MapReduce中间结果

Mapper.run在执行过程中，将处理结果写到内存缓冲区中，当缓冲区被使用完后，需要spill到磁盘中，过程如下图所示：



由IFile.Writer写入磁盘，源码如下：

*while (spindex < mend &&*

*kvmeta.get(offsetFor(spindex % maxRec) + PARTITION) == i) {*

*final int kvoff = offsetFor(spindex % maxRec);*

*int keystart = kvmeta.get(kvoff + KEYSTART);*

*int valstart = kvmeta.get(kvoff + VALSTART);*

*key.reset(kvbuffer, keystart, valstart - keystart);*

*getVBytesForOffset(kvoff, value);*

*writer.append(key, value);*

*++spindex;*

*}*

map执行结束后，spill到磁盘的文件一般为多个，需要对文件进行merge操作，操作的过程中会对文件中的数据进行归并。



这个过程结束后，每个Map会生成一个MapOutputFile，Map结束后通知AM，启动Reduce任务。

MapOutputFile的默认定义文件为：

*static final String MAP\_OUTPUT\_FILENAME\_STRING = "file.out";  
static final String MAP\_OUTPUT\_INDEX\_SUFFIX\_STRING = ".index";  
static final String REDUCE\_INPUT\_FILE\_FORMAT\_STRING = "%s/map\_%d.out";*

其目录为：

*/tmp/hadoop-hadoop/nm-local-dir/usercache/hadoop/appcache/application\_1533738450228\_0004/container\_1533738450228\_0004\_01\_000002*

MapTask#Flush的相关代码如下：

*Path outputPath = mapOutputFile.getOutputFile();  
fileOutputByteCounter.increment(rfs.getFileStatus(outputPath).getLen());  
// If necessary, make outputs permissive enough for shuffling.  
if (!SHUFFLE\_OUTPUT\_PERM.equals(  
 SHUFFLE\_OUTPUT\_PERM.applyUMask(FsPermission.getUMask(job)))) {  
 Path indexPath = mapOutputFile.getOutputIndexFile();  
 rfs.setPermission(outputPath, SHUFFLE\_OUTPUT\_PERM);  
 rfs.setPermission(indexPath, SHUFFLE\_OUTPUT\_PERM);  
}*

MapReduce相关配置如下：

*this.mapOutputFile = ReflectionUtils.newInstance(  
 conf.getClass(MRConfig.TASK\_LOCAL\_OUTPUT\_CLASS,  
 MROutputFiles.class, MapOutputFile.class), conf);  
this.lDirAlloc = new LocalDirAllocator(MRConfig.LOCAL\_DIR);*

*public static final String TEMP\_DIR = "mapreduce.cluster.temp.dir";  
public static final String LOCAL\_DIR = "mapreduce.cluster.local.dir";*

*mapreduce.cluster.local.dir for child: /hadoop/yarn/local/usercache/hdfs/appcache/application\_1533019554063\_0049*

*$localdir/usercache/$user/appcache/$appId/output/${attemptId}/file.out*

https://www.quora.com/When-is-Hadoop-MapReduce-better-than-Spark