Introduction to Spark and PySpark

March 25th, 2024

Shashank

RESEARCH, INNOVATION & IMPACT
Data Science Institute

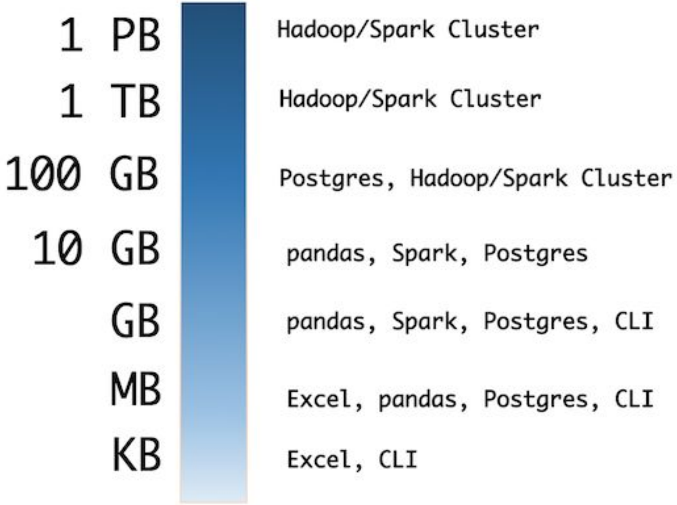
Introduction to Apache Spark

An open-source, distributed computing system that provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.

- **In-Memory Processing (Speed):** Spark's in-memory computing offers fast data processing by reducing disk I/O, ideal for iterative algorithms and real-time analytics.
- **Distributed Computing:** Enables parallel processing by distributing data and computations across a cluster, ensuring efficient resource use, scalability, and high availability.
- **Broad Language Support:** Offers APIs in Scala, Java, Python, and R, accommodating diverse development and data science.
- **Integration with Big Data Ecosystem:** Compatible with various data storage systems like HDFS, HBase, Cassandra, and Amazon S3, facilitating easy data reads/writes and integration with big data tools and frameworks.

Introduction to PySpark

Python API for Spark.



1 PB	Hadoop/Spark Cluster
1 TB	Hadoop/Spark Cluster
100 GB	Postgres, Hadoop/Spark Cluster
10 GB	pandas, Spark, Postgres
GB	pandas, Spark, Postgres, CLI
MB	Excel, pandas, Postgres, CLI
KB	Excel, CLI

- Allows for data transformation and analysis on large data sets, supports SQL queries, streaming data, machine learning, and graph processing, all within Python's syntax

- A vast library of resources, tools, and support available due to the large Python and Spark communities.

PySpark

- Best for batch and real-time data processing that requires fast execution, especially for machine learning algorithms and data transformations.

- Complexity and Flexibility: Ideal for complex data pipelines that involve aggregations, joins, window functions, and more.

Spark/PySpark vs. Hadoop/Hive

- **Processing Speed:** Spark provides in-memory processing which is significantly faster than the disk-based processing of Hadoop.

- **Ease of Use:** PySpark and Spark offer high-level APIs in Python, Java, Scala, and R, making them more accessible than Hadoop's MapReduce model.

- **Real-Time Processing:** Spark supports real-time processing capabilities, whereas Hadoop is primarily designed for batch processing. This makes Spark more suitable for applications requiring live data feeds.

Criteria	Hadoop/Hive	Spark/PySpark
Data Processing Speed	Optimal for batch processing where real-time speed is not critical.	Preferred for real-time analytics and when speed is crucial.
Data Size and Storage	Ideal for very large datasets; cost-effective storage on HDFS.	Best for processing that can fit data in memory; more expensive for storage.
Processing Type	Suited for batch processing and long-running jobs.	Ideal for both batch and real-time/streaming processing.
Complexity of Operations	Good for standard data warehousing operations with SQL-like queries (HiveQL).	Better for complex data transformations and ML algorithms.
Language Support	Primarily uses HiveQL for queries.	Supports Scala, Java, Python, and R, offering broader development flexibility.
Ecosystem Integration	Mature ecosystem with extensive tool integration for data management.	Robust integration with big data tools, but focuses more on analytics.
Cost	More cost-effective for data storage.	In-memory processing can be costly for very large datasets.
Use Cases	Data warehousing and historical data analysis, Large scale ETL jobs	Real-time data processing, Interactive data analysis, Machine learning