Map Reduce vs Spark

01 July 2024 09:53

MapReduce: is in disk operation, There will be multiple intermediate output generated for each mapper job which will result into multiple read/write operation in disk.

MapReduce: primarily uses disk based storage for intermediate data between Map and Reduce Stages. Each MapReduce job typically uses reading from disk and performing computation and writing back to disk, which can lead to slower processing at times.

Spark: Spark utilizes In-Memory processing to perform computations. It keeps intermediate data in memory, which allows for much faster data processing compared to MapReduce.

Spark: also provides fault-tolerance through RDDs(Resilient Distributed Dataset)which are stored in memory and cab be rebuilt if any node fails.

MapReduce: Suitable for batch processing where data processing requirements are not time-sensitive and it involves large volume of data, that can be handled sequentially.

Spark: Spark utilizes in iterative processing(such as machine learning algorithms),interactive queries and streaming data processing.

Spark: Performance benefits become particularly evident when dealing with complex workflows that require multiple iterative steps.

Limitations of MapReduce in Hadoop

-> Unsuitable for real-time processing
-> Unsuitable for trivial operations
-> Unsuitable for large data on network
-> Unsuitable with OLTP(SQL)
-> Unsuitable for processing graphs-Property graph(a type of data structure)
-> Unsuitable for iterative execution(ML algorithms)

Introduction to Apache spark

01 July 2024 10:32

It is suitable for real time processing, trivial operations and processing larger data on a network

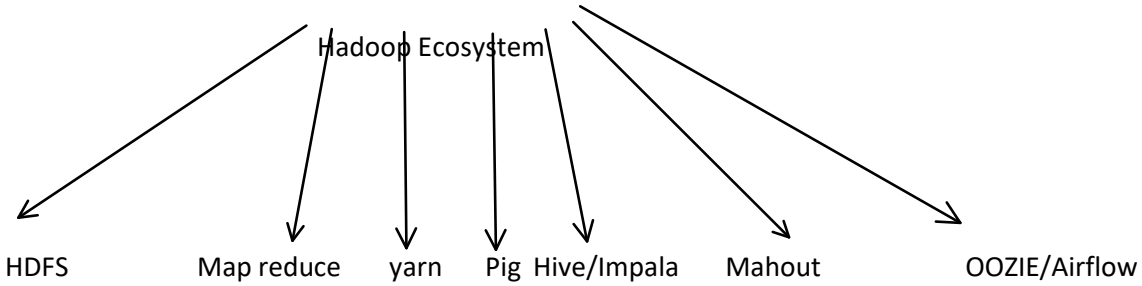
Is opensource cluster computing framework

Provides up to 100 times faster performance for a few applications with in memory primitives, compared to the two stage disk based MapReduce paradigm of Hadoop

Is suitable for ML algorithms, as it allows programs to load and query data repeatedly

Features of spark:

->Spark core and RDDs(Resilient Distributed Datasets)
->Spark SQL
->Spark Streaming-Helps run RDD Transformations on them, ingests data in small batchess
->ML lib(Machine Learning Libraries)
->Graph X- Distributed graph processing framework works on top of Spark ,It provides an API and optimized runtime for pregel abstraction.

