Flink Window API

左元

2021 年 7 月 20 日

尚硅谷大数据组

主要内容

- Window 概念
- Window 类型
- Window API

窗口（Window）

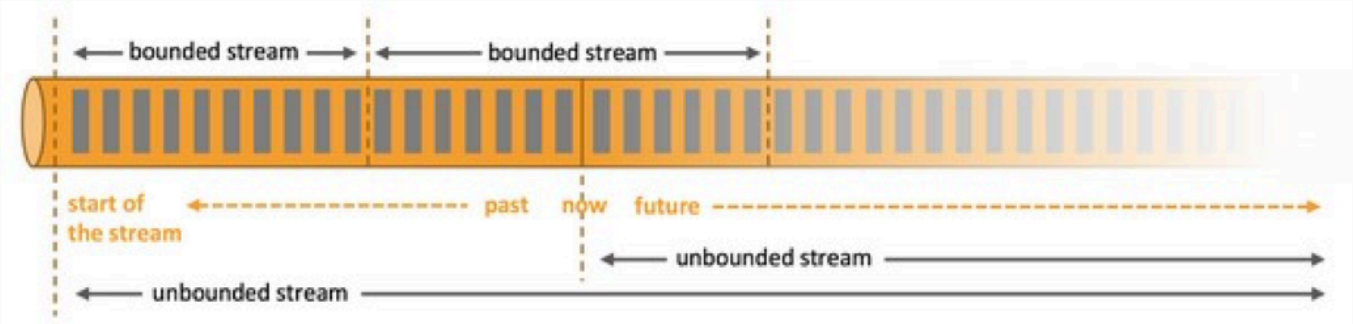

图 1：无界流

- 一般真实的流都是无界的，怎样处理无界的数据？
- 可以把无限的数据流进行切分，得到有限的数据集进行处理——也就是得到有界流
- 窗口（Window）就是将无限流切割为有限流的一种方式，它会将流数据分发到有限大小的桶（bucket）中进行分析

Window 类型

- 时间窗口（Time Window）
  - 滚动时间窗口
  - 滑动时间窗口
  - 会话窗口（只有 Flink 支持）
- 计数窗口（Count Window）
  - 滚动计数窗口
  - 滑动计数窗口

