DATA FELLAS

DISTRIBUTED DATA SCIENCE

O'Reilly Online Training on Integrated Data Science Pipeline, given by

Andy Petrella, Founder **Xavier Tordoir, Founder**

1-3 March, 2016

Streaming Data

In the flow

Data as a stream

With distributed systems, you want to work with immutable data representations (no update)

But it is very likely that the data you want to exploit is not static

Thus most dataset can be viewed as the accumulation of data streams

e.g. logs, events

Real time computing?

What makes a system real-time?

Its total correctness depends also on the time in which operations are performed. Real-time systems have 3 possible levels:

- **Hard: a missed deadline is a total system failure**
- **Firm: few missed deadlines are tolerable but may degrade to overall quality of service, results produced after the deadline are not useful at all**
- **Soft: Results usefulness degrade after deadline**

Real time computing?

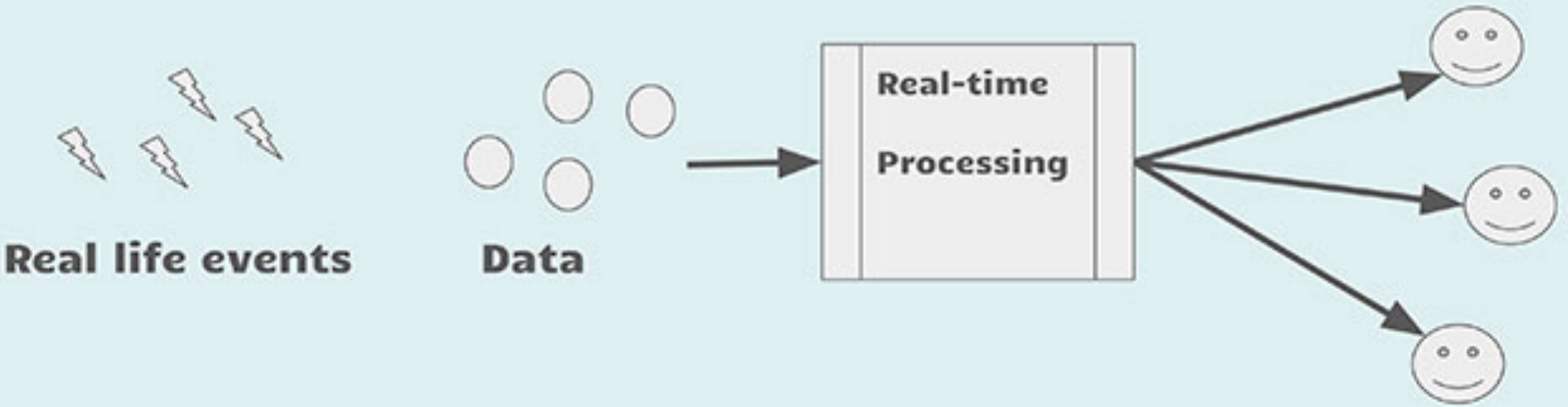
If you really need hard real-time computing, then it is very likely that:

- **the computation engine will be embedded**
- **dedicated resources are not shared with other processes**
- **the consumer of the result is unique and permanently wired**

Doesn't sound like the kind of applications deployed on distributed systems with their inherent latency (network, IO, cpu), and flexibility with data consumption (multiple consumers).

Real time computing?

It can be confusing to classify real time computing, from whom's perspective? A single consumer? What about the producer?

