

组会报告

徐益

2018 年 5 月 24 日

1 本周工作内容

1. 尝试扩大 DPDK 的中 mbuf
2. 实现分块传输和流量控制问题
3. 处理 makefile 相关问题

2 尝试扩大 DPDK 的中 mbuf

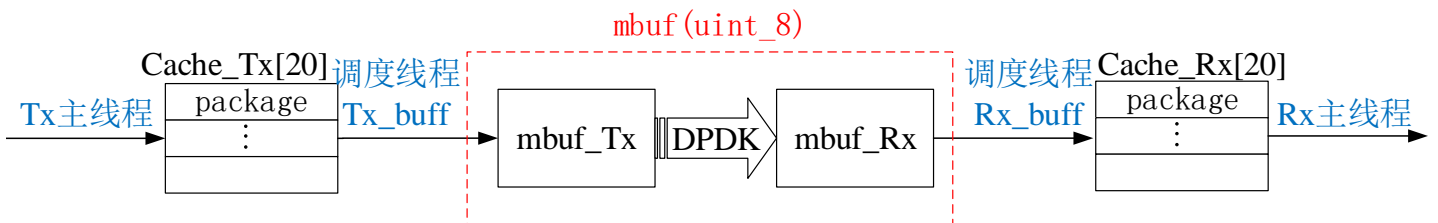


图 1: 原系统结构

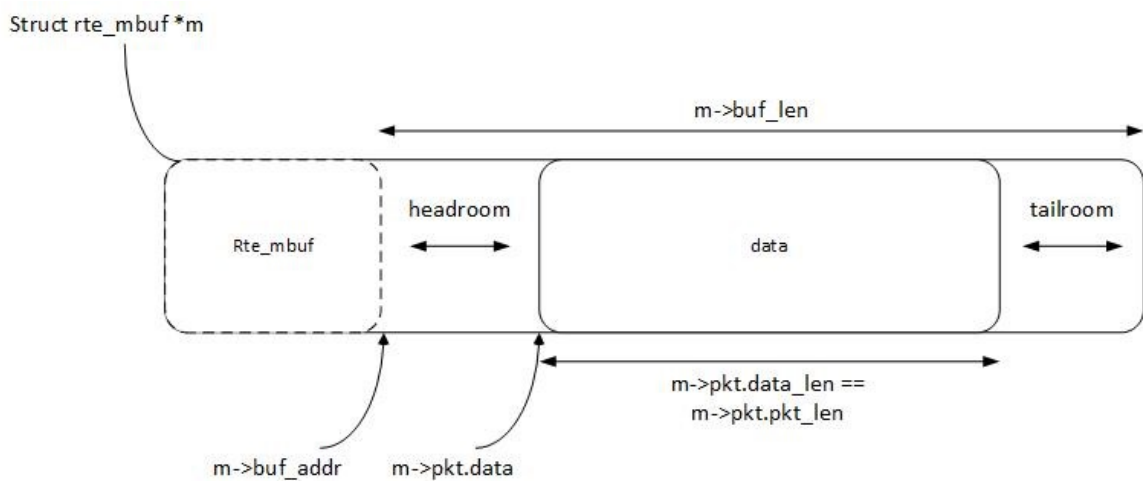


图 2: rtl_mbuf 结构

```
1 struct rte_mempool *  
2 rte_pktmbuf_pool_create(const char *name, unsigned n,  
3     unsigned cache_size, uint16_t priv_size, uint16_t data_room_size,  
4     int socket_id);
```

```
root@ubuntu: /home/xuyi/dataProcess

Port statistics =====
Statistics for port 0 -----
Packets sent:          0
Packets received:      0
Packets dropped:       0
Aggregate statistics =====
Total packets sent:    0
Total packets received: 0
Total packets dropped: 0
=====
2
send_rate= 0.000000 Gb
receive_rate= 0.000000 Gb
Segmentation fault (core dumped)
root@ubuntu:/home/xuyi/dataProcess#
```

图 3: 仅扩大 MBUF_SIZE

```
root@ubuntu: /home/xuyi/dataProcess
EAL: probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:04:00.1 on NUMA socket 0
EAL: probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:45:00.0 on NUMA socket 1
EAL: probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:45:00.1 on NUMA socket 1
EAL: probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:82:00.0 on NUMA socket 2
EAL: probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:82:00.1 on NUMA socket 2
EAL: probe driver: 8086:10fb net_ixgbe
RTE_MBUF_DEFAULT_BUF_SIZE:2176
mempool init done
ring_send create done
ring_receive create done
number of Ethernet ports that are available:4
Initializing port 0... PMD: ixgbe_alloc_rx_queue_mbufs(): RX mbuf alloc failed q
ueue_id=0
PMD: ixgbe_dev_rx_queue_start(): Could not alloc mbuf for queue:0
PMD: ixgbe_dev_start(): Unable to start rxtx queues
PMD: ixgbe_dev_start(): failure in ixgbe_dev_start(): -1
EAL: Error - exiting with code: 1
Cause: rte_eth_dev_start:err=-5, port=0
root@ubuntu:/home/xuyi/dataProcess#
```

