SSR: Structured Streaming for R and Machine Learning

Felix Cheung
Principal Engineer & Spark Committer

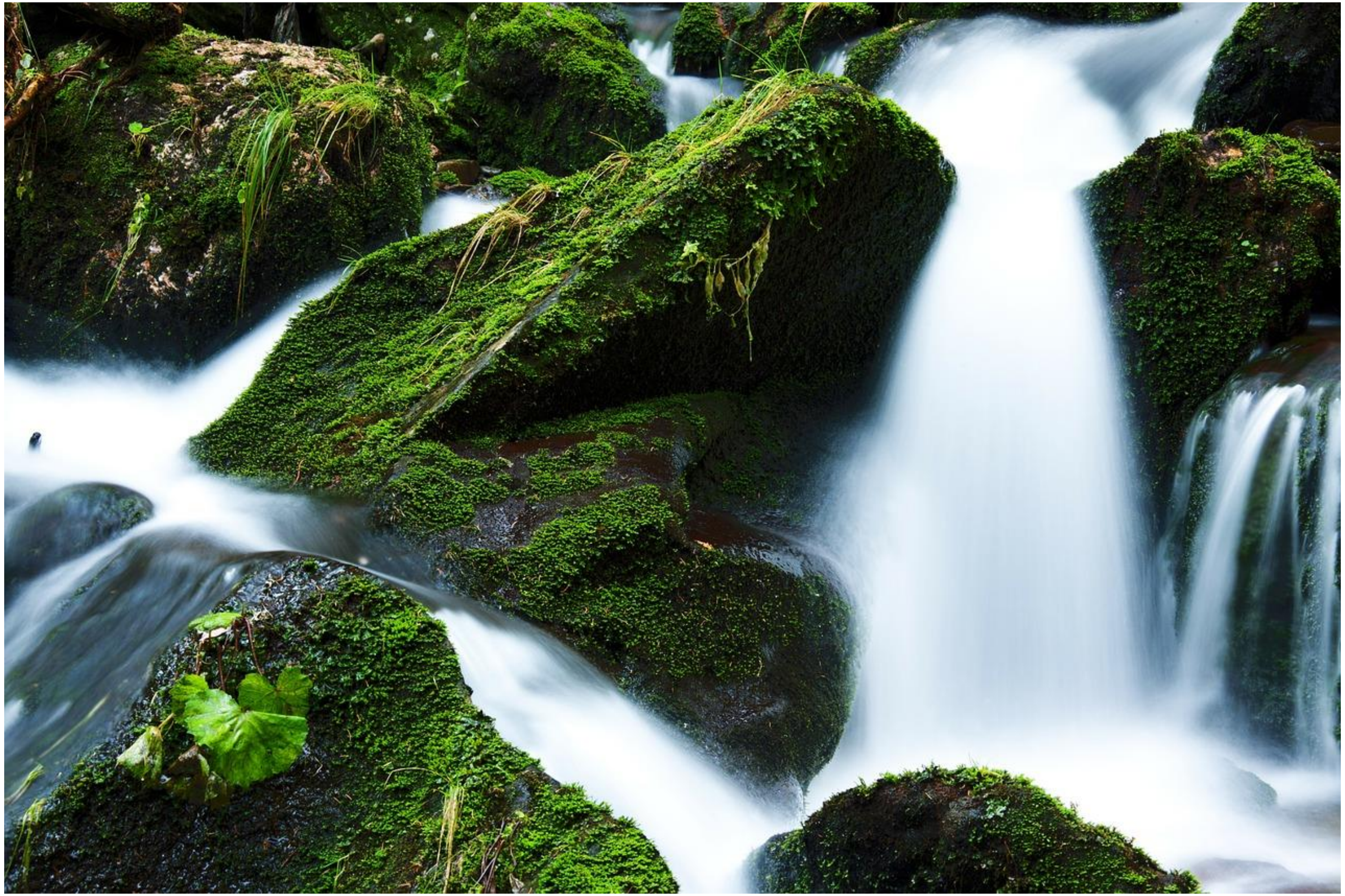
Spark + R

Disclaimer:

Apache Spark community contributions

Agenda

- Structured Streaming
- ML Pipeline
- R - putting it all together
- Considerations