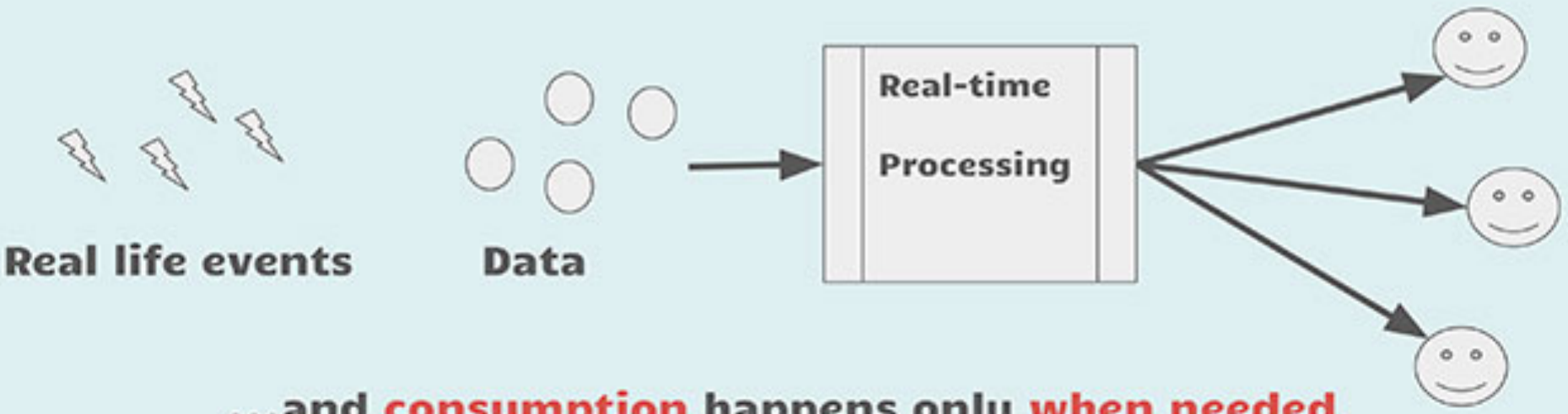


Real time computing?

Non-hard real time service making computations available...

Streaming

Non-hard real time service making computations available...



...and consumption happens only when needed

Streaming

A stream is a sequence of data to be made available over time.

- **potentially unlimited**
- **functions applied to stream are different from batch ones (e.g. how to average a value in a stream...)**

Spark Streaming

Microbatching at scale

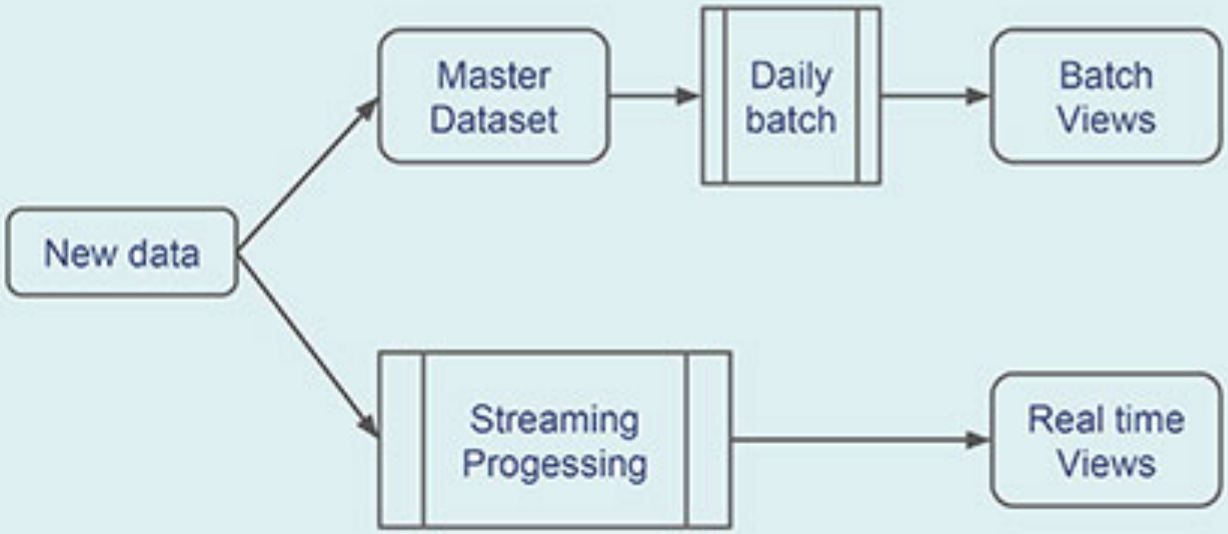
Why batch?

