大数据综合处理实验——课程设计1

许嘉禾 朱佳灏 张杨阳

分工：

许嘉禾：task2， task5

朱佳灏：task1， task4

张杨阳：task3

任务一

任务一较为简单，是统计每场比赛的比赛结果。由于每天每支球队最多只有一场比赛，所以可以在map中返回的key表示为<比赛时间,主队,客队>，value表示为<主队/客队,1/2/3分>，然后在reduce中分别加起来统计即可。

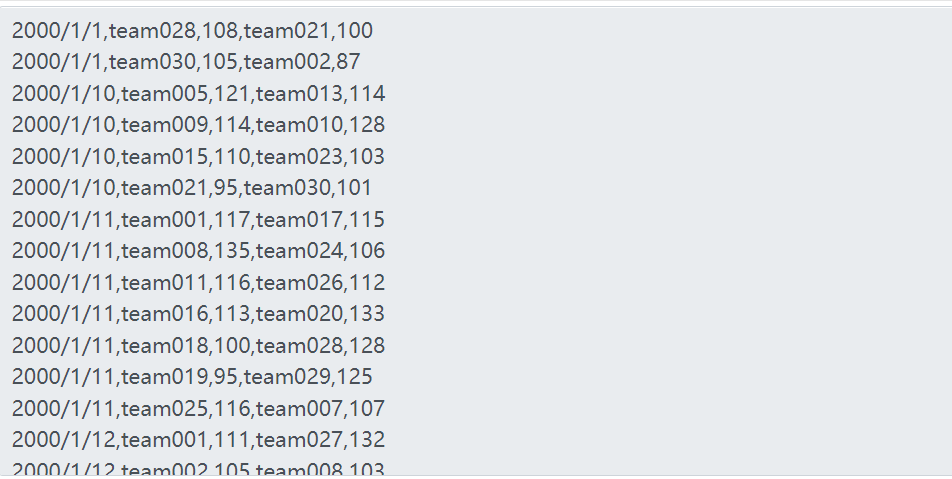
Map

输入格式为默认的FilelnputFormat，value为一行的内容，我们把它转成String后再用split将其分解为各个字段，观察当前PlayBy字段的球队是主队还是客队，是主队则设置value第一个字段为1，否则设为2。接下来再看具体得分（1分、2分还是3分）且结果为make，则在value的第二个字段设为得到的分数。比如客队的2-pt为make，value值则为“2,2”。至于key值，如开头所讲，设为<比赛时间,主队,客队>即可。

Reduce

reduce相对简单，只需遍历所有value值，分别计算主队和客队的得分即可，输出的key为null，value为“比赛时间，主队名，主队比分，客队名，客队比分”。比如“2000/1/1,team028,108,team021,100”。

集群测试结果



任务二

计算各项技术统计前5名的球员。分两个job完成，第一个job用于统计每个球员的各项数据的总数，第二个job再利用上一job的结果在各项技术统计中排序。

Job1的map：

输入格式为默认的FileInputFormat，即key为行偏移，value为一行的内容（即一条日志记录）。用String类的split将value分解为各个字段，然后根据各个字段的值提取出该条记录中球员的数据贡献。输出的key为球员名字，value为某项技术统计值（如“score,3”，“assist,1”这样的形式）。

Job1的reduce：

Key为球员名，values为该球员各项数据贡献的集合。在reduce中对各项数据进行累加即可得到该球员各项数据统计的总值，把该信息输出到一个临时文件中。输出的键值对的格式为<球员名，各项数据统计>，例如<“player1”,“score:50,rebound:8,assist:7,steal:2,block:1”>。

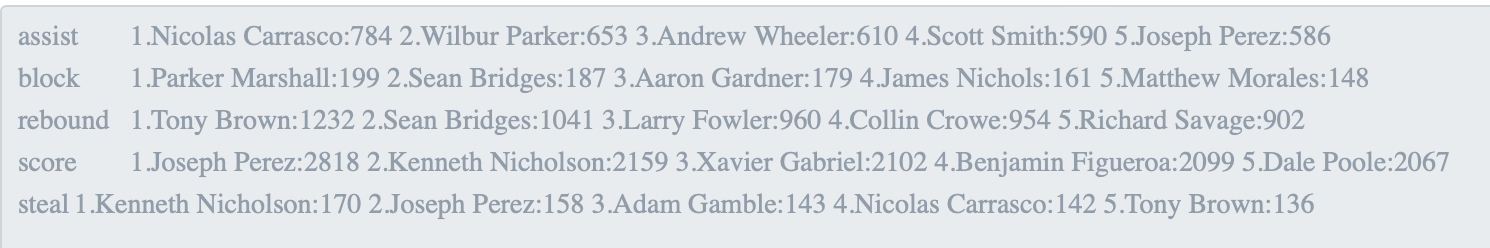
Job2的map：

从job1的输出文件中读取输出，作一个key和value的翻转。Job1得到的是key为球员名，value是数据统计名称和数值，翻转之后，key为数据统计名称，value为球员名和该球员在该项数据统计上的数值。

