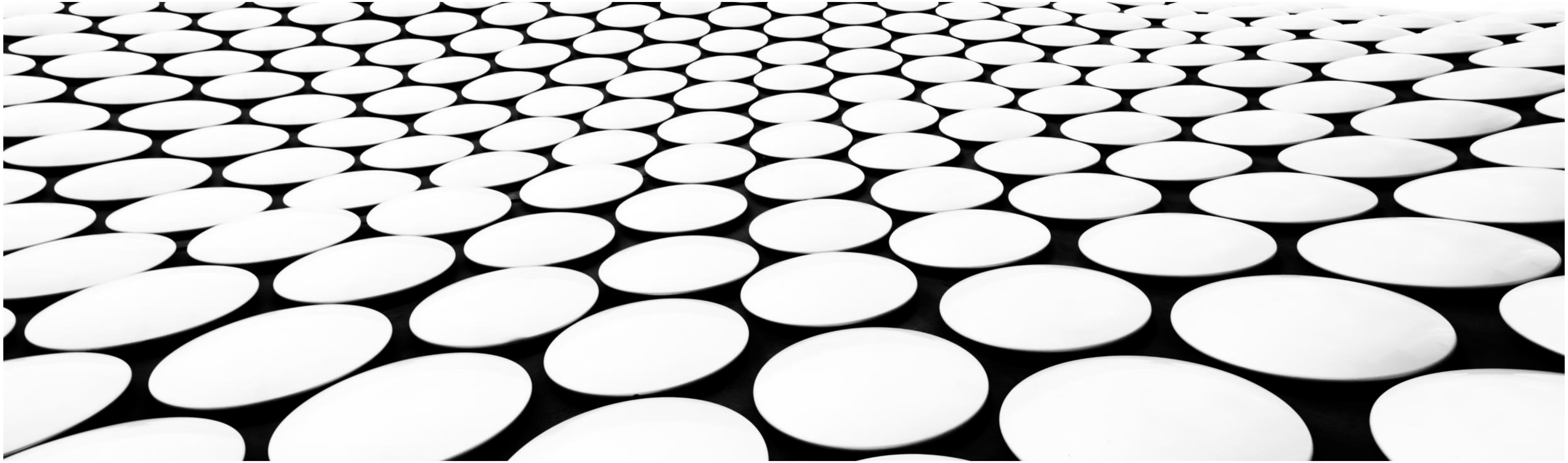
A BEGINNER'S OVERVIEW OF SPARK AND PYSPARK

SAN DIEGO MACHINE LEARNING

MAY 30, 2020



AGENDA

- Provide an overview of what Spark is and why you would use it
- Discuss architecture and key features
- Not a PySpark programming guide
- List some resources to learn more

