



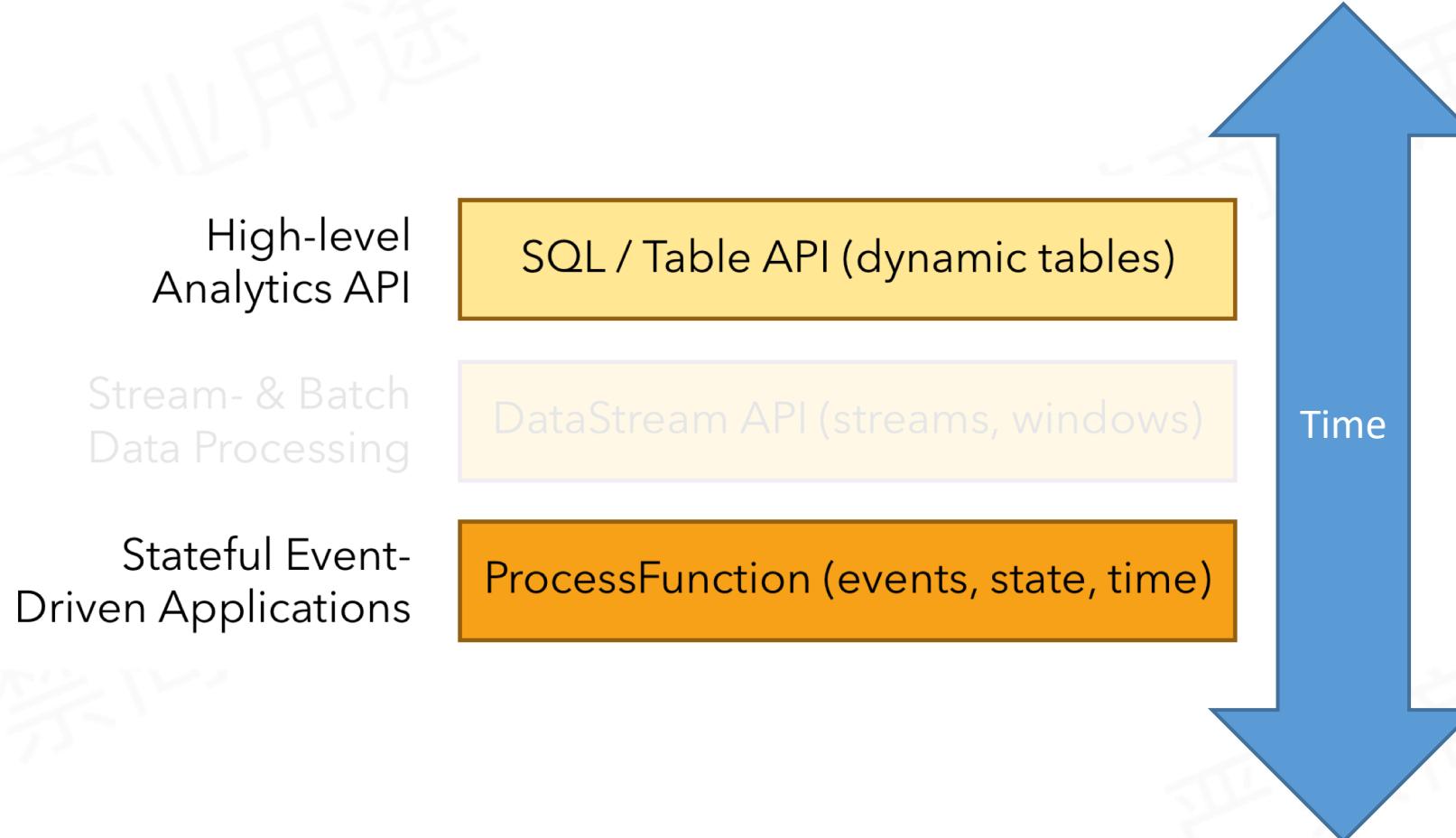
# Flink Time深度解析

崔星灿 · York University / Apache Flink Committer

Aache Flink 在线教程 – 2019年05月21日



# 时间在Flink中的地位





Apache Flink

# CONTENT

## 目录 >>

01 /

Flink时间语义

02 /

Event Time和Watermark

03 /

Table中的时间

04 /

深入思考

# 01

---

Flink时间语义

---

# 时间语义

## 酬乐天扬州初逢席上见赠

唐·刘禹锡

巴山楚水凄凉地，二十三年弃置身。

怀旧空吟闻笛赋，到乡翻似烂柯人。

沉舟侧畔千帆过，病树前头万木春。

今日听君歌一曲，暂凭杯酒长精神。

时间不是绝对的，不同“世界”中的时间语义各不相同



# Flink支持的时间语义

Processing Time	Event Time (Row Time)
真实世界的时间	数据世界的时间
处理数据节点的本地时间	记录携带的Timestamp
处理简单	处理复杂
结果不确定（无法重现）	结果确定（可重现）

