**VirtIO框架**

**1 概要**

Vela基于OpenAMP实现了完整的Virtio框架

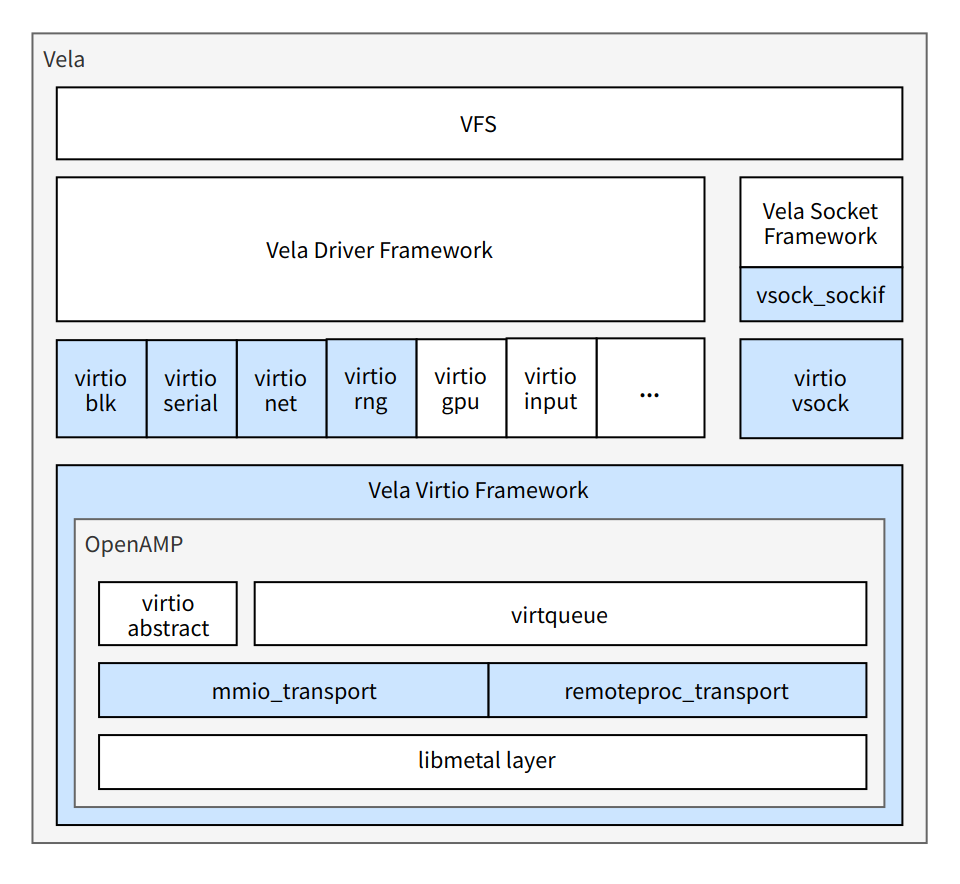
**2 缩写**

**MMIO**：Simple memory mapped device (“virtio-mmio”). Virtual environments without PCI support (a common situation in embedded devices models) might use

**Remoteproc**：Remote Processor

**3 框架**

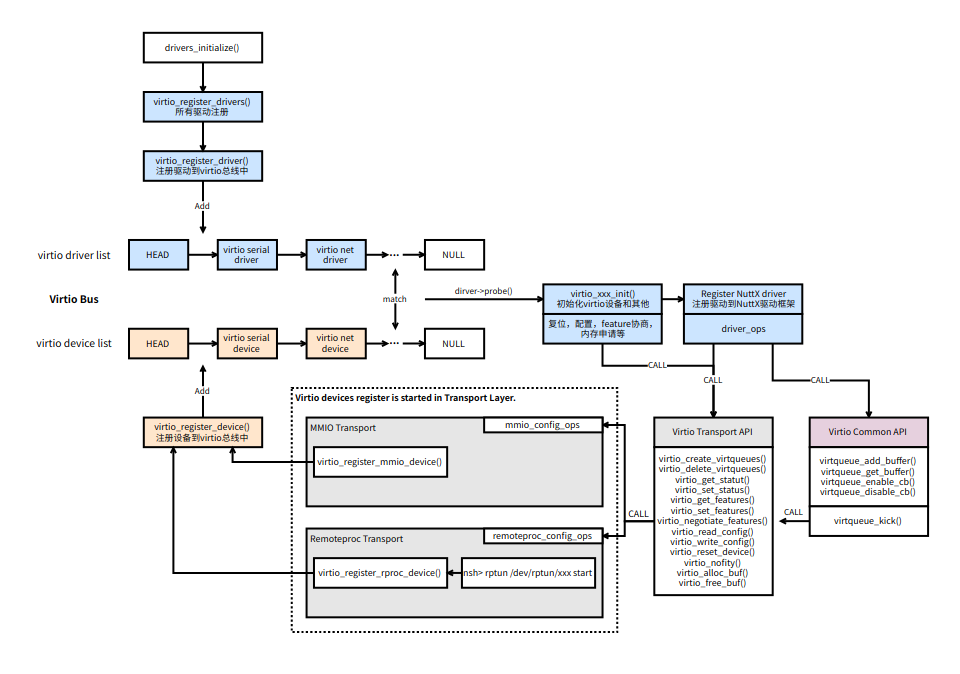
**3.1 框架图**



上图为Vela Virtio框架图，大致可以分为三部分：

1. 驱动层：将virtio和Vela驱动框架进行对接，驱动层调用virtio提供的统一接口进行设备的初始化和数据的交互；
2. Virtio层：提供统一的接口给Driver，实现Driver/Device的注册，卸载和匹配机制；
3. 传输层：提供不同传输层的支持；

