计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(8):232−236 [doi: 10.15888/j.cnki.csa.006469]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于关键阶段分析的 Spark 性能预测模型①

葛庆宝[1,2], 陶耀东[2], 高　岑[2], 田　月[2], 孟祥茹[2]

[1](中国科学院大学, 北京 100049)
[2](中国科学院 沈阳计算技术研究所, 沈阳 110168)

摘　要: Spark 作为目前大数据处理领域广泛使用的计算平台, 合理分配集群资源对 Spark 作业性能优化有着重要的作用. 性能预测是集群资源分配优化的基础和关键, 本文正是基于此提出了一种 Spark 性能预测模型. 文中选取作业执行时间作为 Spark 性能衡量指标, 提出了 Spark 作业关键阶段的概念, 通过运行小批量数据集来获取关键阶段的运行时间和作业输入数据量之间关系, 从而构建了 Spark 性能预测模型. 实验结果表明该模型较为有效.

关键词: Spark; 资源分配; 性能预测; 关键阶段

引用格式: 葛庆宝,陶耀东,高岑,田月,孟祥茹.基于关键阶段分析的 Spark 性能预测模型.计算机系统应用,2018,27(8):232–236. http://www.c-s-a.org.cn/1003-3254/6469.html

Performance Prediction Model for Spark Based on Key Stages Analysis

GE Qing-Bao[1,2], TAO Yao-Dong[2], GAO Cen[2], TIAN Yue[2], MENG Xiang-Ru[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: Spark is widely used as a computing platform for large data processing, reasonable allocation of cluster resources plays an important role in the operation of Spark performance optimization. The performance prediction is the basis and key of cluster resource allocation optimization, thus we put forward a Spark performance prediction model in this paper. This paper selects the job execution time as a measure indicator of Spark performance, and put forward the concept of key Stage of Spark job. Finally, we built the model by analyzing relationships between the key Stages and the amount of input data through running a small quantity of data. The experimental results show that the model is effective

Key words: Spark; resource allocation; performance prediction; key stages

1　引言

Spark 是起源于美国加州大学伯克利分校 AMPLab 的大数据计算平台, 它提出的弹性分布式数据集 (Resilient Distributed Dataset, RDD) 概念能够在分布式内存中快速处理大数据, 因此 Spark 在内存处理速度上比 Hadoop MapReduce 要高 100 倍, 除此之外 Spark 的功能涵盖了数据的离线批处理和实时流数据处理, 机器学习, 图计算和 SQL 数据处理等各种类型的操作[1]. 这一系列因素使得 Spark 在大数据领域被越来越广泛的使用. 在实际使用中通常需要根据需求优化 Spark 平台, 但是因为 Spark 的底层运行机制对用户来讲是透明的, 所以如何合理分配集群资源和优化 Spark 作业性能就成为一个值得研究的问题. 性能预测是集群资源分配优化模型的基础和核心, 本文正是基于此提出了一种基于关键阶段分析的 Spark 性能预测模型. 通过对 Spark 作业运行性能进行预测从而合理分配集群资源提高作业运行效率.

① 收稿时间: 2017-12-08; 修改时间: 2018-01-04; 采用时间: 2018-01-08; csa 在线出版时间: 2018-07-28

2 相关工作

近几年, 随着各种大数据计算平台的流行, 为了高效的管理和利用集群资源, 很多学者提出了一些性能预测模型和方案. 文献[2–8]只是部分针对 Hadoop 平台的性能预测研究.

目前针对 Spark 平台的性能预测研究还比较少. Wang KW 等[9]提出了理论分析模拟 Spark 作业的方法预测其性能 (运行时间和内存使用等), 能够获得较好的准确度, 但是需要对作业进行详细的分析. Wang GL 等[10] 提出了使用机器学习的方式预测作业在不同参数下的性能表现, 对比了决策树、线性回归、支持向量机和神经网络方法在预测 Spark 性能方面的准确度, 但是该方法完全采用黑盒方法, 没有结合 Spark 作业的运行特征, 对不同类型的作业性能预测缺少支持. 陈侨安等[11]提出的基于近邻搜索算法的 Spark 参数自动优化, 主要思路是通过提取 Spark 作业的特征, 比如运行日志和 DAG 图等, 然后在历史数据库中查询相似作业, 并把相似作业执行效率最高的配置参数作为当前作业的最优配置参数,这种方法有几个缺陷,第一是通过图的编辑距离来判断 DAG 图的相似性, 计算性较大且图的编辑距离方法本身已经被证明为是一个 NP 难问题; 第二是该方法的相似性判断不一定能够选取出相似作业.

