From lspci output above, the primary PF of each physical port can be identified as they have the PCIe function 0 or 1: e.g. 07:00.0 and 07:00.1.

## 7.5 General Configuration

### Enabling Physical Functions

Use the sfboot utility from the Solarflare Linux Utilities package to create PFs. Up to 16 PF and 16 MAC addresses are supported *per adapter*.

```
sfboot pf-count=<N>
```

PF VLAN tags can also be assigned using sfboot.

```
sfboot pf-count=4, pf-vlans=100,110,200,210
```

The first VLAN tag is assigned to the first function, thereafter the tags are applied to PFs in MAC address order.

### Enabling Virtual Functions

On RHEL6.5 and later versions, VF creation is controlled through sysfs. Use the following commands (example) to create and view created VFs.

```
echo 2 > /sys/class/net/eth8/device/sriov_numvfs
cat /sys/class/net/eth8/device/sriov_totalvfs
```

On kernels not having this control via sysfs the Solarflare net driver module option `max_vfs` can be used to enable VFs. The `max_vfs` value applies to all adapters and can be set to a single integer i.e. all adapter physical functions will have the same number of VFs, or can be set to a comma separated list to have different numbers of VFs per PF.

The driver module parameter should be enabled in a user-created file (e.g. `sfc.conf`) in the `/etc/modprobe.d` directory and the sfc driver must be reloaded following changes.

```
options sfc max_vfs=4
options sfc max_vfs=2,4,8
```

When specified as a comma separated list, the first VF count is assigned to the PF with the lowest index i.e. the lowest MAC address, then the PF with the next highest MAC address etc. If the sfc driver option is used to create VFs, reload the driver:

```
modprobe -r sfc
modprobe sfc
```

VLAN tags can be dynamically assigned to VFs using libvirt commands, or using the ip command:

```
ip link vf NUM [mac LLADDR] [vlan VLANID]
```

To ensure VLAN tags persist after reboot, these can be configured in the VM XML file.

## Using OpenOnload in a Virtual Machine

Onload users should refer to the Onload User Guide (SF-104474-CD) for further information about using Onload in a KVM.

When Onload and the sfc net driver have been installed in the guest, the sfc driver module option num\_vis is used to allocate the required number of virtual interfaces. One VI is needed for each Onload stack using a VF.

Driver module options should be enabled in a user created file (e.g. sfc.conf) in the /etc/modprobe.d directory.

```
options sfc num_vis=<num>
```

Reload the driver after setting/changing this value:

```
onload_tool reload
```

## 7.6 Feature Summary

**Table 54: Feature Summary**

	Default	SRIOV	Partitioning	Partitioning + SRIOV	PFIOV
Number of PFs (per adapter)	num ports	num ports	≥num ports ≤16	≥num ports ≤16	≥num ports ≤16
All PFs (per port) must be on unique VLANs	N/A	N/A	Yes	Yes	No
Num VFs (per adapter)	0	>0, ≤240	0	>0, ≤240	0
Mode suitable for PF PCIe passthrough	No	No	No	No	Yes
Mode suitable for VF PCIe passthrough	No	Yes	No	Yes	No

**Table 54: Feature Summary**

	<b>Default</b>	<b>SRIOV</b>	<b>Partitioning</b>	<b>Partitioning + SRIOV</b>	<b>PFIov</b>
sfboot settings	switch-mode =default	switch-mode =sriov	switch-mode =partitioning	switch-mode =partitioning -with-sriov	switch-mode =pfiov
	pf-count=1	pf-count=1	pf-count>1	pf-count>1	pf-count>1
	vf-count=0	vf-count>0	vf-count=0	vf-count>0	vf-count=0
L2 switching between PF and associated VFs	N/A	Yes	N/A	Yes	N/A
L2 switching between PFs on the same physical port	N/A	N/A	No	No	Yes

## 7.7 Limitations

Users are advised to refer to the Solarflare net driver release notes for details of all limitations.

### Per Port Configuration

For initial releases, all PFs on a physical port have the same expansion ROM configuration where PXE/UEFI settings are stored. This means that all PFs will PXE boot or none will attempt to PXE boot. Users should ensure that a DHCP server responds to the first MAC address.

The PF (pf-count) configuration is a global setting and applies to all physical ports on an adapter. It is not currently possible to configure ports individually.

### PTP

PTP can only run on the primary physical function of each physical port and is not supported on VF interfaces.

# 8

# SR-IOV Virtualization Using ESXi

This chapter includes procedures for installation and configuration of Solarflare adapters for SR-IOV and Virtualization deployment using VMware® ESXi. For details of installation and configuration on VMware® platforms refer to [Solarflare Adapters on VMware on page 150](#).



**NOTE:** SR-IOV is not supported by the current native ESXi driver. You must instead use the legacy driver. See [Legacy Driver \(vmklinux API\) on page 151](#).

## 8.1 Introduction

This chapter describes SR-IOV and DirectPath I/O using the VMware ESXi hypervisor and Solarflare SFN7000 and SFN8000 series adapters.

SR-IOV enabled on Solarflare adapters provides accelerated cut-through performance and is compatible with hypervisor based services and management tools. The advanced design of Solarflare SFN7000 and SFN8000 series adapters incorporates a number of specific features when deploying the adapter into virtualized environments.

- PCIe Physical Functions (PF)

By partitioning the NIC, each physical network port can be exposed to the host as up to 16 PCIe Physical Functions (PF) with each having a unique interface name and unique MAC address.

- PCIe Virtual Functions (VF)

A PCIe physical function, PF, can support a configurable number of PCIe virtual functions. In total 240 VFs can be allocated between the PFs. The adapter can also support a total of 2048 MSI-X interrupts.

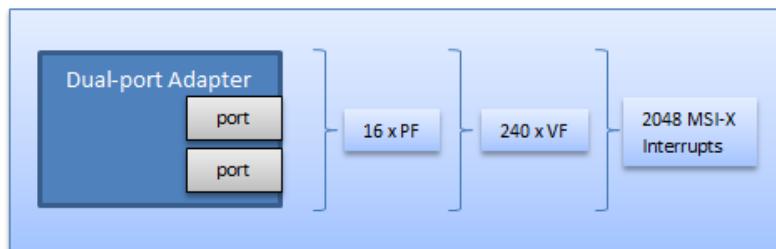


Figure 22: Per Adapter - Configuration Options

## Features Supported

On ESXi Solarflare adapters support the following deployments:

**Table 55: ESXi Virtualization Features**

Feature	Guest OS
VF Passthrough	Linux 6.5 to 7.x
PF Passthrough (DirectPath I/O)	Linux 6.5 to 7.x
	Windows Server 2012 R2

## Recommended Reading

The instructions in this chapter follow the procedures for VF and PF passthrough as documented in the [VMware Networking User Guide for ESXi 5.5](#).

## Platform Compatibility

SR-IOV and DirectPath I/O are not supported on all server platforms and users are advised to check server compatibility.

DirectPath I/O - PF Passthrough does not require platform SR-IOV support.

- Check for SR-IOV support in the VMware compatibility web page:  
<http://www.vmware.com/resources/compatibility/search.php>
- Ensure the BIOS has all SR-IOV/Virtualization options enabled.
- On a server with SR-IOV correctly configured, identify if Virtual Functions (VF) can be exposed to the host OS. Refer to sfboot options below for the procedure to configure VFs on the Solarflare adapter.

## BIOS

To use SR-IOV modes, SR-IOV must be enabled in the platform BIOS where the actual BIOS setting can differ between machines, but may be identified as SR-IOV, IOMMU or VT-d and VT-x on an Intel platform.

There may be other BIOS options which should be enabled to support SR-IOV, for example on DELL servers the following BIOS option must also be enabled:

Integrated Devices, SR-IOV Global Enable

*Users are advised to consult the server vendor BIOS options documentation.*

## Supported Platform OS

### Host

- VMware ESXi 5.5 and 6.0
- Solarflare v4.7 (or later) net drivers

### Guest VM

- Red Hat Enterprise Linux 6.5 to 7.x
- Windows Server 2012 R2
- Solarflare v4.5 (or later) net drivers

Acceleration of Virtual Machines (VM) running guest operating systems not listed above are not currently supported.

## Supported Adapters

All Solarflare adapters fully support SR-IOV.

## Solarflare Driver/Firmware

Features described in the chapter require the following (minimum) Solarflare driver and firmware versions.

```
ethtool -i vmnic<N>
driver: sfc
version: 4.7.0.1031
firmware-version: 4.7.0.1020 rx0 tx0
```

The adapter must be using the *full-feature* firmware variant which can be selected using the sfboot utility and confirmed with **rx0 tx0** appearing after the version number in the output from ethtool as shown above.

The firmware update utility (sfupdate) and boot ROM configuration tool (sfboot) are available in the Solarflare Linux Utilities package (SF-107601-LS issue 36 or later).

## 8.2 Configuration Procedure - SR-IOV

Use the following procedure to configure the adapter and server for SR-IOV.

- [Install the Solarflare Driver on the ESXi host on page 265](#)
- [Install Solarflare Utilities on the ESXi host on page 266](#)
- [Install Solarflare Drivers in the Guest on page 265](#)
- [Configure VFs on the Host/Adapter on page 268](#)
- [Virtual Machine on page 269](#)
- [vSwitch and Port Group Configuration on page 271](#)
- [VF Passthrough on page 275](#)

## 8.3 Configuration Procedure - DirectPath I/O

Use the following procedure to configure the adapter and server for PF passthrough.

- [Install the Solarflare Driver on the ESXi host on page 265](#)
- [Install Solarflare Utilities on the ESXi host on page 266](#)
- [Install Solarflare Drivers in the Guest on page 265](#)
- [Partition the Adapter on page 281](#)
- [Virtual Machine on page 269](#)
- [Make PF Passthrough Devices available to the Guest on page 282](#)
- [Assign PF Passthrough Devices to the VM on page 283](#)

## 8.4 Install Solarflare Drivers in the Guest

For both VF and PF passthrough configurations, the Solarflare adapter driver must be installed in the virtual machine guest OS.

Drivers are available from the Solarflare download site for Linux and Windows guests: <https://support.solarflare.com/>.

Driver installation procedures on a guest are the same as installation for a host.

## 8.5 Install the Solarflare Driver on the ESXi host

Solarflare VMware ESXi drivers are available from: <https://support.solarflare.com/>.

Refer to [Solarflare Adapters on VMware on page 150](#) for instructions to install VIB driver packages through the CLI.

## 8.6 Install Solarflare Utilities on the ESXi host

Solarflare utilities - including sfboot, sfupdate and sfkey are distributed in the Solarflare Linux Utilities package (SF-107601-LS issue 36 or later) from:

<https://support.solarflare.com/>.

Refer to [Install CIM Provider on page 173](#) for instructions to install the utilities on the ESXi host server.



**NOTE:** The Solarflare driver must be installed before using sfboot or any of the utilities.

### sfboot - Configuration Options

The sfboot utility allows the user to configure:

- The number of PFs exposed per port to host and/or Virtual Machine (VM).
- The number VFs exposed per port to host and/or Virtual Machine (VM).
- The number of MSI-X interrupts assigned to each PF or VF.
- Firmware Variant and switch mode.

To check the current adapter configuration run the sfboot command:

```
sfboot
```

```
Solarflare boot configuration utility [v4.7.0]
Copyright Solarflare Communications 2006-2015, Level 5 Networks 2002-2005
```

```
vmnic6:
 Boot image Disabled
 Physical Functions on this port 1
 PF MSI-X interrupt limit 32
 Virtual Functions on each PF 4
 VF MSI-X interrupt limit 16
 Port mode 2x10G
 Firmware variant Full feature / virtualization
 Insecure filters Enabled
 MAC spoofing Disabled
 VLAN tags None
 Switch mode SR-IOV
 RX descriptor cache size 32
 TX descriptor cache size 16
 Total number of VIs 2048
 Rate limits None
 Event merge timeout 8740 nanoseconds
```

An alternative bootable ISO image of the Solarflare Utilities is available from the Solarflare download site under **Downloads > Linux > Misc**.

