白话Flink概念

**Window Assigners：**Window Assigners定义元素如何选择要被发送到指定窗口

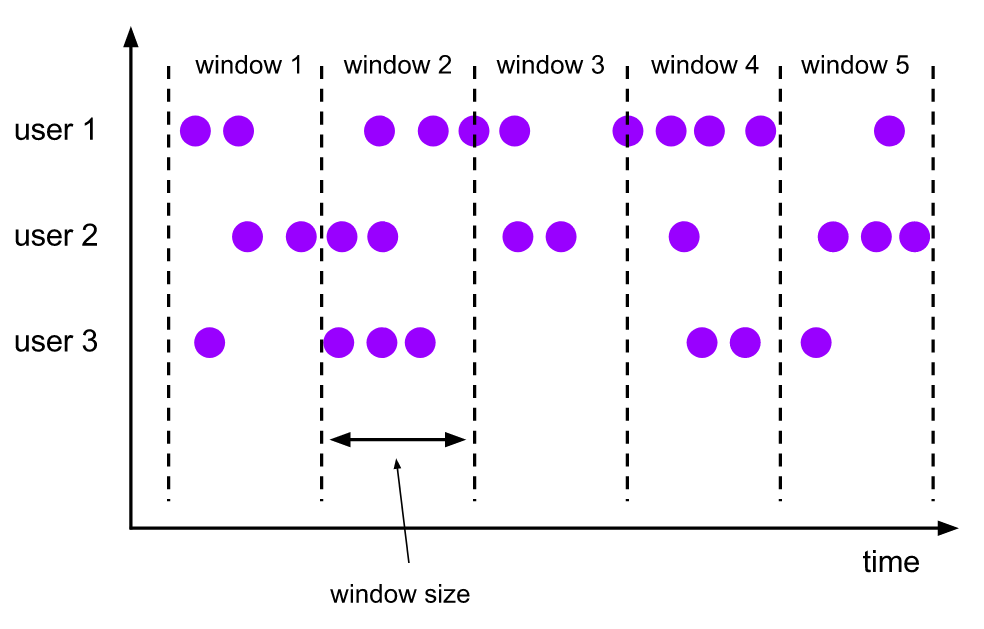
https://ci.apache.org/projects/flink/flink-docs-release-1.2/dev/windows.html#window-assigners

接下来查看一下Flink预定义的window assigners 如何工作，如何在Streaming程序中使用！如下图可视化了每一种assigner，其中紫色圆表示元素，这些元素根据case class user的key进行分区，横轴表示时间：

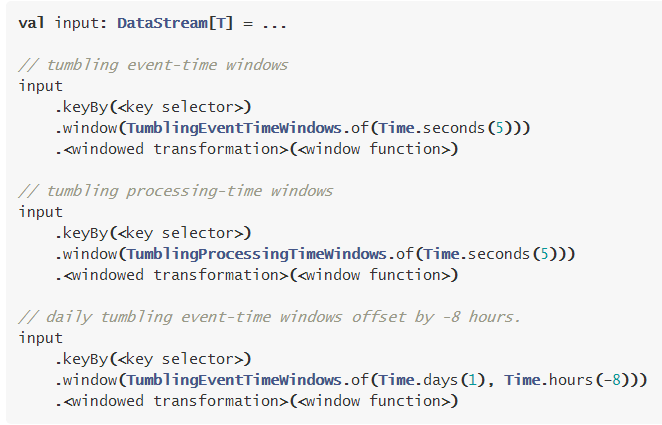
**Tumbling Windows：**

翻动窗口(和滑动窗口区别两个相邻窗口没有重叠元素)

翻动窗口assigner 指定每一个元素到一个指定大小(一般指时间)的窗口中，翻动窗口是固定大小和没有重叠的。例如：如果你指定翻动窗口大小为5min时，将每5分钟开始计算一个窗口和没五分钟创建一个新的窗口，如下图所示：



