December 5 - 6, 2018 | San Jose, CA

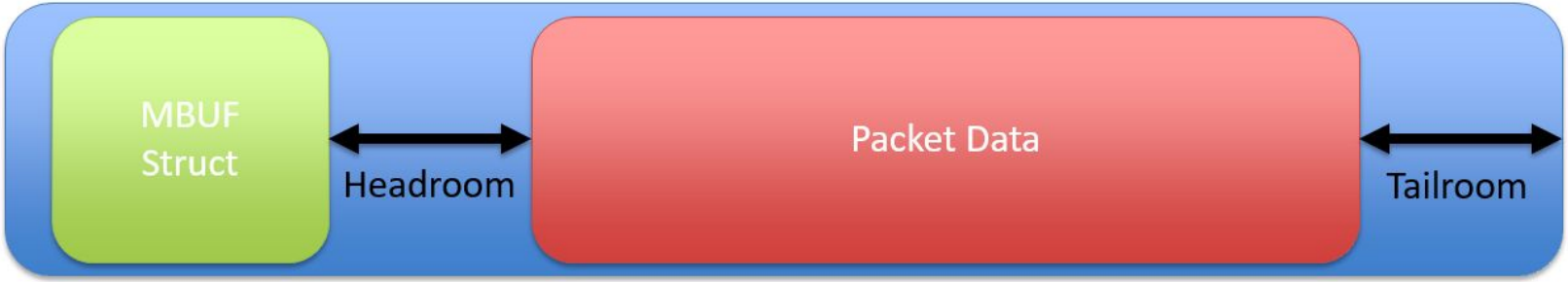
OVS-DPDK: Memory management and debugging

Ian Stokes & Kevin Traynor

Content

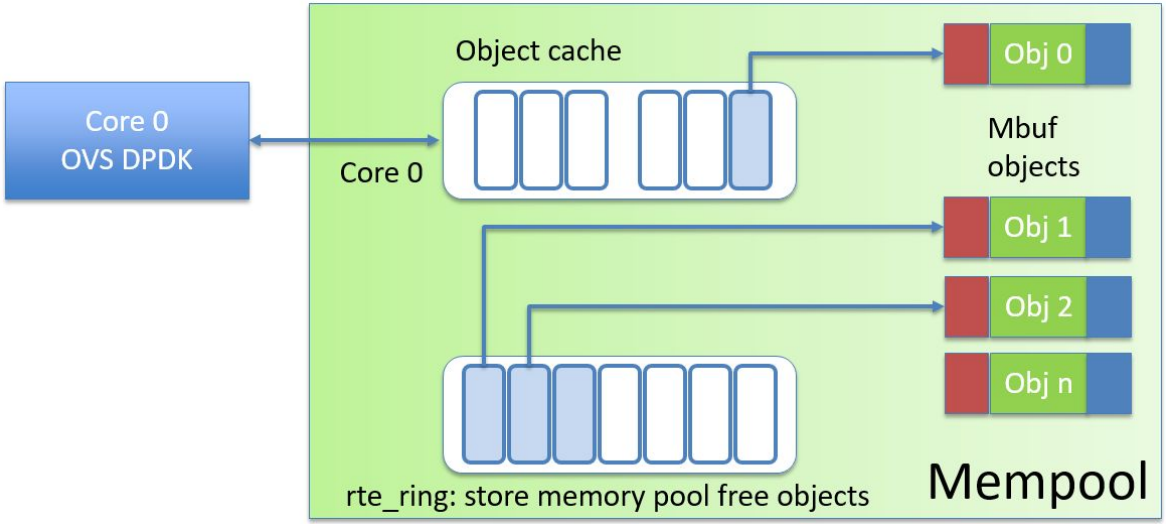
- Mbufs and Mempool
- Shared Memory Overview
- Per Port Memory Overview
- Memory Model Support To Date
- Future Memory Models

MBUF and Mempools



- An rte_mbuf struct
  - Contains metadata control information
  - Packet data i.e. payload
  - Cache aligned
- Can handle single and multiple segments
- Mbufs stored in a mempool

