OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

INTEL® OMNI-PATH FABRIC VIRTUAL NETWORK INTERFACE CONTROLLER (OPA VNIC)

Niranjana Vishwanathapura, Software Development Engineer

Intel Corporation

[March 29, 2017]

intel®

LEGAL DISCLAIMERS

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The cost reduction scenarios described in this document are intended to enable you to get a better understanding of how the purchase of a given Intel product, combined with a number of situation-specific variables, might affect your future cost and savings. Circumstances will vary and there may be unaccounted-for costs related to the use and deployment of a given product. Nothing in this document should be interpreted as either a promise of or contract for a given level of costs.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: Learn About Intel® Processor Numbers

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

The High-Performance Linpack (HPL) benchmark is used in the Intel® FastFabrics toolset included in the Intel® Fabric Suite. The HPL product includes software developed at the University of Tennessee, Knoxville, Innovative Computing Libraries.

Intel, Intel Xeon, Intel Xeon Phi™ are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries.

INTRODUCTION

Supporting Ethernet over Omni-Path fabric allows us to make full use of standard Ethernet support provided by the Operating System (including VLAN etc.) over the fabric without having verbs layering in the stack.

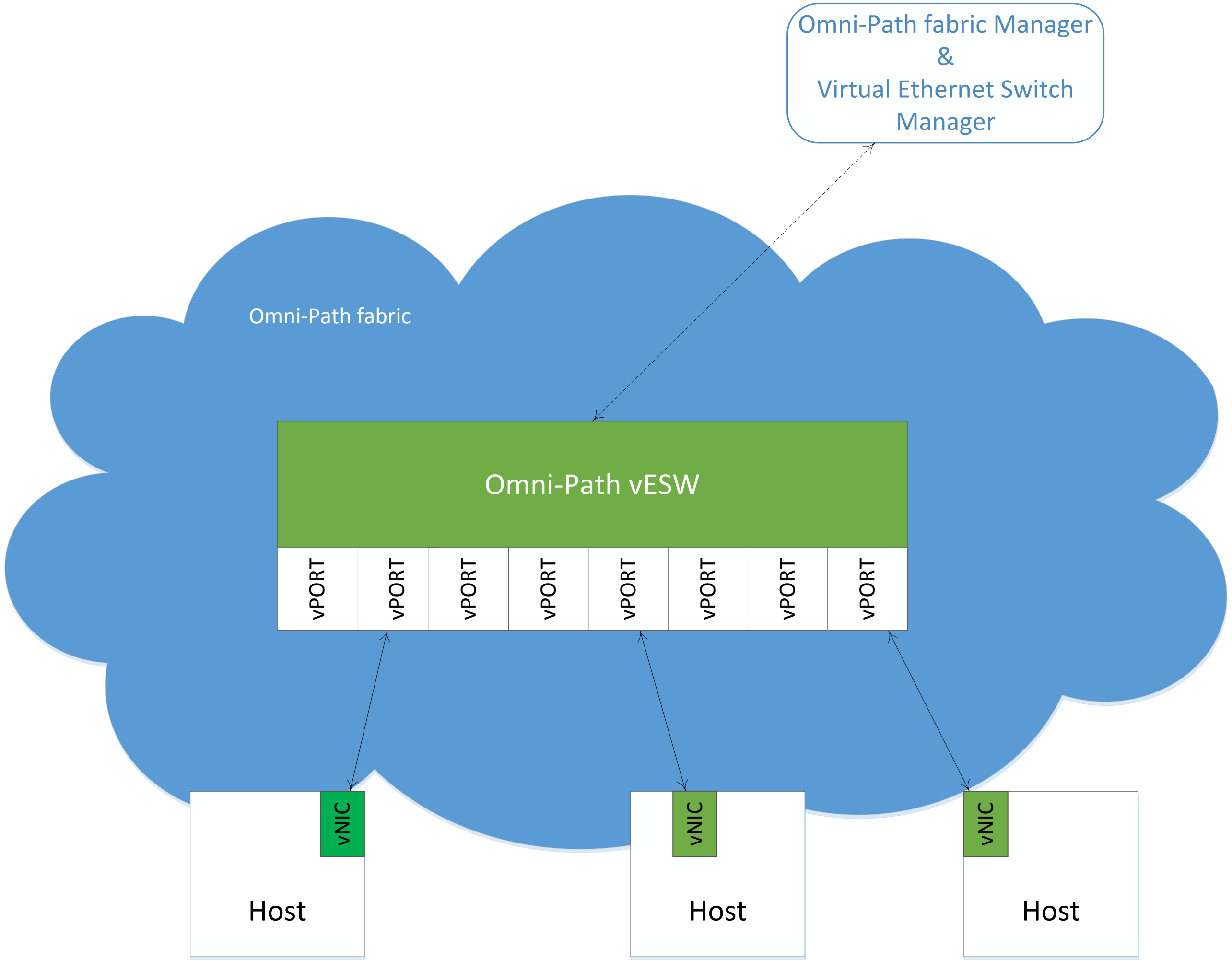
Intel Omni-Path (OPA) Virtual Network Interface Controller (VNIC) feature supports Ethernet functionality over Omni-Path fabric by encapsulating an Ethernet packet within an Omni-Path packet.