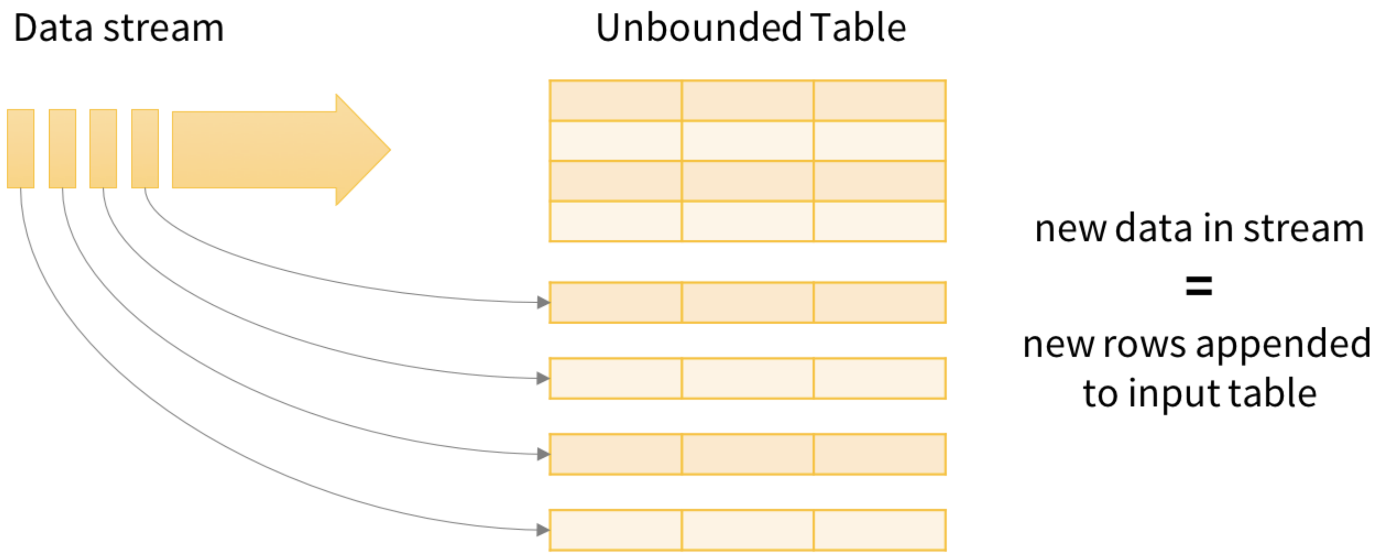
Why Streaming?

- Faster insight at scale
- ETL
- Trends
- Latest data to static data
- Continuous Learning

Spark Streaming

1. Receiver
2. Direct DStream
3. Structured Streaming

Structured Streaming



Data stream as an unbounded Input Table

https://databricks.com/blog/2016/07/28/structured-streaming-in-apache-spark.html

Structured Streaming

- "Streaming Logical Plan"
  - Extending Dataset/DataFrame to include *incremental* execution of unbounded input
  - aka Rinse & Repeat

SPARK SUMMIT 2017

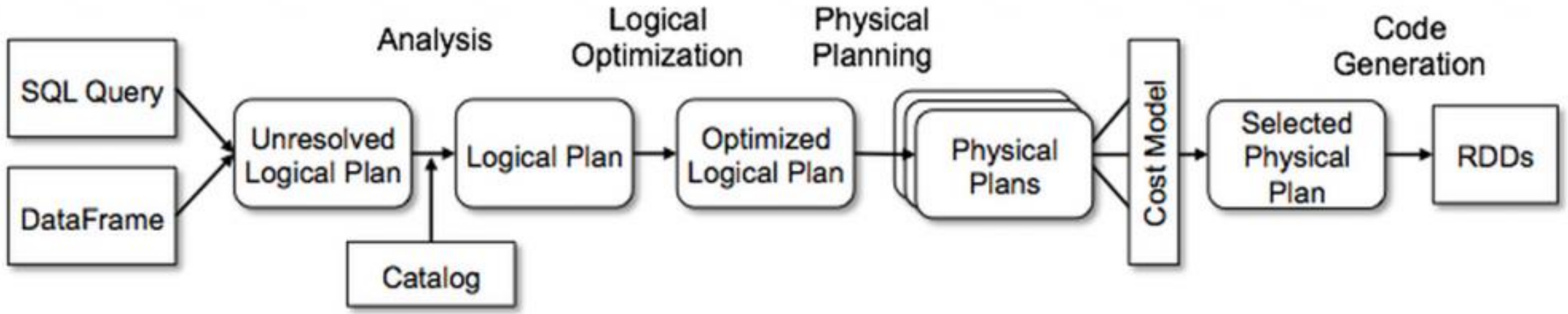
Same

- Transformations:
  map
  filter
  aggregate
  window
  join* (*some limitations)