图 4: 扩大 MBUF_SIZE 的同时缩小 NB_MBUF

3 分段传输方案

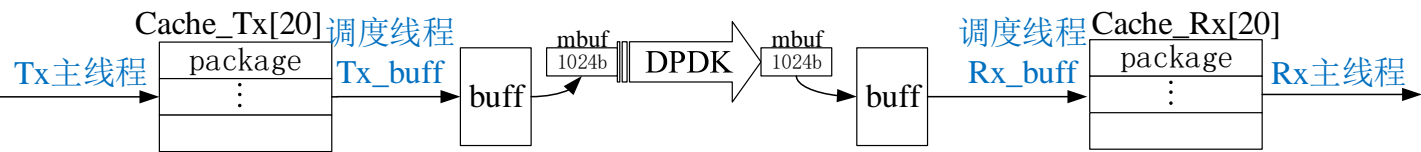


图 5: 分段传输方案系统结构

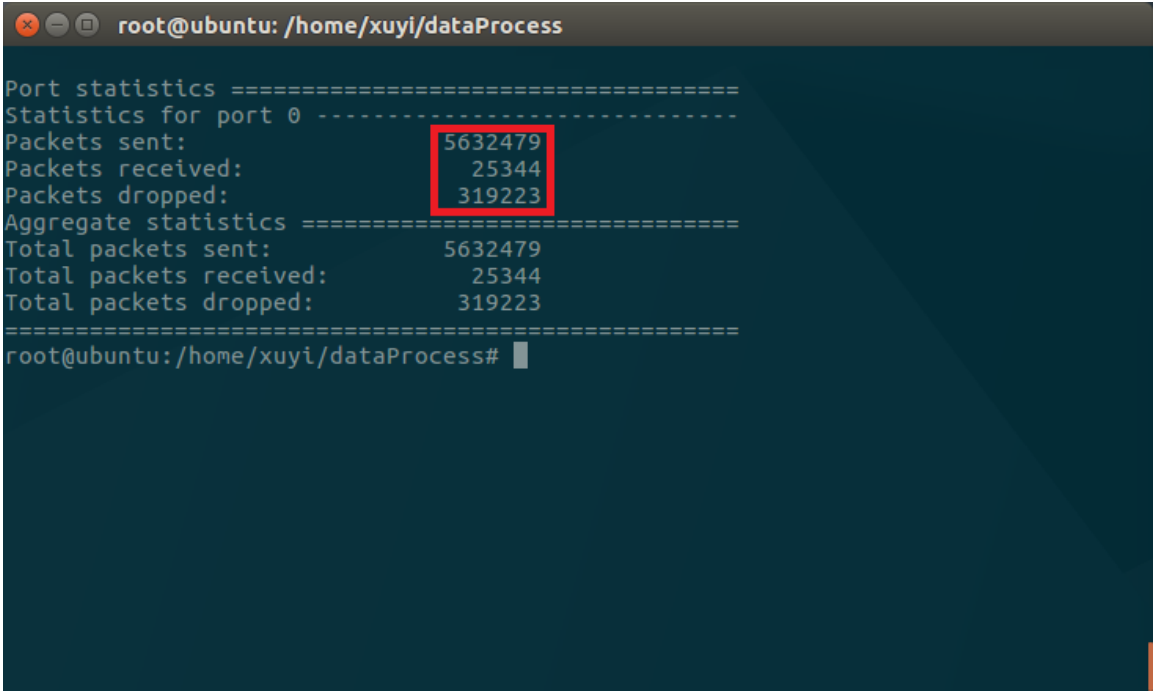


图 6: 未进行流量控制

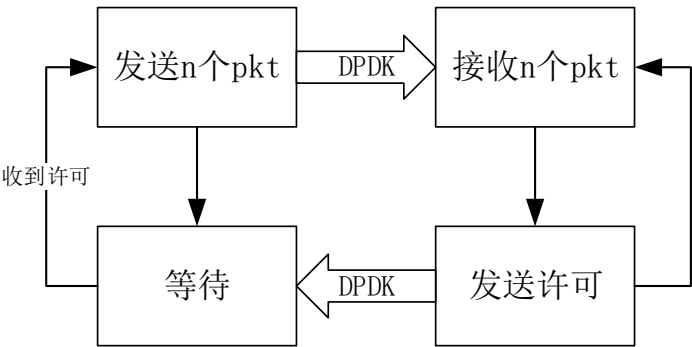


图 7: 流量控制方案

```
root@ubuntu: /home/xuyi/traffic_control/dataProcess_send

Port statistics =====
Statistics for port 0 -----
Packets sent:          78471
Packets received:      78471
Packets dropped:        0
Aggregate statistics =====
Total packets sent:      78471
Total packets received:  78471
Total packets dropped:   0
=====
root@ubuntu: /home/xuyi/traffic_control/dataProcess_send#

root@ubuntu: /home/xuyi/traffic_control/dataProcess_receive

Port statistics =====
Statistics for port 0 -----
Packets sent:          78471
Packets received:      78471
Packets dropped:        0
Aggregate statistics =====
Total packets sent:      78471
Total packets received:  78471
Total packets dropped:   0
=====
root@ubuntu: /home/xuyi/traffic_control/dataProcess_receive#
```

图 8: 进行流量控制后

4 makefile 相关问题

```
M Makefile x
35 RTE_SDK ?= $(HOME)/dpdk
36
37 # Default target, can be overridden by command line or environment
38 RTE_TARGET ?= x86_64-native-linuxapp-gcc
39
40 include $(RTE_SDK)/mk/rte.vars.mk
41
42 # binary name
43 APP = l2fwd
44
45 # all source are stored in SRCS-y
46 SRCS-y := main.c
47
48 CFLAGS += -O3
49 CFLAGS += $(WERROR_FLAGS)
50
51 include $(RTE_SDK)/mk/rte.extapp.mk
52
53
M Makefile x
1 OBJS = crc.o bit.o cbsegm.o debug.o turbocoder.o vector.o tc_interl_ums.o tc_interl_
2 DEST = main
3 LIBS = -lm -lpthread -lmkl_rt -msse4.1 -fopenmp
4
5 CPPFLAGS = -g -Wall -I . -msse4.1 -O3 -mavx
6 main: $(OBJS)
7 gcc -o $(DEST) $(OBJS) $(LIBS)
8 clean:
9 rm -f $(DEST) $(OBJS)
10
```

图 9: DPDK 和编码调制系统 makefile 对比

```
23 # A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
24 # OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
25 # SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
26 # LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
27 # DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
28 # THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
