组会报告

徐益

2018 年 5 月 23 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

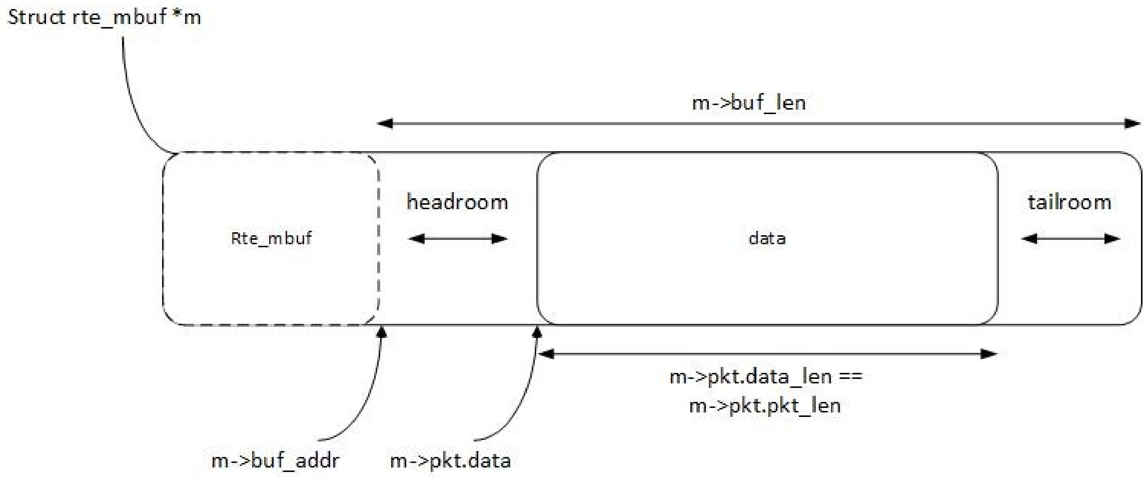
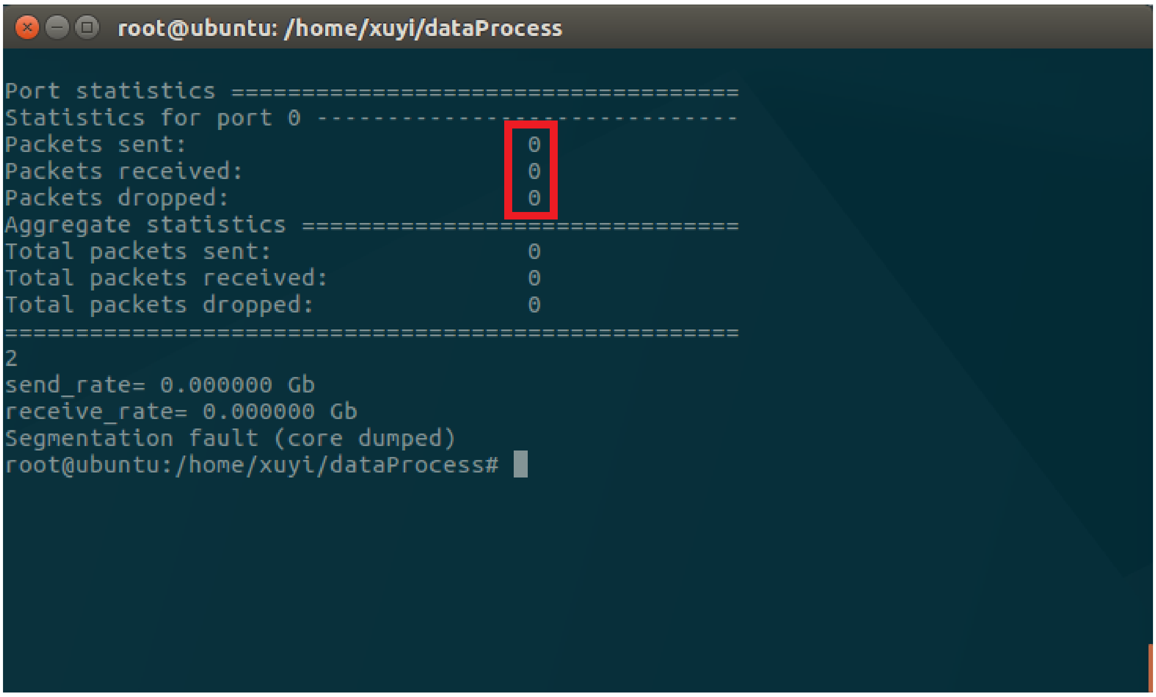