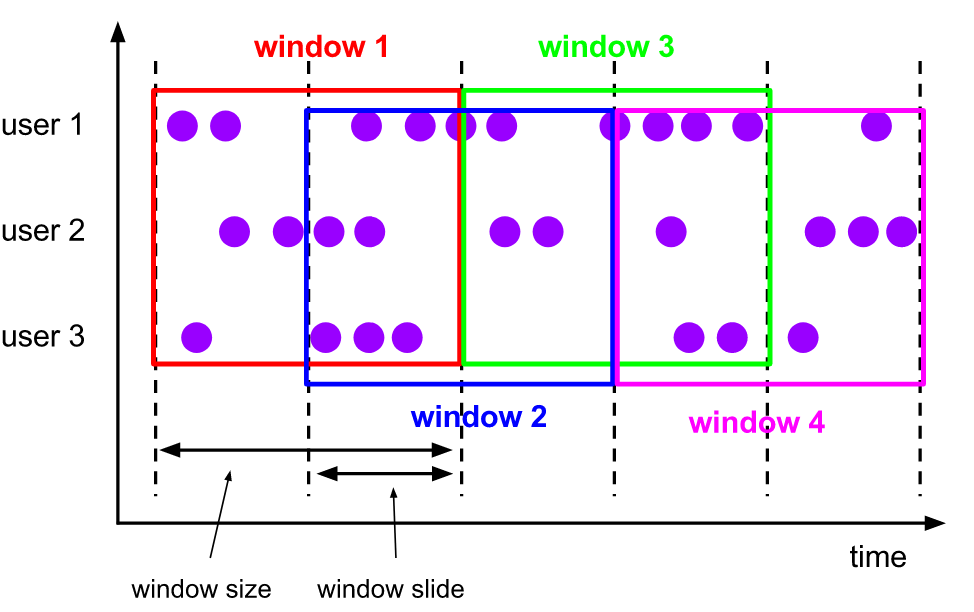
**示例代码：**



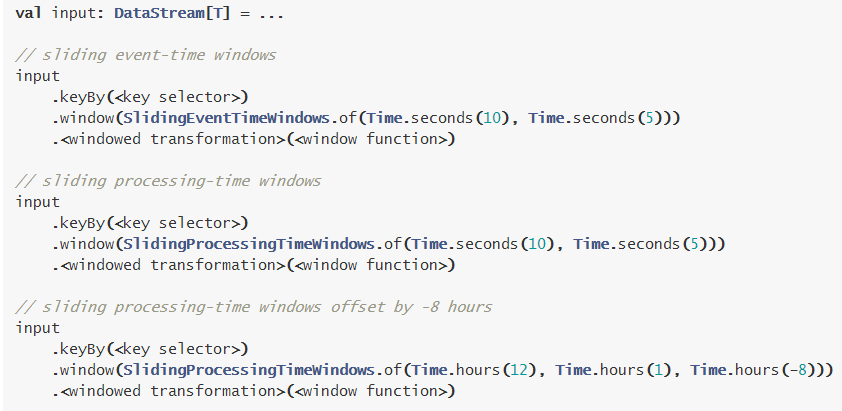
最后注意指定时域，在中国必须指定：Time.hours(-8)，更多细节参见官方文档

**Sliding Windows：**

滑动窗口assigner 指定元素到固定长度的窗口中。类似于翻动的窗口assigner ，窗口大小使用窗口参数设置，额外一个滑动窗口参数是控制滑动的频率，因此：如果滑动长度小于窗口长度的话，滑动窗口内元素是有重叠的，这种情况下一个元素可能在多个窗口中出现。