本文在借鉴前人工作的基础上, 通过分析大量的 Spark 作业运行特征, 提出了 Spark 作业关键阶段的概念, 并基于此建立了一种 Spark 性能预测模型. 本文组织结构如下: 第 3 节阐述 Spark 原理, 提出关键阶段的概念, 并对关键阶段和 Spark 性能之间的关系进行分析; 第 4 节介绍通过关键阶段建立 Spark 性能预测模型; 第 5 节通过实验验证该模型, 证明该模型能够得到较好的预测效果.

3 关键阶段分析

3.1 Spark 作业运行时间分析

用户提交的程序被 Spark 分解成 n 个作业 (job), 每个作业又被分解为 m 个阶段 (stage), 各个阶段之间串行执行. 每个阶段中包含若干任务 (task). 为了能够并行进行计算, 每个阶段中的一批任务并行处理其中的可容错弹性分布数据集 (RDD). 因此我们可以将 Spark 应用的执行时间定义为如下公式:

$$T_{\text{application}} = T_{\text{start}} + \sum_{i=0}^{n} T_{\text{job}_i} + T_{\text{end}} \tag{1}$$

$$T_{\text{job}} = \sum_{j=0}^{m} T_{\text{stage}_j} \tag{2}$$

其中, $T_{\text{application}}$表示用户程序运行时间,T_{start}和T_{end}表示程序提交阶段和运行结束后清理阶段运行时间,T_{job_i}表示第 i 个作业的运行时间 ($0 \leqslant i \leqslant n$), T_{stage_j}表示第 j 个阶段运行的时间 ($0 \leqslant j \leqslant m$).

3.2 关键阶段

因为每个阶段内部的任务是按照批次进行的, 同一批次内部并行执行, 并行执行受到集群的资源分配 (机器核数,executor 个数等) 和作业类型等多个因素的影响, 所以不能按照简单批次时间累加. 如果去分析所有的任务的执行时间会消耗大量的计算成本和时间成本. 我们经过大量的实验发现, 在 Spark 作业的每个阶段在不同输入数据量的情况下对作业执行时间产生影响不同, 如图 1 所示. 我们可以发现一个共同特点:有的阶段运行时间在不同的输入数据量下基本保持不变, 有的则会产生较大变化从而影响整个作业的运行时间, 还有的阶段对应的运行时间为 0, 这是因为该阶段的计算结果因为缓存到了内存之中, 在后续需要使用该结果的时候可以直接取出从而不再需要计算. 以图 1(a) 中的 WordCount 程序为例, 我们可以看出 Stage0 阶段的运行时间在整个作业运行中的时间比重较大, 而且随着数据量的变化产生明显的变化. Stage1 所占比例较小, 且基本稳定. 我们通过实验分析其他几种类型的作业, 比如 Sort, PageRank 和 K-Means, 均有相似性质, 因此我们提出了关键阶段的概念.

定义 1. 关键阶段. 对于一个 Spark 作业其关键阶段需要同时满足以下条件:

(1) 该阶段的运行时间大于整个作业运行时间的均值

(2) 该阶段的运行时间随输入数据量的变化而变化的幅度大于该作业每个阶段运行时间变化幅度的均值

3.3 关键阶段和 Spark 性能之间的关系

本文我们选取 Spark 作业的运行时间作为 Spark 性能的衡量指标, 因此本文的目的即预测在不同的输入数据量的情况下 Spark 作业的运行时间. 为了建立该模型, 我们需要分析关键阶段的运行时间和和 Spark 作业不同输入数据量之间的关系.

