INTEL® 25GBE ADVANCED FEATURES FOR NFV

HELIN ZHANG, INTEL®
JINGJING WU, INTEL®
Agenda

• Key Hardware Features
• Dynamic Device Personalization (DDP)
• Generic Flow API
• Virtual Function Daemon (VFD)
• Good Performance
• Adaptive Virtual Function (AVF)
Key Hardware Features

• PCIe v3.0, x8
• XXV710, 25GbE Link Speed
  • New addition to Intel® Ethernet 700 Series (10/25/40GbE)
• Network Virtualization offloads
  • VXLAN, NVGRE, GENEVE, VXLAN-GPE with NSH, MPLS, and more
• Input Set for RSS and Flow Director (FD)
  • Up to 24 of 56 words can be selected
• 3 HASH Algorithms
  • Toeplitz, Simple XOR, Symmetric Simple XOR
### Key Hardware Features for Virtualization

<table>
<thead>
<tr>
<th>Feature</th>
<th>XXV710</th>
<th>82599EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR-IOV support</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VF to PF mailbox</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Max Number of Virtual functions</td>
<td>128 per device (globally)</td>
<td>64 per port (single queue)</td>
</tr>
<tr>
<td>Max number of Queues</td>
<td>1536</td>
<td>128</td>
</tr>
<tr>
<td>Max number of queues per VF</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Max number of queues per VMDq2 VSI</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Max Number of VMDq2 ports</td>
<td>256 per device (globally)</td>
<td>64 per port (single queue)</td>
</tr>
<tr>
<td>MAC addresses</td>
<td>1024 per device (globally)</td>
<td>128 per port</td>
</tr>
<tr>
<td>VLAN tags</td>
<td>512 per device (global)</td>
<td>64 per port</td>
</tr>
<tr>
<td>Queuing to Pool/VSI method</td>
<td>SA, VLAN pairs or SA or VLAN</td>
<td>SA or VLAN or (SA and VLAN)</td>
</tr>
<tr>
<td>Cloud filter in Switch</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RSS per VF</td>
<td>Yes</td>
<td>No (Single RSS used for all VFs).</td>
</tr>
<tr>
<td>Switching modes</td>
<td>VEB, VEPA</td>
<td>VEB*</td>
</tr>
<tr>
<td>Promiscuous modes per VM</td>
<td>VLAN, Multicast, Unicast</td>
<td>Multicast</td>
</tr>
</tbody>
</table>
Internal Packet Processing

82599EB
• Fixed packets Parse graphic.
• Input set of filtering/steering is fixed.

XXV710
• Configurable input set for RSS and FD.
• DDP to support more protocol steering.
Dynamical Device Personalization (DDP)

- By default, it supported limited protocols, due to hardware resources
  - e.g. VXLAN
- Loadable profiles for packet classification for extra protocols
  - e.g. MPLSoGRE
- Configurable tunnel filters for traffic steering
  - Steering packets to VM queues on QinQ/tunnel ID
Generic Flow API Support

• A generic way to configure the hardware
  • Don’t need to know the HW specific filters

• Flow rule
  • Attributes
  • Matching pattern
  • Actions

• Rule management
  • rte_flow_validate()
  • rte_flow_create()
  • rte_flow_destroy()
  • rte_flow_flush()
Example

