第四节课

开发优化中需要注意的事项

对于读入的数据，做好清洗、转换、分区工作

```
rdd1= sc.textFile("hdfs://test.txt",15)
      .map(_.split("|"))
      .filter{
  //尽量严格过滤
}
.map(id, money)//提前必要的字段，减少数据量
.coalesce(10, True)//分片
```

RDD高效使用

4. 相同数据不要多次IO读取
5. 提高对同一RDD使用次数
6. 对于多次使用的RDD考虑是否持久化

PairRDD状态维护

3. 对于需要多次join的RDD提前repartition
4. 使用PairRDD特有API，不要破坏分区信息
    - mapValue替代map
    - reduceByKey替代reduce
    - flatMapValues替代flatMap

```
val classRDD = rawClassRDD.coalesce(10, True)
val  classRDD = classRDD.mapValue(_+1)
val  classRDD = classRDD.map{line =>
      (line._1, line._2+1)
}
val resultRDD = classRDD.join(studentRDD)
```

使用预聚合功能

使用reduceByKey替代groupByKey

```
rawClassRDD.groupByKey().mapValues{case iter: Iterable[Long] =>
 var number:Long = 0
 iter.foreach(l => number += l)
 number
}

rawClassRDD.reduceByKey(_+_)
```

竞争资源批量处理

2. 使用mapPartitions替代普通map
    - 一次函数调用会处理一个partition所有的数据
    - 适用于访问竞争资源
    - 需要有大内存 或扩大partition

```
    val smallValue = smallRDD.collectAsMap()
val smallBD = sc.broadcast(smallValue)
val resultRDD2 = bigRDD.mapPartitions { iter =>
  val small = smallBD.value
  val arrayBuffer = ArrayBuffer[(String,Long,Long)]()
  iter.foreach{case(className,number) =>{
    if(small.contains(className)){
      arrayBuffer .+= ((className, small.getOrElse(className,0), number))
    }
  }
  arrayBuffer.iterator
}
```

2. 使用foreachPartitions替代foreach

- 一次函数调用会处理一个partition所有的数据
- 适用于访问竞争资源
- 需要有大内存 或扩大partition

```
bigRDD.foreachPartitions(//批量写数据库)
```

数据倾斜

由于存在热点数据，某个key下存在大量记录，在进行计算的时候会出现任务倾斜，某过程有1000个task，999个执行完毕等待1个task长期运行，浪费资源；

热点数据过大，内存不够会发生OOM现象，程序不断的恢复又不停的OOM，最后崩溃退出。

数据倾斜往往伴随shuffle过程，相关API：distinct、 groupByKey、reduceByKey、aggregateByKey、join、cogroup、repartition等

首先查看热点数据：

```
rdd.sample(false, 0.1)
 .countByKey()
 .foreach(println(_))
```

解决方式：

6. 根据业务决定是否可以直接过滤（大部分机器学习在特征工程中需要去噪点数据、裁剪边）
7. 增加partition，提高并行度
8. 利用广播变量调优
9. 拆解热点key
10. 拆解热点key+小表数据扩容

参数调优

资源申请相关

spark.driver.cores
spark.driver.memory
spark.executor.cores
spark.executor.memory

rdd大小、 partition、 spark.executor.memory、 spark.executor.cores联合起来调配

压缩相关
spark.shuffle.compress
spark.shuffle.io.maxRetries
spark.shuffle.spill.compress
spark.broadcast.compress

序列化相关
spark.serializer

shuffle相关
spark.shuffle.manager =sort
spark.shuffle.consolidateFiles

五处错误的代码：

```
  val rawClassRDD = sc.makeRDD(Array("spark", "hadoop", "hive", "yarn", "hbase", "flink", "flink", "flink", "kafka"), 4).zipWithIndex()
   val rawStudentRDD = sc.makeRDD(Array("spark, tony", "hadoop, jack"), 4)
   val studentRDD: RDD[(String, String)] = rawStudentRDD.map{ line =>
     val info = line.split(",")
     info.apply(0) -> info.apply(1)
   }
   val number = sc.makeRDD(Array("spark", "hadoop", "hive", "yarn", "hbase", "flink", "flink","flink", "kafka"), 4).count()
   println("Number of courses: " + rawStudentRDD.count())
   val classRDD: RDD[(String, Long)] = rawClassRDD.map(l => (l._1, l._2+1))
   val resultRDD = classRDD.join(studentRDD)
   resultRDD.foreach(println)
```