Persist and caching in Spark

.cache ( )

Spark has concept of persist and cache to optimize job execution by storing intermediate data in memory / disk.

29 → 2^{10}

⇓ Stored

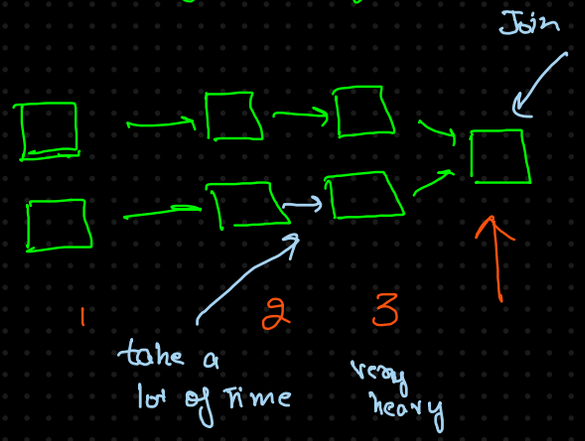
*2

Why caching needed in spark

⇒ in-memory

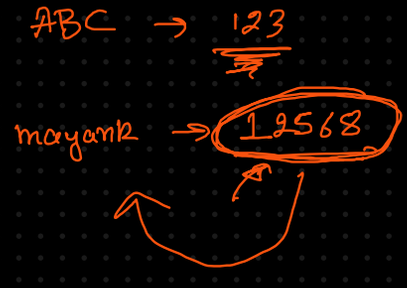
1. Repeated Use of data
2. Reducing disk reads:
   though processing happens in memory, initially data is stored in hdfs/s3
3. Lazy Evaluation Impact

why caching ⇒ in memory



Feature	cache()	persist(storageLevel)
Default Storage Level	MEMORY_AND_DISK (PySpark), MEMORY_AND_DISK_SER (Scala)	User-defined storage level
Control Over Storage	No	Yes
Ease of Use	Simpler, quick to implement	More flexible
Replication	No replication	Can replicate across nodes
Serialization	Not customizable	Customizable (serialized/deserialized)

Caching is a specific type of persist.



How and where caching happens

1. Memory

Cached data stored in deserialized format for fast processing

If memory is insuff., then spark drops the data to disk

2. Disk

data is serialized and written on local disk (not hdfs)

helps when memory is limited but increase disk I/O overhead.

When to use Caching?

1. when data is to be used multiple times
2. data is not very large to overwhelm
3. Avoid caching if memory pressure is a concern, as it may degrade overall cluster performance.

rdd → memory

tables → mem & disk

Say 10 blocks to cache

but space for 5

rdd → only 5 will be cached

table → in b/w memory & disk

2^9

$2^1 \rightarrow 2$

$2^2 \rightarrow 4$

$2^3 \rightarrow 8$

$2^4 \rightarrow 16$

.

.

$2^9 \rightarrow 5$

2^6