DPDK Virtualization Status & Preview

Zhihong Wang <zhihong.wang@intel.com>
DPDK Summit Userspace - Dublin- 2017
Agenda

- Deployment
  - Live migration between different backends
  - VM memory hot plug
  - Container friendly

- Performance
  - Memory copy in the broad sense

- Virtio 1.1 packed ring layout
Live Migration Between Different Backends

**Motivation**
- Migrate from OVS to OVS-DPDK
- Migrate between SW and accelerators

**Gaps**
- Default negotiated features
- Orchestration commands

**<TODO>**
- Align feature bits with vhost-net

Identified feature gap:
- VIRTIO_NET_F_GSO: Device handles packets with any GSO type
- VIRTIO_NET_F_GUEST_ECN: Driver can receive TSO with ECN
- VIRTIO_NET_F_GUEST_UFO: Driver can receive UFO
- VIRTIO_NET_F_HOST_ECN: Device can receive TSO with ECN
- VIRTIO_NET_F_HOST_UFO: Device can receive UFO
- VIRTIO_F_ANY_LAYOUT: Device accepts arbitrary descriptor layouts
- VIRTIO_F_RING_EVENT_IDX: Interrupt & notification suppression
- VIRTIO_NET_F_GUEST_ANNOUNCE: Driver can send gratuitous packets
Memory Hot Plug

- **Motivation**
  - Elasticity: Memory provisioning and de-provisioning

- **Gaps**
  - Virtio-balloon doesn’t work for hugepages
  - DPDK vhost-user handles memory region update inappropriately

- **<TODO>**
  - Fix memory region update in DPDK vhost-user

---

```plaintext
Two monitor commands are used to hotplug memory:

- `object_add`: creates a memory backend object
- `device_add`: creates a front-end pc-dimm device and inserts it into the first empty slot

Two monitor commands are used to hot unplug memory:

- `device_del`: deletes a front-end pc-dimm device
- `object_del`: deletes a memory backend object
```

Container Friendly

- Lightweight memory model
  - Motivation
    - Increase deployment density
  - <TODO>
    - Address too-many-files limitation in virtio-user
    - 4KB page support with VFIO
- Fast boot
  - Motivation
    - Service fast boot, hot upgrade
  - <TODO>
    - Call for proposal!

<table>
<thead>
<tr>
<th>DPDK app init phase</th>
<th>Time cost (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hugepage init (per 1 GB)</td>
<td>320</td>
</tr>
<tr>
<td>Mbuf init (per 1,000,000)</td>
<td>320 ~ 520</td>
</tr>
<tr>
<td>Bus probe (per device)</td>
<td>240 (uio) ~ 400 (vfio)</td>
</tr>
<tr>
<td>Device start (per device)</td>
<td>440</td>
</tr>
</tbody>
</table>
Key to virtio performance

Optimize for core-to-core
  ▶ Copy virtio header and data **CONSECUTIVELY** (Instruction level)
  ▶ Batch copy small packets

The next breakthrough
  ▶ Performance feature tradeoff
  ▶ Ring layout evolution

Patches:
http://dpdk.org/ml/archives/dev/2016-October/048906.html
http://dpdk.org/ml/archives/dev/2016-December/051658.html
http://dpdk.org/ml/archives/dev/2017-September/074898.html

Article:
The new ring layout: 3 rings -> 1 desc ring

What it enables
- Simplified ring operation
- Sequential desc access
- More hardware friendly

DPDK’s effort
- Great incubator for new technologies
- PMD development and optimization

<TODO>
- Call for review!

DPDK patches:
http://dpdk.org/ml/archives/dev/2017-June/068315.html
http://dpdk.org/ml/archives/dev/2017-July/071562.html


23% gain
Questions?

Zhihong Wang
zhihong.wang@intel.com