Job2的reduce：

输入的key为数据统计名（score，rebound等），values为各个球员与数值的集合。对数值进行排序，提取出前5名球员输出。为了排序，自定义一个类实现Comparable接口，类中的两个成员分别是string类型的球员名name和int类型的数值value，然后重写compareTo方法（按value从大到小排序）。

集群测试结果：



任务三

任务三是预测给定两球队比赛的胜率，我是分为三个job来写。首先job1根据日志得到每场比赛的日期，主队名称，主队比分，客队名称，客队比分，也就是任务一的结果。然后job2根据每场比赛的得分情况得到每一队的胜率和场均得分。Job2的结果写到job3的cachefile中，然后存到哈希表里。Job3的输入是predict文件，根据每一行的两队，分别去哈希表找他们的胜率和场均得分，按照胜率权重9，场均得分权重1的比例去计算最后对战胜率。

Job1与任务一相同。

Job2的map

Map的输入是job1的输出，也就是<日期，主队名称，主队得分，客队名称，客队得分>，把队伍名做为key，队伍的输赢和该场得分作为value（如“1,100”，表示赢了 并且该场得了100分）

Job2的reduce

Reduce根据队名，将一个队所有的比赛放在一起，计算胜利场次，总场次，总得分，然后计算出队伍的胜率和场均得分。Key是队伍名，value是胜率和场均得分。

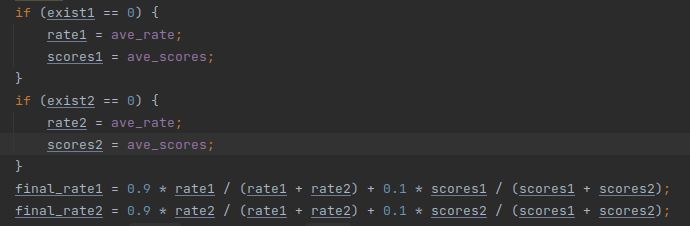
Job3的map

输入是predict文件中的两支需要计算胜率的队伍，注意需要去除第一行的AwayTeam，HomeTeam。然后将<客队名称，主队名称>作为key，value置为“1”，即每场需要预测的比赛是一个键值对。

Job3的reduce

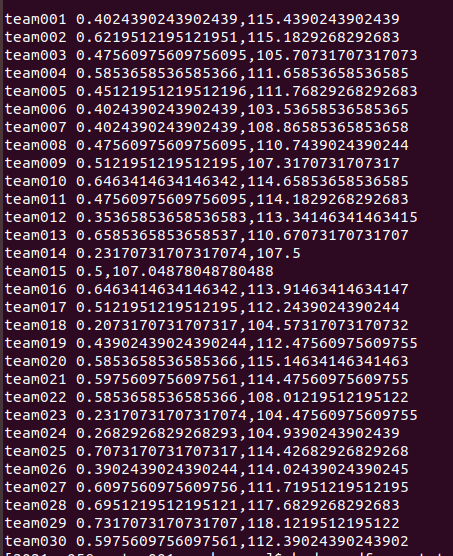
Reduce中加入了哈希表Hashtable<String, List<Double>>来存储job2中的输出。setup从cachefile中读入数据并存入哈希表中，同时计算所有队伍的平均胜率和平均场均进球数。平均胜率和平均场均进球数用来处理没有出现在训练集当中的队伍，那他们的胜率和场均进球数就按照平均的胜率和场均进球数来计算。

Reduce从map中得到两支队伍，再根据两个队伍名在哈希表中查找对应的胜率和场均进球数，如果某队伍没有出现在训练集中，就用平均胜率和平均场均进球数。最后按照9:1的权重，9是胜率，1是场均进球数来计算胜率。

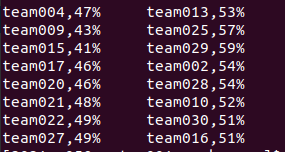


输出是<客队名称，客队胜率 主队名称，主队胜率>

集群测试输出结果：



这是平均胜率和平均场均进球数



这是预测量两队胜率。

比如team004和team013比较，team004的的胜率为58%，场均进球数为111.6；team013的胜率为65%，场均进球数为110.6相差并没有特别大，最后就是47%：53%的胜率比。