## Firmware Variant

The firmware variant must be set to full-feature / virtualization.

```
sfboot --adapter=vmnic6 firmware-variant=full-feature
```

## SR-IOV (VF Passthrough) sfboot Settings

The following example creates 4 VFs for each physical port.

```
sfboot switch-mode=sriov pf-count=1 vf-count=4
```

When used without the --adapter option, the command applies to all adapters

## DirectPath I/O (PF Passthrough) sfboot Settings

The following example partitions the NIC so that each physical port is exposed as 4 PCIe PFs.

```
sfboot switch-mode=partitioning pf-count=4 vf-count=0
```

*For some configuration option changes using sfboot, the server must be power cycled (power off/power on) before the changes are effective. sfboot will display a warning when this is required.*

## 8.7 Configure VFs on the Host/Adapter

The following host procedure is used to expose VFs from the Solarflare adapter.

- 1 Set the sfc driver module parameter for the required number of VFs:

```
esxcli system module parameters set -m sfc -p max_vfs=4
esxcli system module parameters list -m sfc
```

- 2 Use sfboot to create VFs on the adapter:

```
sfboot switch-mode=sriov vf-count=4
```

**The server must be restarted (power off/power on) for these changes to take effect.**

- 3 Following restart - list VFs exposed in the host:

```
lspci | grep Solarflare
0000:04:00.0 Network controller: Solarflare SFC9120 [vmmic6]
0000:04:00.1 Network controller: Solarflare SFC9120 [vmmic7]

0000:04:00.2 Network controller: Solarflare [PF_0.4.0_VF_0]
0000:04:00.3 Network controller: Solarflare [PF_0.4.0_VF_1]
0000:04:00.4 Network controller: Solarflare [PF_0.4.0_VF_2]
0000:04:00.5 Network controller: Solarflare [PF_0.4.0_VF_3]

0000:04:00.6 Network controller: Solarflare [PF_0.4.1_VF_0]
0000:04:00.7 Network controller: Solarflare [PF_0.4.1_VF_1]
0000:04:01.0 Network controller: Solarflare [PF_0.4.1_VF_2]
0000:04:01.1 Network controller: Solarflare [PF_0.4.1_VF_3]
```

The example above is a dual-port adapter. Each physical port is exposed as 1 PF and 4 VFs (PFs are shown in bold text).

## 8.8 Virtual Machine

The procedures in the Chapter assume the VM has already been created. Users should consult the VMware documentation to create the VM. The recommended method is to use the VMware vSphere Web Client:

The VM must be compatible with version 10 (or later).

### VM Compatibility

The VM must be compatible with ESXi 5.5 (or later). When the VM is not compatible, the following procedure via the vSphere Web Client will upgrade compatibility:

- Locate the VM from the listed hosts in the Web Client.
- Right click the VM > **Edit Settings**
- Under the **Virtual Hardware** tab > **Upgrade**
- Check the “Schedule VM Compatibility Upgrade” check box
- Select **ESXi 5.5 and later** from the drop down list
- Click **OK** to close the dialog
- Shutdown and restart the guest

After the VM has been shutdown and restarted, the compatibility will be displayed under the Settings tab:

VM Hardware	
▶ CPU	1 CPU(s), 0 MHz used
▶ Memory	2048 MB, 0 MB used
▶ Hard disk 1	16.00 GB
▶ Network adapter 1	VM Network (disconnected)
CD/DVD drive 1	Power on VM to connect
Floppy drive 1	Power on VM to connect
▶ Video card	8.00 MB
▶ Other	Additional Hardware
Compatibility	ESXi 5.5 and later (VM version 10) (upgrade succeeded)

## 8.9 List Adapters - Web Client

To list available adapters.

Navigate to the **Host > Networking > Physical Adapters**



Device	SR-IOV Status	Number of VFs	Switch	Actual Speed
vmnic2	Not supported	--	--	Down
vmnic3	Not supported	--	--	Down
<b>Solarflare SFC9120</b>				
vmnic6	Enabled	8 (4 currently available)	vSwitch1	10000 Mb
vmnic7	Disabled	--	--	10000 Mb

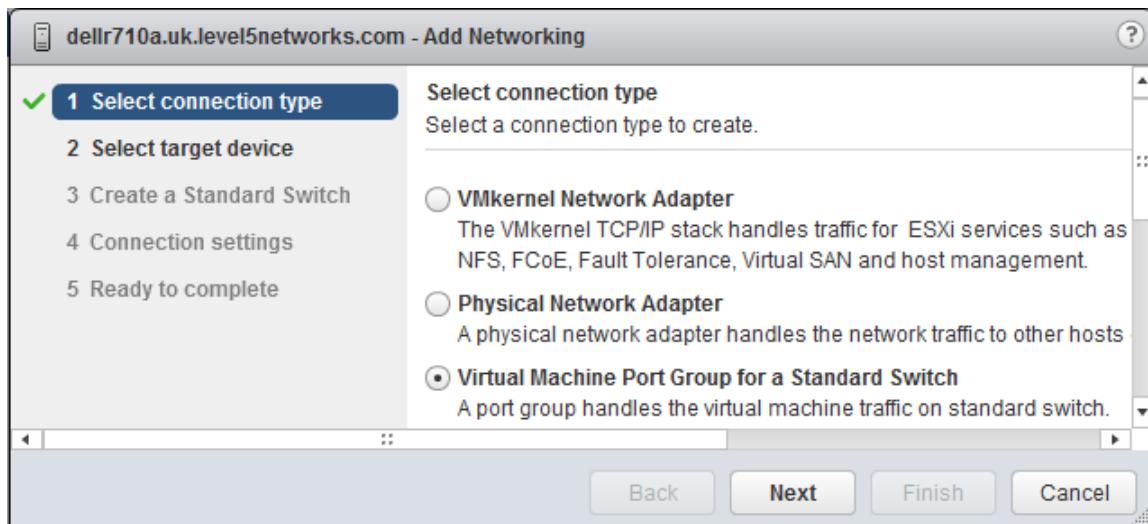
Selecting an adapter and clicking the edit (pencil) icon allows adapter settings to be edited.

## 8.10 vSwitch and Port Group Configuration

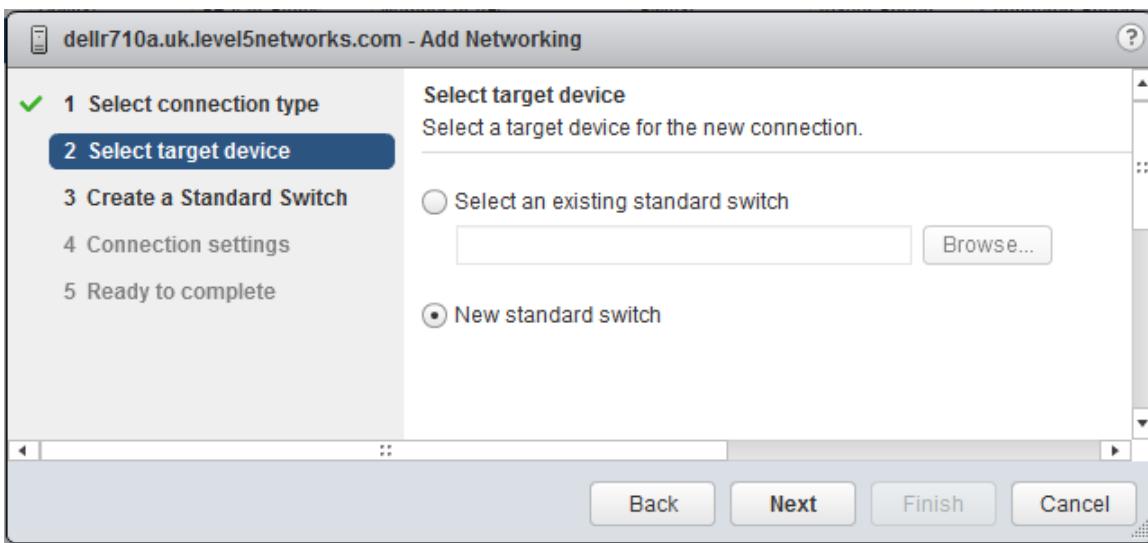
Using the vSphere Web Client, navigate to the host.

Right click and select **All vCenter Actions > Add Networking** to display the **Add Networking** wizard.

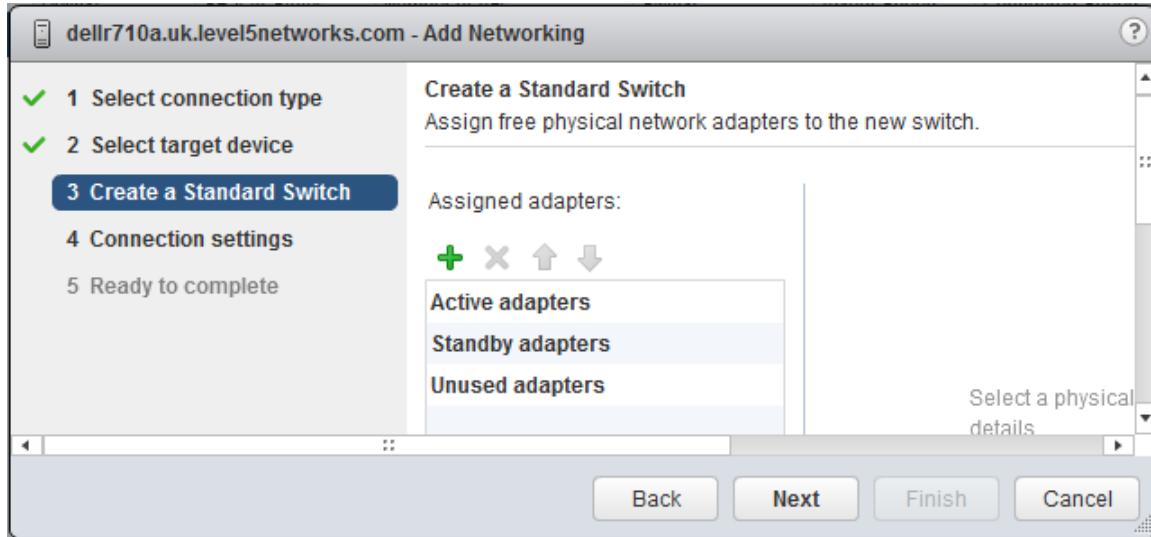
Select **Virtual Machine Port Group for a Standard Switch**, click **Next**.



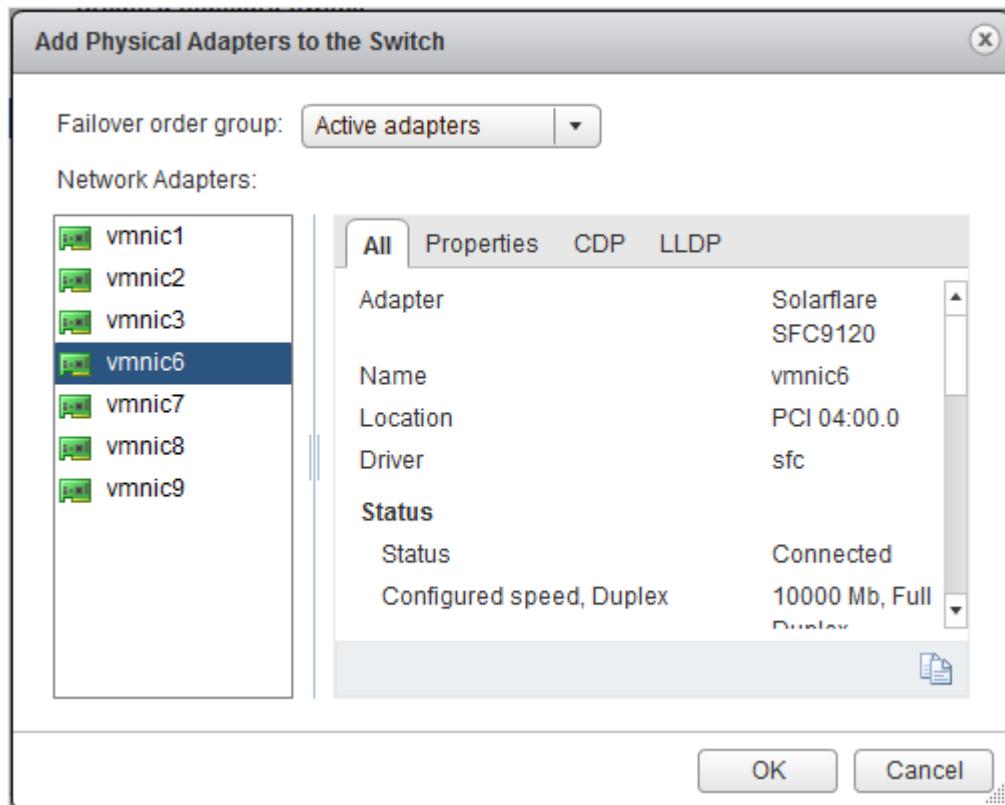
In **Select target device** - select an existing standard switch or create a new switch, click **Next**.



