

Lab2-report

Part1: Question

Q1: What's the purpose of using hugepage?

网络数据包处理中，CPU对内存访问频繁。使用内存大页能减少页表中page table entry的数目，从而有效降低TLB miss，从而降低访存开销。

Q2: Take examples/helloworld as an example, describe the execution flow of DPDK programs?

main函数中，调用 `rte_eal_init`，启动基础运行环境。`RTE_LCORE_FOREACH_SLAVE(lcore_id)` 遍历所有逻辑核，获取其 `lcore_id`。对于每一个从逻辑核，调用 `rte_eal_remote_launch` 启动指定线程。调用 `lcore_hello(NULL)` 启动主逻辑核的线程。调用 `rte_eal_mp_wait_lcore` 等待所有逻辑核运行线程结束。每一个线程调用 `rte_lcore_id` 获取当前逻辑核id，并打印出指定字符串。

Q3: Read the codes of examples/skeleton, describe DPDK APIs related to sending and receiving packets.

- `rte_eth_dev_configure`：对指定端口设置收发队列数目，并可以对端口功能进行配置。
- `rte_pktmbuf_pool_create`：分配一段空间作为内存池，用于存储 `rte_mbuf` 结构体。
- `rte_eth_rx_queue_setup` / `rte_eth_tx_queue_setup`：初始化队列，指定内存、描述符数量、报文缓冲区，并且对队列进行配置。
- `rte_eth_rx_burst` / `rte_eth_tx_burst`：用于收发包，四个参数分别是端口，队列，报文缓冲区以及收发包数。
- `rte_eth_dev_start`：启动端口。
- `rte_eth_promiscuous_enable`：开启混杂模式，使得机器能够接收经过它的所有数据流，无论目标地址是否指向它。

Q4: Describe the data structure of 'rte_mbuf'.

- `rte_mbuf` 的结构报头包含包处理所需的所有数据，大小为2个cache line。对于巨型帧，`rte_mbuf` 包含指向下一个 `rte_mbuf` 结构体的指针来形成链表结构。
- head room用来存储和系统中其他实体交互的信息，如控制信息、帧内容、事件等。数据帧的起始地址保存在 `rte_mbuf` 的 `buff_addr` 中，长度由 `RTE_PKTMBUF_HEADROOM` 定义。在 `rte_mbuf` 的结尾有一段tail room。通过调用 `rte_pktmbuf_prepend` 和 `rte_pktmbuf_append` 可以从head room和tail room中分配一段空间用于扩大数据帧。

- 网络数据帧内容实际长度可通过调用 `rte_pktmbuf_pktlen` 或 `rte_pktmbuf_datalen` 获得。

Part 2: Correctness verification

- 在虚拟机的工作目录下执行 `make`，编译完成后执行 `sudo ./build/sendpkt`。DPDK程序开始连续发包，发包的时间间隔为1秒。
- 在主机上开启wireshark，选择DPDK绑定的虚拟网卡 `vnic1` 进行抓包。
- 检查抓到的包的内容：
 - 抓到包的时间间隔约为1秒
 - Source为我在ipv4 header中指定的源IP地址，即 `10.37.129.2`；Destination为我在ipv4 header中指定的目标IP地址，即 `10.37.129.3`
 - Protocol为UDP
 - Info中显示 `0->0`，即我在udp header中指定的源端口和目标端口；Info中显示Len=12，即我所发送的payload大小
 - 在Data中可以看到我所发送的"hello world"字样
 - IP头的Internet Protocol Version、total length，UDP头的Length都正常显示，没有出现错误警告
- 可知DPDK程序正确发出包

wireshark抓包结果如下图：

eth.addr == 00:1C:42:8E:87:EC

No.	Time	Source	Destination	Protocol	Length	Info
655267	5.202620	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655268	5.202624	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655269	5.202629	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655270	5.202633	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655271	5.202637	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655272	5.202641	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655273	5.202646	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655274	5.202650	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655275	5.202654	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655276	5.202659	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655277	5.202663	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655278	5.202666	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655279	5.202671	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655280	5.202675	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655281	5.202680	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655282	5.202684	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655283	5.202688	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655284	5.202693	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655285	5.202697	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655286	5.202701	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655287	5.202705	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655288	5.202709	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655289	5.202713	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12
655290	5.202717	10.37.129.2	10.37.129.3	UDP	54	0 → 0 Len=12

▶ Frame 655281: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface vnic1, id 0
 ▶ Ethernet II, Src: Parallel_8e:87:ec (00:1c:42:8e:87:ec), Dst: Parallel_8e:87:ec (00:1c:42:8e:87:ec)
 ▶ Internet Protocol Version 4, Src: 10.37.129.2, Dst: 10.37.129.3
 ▶ User Datagram Protocol, Src Port: 0, Dst Port: 0
 ▶ Data (12 bytes)

```

0000  00 1c 42 8e 87 ec 00 1c 42 8e 87 ec 08 00 45 00  ..B....B....E.
0010  00 28 00 00 40 00 00 11 3a c6 0a 25 81 02 0a 25  .(..@.@.:.%..%
0020  81 03 00 00 00 00 00 14 00 00 68 65 6c 6c 6f 20  ..hello
0030  77 6f 72 6c 64 00                                world.
  
```

虚拟机运行DPDK程序输出如下图：

[illegible]