图 1: rtl_mbuf 结构

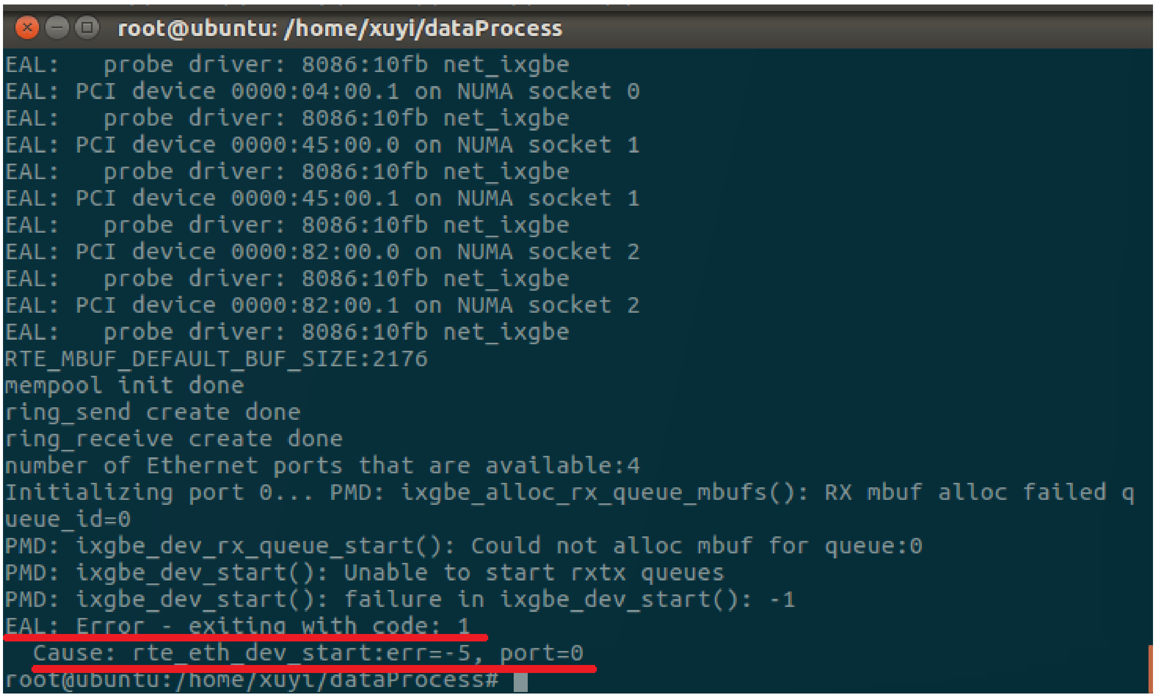
```
/**
* @param name
*   The name of the mbuf pool.
* @param n
*   The number of elements in the mbuf pool. The optimum size (in terms
*   of memory usage) for a mempool is when n is a power of two minus one:
*   n = (2^q - 1).
* @param cache_size
*   Size of the per-core object cache. See rte_mempool_create() for
*   details.
* @param priv_size
*   Size of application private are between the rte_mbuf structure
*   and the data buffer. This value must be aligned to RTE_MBUF_PRIV_ALIGN.
* @param data_room_size
*   Size of data buffer in each mbuf, including RTE_PKTMBUF_HEADROOM.
```

```
 * @param socket_id
 *   The socket identifier where the memory should be allocated. The
 *   value can be *SOCKET_ID_ANY* if there is no NUMA constraint for the
 *   reserved zone.
 */
struct rte_mempool *
rte_pktmbuf_pool_create(const char *name, unsigned n,
        unsigned cache_size, uint16_t priv_size, uint16_t data_room_size,
        int socket_id);
```



```
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 2: 仅扩大 MBUF_SIZE



