Spark Tutorial

I. Introduction.

* not a modified version of hadoop. has its own cluster management
  - Hadoop is just one of the ways to implement Spark.
  - Spark uses Hadoop for storage purpose only.

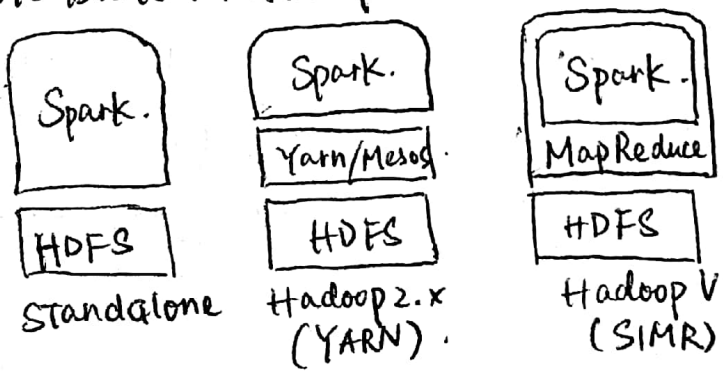
Apache Spark.

- based on Hadoop/MapReduce model, extends MapReduce model for more types of computation: interative query and stream processing.
- In-memory cluster computing.

Features:

- speed: 100 times faster in memory, 10x faster on disk.
  * reduce I/O operations to disk.
  * stores intermediate processing data in memory.
- support multiple languages.
- advanced analytics: support "map", "reduce", "SQL Query", "streaming data", "Machin learning", "Graph" alg

Spark Built on Hadoop.



- Standalone: Spark and MapReduce run side-by-side to cover all jobs.
- Hadoop YARN: Spark runs on Yarn w/o any preinstallation. Yarn helps integrate spark into HDFS.
- Spark in MapReduce (SIMR): launch spark in addition to standalone deployment.

■ Components of Spark.

- Spark Core: In-memory computing, reference datasets in external storage ~~components~~ system.
- Spark SQL: new data abstraction called SchemaRDD → structured data, semi-structured support
- Spark Streaming: provides streaming analytics
  → ingest data in mini batches.
  → perform RDD (resilient distributed datasets).
- MLlib: ML library, 9x faster ~~than~~ as fast as Mahout.
- GraphX.

II. RDD.

■ Resilient Distributed Datasets (RDD):

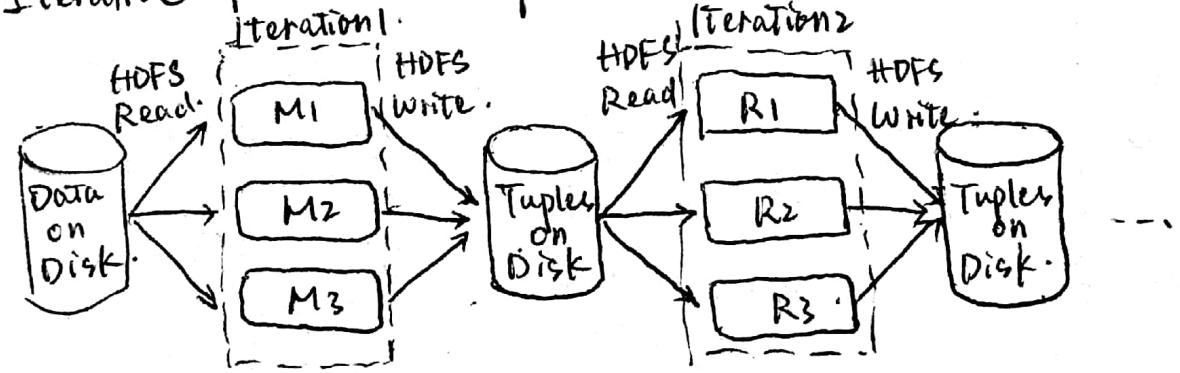
- immutable distributed collection of objects.
- read-only, partitioned collection of records.
- two ways to create RDDs:
  a) parallelizing an existing collection in your driver.
  b) referencing a dataset in an external storage system.

■ Data sharing is slow in M.R.

- Only way to reuse data between computations (between 2 MR jobs) is to write to ~~the~~ external stable storage system (HDFS).

* Data sharing is slow in MR due to: replication, serialization, and disk I/O. (90%).

