一种基于交互式的 Hadoop 作业调度算法

吴 佳,苏 丹,李环媛,袁卫国
(国网冀北电力有限公司信息通信分公司 信息通信工程中心,北京 100053)

摘 要:Hadoop 平台中作业调度是一个重要环节。FIFO 是 Hadoop 默认的调度算法,简单易实现且应用广泛,但其在数据的本地化(data locality)这一特性上考虑不足,会引起网络的负载量增大,任务的等待执行时间长,计算资源得不到充分利用等一系列弊端;同时 Map 阶段和 Reduce 阶段资源槽的静态职能形式也更一步加深了这种缺陷。针对这些缺陷,从数据的本地性、任务分配的角度出发,提出了一种基于主从节点间交互的作业调度算法(Interactive Scheduler,IS)。该算法是对 FIFO 的一种改进,同时也使不同资源槽之间可以动态转换,提高了资源的使用率。通过实验对比,结果表明 IS 调度算法对 Hadoop 平台的作业调度效率有显著的提升。

关键词:Hadoop;MapReduce;交互式;slots 资源槽;IS 调度
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2016)11-0045-04
doi:10.3969/j.issn.1673-629X.2016.11.010

An Job Scheduling Algorithm for Hadoop Based on Interaction

WU Jia,SU Dan,LI Huan-yuan,YUAN Wei-guo
(Center of Information and Communication Engineering,State Grid Jibei Information & Telecommunication Company,Beijing 100053,China)

Abstract:Job scheduling is an important part of Hadoop. FIFO,as a scheduling algorithm by Hadoop,is simple and easy to achieve and widely used,but it is lack of consideration in the characteristic of data locality,that will cause network transmission increased and task waiting long execution time and computing resources cannot be fully utilized and a series of drawbacks. Meanwhile the static function of resource slots in Map and Reduce stages further increases the defects. So a job scheduling algorithm (Interactive Scheduler,IS) based on interacting the master node and slave nodes from the data locality and tasks allocation is proposed,which is improvement for FIFO,and realizes the dynamic conversion of map slots and reduce slots,and increases the usage of resources. Through the comparison of experiment,it proves that the IS has a great improvement in job scheduling for Hadoop.

Key words:Hadoop;MapReduce;interaction;slots;IS scheduling

0 引 言

随着互联网的高速发展及用户量的迅速增长,数据量的产生速度也呈现出爆炸性增长,使得大数据成为时下的热门搜索词。面对越来越庞大的数据量,传统的大机器的处理方式显得越来越力不从心。随着软件技术的飞速发展,分布式计算的概念再一次被提上日程。它是一种低成本、高效率的处理方式,通过将大量廉价的 PC 彼此连接起来组成集群的方式进行海量数据的存储与分析。简单地说它是一种在数据处理领域从“猛虎到群狼”的策略的转变。在众多的分布式处理平台中,Apache Hadoop[1] 无疑是目前最活跃、应用最广泛的一个,同时也是 Apache Software Foundation 下的开源项目之一。Hadoop 平台对开发者隐藏了底层的实现细节。在此平台上,只需考虑算法的本身而不用关注平台的底层实现。用户只需要根据自己的需求编写 Map 和 Reduce 函数就可实现分布式的应用。因此,众多的云计算运营商及 Hadoop 爱好者积极投入到 Hadoop 的阵营中,也使得 Hadoop 的用户数和活跃度在不停攀升。

Hadoop 中有三种调度方法:FIFO Scheduler[2]、Fair Scheduler[3]、Capacity Scheduler[4]。其中默认的调度算法是 FIFO。Fair Scheduler 和 Capacity Scheduler 都是基于多用户、公平性的调度器,区别只是在调度策略上有所不同。Capacity Scheduler 是 Yahoo 公司开发

收稿日期:2016-01-29 **修回日期**:2016-05-13 **网络出版时间**:2016-10-24
基金项目:国家发改委高科技产业化项目(发改高技[2009]1365 号)
作者简介:吴 佳(1986-),女,工程师,硕士,研究方向为分布式系统、中间件。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20161024.1117.070.html

的一种基于计算能力的调度算法,它以队列的形式分配资源,通过对内存的约束来限制每个用户占用的资源数;Fair Scheduler 由 Facebook 公司开发,为了保证公平性,该算法采用资源池的形式组织作业,具有小作业快速响应、大作业确保性能的特点。FIFO 是一种批处理调度器,所有的作业被提交到一个作业队列,依照先来后到的顺序将每个作业切分成不同的任务,再依次对任务进行调度。