```
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 3: 扩大 MBUF_SIZE 的同时缩小 NB_MBUF

3 分段传输方案

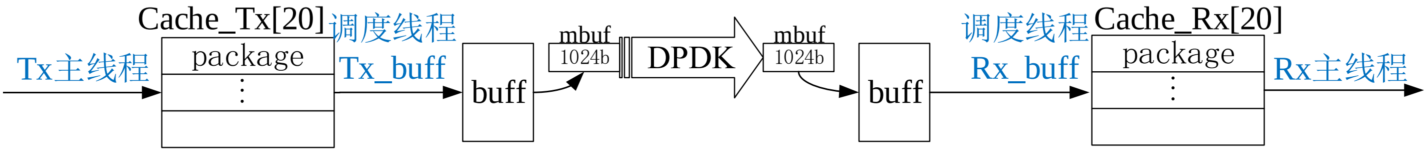
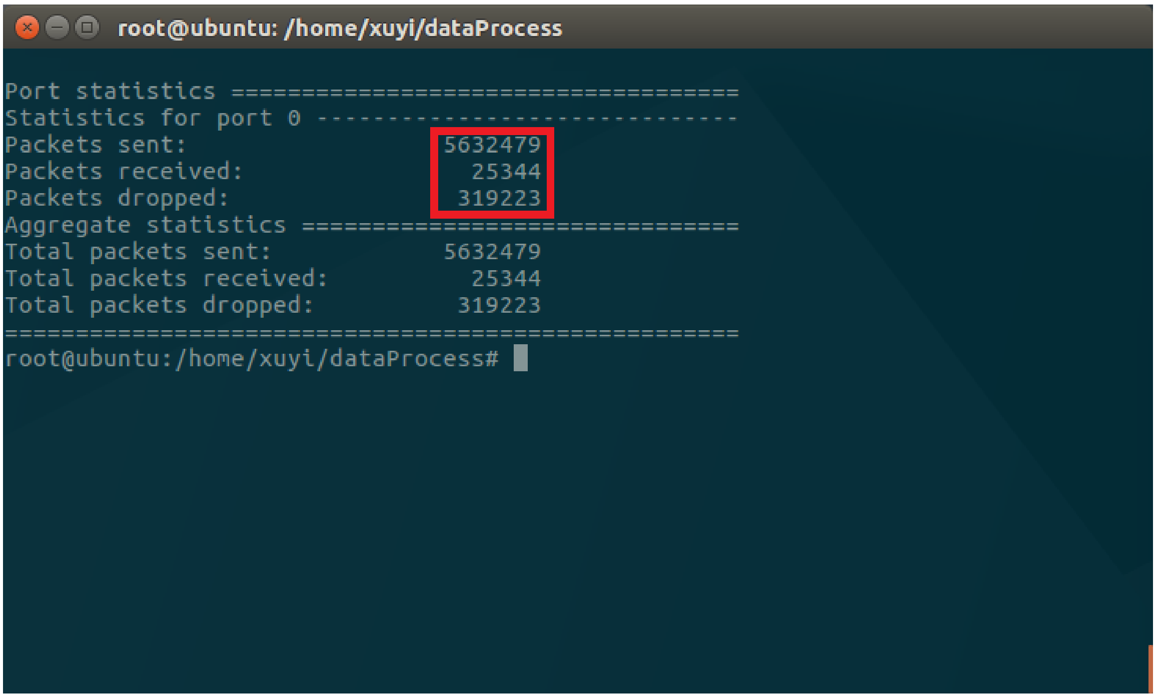


图 4: 分段传输方案系统结构



```
Port statistics ====================================
Statistics for port 0 ------------------------------
Packets sent:                       5632479
Packets received:                     25344
Packets dropped:                     319223
Aggregate statistics ===============================
Total packets sent:                 5632479
Total packets received:               25344
Total packets dropped:               319223
====================================================
root@ubuntu:/home/xuyi/dataProcess# 
```

