

Window Functions



Window Functions

A mechanism to group events by time chunks

- need to timestamp events so Flink can group them
- need to specify a watermarking strategy to discard late data

```
val eventStream: DataStream[ServerEvent] = env
    .fromCollection(events)
    .assignTimestampsAndWatermarks( // extract timestamps for events (event time) + watermarks
        WatermarkStrategy
            .forBoundedOutOfOrderness(java.time.Duration.ofMillis(500))
    // once you get an event with time T, you will NOT accept further events with time < T - 500
        .withTimestampAssigner(new SerializableTimestampAssigner[ServerEvent] {
            override def extractTimestamp(element: ServerEvent, recordTimestamp: Long) =
                element.eventTime.toEpochMilli
        })
    )
```

Window Functions

WindowAll: a way to group all events by time

```
val threeSecondsTumblingWindow =  
  eventStream.windowAll(TumblingEventTimeWindows.of(Time.seconds(3)))
```

AllWindowFunction: a window handler after Flink does the splitting

```
class CountByWindowAll extends AllWindowFunction[ServerEvent, String, TimeWindow] {  
  // ^ input ^ output ^ window type  
  override def apply(window: TimeWindow, input: Iterable[ServerEvent], out: Collector[String]) = {  
    val registrationEventCount = input.count(event => event.isInstanceOf[PlayerRegistered])  
    out.collect(s"Window ${window.getStart} - ${window.getEnd} $registrationEventCount")  
  }  
}
```

```
val registrationsPerThreeSeconds: DataStream[String] =  
  threeSecondsTumblingWindow.apply(new CountByWindowAll)
```

Window Functions

Keyed streams: splitting the events by unique values given by a function

```
val streamByType: KeyedStream[ServerEvent, String] =  
  eventStream.keyBy(e => e.getClass.getSimpleName)
```

WindowFunction: a window handler for keyed streams

```
class CountByWindow extends WindowFunction[ServerEvent, String, String, TimeWindow] {  
  //  
  override def apply(  
    key: String,  
    window: TimeWindow,  
    input: Iterable[ServerEvent],  
    out: Collector[String]  
  ): Unit =  
  out.collect(s"$key: $window, ${input.size}")  
}
```

Window Types

Tumbling event time windows

- windows don't overlap

Sliding event time windows

- windows may overlap
- given by 2 parameters: window length, sliding duration

Session windows

- all events with no more than a time gap between them
- may yield unequal windows

Global windows

- not time-dependent

Custom window assigners

- advanced

Flink rocks