FIFO 调度算法的优点很明显:简单易实现,对 JobTracker 的负担小;但也存在很多不足:任务分配时只是尽量保持本地性,任务槽是面向职能而非作业。因此,众多学者试图从各个方面对此进行改进。文献[5]提出的一种调度算法考虑了当任务调度器不能选择 Data-local 任务时是否允许分配 Non-Local 任务;文献[6-7]提出了一种基于 Map 任务节点数量和数据片复制模式的 Map 任务选择的调度算法;Delay Scheduler[8-10]解决了由非本地数据作业调度引起的局部性问题;Dynamic Proportional Scheduler[11-13]支持用户优先级的改变,通过计算动态为用户按比例分配任务;文献[14]提出了一种基于截止时间的实时调度算法;文献[15]提出了一种在异构环境下基于 MapReduce 任务调度改进机制;文献[16]提出了基于改进遗传算法的 Hadoop 作业调度;文献[17]解决了短作业执行性能优化。但是这些算法都没能兼顾任务分配时数据的本地化特性及 TaskTracker 节点在任务分配中的能动性。基于此,文中提出了一种基于交互式的调度算法(Interactive Scheduler)——IS 调度算法。

1 IS 调度算法的相关定义

传输时间:DataNode 节点将数据片(split)从当前节点传输到等待执行任务的节点所需的时间,用 t_1 表示,则:

$$t_1 = \frac{s}{v} \tag{1}$$

其中,s 表示数据块的大小;v 表示交换机的传输速率。

等待时间:TaskTracker 节点处于执行状态的任务的剩余执行时间,用 t_2 表示,则:

$$t_2 = \mathrm{MAX}\{T_1, T_2, \cdots, T_N\}(1 - w) \tag{2}$$

其中,考虑到不同任务之间的差异性,$\mathrm{MAX}\{T_1, T_2, \cdots, T_N\}$ 表示该 TaskTracker 节点执行队列中执行时间的最大值;w 表示当前任务完成百分比。

Slot 资源槽是 MapReduce 中执行任务的基本单位。默认的配置中每个节点分配 2 个 map slot 和 1 个 reduce slot,且 map slot 只负责执行 map 任务,reduce slot 只负责执行 reduce 任务。slot 的数量可在一定程度上代表集群的计算规模。

有效 slot:map 或者 reduce 任务中处于执行状态下的 slot 数量。假设集群 A 中节点数量为 n,那么默认有效 map slot 的数量 $Q_m \in [0, 2n]$,有效 reduce slot 的数量 $Q_r \in [0, n]$。

有效 slot 率:map 或者 reduce 任务中处于执行状态下的 slot 数量与集群中对应总 slots 的比值,其中:

$$l_m = \frac{Q_m}{Q} * 100\% \tag{3}$$

$$l_r = \frac{Q_r}{Q} * 100\% \tag{4}$$

其中,l_m 表示 map 任务的有效执行率;l_r 表示 reduce 任务的有效执行率;Q_m 和 Q_r 分别表示集群中 map slot 或者 reduce slot 的总数。

2 IS 调度算法

2.1 IS 调度算法的设计目标

FIFO 调度策略是作业按到达的顺序依次排序等候调度。被调度的作业被切分成任务片依次被分配到 TaskTracker 节点执行。等到这个作业全部执行完才进行下一作业的调度。整个过程是一个串行的过程。因此存在以下弊端:

(1)任务片被分配到 TaskTracker 节点执行时,不一定能够保证数据的本地性,而执行非本地性任务需要通过网络拷贝数据,这样会加重网络的负载。

(2)由于机器处理速度的差异性、任务的不确定性、资源的静态分配等,一部分机器在处理完 map 任务后处于等待状态,这对计算资源是一种浪费。

据此,提出的交互式调度算法的主要目标是对 JobTracker 和 TaskTracker 间主从结构的一种改进,充分发挥 TaskTracker 节点在任务分配中的能动性,旨在对于 TaskTracker 节点中不具有数据本地性的任务,通过与 JobTracker 节点的交互协商的方式,选择最佳的任务分配给该 TaskTracker 节点;其次在集群初始化时,资源槽全部设置为 map slot,而对 reduce slot 的初始值设置为 0,在任务的执行过程中由 map slot 动态转换为 reduce slot。这样一方面可以有效减少 map 阶段产生的中间数据在网络中的传输量,另一方面也提升了资源的利用率。