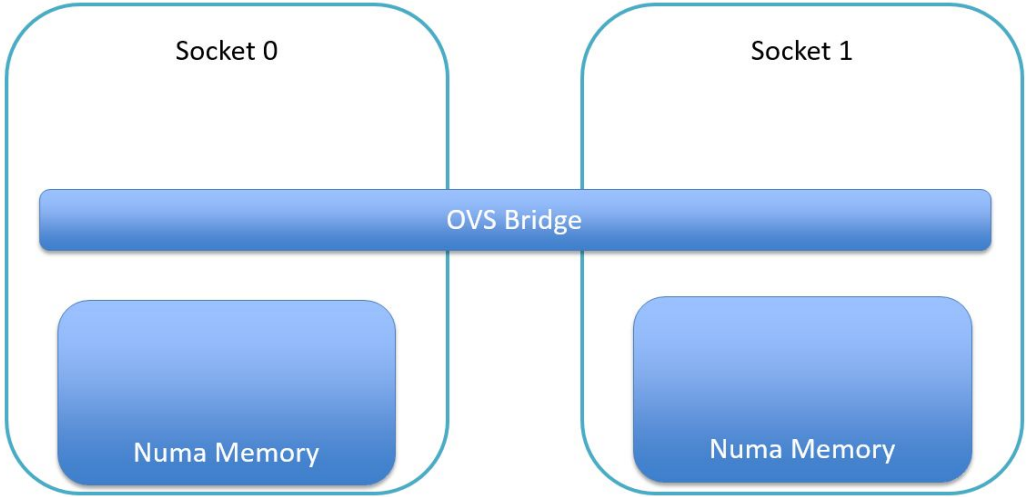
Mempool



- An allocator of a fixed-sized objects i.e. mbuf
- Uses a mempool handler to store free objects
- Maintains a per-core object cache

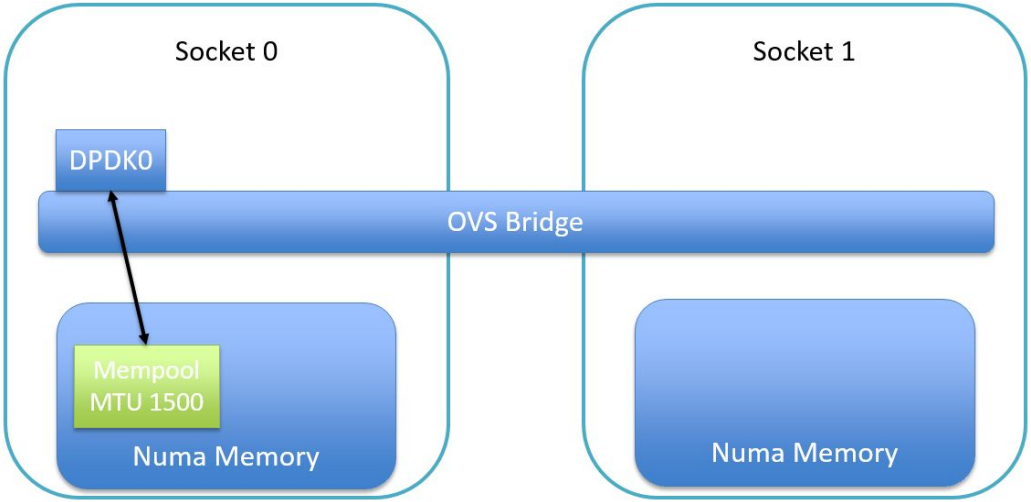
Shared Memory Model Overview

- Mempools shared between interfaces based on:
  - Socket ID
  - MTU Size
- Examples