图 5: 未进行流量控制

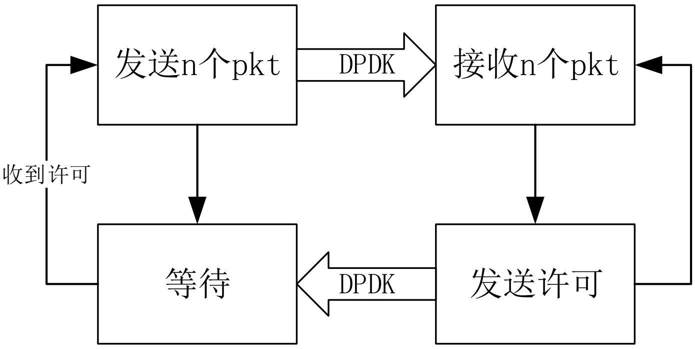
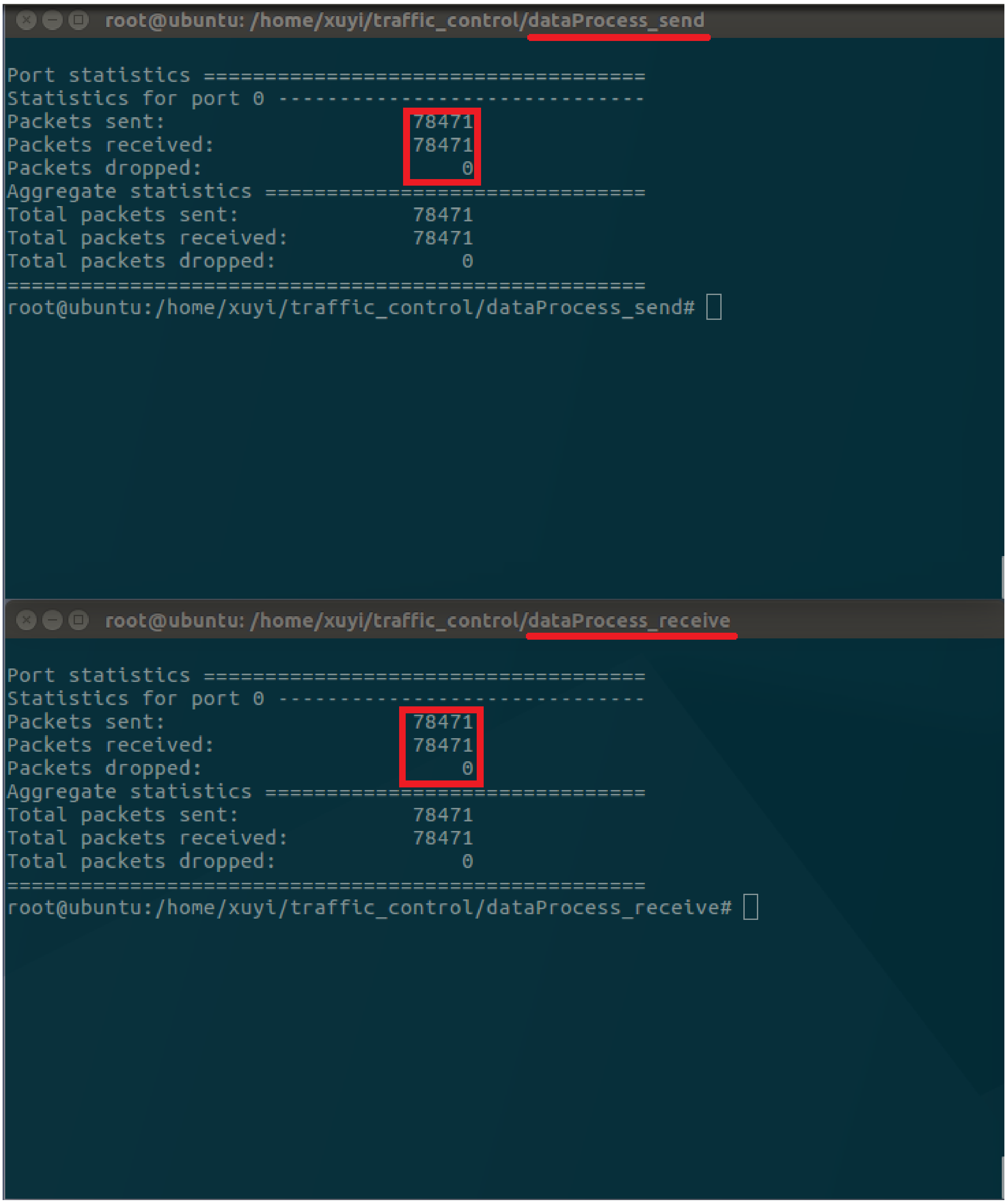


