



# DPDK中DMA框架设计、演进

华为

冯成文

# 目录

■ 缘起

■ 设计

■ 演进

# 缘起：


21年6月我们计划新增鲲鹏DMA驱动，经分析发现：

1. 社区没有专门的DMA设备驱动框架，而是在 [rawdev驱动框架](#) 中，另外 [数据面API](#) 已经有分叉（见右侧）
2. rawdev驱动框架虽然API表面统一，实际使用时必须传入具体驱动的结构体（强转为void \*），[导致应用与具体驱动绑定](#)

path: [root/drivers/raw](#)

Mode	Name
d-----	<a href="#">dpaa2_cmdif</a>
d-----	<a href="#">dpaa2_qdma</a>
d-----	<a href="#">ifpga</a>
d-----	<a href="#">ioat</a>
-rw-r--r--	meson.build
d-----	<a href="#">ntb</a>
d-----	<a href="#">octeontx2_dma</a>
d-----	<a href="#">octeontx2_ep</a>
d-----	<a href="#">skeleton</a>

已有3个DMA驱动



From Me★

Subject **[dppk-dev] RFC: Kunpeng DMA driver API design decision**

To Thomas Monjalon <thomas@monjalon.net>★, Ferruh Yigit <ferruh.yigit@intel.com>★

Cc dev@dppk.org <dev@dppk.org>★, nipun.gupta@nxp.com★, hemant.agrawal@nxp.com★, Richardso

Hi all,

We prepare support Kunpeng DMA engine under rawdev framework, and observed that there are two different implementations of the data plane API:

1. rte\_rawdev\_enqueue/dequeue\_buffers which was implemented by dpaa2\_qdma and octeontx2\_dma driver.
2. rte\_ioat\_enqueue xxx/rte\_ioat\_completed\_ops which was implemented by ioat driver.

Due to following consideration (mainly performance), we plan to implement API like ioat (not the same, have some differences) in data plane:

1. The `rte_rawdev_enqueue_buffers` use opaque buffer reference which is vendor's specific, so it needs first to translate application parameters to opaque pointer, and then driver writes the opaque data onto hardware, this may lead to performance problem.
2. `rte_rawdev_xxx` doesn't provide memory barrier API which may need to extend by opaque data (e.g. add flag to every request), this may introduce some complexity.

Also the example/ioat was used to compare DMA and CPU-memcopy performance, Could we generalized it so that it supports multiple-vendor ?

I don't know if the community accepts this kind of implementation, so if you have any comments, please provide feedback.

Best Regards.

# 结“果”：

From Thomas Monjalon <thomas@monjalon.net> ★  
Subject Re: [dpdk-dev] [PATCH v26 0/6] support dmadev  
To Me ★  
Cc ferruh.yigit@intel.com ★, bruce.richardson@intel.com ★, jerinj@marvell.com ★

13/10/2021 14:24, Chengwen Feng:

This patch set contains six patch for new add dmadev.