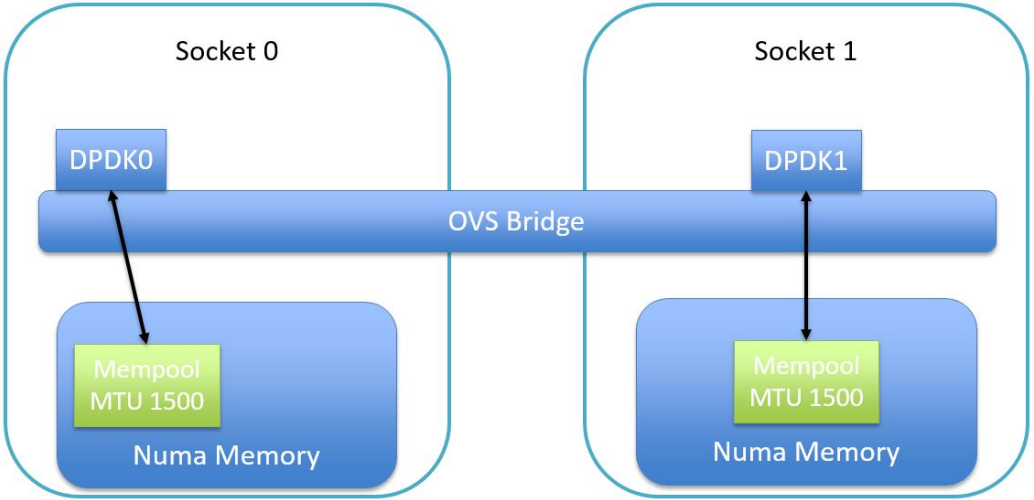
Shared Memory Model Overview

- Mempools shared between interfaces based on:
  - Socket ID
  - MTU Size
- Examples
  - Socket 0 MTU 1500



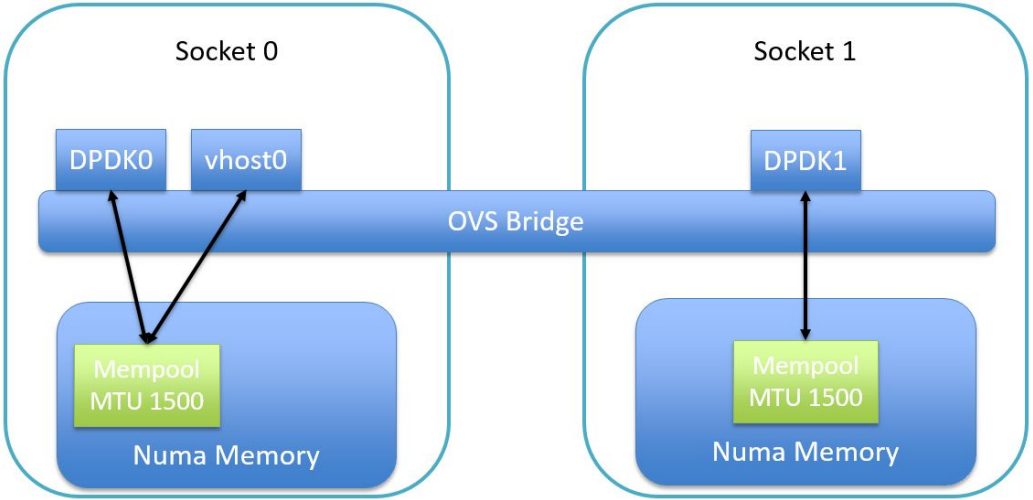
Shared Memory Model Overview

- Mempools shared between interfaces based on:
  - Socket ID
  - MTU Size
- Examples
  - Socket 0 MTU 1500
  - Socket 1 MTU 1500



