Hyperscan中正则规则的逻辑组合

Chang Harry DPDK与SPDK开源社区 2018-09-21

Hyperscan中正则规则的逻辑组合

Hyperscan中正则规则的逻辑组合

在Hyperscan5.0之前的版本中，只支持对正则表达式的匹配。在用户的使用场景中，有时需要在一组正则规则中根据匹配结果的逻辑组合来决定后续行为，比如需要某几条正则规则全部匹配或部分匹配，或要求某些正则规则不匹配，这时需要用户自己记录匹配结果并进行逻辑计算。

Hyperscan 5.0 提供了这样一个新的特性：正则规则间的逻辑组合，用户可以直接定义逻辑组合表式，由Hyperscan替用户进行正则规则匹配结果的逻辑运算并直接报告结果。

01

正则规则间逻辑组合的定义

当用户需要基于在一组正则规则中的部分规则有无匹配的结果来决定后续行为时，新版本的Hyperscan便提供了在给定正则规则集里自定义子规则匹配结果逻辑组合的功能，目前包含“与(AND)”“或(OR)”“非(NOT)”三种逻辑运算。

一个逻辑组合的值取决于其中每个子规则在当前位置的匹配状态。子规则的匹配状态由一个布尔值来描述：如果直到当前位置该规则尚未产生匹配即为FALSE，反之如果在当前位置或之前已经产生过匹配即为TRUE。

特别的，对于“非”运算的值而言，是对它所作用的规则在当前位置匹配状态值取反，例如，NOT 101的意义为规则101直到当前位置尚未产生匹配。

一个逻辑组合以表达式的形式在编译期传入Hyperscan。逻辑组合表达式会在其子规则产生匹配且逻辑表达式值为TRUE的每个位置上产生匹配。

我们使用下面的逻辑组合表达式来举例说明：

((301 OR 302) AND 303) AND (304 OR NOT 305)

假设子规则301在位置10产生匹配，则301的匹配状态逻辑值为TRUE，同时其余子规则的逻辑值为FALSE，因此，整个逻辑组合的值在该位置上为FALSE。

然后，假设子规则303在位置20产生匹配。此刻子模式301和303的逻辑值为TRUE，同时其余子模式的逻辑值仍为FALSE。这样整个逻辑组合的值为TRUE，该因此逻辑组合表达式在位置20会产生一个匹配。

最后，假设子规则305在位置30产生匹配。则此时子模式301，303和305的逻辑值为TRUE，同时其余子模式的逻辑值仍为FALSE。这样整个逻辑组合的值为FALSE，对逻辑组合表达式而言没有匹配产生。

02

逻辑组合的使用方法

Hyperscan中使用逻辑组合的语法，是一个由操作数、操作符和小括号组成的逻辑运算中缀表达式。其中操作数是子规则的ID，操作符为”!”(NOT)，” &”(AND)和”|”(OR)。比如上例中的逻辑组合就用如下表达式来表示：

((301 | 302) & 303) & (304 | !305)

逻辑组合表达式有下列性质：

1. 操作符的优先级为！> & > |。例如：
   A&B|C 将处理为 (A&B)|C，
   A|B&C 将处理为 A|(B&C)，
   A&!B 将处理为 A&(!B)。
2. 允许使用多余的小括号。例如：
   (A)&!(B) 等同于 A&!B，
   (A&B)|C 等同于 A&B|C。
3. 允许使用空白字符，并自动忽略。

使用逻辑组合表达式时，需定义其ID，传入Hyperscan的编译API函数hs_compile_multi()或hs_compile_ext_multi()，同时为其设置HS_FLAG_COMBINATION标志，指明这是一个逻辑组合表达式而非普通正则。一个逻辑组合表达式必须与其中提到的所有子规则作为统一集合一起编译。

当一个表达式设置了HS_FLAG_COMBINATION标志时，除HS_FLAG_SINGLEMATCH和HS_FLAG_QUIET之外的其他标志均会被忽略。

目前Hyperscan在编译期拒绝当没有子规则产生匹配时逻辑值为TRUE的逻辑组合，例如：

```
!101
!101|102
!101&!102
!(101&102)
```

作为逻辑组合中操作数的子规则（如前面例子中的301至305），可以设置HS_FLAG_QUIET标志使之对自身产生的匹配保持沉默。当没有设置HS_FLAG_QUIET标志时，所有产生的匹配（包括子规则自身产生的匹配和逻辑组合产生的匹配）都将被报告。

当某逻辑组合表达式被同时设置了HS_FLAG_QUIET标志时，该逻辑组合产生的所有匹配都不会被报告。

03

代码示例

下面将举例说明逻辑组合功能的用法。

1. 定义单条逻辑组合。部分子规则可以设置HS_FLAG_QUIET标志使它们对自身匹配保持静默，这不影响逻辑组合的结果：

```
hs_database_t *db = nullptr;
hs_compile_error_t *compile_err = nullptr;
const char *expr[] = {"abc",
                      "def",
                      "foobar.*gh",
                      "teakettle{4,10}",
```

```
                "ijkl[mMn]",
                "(101 & 102 & 103) | (104 & !105)"};
unsigned flags[] = {HS_FLAG_QUIET,
                    HS_FLAG_QUIET,
                    HS_FLAG_QUIET,
                    HS_FLAG_QUIET,
                    0,
                    HS_FLAG_COMBINATION};
unsigned ids[] = {101, 102, 103, 104, 105, 1001};
hs_error_t err = hs_compile_multi(expr, flags, ids, 6, HS_MODE_NOSTREAM,
                                  nullptr, &db, &compile_err);
```

2. 定义多条逻辑组合。逻辑组合表达式亦可根据需要设定HS_FLAG_SINGLEMATCH标志：

```
hs_database_t *db = nullptr;
hs_compile_error_t *compile_err = nullptr;
const char *expr[] = {"abc",
                "def",
                "foobar.*gh",
                "teakettle{4,10}",
                "ijkl[mMn]",
                "(101 & 102 & 103) | (104 & !105)",
                "!101 & 102",
                "!(!101 | 102)",
                "101 & !102"};
unsigned flags[] = {HS_FLAG_QUIET,
                    HS_FLAG_QUIET,
                    HS_FLAG_QUIET,
                    HS_FLAG_QUIET,
                    0,
                    HS_FLAG_COMBINATION | HS_FLAG_SINGLEMATCH,
                    HS_FLAG_COMBINATION,
                    HS_FLAG_COMBINATION | HS_FLAG_SINGLEMATCH,
                    HS_FLAG_COMBINATION | HS_FLAG_SINGLEMATCH};
unsigned ids[] = {101, 102, 103, 104, 105, 1001, 1002, 1003, 1004};
hs_error_t err = hs_compile_multi(expr, flags, ids, 9, HS_MODE_NOSTREAM,
                                  nullptr, &db, &compile_err);
```

送！京东电子卡

活动规则：带文字将本文转发至朋友圈，并将截图发至公众号后台，小编将随机抽取十位，每人送出一张京东十元电子卡！中奖名单见下期推送。

领取方式：后台发送卡号及卡密。

推荐阅读

2018 SPDK中国技术峰会系列：基于SPDK通用块层的QoS流量控制以及相关生态工具

DPDK加速FPGA支持，强强联手助力数据中心网络加速（文末福利继续~）

在虚拟机上使用SoftRoCE部署SPDK NVMe-oF

具有DPDK和英特尔®82599网络控制器的内联IPsec（文末福利继续哦~）

2018 SPDK中国技术峰会系列：SPDK现状介绍

DPDK与SPDK　开源社区