Better

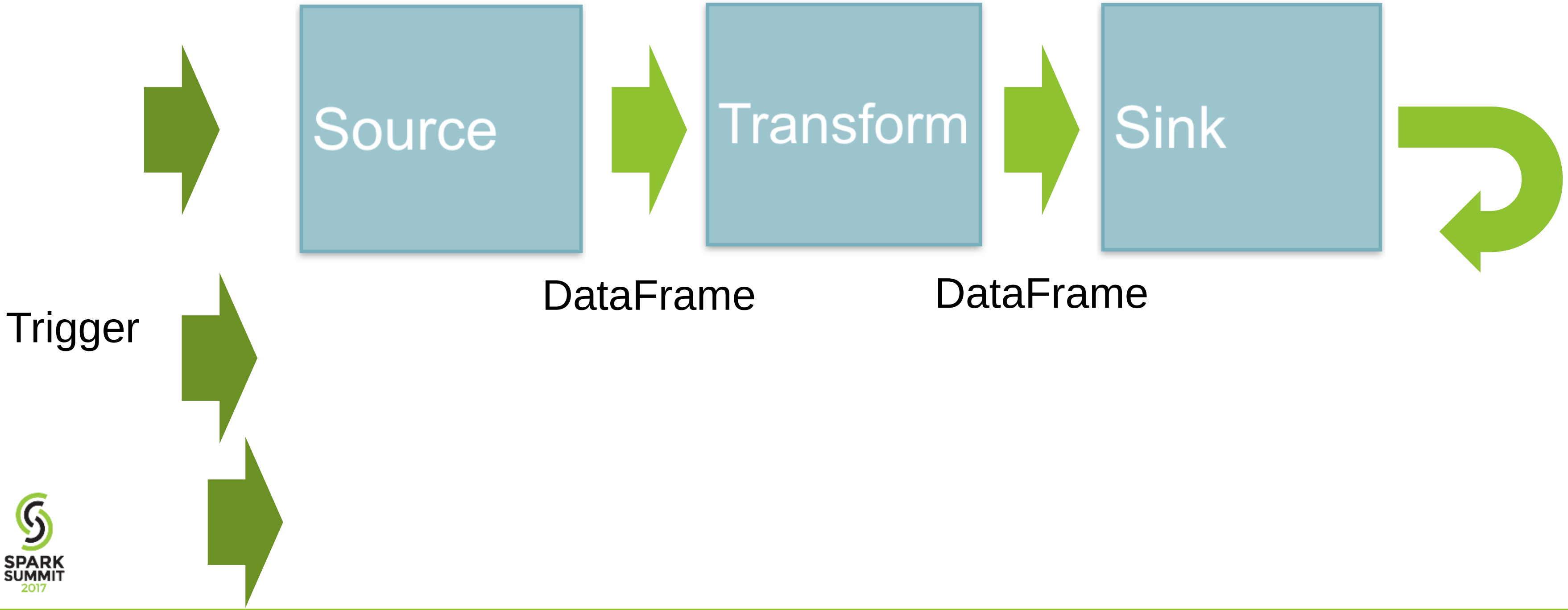
- Trigger
- Consistency
- Fault Tolerance
- Event time – late data, watermark

Execution Plan



https://databricks.com/blog/2015/04/13/deep-dive-into-spark-sqls-catalyst-optimizer.html

SS in a Circuit



Source

File
Kafka
Socket
MQTT

Sink

File (new formats in 2.1+)
Console
Memory (aka Temp View)
Foreach
Kafka (new in 2.2)