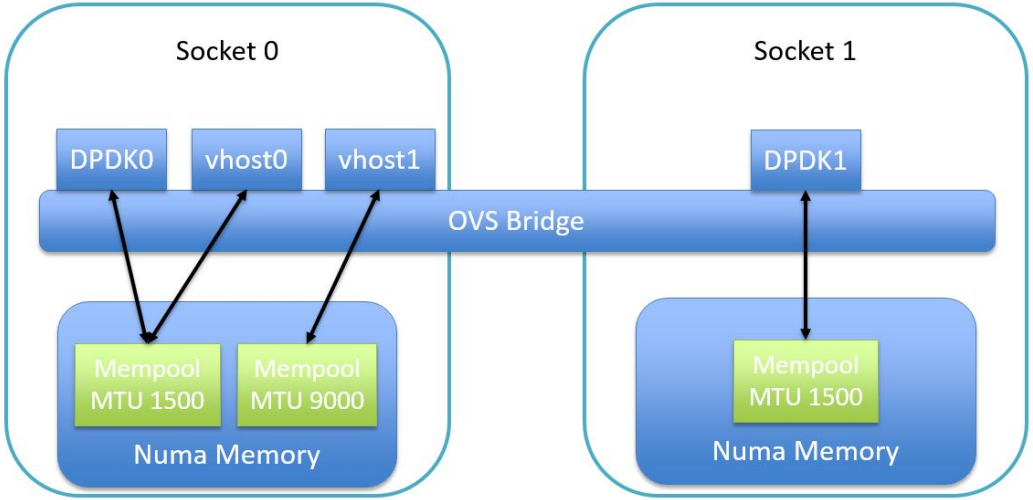
Shared Memory Model Overview

- Mempools shared between interfaces based on:
  - Socket ID
  - MTU Size
- Examples
  - Socket 0 MTU 1500
  - Socket 1 MTU 1500
  - Socket 0 MTU 1500



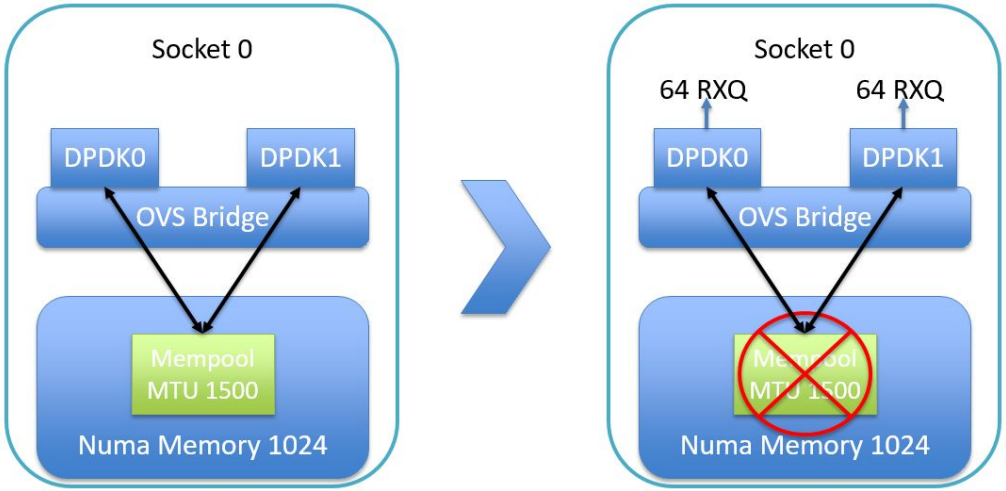
Shared Memory Model Overview

- Mempools shared between interfaces based on:
  - Socket ID
  - MTU Size
- Examples
  - Socket 0 MTU 1500
  - Socket 1 MTU 1500
  - Socket 0 MTU 1500
  - Socket 0 MTU 9000