2.2 HDFS 与 IS 的结合

由于 HDFS 在存放副本时对数据的容错性与可靠性进行了充分的考虑,因此,在 IS 算法中要充分利用这一特性来达到调度目标。同时为了避免跨数据块(block)的任务,Map 任务片(split)的大小要与 HDFS 数据块的大小保持一致。假定在图 1 所示的网络环境中,有 Rack1、Rack2 两个机架通过交换机 1 和交换机 2

互联,采用 HDFS 默认的存储策略,数据块布局会随着节点的意外失效和负载均衡动态改变。在 IS 算法中,当数据不具有本地化特性时,会对存储该数据的最“邻近”节点在数据块传输时间和等待时间两方面的大小进行权衡。在此对这种“邻近”节点有一个距离上的定义。

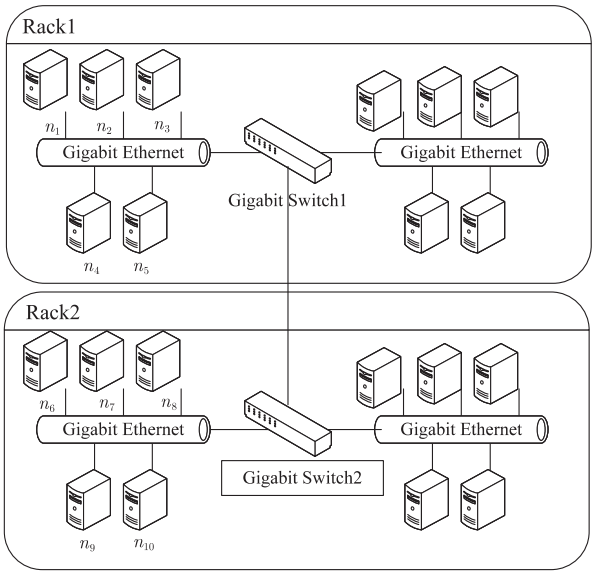

图 1　集群拓扑图

在保证不失一般性的前提下,依默认存储规则任选三个节点。在此取 n_1,n_5,n_{10} 三个节点,数据块 D 在这三个节点均有备份。d 表示连点间的距离。则有:

$$f(d)=\begin{cases} d(n_1,n_1)=0 \\ d(n_1,n_5)=1 \\ d(n_1,n_{10})=2 \end{cases}$$

其中,0 表示同一节点中不同进程间的距离;1 表示同一机架间节点的距离;2 表示不同机架间节点的距离。

JobTracker 在对邻近节点权衡时以距离为依据从大到小依次选取。

2.3　IS 调度算法的数据流程

数据从 Input 到 Output 要经过 4 个过程:

(1)从 Input 到 Map 阶段存在数据本地性的问题。若数据分片在当前 TaskTracker 节点有备份且被分配到该节点执行,这是一种最优的情况;否则转下步。

(2)若该 task 被分配到其他不存在备份数据的节点,则计算“邻近”节点等待时间和传输时间。当“邻近”节点的等待时间小于传输时间,则为“邻近”节点保留此数据块,等待当前任务完成后进行下次调用。否则数据块传输到当前等待分配任务的节点。

(3)如果上述条件都不满足,则认为作业进程已进入到 Reduce 阶段。期间 map slot 转化为 reduce slot 会处理 flush 到本地磁盘的中间数据及 RPC 到本地的数据。

(4)最终所有节点完全进入 Reduce 阶段,处理 Reduce 任务及输出操作。

数据的整体流程图如图 2 所示。

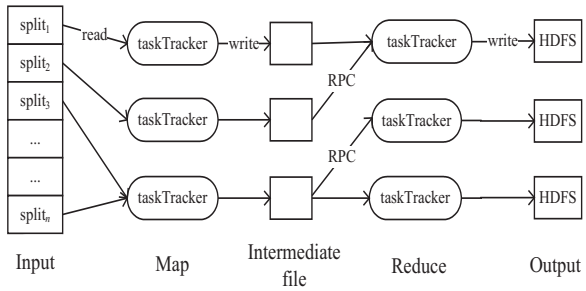

图 2　MapReduce 数据流程图

2.4　IS 的执行流程

JobTracker 与 TaskTracker 之间依据数据的本地性采取一种协调的方式分配任务。首先先试探性地按 FIFO 的顺序分配,在 TaskTracker 节点不满足本地性的情况下反馈回 JobTracker 节点调整再分配,而不是一次性地根据请求的次序依次分配任务。流程图如图 3 所示。

