# Hyperscan的模式选择

原创 DPDK开源社区 2017-03-27

作者 徐迟


在上一篇文章中我们对Hyperscan的整体结构做了初步的介绍。我们知道Hyperscan的调用分为两个阶段：编译期和运行期。在使用前，用户需要提前为运行期选择一个合适的运行模式，并在编译期就将此参数传入Hyperscan。该运行模式的选择取决于用户的使用场景，最常用的两种分别为块模式与流模式。本篇我们将对这两种模式做具体的介绍。

## 1 块模式(Block Mode)

块模式是Hyperscan中基础的模式，调用函数为hs_scan()。用户每次调用时，将对一段完整的数据块进行匹配。匹配只限于该数据块内，而与上一次的hs_scan() 调用无关。在通常情况下块模式是所有模式中最高效的。

### 使用场景：

1. 用户只有一段完整的数据需要扫描，而没有更多关联的上下文时，推荐使用块模式。如对本地的数据库文本进行扫描；
2. 若用户的数据为多个独立的数据包，并希望在各个数据包内寻找匹配时，更适合使用块模式；此时Hyperscan会返回相对于每个数据包头的匹配位置。

## 2 流模式(Streaming Mode)

在真实网络场景下，数据被拆分成多个报文发送，在只接收到部分数据流的情况下使用块模式匹配会导致跨数据流的匹配点被遗漏，可行的方法只有等全部数据流接收完成后再统一进行匹配，此举会增加内存的开销及报文处理的复杂度。

由此，在Hyperscan中我们引入流模式。通过额外的流内存对流匹配信息进行记录，保证在丢弃了过去所有流数据的情况下匹配过程仍然能够正确执行。

### 流内存(Stream State)：

流内存是一段与Hyperscan数据库关联的缓存，它的长度在编译时确定。对于一个编译出的数据库，与它关联的所有流内存都需要各自分配独立的空间。流内存里存储着所有与流的边界相关的状态信

息。

### 流模式的接口：

流模式提供了众多的接口，一次简单的流模式调用就包括了打开(hs_open_stream())，扫描(hs_scan_stream())和关闭(hs_close_stream())三个操作。所有扫描中找到的匹配信息都将交给用户自定义的回调函数进行处理。

除此以外，Hyperscan还提供了对流信息的重置(hs_reset_stream())与拷贝(hs_copy_stream())功能，减少空间开销并提高了复用性。

### 使用场景：

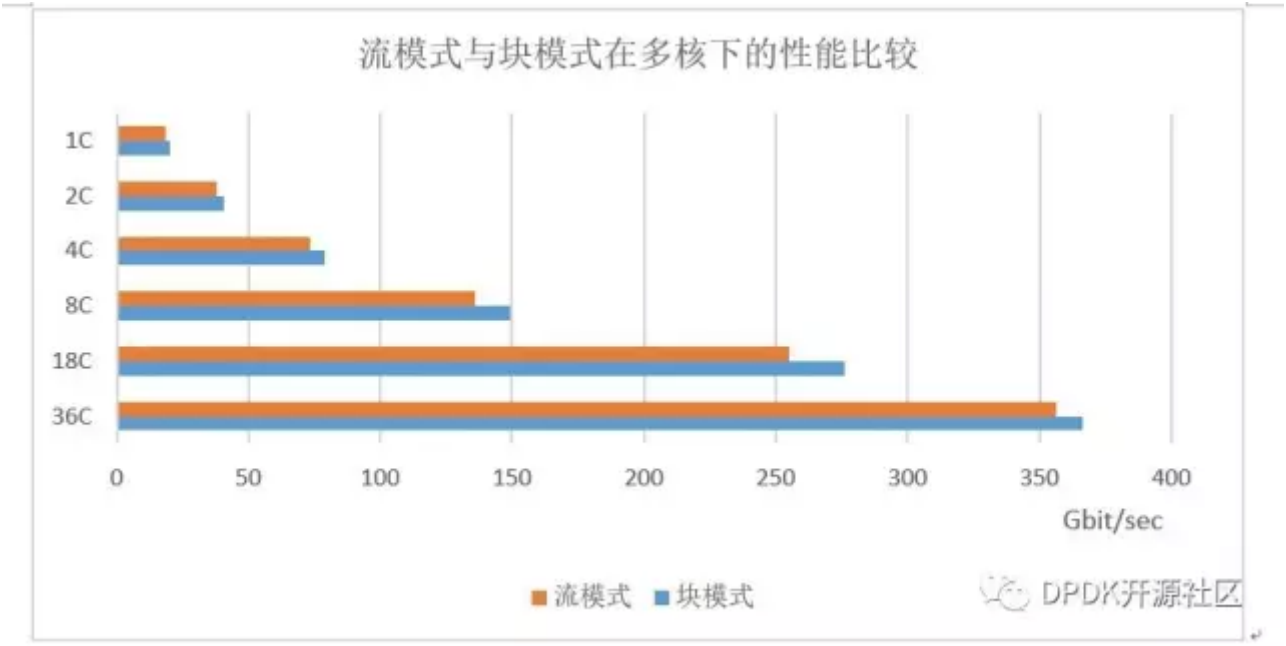
1. 如在网络防火墙或报文检测的应用场景下，数据流是分段接收的，则适用流模式进行匹配；
2. 对一组存储在本地但不连续的数据，并且做memcpy()合并数据开销过大时，可以使用流模式（详见向量模式）。