29 # (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
30 # OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
31
32 # ifeq ($(RTE_SDK),)
33 # $(error "Please define RTE_SDK environment variable")
34 # endif
35
36 RTE_SDK ?= $(HOME)/dpdk
37
38 # Default target, can be overridden by command line or environment
39 RTE_TARGET ?= x86_64-native-linuxapp-gcc
40
41 include $(RTE_SDK)/mk/rte.vars.mk
42
43 # binary name
44 APP = dataProcess
45
46 # all source are stored in SRCS-y
47 SRCS-y := main.c crc.c bit.c cbsegm.c debug.c turbocoder.c vector.c tc_interl_ums.c tc_interl_lte.c rm_turbo.c MQAMMod.c phy_common.c sequence.c turbodecoder.c turbodecoder_gen.c turbodecod
48
49 LTB += -lm -lpthread -lmkl_rt -msse4.1 -fopenmp
50 CFLAGS += -g -Wall -I. -msse4.1 -O3 -maxx
51
52 CFLAGS += -I$(SRCDIR)
53 CFLAGS += -O3 $(USER_FLAGS)
54 CFLAGS += $(WERROR_FLAGS)
55
56 include $(RTE_SDK)/mk/rte.extapp.mk
57
```

图 10: 尝试合并的 makefile

```
root@ubuntu: /home/xuyi/dataProcess_send
/home/xuyi/dataProcess_send/ChannelEstimator.c:413: undefined reference to `Dfti
ComputeBackward'
/home/xuyi/dataProcess_send/ChannelEstimator.c:414: undefined reference to `Dfti
ComputeBackward'
/home/xuyi/dataProcess_send/ChannelEstimator.c:415: undefined reference to `Dfti
FreeDescriptor'
ChannelEstimator_LS.o: In function `Cal_DCT_FFT':
/home/xuyi/dataProcess_send/ChannelEstimator_LS.c:224: undefined reference to `D
ftiCreateDescriptor_s_1d'
/home/xuyi/dataProcess_send/ChannelEstimator_LS.c:225: undefined reference to `D
ftiCommitDescriptor'
/home/xuyi/dataProcess_send/ChannelEstimator_LS.c:246: undefined reference to `D
ftiComputeForward'
/home/xuyi/dataProcess_send/ChannelEstimator_LS.c:247: undefined reference to `D
ftiComputeForward'
/home/xuyi/dataProcess_send/ChannelEstimator_LS.c:256: undefined reference to `D
ftiFreeDescriptor'
TaskScheduler.o: In function `TaskScheduler_tx':
/home/xuyi/dataProcess_send/TaskScheduler.c:579: undefined reference to `cblas_c
gemm'
collect2: 错误: ld 返回 1
make[1]: *** [dataProcess] Error 1
make: *** [all] Error 2
root@ubuntu: /home/xuyi/dataProcess_send#
```

