XDP Socket 在阿里云的发展

2021.10.24

丁雪峰（花名：玄拙）

Xuan Zhuo <xuanzhuo@linux.alibaba.com>

0. 前言

- 随着技术的发展，TCP 不再能满足各种技术场景，越来越多的基于 UDP 的用户态协议开始出现，但是 UDP 的性能就成为了一个急需要优化的领域。
- 我们选择 xdp socket 的最终的目的是实现对于 udp 的加速，为基于 udp 的用户态协议栈赋能。

0. 前言

选择 xdp socket

xdp socket 本身的优秀特点是我们选择 xdp socket 的原因

发展 xdp socket

推动了 xdp socket 在阿里云的发展

virtio 支持 xdp socket

促进我们推动 virtio-net 对 xdp socket 的支持

Express UDP

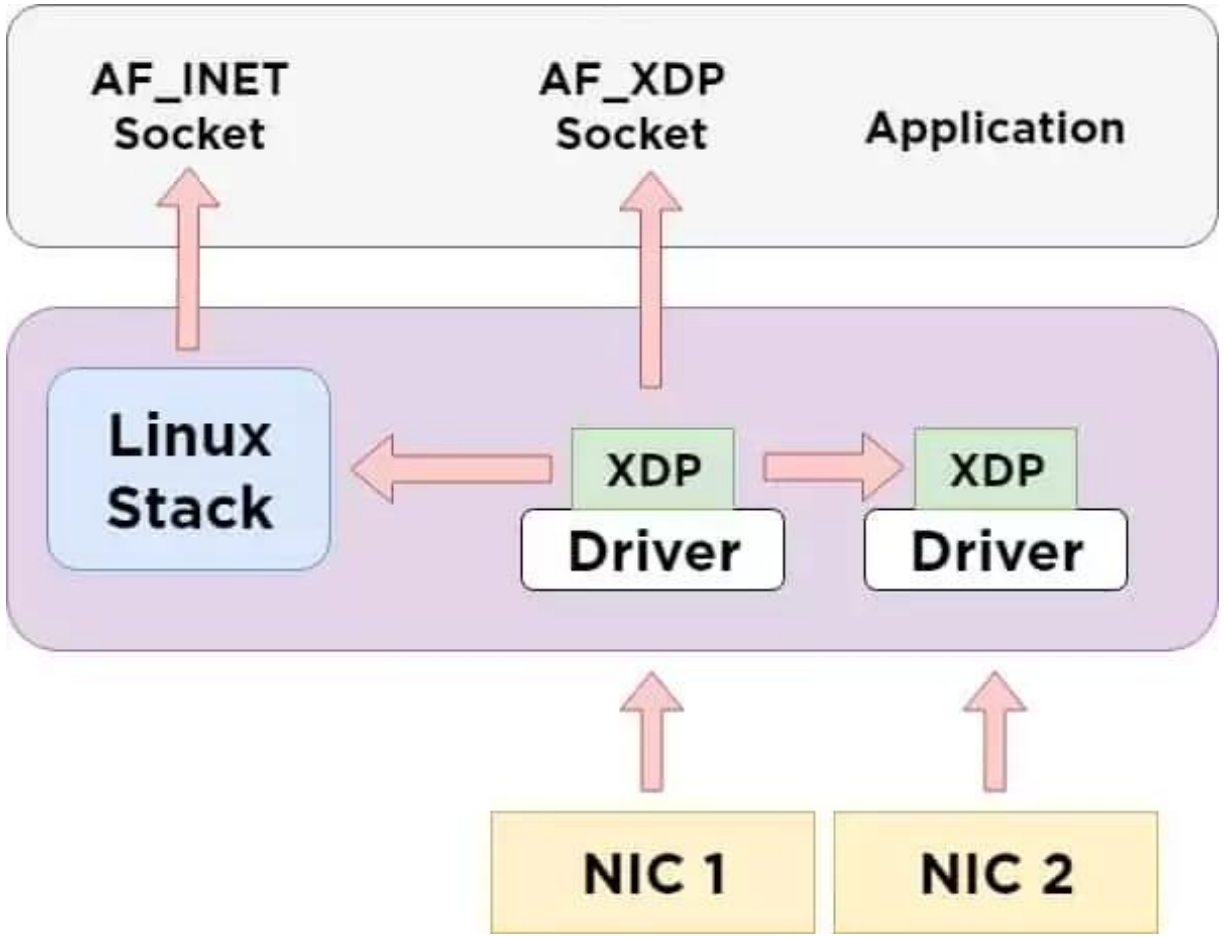
Express UDP 为 UDP 加速

Agenda

1. 为什么要 XDP Socket
2. XDP Socket 方面的工作
3. virtio 支持 XDP Socket
4. Express UDP
5. 下一步工作

1.1 XDP Socket

XDP Socket（AF_XDP）是一种新的 socket，最早出现在 Linux Kernel 4.18。它没有如同 DPDK 那样完全绕过内核，但是可以实现类似于 DPDK 的功能。



1.1 为什么需要 XDP Socket?

