Distributed Processing in the Spark Ecosystem

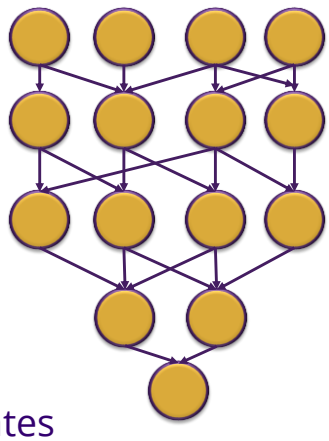
Lesson 10 – Section 3

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Coordinating Data Flow for Job Execution

Facilitating Data Flows

- Hadoop provides a universal framework for writing and executing distributed jobs
- However, a single MapReduce (Hive, Pig, etc.) is rarely sufficient for most use-cases
- Instead – need to coordinate a series of MapReduce jobs
- The primary engineering effort is spent in coordinating a *data flow* for job execution

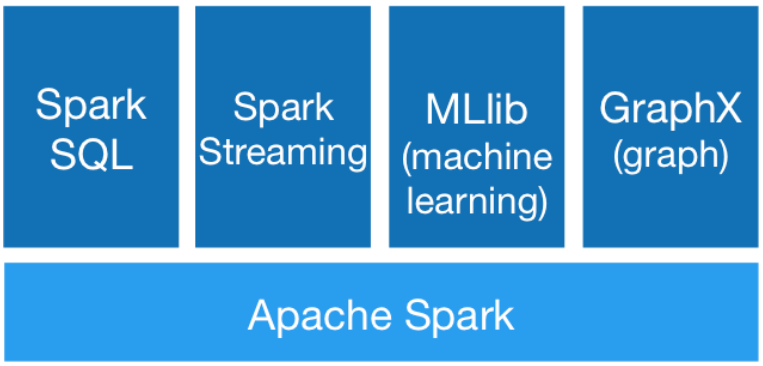


Can we do better than MapReduce, which recreates daemons/readers and writes to disk for each iteration in the data flow? Yes we can—with **Spark**!

Where Spark got it's Spark

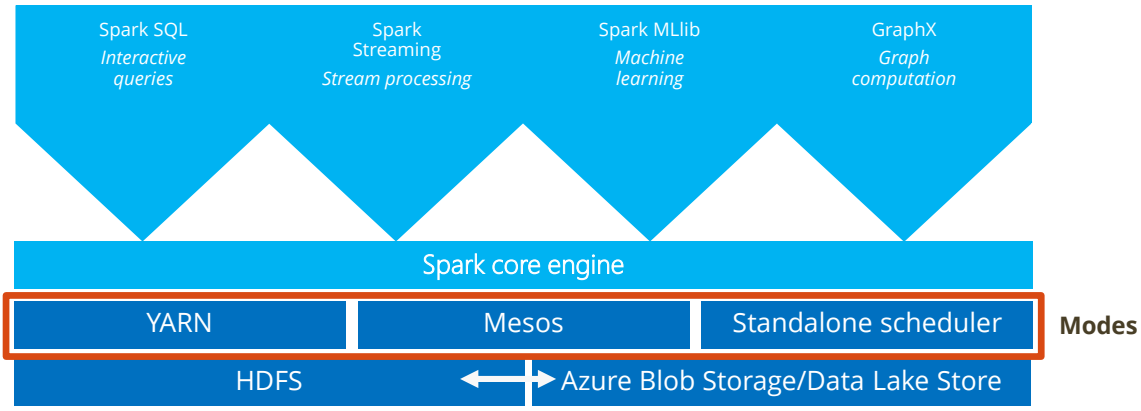
- Started as a research project at UC Berkeley, AMP Lab in 2009, to work on Mesos—a distributed Linux kernel
- Differentiated itself from Hadoop (distributed batch processing) by targeting interactive, iterative workloads like machine learning and data science
- Designed for developers already leveraging Hadoop's HDFS (no storage layer)

Apache Spark: A unified framework



- Goal: unified engine across data sources, workloads, and environments
- Simplified Developer APIs: Scala, Java, Python, R
- Latest Stable Release: http://spark.apache.org/downloads.html