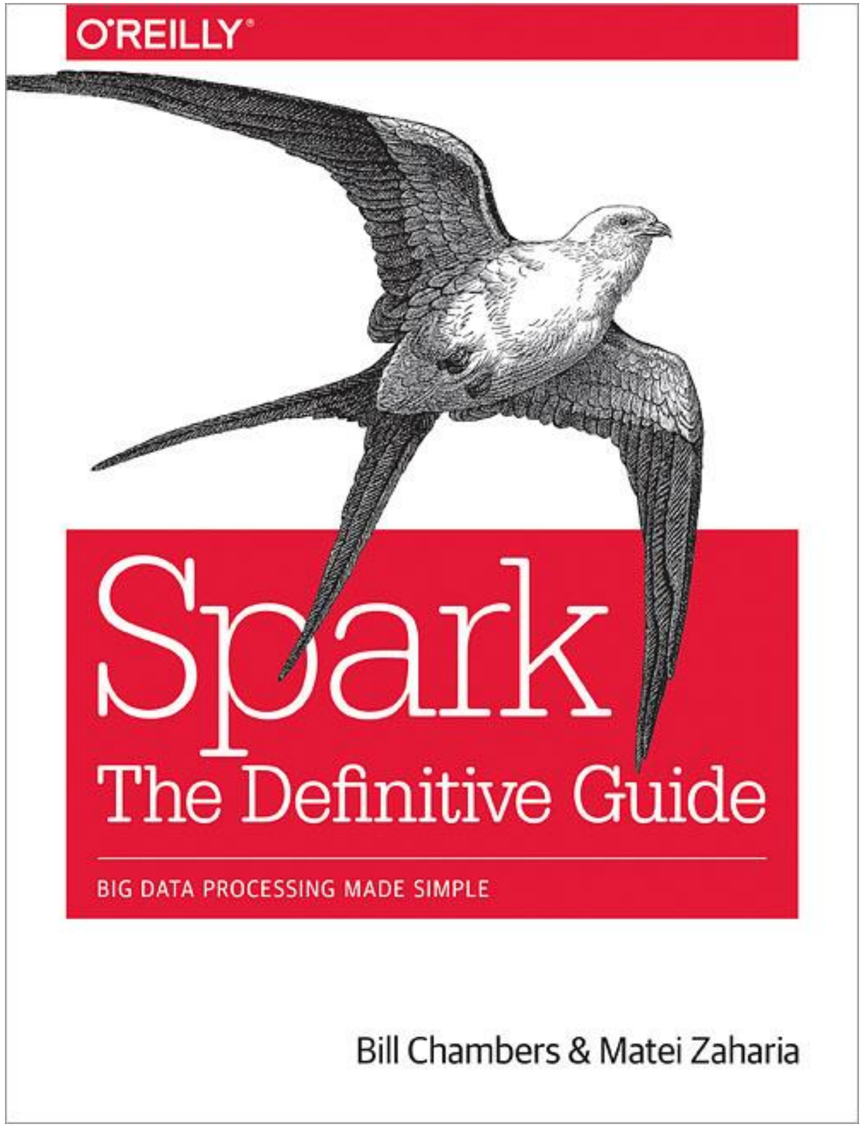

O'REILLY
Spark
The Definitive Guide
BIG DATA PROCESSING MADE SIMPLE
Bill Chambers & Matei Zaharia

*Figure from http://shop.oreilly.com/product/0636920034957.do

WHAT IS SPARK

- Apache Spark is an open source general-purpose framework for cluster computing
- History:
  - Originally developed at Berkeley, starting in 2009
  - Contributed to Apache in 2013, and founded Databricks
  - Version 1.0 in 2014. Spark 2.0 in 2016.
- Spark is about computing, not storage or other functionality
- Written in Scala, runs on JVM
- Pros: fast; real-time; powerful caching; language support for Java, Scala, Python, R, and SQL
- Built-in fault tolerance and parallelism

WHY USE SPARK

- Fast, big data computing
- Plays well with HDFS
- Streaming
- Not for multi-user
- MapReduce was inefficient for iterative or interactive jobs
  - According to authors, MapReduce required multiple passes over the data, and multiple batch jobs
- Easy to use

SPARK ARCHITECTURE

- Master node and compute nodes, called driver and executors
- Cluster manager, either built-in, YARN or Mesos
- Everything done in memory – extremely fast
- Functional programming