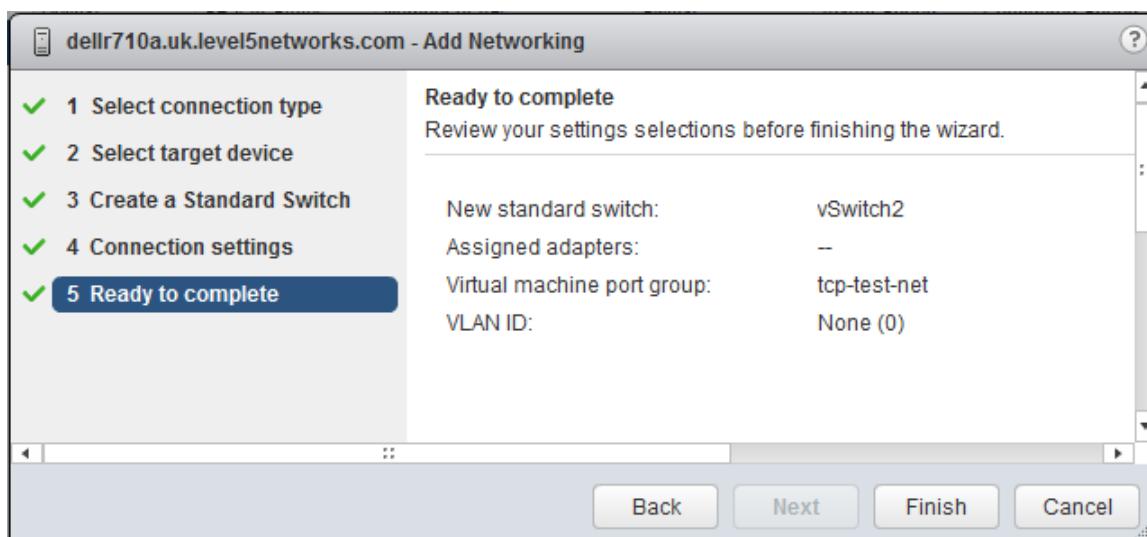
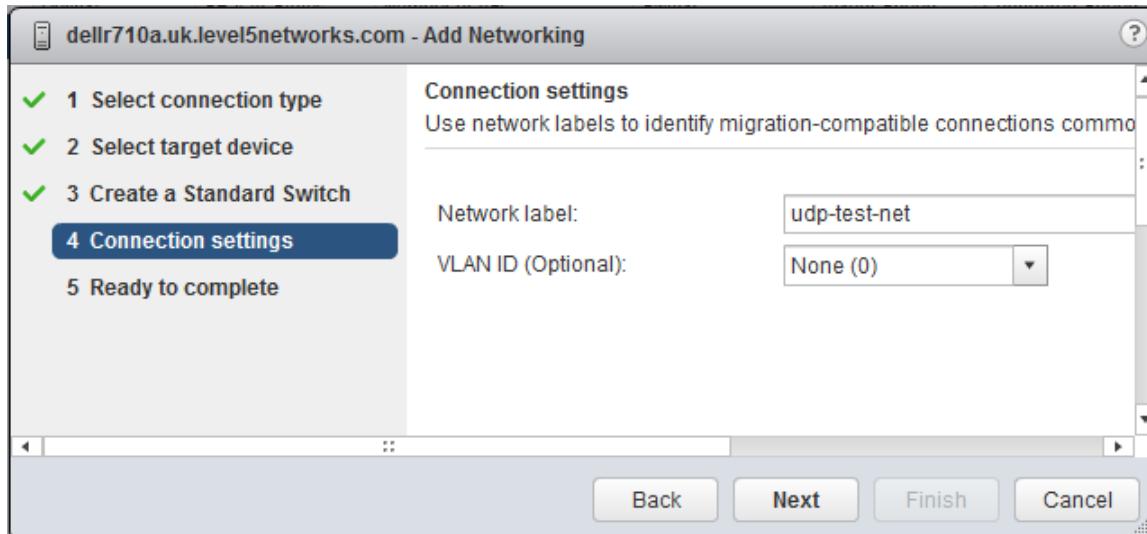
When creating a new standard switch, click the + icon under Assigned adapters.



Select the required physical adapter(s) which, as uplinks, will connect the vswitch with a network.



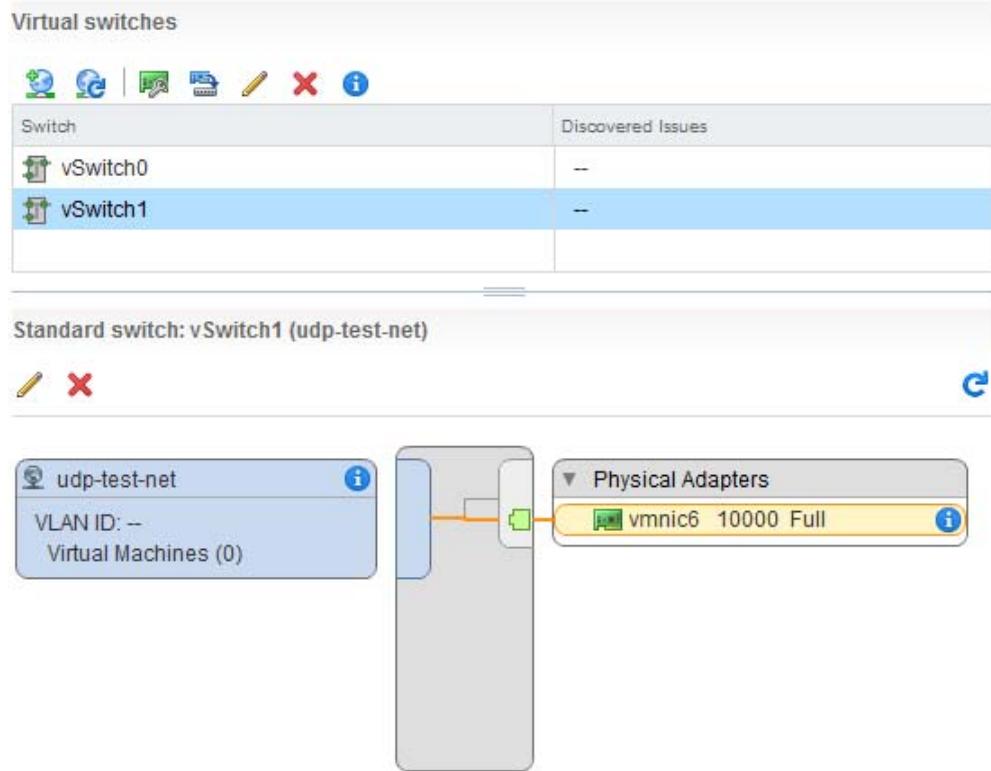
Label the portgroup and assign a network label and VLAN ID if required. VFs will later be assigned to the same portgroup and will be able to send/receive traffic through the uplink adapter(s).



Review settings and click **Finish** to complete.

The vSwitch and associated uplink adapter(s) topology can be viewed as follows:

Select the host > **Manage** tab > **Networking** > **Virtual switches**



## 8.11 VF Passthrough

The following procedure uses the VMware vSphere Web Client to configure SR-IOV VF passthrough.

The procedure is documented in the [VMware Networking User Guide for ESXi 5.5](#).

### Assumptions

The procedure assumes the following tasks are complete:

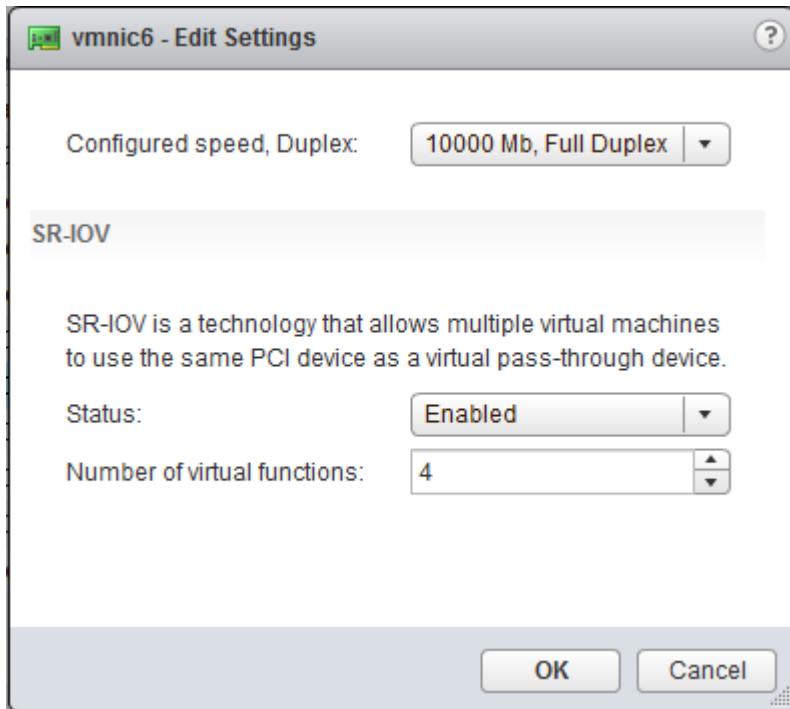
- A VM has been created and the guest OS installed.
- A Solarflare SFN7000 or SFN8000 series adapter is physically installed on the host.
- The Solarflare VMware net driver package is installed on the host.
- A Solarflare net driver is installed in the VM.
- The Solarflare adapter exposes VFs to the host OS.
- On the host a vswitch has been created and the PF(s) from the Solarflare adapter are selected as uplinks.

### Enable SR-IOV on the Host Adapter

- 1 In the vSphere Web Client, navigate to the host.
- 2 Select the **Manage** tab > **Networking** > **Physical Adapters** to list all available host adapters.
- 3 Select the required Solarflare adapter, then select the pencil (**edit**) icon.

Physical adapters					
Device	Switch	SR-IOV Status	Number of VFs	MAC Address	
<b>Solarflare SFC9120</b>					
vmnic6	vSwitch1	Enabled	4	00:0f:53:01:7d:00	
vmnic7	vSwitch1	Disabled	--	00:0f:53:01:7d:01	
<b>Solarflare SFC9140</b>					
vmnic8	--	Disabled	--	00:0f:53:20:9a:20	
vmnic9	--	Disabled	--	00:0f:53:20:9a:21	

- 4 From the adapter **Edit Settings** dialog enable SRIOV and specify the number of VFs which can be used by the VM.