最后结果没有出现特别大的胜率预测差别，一方面没有特别强和特别差的队伍胜率预测，另一方面是算法不会出现特别大的胜率差，毕竟以下克上的可能性还是有的，并不会出现较大的预测差。同时舍弃ELO排位方法是篮球的胜率和小分很大程度能反应球队水平，使用elo机制的话每局的加减分较难平衡。

任务四

任务四是评选出表现最好的五名球员，我借鉴了CBA今年的FMVP评选公式\*10（之所以×10是为了避免浮点数计算），即表现质量值SPR=10×得分-10×2分出手数-8×3分出手数-3×罚球出手数+3×进攻篮板数+2×防守篮板数+5×助攻数+15×抢断数+7×盖帽数-12×失误数。

此任务我分成两个job完成，job1用来统计每个球员的SPR值，job2则用来对球员的SPR值进行排序并输出前五名。

Job1的map

输入格式为默认的FilelnputFormat，value为一行的内容，我们把它转成String后再用split将其分解为各个字段，然后根据字段提取中球员的数据贡献。输出的key为球员的姓名，value为某项数据的统计值<数据类型，数据值>（如“score,2”表示得2分，“3-pt,1”表示3分出手一次，“o \_rebound,1”表示进攻篮板一次）。

Job1的reduce

遍历每个key对应的所有value，分别计算上述数据值，再根据公式计算出该球员的SPR值，最终输出的key为球员名，value为该球员的SPR值。

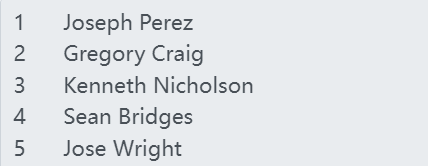
Job2的map

Job2的map是读取job1生成的文件，输入格式为默认的FilelnputFormat，value为一行的内容，由于联盟中的球员不过几百人，所以我们完全可以将所有数据全部放入同一个reduce节点中进行内排序，所以map中输出的key值默认为“1”（目的是将所有的数据传入一个reduce节点），value保持不变，依然为传入map的value（为“球员名\tSPR值”）。

Job2的reduce

Job2的reduce中使用TreeMap<Integer, ArrayList<String>> mp对球员的SPR值进行从大到小排序（我们需要重写compare函数以实现降序排序），我们只需将所有球员的SPR值作为TreeMap的key，球员名作为TreeMap的value中的一个元素，插入到TreeMap中，底层为红黑树的TreeMap就会自动排序，最后我们只需要把排名前五的球员输出即可。输出的key值为球员的排名，value值为球员的姓名。

集群测试结果



任务五

统计两个队的投篮分布，具体来说包括两个队的场均3分出手数、3分命中数、2分出手数、2分命中数、罚球数、罚球命中数以及各项命中率。

Map:

输入格式为默认的FileInputFormat，即key为行偏移，value为一行的内容。用String类的split方法将value分解为各个字段，以PlayBy字段作为输出的关键字，只考虑其值为team025和team028的情况。首先输出<PlayBy, Date>，用于统计该队参加的比赛总数（因为有前提条件一个队一天只打一场比赛）。然后查看ShotType字段，若其不为空，则表示有3分或2分出手，再看ShotOutcome字段，根据是否命中输出“3,1”或“3,0”或“2,1”或“2,0”， “3,1”表示3分命中，“3,0”表示3分miss，2分同理。再根据FreeThrowOutcome字段得到罚球信息，同上输出“1,1”或“1,0”。

Combine:

Map阶段在每一行都会输出一个<PlayBy, Date>，有很多重复的输出。而我们需要的只是场数，即不同Date的数目，因此为了减小传输开销，在Combine中统计所有不同Date的数目，然后发送一个<PlayBy, Count>键值对即可，统计利用的是HashSet。而对于其他类似于<PlayBy, “3,1”>这样的键值对，原样输出。

Reduce:

输入的key为队名（team025或team028），values中包括两种value，一种是count（比赛数），一种是类似于“3,1”“2,0”这样的投篮信息。对所有投篮信息作合并统计，得到3分出手总数、3分命中总数、2分出手总数、2分命中总数、罚球总数、罚球命中总数，分别除以count就得到场均的信息。最后再把命中数除以出手数求得各项命中率。

集群运行结果