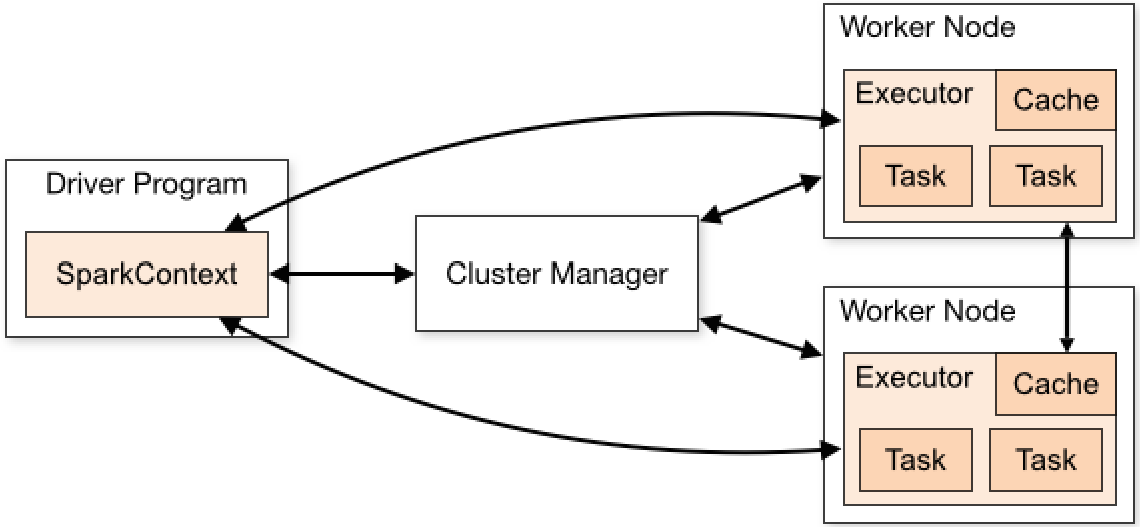


*Figure from https://spark.apache.org/docs/latest/cluster-overview.html

SPARK ARCHITECTURE

- High level data structures includes DataFrames. DataFrames are made up of partitions.
- Low level data structures are RDDs (Resilient Distributed Datasets)
- Core data structures are immutable, so we use transformations
  - Narrow transformations, within partitions
  - Wide transformations require shuffles

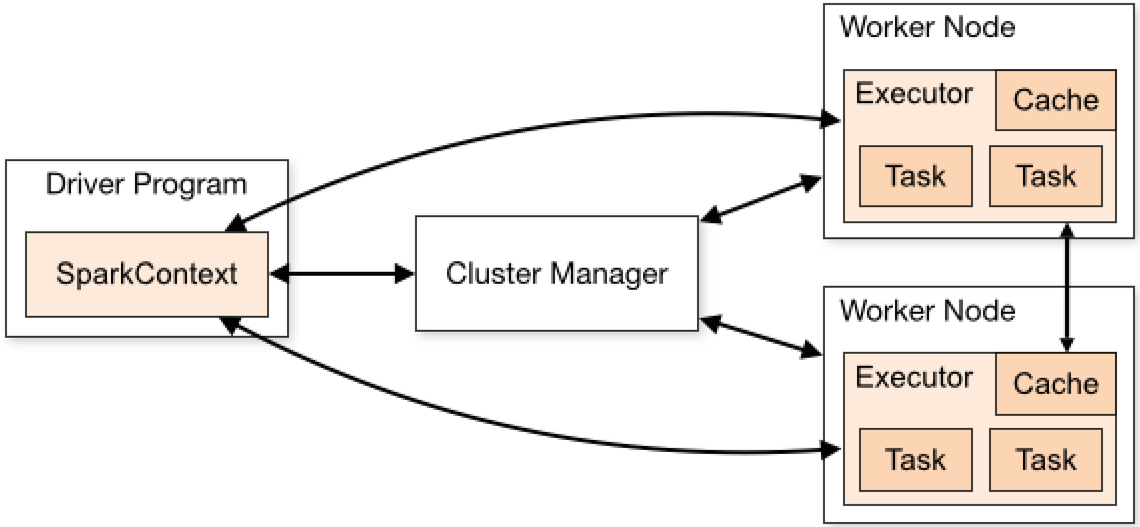


*Figure from https://spark.apache.org/docs/latest/cluster-overview.html

KEY FEATURES

- Low level:
  - RDDs
  - Lazy evaluation, and predicate pushdown
  - Catalyst, cost based optimizer
- High level:
  - DataFrames and Datasets
  - Spark Streaming
- Libraries
  - Spark SQL
  - MLlib
  - Spark NLP