Shared Memory Model Benefits vs Drawbacks

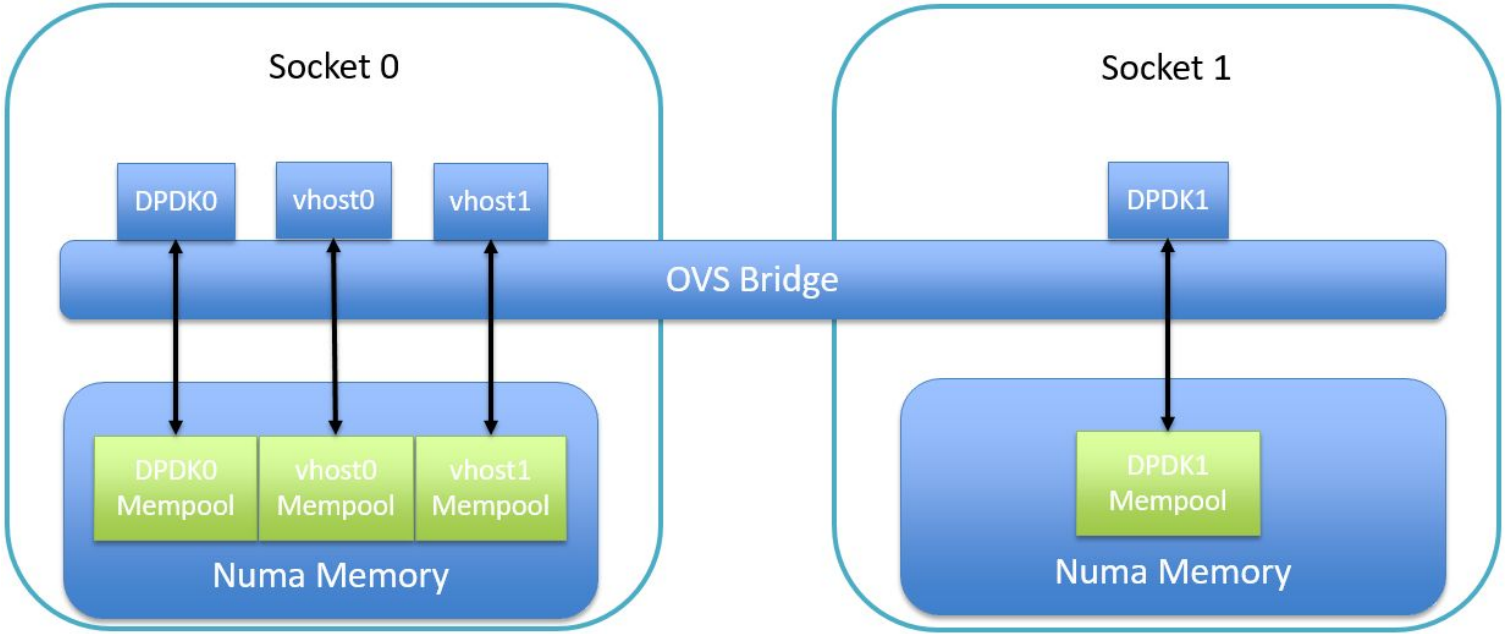
- Benefits
  - Mature solution.
  - Small memory footprint for same socket and MTU config
  - Buffer provisioning accounts for in-flight worst case
- Drawback
  - Configuration of a device could exhaust memory for other devices



https://mail.openvswitch.org/pipermail/ovs-discuss/2016-September/042560.html

Per Port Memory Model Explained

- Mempool now allocated per interface basis, never shared.



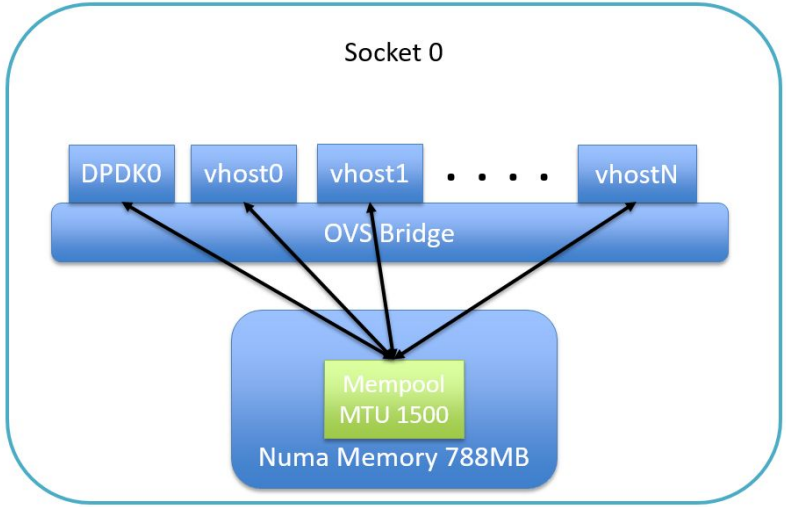
Per Port Memory Model Benefits vs Drawbacks

- Benefits
  - Provides a more transparent memory usage model.
  - Avoids pool exhaustion due to competing memory requirements for interfaces.