It can be an advantage to process events immediately at arrival (event processing)

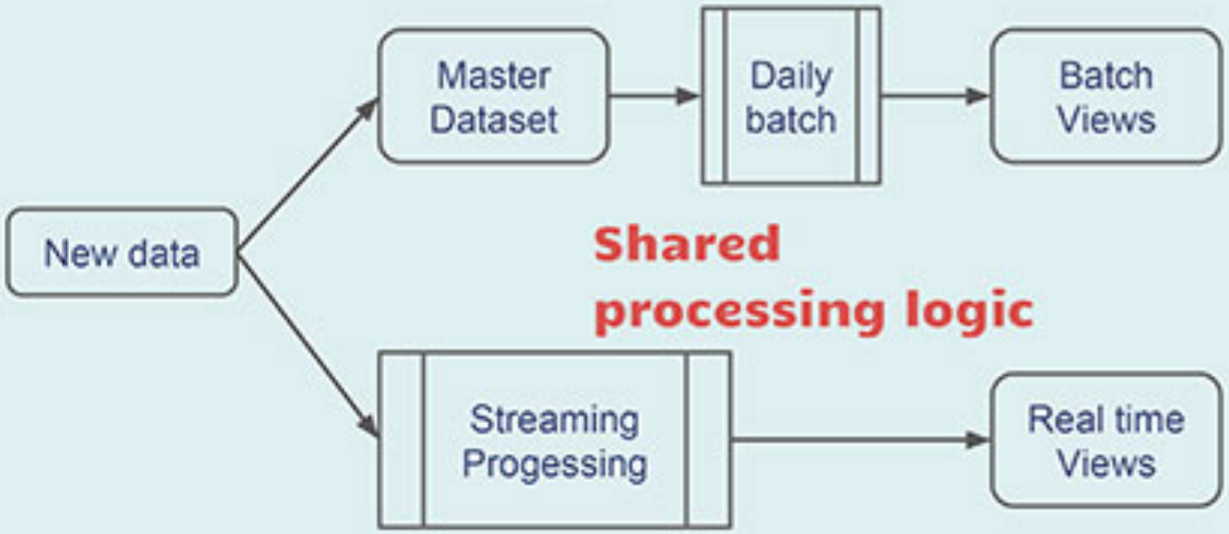
BUT, allocating computing resources to a single event can be costly while the deadlines requirements may not be that high

Spark is a batch processing engine, Spark Streaming leverages the batch engine, hence the code written for batch can be reused (cf lambda architecture)

lambda architecture



lambda architecture



Spark Streaming

Resilient Distributed Datasets (RDDs)*

- **Define computations on Partitionned Datasets**
- **Very similar to Collection API**

Discretized Stream (DStream)

- **Infinite sequence of RDDs**
- **Hence same API as RDDs for streaming transformations**
- **each batch stored as an RDD**
- **fixed time interval per batch (0.5 - 60+ sec)**

*See Big Data Analytics with Spark by Mohammed Guller

Or Advanced Analytics with Spark by Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills

Spark Streaming

Discretized Stream (DStream)

*Spark Streaming Documentation: http://spark.apache.org/docs/latest/streaming-programming-guide.html

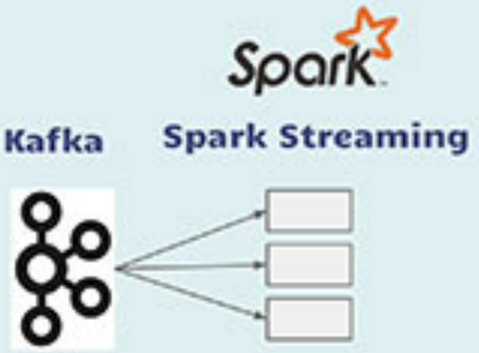
Spark Streaming

*Spark Streaming Documentation: http://spark.apache.org/docs/latest/streaming-programming-guide.html

Consume Streaming

Notebook!

Consume streaming



In-Memory Data

Storage, fat and fast

In-memory

Data is consumed, results are produced

- **These results will be consumed**

Do we know in advance all use of these results?

NO, hence, we need flexibility because exposing all potential results or transformations from the original data is not possible with fast or large datasets

Hence the need to prepare views preparing some work for further detailed/specialized analysis, these views should be fast

Cassandra

Distributed database

Cassandra

Cassandra is a distributed column-oriented NoSQL Database

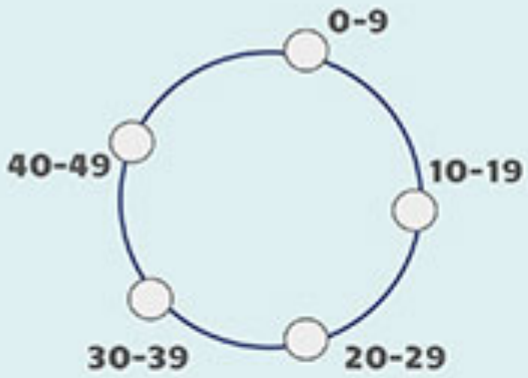
(See Google Big Table, Amazon Dynamo)