- 高吞吐，底消耗，底延时
  - 半 bypass 内核，跳过内核协议栈
  - zerocopy, mmap, xdp……
- 开发方便
  - 对外提供标准 socket，可以使用 socket api。与一般项目框架的兼容性好。
  - 支持 epoll、select、poll 等
- 使用方便
  - 不独占 CPU，网卡等资源
  - 开箱即用

1.2 XDP Socket 的特征

- 对上层表现为一个标准 socket
- 收包：基于 XDP 在驱动层过滤数据包并投递给 xdp socket
- 发包：两种模式
  - 驱动 tx napi poll 直接从 socket 里面获取数据包并 xmit
  - 构造 skb 进入 dev queue

Agenda

1. 为什么要 XDP Socket
2. XDP Socket 方面的工作
3. virtio 支持 XDP Socket
4. Express UDP
5. 下一步工作

2.1 我们的工作

特性支持：

- generic zerocopy xmit
- 虚拟化 DMA 支持

我们的工作部分 patch 已经提交到社区，还有一些 patch 等待后续的工作完成也会提交到社区。

主要包含新特性实现，bugfix，虚拟化支持等。

BugFix:

- Replace datagram_poll by sock_poll_wait
- Change the tx writeable condition
- Fix for xp_aligned_validate_desc() when len == chunk_size
- ……

2.2 xsk generic zerocopy xmit

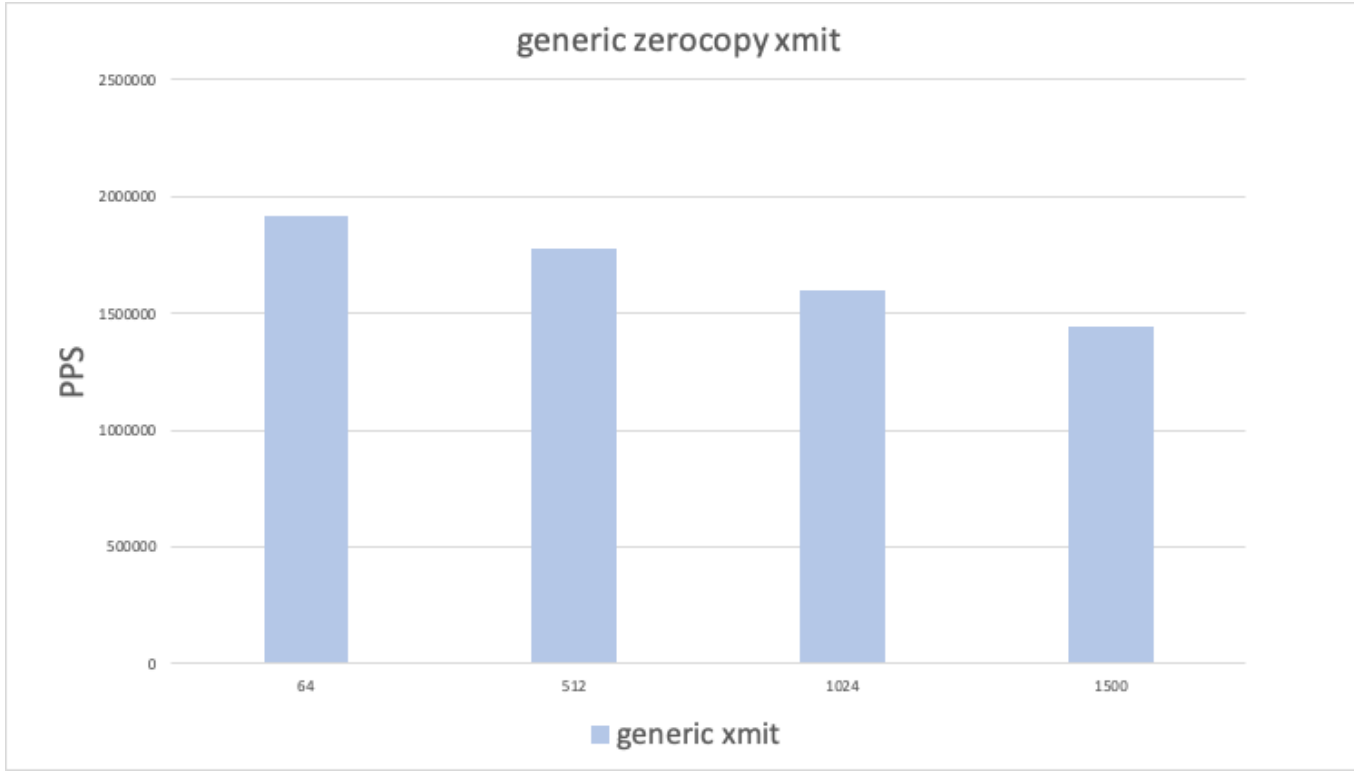
XDP socket 有两种工作模式

- generic xmit （驱动不支持 xdp socket， 要构造 skb 并 copy 数据）
- zerocopy xmit （驱动支持 xdp socket）