示例代码：

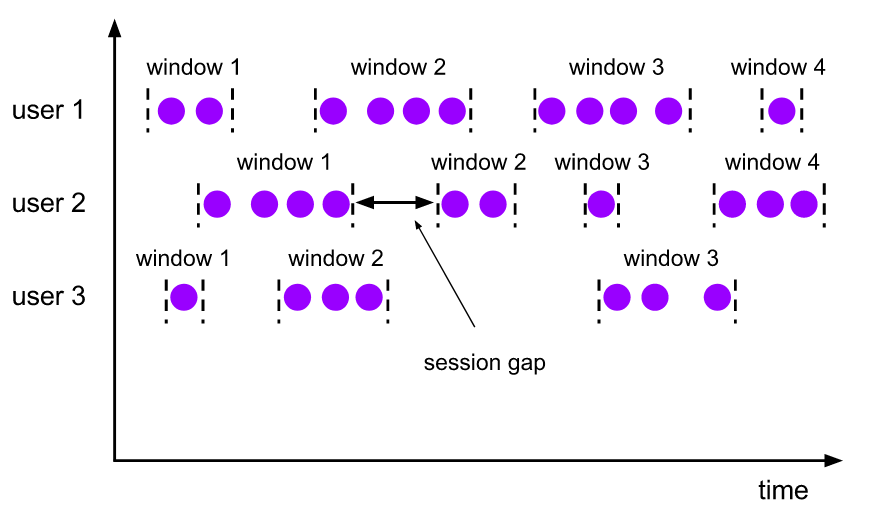


**Session Windows：**

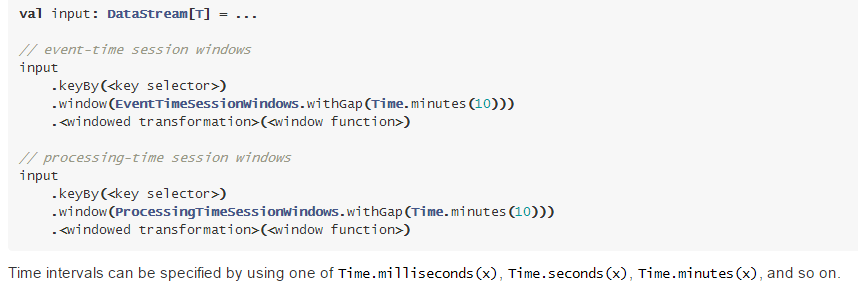
预习一下JavaEE的Session，Session有个超时时间，就是当两次web请求中间的时间间隔超过”TimeOut”的话。Session将失效，这就是JavaEE的Session。

接下来看一下Session Windows：

会话窗口assigner是根据当前活动的Session进行分组。和翻动窗口，滑动窗口对比的话，Session窗口没有重叠而且没有固定大小。相反，当会话窗口持续N个单位时间(N个单位时间就是指超时时间)没有接收到元素的话，会话窗口便会关闭。会话窗口assigner需要配置session gap指定不活动的时间间隔(就是超时时间)。当会话窗口超时时间用尽，当前会话将会关闭，后续元素将会指定到一个新的窗口。



示例代码：

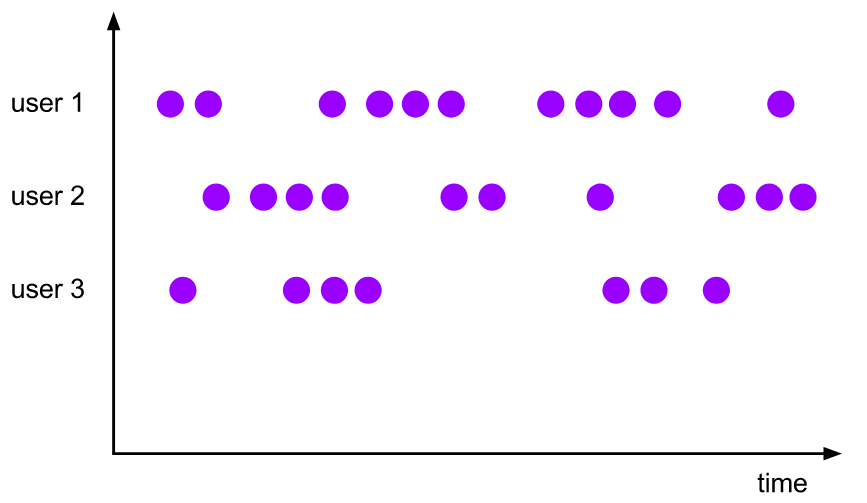


**Attention** ：

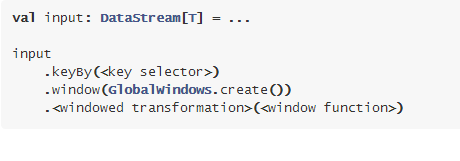
因为会话窗口没有固定开始和结束，因此会话窗口的计算和翻动、滑动窗口不同。内部，会话窗口操作符会创建一个新的窗口为到达的记录，并且合并这些窗口（指这些窗口之间的间隔时间小于超时时间）。为了可以合并，会话窗口操作符需要一个合并的触发器( [Trigger](https://ci.apache.org/projects/flink/flink-docs-release-1.2/dev/windows.html#triggers) )和一个合并窗口函数，例如：ReduceFunction 或者WindowFunction (FoldFunction 函数不能合并.)

**Global Windows：**

全局窗口assigner 将会指定带有相同key的元素到相同的全局窗口中，全局窗口的使用必须指定 “自定义一个[trigger](https://ci.apache.org/projects/flink/flink-docs-release-1.2/dev/windows.html#triggers)”，否则不会触发计算，因为全局窗口没有” 结束标识”供我们进行计算！



示例代码：



**Window Functions：**

窗口计算函数，就是窗口的计算逻辑。用来处理窗口中的元素。窗口函数有： ReduceFunction, FoldFunction 或 WindowFunction前面两个函数比较高效，因为他们可以在元素到时候立即执行计算，而WindowFunction需要窗口元素全部到来才能执行计算。

书签：<https://ci.apache.org/projects/flink/flink-docs-release-1.2/dev/windows.html#window-functions>

Triggers

触发器决定一个准备的窗口何时被窗口函数处理，每一个窗口指定器都有一个默认的触发器，如果默认的触发器无法满足需要的话，你可以指定一个自定义的触发器。

触发器接口有五个方法允许触发器响应不同的事件：

* The onElement() method is called for each element that is added to a window.
* The onEventTime() method is called when a registered event-time timer fires.
* The onProcessingTime() method is called when a registered processing-time timer fires.
* The onMerge() method is relevant for stateful triggers and merges the states of two triggers when their corresponding windows merge, *e.g.* when using session windows.
* Finally the clear() method performs any action needed upon removal of the corresponding window.