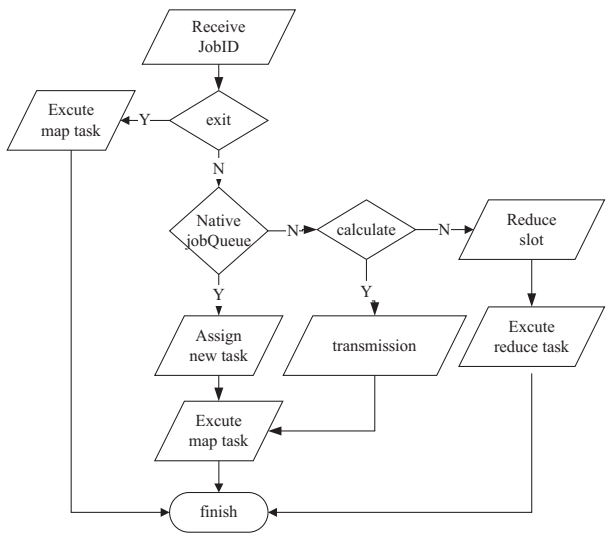

图 3　调度流程图

执行步骤:

(1)TaskTracker 节点接到任务后,首先判断任务对应的数据块是否存在,若存在则直接执行;否则转下步。

(2)TaskTracker 通过心跳把信息反馈回 JobTracker 节点。JobTracker 节点会依据数据本地性在 JobQueue 中重新选择任务。若存在则直接分配执行;否则转下步。

(3)JobTracker 节点会依据距离选择最“邻近”节点计算等待时间与传输时间。若传输时间小于等待时间,则传输数据执行任务,否则转下步。

(4)上述条件都不满足时说明当前 Job 的任务片

正在执行态或即将被执行。总体上 Job 处于即将完成 map 阶段的时期,此时闲置的 map slot 转换为 reduce slot 来执行 reduce 任务。

(5)所有任务执行完成,等待下一作业的到来进行重新的调度分配。

3 实 验

首先用有效 slot 和有效 slot 率预估 FIFO 算法与 IS 算法之间的优劣。从经验中表明 FIFO 的有效 map slot 率与集群的规模有关且是一种反比关系;IS 调度算法是在 FIFO 的基础上着重考虑数据的本地性及资源槽的动态性,因此 slot 率基本会在一个较高的范围内波动,且在 Map 阶段和 Reduce 阶段有一定的并发,一定程度上会缩短任务的执行时间。

实验环境是由 9 个节点组成的 Hadoop 集群。采用 Hadoop1.1.2 版本。测试数据由 RandomWriter 随机生成。其中主节点和四个从节点在一台宿主机中,其余四个节点在另一台宿主机中。两台宿主机通过千兆交换机互联。

通过 slave 节点的有效 slot 数量、总体执行时间及 CPU 的利用率对两种算法进行对比。在一个确定的集群中,有效 slot 率是由有效 slot 数量确定的。因此只需要取有效 slot 数量一个参数进行对比。

集群中每个 slave 节点上运行的 TaskTracker 通过相关的"/proc/stat"命令在预设时间段监控 CPU 的状态。"/proc/stat"中显示了所有 CPU 及每个 CPU 的所有活动信息。在此取其中的 I/O 等待时间信息 iowait 并采集样点对 CPU 的使用率进行计算。对于不同的数据分片设置不同的阈值。当 I/O 等待时间超过指定的阈值,认为此时的数据处于传输中;否则就认为是有效的 slot。

图 4 为在处理同一任务中两种算法的有效 slot 的数量概图。总体上 IS 的有效 slot 数量是多于 FIFO 的,但两者间的差距不大。这是因为节点的数量比较少,即使按顺序依次分配也会有很大的概率选中具有数据本地性的节点;但是在节点较多的集群上,随着这种概率的降低,IS 算法的优势才能体现得更好,两者间的差距才能拉大;同时,还可以看出 IS 的结束时间比 FIFO 提前,这也在一定程度上体现了算法的优势。