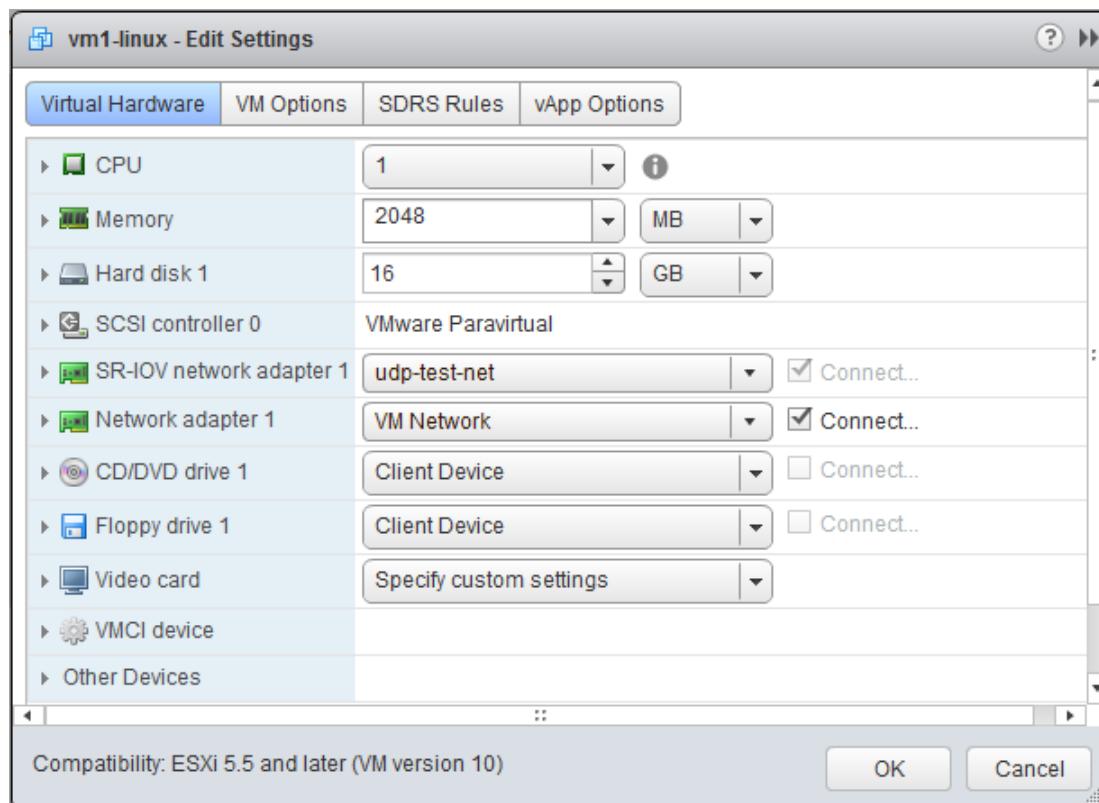


**NOTE:** The number of virtual functions should not exceed the value set by the max-vfs Solarflare driver option.

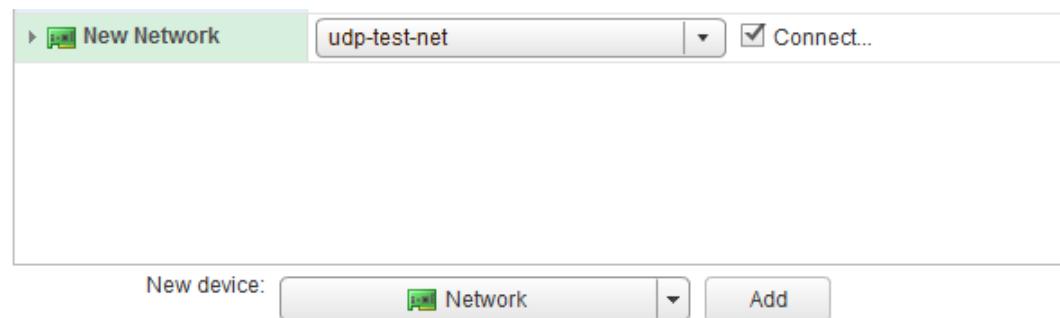
- 5 **The host must be restarted following adapter settings changes.** Select and right click the host in the Web Client window for reboot options.

## Assign a VF as a SR-IOV Passthrough adapter to the VM

- 1 In the vSphere Web Client, navigate to the VM.
- 2 Power **OFF** the VM.
- 3 Select the **Manage** tab > **Settings** > **VM Hardware**, then click the **Edit** button to display the **VM - Edit Settings** window.

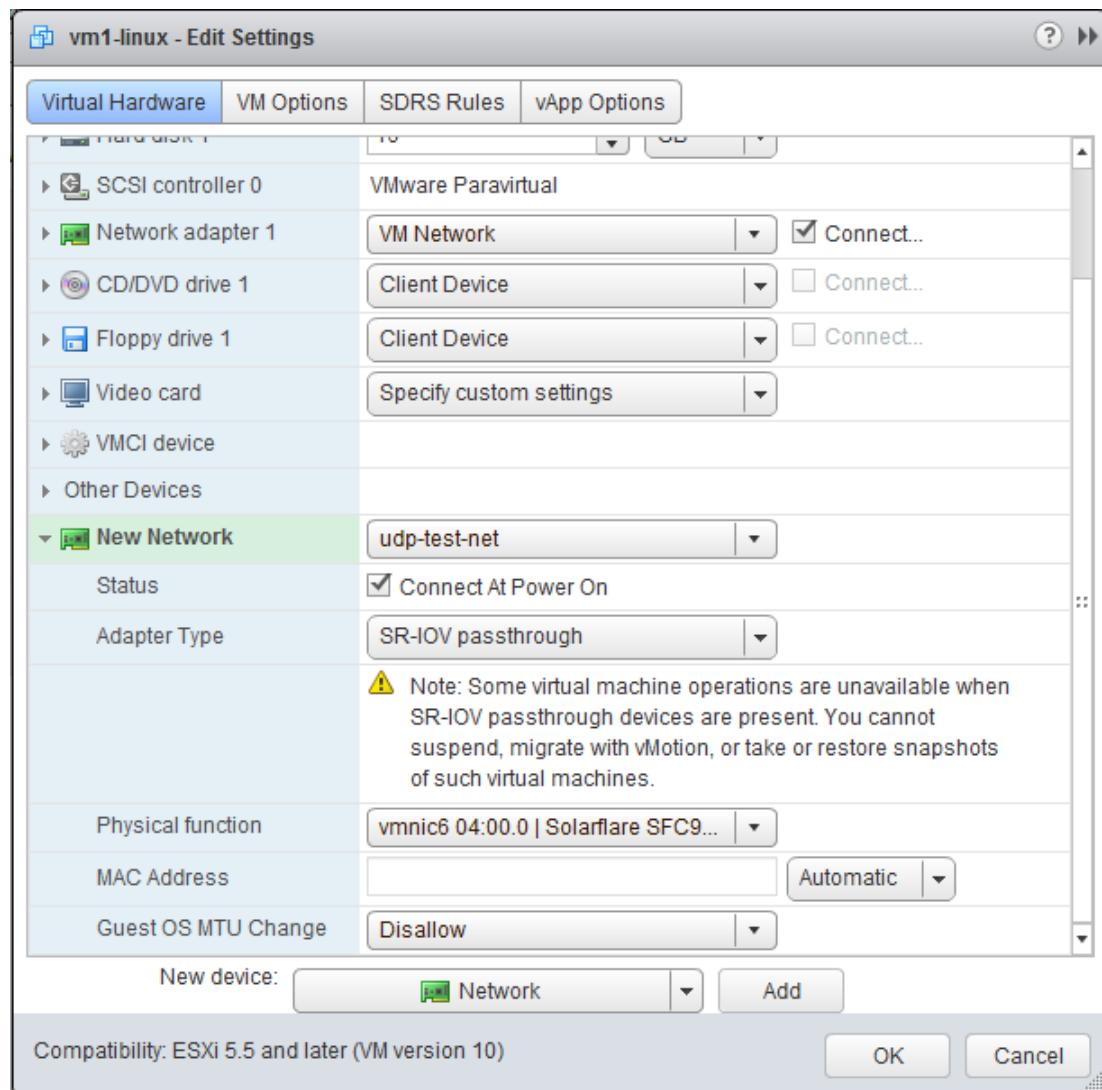


- 4 To add a VF, select **Network** from the **New device** drop down list and click the **Add** button.



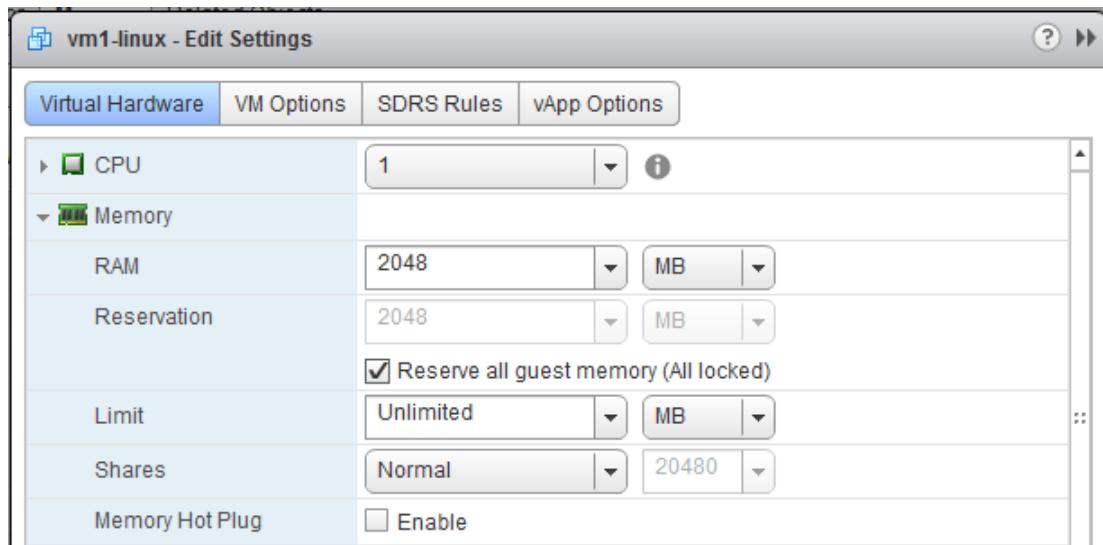
- 5 Expand the **New Network** section:
- 6 Connect the VM to a port group identifying the network the VF is to connect to. If there is no uplink associated with the portgroup, VFs attached to the same portgroup can sent between themselves, but not to any network.

- 7 Select Adapter Type as **SR-IOV passthrough**.
- 8 Select the physical adapter associated with the port group and which will back the passthrough virtual machine adapter. ESXi will select an unused VF associated with this adapter.



- 9 From the Edit Settings window, expand the **Memory** section and check the **Reserve all guest memory (All locked)** check box.

Reserving all memory is required for both SR-IOV and DirectPath configurations because the adapter PF/VF must be able to DMA to the guest memory which needs to be present in physical RAM.



- 10 From the **Edit Settings** window, click the **OK** button to close.

- 11** With the VM selected, click the **Manage** tab and the **VM Hardware** option to view hardware configuration. The Solarflare adapter PF and VF are listed in this window.

VM Hardware	
▶ CPU	1 CPU(s), 0 MHz used
▶ Memory	2048 MB, 0 MB used
▶ Hard disk 1	16.00 GB
▼ SR-IOV network adapter 1	
MAC Address	00:50:56:8b:5c:c0
DirectPath I/O	Not supported 
Network	VM Network 2 (connected)
Allow Guest MTU Change	Disallow
Physical Function	vmnic6 04:00.0   Solarflare SFC9120
Virtual Function	04:00.2   Solarflare <class> Ethernet controller
▶ Network adapter 1	VM Network (connected)
 CD/DVD drive 1	Disconnected
 Floppy drive 1	Disconnected
▶ Video card	8.00 MB
▶ Other	Additional Hardware
Compatibility	ESXi 5.5 and later (VM version 10) (upgrade succeeded)

## **Listing Passthrough Devices in a Linux Guest**

When the VM has been restarted the passed through VF devices are visible in the Linux guest using both `lspci` and the `ifconfig` commands.

```
[root@localhost ~]# lspci | grep Solarflare
0b:00.0 Ethernet controller: Solarflare Communications Device 1903
13:00.0 Ethernet controller: Solarflare Communications Device 1903
[root@localhost ~]# _
```

## 8.12 DirectPath I/O

DirectPath I/O allows a VM access to PF on platforms having an IOMMU. Platform support for SR-IOV is not required.

Solarflare SFN7000 and SFN8000 series adapters can be partitioned into multiple PCIe PFs, supporting up to 16 PCIe physical functions.

For details of NIC Partitioning see [NIC Partitioning on page 59](#).

### Partition the Adapter

The Solarflare NIC can be partitioned to expose up to 16 PFs using the sfboot command from the ESXi host command line interface:

```
sfboot --adapter=vmnic6 vf-count=0 pf-count=4 switch-mode=partitioning
```

**The server must be cold-power cycled.** When the server restarts, PFs will be visible in the host:

```
lspci -vvv | grep Solarflare
```

```
0000:07:00.0 Ethernet controller Network controller: Solarflare SFC9140 [vmnic8]
0000:07:00.1 Ethernet controller Network controller: Solarflare SFC9140 [vmnic9]
0000:07:00.2 Ethernet controller Network controller: Solarflare SFC9140 [vmnic4]
0000:07:00.3 Ethernet controller Network controller: Solarflare SFC9140 [vmnic5]
0000:07:00.4 Ethernet controller Network controller: Solarflare SFC9140 [vmnic10]
0000:07:00.5 Ethernet controller Network controller: Solarflare SFC9140 [vmnic11]
0000:07:00.6 Ethernet controller Network controller: Solarflare SFC9140 [vmnic12]
0000:07:00.7 Ethernet controller Network controller: Solarflare SFC9140 [vmnic13]
```

In the above example a dual-port adapter is partitioned to expose 4 PFs per physical port.

