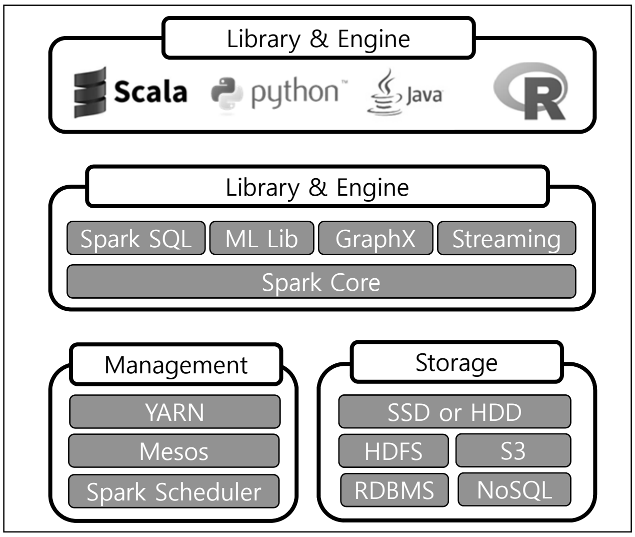
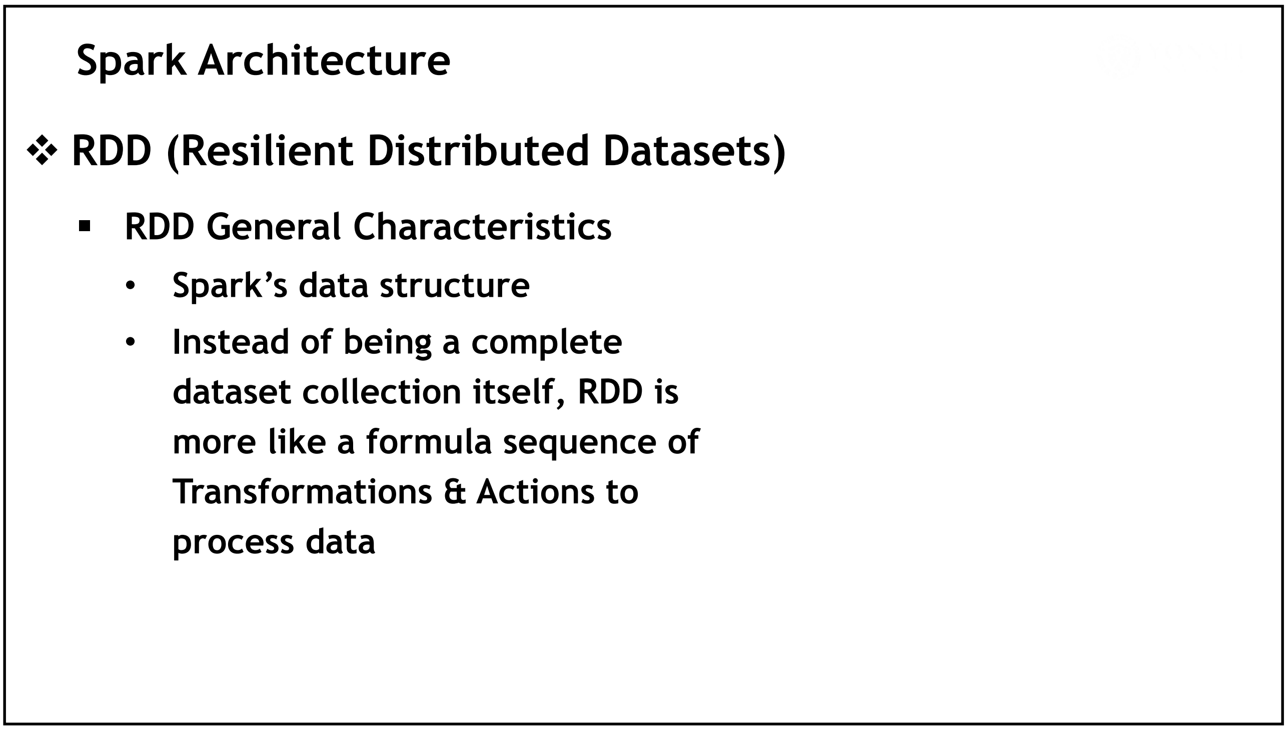
Big Data

