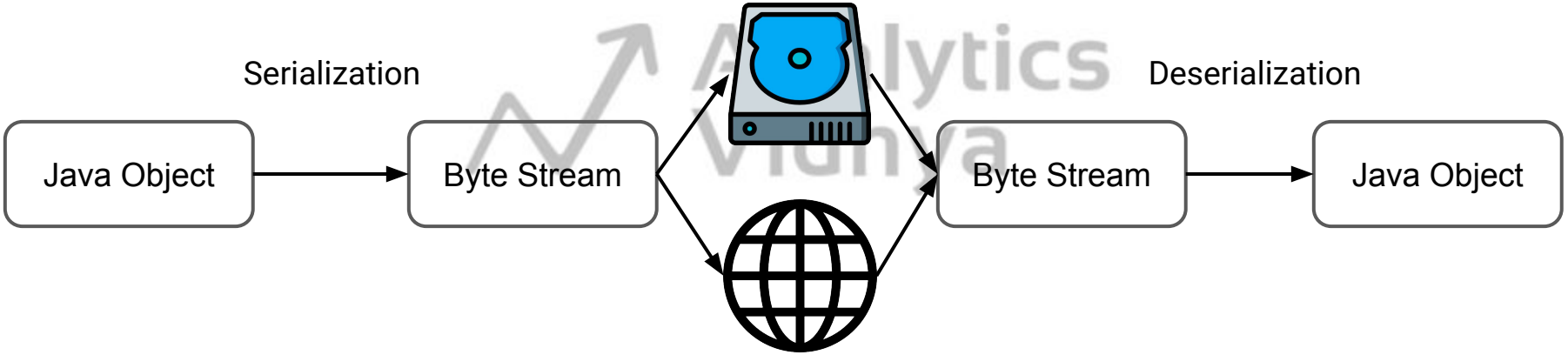
Storage Levels in Apache Spark

Serialization & Deserialization



Storage Levels in Apache Spark

MEMORY_ONLY

Allows storage of RDD as deserialized Java objects

Recomputes any RDDs not fitted in memory

MEMORY_AND_DISK

Allows storage of RDD as deserialized objects

Also stores RDDs on disk

MEMORY_ONLY_SER

Stores RDD as serialized Java objects

Enables better space efficiency

MEMORY_AND_DISK_SER

Similar to **MEMORY_ONLY_SER**, but spills partitions not fitted in memory to disk

Storage Levels in Apache Spark

DISK_ONLY

Store the RDD partitions only on disk

MEMORY_ONLY_2, MEMORY_AND_DISK2

Replicate every partition on 2 cluster nodes

OFF_HEAP

Allows storage of RDD in serialized format in Tachyon

Reduces garbage collection overhead and avoids losing in-memory cache

Features of RDD Persistence

Cost efficient

Spark computations are very expensive hence reusing the computations are used to save cost.

Time efficient

Reusing the repeated computations saves lots of time.

Execution time

Saves execution time of the job and we can perform more jobs on the same cluster.

Storage Levels in PySpark

In Python, stored objects will always be serialized with the Pickle library, so it does not matter whether you choose a serialized level.

DISK_ONLY

DISK_ONLY_2

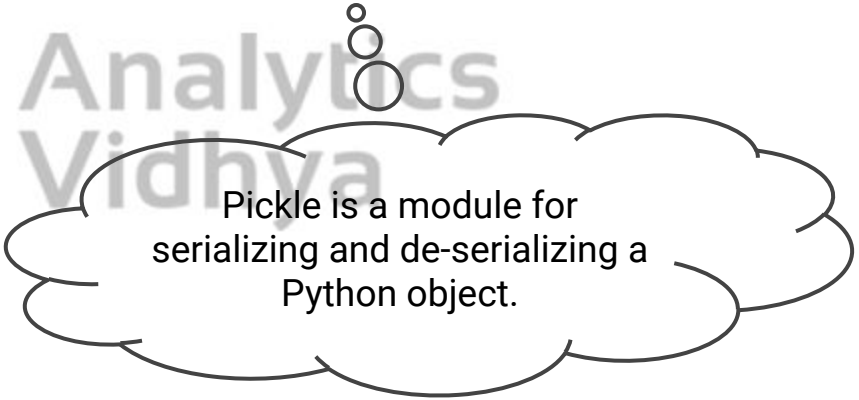
MEMORY_AND_DISK

MEMORY_AND_DISK_2

MEMORY_ONLY

MEMORY_ONLY_2

OFF_HEAP



Which Storage Level to Choose?

Spark's storage levels are meant to provide different trade-offs between memory usage and CPU efficiency.

If your RDDs fit comfortably with the default storage level,
leave them that way.

If not, try using MEMORY_ONLY_SER and selecting a fast serialization library to make the objects much more space-efficient, but still reasonably fast to access.

Use the replicated storage levels
If you want fast fault recovery.

Don't spill to disk unless the functions that computed your datasets are expensive.

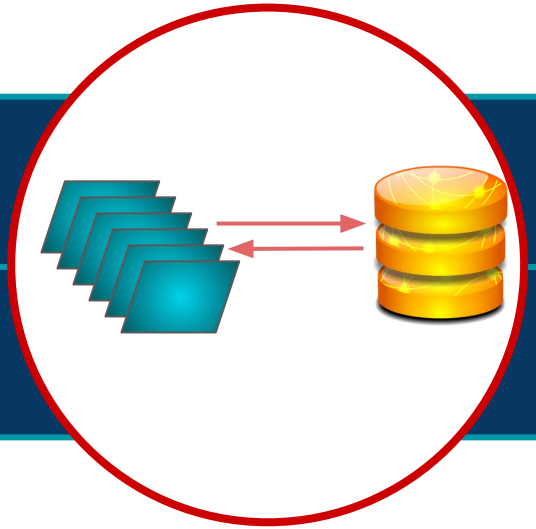
Analytics Vidhya

Changing Persistence Options

To change persistence option on an RDD:

Unpersist the RDD

RDD.Unpersist()



To change RDD persistence to other storage level:

Unpersist the RDD and then mark it for persistence with different storage level.

RDD.persist(pyspark.StorageLevel.MEMORY_ONLY)

Analytics Vidhya

Thank You!!