

# Spark streaming应用调优

韩红根 2014.09

[www.jd.com](http://www.jd.com)



- 问题1：建立一个数据接收流进行数据接收，数据会存放于接收work及备份work上。在任务调度时考虑到数据本地性，task调度到接收流和副本所在两个work上，其他work空闲，形成计算不均
- 解决：1) 设置多个接收流 2) 通过repartition调节RDD中partition数量，将task调度分散，使得计算均匀。

- 问题2: **reduce task**数目不合适
- 解决: 默认为8, 需根据实际情况进行调节。可调节参数**spark.default.parallelism**。通常, **reduce**数目设置为**core**数目的2---3倍。数量太大, 造成很多小任务, 增加启动任务的开销; 数目太少, 任务运行缓慢。

- 问题3: shuffle磁盘IO时间长
- 解决: 可以设置spark.local.dir为一组磁盘, 并尽量设置磁盘为IO速度快的磁盘。通过增加IO来优化shuffle性能。

- 问题4: map|reduce数量大, 造成shuffle小文件数目多, default: shuffle文件数目为  $\text{map tasks} * \text{reduce tasks}$
- 解决: 通过设置 `spark.shuffle consolidateFiles` 为 true, 来合并shuffle中间文件, 文件数为 `reduce tasks` 数目。

- 问题5: GC或OOM问题严重
- 解决: 调整spark.storage.memoryFraction。  
Default:0.6。Further, 观察app运行过程中的GC实际情况, 进行其他调节。

- 问题6: block not found
- 解决: 调整spark.cleaner.ttl。RDD及元数据的过期时间。

- 问题7： 序列化时间长或结果大。
- 解决： 通过设置spark.serializer为  
org.apache.spark.serializer.KryoSerializer。  
使用广播变量。



- 问题8: 系统吞吐量不高
- 解决: 设置  
`spark.streaming.concurrentJobs.`

- 问题9：单条记录消耗大
- 解决：使用mapPartition替换map.

# 谢谢

We are hiring!  
[ode@jd.com](mailto:ode@jd.com)

