Qubole

Hive Tuning - Quick Reference Guide

HDFS Blocks & MapReduce Splits

HDFS block size manages the storage of the data in the cluster and the Split Size drives how that data is read for processing by MapReduce.

dfs.blocksize	Sets the HDFS Block Size for storage - defaults to 128 MB
mapred.min.split.size	Sets the minimum split size - defaults to dfs.blocksize
mapred.max.split.size	Sets the maximum split size - defaults to dfs.blocksize

Block & Split Tuning

Configuring the Split Size boundaries for MapReduce may have cascading effects on the number of mappers created and the number of files each Mapper will access.

Blocks Required	Dataset Size / dfs.blocksize
Maximum Mappers Required	Dataset Size / mapred.min.split.size
Minimum Mappers Required	Dataset Size / mapred.max.split.size
Maximum Mappers per Block	Maximum Mappers Required / Blocks Required
Maximum Blocks per Mapper	Blocks Required / Minimum Mappers Required

Parallelism Tuning

The number of tasks configured for slave nodes determines the parallelism of the cluster for processing Mappers and Reducers. As the slots get used (by map/reduce jobs) if the number of slots was not appropriately configured there may job delays due to constrained resources.

mapred.tasktracker.map.tasks.maximum	Maximum number of map tasks
mapred.tasktracker.reduce.tasks.maximum	Maximum number of reduce tasks

Memory Tuning

If analysis of the tasks reveals that the memory utilization is low consider modifying the memory allocation for the Hadoop cluster. Reducing the allocated memory for the tasks will free up space on the cluster and allow for an increased in the number of Mappers or Reducers.

mapred.map.child.java.opts	Java heap memory setting for the map tasks
mapred.reduce.child.java.opts	Java heap memory setting for the reduce tasks