Agenda:

- OPA VNIC Architecture
- OPA VNIC Driver Design

OMNI-PATH VNIC ARCHITECTURE

ARCHITECTURE

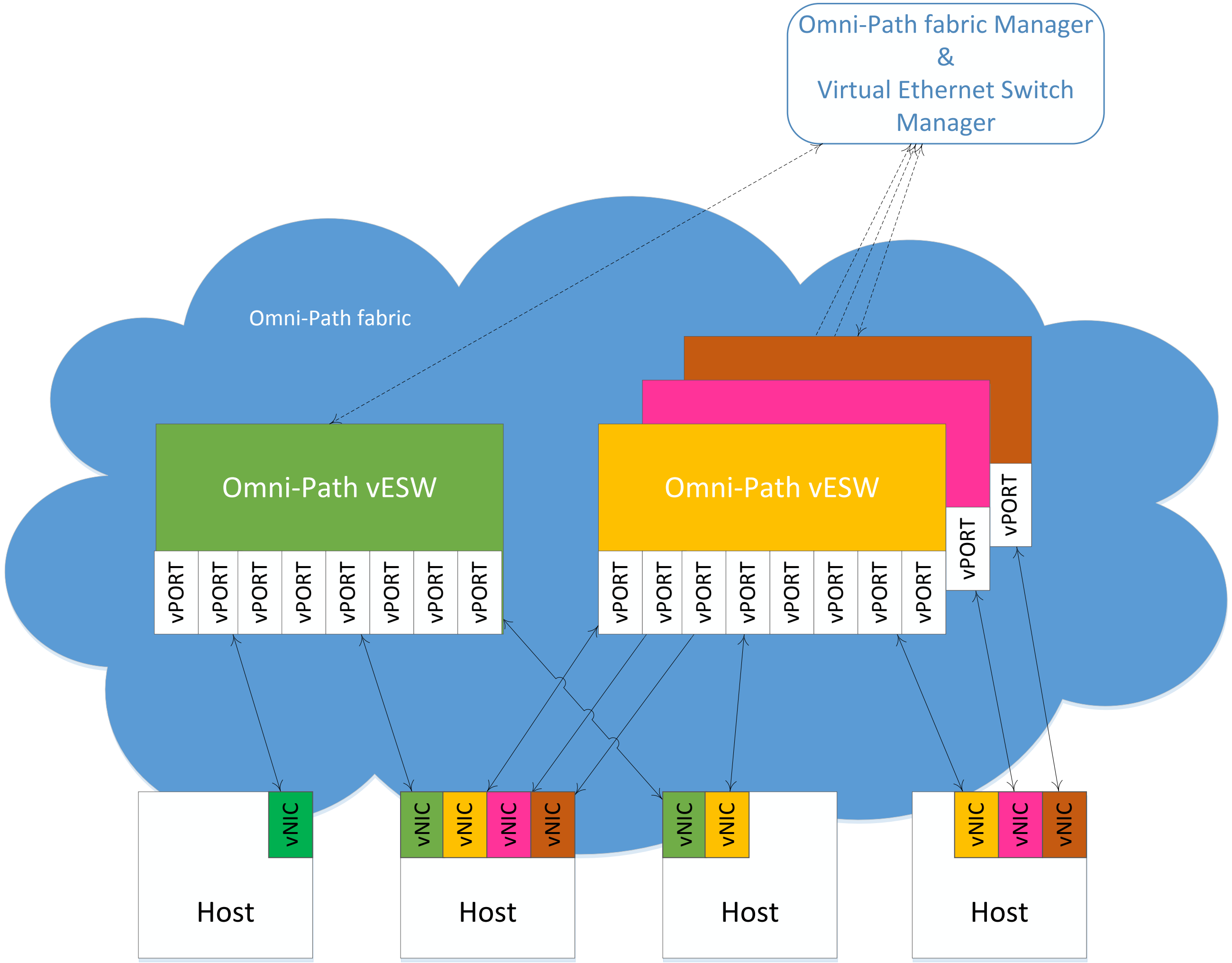


- An Omni-Path virtual Ethernet switch (vESW) is a **logical abstraction** achieved by configuring the hosts on the fabric for header generation and processing

