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

```
34
35  Two monitor commands are used to hotplug memory:
36
37   - "object_add": creates a memory backend object
38   - "device_add": creates a front-end pc-dimm device and inserts it
39                   into the first empty slot


84  Two monitor commands are used to hot unplug memory:
85
86   - "device_del": deletes a front-end pc-dimm device
87   - "object_del": deletes a memory backend object
```

Reference: *https://github.com/qemu/qemu/blob/master/docs/memory-hotplug.txt*

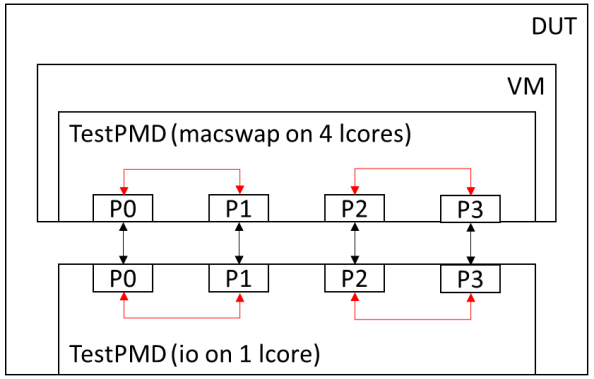
Container Friendly

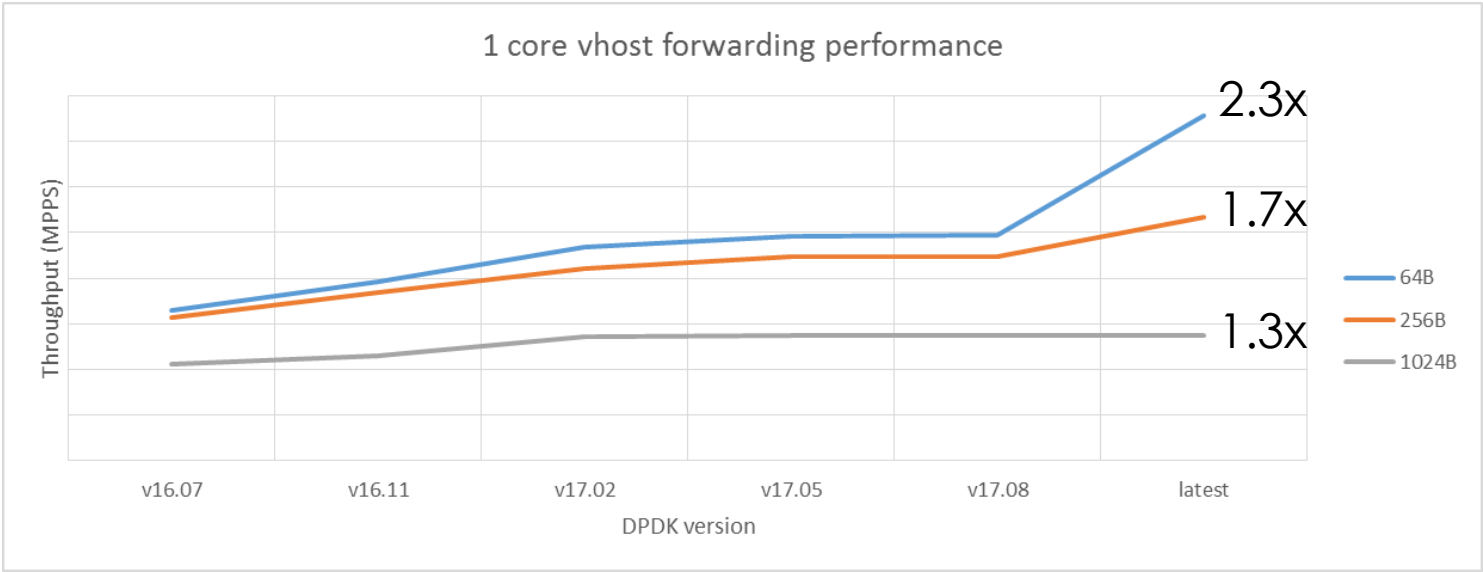
- Lightweight memory model
  - Motivation
    - Increase deployment density
  - <TODO>
    - Address too-many-files limitation in virtio-user
    - 4KB page support with VFIO
- Fast boot
  - Motivation
    - Service fast boot, hot upgrade
  - <TODO>
    - Call for proposal!

DPDK app init phase	Time cost (ms)
Hugepage init (per 1 GB)	320
Mbuf init (per 1,000,000)	320 ~ 520
Bus probe (per device)	240 (uio) ~ 400 (vfio)
Device start (per device)	440

Memory Copy In The Broad Sense

- 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

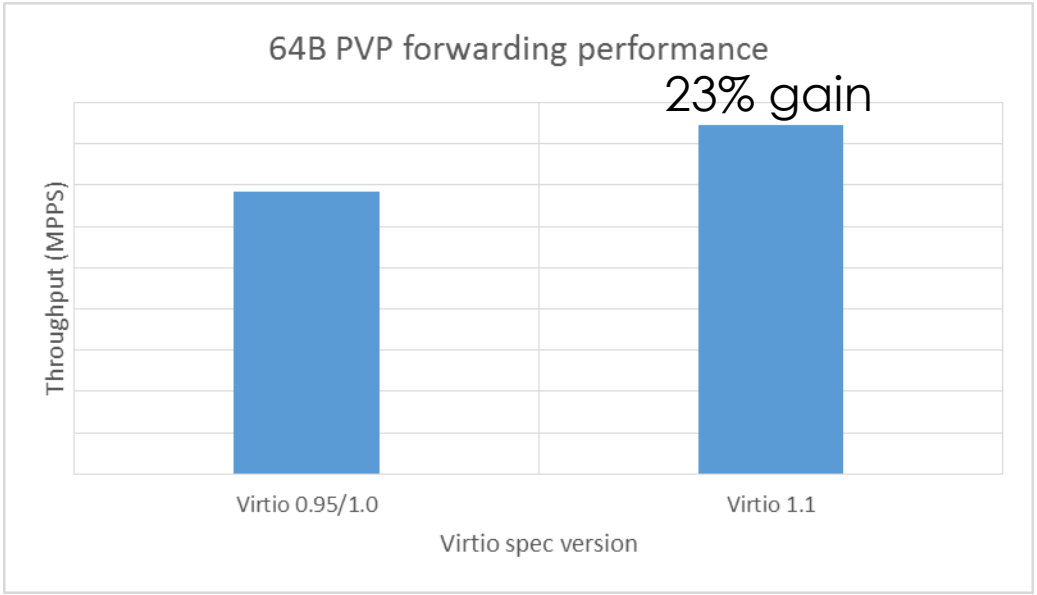




Patches:
http://dpdk.org/ml/archives/dev/2016-October/048906.html
http://dpdk.org/ml/archives/dev/2016-December/051658.html
http://dpdk.org/ml/archives/dev/2017-September/074898.html
Article:
https://software.intel.com/en-us/articles/performance-optimization-of-memcpy-in-dpdk

Virtio 1.1 Packed Ring Layout

- 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!



Test setup: http://dpdk.org/doc/guides/howto/pvp_reference_benchmark.html

DPDK patches:
http://dpdk.org/ml/archives/dev/2017-June/068315.html
http://dpdk.org/ml/archives/dev/2017-July/071562.html
Intel's proposal: https://lists.oasis-open.org/archives/virtio-comment/201708/msg00000.html
The latest v3 proposal: https://lists.oasis-open.org/archives/virtio-dev/201709/msg00013.html

Questions?

Zhihong Wang

zhihong.wang@intel.com