Spark Architecture

Spark Architecture

❖ Spark Architecture

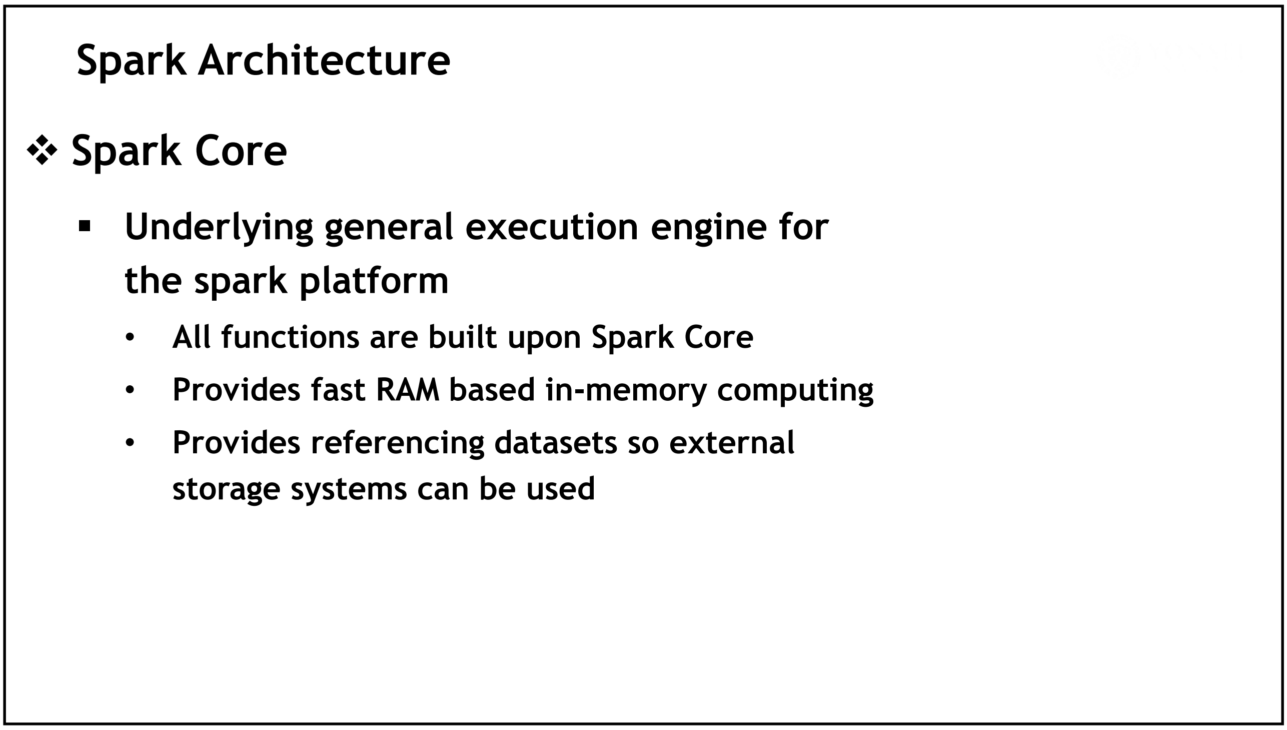




Spark Architecture

❖ RDD (Resilient Distributed Datasets)

- RDD General Characteristics
  - Spark’s data structure
  - Instead of being a complete dataset collection itself, RDD is more like a formula sequence of Transformations & Actions to process data



Spark Architecture

❖ Spark Core

- Underlying general execution engine for the spark platform
  - All functions are built upon Spark Core
  - Provides fast RAM based in-memory computing
  - Provides referencing datasets so external storage systems can be used

Spark Architecture

❖ Spark SQL

- SchemaRDD based data abstraction
- Provides support for structured and semi-structured data

Spark Architecture

❖ Spark Streaming

- Performs analysis on streaming data using the fast scheduling capability of Spark Core
- Input data is formed into mini-batches
- RDD (Resilient Distributed Datasets) transformation is performed on the mini-batches