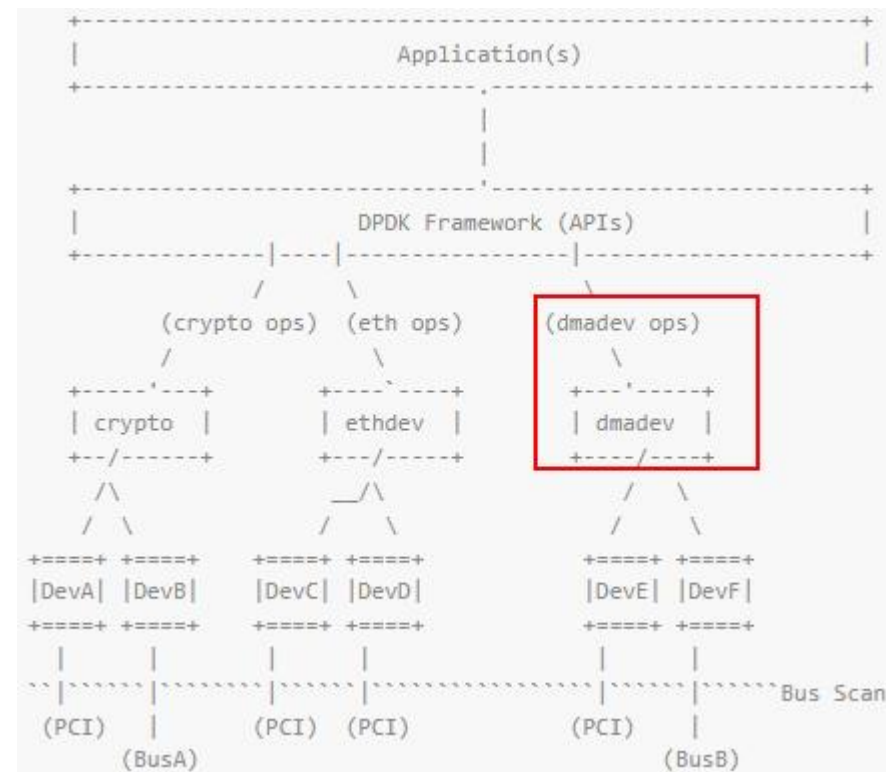
Chengwen Feng (6):

- dmadev: introduce DMA device library
- dmadev: add control plane API support
- dmadev: add data plane API support
- dmadev: add multi-process support
- dma/skeleton: introduce skeleton dmadev driver
- app/test: add dmadev API test

Applied, thanks for the big work.

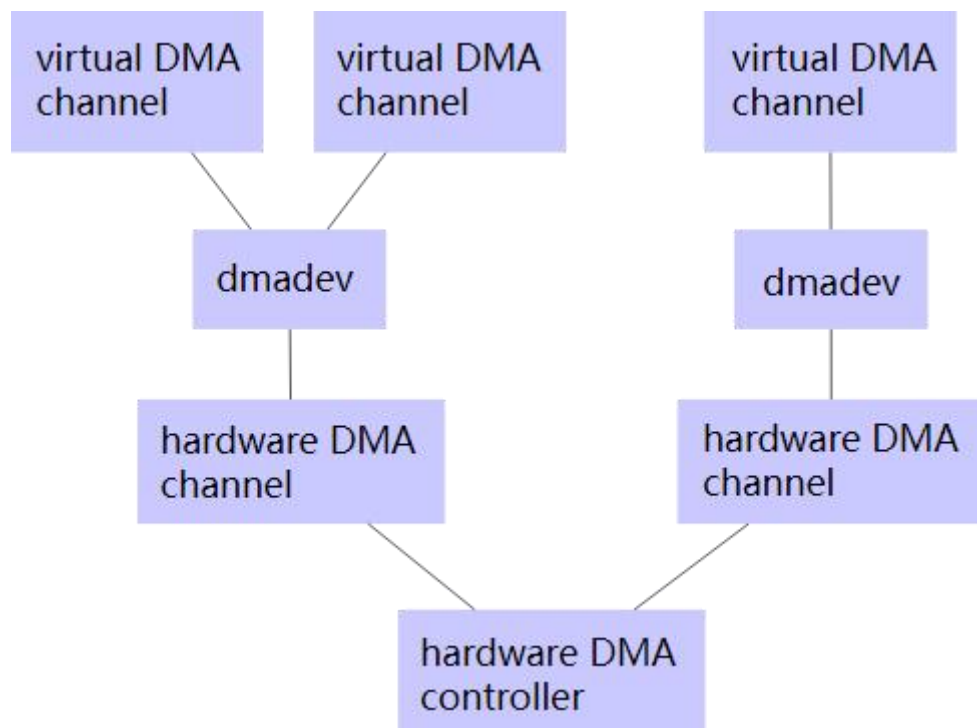
rte\_dmadev.h:

```
/* SPDX-License-Identifier: BSD-3-Clause  
 * Copyright(c) 2021 HiSilicon Limited  
 * Copyright(c) 2021 Intel Corporation  
 * Copyright(c) 2021 Marvell International Ltd  
 * Copyright(c) 2021 SmartShare Systems  
 */
```



