



## OVS 总体架构、源码结构及数据流程全面解析

本文首发于我的公众号 **Linux云计算网络 (id: cloud\_dev)**，专注于干货分享，号内有 **10T** 书籍和视频资源，后台回复「**目录**」，欢迎大家关注，二维码文末可以扫。

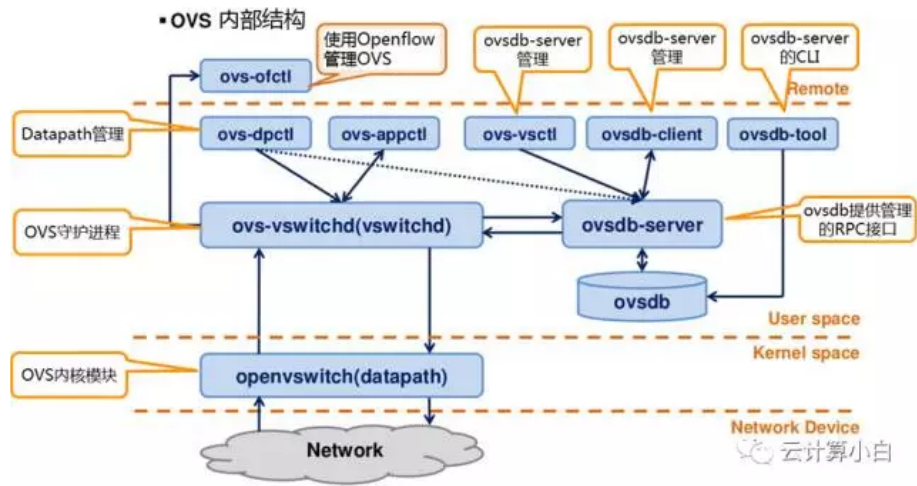
CONTENTS ×

1. OVS 架构
2. OVS 源码结构
3. 数据转发流程
4. 总结

在前文「[从 Bridge 到 OVS](#)」中，我们已经对 OVS 进行了一番探索。本文决定从 OVS 的整体架构到各个组件都进行一个详细的讲解。

### OVS 架构

OVS 是产品级的虚拟交换机，大量应用在生产环境中，支撑整个数据中心虚拟网络的运转。OVS 基于 SDN 的思想，将整个核心架构分为控制面和数据面，数据面负责数据的交换工作，控制面实现交换策略，指导数据面工作。



从整体上看，OVS 可以划分为三大块，管理面、数据面和控制面。

数据面就是以用户态的 ovs-vswitchd 和内核态的 datapath 为主的转发模块，以及与之相关联的数据库模块 ovsdb-server，控制面主要是由 ovs-ofctl 模块负责，基于 OpenFlow 协议与数据面进行交互。而管理面则是由 OVS 提供的各种工具来负责，这些工具的提供也是为了方便用户对底层各个模块的控制管理，提高用户体验。下面就对这些工具进行一个逐一的阐述。

**ovs-ofctl**：这个是控制面的模块，但本质上它也是一个管理工具，主要是基于 OpenFlow 协议对 OpenFlow 交换机进行监控和管理，通过它可以显示一个 OpenFlow 交换机的当前状态，包括功能、配置和表中的项。使用时，有很多参数，我们可以通过 `ovs-ofctl --help` 查看。

Copy

常用命令：

`ovs-ofctl show switch-name`：输出交换机信息，包括其流量表和端口信息。

`ovs-ofctl dump-ports switch-name`：输出交换机的端口统计信息，包括收发包、丢包、错误包等数量。

`ovs-ofctl add-flow switch-name`：为交换机配置流策略。

**ovs-dpctl**：用来配置交换机的内核模块 datapath，它可以创建，修改和删除 datapath，一般，单个机器上的 datapath 有 256 条（0-255）。一条 datapath 对应一个虚拟网络设备。该工具还可以统计每条 datapath 上的设备通过的流量，打印流的信息等，更过参数通过 `ovs-dpctl --help` 查看。

Copy

常用命令：

`ovs-dpctl show`：显示所有 datapath 的基本信息。

`ovs-dpctl dump-dps`：显示所有 datapath 的名字。

`ovs-dpctl dump-flows DP`：显示一条 datapath DP 上的流信息。

**ovs-appctl**：查询和控制运行中的 OVS 守护进程，包括 ovs-switchd，datapath，OpenFlow 控制器等，兼具 ovs-ofctl、ovs-dpctl 的功能，是一个非常强大的命令。ovs-vswitchd 等进程启动之后就以一个守护进程的形式运行，为了能够很好的让用户控制这些进程，就有了这个命令。详细可以 `ovs-appctl --help` 查看。