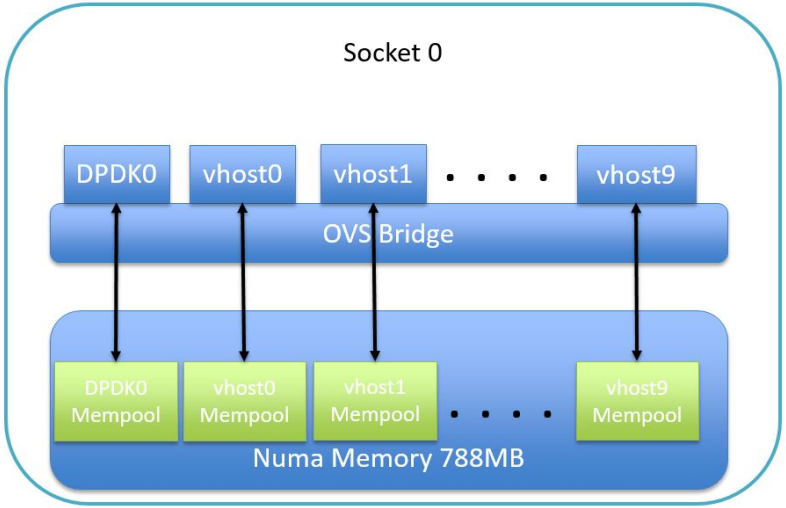
- Drawbacks
  - Memory footprint now impacted by
    - Num RX/TX queues, RX/TX queue size, Num of PMD etc.
  - Memory requirements change for a given deployment between OVS releases.