2.2 xsk generic zerocopy xmit

问题：xsk generic xmit 的性能受到 msg size 的影响。随着 msg size 增加，copy 的开销就会增加了，导致 PPS 下降。

解决方案：数据 page 作为 frag 直接插入 skb，实现 zerocopy。

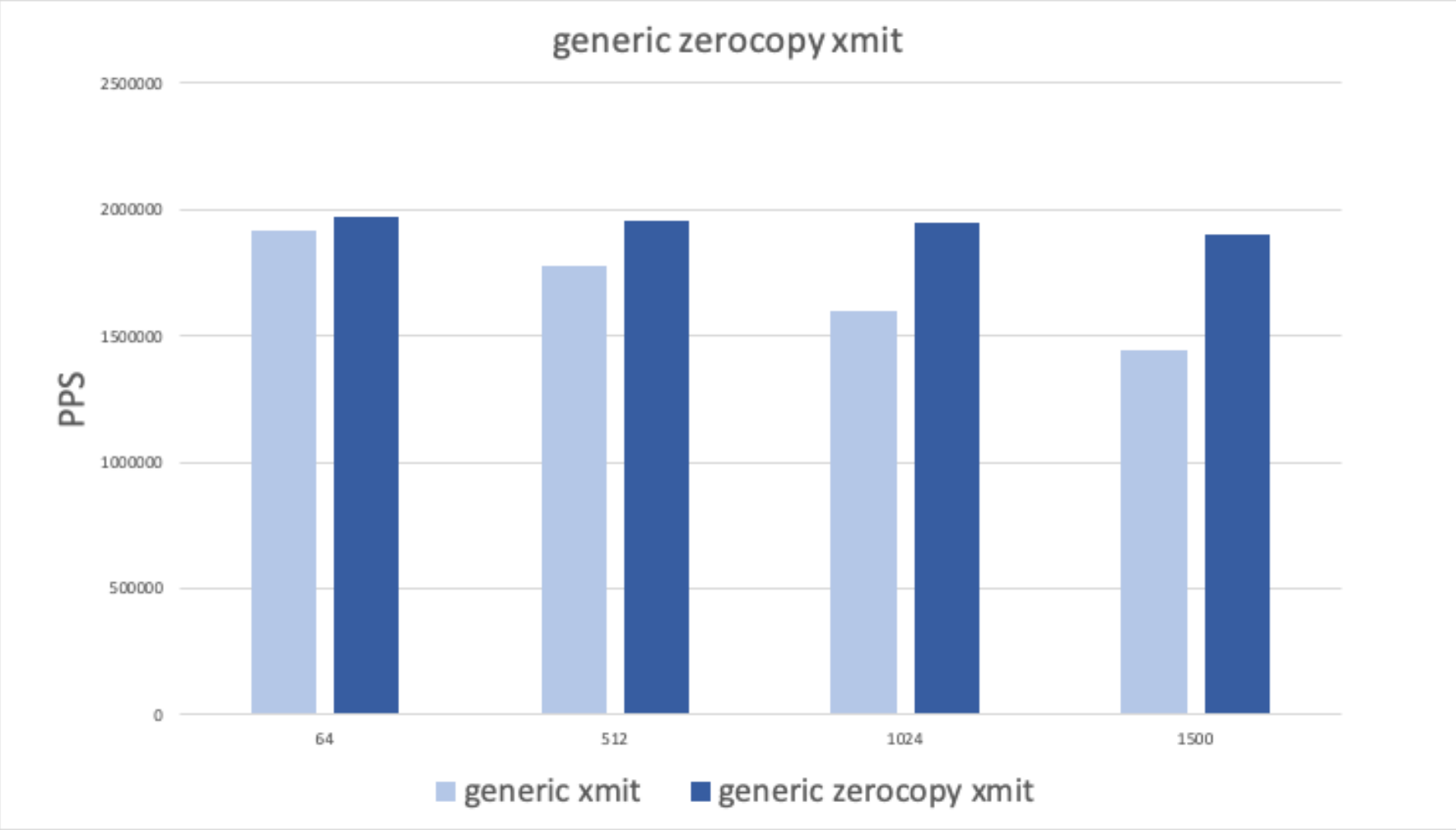


2.2 困难！！！

```
commit c2ff53d8049f30098153cd2d1299a44d7b124c57
Author: Xuan Zhuo <xuanzhuo@linux.alibaba.com>
Date:   Thu Feb 18 20:50:02 2021 +0000

    net: Add priv_flags for allow tx skb without linear

    In some cases, we hope to construct skb directly based on the existing
    memory without copying data. In this case, the page will be placed
    directly in the skb, and the linear space of skb is empty. But
    unfortunately, many the network card does not support this operation.
    For example Mellanox Technologies MT27710 Family [ConnectX-4 Lx] will
    get the following error message:

        mlx5_core 0000:3b:00.1 eth1: Error cqe on cqn 0x817, ci 0x8,
        qn 0x1dbb, opcode 0xd, syndrome 0x1, vendor syndrome 0x68
        00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        00000030: 00 00 00 00 60 10 68 01 0a 00 1d bb 00 0f 9f d2
        WQE DUMP: WQ size 1024 WQ cur size 0, WQE index 0xf, len: 64
        00000000: 00 00 0f 0a 00 1d bb 03 00 00 00 08 00 00 00 00
        00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        00000020: 00 00 00 2b 00 08 00 00 00 00 00 05 9e e3 08 00
        00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        mlx5_core 0000:3b:00.1 eth1: ERR CQE on SQ: 0x1dbb

    So a priv_flag is added here to indicate whether the network card
    supports this feature.

    Suggested-by: Alexander Lobakin <alobakin@pm.me>
    Signed-off-by: Xuan Zhuo <xuanzhuo@linux.alibaba.com>
    Signed-off-by: Alexander Lobakin <alobakin@pm.me>
    Signed-off-by: Daniel Borkmann <daniel@iogearbox.net>
    Acked-by: John Fastabend <john.fastabend@gmail.com>
    Link: https://lore.kernel.org/bpf/20210218204908.5455-3-alobakin@pm.me
```

```
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 3b6f82c2c271..6cef47b76cc6 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -1518,6 +1518,8 @@ struct net_device_ops {
  * @IFF_FAILOVER_SLAVE: device is lower dev of a failover master device
  * @IFF_L3MDEV_RX_HANDLER: only invoke the rx handler of L3 master device
  * @IFF_LIVE_RENAME_OK: rename is allowed while device is up and running
+ * @IFF_TX_SKB_NO_LINEAR: device/driver is capable of xmitting frames with
+ *     skb_headlen(skb) == 0 (data starts from frag0)
  */
 enum netdev_priv_flags {
        IFF_802_1Q_VLAN                 = 1<<0,
@@ -1551,6 +1553,7 @@ enum netdev_priv_flags {
        IFF_FAILOVER_SLAVE              = 1<<28,
        IFF_L3MDEV_RX_HANDLER           = 1<<29,
        IFF_LIVE_RENAME_OK              = 1<<30,
+       IFF_TX_SKB_NO_LINEAR            = 1<<31,
 };

 #define IFF_802_1Q_VLAN                        IFF_802_1Q_VLAN
@@ -1584,6 +1587,7 @@ enum netdev_priv_flags {
 #define IFF_FAILOVER_SLAVE             IFF_FAILOVER_SLAVE
 #define IFF_L3MDEV_RX_HANDLER          IFF_L3MDEV_RX_HANDLER
 #define IFF_LIVE_RENAME_OK             IFF_LIVE_RENAME_OK
+#define IFF_TX_SKB_NO_LINEAR           IFF_TX_SKB_NO_LINEAR
```

