Arbitrary Stateful Processing

Objective

Aggregate data in an arbitrary way

Manage stream data "manually"

Scenario

You're running a social network and people share updates

Your social network needs to store everything

You want to compute the average storage used by every type of post

```
text,3,4812
video,1,5328585
text,4,5326
audio,5,523662

// type of posts, count of posts, total storage used
```

Average storage = total storage / total count, grouped by post type

Takeaways

Define a function to update state based on new records

```
socialPosts
     .groupByKey(_.postType)
     .mapGroupsWithState(GroupStateTimeout.NoTimeout())(updateAvgStorage)
     .writeStream
     .format("console")
     .outputMode("update")
     .start()
     .awaitTermination()
```

group first

append not supported

```
def updateAvgStorage(
     postType: String,
     group: Iterator[SocialPostRecord],
     state: GroupState[SocialPostBulk]
): AveragePostStorage = {

    // your code here
}
```

key by which the records are grouped

new records to aggregate

current state

final value to be returned for the whole group

Also flatMapWithState when a group produces multiple outputs

Spark rocks