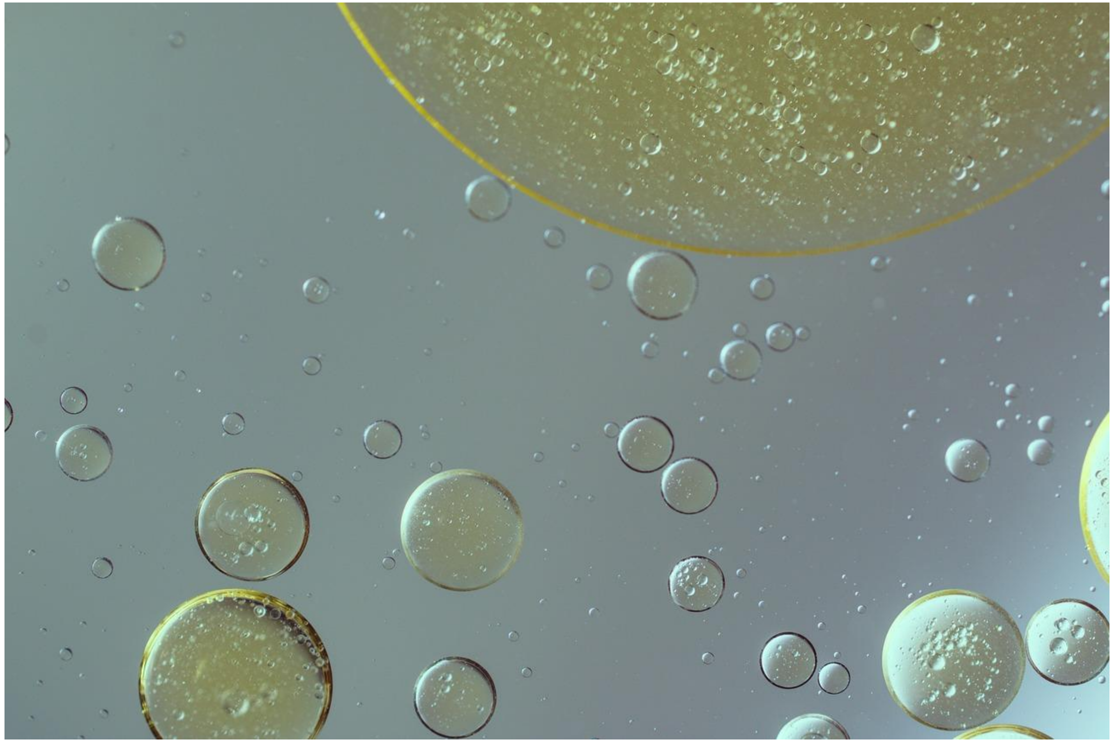
Output Mode

Append (default)

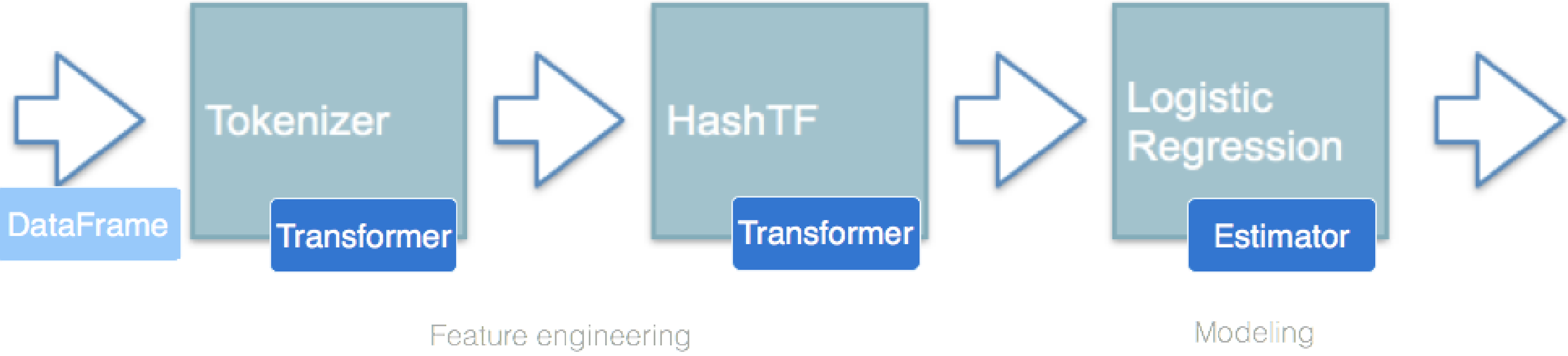
Complete

Update (new in 2.1.1)

