PySpark

By: Yizhak Cohen

01

About PySpark

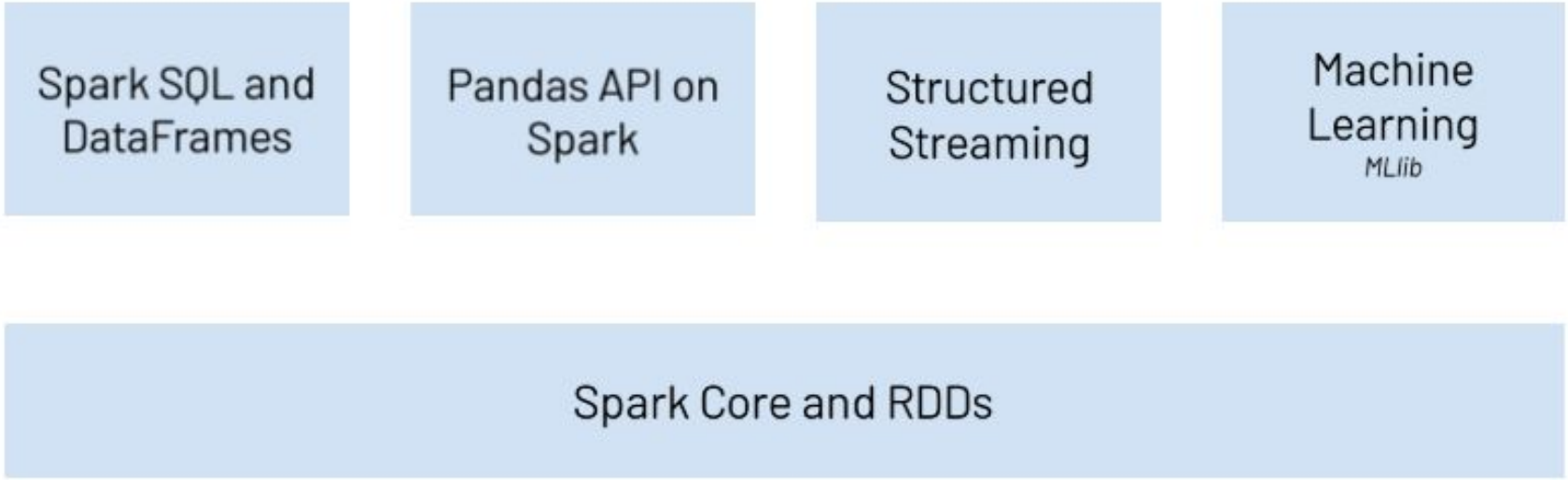
Overview

- Python API for Apache Spark, which enables large-scale data processing in a distributed environment
- Spark Core provides framework for the distributed environment using the Resilient Distributed Dataset (RDD)
- Can utilize SQL and Pandas API with Spark programs and run with multiple nodes
    - With Pandas, you can migrate code to production with Spark but keep pandas API for smaller datasets and local testing

RDD

- Distributed collection of objects cached in memory across a collection of machines (cluster)
- Results in faster execution by reading data into memory, processing it and writing results back in one step
- Reuses data through DataFrames, an abstraction of RDD, which can be reused in further Spark operations
- Data is distributed across the cluster and can be computed or moved into data store, which is managed by Spark Core
    - Usually no need to tinker with RDD for most purposes

