

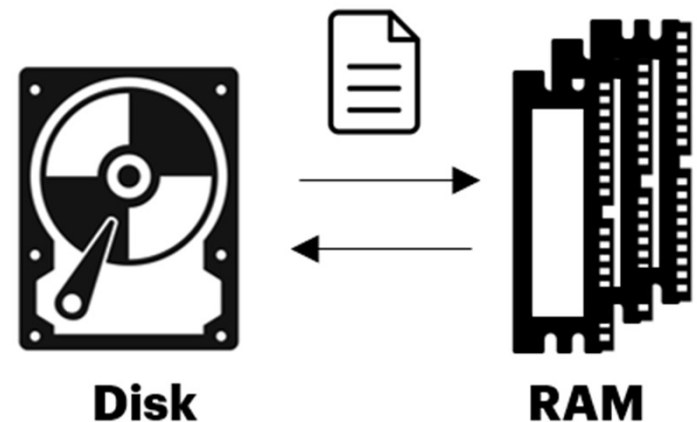
Spark Performance Issues

Spill Memory



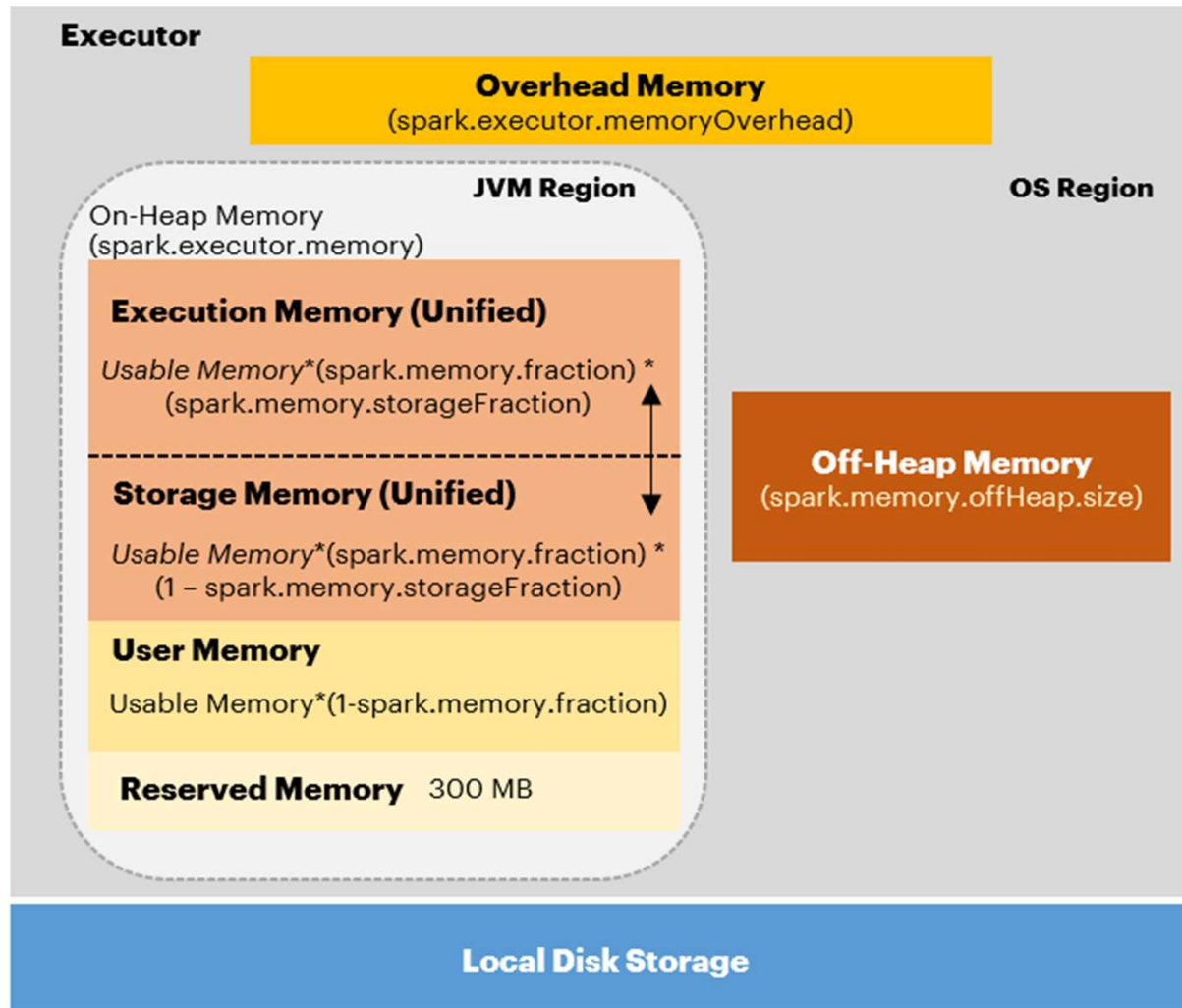
Spill problem

- Happens when the data moves from RAM to disk and then back to RAM again.
- This behaviour occurs when a given data partition is too large to fit within the RAM of the executor.
- Spark will read and write the surplus data into disk to free up memory space in the local RAM for the remaining tasks within the job
- This is an expensive process and slow!