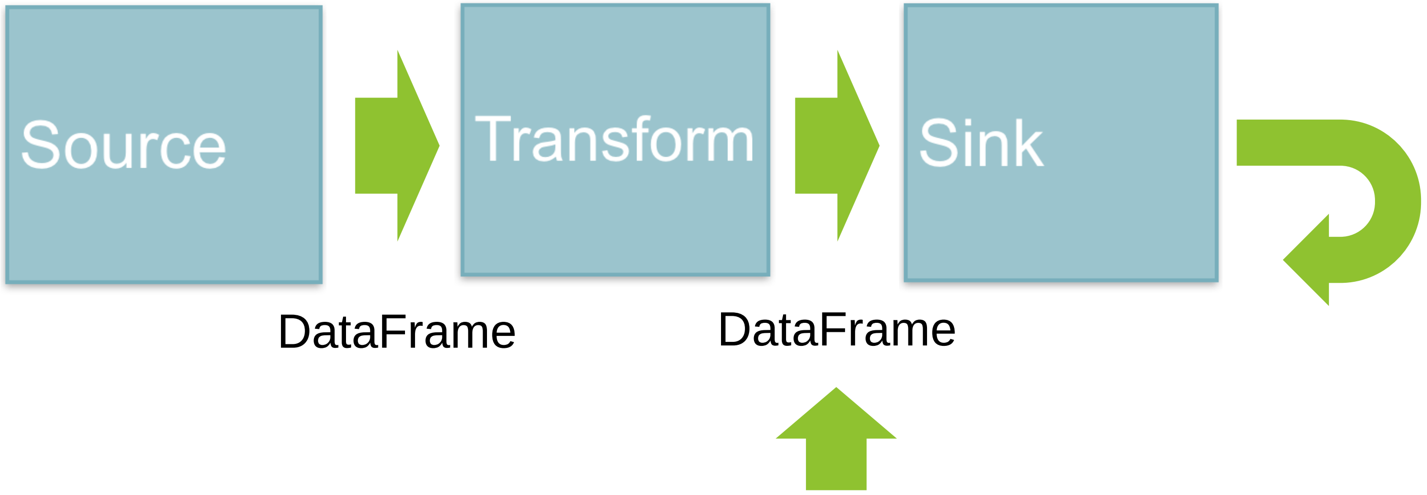
Streaming & ML Don't Mix*



ML Pipeline Model



Remember the SS Flow?

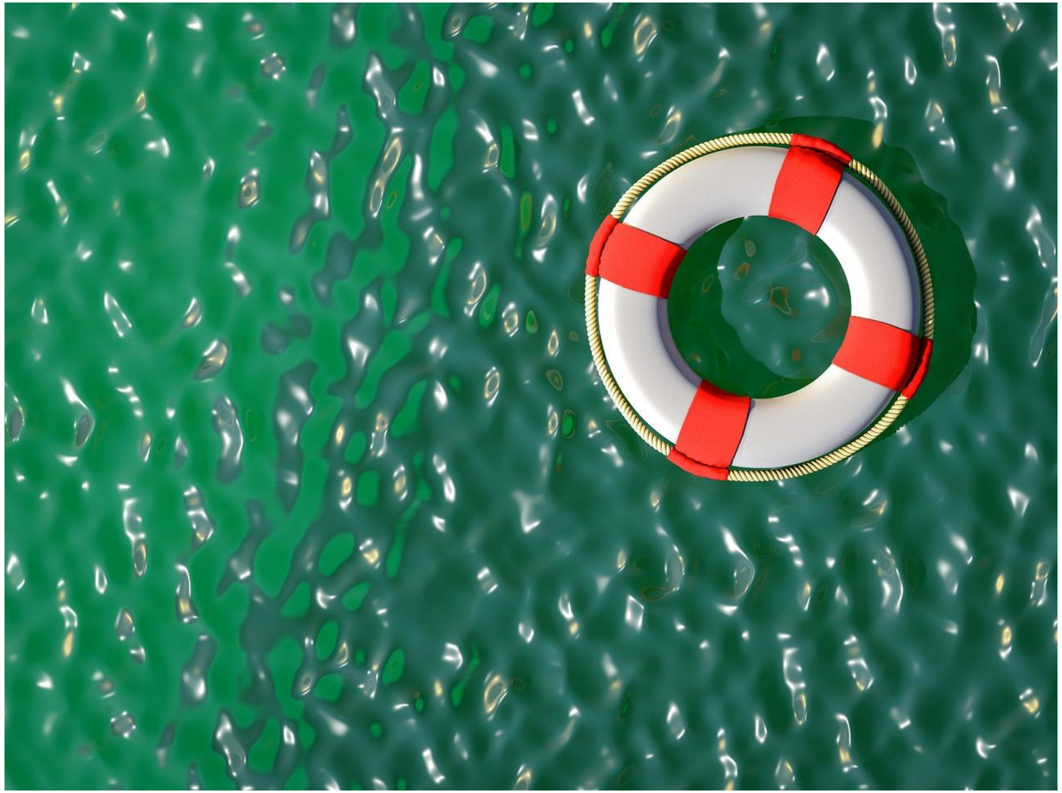


ML Pipeline fit()

- Essentially an Action
- Results in a Model

- Sink start() also an Action
- Structured Streaming circuit must be completed with Sink start()

R to the Rescue



R

- Statistical computing and graphics
- 10.7k+ packages on CRAN

Why Streaming in R

- Single integrated job for everything
  1. Ingest
  2. ETL
  3. Machine Learning
- Use your favorite packages - freedom to choose
- rkafka – last published 2015