## 3 向量模式(Vectored Mode)

向量模式是Hyperscan为了特定的用户需求而支持的第三种扫描模式。它和块模式的区别在于: 当用户已经准备好了所有的输入数据，但数据在内存中并不连续，而是分散为多个块时，可以调用向量模式的扫描函数(hs_scan_vector())进行匹配。向量模式在内部实现上将字符串数组当作若干段数据流，依次调用流模式完成匹配工作。其性能一般情况下不会优于块模式，通常不做推荐。

## 4 流模式与块模式的性能比较

我们基于防火墙厂商的真实规则，在Intel(R) Xeon(R) CPU E5-2699 v3 @2.30GHz环境下，对IPS真实网络流量进行测试。以下数据分别是Hyperscan在单核和多核运行时，流模式与块模式的性能对比，我们同时使用了70条规则进行匹配：




可以看出，块模式在相同情况下比流模式在性能上有着约8%的提升。

## 5 模式选择的建议

1. 轻度需求请使用块模式:

在不必要时使用流模式可能会使Hyperscan达不到最优化，因此降低扫描速度；

若一组规则集中同时包含需要流模式匹配的规则和块模式匹配的规则，除非需要流模式的规则占了绝大多数，否则我们建议将其分解为两个子规则集分开编译。

2. 在流模式下具有重复特征(Bounded Repeats)的规则会使性能大幅下降：

像规则 /X.{1000,1001}abcd/中的重复特征 /.{1000,1001}/，在流模式下会对性能产生较大影响。它需要在流内存中用近千个比特位记录X出现的位置以防X与abcd中间跨越了若干条流。

若在一个规则的匹配中，如果规则中一部分特征指向是极其具体的，已经能很有效地匹配到相应的文本，且重复特征部分对准确率的提升不明显，则用户可以考虑精简规则换取性能的提升。例如在一个病毒规则/mz.{100}abcd/ 中，病毒特征abcd已经可以完成高效匹配，那么舍弃它的通用可执行前缀mz以及重复特征.{100}可以带来接近一倍的匹配速度提升。

**总结：**

块模式与流模式作为Hyperscan的两种重要模式各有不同的使用场景。根据自己的需求选择合适的匹配模式，可以使Hyperscan的算法效率达到最优。

作者介绍

徐迟

英特尔软件工程师，负责Hyperscan研发。主要研究领域包括自动机与正则表达式匹配等。

感谢王翔的建议和修改

“DPDK开源社区”精选文章

欢迎搭乘Hyperscan号极速列车~

Hyperscan Release 4.4.0

DPDK Release 17.02

无锁队列详细分解 — 顶层设计

从计算机架构师的角度看DPDK性能

基于virtio-user的新exception path方案

玩物志 | 什么！DPDK在盒子里？

关于DPDK Cryptodev，你不得不明白的几点！