# 时间的特性

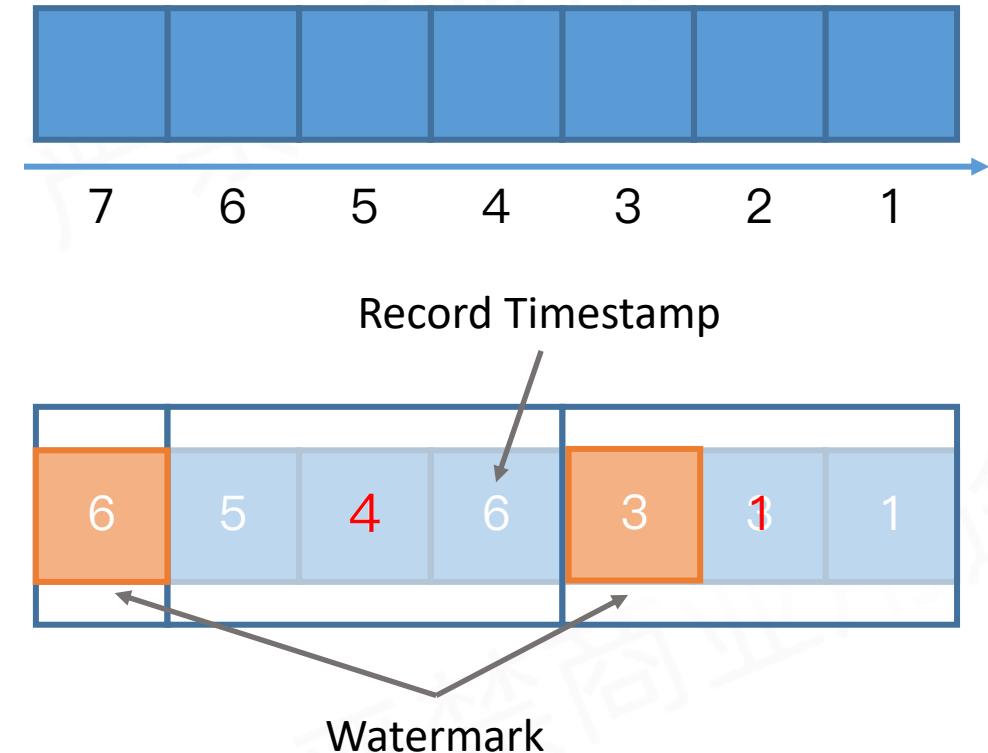


逝者如斯夫，不舍昼夜。

——《论语》

Processing Time  
递增

Event Time  
一定程度的乱序



# 02

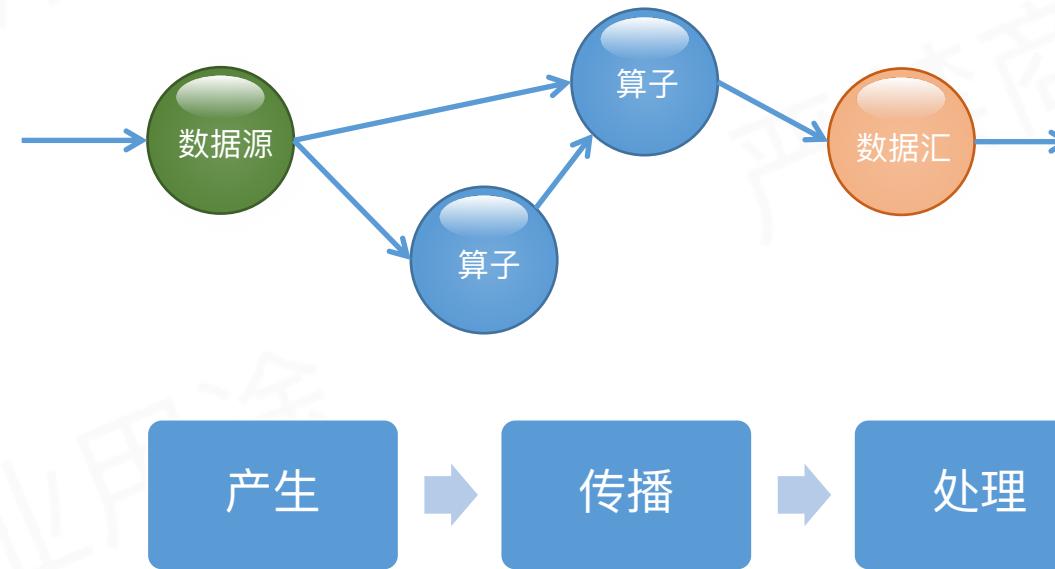
---

## Timestamp和Watermark

---



# Timestamp和Watermark行为概览





# Timestamp分配和Watermark生成

- 在SourceFunction中产生
  - collectWithTimestamp(T element, long timestamp)
  - emitWatermark(Watermark mark)
- 在流程中指定
  - DataStream.assignTimestampsAndWatermarks(...)

定期生成	根据特殊记录生成
现实时间驱动	数据驱动
每隔一段时间调用生成方法	每一次分配Timestamp都会调用生成方法
实现AssignerWithPeriodicWatermarks	实现AssignerWithPunctuatedWatermarks

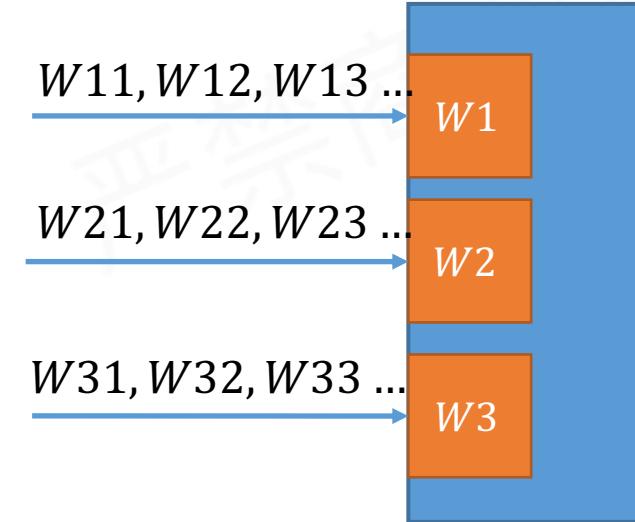


# Watermark传播

- Watermark以广播的形式在算子之间进行传播
- Long.MAX\_VALUE表示不会再有数据
- 单输入取其大，多输入取其小

局限：

没有区分逻辑上的单流和多流，强制同步时钟



单个输入  $W_n = \max(W_{n_1}, W_{n_2}, W_{n_3} \dots)$

整个任务  $W = \min(W_1, W_2, W_3)$



# ProcessFunction

---

- 获取记录的Timestamp或当前ProcTime
- 获取算子时间 (Watermark)
- 注册Timer并提供回调逻辑
  - registerEventTimeTimer()
  - registerProcessingTimeTimer()
  - onTimer()

时间服务 (Time Service)



Apache Flink

# Watermark处理



# 03

---

Table中的时间

---



# Table中指定时间列

Processing Time

从DataStream转化	通过TableSource生成
<code>tEnv.fromDataStream(stream, "f1, f2, f3.proctime")</code>	TableSource实现 <code>DefinedProctimeAttributes</code> 接口

Event Time

从DataStream转化	通过TableSource生成
原始DataStream必须有Timestamp及Watermark	数据中存在类型为long或timestamp的时间字段
<code>tEnv.fromDataStream(stream, "f1, f2, f3.rowtime")</code> <code>tEnv.fromDataStream(stream, "f1.rowtime, f2, f3")</code>	TableSource实现 <code>DefinedRowtimeAttributes</code> 接口



# 时间列和Table操作

- Over窗口聚合 (Over Window Aggregation)
- Group By窗口聚合 (Group By Window Aggregation)
- 时间窗口连接 (Time-WINDOWED Join)
- 排序 (Order By)