图 6: 流量控制方案



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

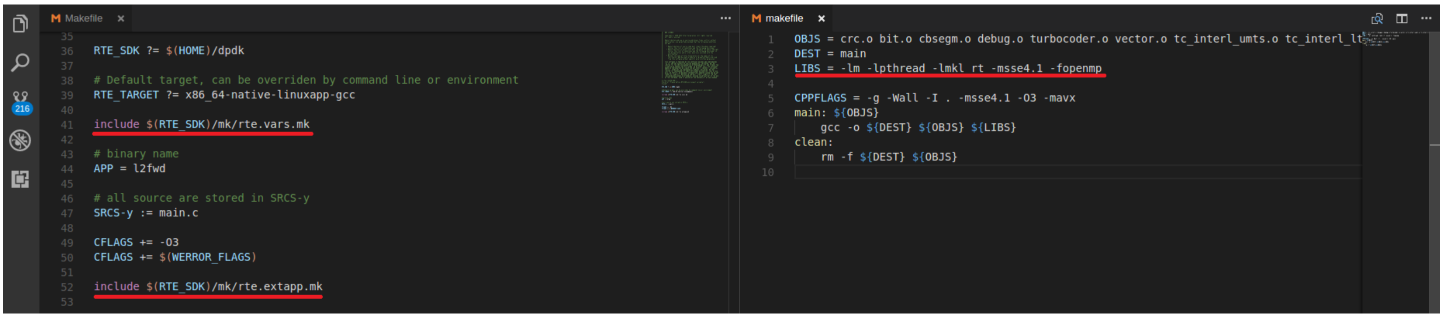
```
Port statistics ====================================
Statistics for port 0 ------------------------------
Packets sent:                      78471
Packets received:                  78471
Packets dropped:                       0
Aggregate statistics ===============================
Total packets sent:                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 7: 进行流量控制后

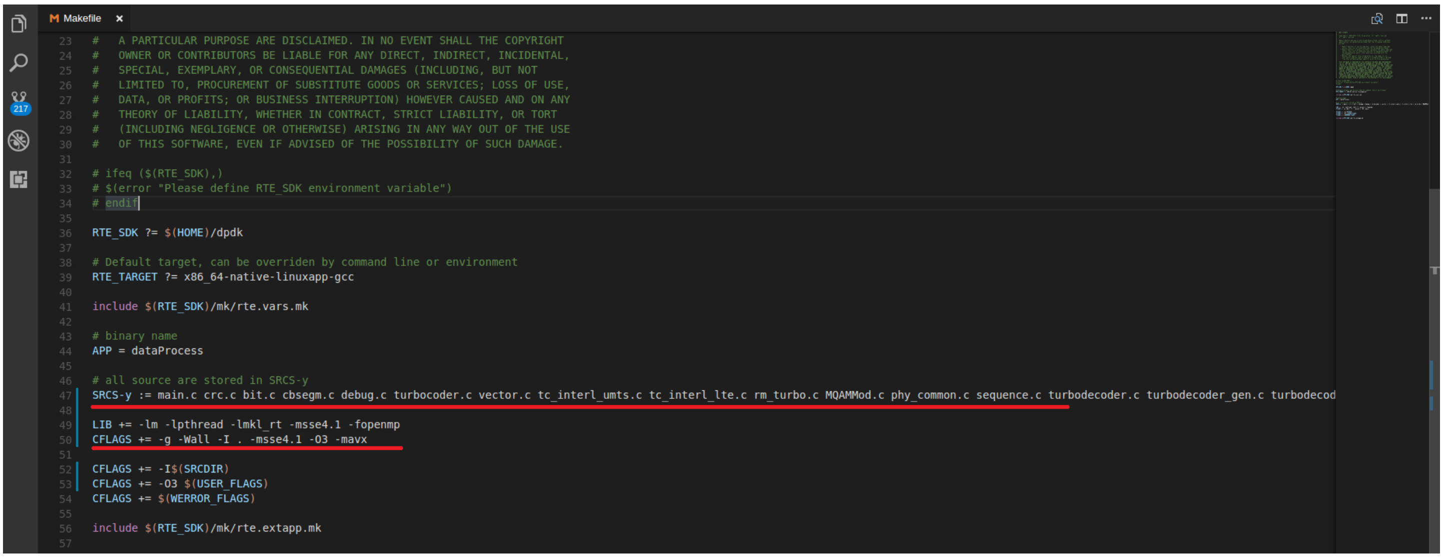
4 makefile 相关问题