Spark Architecture

❖ MLlib (Machine Learning Library)

- Built-in library of ML (Machine Learning) functions
- MLlib can be easily applied to the distributed memory based Spark architecture

Spark Architecture

❖ GraphX

- Distributed graph-processing framework
- Graph computation
  API (Application Programming Interface)
  - Optimized runtime processing of user-defined graphs

Spark Architecture

❖ Spark Cluster Management Methods

Spark Mode	Spark Central Master	Executors Initiation	Tasks Execution
Standalone	Standalone Master	Worker JVM	Executor
YARN	YARN App Master	Node Manager	Executor
Mesos	Mesos Master	Mesos Slave	Executor

Big Data

Spark Family

Spark Family

- YARN
  - YARN (Yet Another Resource Negotiator)
  - Apache Hadoop's cluster based resource management and job scheduling platform
  - Apache Hadoop NextGen MapReduce

Spark Family

- YARN
  - Divides Resource Management and Job Scheduling into separate daemons
    - Global RM (ResourceManager)
    - Per-application AM (ApplicationMaster)
  - Applications can be a Single job or a DAG (Directed Acyclic Graph) of jobs

Spark Family

❖ Mesos

- Apache Mesos is a distributed cluster resource management and scheduling platform
- Mésos in ancient Greek (μέσος) means "middle"or "mediator"

Spark Family

❖ Mesos

- Provides resource management and scheduling APIs for the entire datacenter and cloud environments
- Linear scalability for large clusters with 10,000s of nodes

Spark Family

❖ Mesos

- HA (High Availability) fault-tolerance through ZooKeeper replicated master/agents and non-disruptive upgrades
- Supports scheduling for legacy Cloud applications and added (pluggable) scheduling policies

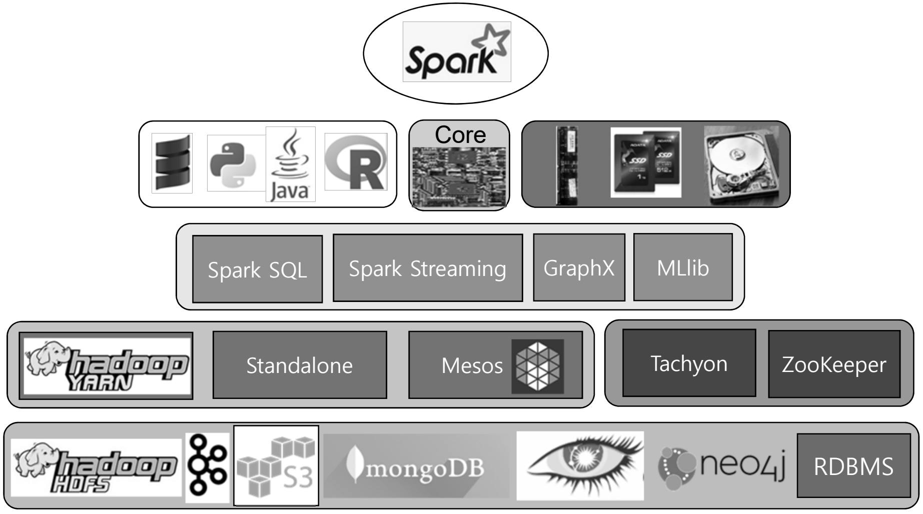
Spark Family

❖ Mesos

- Includes HTTP APIs to help develop cluster based distributed applications
- Built-in Web UI enables Cluster state monitoring and management
- Operates well on various OSs (Linux, OSX, Windows) and Cloud platforms

