# 资源配置及测试说明

首先，说明在上次讨论时，为什么测试groupByTest, spark集群会只有54个core？

spark-defaults.conf中配置spark.executor.cores 2 spark.executor.memory 4g 按理说集群的executor数量为 9(worker数量) \* 8(每个worker的core数量) / 2 = 36 core数量为36 \* 2 = 72

但是，注意这里其实并没有起36个executor，因为memory不够分, 每个worker的memory才14.6g，还要被守护进程占去一些，所以每个worker不能起4个executor, 只能起3个executor。这时集群的core数量就是3\*9\*2 = 54

然后，说明--num-executors、--executor-memory、--executor-cores、--total-executor-cores配置参数的使用

1. 使用范围：

通用：--executor-memory

Only Spark Standalone : --total-executor-cores

Spark Standalone and Yarn : --executor-cores

Only Yarn : --num-executors

所以，大家知道为什么—num-executors不起作用了…

1. 参数意义：

--executor-memory ：每个executor的memory大小

--executor-cores ：每个executor的core个数

--total-executor-cores ：分配给application(eg. groupByTest)的总core数

所以，并没有直接指出分配给application的executor数量，而是通过total-executor-cores/ executor-cores得到executor数量。

还有为什么要有--total-executor-cores？因为application默认是占满集群资源的，如果不配置，我随便运行个spark-shell，其他人就啥也干不了。

最后，说明我们怎么测试？

现在配置每个Worker分配的memory为12g，spark.executor.cores 2，spark.executor.memory 3g。即现在application所用资源，是36个executor，72个core。

那么，测试时，我们可以设置 --executor-cores 4 --executor-memory 6g ，即现在application所用资源，是18个executor，72个core。

还有，通过--total-executor-cores、--executor-memory、--executor-cores组合来测试扩展性，eg. spark.executor.cores 2，spark.executor.memory 3g --total-executor-cores 36，即现在application所用资源，是18个executor，36个core。剩下的大家自己组合吧。