2.2 效果


generic zerocopy xmit
2500000
2000000
1500000
1000000
500000
0
PPS
64
512
1024
1500
generic xmit
generic zerocopy xmit

Agenda

1. 为什么要 XDP Socket
2. XDP Socket 方面的工作
3. virtio 支持 XDP Socket
4. Express UDP
5. 下一步工作

3.1 Virtio-net load xdp?

```
if (prog)
        xdp_qp = nr_cpu_ids;

/* XDP requires extra queues for XDP_TX */
if (curr_qp + xdp_qp > vi->max_queue_pairs) {
        NL_SET_ERR_MSG_MOD(extack, "Too few free TX rings available");
        netdev_warn(dev, "request %i queues but max is %i\n",
                    curr_qp + xdp_qp, vi->max_queue_pairs);
        return -ENOMEM;
}
```

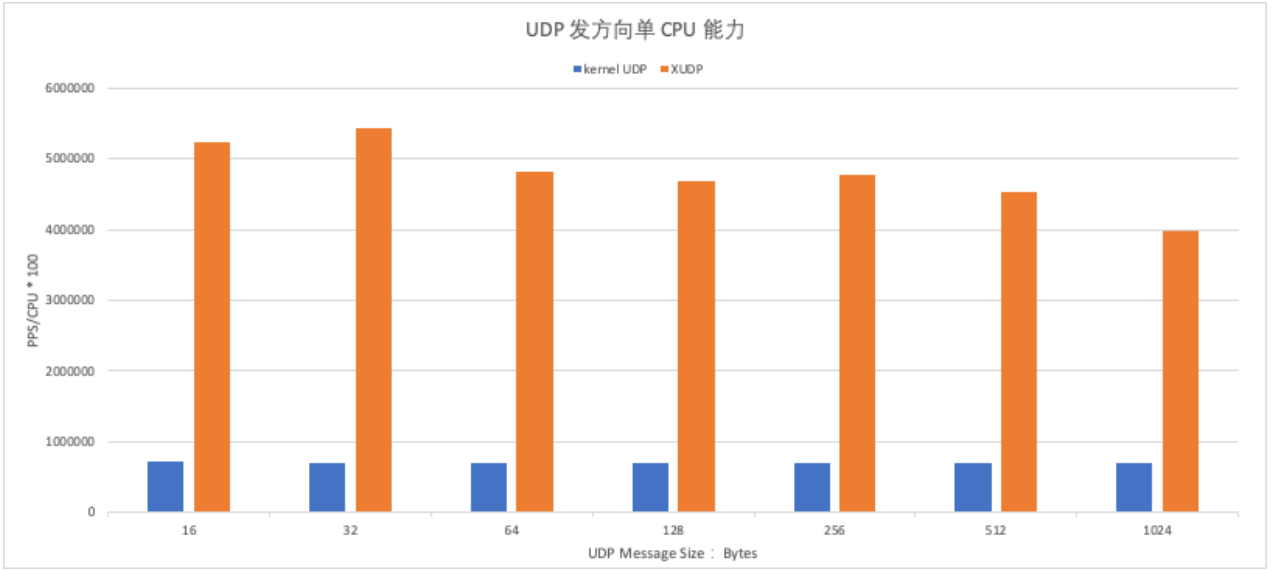
3.1 Virtio-net load xdp

```
commit 97c2c69e1926260c78c7f1c0b2c987934f1dc7a1
Author: Xuan Zhuo <xuanzhuo@linux.alibaba.com>
Date:   Wed Mar 10 10:24:45 2021 +0800

    virtio-net: support XDP when not more queues

    The number of queues implemented by many virtio backends is limited,
    especially some machines have a large number of CPUs. In this case, it
    is often impossible to allocate a separate queue for
    XDP_TX/XDP_REDIRECT, then xdp cannot be loaded to work, even xdp does
    not use the XDP_TX/XDP_REDIRECT.

    This patch allows XDP_TX/XDP_REDIRECT to run by reuse the existing SQ
    with __netif_tx_lock() hold when there are not enough queues.

    Signed-off-by: Xuan Zhuo <xuanzhuo@linux.alibaba.com>
    Reviewed-by: Dust Li <dust.li@linux.alibaba.com>
    Acked-by: Jason Wang <jasowang@redhat.com>
    Signed-off-by: David S. Miller <davem@davemloft.net>
```

3.2 Virtio-net support XDP Socket Zerocopy

我们在业界首次实现了 virtio-net 对于 XDP Socket zerocopy 的支持。

让 virtio-net 上运行 XDP socket 性能有了成倍的性能提升。



3.3 Virtio virtqueue reset

```
353 \subsection{Virtqueue Reset}\label{sec:Basic Facilities of a Virtio Device / Vi
354
355 When VIRTIO_F_RING_RESET is negotiated, the driver can reset a virtqueue
356 individually. The way to reset the virtqueue is transport specific.
357
358 Virtqueue reset is divided into two parts. The driver first resets a queue and
359 can afterwards optionally re-enable it.
```