滚动窗口（Tumbling Windows）

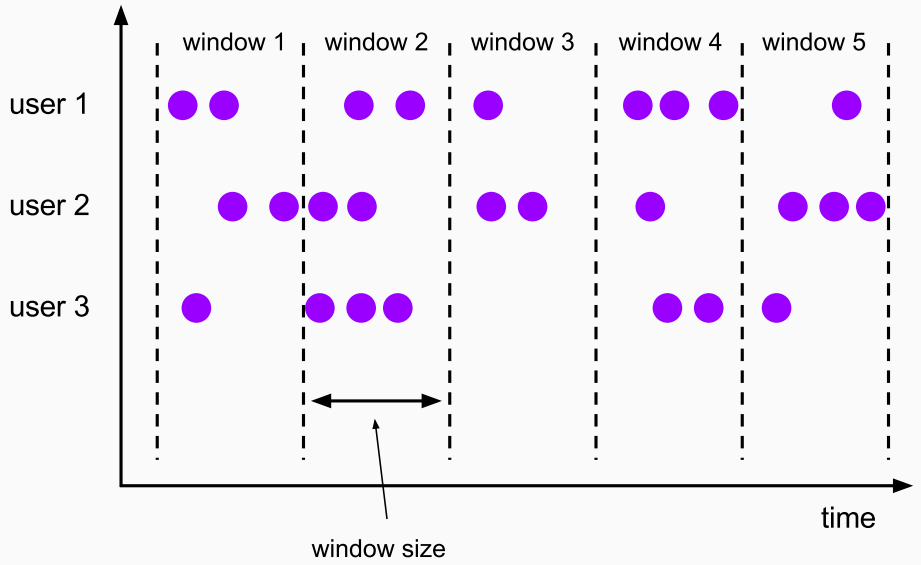

图 2: 滚动窗口

- 将数据依据固定的窗口长度对数据进行切分
- 时间对齐，窗口长度固定，没有重叠

滑动窗口（Sliding Windows）

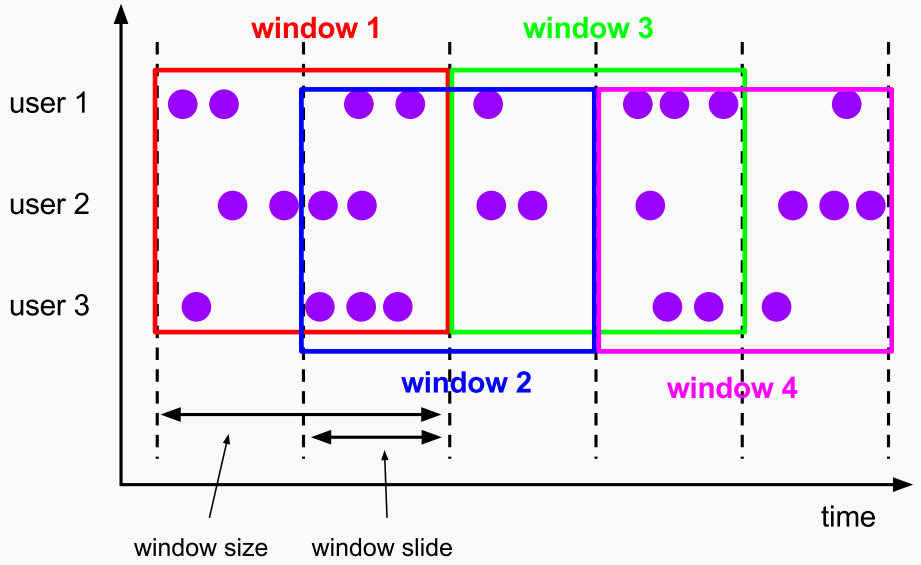

图 3: 滑动窗口

- 滑动窗口是固定窗口的更广义的一种形式，滑动窗口由固定的窗口长度和滑动间隔组成
- 窗口长度固定，可以有重叠

会话窗口（Session Windows）

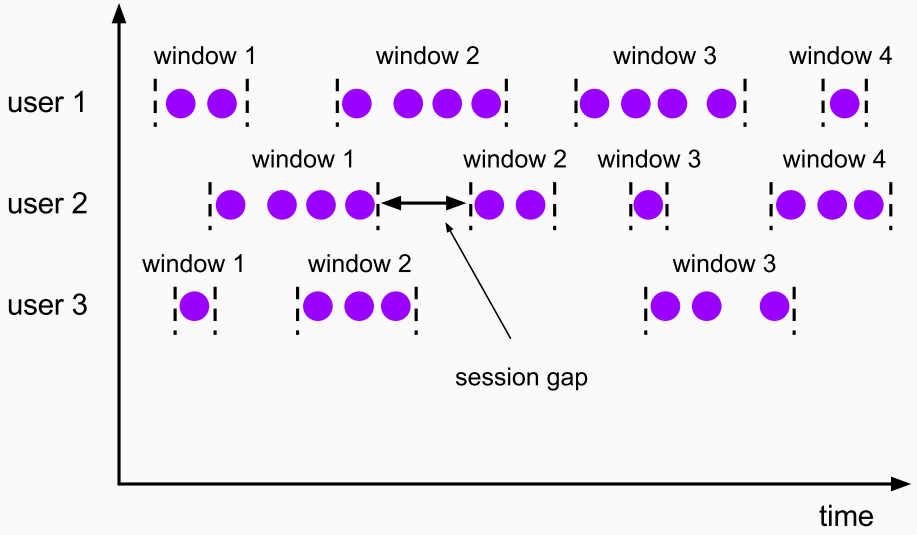

图 4: 会话窗口

- 由一系列事件组合一个指定时间长度的 timeout 间隙组成，也就是一段时间没有接收到新数据就会生成新的窗口
- 特点：时间无对齐
- 只有 Flink 支持会话窗口

Window API

- 窗口分配器——window() 方法
- 我们可以用.window() 来定义一个窗口，然后基于这个 window 去做一些聚合或者其它处理操作。

创建不同类型的窗口

- 处理时间窗口
- 事件时间窗口

处理时间窗口

- 滚动窗口

```
.window(TumblingProcessingTimeWindows.of(Time.seconds(5)))
```

- 滑动窗口

```
.window(SlidingProcessingTimeWindows.of(Time.seconds(10),
↪  Time.seconds(5)))
```

- 会话窗口

```
.window(ProcessingTimeSessionWindows.withGap(Time.seconds(10)))
```

事件时间窗口

- 滚动窗口

```
.window(TumblingEventTimeWindows.of(Time.seconds(5)))
```

- 滑动窗口

```
.window(SlidingEventTimeWindows.of(Time.seconds(10),
↪  Time.seconds(5)))
```