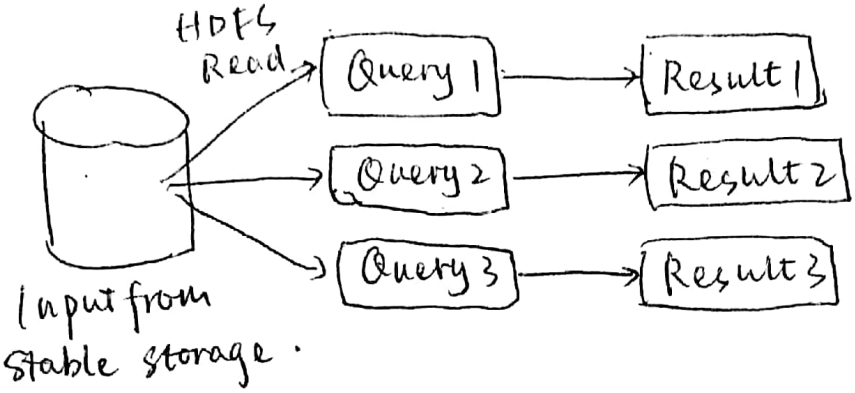
■ Iterative Operations on MapReduce.



Spark Tutorial

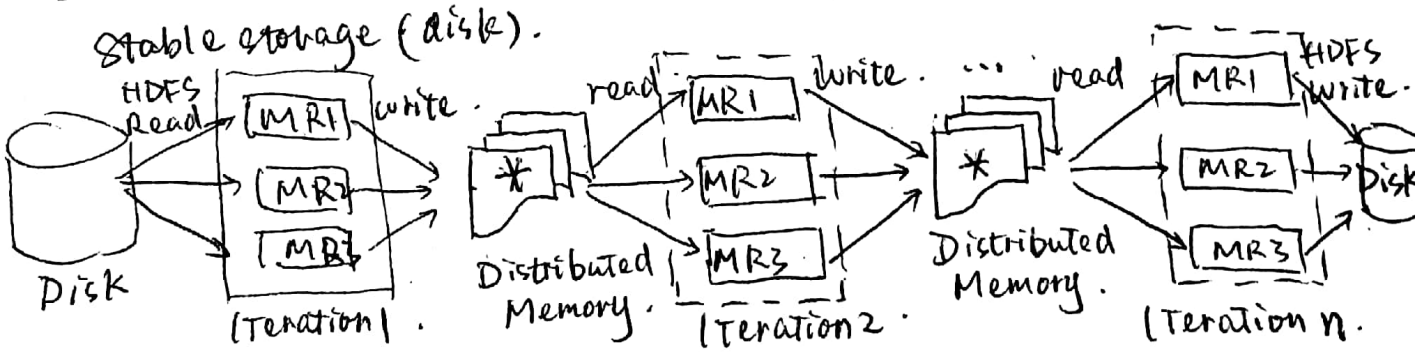
II. RDD (Con't).

- Interative operations on MapReduce

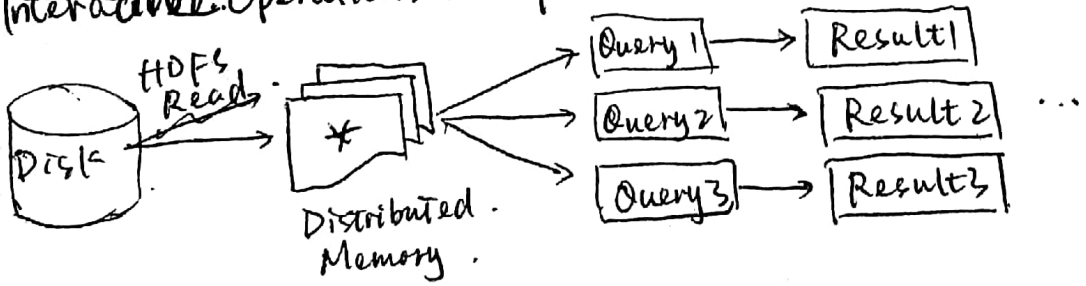


Disk I/O time consuming.

- Data sharing using Spark RDD.
  - RDD supports in-memory processing computation.
  - Stores the state of memory as an object across jobs. (sharable).
- Iterative operations on Spark RDD.
  - Store intermediate results in distributed memory instead of stable storage (disk).



- Interative Operations on Spark RDD



III. Spark Installation (Skip).

IV. Spark core programming.

- Spark Shell: (scala or python) create RDD from reading a file.
- RDD transformations:
  - map(func) • filter(func). • flatMap(func).
  - sample(....) • union(otherDataset).

Spark Tutorial

IV Spark Core Programming (Con't).

- Actions.
  - reduce(func). • collect(). ...
- Programming with RDD.