1 / 3

Spark和MapReduce都是可以处理海量数据，但是在处理方式和处理速度上存在着差异

spark 和 mapreduce 的区别

TaskUmbilicalProtocol (Child && TT)

JobSubmissionProtocol (Client && JT)

InterTrackerProtocol (TT && JT)

Map/Reduce 中有3种协议:

NamenodeProtocol (SNN && NN)

ClientProtocol (Client && NN)

ClientDatanodeProtocol (Client && DN)

InterDatanodeProtocol (DN && DN)

DatanodeProtocol (DN && NN)

HDFS中有5种协议

Hadoop的通信是建立在RPC的基础上

Hadoop节点之间的通信协议

排序：需要根据值排序时，需要在 reducer 阶段写出是，将写入的键值对反写

Reducer Join: 在 reducer 对表进行合并

过滤：在 reducer 阶段去除重复

序列化：在 reducer 阶段对map后的数据进行处理操作

修改

Map Join: 合并操作在 Mapper 阶段进行，到 reducer 后已经是合并好的数据

分区过程中没有修改：分区的过程是在 mapper 之后， reducer 之前进行的操作，在reducer 无需操作

未修改

那个过程中 Reduce 没有修改

core-site.xml、hdfs-site.xml、yarn-site.xml、mapred-site.xml, workers

hadoop配置环境的时候需要修改哪些配置文件？

HADOOP 思考题

HADOOP 思考题



2 / 3

规模数据,但对于需要低延迟的实时业务场景来说,并不太合适

Hive 主要采用批处理的方式,每隔一定时间对数据进行一次批量处理。这种方式虽然可以高效地处理大

方面也有一些限制。

Hive 作为一个基于 Hadoop 的数据仓库系统,在处理大规模批量数据方面确实有很多优势,但实时处理

hive 适不适合做实时

驱动器而非内存，在执行过程中失败可以从中断处恢复，Spark是从头重新计算

一个是RDD的血缘机制，一个checkpoint, 失败后可以通过血缘机制重新计算。MapReduce依赖于硬盘

4. Spark 与 MapReduce 的容错不同

能够充分利用，但是这样任务运行的很慢。

MapReduce是细粒度申请资源，提交任务，task自己申请资源自己运行程序，自己释放资源，虽然资源

果申请不到资源就等待，如果申请到就运行task任务，而不需要task再去申请资源。

粗粒度申请资源指的是在提交资源时，spark会提前向资源管理器（yarn，mess）将资源申请完毕，如

3.Spark是粗粒度资源申请，而MapReduce是细粒度资源申请

加载耗时，所以 Spark 跑机器学习算法比较在行（需要对数据进行反复迭代）

载到内存，Hadoop 也是如此，只不过 Spark 支持将需要反复用到的数据给 Cache 到内存中，减少数据

盘的！有一个误区，Spark 是基于内存的计算，所以快，这不是主要原因，要对数据做计算，必然得加

盘，减少了磁盘 IO 的操作。但是，如果计算过程中涉及数据交换，Spark 也是会把 shuffle 的数据写磁

算不涉及与其他节点进行数据交换，Spark 可以在内存中一次性完成这些操作，也就是中间结果无须落

多数情况下可以减少 shuffle 次数。Spark 的 DAGScheduler 相当于一个改进版的 MapReduce，如果计

Spark 计算比 MapReduce 快的根本原因在于 DAG 计算模型。一般而言，DAG 相比MapReduce 在大

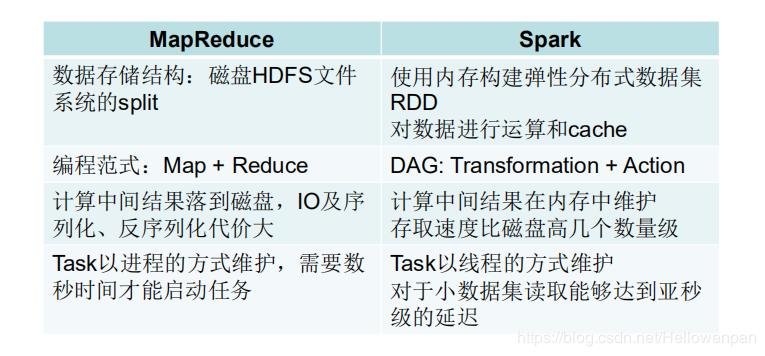
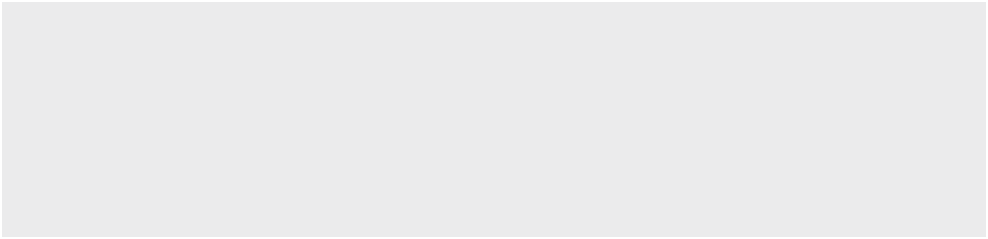
2.Spark在处理数据时构建了DAG有向无环图，减少了shuffle和数据落地磁盘的次数

Spark是将计算的中间结果保存到内存中，可以反复利用，提高了处理数据的性能。

MapReduce是将中间结果保存到磁盘中，减少了内存占用，牺牲了计算性能。

1.spark处理数据是基于内存的，而MapReduce是基于磁盘处理数据的。

HADOOP 思考题



3 / 3

用包括，命名服务，配置管理，分布式一致性锁，HA(High available)

zookeeper是用来管理分布式集群的，是一种管理和协调工具。它提供了文件系统和通知机制，它的主要作

zookeeper的作用与主要功能

HADOOP 思考题