- **Continuous availability and resilience / Multi-site**
- **Linear scale: adding nodes for performance and size**
- **Works with commodity hardware, on-premise or in the cloud**
- **True peer-to-peer architecture (one node type)**

Key - Hashing

Data is addressed with keys

Consistent key hashing allow every server to know where to find data

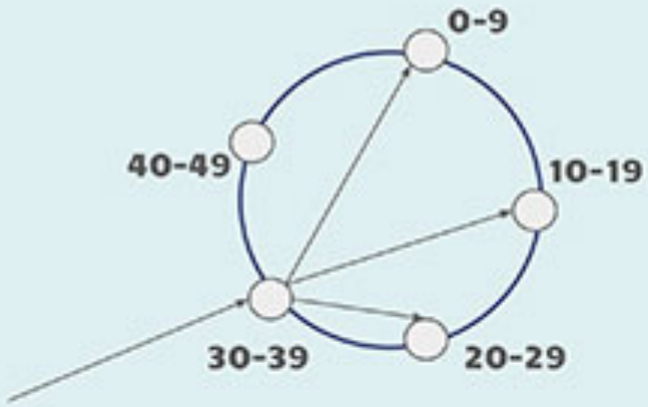


Replication & consistency

Data is replicated with network topology awareness

Consistency tunable

- **one**
- **quorum**
- **all**



Keys

Partition key/cluster key

```
CREATE TABLE quotes (
  symbol text,
  ts timestamp,
  price: double
  PRIMARY KEY (symbol, ts)
);
```

key **columns**

symbol	ts:9:00	ts:2	ts:3	ts:4	ts:5
BAC	10.01	10.03	9.98	9.87	9.78

Efficient at write because data is written in sequence, and with range queries because columns are sorted, the cluster knows which nodes to query to get a complete range!

Storing the stream

Notebook!

Storing the stream

Data analysis

Statistics, ML & co

Data analysis

Hey you want to do something with this data...

Model a facet of the data to answer specific questions like:

- **What is the chance we lose a customer in the next period?**
- **What is the best recommendation we can give?**
- **Is there an arbitrage in the current price of an asset?**
- **What is the chance for a transaction to be fraudulent?**

Data analysis

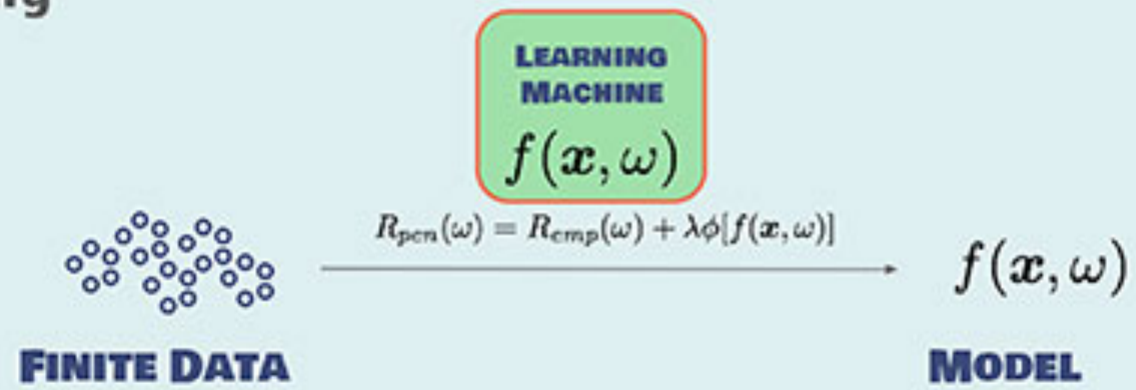
To answer these questions we need several things:

- **Access to data**
- **Data cleaning tools**
- **Data transformation tools (e.g. features engineering)**
- **Descriptive statistics**
- **Machine learning (training models and predicting)**

All this better be done within a single interactive environment,...