Shared VS Per Port Memory Footprint

Shared Mempool



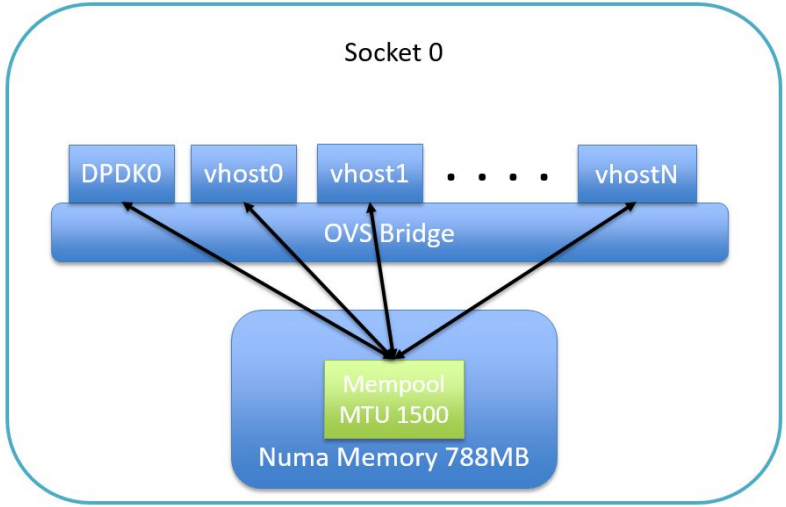
Per Port Mempool



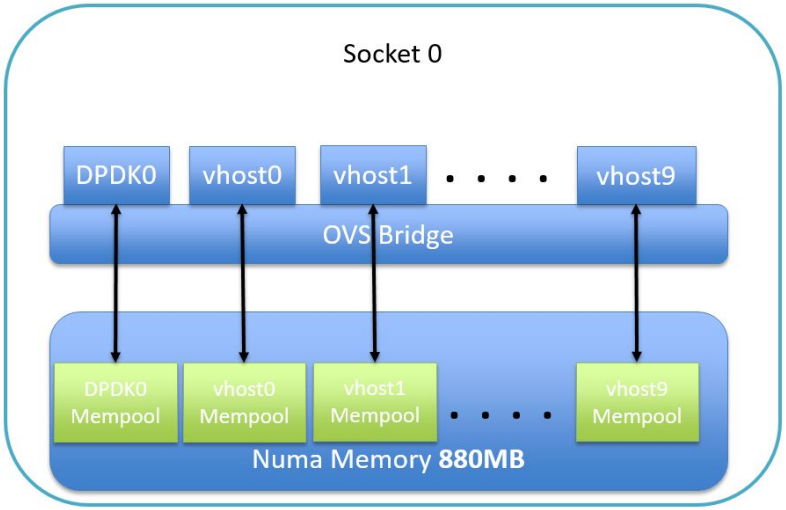
- MTU 1500
- 1 x PMD
- 1 x RXQ

Shared VS Per Port Memory Footprint

Shared Mempool



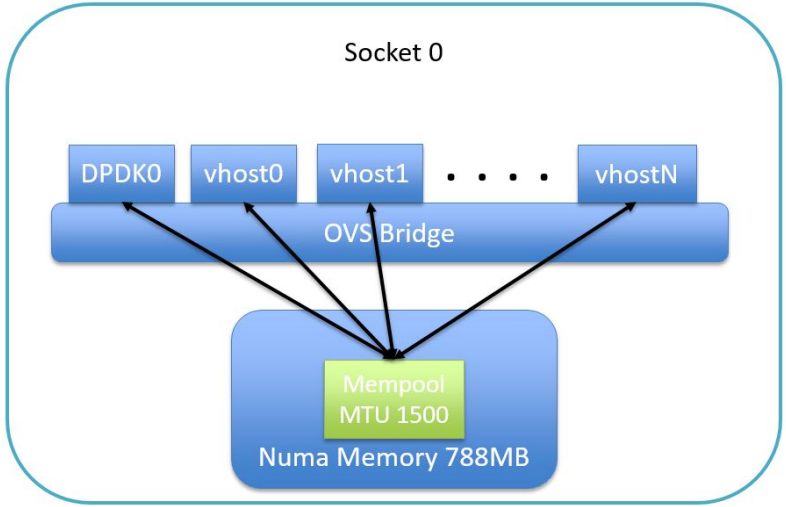
Per Port Mempool



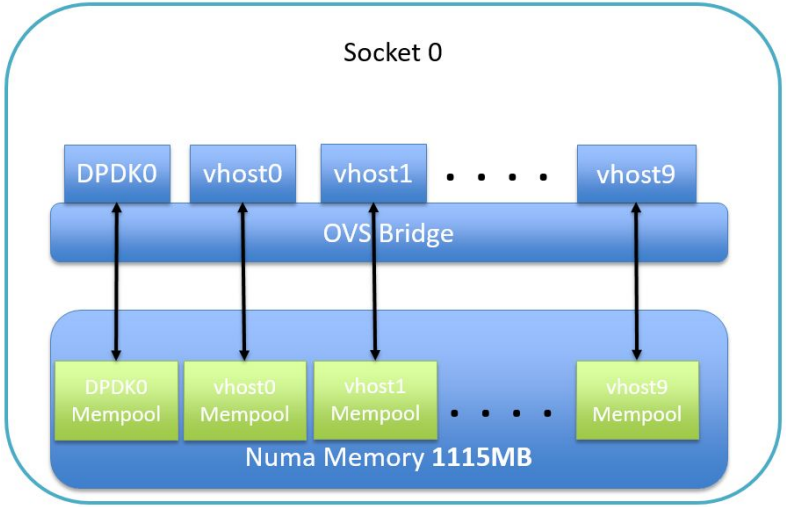
- MTU 1500
- 2 x PMD
- 2 x RXQ

Shared VS Per Port Memory Footprint

Shared Mempool



Per Port Mempool



- MTU 1500
- 4 x PMD
- 4 x RXQ

Memory Model Support to Date

- OVS 2.5 -> 2.9
  - Shared Memory model used

- OVS 2.10 provides support for both models
  - Shared memory enabled by default
  - Per port memory enabled by request

```
$ ovs-vsctl set Open_vSwitch . other_config:per-port-memory=true
```

Future Memory Models

- DPDK 18.05 reworked DPDK memory model
  - Hotplug capabality now available
  - Min and Max memory now provisioned for in dynamic manner.
  - Will be available to OVS via DPDK 18.11

- OVS DPDK Mempool re-design
  - Mempool per PMD?