```

RTE_SDK ?= $(HOME)/dpdk

# Default target, can be overriden by command line or environment
RTE_TARGET ?= x86_64-native-linuxapp-gcc

include $(RTE_SDK)/mk/rte.vars.mk

# binary name
APP = l2fwd

# all source are stored in SRCS-y
SRCS-y := main.c

CFLAGS += -O3
CFLAGS += $(WERROR_FLAGS)

include $(RTE_SDK)/mk/rte.extapp.mk

```

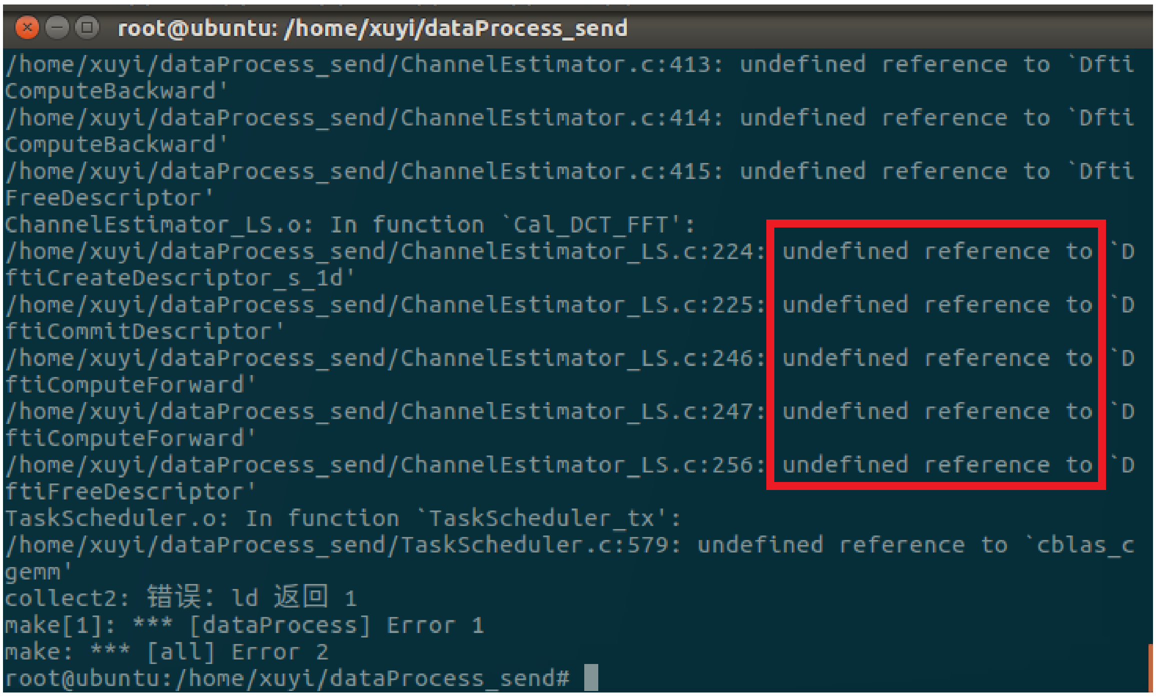
```
OBJS = crc.o bit.o cbsegm.o debug.o turbocoder.o vector.o tc_interl_umts.o tc_interl_lt
DEST = main
LIBS = -lm -lpthread -lmkl_rt -msse4.1 -fopenmp

