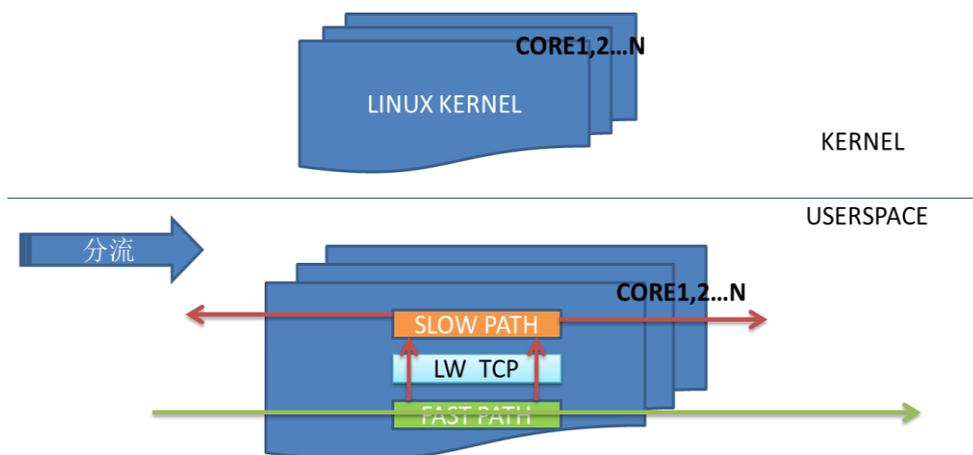


### 1.1.1 设计思路

BOS 产品使用多核 CPU 平台，多核分为 3 组，0 核为 1 组运行控制层面业务，1-M 核运行分流器，M-N 核运行数据业务程序。（单核情况下，不启用分流器，数据与控制业务共用 1 个核心）。设计的核心思想为：分流+全并行业务处理+主业务资源隔离。



#### 数据业务

采用 ZOL 用户进程方式，全并行。每个数据业务核心，既跑高性能模式,又跑代理模式,数据流根据配置走到相应流程. 由于有分流器的支撑，可以大大减少竞争访问。

转发业务程序根据业务区分高性能模式与代理模式，高性能模式采用 4 层转发+NAT 处理 L4 负载均衡。代理模式处理 L7 负载均衡业务，通过高性能 TCP 协议栈代理业务，业务连接到设备终结。

L4 通过 NAT 实现，性能可以保证。L7 通过轻量级 TCP 协议栈（零拷贝，无任务切换)支撑的 TCP 代理实现。

#### 高性能模式

处理业务：LAYER4 负载均衡

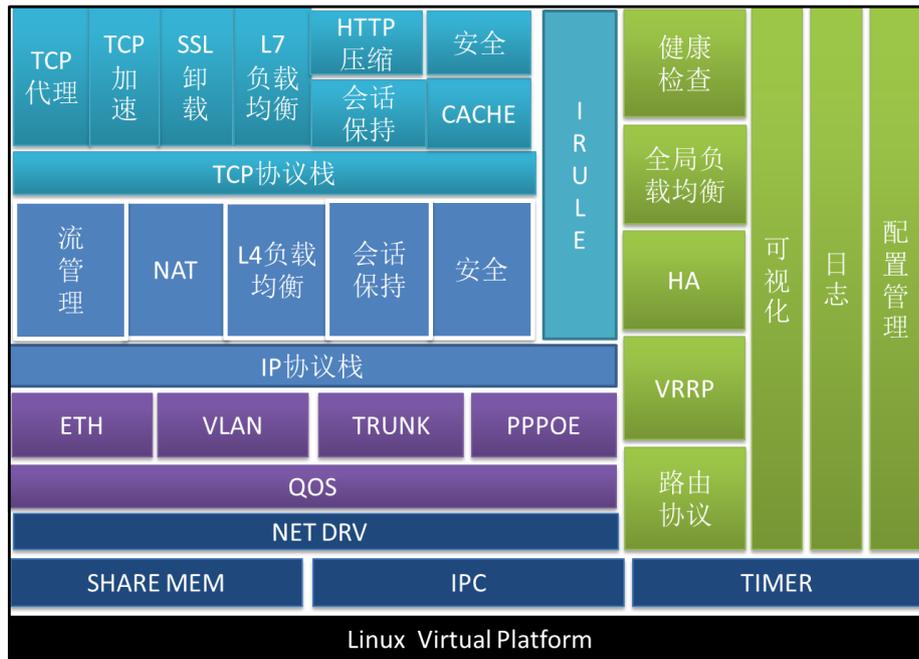
实现方式：4 层转发+NAT +LB

#### 代理模式

处理业务：LAYER7 全代理，包括 SSL 卸载，CACHE 等。

实现方式：高性能 TCP 协议栈+ TCP PROXY+HTTP 协议处理+LB

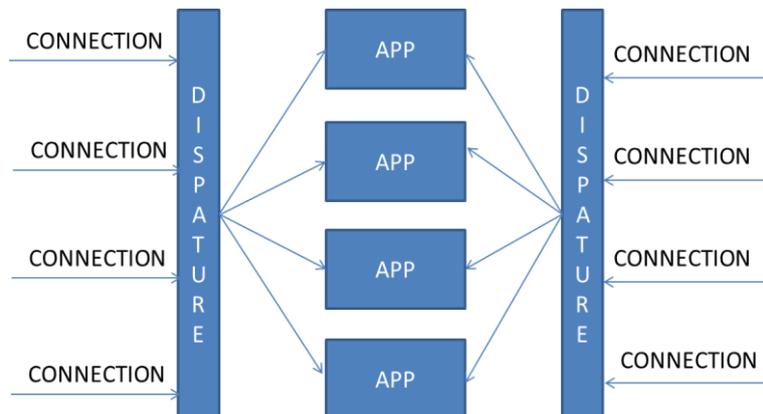
### 1.1.2 系统框图



### 1.1.3 关键技术

#### 分流器(DISPATURE)

分流器把流量均衡的分配到各个核心，可降低各核之间的竞争访问，提高性能。并且可降低共享数据并发访问处理的复杂度。



分流器主要通过报文五元组 HASH 来分配报文。

#### TCP代理

Tcp 协议栈为应用代理提供标准 TCP 协议服务。通过与应用层之间的零拷贝与零任务调度机制达到高性能的目标。设计要点如下：

- 多线程方式和回调方式同时提供, 采用多线程方式。
- 协议栈、应用程序、代理之间共享内存, 以实现报文零拷贝。
- 协议栈与应用程序之间数据通信支持非连续数据和连续数据。
- 一个 Core 中存在一个协议栈线程和一个或多个应用程序。

其基本框架如下图:

