Spark

In this presentation I will show you one apprach to building a local Spark application that consumes, transforms and stores data for just-in-time visual analysis.

Stream

Core to our application is a stream — composing a *Source ~> Flow ~> Sink*.

Stream components might include:

- Source ~ Cassandra, Hdfs, Kafka, Text, …
- Flow ~ Spark Transformations and Actions
- Sink ~ Cassandra, Hdfs, Kafka, Text, …

Technologies

We'll use these technologies:

- Scala
- ScalaFX
- Spark
- Kafka
- Cassandra
- Cassandra-Spark Connector
- Zookeeper

Visuals

Spark visuals can be built with ScalaFX, a JavaFX wrapper, without having to use Javascript. It's worth noting:

- JavaFX includes Charts, 2D, 3D and Animations.
- JavaFX apps and controls are decorated by CSS.
- Third party JavaFX charts and components exist.
- JavaFX includes a WebView component that allows for the rendering of Javascript content. For instance, D3 could be used to build an exotic chart.

Visual Spark

A ScalaFX application that executes and visualizes a *Kafka Source ~> Spark Flow ~> Casandra Sink* stream.

The next slide contains a screen shot of Visual Spark after executing and visualizing a stream.

Visual Spark

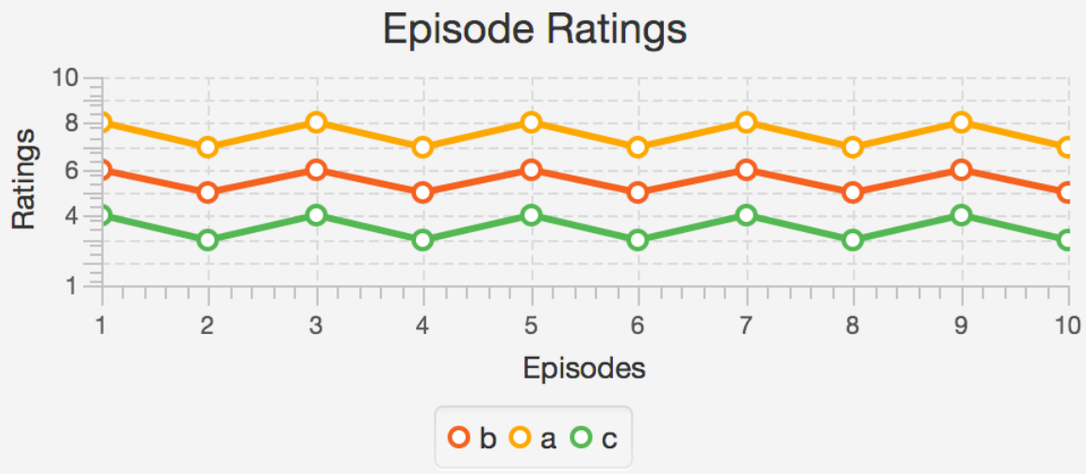
Play

30 messages processed in 22 seconds.

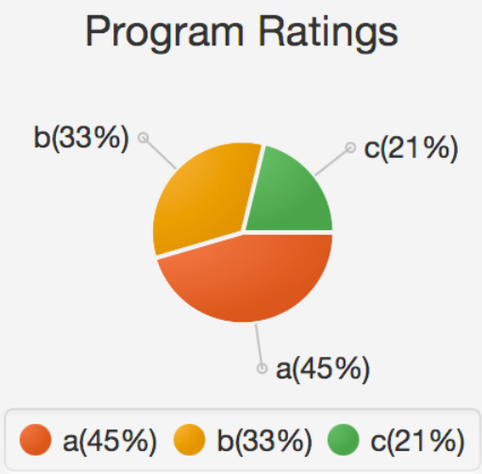
Source

Program	Season	Episode	Rating
a	1	1	8
a	1	2	7
a	1	3	8
a	1	4	7
a	1	5	8
a	1	6	7
a	1	7	8
a	1	8	7

Flow



Sink



Next

Let’s examine the project* and view a demo.

* For details, see: github.com/objektwerks