3.3 Virtio virtqueue reset

```
Message-Id: <20210930175008.88352-1-xuanzhuo@linux.alibaba.com>
Date: Fri,  1 Oct 2021 01:50:05 +0800
From: Xuan Zhuo  @linux.alibaba.com
To:   cohuck     @redhat.com
      jasowang   @redhat.com
Cc:   virtio-dev @lists.oasis-open.org

[virtio-dev] [PATCH v5 0/3] virtio: introduce VIRTIO_F_RING_RESET for reset queue

================================================================================
Hi All:

This is a new version to support VIRTIO_F_RING_RESET. The feautre
extends the basic facility to allow the driver to reset a virtqueue.
This main motivation is to support the reset function of the queue of the
network device.

Please review.

v5:
   It is defined in the transports that the device can modify the default
   value after reset, and the driver can use a different configuration to
   re-enable the device.

v4:
   Cornelia Huck helped me more. Thanks.
   MMIO support this.

Thanks

Xuan Zhuo (3):
  virtio: introduce virtqueue reset as basic facility
  virtio: pci support virtqueue reset
  virtio: mmio support virtqueue reset

 content.tex | 111 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 1 file changed, 110 insertions(+), 1 deletion(-)

--
```

Agenda

1. 为什么要 XDP Socket
2. XDP Socket 方面的工作
3. virtio 支持 XDP Socket
4. Express UDP
5. 下一步工作

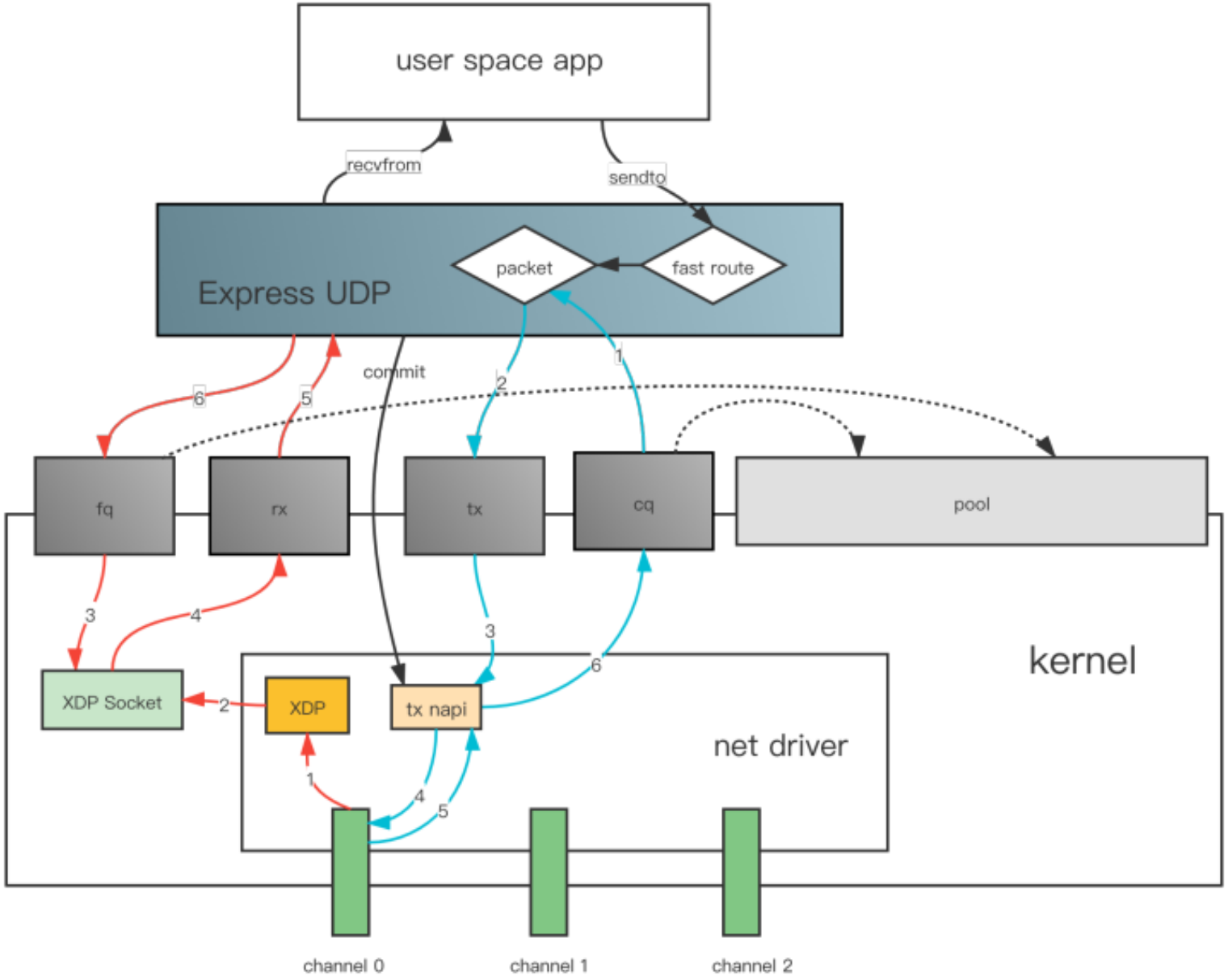
4. Express UDP（xudp）

- Express UDP（简称 xudp）是基于 XDP Socket（AF_XDP）实现的 bypass 内核的用户态的高性能 UDP 通信软件库。
- xudp 利用 XDP 和 XDP Socket 封装了一套高性能 UDP 通信接口，为应用提供了一种高性能 UDP 通信编程框架。可以为高 PPS UDP 通信场景带来网络性能上的显著提升。
- 随着 QUIC、低时延直播等的兴起，越来越多的应用开始采用 UDP 来进行通信。