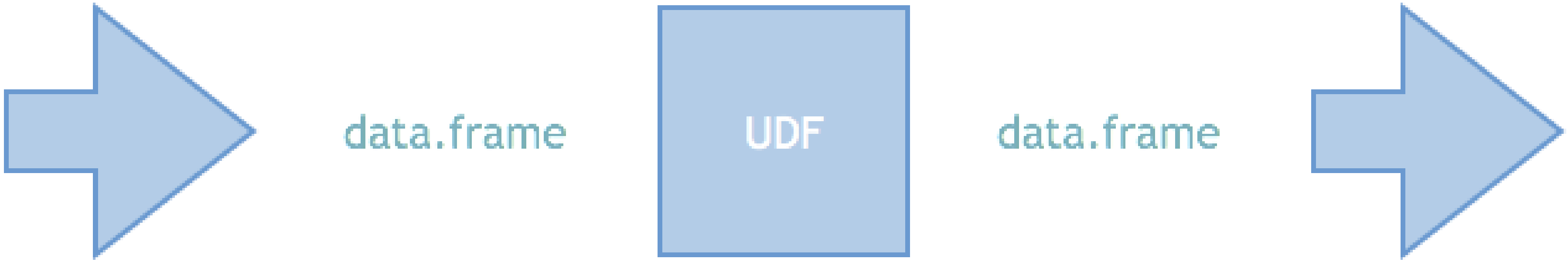
Spark + R

SparkR

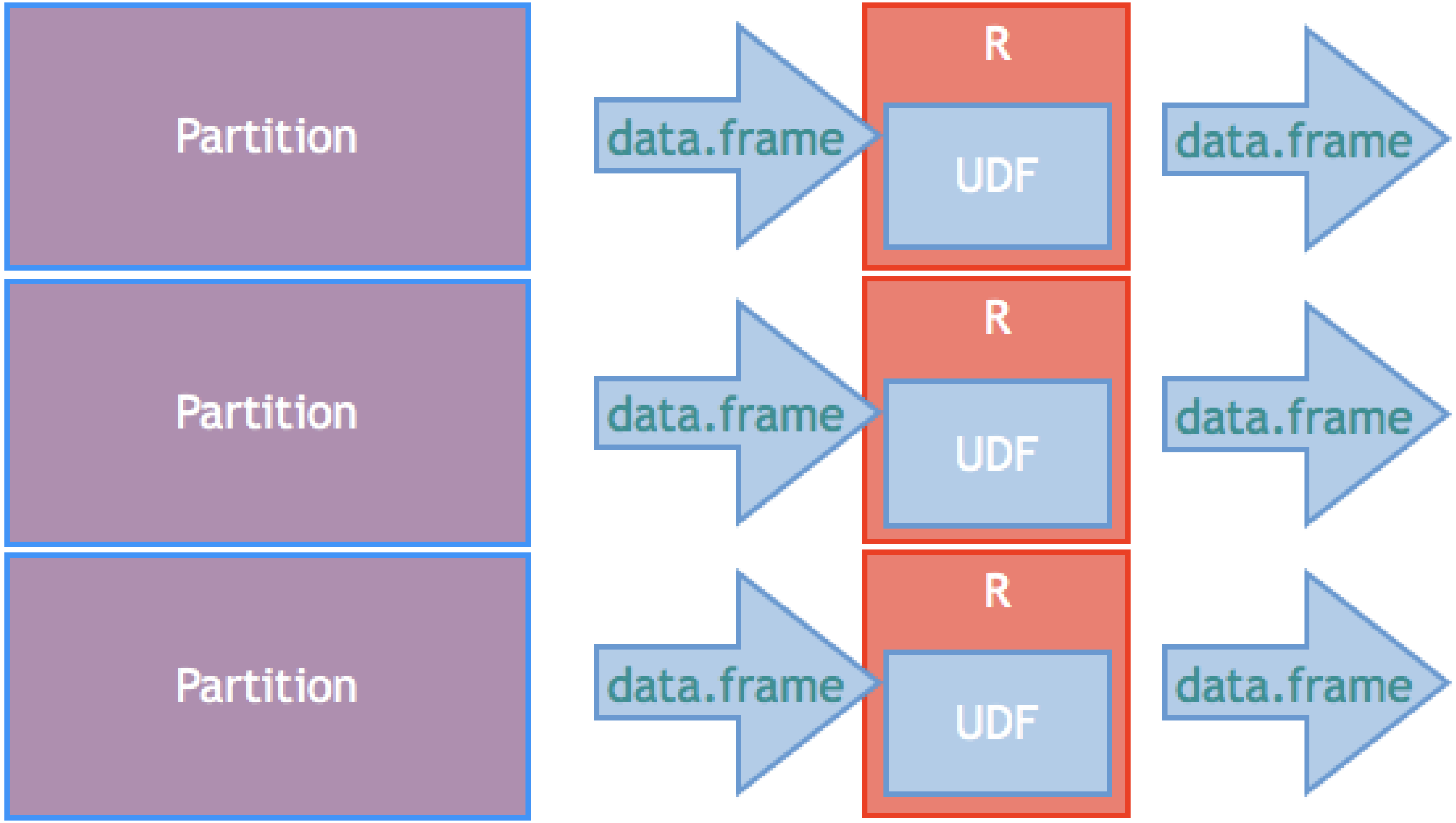
- DataFrame API like R data.frame, dplyr
  - Full Spark optimizations
- SQL, Session, Catalog
- “Spark Packages”
- ML
- R-native UDF
- SS