## Make PF Passthrough Devices available to the Guest

This procedure uses the vSphere Web Client and follows the procedure from the vmware documentation for DirectPath I/O:

[VMware Networking User Guide for ESXi 5.5](#)

- 1 Navigate to the host. Select the **Manage** tab > **Settings** option.
- 2 Under the **Hardware** section, select **PCI Devices**.

DirectPath I/O PCI Devices Available to VMs		
ID	Status	Vendor Name
04:00.2	Available	Solarflare
00:09.0	Not Configurable	Intel Corporation
07:00.3	Available	Solarflare
07:00.7	Available	Solarflare
07:00.0	Unavailable	Solarflare
07:00.2	Available	Solarflare
07:00.5	Available	Solarflare
07:00.6	Available	Solarflare
07:00.1	Unavailable	Solarflare
07:00.4	Available	Solarflare

- 3 Right click any device listed and select **Edit** from the pop-up menu. Or click the edit (pencil) icon.
- 4 From the **All PCI Devices** window, tick the check-box of the required PF devices:

All PCI Devices				
ID	Status	1 ▲	Vendor Name	Device Name
07:00.5	Available		Solarflare	SFC9140
07:00.7	Available		Solarflare	SFC9140
07:00.6	Available		Solarflare	SFC9140
07:00.2	Available		Solarflare	SFC9140
07:00.3	Available		Solarflare	SFC9140
07:00.4	Available		Solarflare	SFC9140
07:00.0	Unavailable		Solarflare	SFC9140
07:00.1	Unavailable		Solarflare	SFC9140



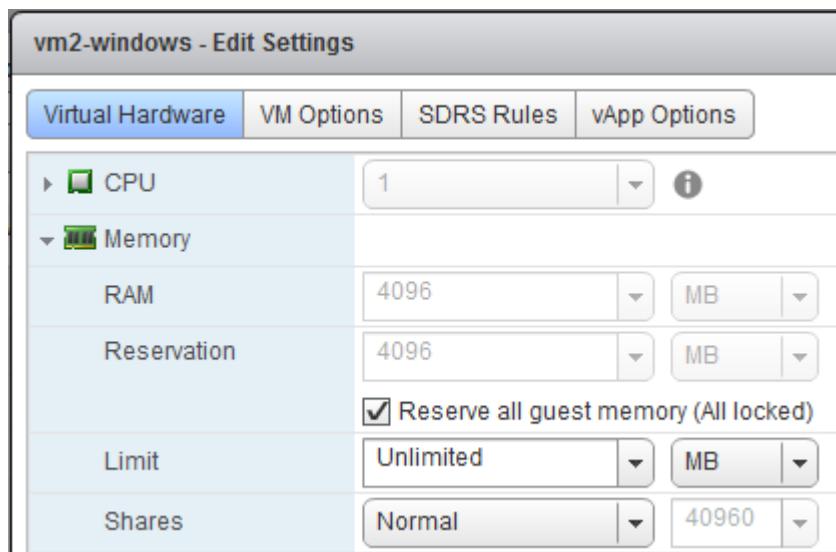
**CAUTION:** The Primary PF from each physical port cannot be passed through to a guest. PFO, having configuration privileges, is used by the sfc driver in the hypervisor and should not be passed to a VM.



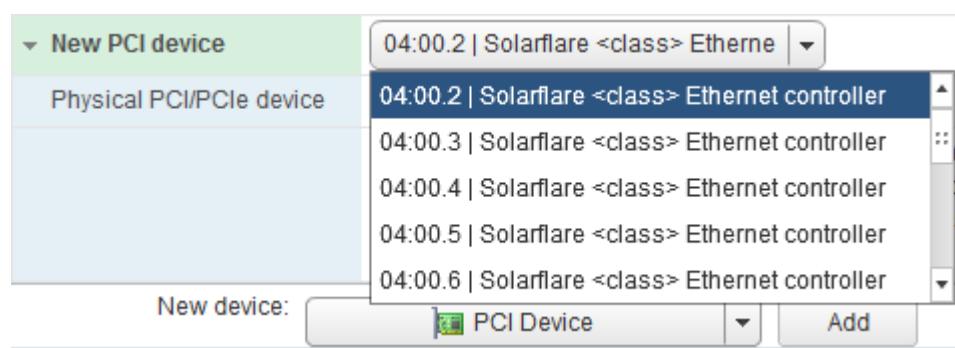
**NOTE:** When selecting new devices these will be marked as “Available (pending)”. The host must be restarted to effect the changes.

## Assign PF Passthrough Devices to the VM

- 1 Locate and select the VM in the vSphere Web Client.
- 2 Power **OFF** the VM.
- 3 Select the **Manage** tab, select **Settings** then **VM Hardware**, then click the **Edit** button to open the VM Edit Settings window.
- 4 Select the **Virtual Hardware** tab and then the **Memory** option:



- 5 For the **Memory** option, set the **Limit** to “Unlimited”.
- 6 Next, select **PCI Device** from the **New device** drop-down list, then click the **Add** button.



- 7 Select the required Solarflare PF device to be passed through.



- 8** Repeat - selecting **New Device** then **Add** for each PF to be passed through.
- 9** Click the **OK** button when done.
- 10** Power ON the VM.

# 9

# Solarflare Boot Manager

## 9.1 Introduction

Solarflare adapters support the following Preboot Execution Environment (PXE) options enabling diskless systems to boot from a remote target operating system:

- Solarflare Boot Manager (based on gPXE) - the default PXE client installed on all Solarflare adapters.
- UEFI network boot.
- iPXE - supported on Solarflare XtremeScale™ and X2 series adapters.

The Solarflare Boot Manager complies with the PXE 2.1 specification.

### Boot Manager Exposed

Solarflare adapters are shipped with boot ROM support ‘exposed’, that is the Boot Manager runs during the machine bootup stage allowing the user to enter the setup screens (via **Ctrl+B**) and enable PXE support when this is required.

The Boot Manager can also be invoked using the Solarflare supplied **sfboot** utility. For instructions on the **sfboot** method refer to the **sfboot** commands in the relevant OS specific sections of this user guide.

Using the **sfboot** utility, the **boot-image** options identify which boot images are exposed on the adapter during boot time. The **boot-image=uefi** option allows the Solarflare UEFI driver to be loaded by the UEFI platform which can be configured to PXE boot from the Solarflare adapter.

The **boot-type** options allows the user to select PXE boot or to disable PXE boot. This is effective on the next reboot.



**NOTE:** If network booting is not required, startup time can be decreased when the **boot-image** option is ‘disabled’ so that the **CTRL-B** option is not exposed during system startup.

### PXE Enabled

Some Solarflare distributors are able to ship Solarflare adapters with PXE boot enabled. Customers should contact their distributor for further information.

Solarflare XtremeScale and X2 series adapters are shipped with the PXE boot enabled and set to PXE boot.



**NOTE:** PXE, UEFI network boot is not supported for Solarflare adapters on IBM System p servers.

## Firmware Upgrade - Recommended

Before configuring the Solarflare Boot Manager, it is recommend that servers are running the latest adapter firmware which can be updated as follows:

- From a Windows environment use the supplied Command Line Tool sfupdate.exe.
- From a Linux or VMware environment update the firmware via sfupdate. See OS specific sections of this document for sfupdate commands.

This section covers the following subjects.

- [Solarflare Boot Manager on page 286](#)
- [iPXE Support on page 287](#)
- [sfupdate Options for PXE upgrade/downgrade on page 287](#)
- [Starting PXE Boot on page 289](#)
- [iPXE Image Create on page 293](#)
- [Multiple PF - PXE Boot on page 295](#)

## 9.2 Solarflare Boot Manager

**The standard Solarflare Boot Manager, based on gPXE, is supported on all Solarflare adapters.**

The boot ROM agent, pre-programmed into the adapter's flash image, runs during the machine bootup stage and, if enabled, supports PXE booting the server.

The Boot Manager can be configured using its embedded setup screens (entered via **Ctrl+B** during system boot) or via the Solarflare-supplied sfboot utility.

The boot ROM agent firmware version can be upgraded using the Solarflare-supplied **sfupdate** utility. Refer to the OS specific sections of this document for details of sfupdate commands.

The use of the Solarflare Boot Manager is fully supported by Solarflare (including meeting any SLA agreements in place for prioritised and out-of-hours support).

## 9.3 iPXE Support

**iPXE boot is supported on Solarflare XtremeScale™ and X2 series adapters.**

An iPXE boot image can be programmed into the adapter's flash via the Solarflare-supplied sfupdate utility.

iPXE is an alternative open-source network boot firmware providing both PXE support and additional features such as HTTP and iSCSI boot. Solarflare have integrated, maintained and support iPXE drivers in the iPXE open source code base.

Users can use iPXE features not provided within the gPXE based Solarflare Boot ROM agent. However, iPXE is an open source project with its own development and test process not under the direct control of the Solarflare engineering team. Solarflare will monitor the iPXE development mailing lists and participate to ensure the iPXE driver for Solarflare adapters operates correctly.



***NOTE: It is recommended that customers having support questions on the iPXE feature set work directly with the iPXE open source community.***

## 9.4 sfupdate Options for PXE upgrade/downgrade

This section describes sfupdate when used to install/upgrade/downgrade PXE images. Refer to sfupdate in OS specific sections of this document for a complete list of sfupdate options.

Each version of sfupdate contains a firmware image and Solarflare Boot Manager image.

### Current Versions

Run the sfupdate command to identify current image versions:

```
enp5s0f0 - MAC: 00-0F-53-41-C7-00sf
Firmware version: v6.3.0
Controller type: Solarflare SFC9200 family
Controller version: v6.2.5.1000
Boot ROM version: v5.0.3.1002
UEFI ROM version: v2.4.3.1
```

When an iPXE image has been flashed over the Solarflare Boot Manager:

```
enp5s0f0 - MAC: 00-0F-53-41-C7-00
Controller type: Solarflare SFC9200 family
Controller version: v6.2.5.1000
Boot ROM version: iPXE
UEFI ROM version: v2.4.3.1
```

***(NOTE: sfupdate is not able to display version numbers for iPXE images.)***



## sfupdate - Solarflare Boot Manager image

- To install the firmware image and Solarflare, gPXE based, Boot Manager image:  
`# sfupdate [--adapter=] --write [--force] [--backup]`
- To reinstall firmware and Solarflare Boot Manager image from sfupdate:  
`# sfupdate [--adapter=] --write --restore-bootrom`
- To reinstall only a Solarflare Boot Manager image from backup:  
`# sfupdate [--adapter=] --write --img=<image.dat>`

Use the --force option when downgrading. Use the --backup option to create a backup image (.dat) file of the current firmware and Solarflare Boot Manager image.

## sfupdate - iPXE image

- To install the iPXE image, but keep current firmware:  
`# sfupdate [--adapter=] --write [--backup] --ipxe-image=<image.mrom>`
- Use the --backup option to create a backup of the existing firmware and PXE boot ROM image.
- To upgrade firmware and retain the iPXE image:  
`# sfupdate [--adapter=] --write [--force]`

Using the --force option allows firmware to be downgraded but keeps the current iPXE image.



**CAUTION:** sfupdate does not do version checking for iPXE therefore it is possible to downgrade the image without any displayed warning and without using the --force option.

## 9.5 Starting PXE Boot

The Boot Manager can be configured using any of the following methods:

- On server startup, press **Ctrl+B** when prompted during the boot sequence. This starts the Solarflare Boot Manager GUI.
  - Use the supplied `sfboot` command line tool.
- From a Linux environment, you can use the `sfboot` utility. See [Configuring the Boot Manager with sfboot on page 74](#).