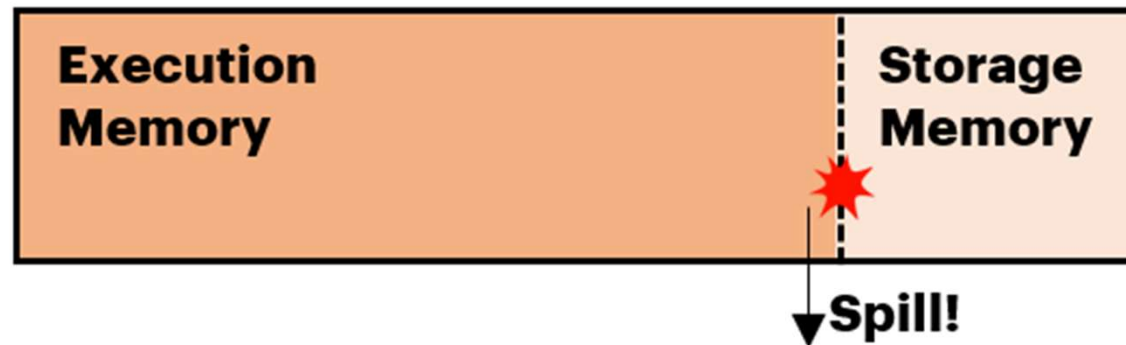
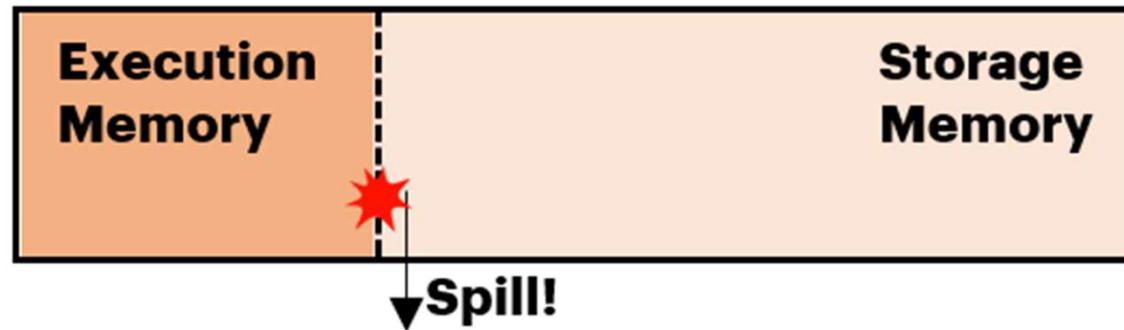
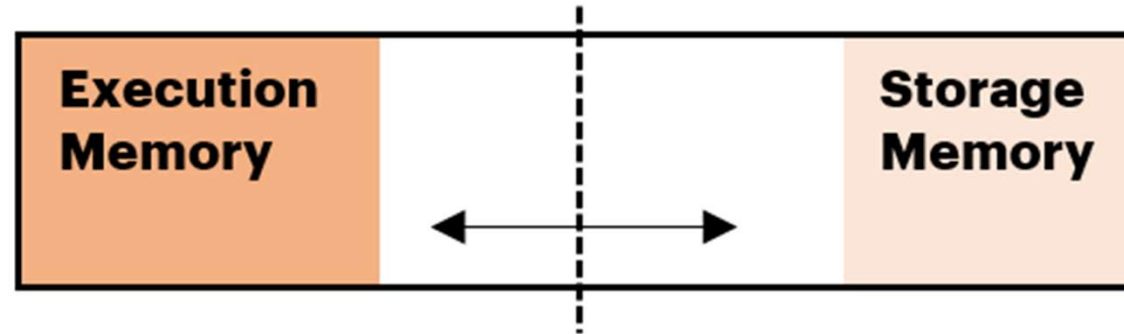


Structure of Memory in Spark Cluster

Worker Node



Unified memory



When Spill occurs

- Whenever total data size exceeds usable memory size (sum up from both execution task and storage task), this is when the spill occurs!

Mitigation of Spill

- If the spill is due to a skew problem, solve the skew first!
- Set the cluster with more memory per worker (increase worker size).
- Manage `spark.sql.shuffle.partitions` to reduce size per partition.
- Perform `.repartitioning()` in transformation code explicitly.
- Set `spark.sql.files.maxPartitionBytes` to fit your nature of data.

Thanks