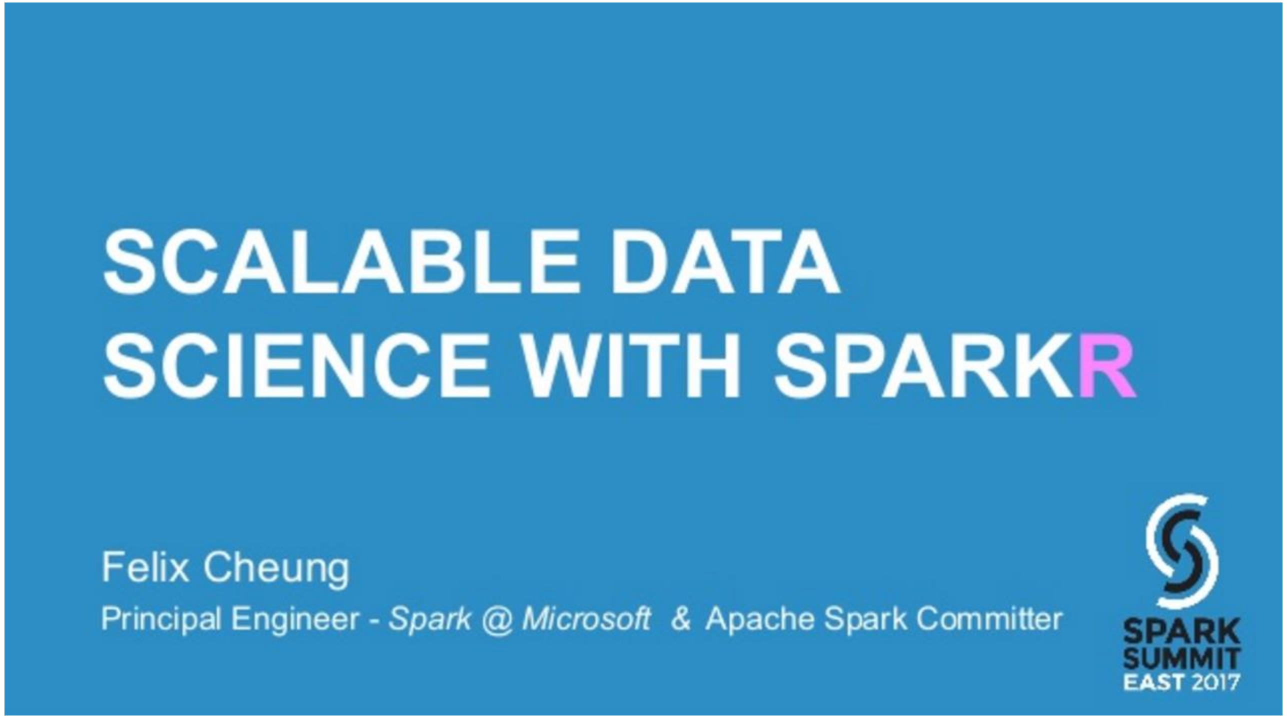
Native R UDF

- User-Defined Functions - custom transformation
- Apply by Partition
- Apply by Group