常用命令：

ovs-vsctl show：显示主机上已有的网桥及端口信息。

ovs-vsctl add-br br0：添加网桥 br0。

**ovsdb-client**：访问 ovsdb-server 的客户端程序，通过 ovsdb-server 执行一些数据库操作。

常用命令：

ovsdb-client dump：用来查看ovsdb内容。

ovsdb-client transact：用来执行一条类 sql。

**ovsdb-tool**：和 ovsdb-client 要借助 ovsdb-server 才能进行相关数据库操作不同，ovsdb-tool 可以直接操作数据库。

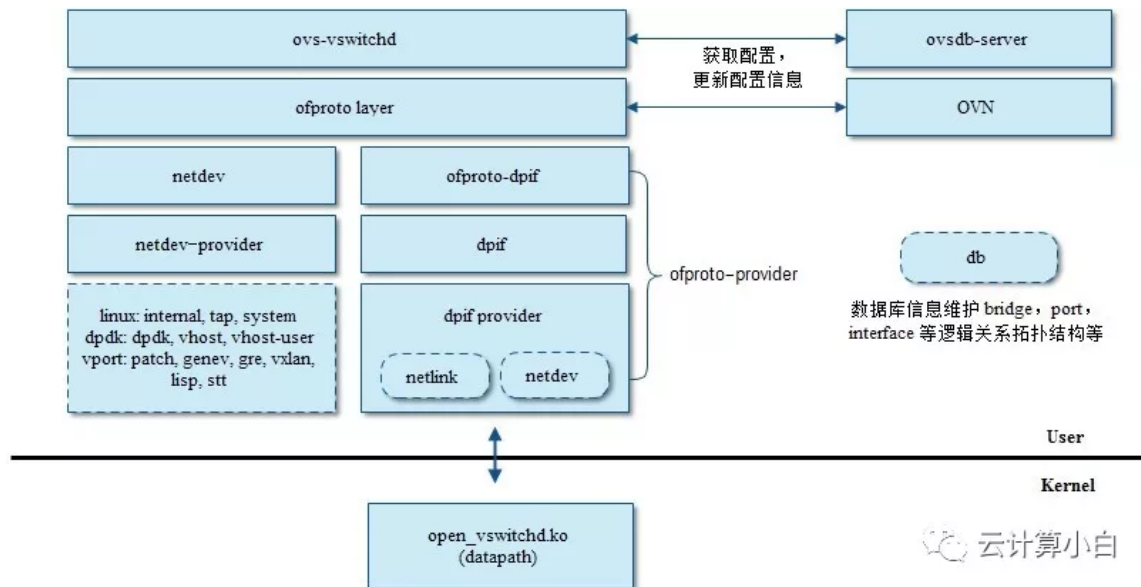
Copy

## CONTENTS

1. OVS 架构
2. OVS 源码结构
3. 数据转发流程
4. 总结

## OVS 源码结构

OVS 源码结构中，主要包含以下几个主要的模块，数据交换逻辑在 vswitchd 和 datapath 中实现，vswitchd 是最核心的模块，OpenFlow 的相关逻辑都在 vswitchd 中实现，datapath 则不是必须的模块。ovsdb 用于存储 vswitch 本身的配置信息，如端口、拓扑、规则等。控制面部分采用的是 OVS 自家实现的 OVN，和其他控制器相比，OVN 对 OVS 和 OpenStack 有更好的兼容性和性能。