4.1 Express UDP 优势

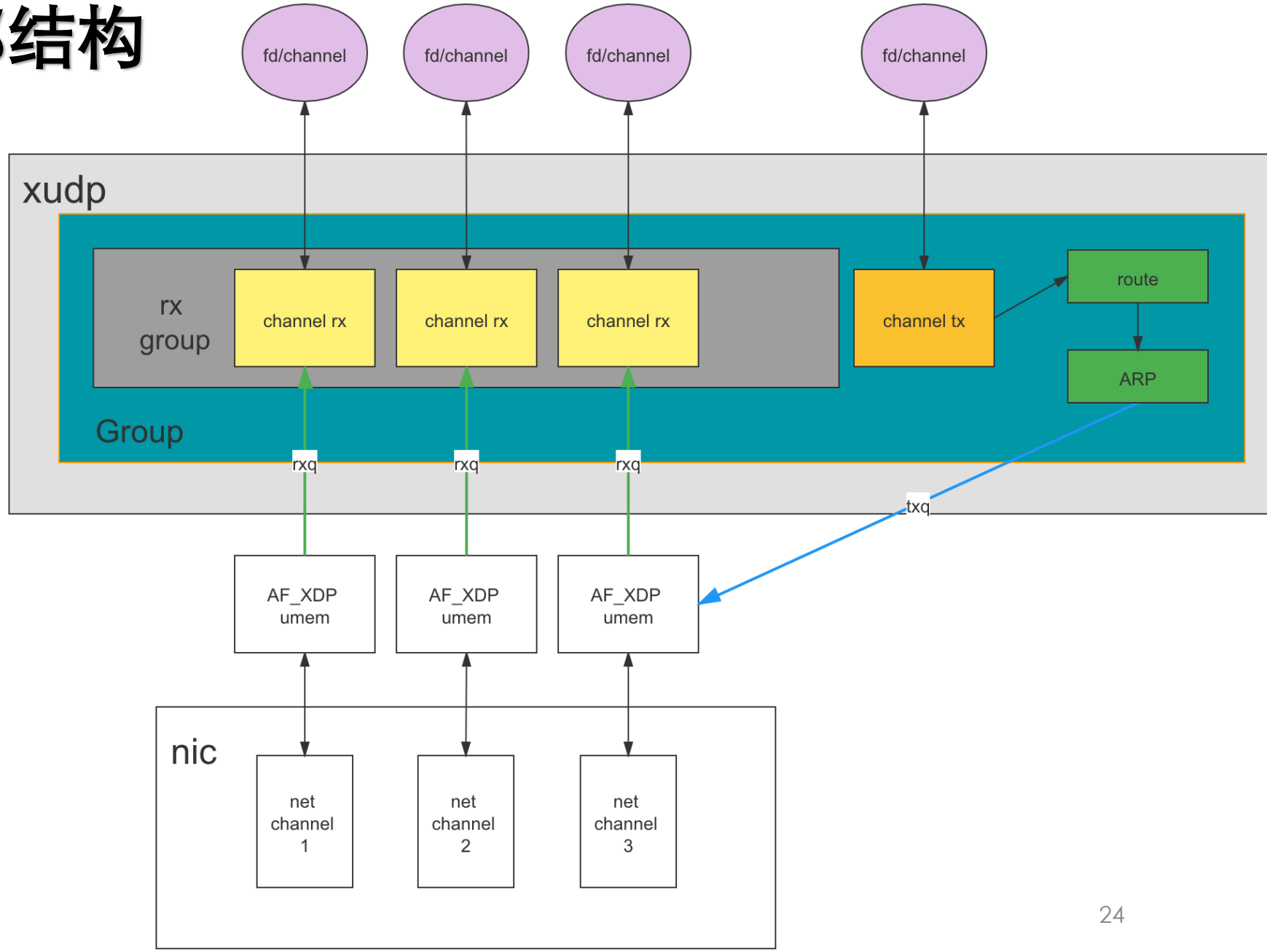
- 更低的传输时延与更低的 CPU 占用
- 驱动层 XDP 可以编程
- 运维更加方便，开发配合难度底