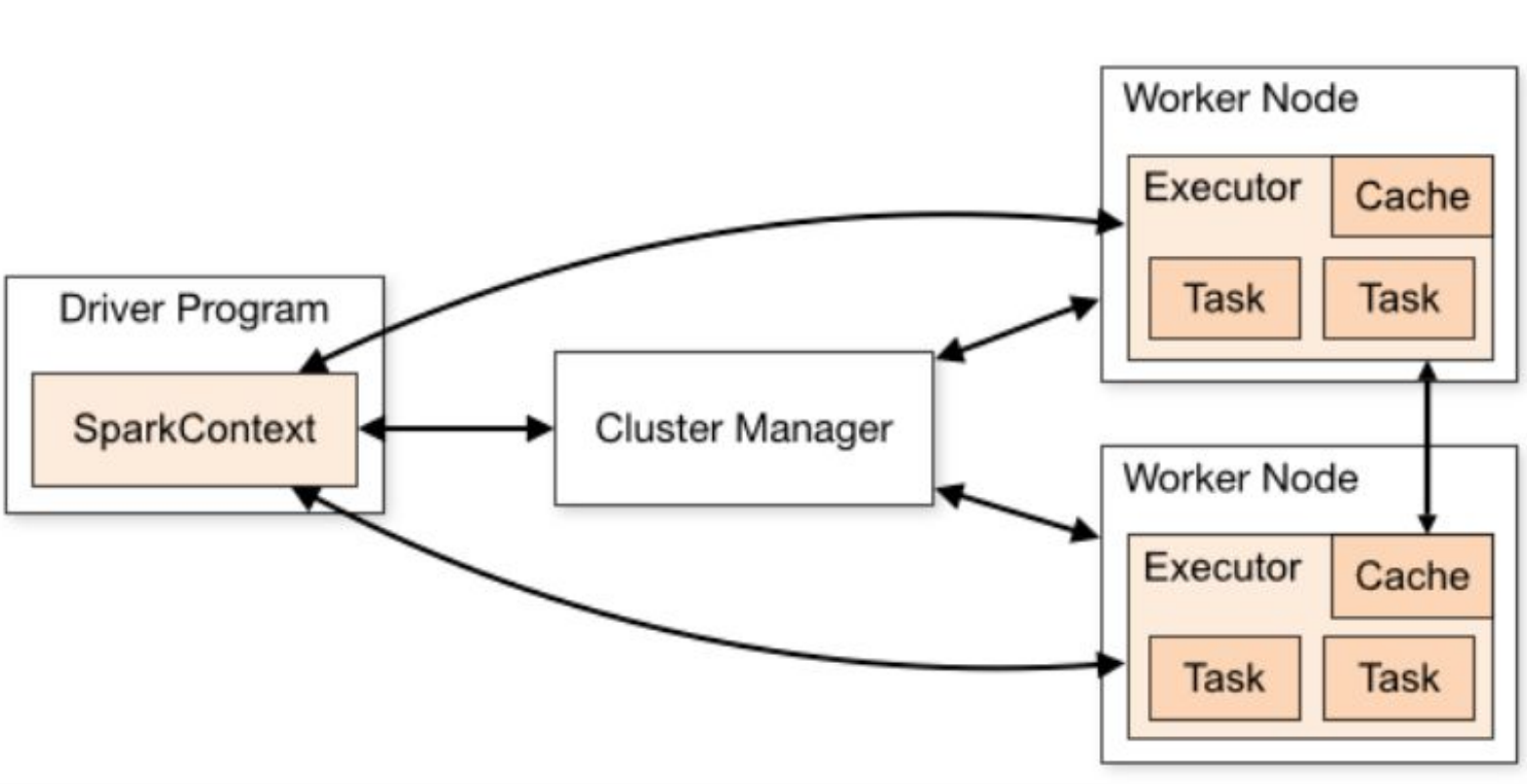
PySpark Architecture



Cluster

- Spark applications consists of multiple, independent processes on a cluster
  - The SparkContext object connects to the cluster through the cluster manager, which allocates resources for the application
    - (i.e. Mesos, YARN, Kubernetes, Spark's Manager)
- Once connected, it acquires executors (processes) on the nodes
  - Executors conduct operations and store data (caching) in the cluster
- SparkContext sends code to the executors and distributes tasks for them to run

Architecture



Benefits

- Speed
  - Caches data in memory in multiple parallel operations, reducing number of read/write disk operations, increasing speed
  - Faster than Hadoop MapReduce due to its sequential multi-step process of reads/write disk operations and data processing
- Easy to use with Python’s learnability
- Advanced Analytics
  - Able to do fast queries, stream processing, machine learning process, and graph processing

Spark Concepts

- DataFrames are primary objects in Spark (similar to Pandas DataFrame)
  - Schema defines column names/types
  - Records are row objects
  - Columns represent simple types or arrays, map, null
  - Immutable, instead save a new dataframe into a variable
- Processing:
  - Utilizes lazy evaluation where transformations are not computed until a specific action is called (i.e. display, head) and you can chain functions together

02

Testing

Resources

- https://spark.apache.org/docs/latest/api/python/index.html
- https://www.databricks.com/glossary/what-is-apache-spark
- https://www.databricks.com/spark/getting-started-with-apache-spark/quick-start
- https://aws.amazon.com/what-is/apache-spark/
- https://docs.databricks.com/en/pyspark/index.html
- https://docs.databricks.com/en/pyspark/basics.html
- https://spark.apache.org/docs/3.5.3/cluster-overview.html
- https://spark.apache.org/docs/latest/quick-start.html
- https://sparkbyexamples.com/
- https://andfanilo.github.io/pyspark-tutorial/#/
- https://medium.com/the-researchers-guide/introduction-to-pyspark-a61f7217398e