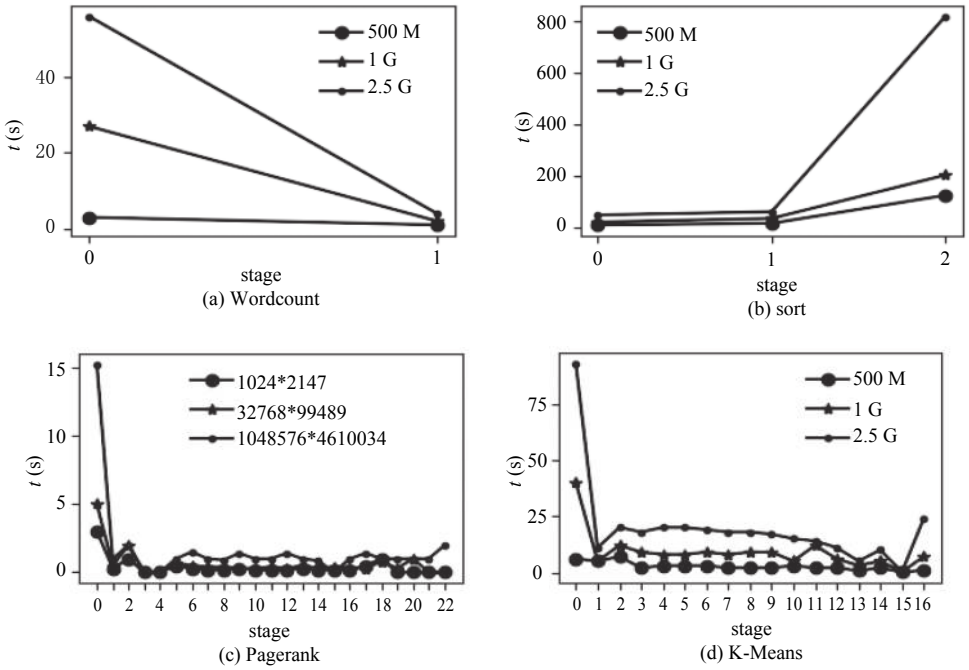

图 1 几种常见 Spark 程序处理在不同输入数据量时的各个阶段运行时间

以 WordCount 程序为例, 我们通过实验得出关键阶段的运行时间在不同输入数据量情况下的关系图, 如图 2 所示.

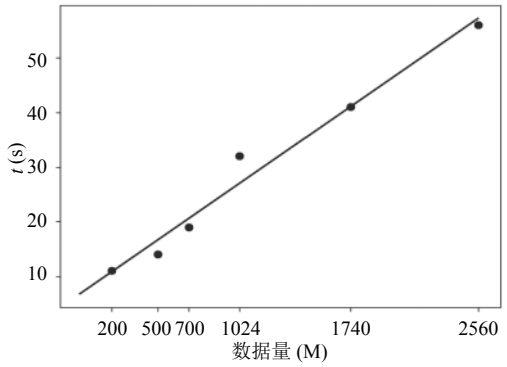

图 2 关键阶段运行时间和不同输入数据量关系图

在图中我们可以看出在 WordCount 程序中关键阶段的运行时间和数据量之间可以拟合成一种线性回归关系, 即

$$T_{\text{stage}}^{key} = \alpha D_{\text{input}} + \beta \tag{3}$$

虽然不同类型的 Spark 作业的资源消耗情况等各种运行情况不尽相同, 比如 CPU 密集型作业和 I/O 密集型作业需要的集群资源不同, 但是我们通过实验发现多数作业的关键阶段运行时间和数据量之间的关系具有回归关系. 所以公式 (3) 可以表达为以下公式:

$$T_{\text{stage}}^{key} = f(D_{\text{input}}) \tag{4}$$

4 基于关键阶段分析的性能预测模型

通过上述两节的分析以及公式 (2), 我们定义对 Spark 作业执行时间的预测公式:

$$T_{\text{job}}^{p} = \sum_{i=0}^{m} T_{\text{stage}_i}^{key} + \sum_{j=0}^{n} T_{\text{stage}_j}^{r} \tag{5}$$

$$T_{\text{stage}}^{r} = \frac{\sum_{i=0}^{N} T_{\text{stage}_i}^{r}}{N} \tag{6}$$

其中, $T^{key}{}_{\text{stage}}$ 表示关键阶段的运行时间, $T^{r}{}_{\text{stage}}$ 表示其它阶段的时间, 这些阶段的运行时间在不同输入数据量的情况下基本可以认为保持不变.

