**Spark Streaming**

**Twitter Stream Processor Application**

1 Create twitter credentials

<https://apps.twitter.com/>

2. build scala code

**mvn scala:compile -DdisplayCmd=true**

3. run scala code

4. build java code

**mvn clean package**

note the equivalent java code is more verbose than the scala code

verbosity

the fact or quality of using more words than needed; wordiness.

When an output operator is called, it triggers the computation of a stream.

Without output operator on DStream no computation is invoked. basically you will need to invoke any of below method on stream

print()

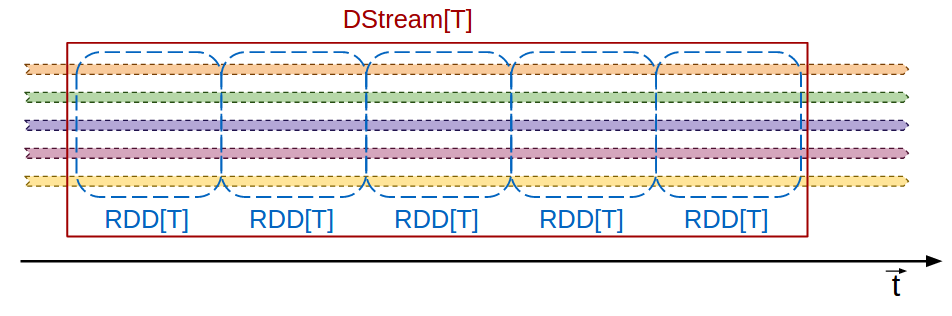
foreachRDD(func)

saveAsObjectFiles(prefix, [suffix])

saveAsTextFiles(prefix, [suffix])

saveAsHadoopFiles(prefix, [suffix])

An DStream is a collection of RDDs over time. Each RDD is composed of partitions of data distributed across the cluster of Spark workers.



In this illustration, each colored line represents a stream of data. When it's bounded in an rdd, it's a partition of the RDD. The RDD is composed by the data collected at each time interval, represented here by the blue box.

DStream.saveAsTextFile will create a file for the RDD created at interval (bluc box) each part-file corresponds to the piece of the colored line bounded by such RDD (colored line within blue blox).

In a distributed file system, like hdfs, the fs will abstract out the partitioning, presenting you with a single logical file, or 1 file per RDD.

When you use the local file system, we will see the partitions as part-files

#### **Window Operations**

