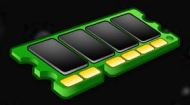
MEMORY



Spark uses memory for:

RDD Storage: when you call .persist() or .cache(). Spark will limit the amount of memory used when caching to a certain fraction of the JVM's overall heap, set by `spark.storage.memoryFraction`

Shuffle and aggregation buffers: When performing shuffle operations, Spark will create intermediate buffers for storing shuffle output data. These buffers are used to store intermediate results of aggregations in addition to buffering data that is going to be directly output as part of the shuffle.

User code: Spark executes arbitrary user code, so user functions can themselves require substantial memory. For instance, if a user application allocates large arrays or other objects, these will content for overall memory usage. User code has access to everything "left" in the JVM heap after the space for RDD storage and shuffle storage are allocated.

DETERMINING MEMORY CONSUMPTION

1. Create an RDD
2. Put it into cache
3. Look at `SparkContext` logs on the driver program or Spark UI

logs will tell you how much memory each partition is consuming, which you can aggregate to get the total size of the RDD

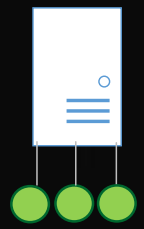
DATA SERIALIZATION

databricks

SERIALIZATION

Serialization is used when:

Transferring data over the network



Spilling data to disk

Caching to memory serialized

Broadcasting variables