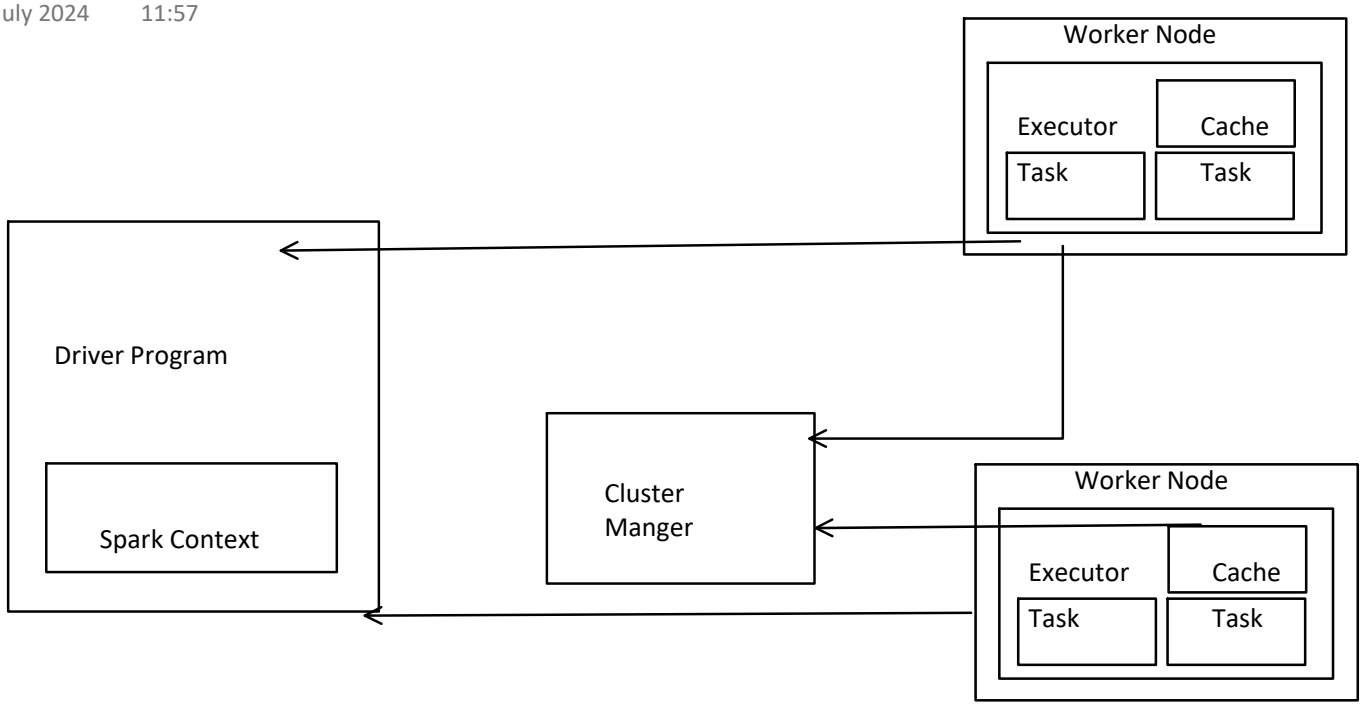


Hadoop Ecosystem vs Spark

Batch Processing
Structured Data Analysis
Interactive SQL Analysis
Realtime Streaming Data Analysis
Machine Learning Analysis

SPARK ARCHITECTURE

01 July 2024 11:57



1. Spark Context:

- This is Driver Program, which is entry gate to Spark Architecture while running or executing Job
- It communicates with the cluster manager to allocate resources and executes tasks

2.Cluster Manager:

- Manages physical resources across cluster
- There are three managers
- Eg: YARN, Mesos, Spark Standalone ,Kubernetes -DevOps etc.

3.Worker Node(Executors):

- Executors are launched on worker node by cluster management
- Data will be partitioned among worker nodes to execute tasks parallelly
- Worker node executes tasks or computations actively as assigned by the driver program
- Cache memory is used to store data , across all stages of job
- Spark has multiple options to store data - In memory
  - In Disk
  - In Memory and disk both

- Worker node will be physical node and processing its data will be dependent on Core of CPU, Spark app can run parallelize tasks across all cores of CPU within each worker node to achieve distributed data processing

4.Executor:

- Executor will be allocated within worker Node and it will be used for monitoring Tasks and it will destroy itself when task is finished.
- Executors are responsible for executing code which is assigned by driver program
- Storing and caching of data(Storing in memory) for their tasks in memory or in disk
- Report the status of tasks to Cluster manager

- Intermediate output generated will be stored in cache or in disk

Spark RDD

01 July 2024 12:39

Spark RDD is an immutable collection of objects which defines the data structure of Spark

Features of Spark RDD:

->Lazy Evaluation
->Fault Tolerant(Uses Acyclic graph to store states and all to recall when one stage fails)
->Persist
->Partition
->Coarse Grained Operation
->In memory computation
->Immutable
->Location stickiness

Creating Spark RDD

->Using Paralellized Collections

```
sc.parallelize([10,20,30,40,50])
```

->

Storage Levels

02 July 2024 09:24

MEMORY_ONLY
MEMORY_AND_DISK
MEMORY_ONLY_SER
MEMORY_AND_DISK_SER
DISK_ONLY
OFF_HEAP

Changing persistence

rdd.unpersist(), rdd.persist(new option)

DAG

02 July 2024 09:57

It's a graph where RDDs and the operations to be performed on TDDs are represented in the form of vertices and edges, respectively

Vertex-stages
Edges-Relations

Main function is Fault Tolerance

inferSchema used to set datatype automatically based on data, other wise all are saved as string

03 July 2024 09:56

3 TYPES OF TABLES IN HIVE TEMPORARY TABLE,MANAGE TABLE, EXTERNAL TABLE