- 会话窗口

```
.window(EventTimeSessionWindows.withGap(Time.seconds(10)))
```

窗口聚合函数

- 窗口聚合函数定义了要对窗口中收集的数据做的计算操作
- 可以分为两类
  - 增量聚合函数
  - 全窗口聚合函数

增量聚合函数

- 每条数据到来就进行计算，只保存一个简单的状态（累加器）
- ReduceFunction, AggregateFunction
- 当窗口闭合的时候，增量聚合完成
- 处理时间：当机器时间超过窗口结束时间的时候，窗口闭合
- 来一条数据计算一次

增量聚合函数

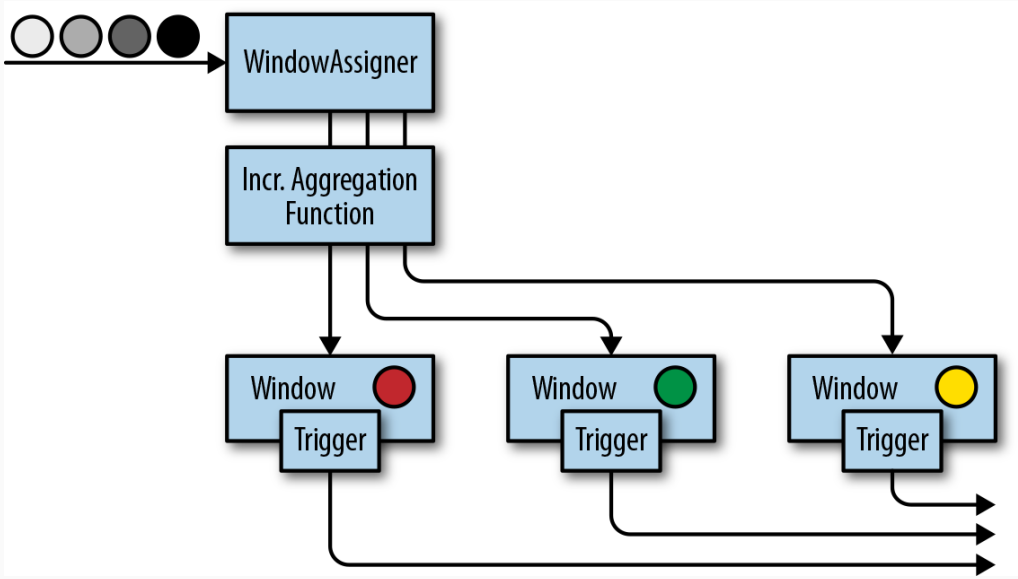

图 5: 增量聚合函数

全窗口聚合函数

- 先把窗口所有数据收集起来，等到计算的时候会遍历所有数据
- ProcessWindowFunction

全窗口聚合函数

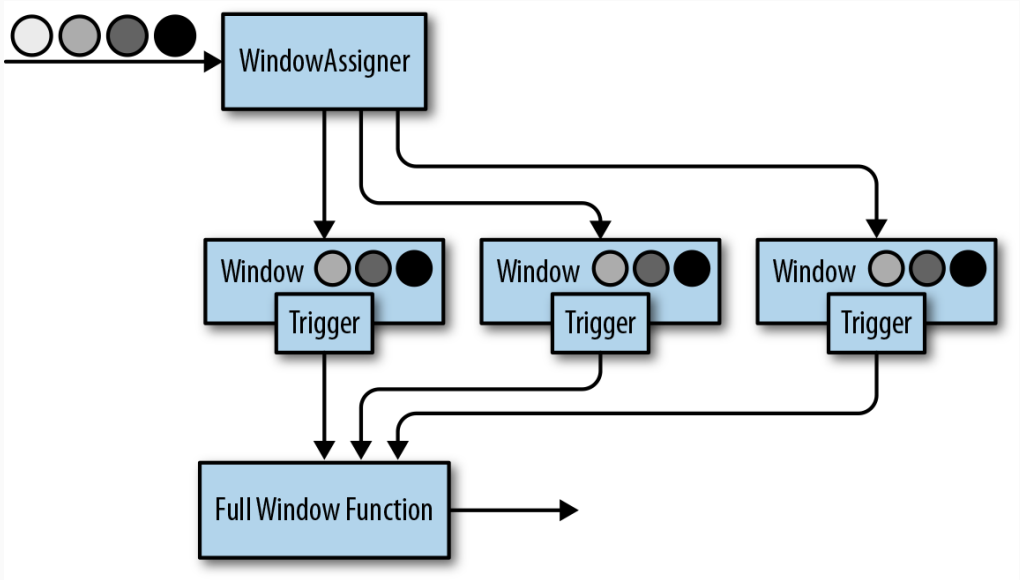

图 6: 全窗口聚合函数

增量聚合和全窗口聚合结合使用

- 可以访问窗口信息
- 不需要收集窗口中的所有元素，只需要维护一个累加器，节省内存

增量聚合和全窗口聚合结合使用

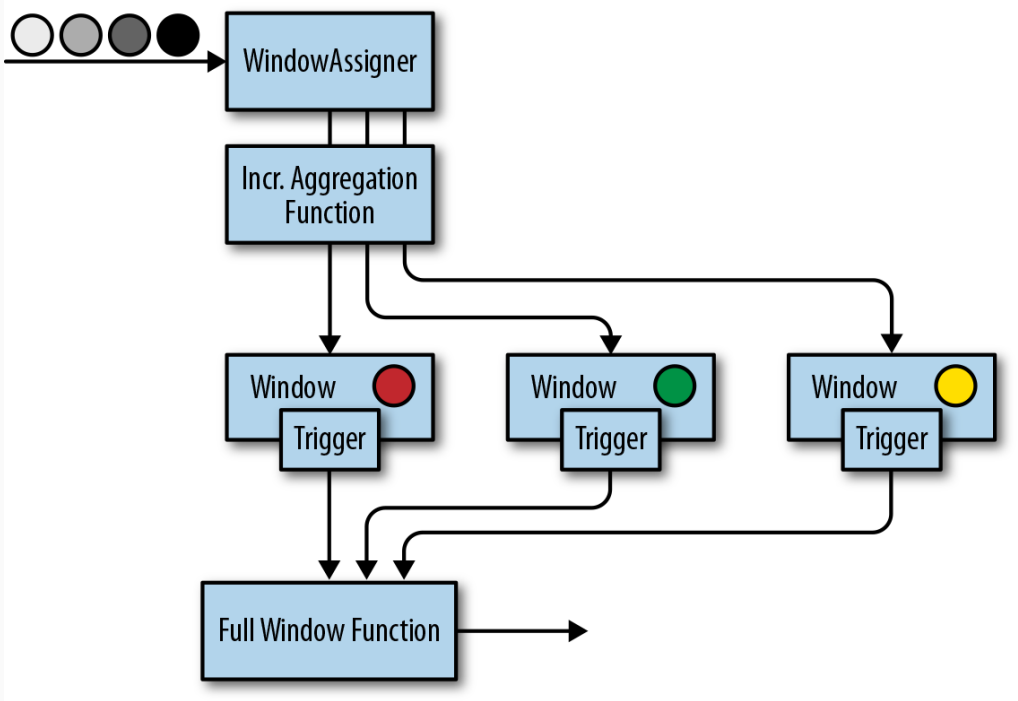

图 7：增量聚合和全窗口聚合结合使用

其他可选 API

- .trigger() ——触发器
  - 定义窗口什么时候关闭，触发计算并输出结果
- .evictor() ——移除器
  - 定义移除某些数据的逻辑
- .allowedLateness() ——允许处理迟到的数据
- .sideOutputLateData() ——将迟到的数据放入侧输出流
- .getSideOutput() ——获取侧输出流

基于 Key 的窗口

```
stream
    .keyBy(...)               <-  keyed versus non-keyed windows
    .window(...)              <-  required: "assigner"
    [.trigger(...)]            <-  optional: "trigger" (else default trigger)
    [.evictor(...)]            <-  optional: "evictor" (else no evictor)
    [.allowedLateness(...)]    <-  optional: "lateness" (else zero)
    [.sideOutputLateData(...)] <-  optional: "output tag" (else no side output for late data)
    .reduce/aggregate/fold/apply()      <-  required: "function"
    [.getSideOutput(...)]      <-  optional: "output tag"
```

不分流直接开窗口

```
stream
    .windowAll(...)           <-  required: "assigner"
    [.trigger(...)]            <-  optional: "trigger" (else default trigger)
    [.evictor(...)]            <-  optional: "evictor" (else no evictor)
    [.allowedLateness(...)]    <-  optional: "lateness" (else zero)
    [.sideOutputLateData(...)] <-  optional: "output tag" (else no side output for late data)
    .reduce/aggregate/fold/apply()      <-  required: "function"
    [.getSideOutput(...)]      <-  optional: "output tag"
```

Q & A