具体建模步骤:

(1) 运行小批量不同大小的数据集, 收集作业的运行信息: 阶段个数, 每个阶段的起止时间等.

(2) 根据算法 1 求出关键阶段.

算法 1. 求解关键阶段

输入: 步骤 1 获取的运行信息.

输出: 关键阶段相关信息.

① 根据每个阶段的起止时间计算每个阶段的运行时间和整个

作业的运行时间均值 T_{mean}.

② 依次判断每个阶段的运行时间是否大于 T_{mean}, 并将大于均值的编号存储到 List1.

③ 遍历 List1 中的编号, 获取对应的每个阶段运行时间的变化幅度和求出总的变化幅度的均值 D_{mean}.

④ 返回 List1 中大于 D_{mean} 的阶段即为所求.

(3) 根据步骤 (1) 的信息结合有关机器学习回归算法获取公式 (4) 的具体表达式.

(4) 根据公式 (5)(6) 获取作业的预测运行时间.

5 实验验证

本文所使用的实验环境的是三个节点组成的 Spark 分布式集群, 集群采用主从架构, 其中的一个节点是主节点, 另外两台为从节点. 节点的操作系统是 CentOS Linux release 7.2.1511 64 bit, 内存为 8 G, 处理器 4 核 4 线程; 使用的 Spark 版本是 2.0.1, Hadoop 版本为 2.7.3, 并且使用 Hadoop 的 YARN 作为分布式集群的资源调度器; java 版本是 1.8.0_131; 基准测试程序 BigDataBench[12]. 实验中我们分别针对 WordCount, PageRank 和 K-Means 程序进行作业性能预测. 首先使用小数据量数据集得出公式 (4) 的表达式, 然后根据该公式预测大数据量情况下作业的运行时间.

针对 WordCount 和 K-Means 程序, 采用的训练数据量是 100 M~1.5 G, 间隔 200 M, 预测 5 G 数据量下运行时间; 针对 PageRank 程序, 采用的训练数据图为节点 1024 到 32 768, 边数为 2147 到 99 489, 预测节点数为 3 048 576, 边数为 4 610 034 图的处理时间. 得到结果如图 3.

如图 3, 实验结果显示预测模型能够整体上反映 Spark 作业的运行态势, 而且可以较为准确的预测出 Spark 作业的运行时间. 同时分析图 3(c) 中的 PageRank 预测结果可以发现后面有几个阶段 (18, 19, 20) 的预测运行时间均为 0, 这是因为实验中运行的小批量训练数据集和验证运行的大数据集的阶段个数不一致导致的. 因为这些阶段不是关键阶段, 也就是在整个作业运行时间中占有极小的一部分比例, 所以对整体运行时间的预测结果影响不大. 将实验结果和文献 9 的结果相对比发现两者所预测的作业执行时间准确度相似, 但是本文所使用的方法不需要人工对作业进行详细的分析从而能具有更好的效率.