- The configuration is performed by an Ethernet Manager (EM) which is part of the trusted Fabric Manager (FM) application

ARCHITECTURE

Multiple Omni-Path vESW example

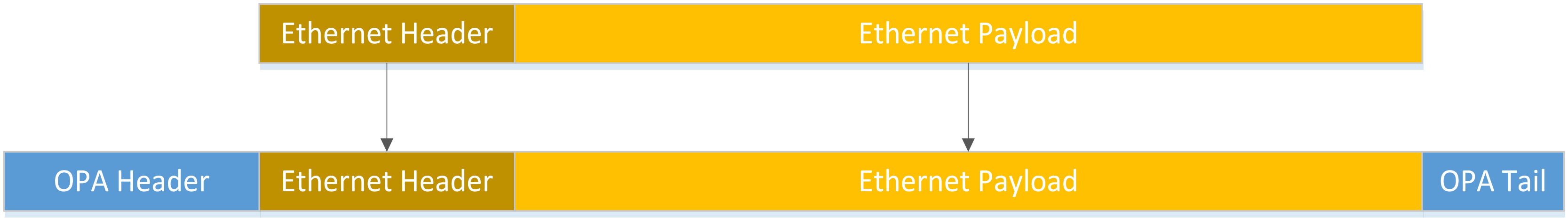


- There can be multiple Omni-Path vESWs in the fabric

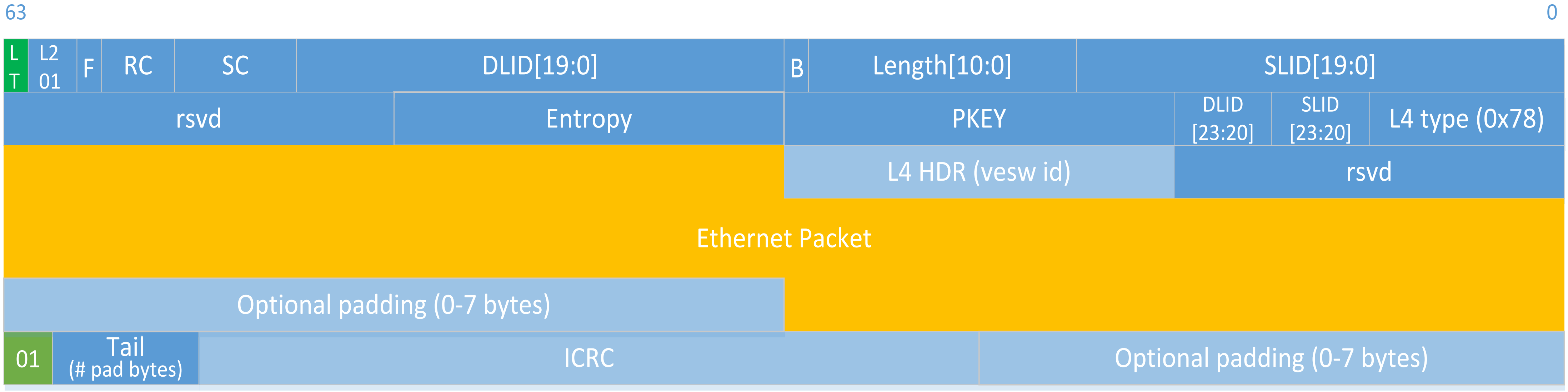
- Hosts can have multiple vNICs each connected to a different Omni-Path vESW