CPPFLAGS = -g -Wall -I . -msse4.1 -O3 -mavx
main: ${OBJS}
    gcc -o ${DEST} ${OBJS} ${LIBS}
clean:
    rm -f ${DEST} ${OBJS}

```

图 8: DPDK 和编码调制系统 makefile 对比



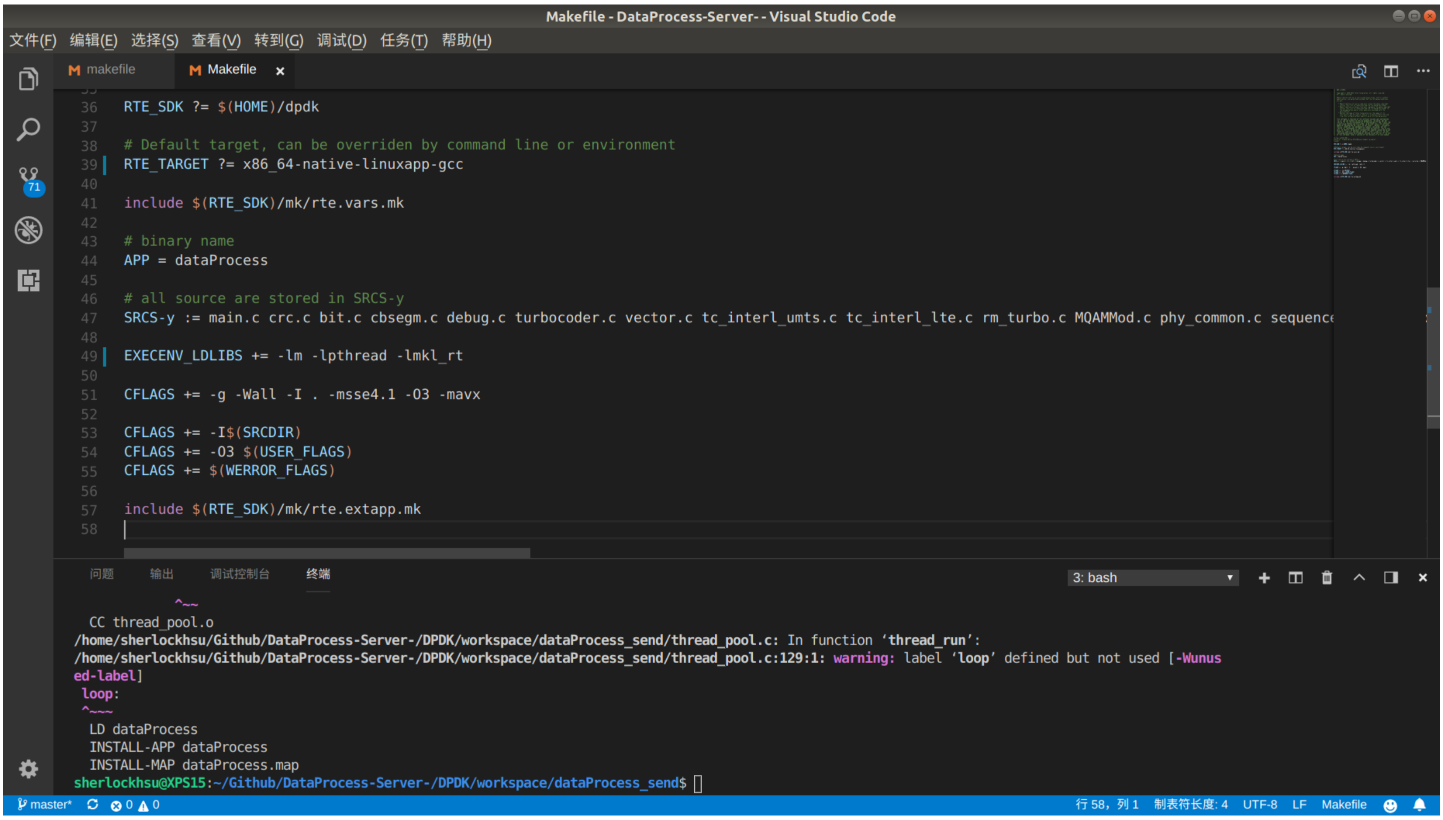
```
#   A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
#   OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
#   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
#   LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
#   DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
#   THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
#   (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
#   OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# ifeq ($(RTE_SDK),)
# $(error "Please define RTE_SDK environment variable")
# endif