4.2 XUDP 全局结构

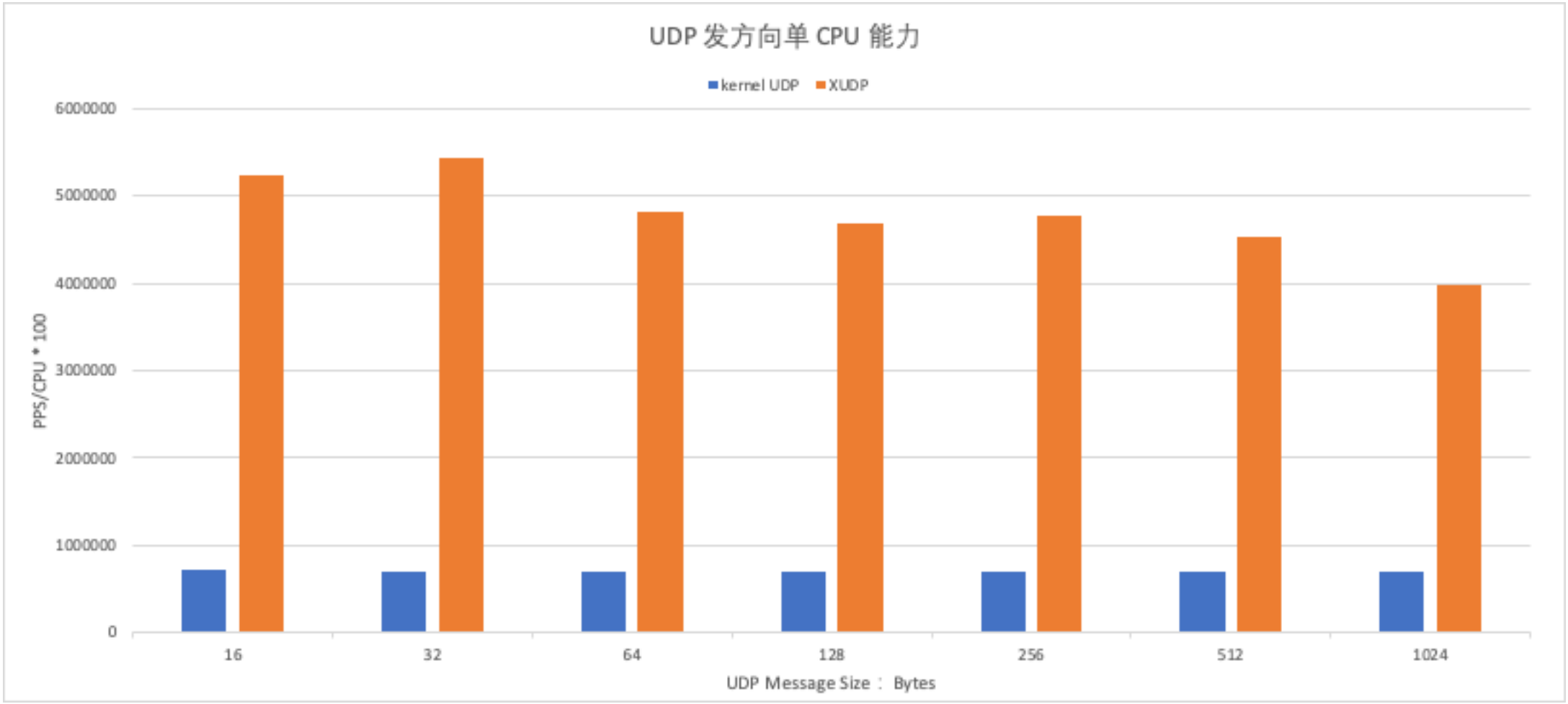


1. tx, rx, fq, cq, pool: 都是内核与用户态共享的内存
2. tx, rx, fq, cq: 都是队列, 队列里面的 item, 指向 pool 里面的内存
2. 只有 sendto 是 syscall, 并且不是每次都必须调用