How much hugepage memory ?

- Shared mempools
  - MTU’s, NUMA node of ports
- Per port mempools
  - Num of rxqs
  - Num of txqs
  - Size of rxqs/txqs
- Metadata / rounding at multiple layers
- Best to just estimate and test

Shared mempool estimation

- Mempools are per MTU, per NUMA
- Ports on 2 NUMA nodes with 9K MTU
- + metadata/rounding per buffer: 9KB → ~10KB
- Number of buffers in mempool: 256K
- 10KB * 256K = 2.7 GB per NUMA node
- If not available, retries for smaller size mempool

https://developers.redhat.com/blog/2018/03/16/ovs-dpdk-hugepage-memory/

Init

```
$ ovs-vsctl --no-wait set Open_vSwitch .
other_config:dpdk-socket-mem="4096,4096"
```

- Hugepages not mounted

```
|dpdk|INFO|EAL ARGS: ovs-vswitchd -c 0x1 --socket-mem 4096,4096
|dpdk|INFO|EAL: 32 hugepages of size 1073741824 reserved, but no
mounted hugetlbfs found for that size
```

- Not enough memory

```
|dpdk|INFO|EAL ARGS: ovs-vswitchd -c 0x1 --socket-mem 32768,0
|dpdk|ERR|EAL: Not enough memory available on socket 0! Requested:
32768MB, available: 4096MB
```

http://docs.openvswitch.org/en/latest/intro/install/dpdk/#setup-ovs

http://docs.openvswitch.org/en/latest/intro/install/dpdk/#setup-hugepages

Add port / Change MTU / Start VM

- May require creating a mempool
- May need to retry for smaller mempool

```
|dpdk|ERR|RING: Cannot reserve memory
```

- Retries might fail

```
|netdev_dpdk|ERR|Failed to create memory pool for netdev dpdk0, with
MTU 9000 on socket 0: Cannot allocate memory
```

Pool of buffers exhausted

- Excessive ports/queues/descriptor lengths

```
|dpdk|ERR|PMD: ixgbe_alloc_rx_queue_mbufs(): RX mbuf alloc failed
...
|netdev_dpdk|ERR|Interface dpdk0 start error: Input/output error
```

```
|dpdk(pmd91)|ERR|VHOST_DATA: Failed to allocate memory for mbuf.
```

- Use per port mempools
- Reduce queues/descriptor lengths

```
$ ovs-vsctl set Interface dpdk0 options:n_rxq=4
$ ovs-vsctl set Interface dpdk0 options:n_rxq_desc=1024
```

Further debug

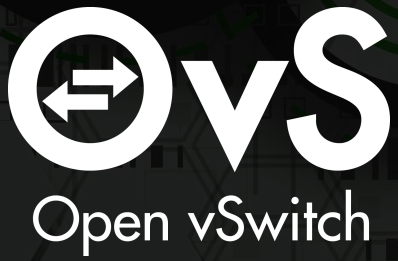
- Mempool create / reuse / free

```
$ ovs-appctl vlog/set netdev_dpdk:file:dbg
```

```
|netdev_dpdk|DBG|Allocated "ovs_mp_2030_0_262144" mempool with 262144
mbufs
|netdev_dpdk|DBG|Reusing mempool "ovs_mp_2030_0_262144"
|netdev_dpdk|DBG|Freeing mempool "ovs_mp_2030_0_262144"
```

- Mempool used by a port

```
$ ovs-appctl netdev-dpdk/get-mempool-info dpdk0
...
mempool <ovs_mp_2030_0_262144>
```



ian.stokes@intel.com
ktraynor@redhat.com