# 04

---

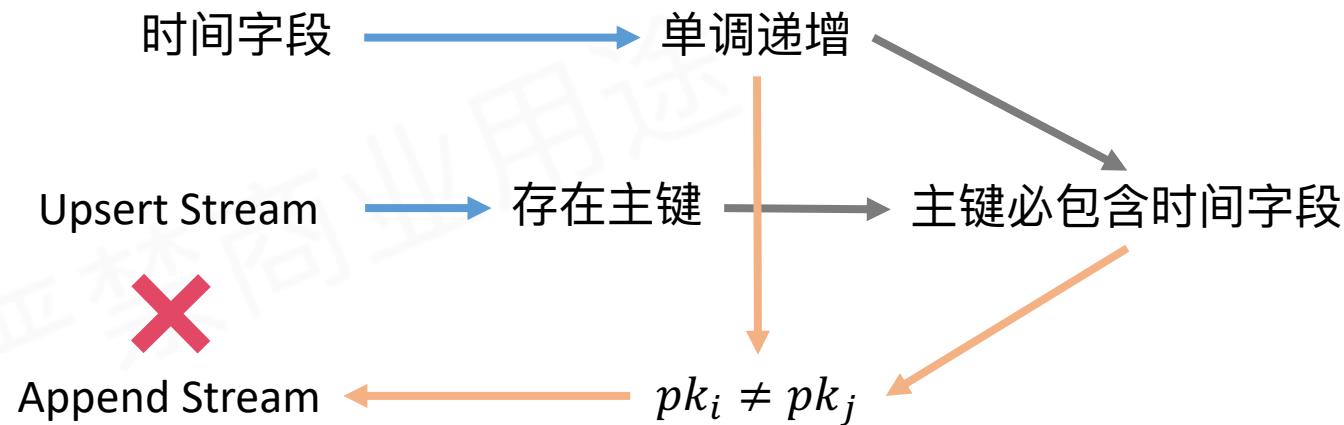
## 深入思考

---



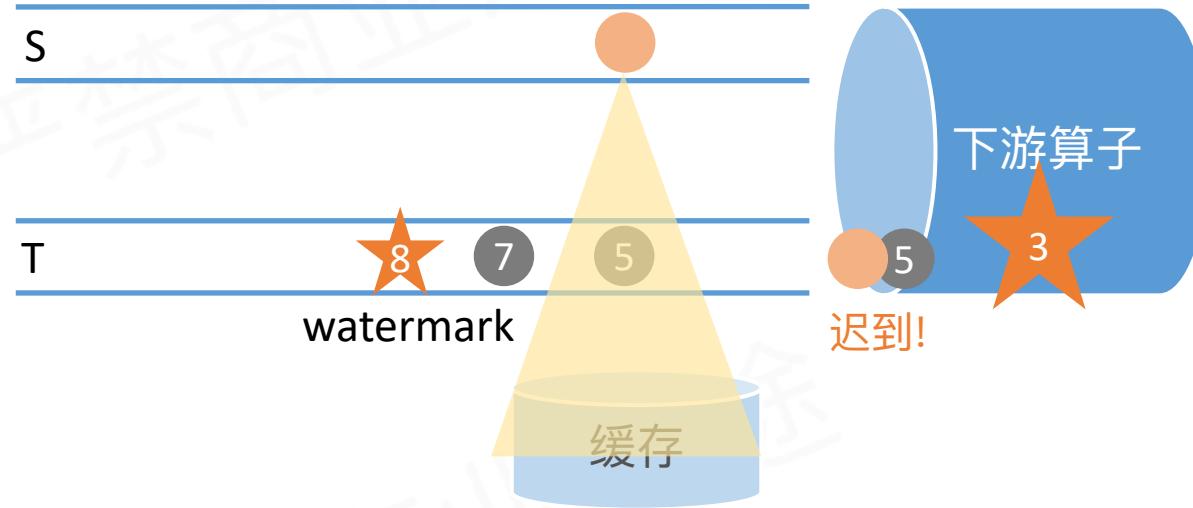
# 数据 or 元数据?

- DataStream API中，时间是元数据
- SQL / Table API中，时间是数据
- 在Upsert Stream中不允许出现时间字段





# 干涉Watermark传播



在发送watermark之前减去一个固定值



# 统一EventTime和ProcTime处理

目标	Event Time	Processing Time
获取记录时间	context.getTimestamp()或从数据字段中获取	timerService().currentProcessingTime()
获取算子时间	timerService().currentWatermark()	当前Processing Time
注册计时器	timerService.registerEventTimeTimer()	timerService.registerProcessingTimeTimer()



# 时间和Watermark本质

---

- 流处理中时间本质上就是一个普通的递增字段，不一定真的表示时间
- Watermark只是应对乱序的办法之一，大多是启发式的，在延迟和完整性之间抉择



Apache Flink

# THANKS

Flink China社区大群



扫一扫群二维码，立刻加入该群。