Machine learning in a nutshell

Machine Learning is the process of learning a function from the data only



The learning process is an optimization procedure to minimize expected risk (error) on unseen samples

Complexity control

An infinitely complex function could perfectly fit any finite training sample

BUT it would capture specifics of the dataset (noise) and not more general features of the process generating the data

This is overfitting

We avoid this by setting contraints on the function complexity

Complexity control

A too simple function is not able the fit characteristics of the generated data

This is underfitting

Procedures to have model complexity in balance are available, for example regularization

A penalty on complexity

Tuning the penalty can be done with resampling

Statistics and Machine learning at scale

New (and old) algorithms are designed to compute statistics and Machine Learning models efficiently on distributed datasets.

They are based on partial computations on batches (partitions) that can be aggregated in a single statistics (model)

We need this because we want to work on the dataset, not on a sample in a local mode with lots of data transfer and information loss risk

Spark and MLLib

ML at scale

MLLib - data structures and linear algebra

Distributed implementation of common Machine Learning algorithms

Data represented as ADDs

Individual samples (row) use the LabeledPoint (for supervised learning) type and Breeze library Vectors

Manipulation of disributed matrices

All the required linear algebra...

MLLib - Basic statistics

Summary statistics (mean/variance), Correlations

Hypothesis testing

Kernel density estimation

Of course on simple data (single Double values)

MLLib - Machine Learning

Classification and regression

- **Linear models, naive bayes, trees and random forests**

Collaborative filtering (for recommendation systems)

Feature extraction and transformation

- **NLP routines (TF-IDF, Word2Vec)**
- **Scalers**
- **PCA**

Other Machine Learning

Other ML libraries

Distributed Machine Learning

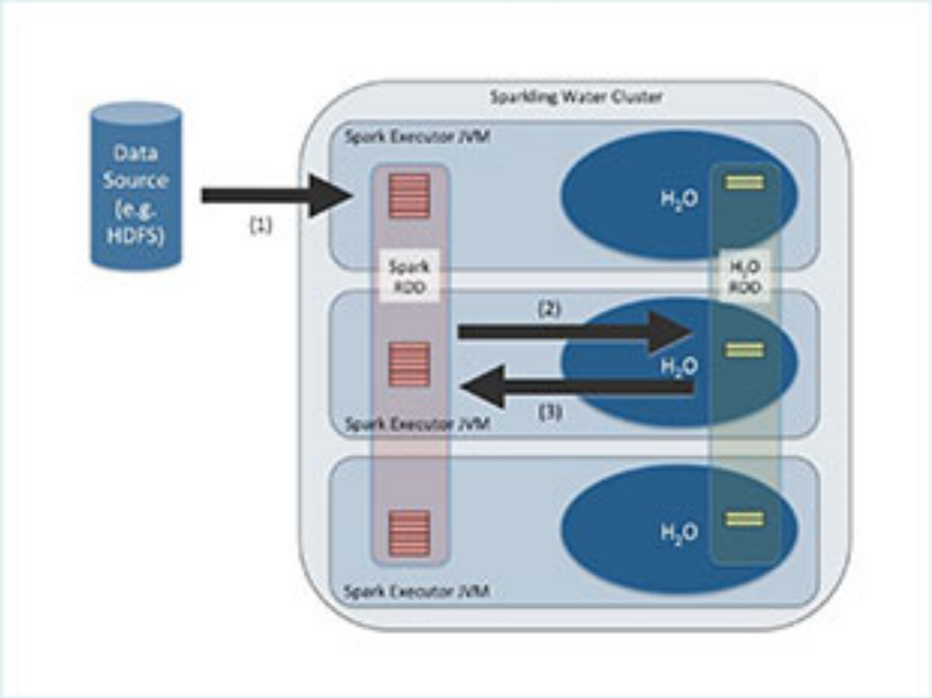
Many libraries for distributed machine learning are popping

Some are compatible with Spark:

- **Spark ML Pipeline (part of Spark itself, working with Dataframes and chaining operations)**
- **ML Keystone by Amplab (typesafe chainable API)**
- **H2O: memory implementation of map-reduce/Machine Learning**
- **DL4J: deeplearning by skymind**
- **DAAL: Linear algebra by Intel**
- **Needle: Deep learning by Nitro**

H2O integration architecture

Based on in-memory data exchange between Spark and H2O



The Model

Notebook!

Our Model

We have timeseries data for a bunch of stocks

These timeseries certainly correlate and one can be inferred from others

We want a model for pricing JPM from the others, thus building a "proxy" for JPM stock price

We'll use supervised learning from the complete database

A linear regression model should make the cut...

Analyse Stock Market: Concretely