图 11: 编译报错

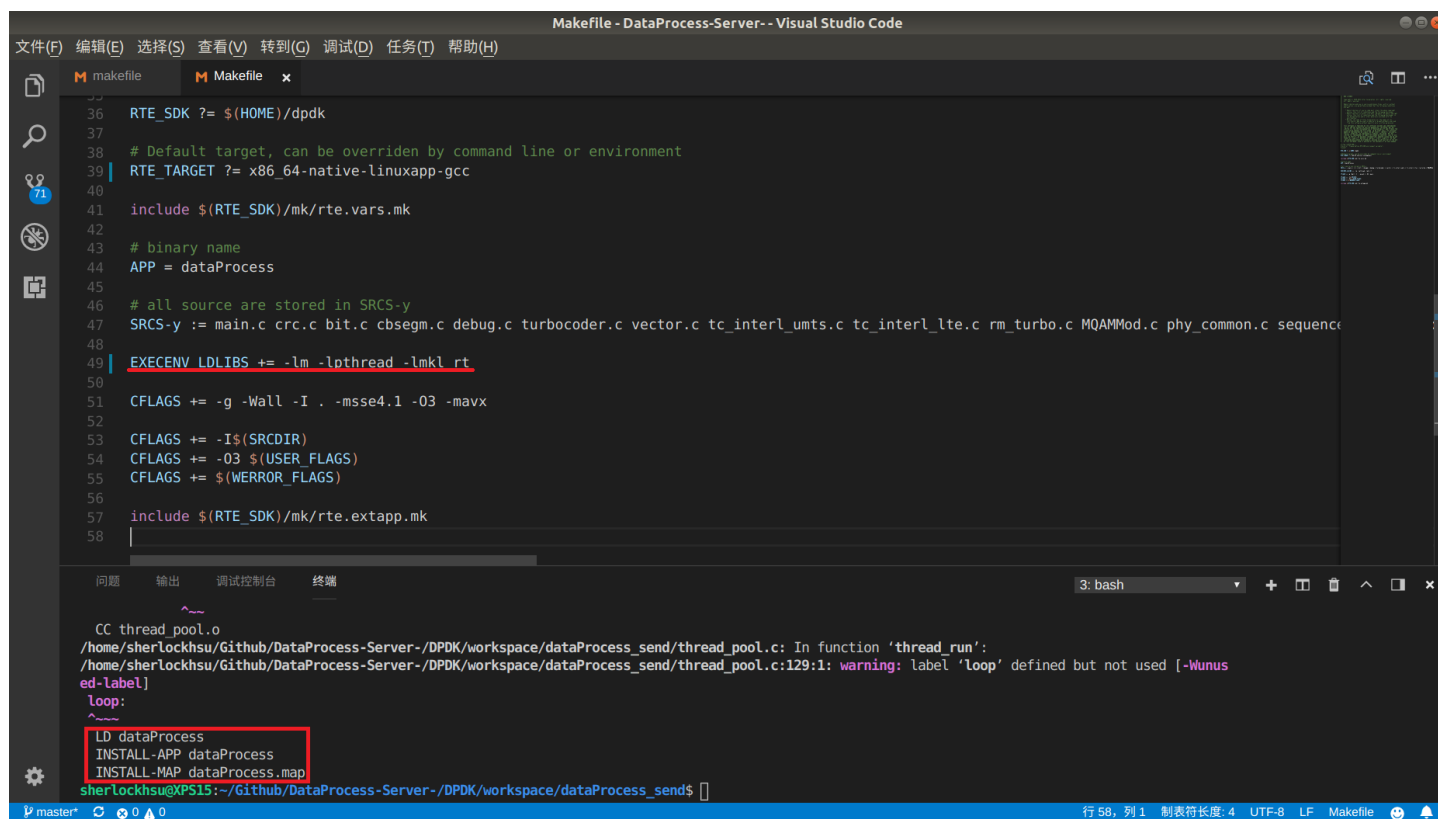


图 12: 最终解决

5 其他改进方向

1. 选择更大的 DPDK 发送页。
2. 选择更优的流量控制策略。

6 下周计划

1. 继续完成数据处理 +DPDK 系统
2. 学习 LDPC 相关内容