Spark Family

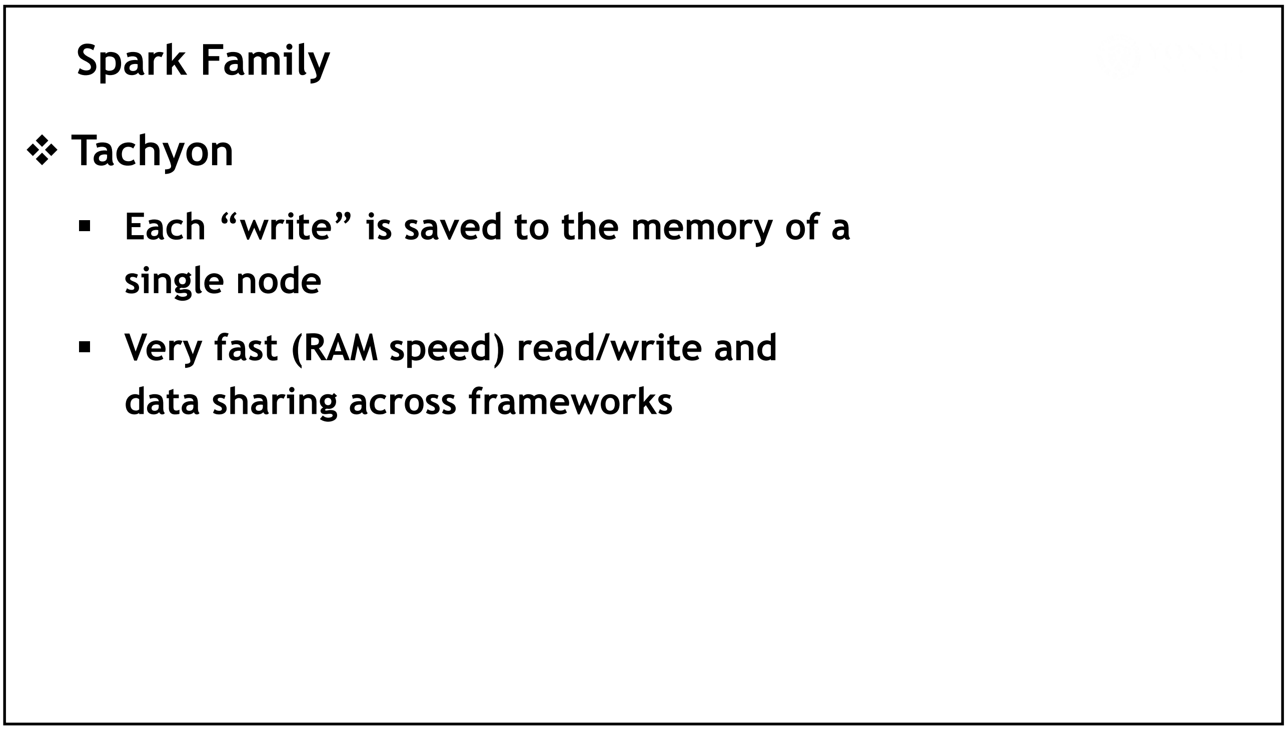
❖ Spark Family



Spark Family

❖ Tachyon

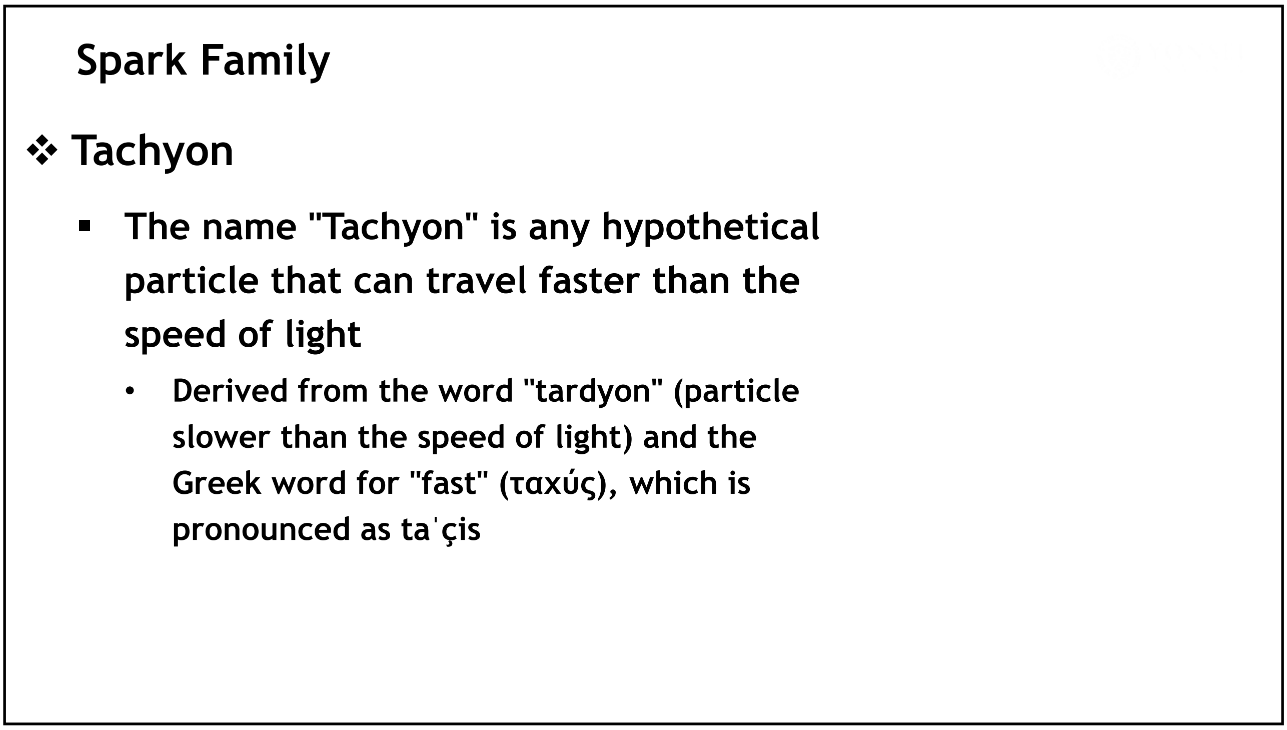
- Open source, distributed, fault-tolerant, and fast in-memory (RAM) file system
- Fast high-performance data sharing feature
  - In-memory (RAM) file system enables very fast data sharing
  - Lineage (log of transformation sequences) is used to build and recover datasets
  - Stores binary programs used to generate datasets as well as the input datasets



Spark Family

❖ Tachyon

- Each “write” is saved to the memory of a single node
- Very fast (RAM speed) read/write and data sharing across frameworks



Spark Family

❖ Tachyon

- The name "Tachyon" is any hypothetical particle that can travel faster than the speed of light
  - Derived from the word "tardyon" (particle slower than the speed of light) and the Greek word for "fast" (ταχύς), which is pronounced as taˈçis

Spark Family

❖ ZooKeeper

- Apache ZooKeeper provides highly reliable distributed coordination for clusters
- HA (High Availability) fault-tolerance provided through replicated master/agents and non-disruptive upgrades

Spark Family

❖ ZooKeeper

- Cluster based centralized control and easy management functions
- Maintains configuration information
- Naming
- Distributed synchronization
- Group services

Spark Family

❖ Cassandra

- Apache Cassandra is a distributed NoSQL database management system
- Supports large datasets and data exchange across various commodity servers
- Provides robustness through HA (High Availability) with no SPOF (Single Point of Failure)

Spark Family

❖ Cassandra

- Provides reliable management capability for multiple datacenter based clusters
- Low latency asynchronous masterless replication operations for all clients

Spark Family

❖ Kafka

- Apache Kafka is a distributed data streaming platform
- Stream data read (input) and write (output) management system
- Enables easy programming of scalable stream processes of real-time applications

