MapReduce作业Uber模式介绍

大家在提交MapReduce作业的时候肯定看过如下的输出：

```
17/04/17 14:00:38 INFO mapreduce.Job: Running job: job_1472052053889_0001
17/04/17 14:00:48 INFO mapreduce.Job: Job job_1472052053889_0001 running in uber mode :
 false
17/04/17 14:00:48 INFO mapreduce.Job:  map 0% reduce 0%
17/04/17 14:00:58 INFO mapreduce.Job:  map 100% reduce 0%
17/04/17 14:01:04 INFO mapreduce.Job:  map 100% reduce 100%
```

注意上面日志的第二行，显示job_1472052053889_0001不是以uber模式运行的。本博客讲介绍uber模式的启用，特点等。

什么是uber模式

Uber模式简单地可以理解成JVM重用，该模式是Hadoop
2.x开始引入的；以Uber模式运行MR作业，所有的Map Tasks和Reduce Tasks将会在Application
Master所在的容器（container）中运行，也就是说整个MR作业运行的过程只会启动AM
container，因为不需要启动mapper 和 reducer
containers，所以AM不需要和远程containers通信，整个过程简单了。

不是所有的MR作业都可以启用Uber模式，如果我们的MR作业输入的数据量非常小，启动Map container或Reduce container的时间都比处理数据要长，那么这个作业就可以考虑启用Uber模式运行，一般情况下，对小作业启用Uber模式运行会得到2x-3x的性能提升。

启用uber模式的要求非常严格，代码如下：

```
isUber = uberEnabled && smallNumMapTasks && smallNumReduceTasks
  && smallInput && smallMemory && smallCpu
  && notChainJob && isValidUberMaxReduces;
```

- uberEnabled：其实就是 mapreduce.job.ubertask.enable 参数的值，默认情况下为 false ；也就是说默认情况不启用Uber模式；
- smallNumMapTasks：启用Uber模式的作业Map的个数必须小于等于 mapreduce.job.ubertask.maxmaps 参数的值，该值默认为9；也计算说，在默认情况下，如果你想启用Uber模式，作业的Map个数必须小于10；
- smallNumReduceTasks ：同理，Uber模式的作业Reduce的个数必须小于等于mapreduce.job.ubertask.maxreduces，该值默认为1；也计算说，在默认情况下，如果你想启用Uber模式，作业的Reduce个数必须小于等于1；
- smallInput：不是任何作业都适合启用Uber模式的，输入数据的大小必须小于等于 mapreduce.job.ubertask.maxbytes 参数的值，默认情况是HDFS一个文件块大小；
- smallMemory ：因为作业是在AM所在的container中运行，所以要求我们设置的Map内存（mapreduce.map.memory.mb）和Reduce内存（mapreduce.reduce.memory.mb）必须小于等于AM所在容器内存大小设置（yarn.app.mapreduce.am.resource.mb）；
- smallCpu：同理，Map配置的vcores（mapreduce.map.cpu.vcores）个数和 Reduce配置的vcores（mapreduce.reduce.cpu.vcores）个数也必须小于等于AM所在容器vcores个数的设置（yarn.app.mapreduce.am.resource.cpu-vcores）；
- notChainJob：此外，处理数据的Map class（mapreduce.job.map.class）和Reduce class（mapreduce.job.reduce.class）必须不是 ChainMapper 或 ChainReducer 才行；
- isValidUberMaxReduces：目前仅当Reduce的个数小于等于1的作业才能启用Uber模式。

同时满足上面八个条件才能在作业运行的时候启动Uber模式。下面是一个启用Uber模式运行的作业运行成功的日志：

```
File System Counters
  FILE: Number of bytes read=215
  FILE: Number of bytes written=505
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=1200
  HDFS: Number of bytes written=274907
  HDFS: Number of read operations=57
  HDFS: Number of large read operations=0
```

```
    HDFS: Number of write operations=11
  Job Counters
    Launched map tasks=2
    Launched reduce tasks=1
    Other local map tasks=2
    Total time spent by all maps in occupied slots (ms)=3664
    Total time spent by all reduces in occupied slots (ms)=2492
    TOTAL_LAUNCHED_UBERTASKS=3
    NUM_UBER_SUBMAPS=2
    NUM_UBER_SUBREDUCES=1
  Map-Reduce Framework
    Map input records=2
    Map output records=8
    Map output bytes=82
    Map output materialized bytes=85
    Input split bytes=202
    Combine input records=8
    Combine output records=6
    Reduce input groups=5
    Reduce shuffle bytes=0
    Reduce input records=6
    Reduce output records=5
    Spilled Records=12
    Shuffled Maps =0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=65
    CPU time spent (ms)=1610
    Physical memory (bytes) snapshot=1229729792
    Virtual memory (bytes) snapshot=5839392768
    Total committed heap usage (bytes)=3087532032
  File Input Format Counters
    Bytes Read=50
  File Output Format Counters
    Bytes Written=41
```

细心的同学应该会发现里面多了 TOTAL_LAUNCHED_UBERTASKS、NUM_UBER_SUBMAPS 以及 NUM_UBER_SUBREDUCES 信息，以前需要启用Map Task 或 Reduce Task运行的工作直接在AM中运行，所有出现了NUM_UBER_SUBMAPS和原来Map Task个数一样；同理，NUM_UBER_SUBREDUCES 和Reduce Task个数一样。

本文链接: 【】（）