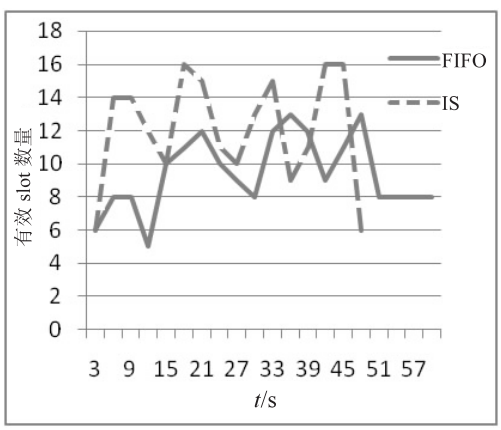

图 4 有效 slot 对比图

图 5 是对同样数据量完成时间的测试的归一化表示。首先,在执行时间上显然是 IS 快于 FIFO;另一方面可以看出,随着数据量的增大执行时间差呈增大的趋势,这同时也说明了算法在大数据量上的优势。

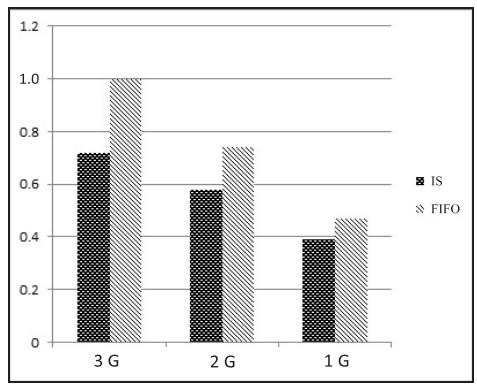

图 5 不同数据量执行时间对比图

4 结束语

文中提出了一种交互式作业调度算法(IS),该算法在主从节点之间以一种交互机制来最大化数据的本地性,减少网络中数据的传输量,加快任务的完成速度,是对 FIFO 算法的一种优化改进,对作业调度中任务执行时间有很大提升。但该算法还有一些瑕疵,几个参数计算式在精确度上难以把握,会在一定程度上影响任务的分配;同时过多的权衡计算也加大了主节点的工作量。这是下一步研究的重点。

参考文献:

[1] ApacheHadoop. Hadoop[EB/OL]. 2015-12-28. http://hadoop.apache.org.

[2] He C, Lu Y, Swanson D. Matchmaking: a new MapReduce scheduling technique[C]//Proceedings of IEEE third international conference on cloud computing technology and science. [s.l.]: IEEE, 2011: 40-47.

[3] Zaharia M, Borthakur D, Sarma J S, et al. Job scheduling for multi-user mapreduce clusters[R]. Berkeley: University of California, 2009.

[4] Raj A, Kaur K, Dutta U, et al. Enhancement of Hadoop clusters with virtualization using the capacity scheduler[C]// Third international conference on services in emerging markets. [s.l.]: IEEE, 2012: 50-57.

[5] Zhang Xiaohong, Feng Yuhong, Feng Shengzhong, et al. An effective data locality aware task scheduling method for MapRe-

(下转第 53 页)

测率,其中有主观因素和客观因素。例如,待检测图像中人的眼睛较小,会在欧拉检测过程中被排除掉,从而影响检测率,如图 6(b)所示;有些图片中人的姿势不规则,也会导致最后的检测率下降。

表 2　实验数据与对比

方法	样本人脸数	检测正确数	检测率/%
基于肤色信息的人脸检测	300	259	86.3
基于肤色信息和模板匹配的人脸检测	300	291	97

4　结束语

文中给出了一种结合肤色信息和模板匹配的人脸检测方法。实验结果表明,此方法较传统方法检测率有所提高。但检测率还有待于进一步提升,候选区域筛选方法需要做出进一步改进。

参考文献:

[1] 陈兆华,徐汀荣,韩志远. 基于运动信息的快速局部遮挡人脸检测[J]. 计算机应用与软件,2013,30(1):63-66.

[2] Jones M J,Rchg J M. Statistical color models with application to skin detection[J]. International Journal of Computer Vision,2002,46(1):81-96.

[3] Tofighi A,Monadjemi S A. Face detection and recognition using skin color and AdaBoost algorithm combined with Gabor features and SVM classifier[C]//International conference on multimedia & signal processing. [s. l.]:[s. n.],2011:141-145.