ARCHITECTURE

Packet format



Omni-Path encapsulation of Ethernet Packet

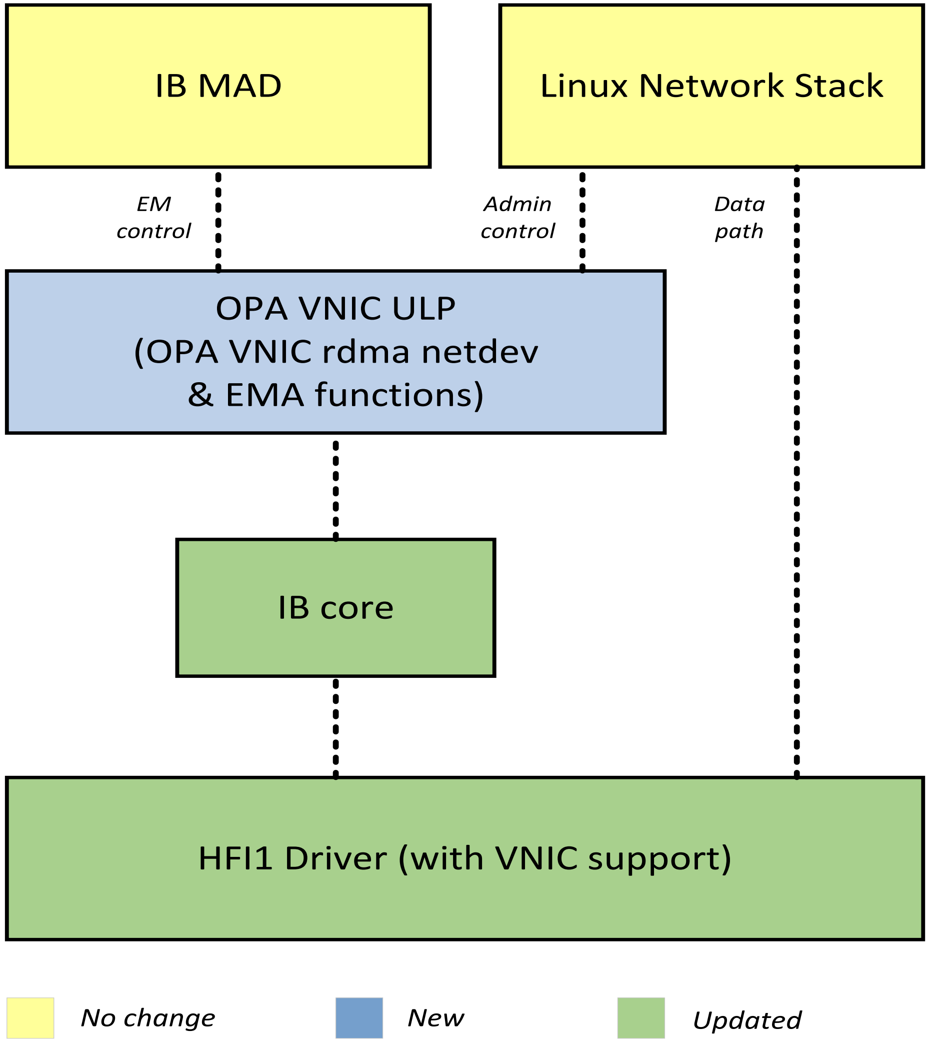


Omni-Path encapsulated Ethernet Packet format

OMNI-PATH VNIC DRIVER DESIGN

DRIVER DESIGN

Omni-Path VNIC SW stack



IB MAD

Linux Network Stack

EM control

Admin control

Data path

OPA VNIC ULP
(OPA VNIC rdma netdev
& EMA functions)