LOW LEVEL FEATURES - RDDS

- RDDs
  - In-memory
  - Fault tolerant
  - Partitioned
  - Immutable
- Can be multiple partitions per worker node
- Often partitioning strategy is matched to data source

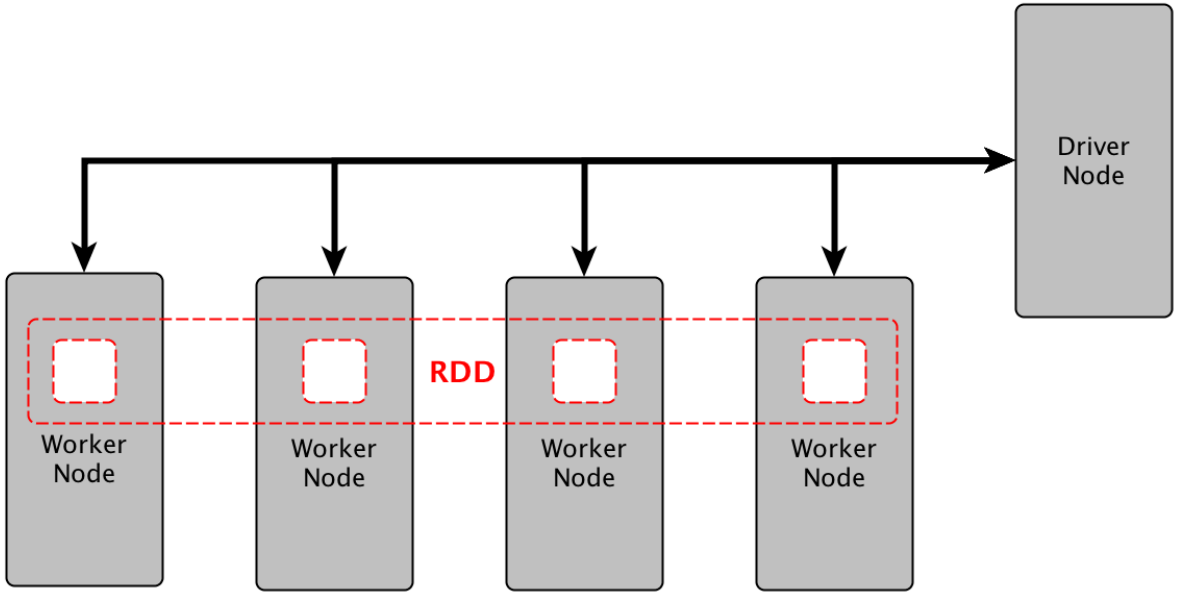
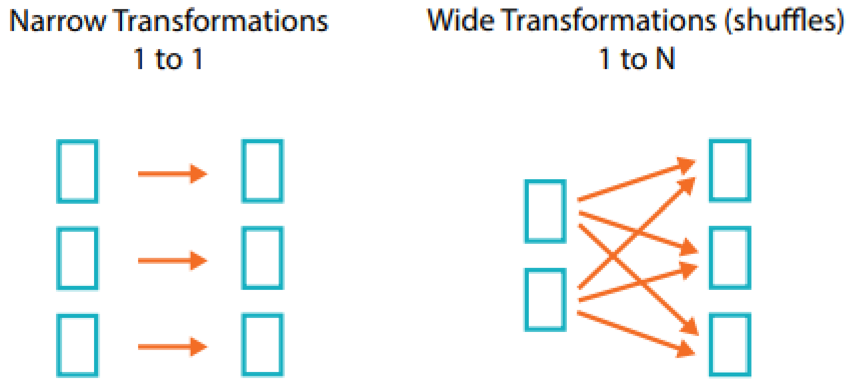