1. 历时4月+, 经过大量与社区交流, 社区于21年10月18日终于接收了v26版本
2. 该库开发也得到众多厂家支持, 如上图中对外API文件 (rte\_dmadev.h)

# 设计-设备模型:



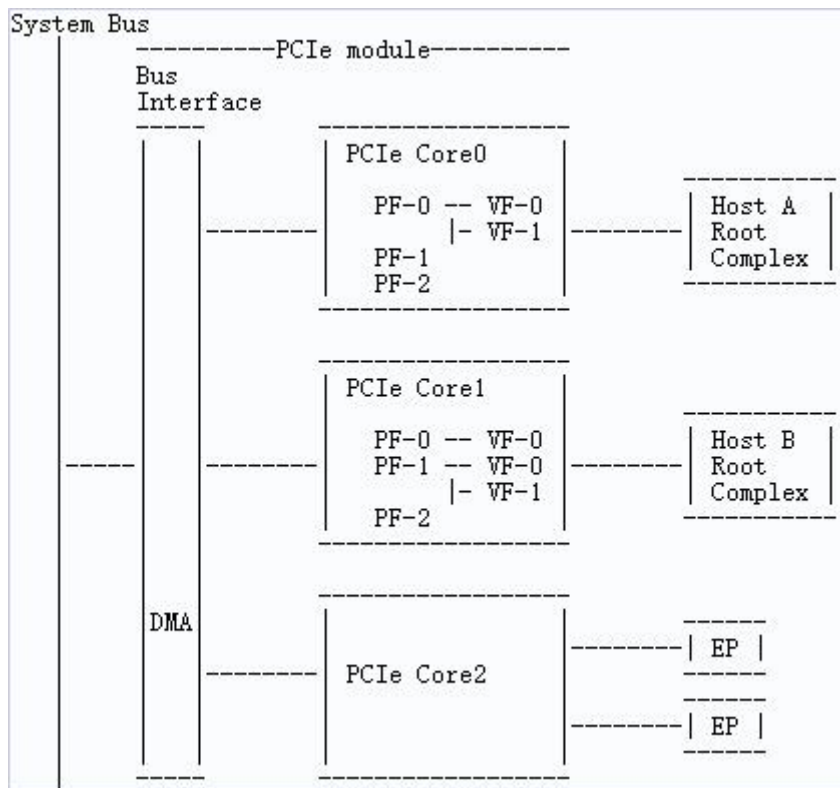
## Device Control Plane API:

1. Device Discovery/Shutdown
2. Device Capability Reporting
3. Device/Channel Configuration
4. Device Status Query & Dump
5. Device Error Handling

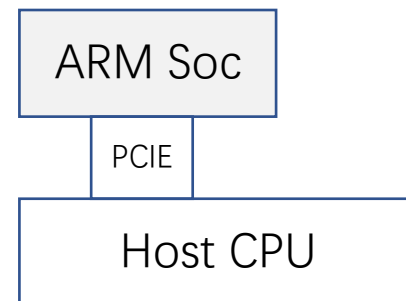
## Device Data Plane API:

1. Copy/Copy\_SG/Fill/Submit
2. Completed(\_status)

# 设计-DMA DIR:



Example:



1. DIR\_MEM\_TO\_MEM: From ARM local memory to local memory
2. DIR\_MEM\_TO\_DEV: From ARM local memory to host
3. DIR\_DEV\_TO\_MEM: From host to ARM local memory
4. DIR\_DEV\_TO\_DEV: From host 1 to host 2

涉及DEV的访问需配置通路参数信息: `rte_dma_port_param`

# 设计-性能:

## 1. 提交任务:

- a) 不提供burst接口, 转而提供单个copy/copy\_sg/fill等接口
- b) 支持批量submit: do多次copy/copy\_sg/fill等操作后, 调用submit一次提交
- c) 支持快速submit: copy/copy\_sg/fill等操作支持携带submit标记快速提交, 避免再次调用submit来提交