Apache Spark: A unified framework



YARN vs. Mesos – Pooling Resources

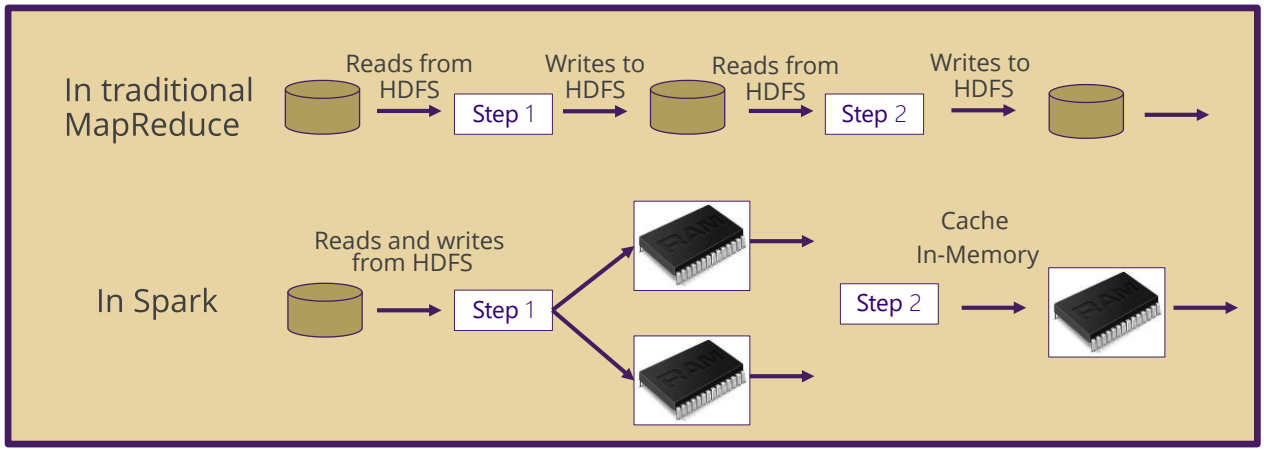
YARN

- Hadoop Cluster Resource Manager
- Memory scheduler
- Linux container groups (more isolation)
- Java-based
- Corporate contributor: Hortonworks

Apache Mesos

- Generic Cluster Resource Manager
- Memory and CPU scheduling
- Simple Unix processes (faster)
- C++ based
- Corporate contributors/Users: LOTS

What makes Spark fast?



DataFrames API – Motivation/Features

- Designed for big data and data science applications
- Inspired by dataframes in R and Python (Pandas) and has APIs for R, Python, Java and Scala.
- Can easily scale from kilobytes of data on a single laptop to petabytes on a large cluster
- Support for a wide array of data formats and storage systems

Spark SQL

Interface for working with structured and semi-structured data.

- –Provides a DataFrame abstraction in Python, Java, and Scala to simplify working with structured datasets
- –Does read/write in a variety of structured formats (e.g., JSON, Hive Tables, and Parquet)
- –Enables SQL querying, both inside a Spark program and from external tools that connect to Spark SQL through standard database connectors (JDBC/ ODBC), such as business intelligence tools like Excel

Spark Streaming

- Provides a way of processing "unbounded" data or continuous data.
- Previously it broke data into micro batches—which could interfere with ordering and timestamps, but has now moved to continuous which has 100x improvement in latency
- New API allows for swapping out for lower latency options such as Apache Storm or Apache Flink

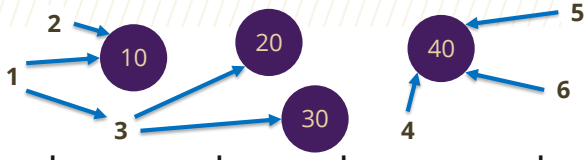
MLlib:

http://spark.apache.org/mllib/

Apache Spark's scalable machine learning library

- **ML Algorithms**: common learning algorithms such as classification, regression, clustering, and collaborative filtering
- **Featurization**: feature extraction, transformation, dimensionality reduction, and selection
- **Pipelines**: tools for constructing, evaluating, and tuning ML Pipelines
- **Persistence**: saving and load algorithms, models, and Pipelines

Spark GraphX



- Many ML problems can be stated as graphs such as search result page rank; count Meetup members and overlaps; similar purchasing habits
- GraphX is Apache Spark's API for graphs and graph-parallel computation
  - –Unifies extract/transfer/load, exploratory analysis, and iterative graph computation
  - –Competitive performance with the fastest graph systems
  - –Has a variety of built-in graph algorithms
  - –Currently only supports Scala

Conclusion

- Spark is the culmination of several successful high performance computing breakthroughs
- It was specifically designed for ML workloads and is highly performant for these types of applications
- It had considerable support from both the open source community and industry, which is why it is being adopted and adapted so quickly