IB core

HFI1 Driver (with VNIC support)

🟨 *No change* 🟦 *New* 🟩 *Updated*

- OPA VNIC ULP is an '*ib_client*'
- EMA is an '*ib_mad_agent*'
- Linux network stack's SKB interface is used and no translation to verbs API required
- HW Driver (HFI1) defines '*net_device_ops*' and can interact directly with the network stack on data path
- OPA VNIC module can override the '*net_device_ops*' defined by HW driver to implement control plane operations and encapsulation

DRIVER DESIGN

rdma netdev

- Requirements
  - Allow OFA device drivers to interface directly with Linux network stack
  - No translation to Verbs Interface required thus providing optimization
- ‘**rdma netdev**’ - A generic netdev interface to OFA device drivers where Linux network stack interfacing is required
- Ability to support different kind of rdma netdev devices
- Address OPA_VNIC and IPoIB use case requirements
- Not adding any overhead on the data path

DRIVER DESIGN

rdma netdev (Omni-Path VNIC example)

```
/**
 * struct rdma_netdev - rdma netdev
 * For cases where netstack interfacing is required.
 */
struct rdma_netdev {
        void                    *clnt_priv;
        struct ib_device  *ibdev;
        u8                       port_num;

        /* control functions */
        void (*set_id)(struct net_device *netdev, int id);
};

/* rdma netdev type - specifies protocol type */
enum rdma_netdev_t {
        RDMA_NETDEV_OPA_VNIC
};

struct ib_device {
        …
        /* rdma netdev operations */
        struct net_device *(*alloc_rdma_netdev)(struct ib_device *device, u8 port_num,
                                               enum rdma_netdev_t type, const char *name,
                                               unsigned char name_assign_type, void (*setup)(struct net_device *));
        void (*free_rdma_netdev)(struct net_device *netdev);
        …
}
```

```
/* opa vnic rdma netdev's private data structure */
struct opa_vnic_rdma_netdev {
        struct rdma_netdev rn;  /* keep this first */

        /* followed by device private data */
        char *dev_priv[0];
};

/* Get ULP's (OPA_VNIC) private data */
static inline void *opa_vnic_priv(const struct net_device *dev)
{
        struct rdma_netdev *rn = netdev_priv(dev);

        return rn->clnt_priv;
}

/* Get driver's (HFI1's VNIC) private data */
static inline void *opa_vnic_dev_priv(const struct net_device *dev)
{
        struct opa_vnic_rdma_netdev *opa_rn = netdev_priv(dev);

        return opa_rn->dev_priv;
}
```

DRIVER DESIGN

Omni-Path VNIC ULP

- Implements required netdev control operations. Allocates rdma netdev and registers netdev with network stack.
- Does OPA encapsulation of Ethernet packets
- Implements EMA IB MAD agent to interact with EM
- Implements Ethtool interface

- **EM Interface**
  - Attributes:
    - CLASS_PORT_INFO
    - VESWPORT_INFO
    - VESWPORT_MAC_ENTRIES
    - IFACE_UCAST_MACS
    - IFACE_MCAST_MACS
    - DELETE_VESW
    - VESWPORT_SUMMARY_COUNTERS
    - VESWPORT_ERROR_COUNTERS
  - Traps:
    - IFACE_UCAST_MAC_CHANGE
    - IFACE_MCAST_MAC_CHANGE
    - ETH_LINK_STATUS_CHANGE