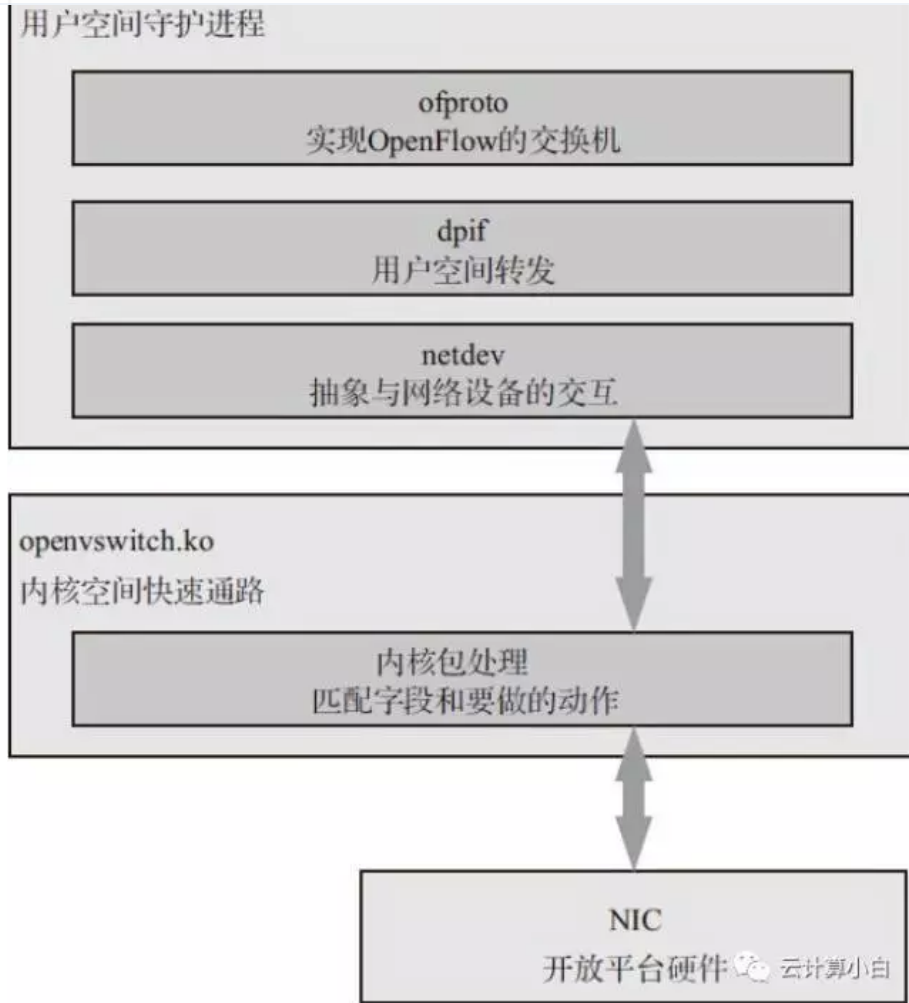
从图中可以看出 OVS 的分层结构，最上层 vswitchd 主要与 ovsdb 通信，做配置下发和更新等，中间层是 ofproto，用于和 OpenFlow 控制器通信，并基于下层的 ofproto provider 提供的接口，完成具体的设备操作和流表操作等工作。

dpif 层实现对流表的操作。

netdev 层实现了对网络设备（如 Ethernet）的抽象，基于 netdev provider 接口实现多种不同平台的设备，如 Linux 内核的 system, tap, internal 等，dpdk 系的 vhost, vhost-user 等，以及隧道相关的 gre, vxlan 等。

## 数据转发流程

通过一个例子来看看 OVS 中数据包是如何进行转发的。



☰ CONTENTS ×

1. OVS 架构

2. OVS 源码结构

3. 数据转发流程

4. 总结

- 1) ovs 的 datapath 接收到从 ovs 连接的某个网络端口发来的数据包，从数据包中提取源/目的 IP、源/目的 MAC、端口等信息。
- 2) ovs 在内核态查看流表结构（通过 hash），如果命中，则快速转发。
- 3) 如果没有命中，内核态不知道如何处置这个数据包，所以，通过 netlink upcall 机制从内核态通知用户态，发送给 ovs-vswitchd 组件处理。
- 4) ovs-vswitchd 查询用户态精确流表和模糊流表，如果还不命中，在 SDN 控制器接入的情况下，经过 OpenFlow 协议，通告给控制器，由控制器处理。
- 5) 如果模糊命中， ovs-vswitchd 会同时刷新用户态精确流表和内核态精确流表，如果精确命中，则只更新内核态流表。
- 6) 刷新后，重新把该数据包注入给内核态 datapath 模块处理。
- 7) datapath 重新发起选路，查询内核流表，匹配；报文转发，结束。

## 总结

OVS 为了方便用户操作，提供了很多管理工具，我们平常在使用过程中只需记住每个工具的作用，具体的命令可以使用 -h 或 --help 查看。

我的公众号「Linux云计算网络」(id: cloud\_dev)，号内有 10T 书籍和视频资源，后台回复「1024」即可领取，分享的内容包括但不限于 Linux、网络、云计算虚拟化、容器 Docker、OpenStack、Kubernetes、工具、SDN、OVS、DPDK、Go、Python、C/C++ 编程技术等内 容，欢迎大家关注。





云计算 | 网络 | Linux | 干货

获取学习大礼包后台  
回复“1024”

加群交流后台回复“加群”



≡ CONTENTS ×

1. OVS 架构
2. OVS 源码结构
3. 数据转发流程
4. 总结

作者：公众号「Linux云计算网络」，专注于Linux、云计算、网络领域技术干货分享

出处：<https://www.cnblogs.com/bakari/p/8097478.html>

本站使用「署名 4.0 国际」创作共享协议，转载请在文章明显位置注明作者及出处。

分类：云计算, 虚拟化

标签：虚拟化, 云计算, OVS

推荐 1

赞赏

收藏

反对 0

« 上一篇：从 Bridge 到 OVS，探索虚拟交换机

» 下一篇：KVM 初探

posted @ 2017-12-24 21:33 CloudDeveloper 阅读(7874) 评论(0) 编辑 收藏

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