`sfboot` is a command line utility program from the Solarflare Linux Utilities RPM package (SF-107601-LS) available from support@solarflare.com.

PXE requires DHCP and TFTP Servers, the configuration of these servers depends on the deployment service used.

### Linux

See [Unattended Installations on page 300](#) for more details of unattended installation on Linux

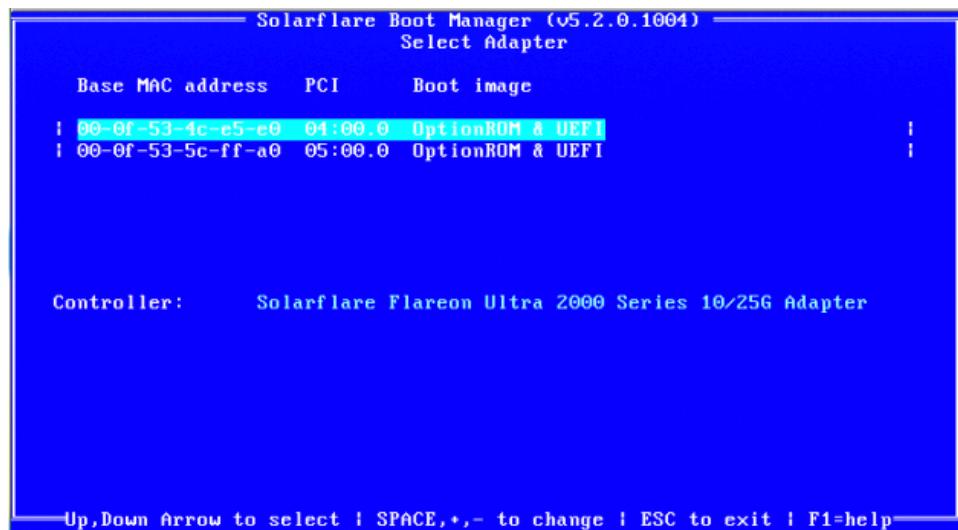
## Configuring the Boot Manager for PXE

This section describes configuring the adapter via the **Ctrl+B** option during server startup.



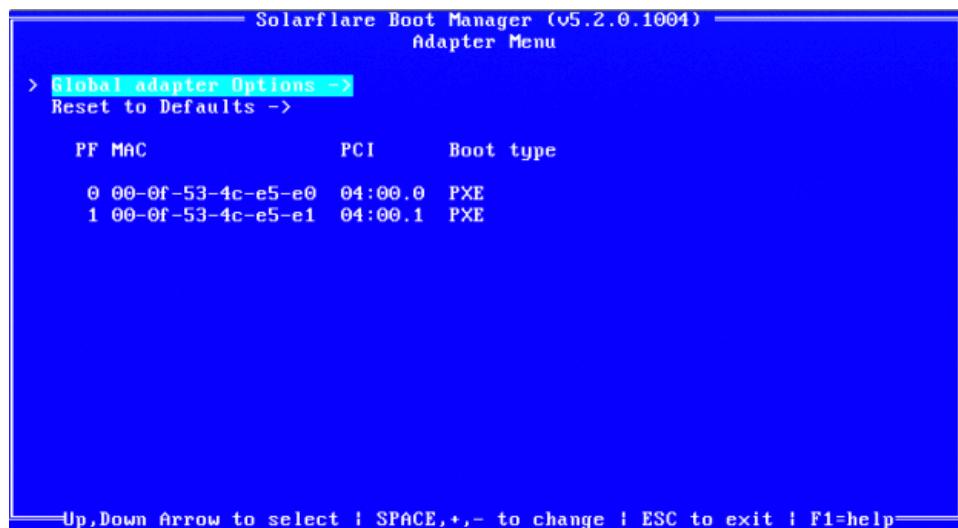
**NOTE:** If the BIOS supports console redirection, and you enable it, then Solarflare recommends that you enable ANSI terminal emulation on both the BIOS and your terminal. Some BIOSs are known to not render the Solarflare Boot Manager properly when using vt100 terminal emulation.

- 1 On starting or re-starting the server, press **Ctrl+B** when prompted. The Solarflare Boot Manager is displayed.

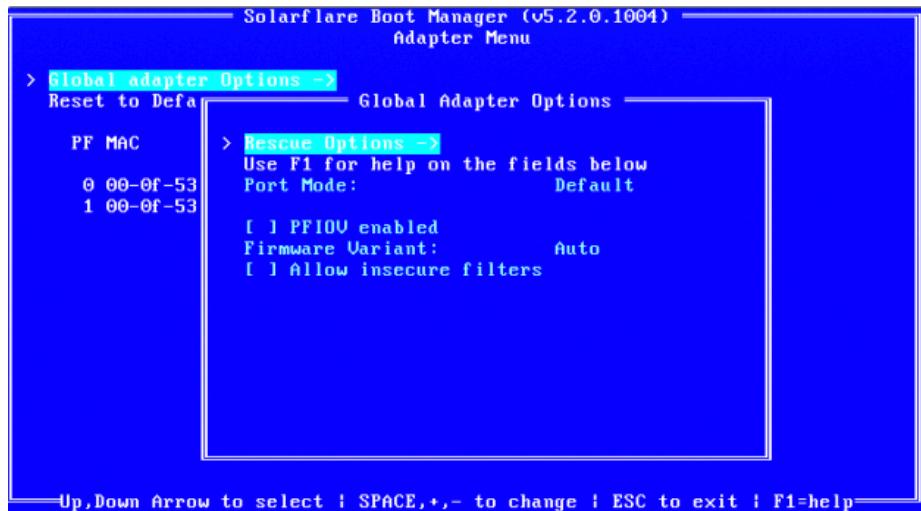


The initial **Select Adapter** page lists the available adapters. In the above example, there are two adapters, on PCI bus 04 and PCI bus 05.

- 2 Use the arrow keys to highlight the adapter you want to boot via PXE and press **Enter**. The **Adapter Menu** is displayed.



- 3 Use the arrow keys to highlight the **Global adapter Options** and press *Enter*. The **Global Adapter Options** menu is displayed.



- 4 The Rescue Options Window

The default setting from the Rescue Menu is OptionROM & UEFI and it should not be necessary to change this.



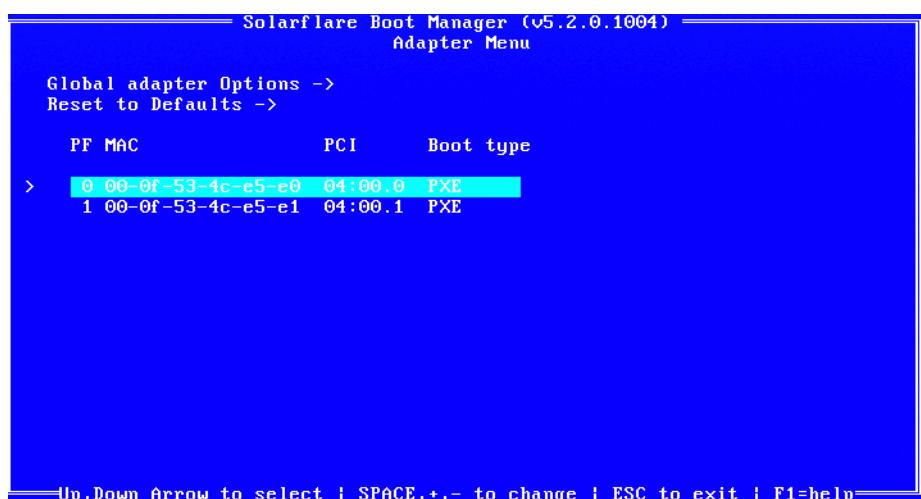
**CAUTION:** This is not a standard PXE procedure. Customers with a PXE boot problem should contact support@solarflare.com

- 5 Select the required boot image:

- a) Use the arrow keys to highlight the **Boot Image**.
- b) Use the space bar to choose the required image.
- c) Press the *Esc* key to exit the **Global Adapter Options**.

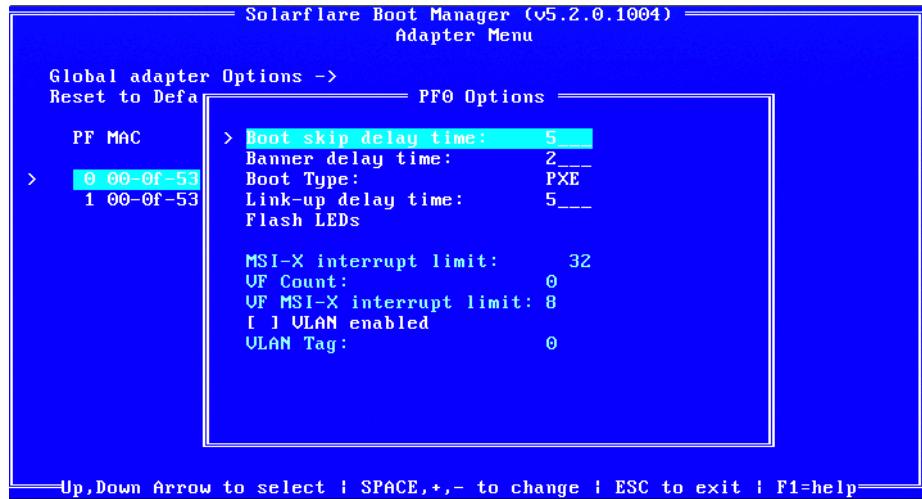
The **Adapter Menu** is again displayed.

- 6 Use the arrow keys to highlight the PF you want to boot via PXE and press *Enter*.



The **PF Options** menu is displayed.

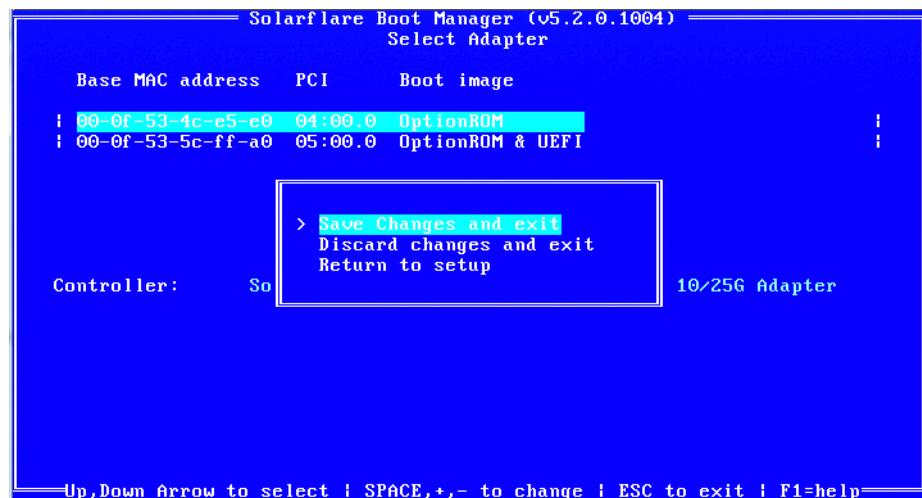
**7 Set the PF to use PXE boot:**



- a) Use the arrow keys to highlight the **Boot Type**.
- b) Use the space bar to select **PXE**.

Solarflare recommend leaving other settings at their default values. For details on the default values for the various adapter settings, see [Table 9.8 on page 298](#).

- 8 Press the *ESC* key repeatedly until the Solarflare Boot Manager exits.**
- 9 Choose **Save Changes and exit**.**



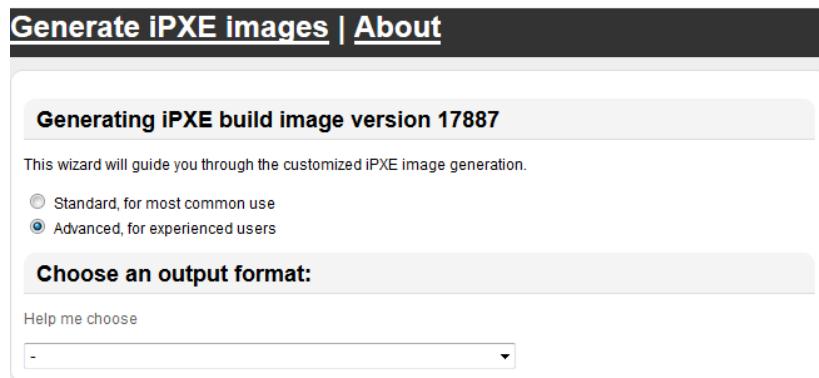
## 9.6 iPXE Image Create

**Solarflare do not provide pre-built iPXE images.**

The Solarflare iPXE boot ROM image can be generated from the **rom-o-matic** iPXE web builder wizard available from:

<https://rom-o-matic.eu/>

- 1 Select the Advanced option:



- 2 Select an Output format:



**3 Enter NIC Details:**

**Generating iPXE build image version 17887**

This wizard will guide you through the customized iPXE image generation.

