

Spark

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

Stream

Core to our application is the concept of a stream, composing a *Source* \leadsto *Flow* \leadsto *Sink*.

Stream components might include:

- Source \sim Cassandra, Hdfs, Kafka, Text, ...
- Flow \sim Spark Transformations and Actions
- Sink \sim Cassandra, Hdfs, Kafka, Text, ...

Technologies

The application is built with these technologies:

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

Visuals

Visuals are built with ScalaFX, a JavaFX wrapper. JavaFX features include:

- Charts, 2D, 3D and Animations.
- CSS styling for all components.
- Third party components.
- WebView component for Javascript content, such as D3.

Visual Spark

Visual Spark is a ScalaFX application that executes and visualizes a *Kafka Source* ~> *Spark Flow* ~> *Cassandra Sink* stream.

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

Play

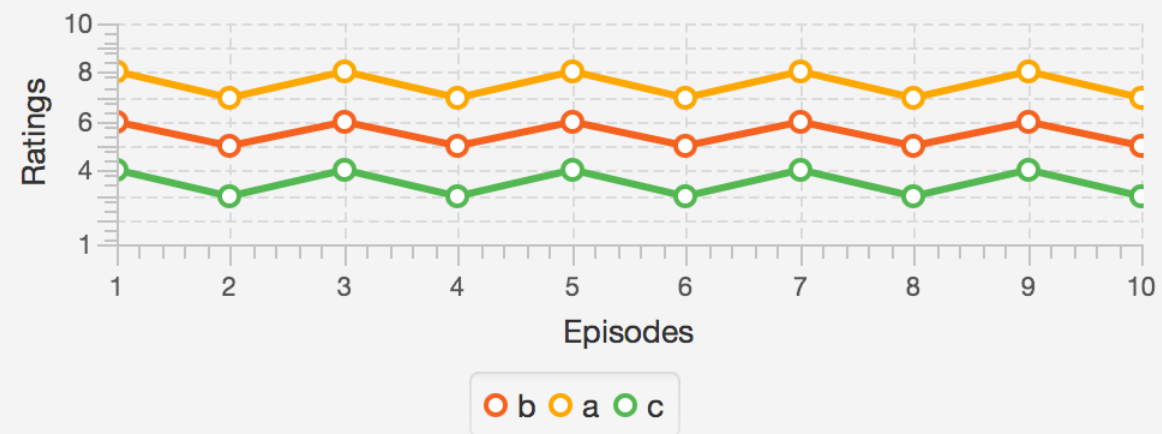
30 messages processed in 22 seconds.

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	

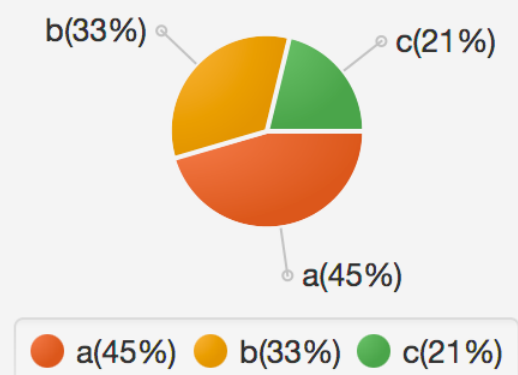
Flow

Episode Ratings



Sink

Program Ratings



Next

Next let's examine the project source code and view a demo.*

* For details, please see: github.com/objektwerks