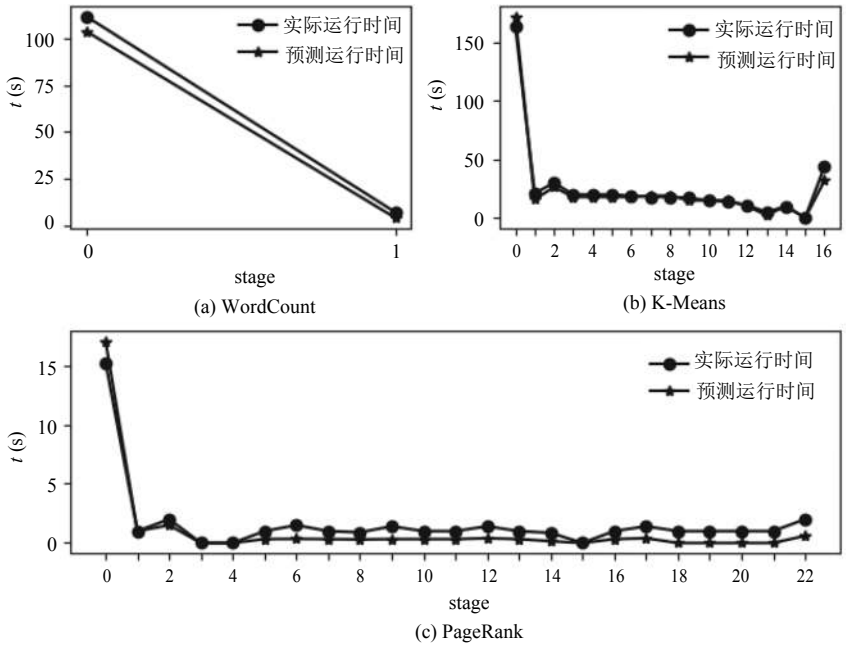

图 3 几种常见 Spark 程序在该模型下预测各阶段运行时间和实际运行时间的对比

6 总结

本文首先通过实验分析 Spark 作业的不同阶段运行时间在整个作业运行时间所占比重的不同, 提出了 Spark 作业的关键阶段这个概念, 然后通过分析 Spark 作业输入数据量和关键阶段运行时间之间的关系提出了一种基于关键阶段分析的 Spark 性能预测模

型, 并且通过实验验证该模型预测结果较为有效. 除此之外, 关键阶段分析对 Spark 平台的其它资源消耗预测 (比如 CPU 和内存等) 具有参考意义.

参考文献

1 Apache Spark. http://spark.apache.org/.

2 Rizvandi NB, Taheri J, Moraveji R, *et al*. On modelling and prediction of total CPU usage for applications in mapreduce environments. Proceedings of the 12th International Conference on Algorithms and Architectures for Parallel Processing. Fukuoka, Japan. 2012. 414–427.

3 李振举, 李学军, 刘涛, 等. MapReduce 性能预测模型构建. 计算机技术与发展, 2016, 26(1): 70–73.

4 Khan M, Jin Y, Li MZ, *et al*. Hadoop performance modeling for Job estimation and resource provisioning. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(2): 441–454. [doi: 10.1109/TPDS.2015.2405552]

5 周世龙, 陈兴蜀, 罗永刚. 基于灰盒模型的 Hadoop MapReduce job 参数性能分析与预测. 四川大学学报 (工程科学版), 2014, 46(Z1): 146–154.

6 Kavulya S, Tan JQ, Gandhi R, *et al*. An analysis of traces from a production MapReduce cluster. Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. Melbourne, VIC, Australia. 2010. 94–103.

7 Yang HL, Luan ZZ, Li WJ, *et al*. MapReduce workload modeling with statistical approach. Journal of Grid Computing, 2012, 10(2): 279–310. [doi: 10.1007/s10723-011-9201-4]

8 Popescu AD, Balmin A, Ercegovac V, *et al*. PREDIcT: Towards predicting the runtime of large scale iterative analytics. Proceedings of the VLDB Endowment, 2013, 6(14): 1678–1689. [doi: 10.14778/2556549]

9 Wang KW, Khan MMH. Performance prediction for apache spark platform. Proceedings of the 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, 2015 IEEE 12th International Conference on Embedded Software and Systems. New York, NY, USA. 2015. 166–173.

10 Wang GL, Xu JG, He B. A novel method for tuning configuration parameters of spark based on machine learning. Proceedings of the 2016 IEEE 18th International Conference on High Performance Computing and Communications, IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems. Sydney, NSW, Australia. 2016. 586–593.

11 陈侨安, 李峰, 曹越, 等. 基于运行数据分析的 Spark 任务参数优化. 计算机工程与科学, 2016, 38(1): 11–19. [doi: 10.3969/j.issn.1007-130X.2016.01.002]

12 詹剑锋, 高婉铃, 王磊, 等. BigDataBench: 开源的大数据系统评测基准. 计算机学报, 2016, 39(1): 196–211. [doi: 10.11897/SP.J.1016.2016.00196]