Spark性能改进实现方案

# 范围

## 标识

Spark版本号：Spark1.2.1

文档版本：V1.0

## 系统概述

Apache Spark是一种快速、通用、可扩展的大数据分析引擎。它是不断壮大的大数据分析解决方案家族中备受关注的明星成分，为分布式数据集的处理提供了一个有效框架，并以高效的方式处理分布式数据集。Spark集批处理、实时流处理、交互式查询与图计算于一体，避免了多种运算场景下需要部署不同集群带来的资源浪费。无论是工业界还是学术界，都已经广泛使用高级集群编程模型来处理日益增长的数据。这些系统将分布式编程简化为自动提供位置感知性调度、容错及负载均衡，使得大量用户能够在商用集群上分析超大数据集。

然而，集群环境对于编程来说带来了很多挑战，首先就是并行化：这就要求以并行化的方式重写应用程序，以便利用更大范围节点的计算能力。集群环境的第二个挑战就是对单点失败的处理，节点宕机以及个别节点计算缓慢在集群环境中非常普遍，这会极大影响程序的性能。最后一个挑战就是集群在大多数情况下都会被多个用户共享，那么动态地进行计算资源分配，也会干扰程序的执行。

因此，针对集群环境出现了大量的大数据框架。如Google的Mapreduce、Storm、Impala和GrapLab。但是这些专有系统存在一些不足。针对MapReduce及各种专有系统中的不足，伯克利大学退出了全新的统一大数据处理框架Spark，创新性地提出了RDD概念，在某种程度上Spark是对MapReduce模型的一种扩展。Spark有速度快、易用，可通用性和可融合性等从其他的大数据分析平台中脱颖而出的优点。但是Spark仍存在着它的一些缺陷。

## 文档概述

本文档是对Spark1.2.1版本的各模块的性能优化方案计划书，用于计划和确定对Spark性能的改进。

本文档不涉密，可自由用于学习交流。

## 术语和缩略词

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 术语 | 英文 | 说明 | |
| 1 | Spark | Spark | Apache Spark是一种快速、通用、可扩展的大数据分析引擎。 | |
| 2 | 弹性分布式数据集 | RDD | Resilient Distributed Dataset,即弹性分布式数据集，是Spark中只读的、分区记录的集合。 |
| 3 | 记录 |  |  | |
| 4 | 任务 | Task | 用户程序划分的实际执行单元。 | |
| 5 | Shuffle | Shuffle | 将不同Stage的数据迁移到下一Stage的过程。 |
| 6 | Collect | Collect | 一个对象数组返回给驱动程序 | |
| 7 | 工作节点 | Worker | 执行任务Task的物理单元。 | |
| 8 | 主节点 | Master | 与Slave节点相对应 | |
| 9 | 从节点 | Slave | 与Master节点相对应 |

# 引用文档

《G-系统需求规格说明书-V3.1》

# 改进方案

## 数据编程模型

问题1：单条记录消耗大

改进方式：使用mapPartition替换map，mapPartition是对每个Partition进行计算，而map是对partition中的每条记录进行计算；

问题2：通过多步骤的RDD操作后有很多空任务或者小任务产生

改进方式：使用coalesce或repartition去减少RDD中partition数量；

## 数据存储

问题3：shuffle磁盘IO时间过长

改进方式：设置spark.local.dir为多个磁盘，并设置磁盘为IO速度快的磁盘，通过增加IO来优化shuffle性能；

问题4：collect输出大量结果时速度慢

改进方式：collect源码中是把所有的结果以一个Array的方式放在内存中，可以直接输出到分布式文件系统，然后查看文件系统中的内容；

## 任务执行

问题5：默认reduce task数目不合适

改进方式：需根据实际情况调节默认配置，调整方式是修改参数spark.default.parallelism。通常，reduce数目设置为core数目的2到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6：map|reduce数量大，造成shuffle小文件数目多

改进方式：默认情况下shuffle文件数目为map tasks \* reduce tasks

通过设置spark.shuffle.consolidateFiles为true，来合并shuffle中间文件，此时文件数为reduce tasks数目；

问题7：任务执行速度倾斜

改进方式：如果是数据倾斜，一般是partition key取的不好，可以考虑其它的并行处理方式 ，并在中间加上aggregation操作；

如果是Worker倾斜，例如在某些worker上的executor执行缓慢，可以通过设置spark.speculation=true 把那些持续慢的节点去掉。