• Direct the VXLAN packet with specific inner MAC and VNI to queue #2.

```
struct rte_eth_tunnel_filter_conf tunnel_filter_conf = {
  .outer_mac = {0x11, 0x22, 0x33, 0x44, 0x55, 0x66);
  .inner_mac = {0x00, 0x11, 0x22, 0x33, 0x44, 0x55};
  .inner_vlan = 0;
  .ip_type = RTE_TUNNEL_IPTYPE_IPV4;
  .ip_addr.ipv4_addr = 1;
  .filter_type = RTE_TUNNEL_FILTER_IMAC_TENID;
  .tunnel_type = RTE_TUNNEL_TYPE_VXLAN;
  .tenant_id = 1;
  .queue_id = 2;
};

int ret;
ret = rte_eth_dev_filter_ctrl(port_id, RTE_ETH_FILTER_TUNNEL,
  RTE_ETH_FILTER_ADD, &tunnel_filter_conf);
```

```
const struct rte_flow_item pattern[] = {
  { RTE_FLOW_ITEM_TYPE_ETH, NULL, NULL, NULL},
  { RTE_FLOW_ITEM_TYPE_IPV4, NULL, NULL, NULL},
  { RTE_FLOW_ITEM_TYPE_UDP, NULL, NULL, NULL},
  { RTE_FLOW_ITEM_TYPE_VXLAN, {.vni = 1}, NULL, {.vni = "\xff\xff\xff"}},
  { RTE_FLOW_ITEM_TYPE_ETH,
    {.dst = {0x00, 0x11, 0x22, 0x33, 0x44, 0x55}}, NULL,
    {.dst = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}}},
  { RTE_FLOW_ITEM_TYPE_END, NULL, NULL, NULL},
};

const struct rte_flow_action actions[] = {
  { RTE_FLOW_ACTION_TYPE_PF, NULL},
  { RTE_FLOW_ACTION_TYPE_QUEUE, { .index = 2}},
  { RTE_FLOW_ACTION_TYPE_END, NULL},
};

struct rte_flow_error flow_err;
flow_err = rte_flow_create(port_id, NULL, pattern, actions, &flow_err);
```
Virtual Function Daemon (VFD) Support

- DPDK as the host driver to support both DPDK and kernel VF
- Lots of VF management features are added
- Mailbox messages management are added
  - VF requests can be accepted/rejected by VFD
- Kernel driver does not support those features
- Only Intel® Ethernet 500 (ixgbe) and 700 (i40e) series are enabled
- Refer to https://github.com/att/vfd
<table>
<thead>
<tr>
<th>Packet Size (Bytes)</th>
<th>Wire Speed (Mpps)</th>
<th>Packet Rate (Mpps)</th>
<th>%Wire Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>37.2</td>
<td>35.63</td>
<td>95.78%</td>
</tr>
<tr>
<td>128</td>
<td>21.1</td>
<td>21.1</td>
<td>100%</td>
</tr>
<tr>
<td>256</td>
<td>11.3</td>
<td>11.3</td>
<td>100%</td>
</tr>
<tr>
<td>512</td>
<td>5.87</td>
<td>5.71</td>
<td>97.27%</td>
</tr>
<tr>
<td>1024</td>
<td>3.03</td>
<td>2.91</td>
<td>97.32%</td>
</tr>
</tbody>
</table>

*Packet sizes and wire speeds are measured in bytes and Mbps respectively.*

**2 Cards**

**1 Cards**

---

Adaptive Virtual Function (AVF)
AVF -- Adaptive Virtual Function

• **Needs:**
  • A single VF driver for all generations of Devices.

• **Solution:**
  • Adaptive Virtual Function
    • Base features
    • Negotiated Advanced Features

• **Benefits:**
  • Existing VM Images will run on the new hardware with **no change**.

• **From:**
  • Intel® 700 series Ethernet Controller
AVF – HW upgrade

Hypervisor

VM

PF driver

PF
VF

Gen 1

PF
VF

Gen 2

PF
VF

Gen N

Hardware upgrade
AVF -- Adaptive Virtual Function

• **Base mode supported**
  • Single device ID
  • Support for single level checksum and TSO offload
  • Multi-queue support
  • RSS

• **Advanced features**
  • Advanced feature introduced by new generation HW.
  • Negotiate with PF driver to expose.
Preserving in Hardware and Software

- **Preserving Base mode**
  - Fixed Minimum Register definition
  - A fixed Meta data format for DMA
  - A Hardware generic mailbox to talk to the PF
  - A Software defined Virtual channel layered on top of Hardware mailbox for expansion

- **Room for expansion**
  - Uncompromising on the base functionality.
  - A large range for hot path registers (Queue and Interrupt)
  - Expandable Virtual channel capability negotiation over the agreed upon communication channel between PF and VF.
  - More advanced features would be added with new drops of AVF driver if the underlying HW device supports.
  - Intel is working on the AVF specification.
Key Takeaways

• 25GbE speed, and better hardware capability
• Generic, flexible and configurable flow classification
• NFV enabled with VFD
• Good performance
• Adaptive VF driver for all Intel® NICs from 700 series
End

• Helin Zhang, helin.zhang@intel.com
• Jingjing Wu, jingjing.wu@intel.com