MapReduce内存参数自动推断

在hadoop 2.0中，为MapReduce作业设置内存参数非常繁琐，涉及到两个参数：mapreduce.{map,reduce}.memory.mb和mapreduce.{map,reduce}.java.opts，一旦设置不合理，则会使得内存资源浪费严重，比如将前者设置为4096MB，但后者却是“-Xmx2g”，则剩余2g实际上无法让java heap使用到。Task相关参数配置如下：

|  |  |  |
| --- | --- | --- |
| 参数 | 描述 | 默认值 |
| mapreduce.map.memory.mb | Map Task申请内存大小 | 1G |
| mapreduce.map.java.opts | Map Task JVM的参数 | “” |
| mapreduce.reduce.memory.mb | Reduce Task申请内存大小 | 1G |
| mapreduce.reduce.java.opts | Reduce Task JVM的参数 | “” |
| yarn.nodemanager.vmem-pmem-ratio | Task的虚拟内存 | 2.1 |

理论上mapreduce.{map,reduce}.java.opts设置的最大堆内存要比{map|reduce}.memory.mb小，一般设置为0.75倍的memory.mb即可。yarn.nodemanager.vmem-pmem-ration被设置为2.1，Task任务只能使用2.1大小的虚拟内存，如果使用的量超出则会被nm kill掉。

在3.0以后，mapreduce.map.memory.mb配置为-1，默认配置为1024，并且增加参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 描述 | 默认值 |
| mapreduce.job.heap.memory-mb.ratio | memory.mb与java.opts的比例 | 0.8 |

如果有memory.mb或者java.opts没有配置，则计算公式如下：

*mapreduce.{map|reduce}.java.opts =*

*mapreduce.{map|reduce}.memory.mb \* mapreduce.heap.memory-mb.ratio*

其他参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 描述 | 默认值 |
| mapreduce.task.io.sort.mb | spill的缓存大小 | 100 |
| mapreduce.task.io.sort.factor | merge的spill文件格式 | 10 |

map输出到缓存中，先放到KvBuffer中，其缓存通过io.sort.mb参数进行配置，默认100m，当数据在Buffer使用大小超过80%时，开始spill，在Spill之前会进行sort。这个过程中会产生多个spill，要进行merge操作。merge操作的spill文件个数通过io.sort.factor进行配置，默认10个文件就可以进行merge

https://issues.apache.org/jira/browse/MAPREDUCE-5785