

# Customize Memory Settings for MapReduce v1

The map task memory is based on the chunk size. For most purposes, set map task memory to 800 MB. Reducer memory should be as high as practical, because the output from the mappers can be quite large. The sort code in the reducer uses a simple byte=array construct to hold all the data from the mappers before starting the merge. Because the JVM uses a 4-byte integer to track array size, the array does not require more than 2GB of memory at any point. To account for Java overhead, set the reducer memory to 3.5 GB. Make sure the JobTracker has plenty of memory, and plenty of IPC handlers (threads). On large clusters (50 nodes or more), give the JobTracker 40 to 50 handlers, and plenty of heap space – up to 30GB, if necessary.

As a general guideline, make sure each reducer has 3.5 GB of memory by setting the value of the `mapred.reduce.child.java.opts` parameter to `-Xmx3500m`. Set map task memory to 800 MB by adding `-Xmx800m` to the value of the `mapred.map.child.java.opts` parameter.

For more specific tuning, determine how much memory is left over after the operating system and MapR are accounted for, then determine whether to raise or lower the number of map or reduce slots, or the memory to allocate to each.

For information about memory allocation, see [Memory Allocation for Nodes](#).