Balance the Load Across the Cluster

This section includes the following topics:

Tune the Split Size to Maximize Slot Availability

If the input data for the job is spread out evenly in the cluster, it improves MapReduce parallelism as more mappers can be scheduled to work on local data. The first task of any job is a single task called `setup`. The setup task examines the job input data to determine how many splits to use. For each split, the setup task finds the locations of the data to determine where to run the map task (one map task for each split). To balance the load evenly across the cluster, pick a split size that will give you a number of splits that fill at least a majority of the slots available to you, keeping in mind other jobs that may be running at the same time. If the input data for the job is spread out evenly in the cluster, it improves MapReduce parallelism as more mappers can be scheduled to work on local data.

Use Partition Lists

If your data is not distributed evenly throughout key ranges, you can create a list of partition keys (partition.lst) instead. To build this list, run a small MapReduce job that samples a small percentage of your data and divides it into even key ranges. The mappers use the partition list to determine the splits before sorting.