Spark Family

❖ Kafka

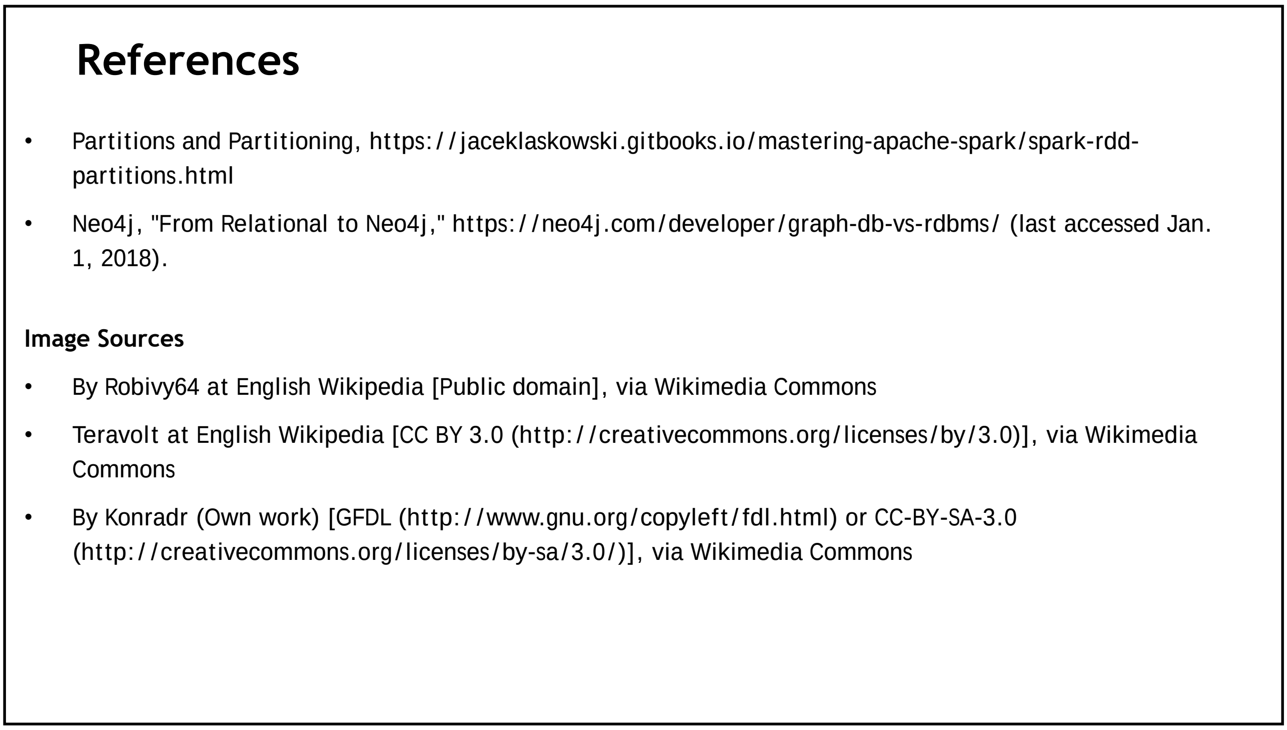
- Safely stores data streams in a distributed, replicated, fault-tolerant cluster
- Streaming features of Kafka are used in Spark Streaming

Big Data

References

References

- Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*. 1st Edition. O'Reilly, 2015.
- Sameer Farooqui, Databricks, **Advanced Apache Spark Training**, Devops Advanced Class, Spark Summit East 2015, http://slideshare.net/databricks, www.linkedin.com/in/blueplastic, March 2015.
- Apache Spark documents (all documents and tutorials were used)
  - http://spark.apache.org/docs/latest/rdd-programming-guide.html
  - http://spark.apache.org/docs/latest/rdd-programming-guide.html#working-with-key-value-pairs
  - https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html#rdd-persistence
- Wikipedia, www.wikipedia.org
- Stackoverflow, https://stackoverflow.com/questions
- Bernard Marr, "Spark Or Hadoop -- Which Is The Best Big Data Framework?," Forbes, Tech, June 22, 2015.
- Quick introduction to Apache Spark, https://www.youtube.com/watch?v=TgiBvKcGL24
- Wide vs Narrow Dependencies, https://github.com/rohgar/scala-spark-4/wiki/Wide-vs-Narrow-Dependencies



References

- Partitions and Partitioning, https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-rdd-partitions.html
- Neo4j, "From Relational to Neo4j," https://neo4j.com/developer/graph-db-vs-rdbms/ (last accessed Jan. 1, 2018).

Image Sources

- By Robivy64 at English Wikipedia [Public domain], via Wikimedia Commons
- Teravolt at English Wikipedia [CC BY 3.0 (http://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons
- By Konradr (Own work) [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/)], via Wikimedia Commons