[4] 王　蓉,李丽华,杨晓刚. 肤色信息在人脸检测中的应用研究[J]. 中国人民公安大学学报:自然科学版,2012,18(3):75-77.

[5] 刘喜荣. 基于肤色模型和模板匹配的人脸检测研究[D]. 太原:太原科技大学,2010.

[6] 陈冠潼. 人脸检测与识别算法的研究与应用[D]. 大连:大连理工大学,2013.

[7] Ghazali K H B,Ma J,Xiao R. An innovative face detection based on YCgCr color space[J]. Physics Procedia,2012,25:2116-2124.

[8] Park J H,Choi H C,Kim S D. Bayesian face detection in an image sequence using face probability gradient ascent[C]//IEEE international conference on image processing. [s. l.]:IEEE,2005.

[9] 郭　佳,刘晓玉,吴　冰,等. 一种光照不均匀图像的二值化方法[J]. 计算机应用与软件,2014,31(3):183-186.

[10] 武　瑛. 形态学图像处理的应用[J]. 计算机与现代化,2013(5):90-94.

[11] 李凤慧. 基于数学形态学的图像噪声处理[J]. 信息技术,2006,30(6):45-46.

[12] 王　骅. 彩色图像中人脸检测与跟踪研究[D]. 南京:南京理工大学,2009.

[13] 卢绪军,赵勋杰. 一种基于肤色和模板匹配的人脸检测方法[J]. 计算机应用与软件,2011,28(7):112-114.

[14] Kasturi S,Vanjare A,Omkar S N. Automatic human face recognition using multivariate Gaussian model and fisher linear discriminative analysis[J]. ICTACT Journal on Image and Video Processing,2014,5(1):899-902.

(上接第 48 页)

duce framework in heterogeneous environments[C]//Proceedings of international conference on cloud and service computing. Hong Kong:IEEE,2011:235-242.

[6] Ibrahim S,Jin H,Lu L,et al. Maestro:replica-aware map scheduling for mapreduce[C]//Proceedings of 12th IEEE/ACM international symposium on cluster,cloud and grid computing. Ottawa,Canada:IEEE,2012:435-442.

[7] 郑晓薇,项　明,张大为,等. 基于节点能力的 Hadoop 集群任务自适应调度方法[J]. 计算机研究与发展,2014,51(3):618-626.

[8] Zaharia M,Borthakur D,Sen Sarma J,et al. Delay scheduling:a simple technique for achieving locality and fairness in cluster scheduling[C]//Proceedings of the 5th European conference on computer systems. New York,NY,USA:ACM,2010:265-278.

[9] 李丽英,唐　卓,李仁发. 基于 LATE 的 Hadoop 数据局部性改进调度算法[J]. 计算机科学,2011,38(11):67-70.

[10] 宁文瑜,吴庆波,谭郁松. 面向 MapReduce 的自适应延迟调度算法[J]. 计算机工程与科学,2013,35(3):52-57.

[11] Sandholm T,Lai K. Dynamic proportional share scheduling in Hadoop[C]//Job scheduling strategies for parallel processing. Berlin:Springer,2010:110-131.

[12] 邹伟明,于　炯,英昌甜,等. 基于动态等待时间阈值的延迟调度算法[J]. 计算机应用研究,2012,29(11):4073-4078.

[13] Kurazumi S,Tsumura T,Saito S,et al. Dynamic processing slots scheduling for I/O intensive jobs of Hadoop MapReduce[C]//Proceedings of the third international conference on networking and computing. [s. l.]:[s. n.],2012:288-292.

[14] Kc K,Anyanwu K. Scheduling Hadoop jobs to meet deadlines[C]//Proceedings of the 2nd IEEE international conference on cloud computing technology and science. [s. l.]:IEEE,2012:388-392.

[15] 何　翔,李仁发,唐　卓. 一种异构环境下的基于 MapReduce 任务调度改进机制[J]. 计算机应用研究,2013,30(11):3370-3373.

[16] 徐　肖,胡吉明. 一种 Hadoop 中基于改进遗传算法的作业调度算法[J]. 计算机技术与发展,2013,23(3):10-13.

[17] 顾　荣,严金双,杨晓亮,等. Hadoop MapReduce 短作业执行性能优化[J]. 计算机研究与发展,2014,51(6):1270-1280.