Parallel Processing By Partition





SCALABLE DATA SCIENCE WITH SPARKR

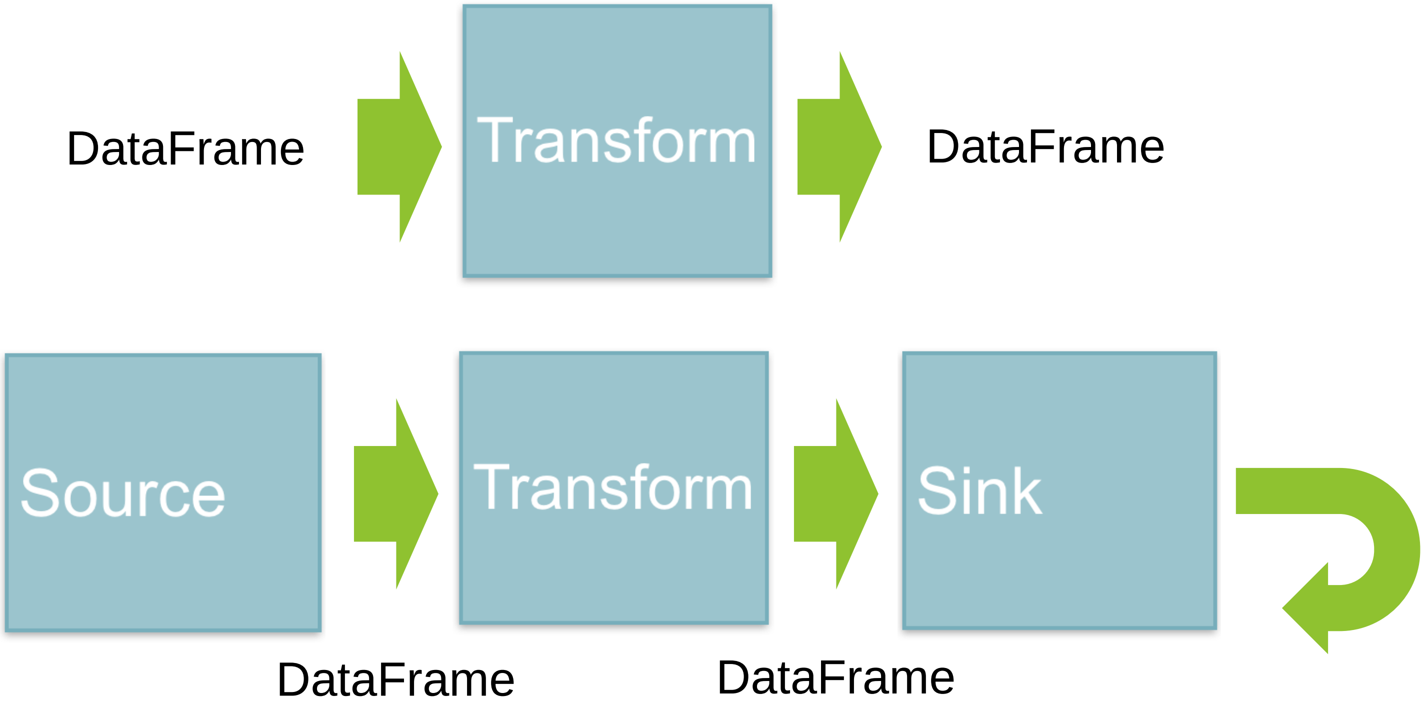
Felix Cheung

Principal Engineer - *Spark @ Microsoft* & Apache Spark Committer

SPARK SUMMIT EAST 2017

https://spark-summit.org/east-2017/events/scalable-data-science-with-sparkr/

Native R UDF = DF Transform



SS in R

1. DataStreamReader/Writer
2. StreamingQuery
3. Extending DataFrame (`isStreaming`)

About Demo

- Create a job to discover trending news topics
  - Structured Streaming
  - Machine Learning with native R package in UDF

Demo!

https://goo.gl/0v6YxF

Demo

- SS – read text stream from Kafka
- R-UDF – a partition with lines of text
  - – RTextTools – text vector into DTM – scrubbing
  - – LDA
  - – terms
- SQL – group by words, count
- SS – write to console

Read DataFrame vs Stream

```
read.df(datapath, source = "parquet")

read.stream("kafka",
            kafka.bootstrap.servers = servers,
            subscribe = topic)
```

Streaming WordCount in 1 line

```
library(magrittr)
kbsrvs <- "kafka-0.broker.kafka.svc.cluster.local:9092"
topic <- "test1"

read.stream("kafka", kafka.bootstrap.servers = kbsrvs, subscribe = topic) %>%
selectExpr("explode(split(value as string, ' ')) as word") %>%
group_by("word") %>%
count() %>%
write.stream("console", outputMode = "complete")
```

SPARK SUMMIT 2017

Challenges

Streaming and ML

- Streaming – small batch
- ML – *sometimes* large data to build model
  => pre-trained model
  => online machine learning
- Adopting to data schema, pattern changes
- Updating model (when?)

Practical Implementation

- LSI – online training
- Online LDA
- kNN
- k-means with predict on new data

SS Considerations

- Schema of DataFrame from Kafka:
  key (object), value (object), topic, partition, offset, timestamp, timestampType
- OutputMode requirements

ML with R-UDF

- Native code UDF can break the job
  - eg. ML packages could be sensitive to empty row
  - more data checks In Real Life
- Debugging can be challenging – run separately first
- UDF must return that matches schema
- Model as state to distribute to each UDF instance

Future – SSR

- Configurable trigger
- Watermark for late data

SPARK SUMMIT 2017

Thank You.

https://github.com/felixcheung
linkedin: http://linkd.in/1OeZDb7
blog: http://bit.ly/1E2z6OI