## 2. 获取完成:

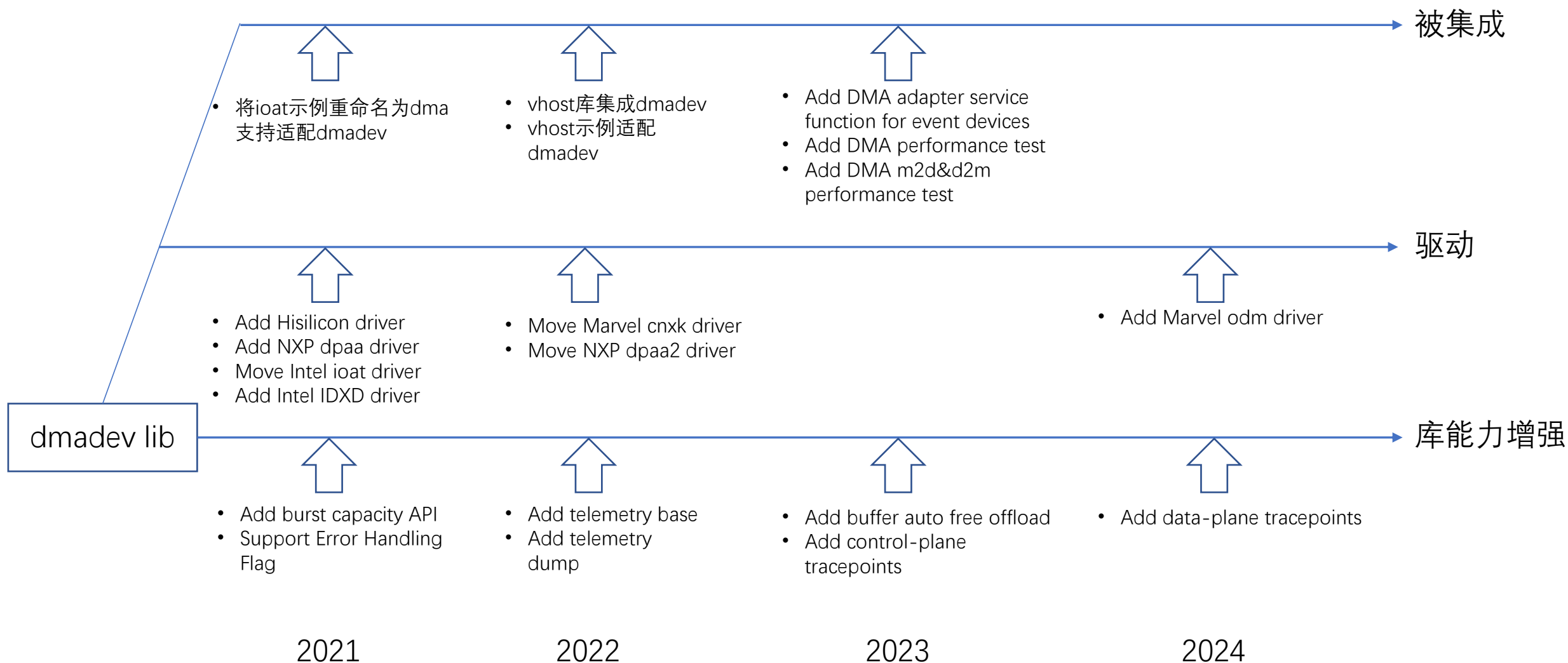
- a) 支持ring\_idx (16bit), 应用根据该ring\_idx和请求的ring\_idx可计算出哪些请求已完成

## 3. 其它:

- a) 支持RTE\_DMA\_OP\_FLAG\_LLC, 标记是否将数据写入low level cache



# 演进 (merged) :



注：很多创新点是由ARM厂家引领的