如上需要注意的两个方法：

1) The first three decide how to act on their invocation event by returning a TriggerResult. The action can be one of the following:

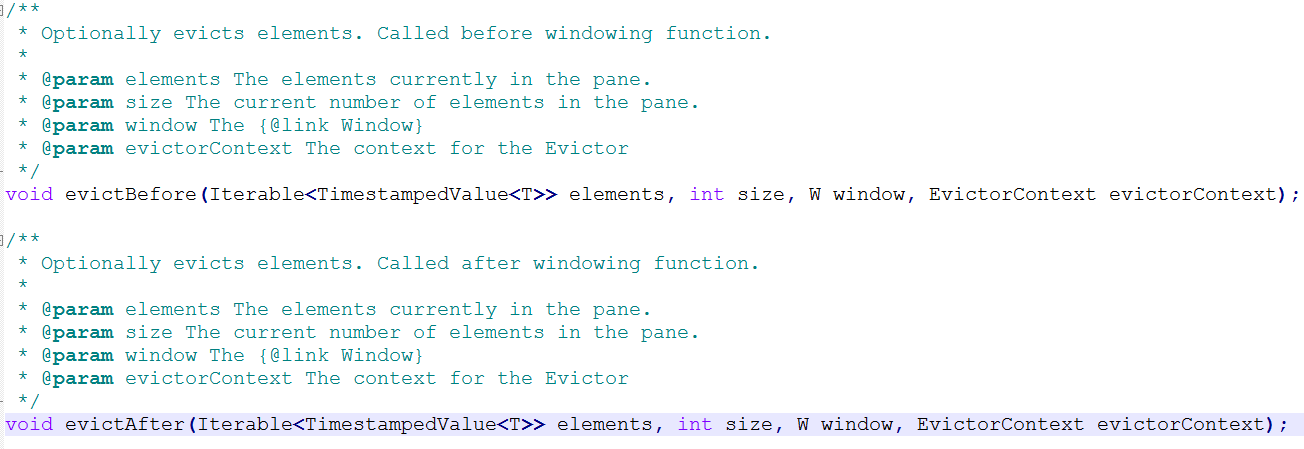
* CONTINUE: do nothing,
* FIRE: trigger the computation,
* PURGE: clear the elements in the window, and
* FIRE\_AND\_PURGE: trigger the computation and clear the elements in the window afterwards.

2) Any of these methods can be used to register processing- or event-time timers for future actions.

Fire and Purge：

Evictors：

Flink窗口模型除了可以指定WindowAssigner和 Trigger，也允许指定一个Evictor ，使用evictor(...) 方法完成指定，evictor 可以在启动计算之后，在应用窗口函数之前移除元素。Evictor 接口有两个方法：



evictBefore方法包含在应用窗口函数之前的移除元素的逻辑，而evictAfter方法包含应用窗口函数处理完毕之后的逻辑。在应用窗口函数之前被移除的元素是不会被处理的。

Flink具备三个预先实现的Evictor：

* CountEvictor: keeps up to a user-specified number of elements from the window and discards the remaining ones from the beginning of the window buffer.
* DeltaEvictor: takes a DeltaFunction and a threshold, computes the delta between the last element in the window buffer and each of the remaining ones, and removes the ones with a delta greater or equal to the threshold.
* TimeEvictor: takes as argument an interval in milliseconds and for a given window, it finds the maximum timestamp max\_ts among its elements and removes all the elements with timestamps smaller than max\_ts - interval.

**Default** By default, all the pre-implemented evictors apply their logic before the window function.

**Attention** Specifying an evictor prevents any pre-aggregation, as all the elements of a window have to be passed to the evictor before applying the computation.（指定evictor之后将会预先聚集，例如窗口函数的reduce算子会元素到达时候进行处理）

**Attention** Flink provides no guarantees about the order of the elements within a window. This implies that although an evictor may remove elements from the beginning of the window, these are not necessarily the ones that arrive first or last.（窗口中元素先后顺序不做保证）

Allowed Lateness：

撒大声地