**3.2 流程图**



上图显示了virtio device和virtio driver的匹配流程图和调用关系

1. Driver注册：Vela初始化时调用virtio\_register\_drivers()注册所有已经支持的（且已配置的）virtio driver到virtio总线中；
2. Device注册：由传输层发起，在初始化或者运行过程中，调用virtio\_register\_device()注册virtio device到virtio总线中；
3. 匹配：总线在注册设备时会进行匹配，如果Driver和Device匹配上，执行Driver实现的probe函数，probe函数中，驱动对virtio device进行初始化，配置，featrue协商等操作，根据设备的复杂度和类型的不同，有时还需要初始化自身的一些私有结构或者调用Vela接口进行一些额外的操作；
4. 注册Vela驱动：调用Vela驱动框架提供的API将驱动注册到VFS中；
5. 运行：运行过程中，Driver会通过调用OpenAMP提供的virtqueue的通用接口，按照Virtio的标准和Device进行数据的交换和相互通知，实现驱动的功能；

**4 API说明**

这里对virtio驱动适配过程中需要调用到的接口进行说明，可以参照上面的Virtio Serial举例说明进行理解

**4.1 Vela Log接口**

|  |
| --- |
| C vrtinfo(...) |

描述：INFO级别的virtio syslog接口

|  |
| --- |
| C vrtwarn(...) |

描述：WARNING级别的virtio syslog接口

|  |
| --- |
| C vrterr(...) |

描述：ERROR级别的virtio syslog接口

**4.2 Vela Virtio框架接口**

* int virtio\_register\_driver(FAR struct virtio\_driver \*driver)

描述：注册一个virtio driver到virtio总线，当总线中已经有对应的设备时，会立即匹配上并调用驱动实现的probe函数，如果总线中没有对应的设备，则直到有对应的virtio设备注册到virtio总线上，回调驱动的probe进行驱动初始化

**4.3 OpenAmp接口**

**4.3.1 前置知识**

Driver Tx virtqueue：驱动的发送队列，从txvq的used ring中获取buffer，**填充需要发送的数据后**，再添加到txvq的avail ring中，是发送数据的流程

Driver Rx virtqueue：驱动的接收队列，从rxvq的used ring中获取buffer，**获取其中的数据后**，再返回给rxvq的avail ring中，是接收数据的流程

**4.3.2 接口说明**

|  |
| --- |
| C void \*virtqueue\_get\_buffer(struct virtqueue \*vq, uint32\_t \*len, uint16\_t \*idx); |

描述：从vq的used ring中获取一个buffer

参数：

vq- virtqueue

len- 获取的buffer的长度

idx- 获取的buffer在used ring中的索引

|  |
| --- |
| C int virtqueue\_add\_buffer(struct virtqueue \*vq, struct virtqueue\_buf \*buf\_list, int readable, int writable, void \*cookie) |

描述：添加一个buffer到vq的avail ring中

参数：

vq- virtqueue

buf\_list- 需要添加的buf数组

readable- buf\_list中读buffer的个数，表示希望device读取的；

writable- buf\_list中写buffer的个数，表示希望device填充的；

cookie - 缓存，会在获取buffervirtqueue\_get\_buffer时返回；

|  |
| --- |
| C void virtqueue\_kick(struct virtqueue \*vq) |

描述：通知device，一般在发送了数据到device端后或者是返回了buffer给device端后通知对方可以进行下一步操作了；

参数：

vq - virtqueue

|  |
| --- |
| C virtqueue\_enable\_cb(struct virtqueue \*vq) virtqueue\_disable\_cb(struct virtqueue \*vq) |

描述：使能或关闭是virtqueue中断

参数：

vq - virtqueue

**5 Driver适配说明**

在Vela中基于OpenAmp实现VirtIO框架后，需要各方基于该套框架实现各类设备的Virtio Driver，这里主要针对VirtIO标准中定义好的VirtIO Driver的适配进行说明，描述如何使用OpenAMP/Vela VirtIO框架的接口来进行驱动的初始化、数据传输等操作。

下面的文档以Virtio-RNG和Virtio-Serial举例，说明了如何基于Vela Virtio框架实现一个Virtio Driver：

[VirtIO Rng](https://xiaomi.f.mioffice.cn/wiki/wikk4EcCZMCWffVZbzs65YymWSc) 和 [VirtIO Serial](https://xiaomi.f.mioffice.cn/wiki/wikk4ideCKKtWZI6p1qnaSSvenf)

**参考文档**

1. https://ozlabs.org/~rusty/virtio-spec/virtio-paper.pdf
2. https://docs.oasis-open.org/virtio/virtio/v1.2/csd01/virtio-v1.2-csd01.pdf
3. [OpenAMP.pptx](https://xiaomi.f.mioffice.cn/wiki/wikk4qeQOlrTjm9OiGwXx3b7xG5)
4. [VirtIO/QEMU介绍](https://xiaomi.f.mioffice.cn/wiki/wikk4r8ZUa03fYh4zPMGYDCf1ep)