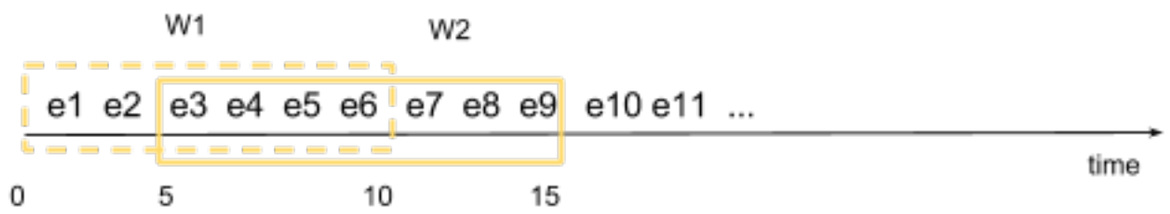
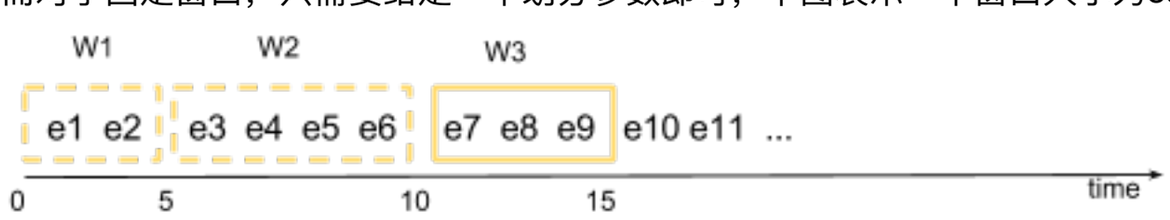
一. 窗口介绍

在分布式计算中，基于数据窗口的计算是一个非常常见的应用场景，比如说聚类、模式识别等。

storm支持两种方式的窗口：滑动窗口和固定窗口，并且支持从两种维度进行窗口分割：时间或tuple数。比如说用时间分割来实现一个滑动窗口，需要给定两个数值，窗口大小和滑动时间。一个窗口大小为10sec，滑动时间为5sec的窗口示意如下：



如图中所示，对这个流进行窗口计算的bolt会收到两次execute调用，一次是包含6个tuple的w1，一次是包含7个tuple的w2，是通过时间来进行划分的。而对于固定窗口，只需要给定一个划分参数即可，下图表示一个窗口大小为5sec的固定窗口：



Storm可同时处理窗口内的所有tuple。窗口可以从时间或数量上来划分，由如下两个因素决定：

- 窗口的长度，可以是时间间隔或Tuple数量；
- 滑动间隔(sliding Interval)，可以是时间间隔或Tuple数量；

1. **Sliding Window：滑动窗口**
   按照固定的时间间隔或者Tuple数量滑动窗口。
   - 如果滑动间隔和窗口大小一样则等同于滚窗
   - 如果滑动间隔大于窗口大小则会丢失数据
   - 如果滑动间隔小于窗口大小则会窗口重叠

2. **Tumbling Window：滚动窗口**
   元组被单个窗口处理，一个元组只属于一个窗口，不会有窗口重叠，一般用滚动就可以了。

二. 窗口实战

首先我们需要一个处理窗口的bolt，这个bolt需要实现IWindowedBolt接口，它与IBolt几乎相同，唯一的差异是其execute函数的参数为TupleWindow。

通常来讲我们都不要直接去implement这个接口，而是继承BaseWindowedBolt，因为实现接口的话需要提供一个windowConfigure的map来指定窗口参数，而BaseWindowedBolt用fluent风格实现了配置api，直接调用即可。

看一下几个API函数的定义：

```
1. public BaseWindowedBolt withWindow(Count windowLength, Count slidingInterval);
2. public BaseWindowedBolt withWindow(Count windowLength, Duration slidingInterval);
3. public BaseWindowedBolt withWindow(Duration windowLength, Count slidingInterval);
4. public BaseWindowedBolt withWindow(Duration windowLength, Duration slidingInterval);
5. public BaseWindowedBolt withWindow(Count windowLength);
6. public BaseWindowedBolt withWindow(Duration windowLength);
7. public BaseWindowedBolt withTumblingWindow(Count count);
8. public BaseWindowedBolt withTumblingWindow(Duration duration);
9. public BaseWindowedBolt withTimestampField(String fieldName);
10. public BaseWindowedBolt withLag(Duration duration);
11. public BaseWindowedBolt withWatermarkInterval(Duration interval);
```

可以看到，API支持用duration和tuple count两种方式来配置窗口