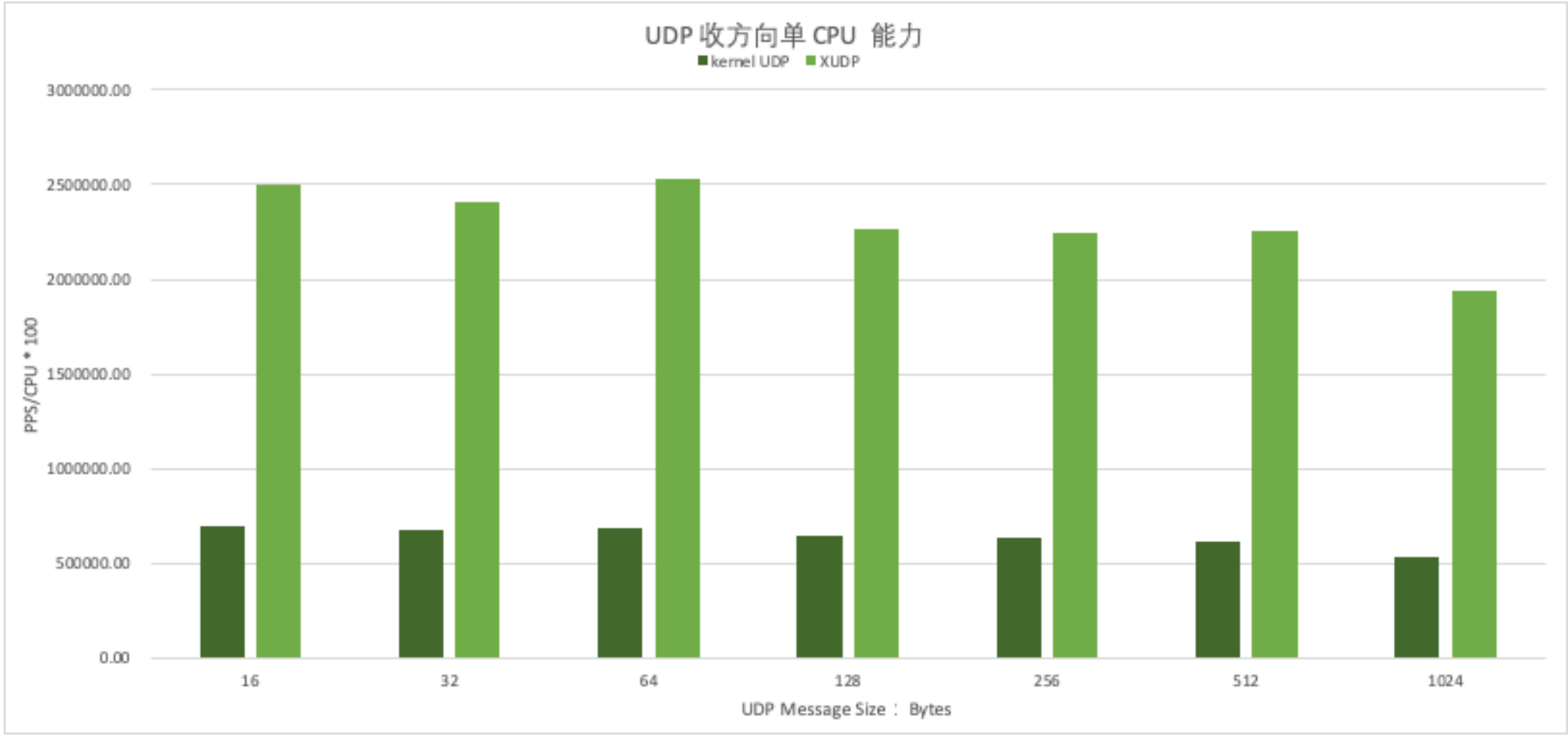
4.3 xudp 内部结构


fd/channel
fd/channel
fd/channel
fd/channel
xudp
rx
group
channel rx
channel rx
channel rx
channel tx
route
ARP
Group
rxq
rxq
rxq
txq
AF_XDP
umem
AF_XDP
umem
AF_XDP
umem
nic
net
channel
1
net
channel
2
net
channel
3

4.4 XUDP 性能数据 - send



4.5 XUDP 性能数据 - recv



4.6 XUDP 应用

基于以上的在 virtio_net 和 XDP Socket 的工作，xudp 结合阿里自研的 XQUIC 应用到了阿里巴巴的入口 QUIC 服务端上，共同支持阿里巴巴业务大规模落地 QUIC。

4.6 XUDP 对 XQUIC 的性能优化

提升 30.2%

xudp

```
$tsar --cpu --traffic -s bytin,bytout,retran,iseg,outseg --live -i1
Time                ----------------------cpu----------------------- -----traffic----
Time                  user     sys    wait    hirq    sirq    util    bytin   bytout
22/04/21-16:23:43     0.04    0.11    0.00    0.00    0.00    0.14    27.0K    13.3K
22/04/21-16:23:44     0.07    0.11    0.00    0.00    0.00    0.18     1.8K     2.7K
22/04/21-16:23:45    47.24   17.57    0.00    0.00    6.88   71.69     1.5M     4.1G
22/04/21-16:23:46    50.41   19.09    0.00    0.00    8.48   78.01     1.5M     4.5G
22/04/21-16:23:47    52.04   19.44    0.00    0.00    8.69   80.17     1.8M     4.6G
22/04/21-16:23:48    52.89   19.56    0.00    0.00    9.28   81.77     2.1M     4.6G
22/04/21-16:23:49    53.60   13.92    0.00    0.00    8.48   76.04     2.2M     4.6G
22/04/21-16:23:50    54.02   12.88    0.00    0.00    8.33   75.29     2.5M     4.7G
22/04/21-16:23:51    51.42   10.32    4.84    0.00    8.40   70.21     2.8M     4.5G
22/04/21-16:23:52    45.68    9.02    7.09    0.00    6.75   61.44     2.6M     4.0G
```

kernel udp

```
22/04/21-16:30:31    23.18   30.18    0.00    0.00    5.62   58.98     1.8M     2.8G
22/04/21-16:30:32    36.68   42.35    0.00    0.00    9.58   88.65     2.2M     4.0G
22/04/21-16:30:33    36.42   41.98    0.00    0.00    9.84   88.27     2.3M     4.0G
22/04/21-16:30:34    37.18   41.82    0.00    0.00   10.71   89.77     2.6M     4.0G
22/04/21-16:30:35    37.60   42.22    0.00    0.00   10.57   90.42     3.0M     4.1G
22/04/21-16:30:36    38.10   42.10    0.00    0.00   11.01   91.28     3.3M     4.1G
22/04/21-16:30:37    35.98   39.45    0.00    0.00    9.84   85.34     3.0M     3.9G
22/04/21-16:30:38    35.31   38.98    0.00    0.00    9.60   83.92     3.5M     3.8G
22/04/21-16:30:39    33.11   37.86    0.00    0.00    8.16   79.17     4.1M     3.7G
22/04/21-16:30:40    30.37   34.45    0.00    0.00    7.55   72.40     4.8M     3.3G
22/04/21-16:30:41    26.01   32.37    0.00    0.00    6.13   64.55     4.4M     3.1G
22/04/21-16:30:42    22.22   27.39    0.00    0.00    4.72   54.33     4.1M     2.6G
```

4.7 开源计划

- 内核相关的工作都已经推送到社区，后续工作也都会同步到 Linux 社区
- Express UDP 已经在 OpenAnolis 社区的“高性能网络技术”SIG 开源了

https://openanolis.cn/sig/high-perf-network

Agenda

1. 为什么要 XDP Socket
2. XDP Socket 方面的工作
3. virtio 支持 XDP Socket
4. Express UDP
5. 下一步工作

5. 下一步工作

- 基于 virtio 新的特性 virtqueue reset 实现 virtio-net 对于 XDP Socket 的支持
- 持续优化虚拟化环境下的 XDP Socket 及 Express UDP 的性能
- 推动 Express UDP 在更多场景的应用，比如视频、直播等

6. Q&A

开源及后续的工作都会在 OpenAnolis 社区的“高性能网络技术”SIG 呈现。

欢迎大家围观

Thanks

OpenAnolis
龙 蜥 社 区

阿里云

奥运会全球指定云服务商