Standard, for most common use  
 Advanced, for experienced users

**Choose an output format:**

Help me choose  
 ROM binary (flashable) for larger ROM images (.mrom)

**Enter NIC device details:**

You have chosen Binary ROM image as your output format. To match the image to your NIC device, please enter its PCI VENDOR CODE and PCI DEVICE CODE.

Information on how to determine NIC PCI IDs:  
 PCI VENDOR CODE: 1924 PCI DEVICE CODE: 0a03

iPXE does not support all possible PCI IDs for supported NICs.

**Embedded script:**

Read about embedded scripts  
 Paste your script:  
`#!ipxe`

Or import your script:  
 Browse... No file selected.

Or drop your script:  
 Drop your script here

**Which revision?**

Read about GIT revision  
 Default master (recommended) master

**Ready to build?**

**Proceed >>** **Save buildcfg**

- a) Enter the Solarflare PCI vendor identifier: 1924.
- b) Enter the adapter PCI Device Code: [0a03 | 0b03]<sup>1</sup>
- c) Select the GIT version (master is recommended).

**4 Generate the image file.**

Click the Proceed button to start image generation then download the created image file to the target server.

**5 Apply the image to the Solarflare adapter using sfupdate:**

```
sfupdate [--adapter=] --write --ipxe-image=<filename.mrom>
```

<sup>1</sup> 0a03 - SFN8000 series adapter, 0b03 - X2 series adapter

## 9.7 Multiple PF - PXE Boot

Using the sfboot command line utility v4.5.0 (or later version) it is possible to PXE boot when multiple Physical Functions have been enabled. The primary function on each port (PF0/PF1) is a privileged function and can be selected for configuration. Other PFs inherit from their privileged function- so, for example, with two physical ports and 2 PFs per port:

- PF0 and PF2 will have the same boot-type
- PF1 and PF3 will have the same boot-type

Configuration of non-privileged functions is not currently supported.

In the following example 2 PFs (and 2 VFs) are enabled for each physical interface.

```
sfboot
Solarflare boot configuration utility [v4.5.0]

eth2:
 Boot image Option ROM only
 Link speed Negotiated automatically
 Link-up delay time 5 seconds
 Banner delay time 2 seconds
 Boot skip delay time 5 seconds
 Boot type Disabled
 Physical Functions per port 2
 MSI-X interrupt limit 32
 Number of Virtual Functions 2
 VF MSI-X interrupt limit 8
 Firmware variant full feature / virtualization
 Insecure filters Disabled
 MAC spoofing Disabled
 VLAN tags 100,110
 Switch mode Partitioning with SRIOV

eth3:
 Boot image Option ROM only
 Link speed Negotiated automatically
 Link-up delay time 5 seconds
 Banner delay time 2 seconds
 Boot skip delay time 5 seconds
 Boot type Disabled
 Physical Functions per port 2
 MSI-X interrupt limit 32
 Number of Virtual Functions 2
 VF MSI-X interrupt limit 8
 Firmware variant full feature / virtualization
 Insecure filters Disabled
 MAC spoofing Disabled
 VLAN tags 100,110
 Switch mode Partitioning with SRIOV
```

```

eth4:
 Interface-specific boot options are not available. Adapter-wide options
 are available via eth2 (00-0F-53-25-39-90).

```

```

eth5:
 Interface-specific boot options are not available. Adapter-wide options
 are available via eth2 (00-0F-53-25-39-90).

```

## Using the Boot Manager

When multiple Physical Functions have been enabled, the Solarflare Boot Manger GUI utility (CTRL-B) will list them:

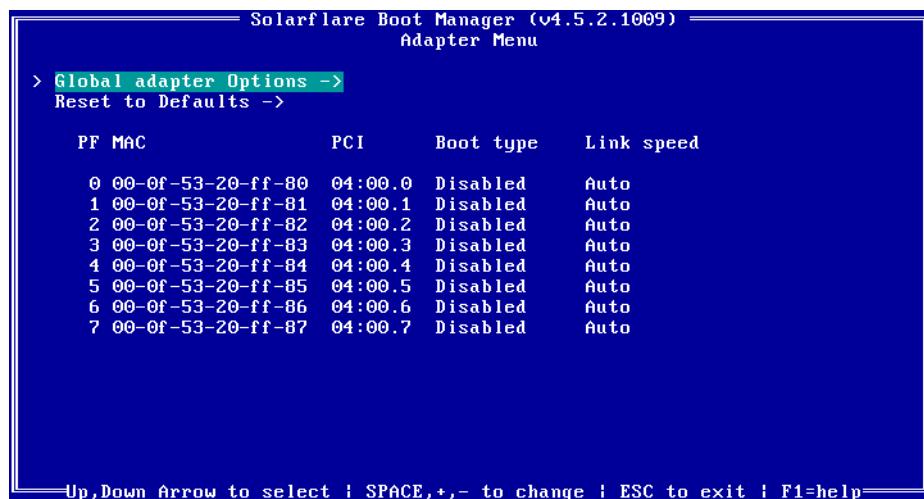


Figure 23: Boot Manager lists multiple PFs

The settings for each PF are read-only, and the only supported action is to **Flash LEDs** on the port being used.

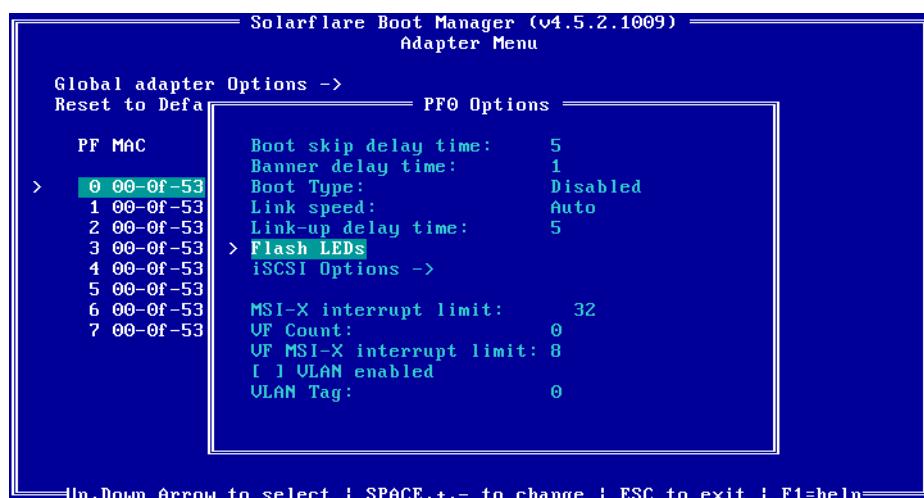


Figure 24: Read-only settings for multiple PFs

## Recovery from incorrect settings

Certain settings must be correct for successful PXE booting, such as:

- port mode
- VLAN tagging.

If these settings become incorrect, for example because a server is moved to a different part of the network. PXE booting will then fail.

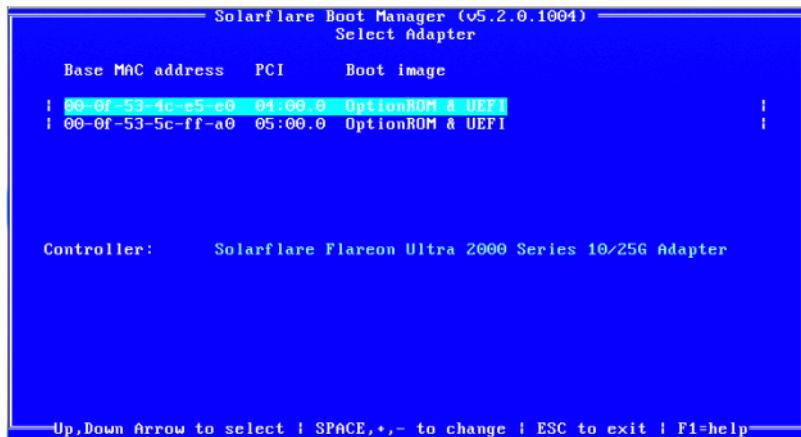
To correct these settings, you must use the Solarflare Boot Manger GUI utility. (You cannot use the sfboot command line utility, because there is no OS to host it.) This is possible only in single Physical Function mode.

If multiple Physical Functions have been enabled, the incorrect settings are read-only. In such cases, you must reset the adapter to its default settings (see [Default Adapter Settings on page 298](#)). This returns the adapter to a single Physical Function mode, and removes all VLAN tags. You can then use the Boot Manger to make the settings that you require.

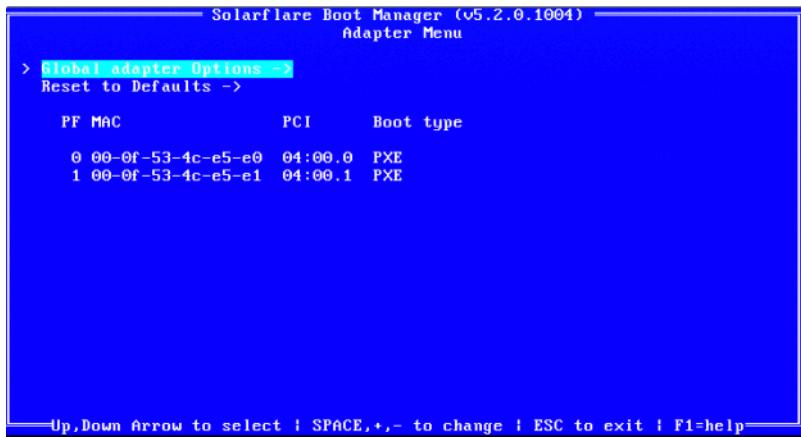
## 9.8 Default Adapter Settings

Resetting an adapter does not change the boot ROM image. To reset an adapter to its default settings:

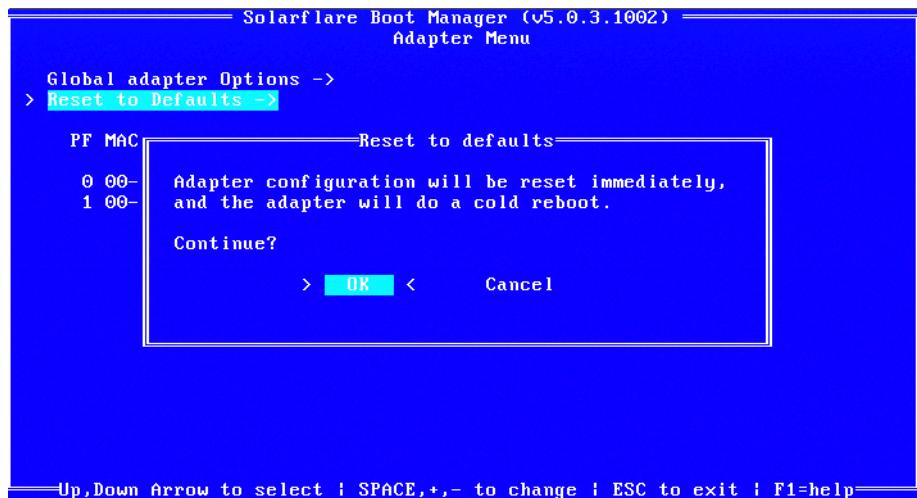
- 1 Start or re-start the server and when prompted, press **Ctrl+B**. The Solarflare Boot Manager is displayed.



- 2 Use the arrow keys to highlight the adapter you want to restore and press *Enter*. The **Adapter Menu** is displayed.



- 3 Use the arrow keys to highlight **Reset to Defaults** and press *Enter*. The **Reset to Defaults** confirmation is displayed.



- 4 Use the arrow keys to highlight **OK** and press *Enter*. The settings are reset to the defaults.