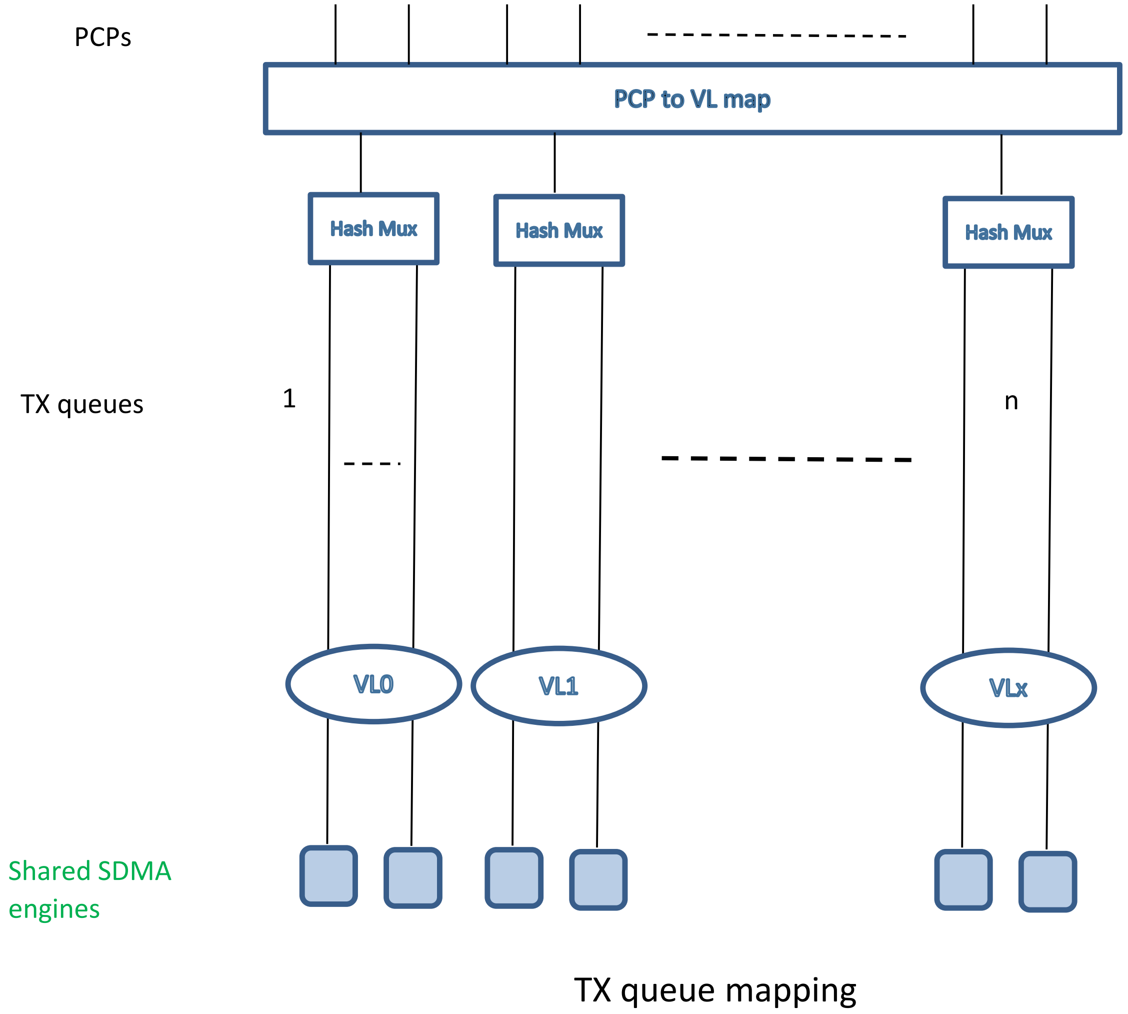
DRIVER DESIGN

HFI1 VNIC support

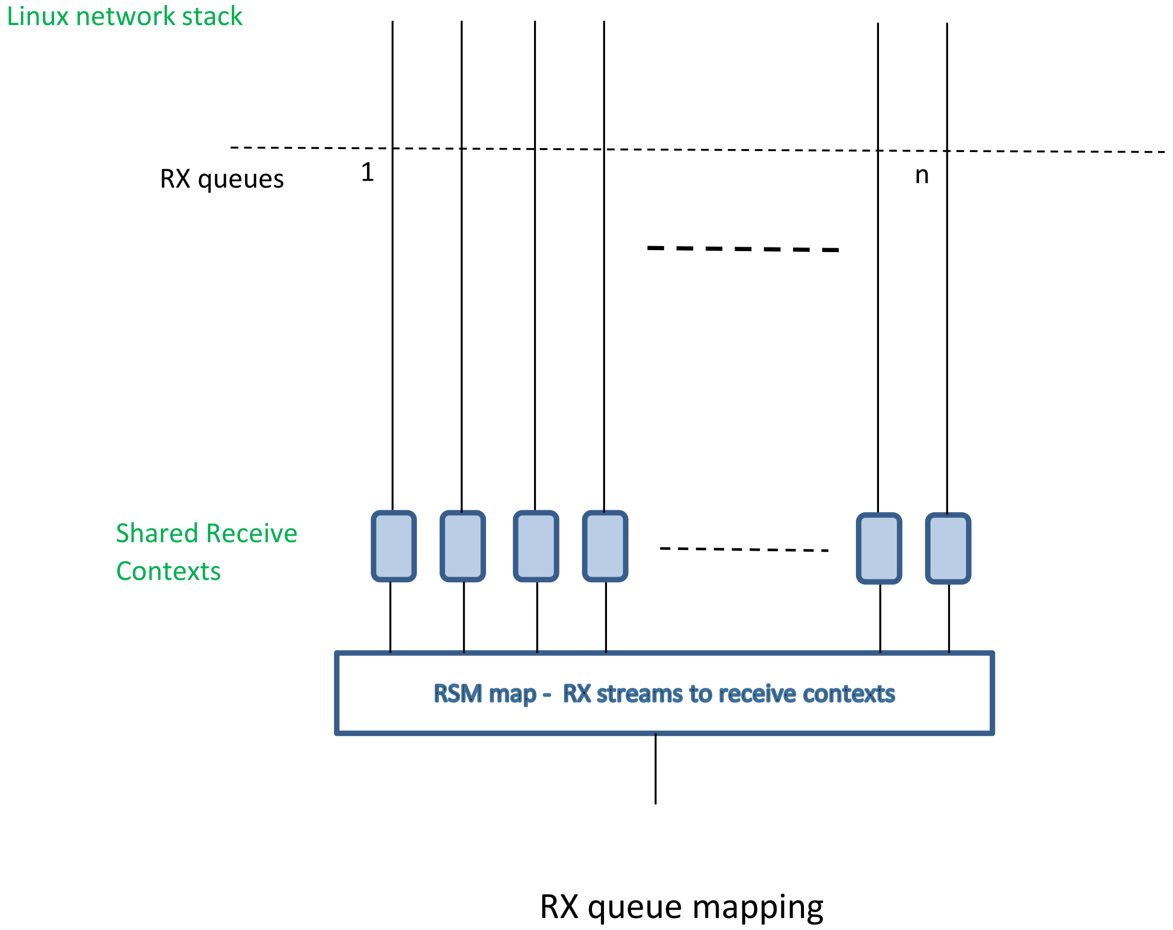
- HW resource management for VNIC traffic
  - Allocates and frees receive contexts
  - Implements RSS using HFI1 RSM engine
- Implements TX path
  - uses hfi1 SDMA engines
  - supports multiple TX queues (VL based)
  - supports TX queue halt and wakeup
- Implements the Rx path
  - Implements multiple Rx queues (RSM)
  - Implements NAPI interface
- Implements VNIC statistics support
  - Supports standard netdev and rmon counters
  - Supports EM defined counters

DRIVER DESIGN

Queue Mapping



TX queue mapping



RX queue mapping

STATUS & NEXT STEPS

Status:

- **Currently the OPA_VNIC patch series is posted on LKML**
  - https://www.spinics.net/lists/linux-rdma/msg46604.html

Next Steps:

- **RDMACM address resolution using VNIC interface**
  - Currently exploring options for RDMACM to use VNIC interface (instead of IPoIB) to translate destination node's IP address to LID

OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

THANK YOU

Niranjana Vishwanathapura, Software Development Engineer

Intel Corporation