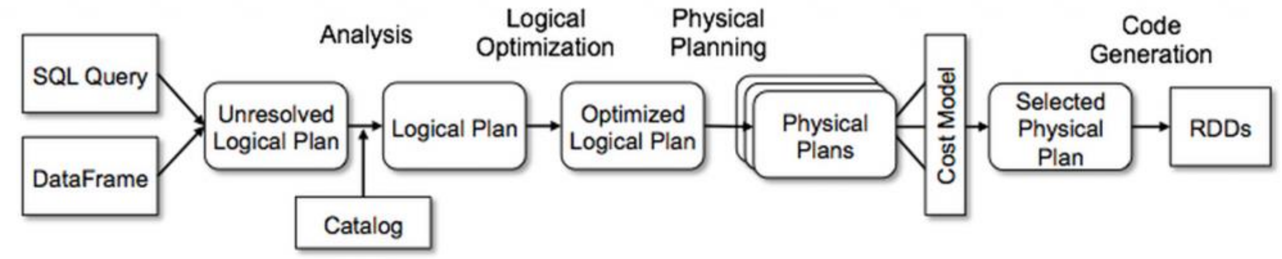
*Figure from https://medium.com/%40lavishj77/spark-fundamentals-part-2-a2d1a78eff73

LOW LEVEL FEATURES - COMPUTATION

- Computation graph is a tree/DAG
- Transformations
  - Narrow: select, filter, sample, map
  - Wide require "*shuffles*": aggregate, join, sort



- Lazy evaluation
- Catalyst, cost based optimizer
  - Optimized logical plan, then physical plan
  - Example: predicate pushdown



- I see many correlations between Spark computation and RDBMS computation

*Figures from https://databricks.com/glossary/what-are-transformations and https://databricks.com/glossary/catalyst-optimizer

HIGH LEVEL FEATURE - DATAFRAMES

- DataFrames
  - Set of rows with defined schema (not a set of columns)
  - Slow when first introduced, but no longer performance penalty
- Datasets
  - Type-safe data
  - Only available in Scala and Java
- Spark Streaming (separate slide)

LIBRARIES AND PACKAGES

- Spark SQL
  - Large subset of ANSI SQL:2003
- MLlib
  - Similar to NumPy and Scikit-Learn
  - Place numeric features into a vector, then call fit() and transform()
- Spark NLP
- GraphX
- Ecosystem of packages, https://spark-packages.org

SPARK STREAMING

- Structured Streaming uses micro-batches
  - Programs for batch and streaming are virtually identical
- Built on top of DataFrames

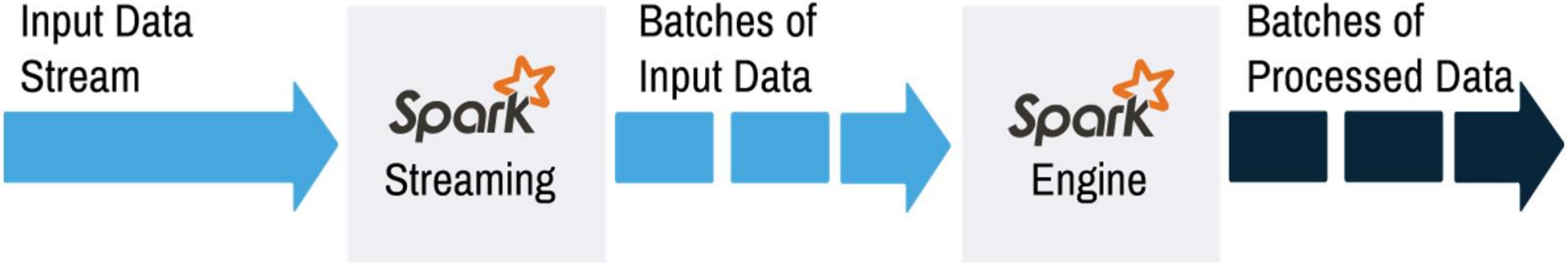


*Figure from https://www.bigdata-toronto.com/2016/assets/getting_started_with_apache_spark.pdf

SPARK STREAMING

- Native support for event time data
- Three output modes:
  - Append
  - Update
  - Complete
- Watermark
  - How late do you expect data to arrive