RTE_SDK ?= $(HOME)/dpdk

# Default target, can be overriden by command line or environment
RTE_TARGET ?= x86_64-native-linuxapp-gcc

include $(RTE_SDK)/mk/rte.vars.mk

# binary name
APP = dataProcess

# all source are stored in SRCS-y
SRCS-y := main.c crc.c bit.c cbsegm.c debug.c turbocoder.c vector.c tc_interl_umts.c tc_interl_lte.c rm_turbo.c MQAMMod.c phy_common.c sequence.c turbodecoder.c turbodecoder_gen.c turbodecod

LIB += -lm -lpthread -lmkl_rt -msse4.1 -fopenmp
CFLAGS += -g -Wall -I . -msse4.1 -O3 -mavx

CFLAGS += -I$(SRCDIR)
CFLAGS += -O3 $(USER_FLAGS)
CFLAGS += $(WERROR_FLAGS)

include $(RTE_SDK)/mk/rte.extapp.mk
```

图 9: 尝试合并的 makefile



```
/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 10: 编译报错



Makefile - DataProcess-Server- - Visual Studio Code
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 任务(T) 帮助(H)
makefile
Makefile
问题
输出
调试控制台
终端
3: bash
master*
0
0
行 58，列 1
制表符长度: 4
UTF-8
LF
Makefile

```
RTE_SDK ?= $(HOME)/dpdk

# Default target, can be overriden by command line or environment
RTE_TARGET ?= x86_64-native-linuxapp-gcc

include $(RTE_SDK)/mk/rte.vars.mk

# binary name
APP = dataProcess

# all source are stored in SRCS-y
SRCS-y := main.c crc.c bit.c cbsegm.c debug.c turbocoder.c vector.c tc_interl_umts.c tc_interl_lte.c rm_turbo.c MQAMMod.c phy_common.c sequence

EXECENV_LDLIBS += -lm -lpthread -lmkl_rt

CFLAGS += -g -Wall -I . -msse4.1 -O3 -mavx

CFLAGS += -I$(SRCDIR)
CFLAGS += -O3 $(USER_FLAGS)
CFLAGS += $(WERROR_FLAGS)

include $(RTE_SDK)/mk/rte.extapp.mk

```

```
         ^~~
  CC thread_pool.o
/home/sherlockhsu/Github/DataProcess-Server-/DPDK/workspace/dataProcess_send/thread_pool.c: In function ‘thread_run’:
/home/sherlockhsu/Github/DataProcess-Server-/DPDK/workspace/dataProcess_send/thread_pool.c:129:1: warning: label ‘loop’ defined but not used [-Wunus
ed-label]
 loop:
 ^~~~
  LD dataProcess
  INSTALL-APP dataProcess
  INSTALL-MAP dataProcess.map
sherlockhsu@XPS15:~/Github/DataProcess-Server-/DPDK/workspace/dataProcess_send$ 
```

图 11: 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