**Table 9.8** lists the various adapter settings and their default values.

**Table 56: Default Adapter Settings**

Setting	Default Value
Boot Image	OptionROM & UEFI
Link up delay	5 seconds
Banner delay	2 seconds
Boot skip delay	5 seconds
Boot Type	PXE
MPIO attempts	
MSIX Limit	32

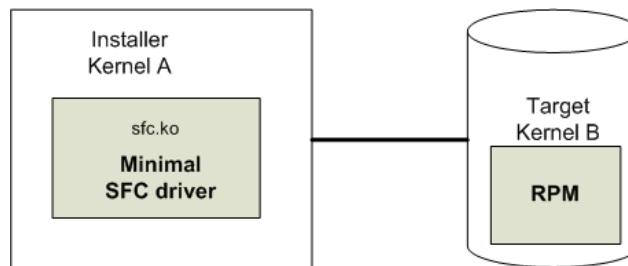
# 10

# Unattended Installations

## Building Drivers and RPMs for Unattended Installation

Linux unattended installation requires building two drivers:

- A minimal installation Solarflare driver that only provides networking support. This driver is used for network access during the installation process.
- An RPM that includes full driver support. This RPM is used to install drivers in the resultant Linux installation.



**Figure 25: Unattended Installation RPM**

[Figure 25](#) shows how the unattended installation process works.

- 1 Build a minimal Solarflare driver needed for use in the installation kernel (Kernel A in the diagram above). This is achieved by defining “sfc\_minimal” to rpmbuild. This macro disables hardware monitoring, MTD support (used for access to the adapters flash), I2C and debugfs. This results in a driver with no dependencies on other modules and allows networking support from the driver during installation.

```

as normal user
$ mkdir -p /tmp/rpm/BUILD
$ rpm -i sfc-<ver>-1.src.rpm
$ rpmbuild -bc -D 'sfc_minimal=1' -D 'kernel=<installer kernel>' \
/tmp/rpm/SPECS/sfc.spec

```

- 2 The Solarflare minimal driver sfc.ko can be found in /tmp/rpm/BUILD/sfc-<ver>/linux\_net/sfc.ko. Integrate this minimal driver into your installer kernel, either by creating a driver disk incorporating this minimal driver or by integrating this minimal driver into initrd.
- 3 Build a full binary RPM for your Target kernel and integrate this RPM into your Target (Kernel B).

## Driver Disks for Unattended Installations

**Table 57** below identifies the various stages of an unattended installation process:

**Table 57: Installation Stages**

In Control	Stages of Boot	Setup needed
BIOS	PXE code on the adapter runs.	Adapter must be in PXE boot mode. See <a href="#">Solarflare Boot Manager on page 285</a> .
SF Boot ROM (PXE)	DHCP request from PXE (SF Boot ROM).	DHCP server filename and next-server options.
SF Boot ROM (PXE)	TFTP request for filename to next-server, e.g. pxelinux.0	TFTP server.
pxelinux	TFTP retrieval of pxelinux configuration.	pxelinux configuration on TFTP server.
pxelinux	TFTP menu retrieval of Linux kernel image initrd.	pxelinux configuration Kernel, kernel command, initrd
Linux kernel/installer	Installer retrieves kickstart configuration, e.g. via HTTP.	Kickstart/AutoYaST configuration.
Target Linux kernel	kernel reconfigures network adapters.	DHCP server.

### 10.1 Unattended Installation - Red Hat Enterprise Linux

Documentation for preparing for a Red Hat Enterprise Linux network installation can be found at:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Installation\\_Guide/](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/)

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Installation\\_Guide/](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/)

The prerequisites for a Network Kickstart installation are:

- Red Hat Enterprise Linux installation media.
- A Web server and/or FTP Server for delivery of the RPMs that are to be installed.
- A DHCP server for IP address assignments and to launch PXE Boot.

- A TFTP server for download of PXE Boot components to the machines being kickstarted.
- The BIOS on the computers to be Kickstarted must be configured to allow a network boot.
- A Boot CD-ROM or flash memory that contains the kickstart file or a network location where the kickstart file can be accessed.
- A Solarflare driver disk.

Unattended Red Hat Enterprise Linux installations are configured with Kickstart. The documentation for Kickstart can be found at:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Installation\\_Guide/ch-kickstart2.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-kickstart2.html)

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Installation\\_Guide/chap-kickstart-installations.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/chap-kickstart-installations.html)

To install Red Hat Enterprise you need the following:

- 1 A modified `initrd.img` file with amended `modules.alias` and `modules.dep` which incorporates the Solarflare minimal driver for the installation kernel.

Find current aliases with the `modinfo` command:

```
modinfo sfc | grep alias
```

Then add the aliases found to the `modules.alias` file:

```
pci:v00001924d00001A03sv*sd*bc*sc*i*
pci:v00001924d00000A03sv*sd*bc*sc*i*
pci:v00001924d00001923sv*sd*bc*sc*i*
pci:v00001924d00000923sv*sd*bc*sc*i*
pci:v00001924d00001903sv*sd*bc*sc*i*
pci:v00001924d00000903sv*sd*bc*sc*i*
pci:v00001924d00000813sv*sd*bc*sc*i*
pci:v00001924d00000803sv*sd*bc*sc*i*
```

- 2 Identify the driver dependencies using the `modinfo` command:

```
modinfo ./sfc.ko | grep depends
depends: i2c-core,mii,hwmon,hwmon-vid,i2c-algo-bit mtdcore mtdpart
```

All modules listed as depends must be present in the initrd file image. In addition the user should be aware of further dependencies which can be resolved by adding the following lines to the `modules.dep` file:

```
sfc: i2c-core mii hwmon hwmon-vid i2c-algo-bit mtdcore mtdpart1
i2c-algo-bit: i2c-core
mtdpart: mtdcore
```

- 3 A configured kickstart file with the Solarflare Driver RPM manually added to the `%Post` section. For example:

```
%post
```

```
/bin/mount -o ro <IP Address of Installation server>:/<path to
```

---

1. For Red Hat Enterprise Linux from version 5.5 add `mdio` to this line.

```
location directory containing Solarflare RPM> /mnt
/bin/rpm -Uvh /mnt/<filename of Solarflare RPM>
/bin/umount /mnt
```

## 10.2 Unattended Installation - SUSE Linux Enterprise Server

Unattended SUSE Linux Enterprise Server installations are configured with AutoYaST. The documentation for AutoYaST can be found at:

[https://www.suse.com/documentation/sles11/book\\_autoplayast/data/book\\_autoplayast.html](https://www.suse.com/documentation/sles11/book_autoplayast/data/book_autoplayast.html)

[https://www.suse.com/documentation/sles-12/book\\_autoplayast/data/book\\_autoplayast.html](https://www.suse.com/documentation/sles-12/book_autoplayast/data/book_autoplayast.html)

The prerequisites for a Network AutoYaST installation are:

- SUSE Linux Enterprise installation media.
- A DHCP server for IP address assignments and to launch PXE Boot.
- A NFS or FTP server to provide the installation source.
- A TFTP server for the download of the kernel boot images needed to PXE Boot.
- A boot server on the same Ethernet segment.
- An install server with the SUSE Linux Enterprise Server OS.
- An AutoYaST configuration server that defines rules and profiles.
- A configured AutoYaST Profile (control file).

### Further Reading

- SUSE Linux Enterprise Server remote installation:  
[https://www.suse.com/documentation/sles11/book\\_sle\\_deployment/data/cha\\_deployment\\_remoteinst.html](https://www.suse.com/documentation/sles11/book_sle_deployment/data/cha_deployment_remoteinst.html)  
[https://www.suse.com/documentation/sles-12/book\\_sle\\_deployment/data/cha\\_deployment\\_remoteinst.html](https://www.suse.com/documentation/sles-12/book_sle_deployment/data/cha_deployment_remoteinst.html)
- SUSE install with PXE Boot:  
[https://www.suse.com/documentation/sles11/book\\_sle\\_deployment/data/sec\\_deployment\\_remoteinst\\_boot.html#sec\\_deployment\\_remoteinst\\_boot\\_pxe](https://www.suse.com/documentation/sles11/book_sle_deployment/data/sec_deployment_remoteinst_boot.html#sec_deployment_remoteinst_boot_pxe)  
[https://www.suse.com/documentation/sles-12/book\\_sle\\_deployment/data/sec\\_deployment\\_remoteinst\\_boot.html#sec\\_deployment\\_remoteinst\\_boot\\_pxe](https://www.suse.com/documentation/sles-12/book_sle_deployment/data/sec_deployment_remoteinst_boot.html#sec_deployment_remoteinst_boot_pxe)  
[http://www.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5.4/html/Deployment\\_Guide/s2-modules-bonding.html](http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Deployment_Guide/s2-modules-bonding.html)
- RHEL6:



[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/  
Deployment\\_Guide/s2-networkscripts-interfaces-chan.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-networkscripts-interfaces-chan.html)

- SLES:

[http://www.novell.com/documentation/sles11/book\\_sle\\_admin/data/  
sec\\_basicnet\\_yast.html#sec\\_basicnet\\_yast\\_netcard\\_man](http://www.novell.com/documentation/sles11/book_sle_admin/data/sec_basicnet_yast.html#sec_basicnet_yast_netcard_man)

# Index

## Numerics

25G Link Speed 35

## A

Adapter Configuration 130

Advanced Features and Benefits 2

Auto-negotiation/Link Training 35

## B

Boot ROM Exposed 285

## C

Configure MTU

FreeBSD 217

Linux 91

CPU Speed Service

Tuning on Linux 102

## D

Default 298

DirectPath I/O 281

## E

ESXCLI 179

Ethtool

Configure Interrupt moderation on  
Linux 91

Configure segmentation offload 93

Running adapter diagnostics on  
Linux 120

ethtool statistics 111

## F

Fiber Optic Cable

Attaching 21

Forward Error Correction 35

## I

Identify NetQueue Interrupts 161

Inserting the adapter 19

Install Solarflare SfTools 129

Installing Solarflare Driver Package 127

Intel QuickData

On Linux 104

Interrupt Affinity 99

Interrupt and Irqbalance

Tuning on Linux 99

Interrupt Moderation

Configure on FreeBSD 218

iPXE Image Create 293

iPXE Support 287

## J

Jumbo Frames

Configuring on Linux 57

## K

Kernel Module Packages (KMP) 50

KVM Direct Bridged 248

KVM libvirt Bridged 244

KVM Libvirt Direct Passthrough 251

KVM Network Architectures 244

## L

Large Receive Offload (LRO)

Configure on FreeBSD 220

Configure on Linux 93

Legacy Driver 124

Linux

Configure MTU 91

## M

Memory bandwidth  
Tuning on Linux 103

## N

NIC Partitioning 59

## P

PCI Express Lane Configuration  
On FreeBSD 222  
On Linux 102

PF-IOV 257

Port Modes 38

Powerd Service  
Tuning on FreeBSD 223

PXE

Configure with the Boot ROM  
agent 290

PXE Enabled 285

## R

Receive Flow Steering (RFS) 68

Receive Side Scaling (RSS) 66

RJ-45 cable

Attaching 20  
Specifications 21

## S

Segmentation offload

Configure on FreeBSD 219  
Configure on Linux 93

SFP 10GBASE-T Transceivers 30

SFP+ to SFP+ Breakout Cables 33

sfupdate

On FreeBSD 214

sfupdate Options for PXE upgrade/  
downgrade 287

Single Optical Fiber - RX Configuration

44

Solarflare Accelerated RFS (SARFS) 70

Solarflare AppFlex™ Technology  
Licensing 15

Solarflare Boot Manager 286

Solarflare Boot ROM Agent 285

SR-IOV 242

SR-IOV Virtualization Using ESXi 262

SUSE

Installing on 52

System Requirements  
FreeBSD 200  
Linux 46

## T

TCP Protocol Tuning  
On Linux 95

Teaming

Setting up on Linux 58

Transmit Packet Steering (XPS) 71

Tuning Recommendations  
On FreeBSD 223  
On Linux 104

## U

Unattended Installation  
Driver disks 301  
SUSE 303

## V

VF Passthrough 275

Virtual Machine 269

vSwitch and Port Group Configuration  
271

## W

Windows 2016 Driver 124

Windows Feature Set 126