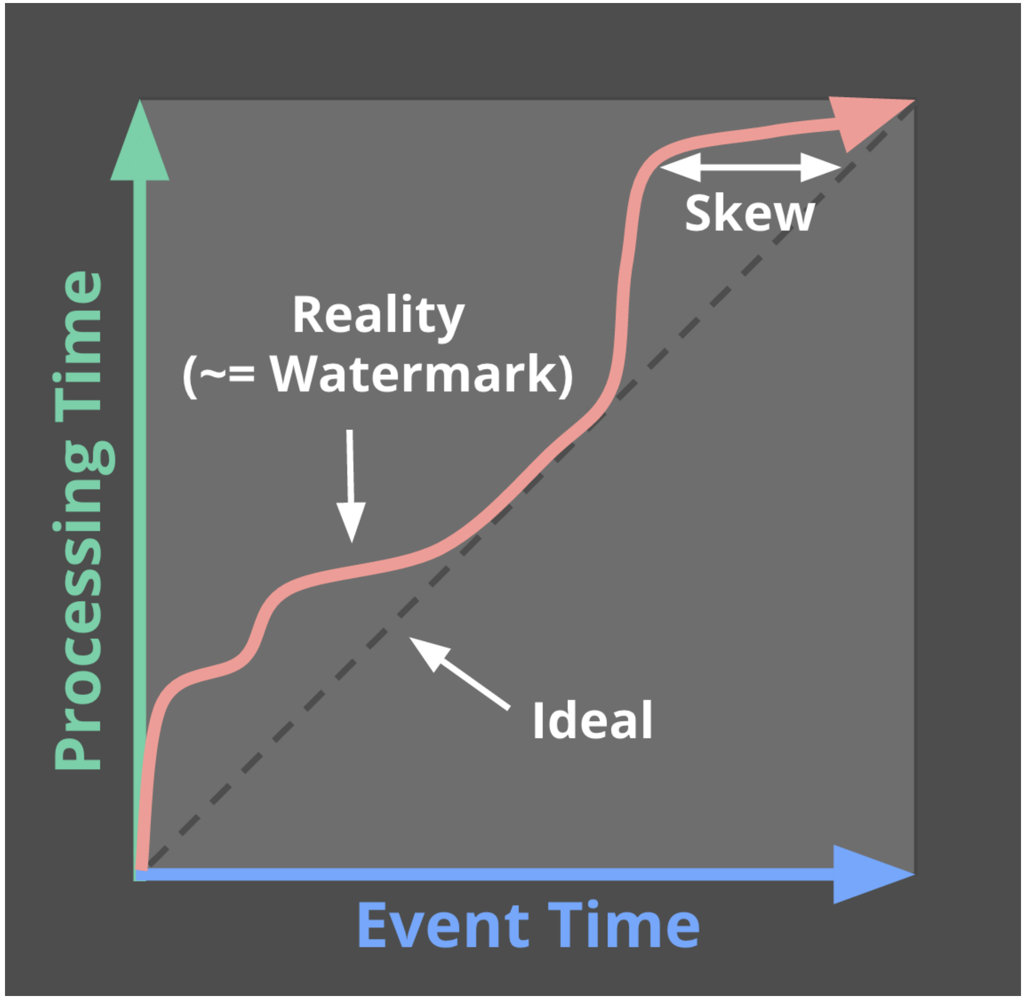


*Figure from https://www.oreilly.com/radar/the-world-beyond-batch-streaming-102/

RESOURCES

- Slides and Python code available at https://github.com/tedkyi/spark_talk
- Spark: The Definitive Guide by Bill Chambers & Matei Zaharia: http://shop.oreilly.com/product/0636920034957.do
- The Apache Spark website documentation: https://spark.apache.org/docs/latest/index.html
- Lots of tutorials and code example on Databricks site: https://docs.databricks.com/getting-started/index.html
- Sign up for your forever free Databricks community account: https://community.cloud.databricks.com

This Photo by Unknown Author is licensed under CC BY-NC-ND