25/25

Array Shah
60004210073
TE Comps - C12

Experiment-7

Aim:- To implement RDD in Pyspark

Theory:- Resilient Distributed Datasets (RDD) in Apache Spark are a core data structure that offers several features.

They are:-

i) Lazy Evaluation :- Transformations in RDDs are lazy, meaning they do not compute their results immediately. This allows Spark to optimize the execution plan.

ii) Fault Tolerance - RDDs track data lineage information, enabling them to rebuild lost data automatically in case of node failures

iii) Immutability - Once an RDD is created, its value cannot be changed, ensuring data consistency & stability

iv) Partitioning - RDDs are partitioned across the nodes in the cluster, which is the fundamental unit of parallelism in Spark.

Advantages:-

i) Cost and Time Efficiency - Persisting RDDs can save computational costs and execution time, making it more efficient to reuse computations

ii) Performance : The RDD-s in-memory computations and lazy evaluation lead to high performance for big data processing tasks.

iii) Faster iterations - Caching intermediate RDDs in iterative algorithms accelerates each iteration by eliminating the need to recompute it from scratch.

iv) Resource Efficiency - Cached RDDs efficiently utilizes

available memory and reduce unnecessary recomputation, ensuring efficient resource utilization.

Disadvantages:-

i) Not suitable for stateful applications - RDDs are not ideal for applications that require updates to the state store, such as web apps.
ii) No changes can be made in RDD once it is created
iii) RDD lacks enough storage memory
iv) The run-time type safety is absent in RDDs.
v) There is no input optimizations available in RDDs.

Conclusion:- RDDs are powerful for big data processing due to their performance, consistency and fault tolerance. However, their limitations such as handling structured data and lack of automated optimization, make them less ideal for certain use cases. Hence we implement RDD using PySpark.