CEP In Flink (4) - 使用瓶颈

> 前三篇博客主要是描述了Apache Flink中CEP的相关原理，虽然Flink采用NFA和SharedBuffer对CEP做了优化，但我作为一个开发者，在使用CEP时，很多地方的稳定性或者说是可用性确实无法让我放心。

回顾

先贴上之前的三篇博客:

1. CEP In Flink (1) – CEP规则解析
2. CEP In Flink (2) – CEP规则匹配
3. CEP In Flink (3) – 匹配事件提取

瓶颈/缺陷

> 注意本文所涉及到的CEP相关内容均取自release-1.6。

Pattern

Pattern这方面，有两个问题：

- 不能同时匹配多个Pattern
- 不能动态修改Pattern
- 不支持NotFollowBy结尾语法

其实前两种情况在实际业务中非常常见的，例如每个公司会同时进行多个活动，或者在用户触达中动态修改策略实现用户的及时促活。目前社区中相关的ticket有FLINK-7129。

其实第三种加上Timeout也是一种很常见的CEP处理逻辑，目前现有代码微调后即可支持NotFollowBy和Timeout并存的情况，但是Flink在规则解析中还没解除这个限制。

人群过滤

我所了解的一些场景里，很多人使用CEP作为人群筛选的工具，比如在一个活动推广中点击了活动链接但是没有参与的人。如果要将这个场景放入Flink的CEP中，那么不得不针对每一个user创建一个NFA，想象一下，如果这个user的数量达到千万级，这对内存的压力会是一个什么样的结果。

EventTime处理逻辑

CEP当然在流式处理中是要支持EventTime的，那么相对应的要支持数据的晚到现象，也就是watermark的处理逻辑。在Flink的处理逻辑中，将晚到数据明细存储在了Map<Long, List<IN>>的结构中，也就是说，如果

watermark设置为当前时间减去5分钟，那么内存中就会存储5分钟的数据，这在我看来，也是对内存的极大损伤之一。