

Partitioning Problems



Objective

Too few/too many partitions

Optimal partition sizes

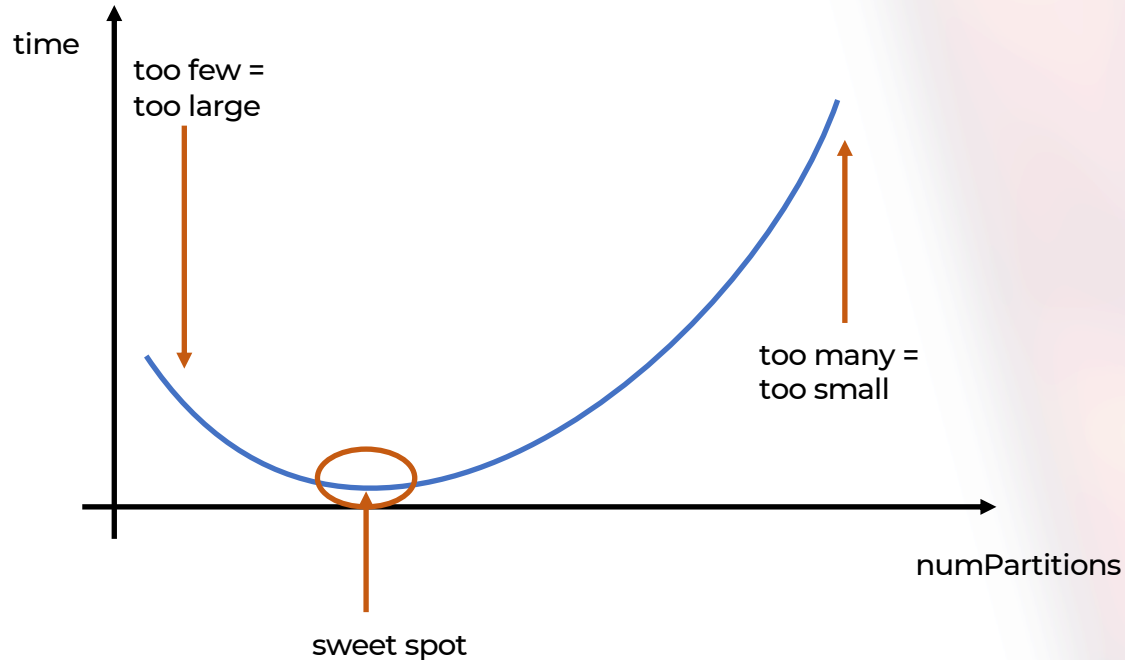
Estimating the size of an RDD/DataFrame



Perf = f(#partitions)

Number of partitions = degree of parallelism

- one partition = serial computation
- multiple partitions & multiple cores = parallel/distributed processing



Partitioning

Optimal partition size between 10 and 100MB

- not hard numbers – variations acceptable
- an order of magnitude difference usually not acceptable

Can configure the number of partitions at a shuffle

- default: 200 for DFs, #cores for RDDs
- set in `spark.sql.shuffle.partitions` for DFs
- set in `spark.default.parallelism` for RDDs

```
spark.sql.shuffle.partitions = 1000  
spark.default.parallelism = 100
```

Partitioning

Optimal #partitions

- too few = not enough parallelism
- too many = thread context switch for executors

Optimal partition size = 10 – 100MB of uncompressed data

```
spark.sql.shuffle.partitions = 1000  
spark.default.parallelism = 100
```

Determining the size of data:

- cache size: DF "native" size (compressed), uncompressed for RDDs
- SizeEstimator: not super accurate, but worth getting the order of magnitude
- query plan size in bytes: uncompressed data (DFs only)

Trial and error

Spark rocks

