

文章编号: 2096-1472(2016)-12-36-03

## Hadoop集群环境下本地性调度算法改进

王越峰<sup>1</sup>, 陈福洪<sup>2</sup>

(1.沈阳工业大学, 辽宁 沈阳 110023;

2.辽宁大学, 辽宁 沈阳 110136)

**摘要:** Hadoop集群环境下本地性调度算法是提高数据本地性的算法。算法本质是提高数据本地性, 减少数据传输时间, 减少集群的网络I/O, 提高资源利用率。由于调度算法采用FIFO方式, 当前作业数据量大时将影响其他紧急性高的作业响应时间, 降低系统性能。本文提出一种新的调度策略, 即在保证原算法数据本地性的前提下, 集成静态优先级的抢占调度策略。实验结果表明, 在相同的数据集上, 采用集成静态优先级抢占的调度策略, 优先级高的作业响应时间较优先级低的作业响应时间减少。

**关键词:** 数据本地性; 静态优先级抢占; 作业响应时间

**中图分类号:** TP316.4

**文献标识码:** A

## Improvement of Local Scheduling Algorithm in the Hadoop Cluster Environment

WANG Yuefeng, CHEN Fuhong

(1. Shenyang University of Technology, Shenyang 110023, China;

2. Liaoning University, Shenyang 110136, China)

**Abstract:** Local scheduling algorithm can improve data locality in the Hadoop cluster environment. The essence of the algorithm is to improve data locality, reduce data transmission time, reduce the network I/O of the cluster, and increase the resource utilization rate. As the scheduling algorithm adopts the FIFO mode, the current large amount of data will affect the response time of other jobs with high emergency, which decreases the system performance. This paper proposes a new scheduling strategy, which can guarantee the data locality of the original algorithm and integrate the static priority preemption scheduling strategy. The experiment results show that, on the same data set, adopting the integrated static priority preemption scheduling strategy, the response time of the job with higher priority is less than that of the job with lower priority.

**Keywords:** data locality; static priority preemption; job response time

### 1 引言(Introduction)

在大数据持续发展的今天Hadoop集群环境下调度算法的研究越来越受到重视。对于作业调度算法的改进一般都是为了减少作业的完成时间, 在同样资源的基础上减少系统消耗。例如大多数的算法都要研究数据本地性, 通过减少机架间的网络传输减少传输时间, 提高系统性能。

本文在对已有的调度策略改进时不仅注意提高作业的完成时间, 还注意了系统对作业需要的优先程度, 即一般作业使用FIFO默认调度策略思路。导致一些优先级高的作业没有在需要的时间完成, 造成系统性能降低。在其他操作系统中遇到类似情况, 一般使用优先级抢占策略, 使优先级高的作业可以抢占正在执行的优先级低的作业的资源, 达到可以降低紧急作业的响应时间。本文沿用了这一思路, 提出基于静态优先级的抢占策略。以解决作业优先级不同时如何降低紧

急作业的响应时间等问题。

### 2 Hadoop平台(Hadoop platform)

Hadoop平台是Apache基金组织引入<sup>[1]</sup>, 受到Google开发的GPS(Google File System)的启发, 主要由Hadoop分布式文件系统HDFS(Hadoop Distributed Files System)<sup>[2]</sup>和分布式计算框架MapReduce<sup>[3]</sup>计算架构组成。

Hadoop平台在大数据的背景下发展飞速, 在这种背景下大量数据出现了中心聚集的问题, 每日的数据处理、作业处理在逐步上升。作业调度性能是衡量大型Hadoop平台性能的首要问题, 一个好的调度策略可以减少作业的平均完成时间, 减少系统的负荷, 提高作业的完成效率和准确性, 同时也可以有效使用平台资源<sup>[4]</sup>。在Hadoop平台中, 作业调度策略是通过作业调度器(HadoopTask Schedule)对作业进行调度, 如图1所示。那么设计、使用好的Task

Schedule, 对Hadoop集群平台的性能提高特别主要<sup>[5]</sup>。Hadoop中MapReduce原有三种调度器<sup>[6]</sup>: 默认的调度器FIFO Scheduler(先入先出调度)、计算能力调度器(Capacity Scheduler)、公平调度器(Fair Scheduler)。

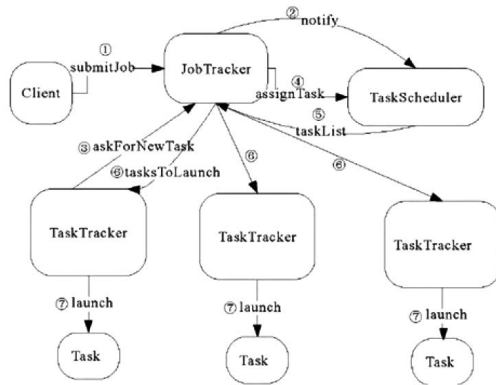


图1 作业调度器工作原理

Fig.1 Job scheduler work principle

默认调度器FIFO是HadoopMap/Reduce计算架构中最早的, JobTracker在进行作业调度时使用的是FIFO(First In First Out)算法。所有用户的作业都被提交到一个队列中, 然后由JobTracker先按照作业的优先级高低, 再按照作业提交时间的先后顺序选择将被执行的作业。优点是调度算法简单明了, JobTracker工作负担轻。同样缺点是忽略了不同作业的需求差异。例如如果类似对海量数据进行统计分析的作业长期占据计算资源, 那么在其后提交的交互型作业有可能迟迟得不到处理, 从而影响到用户的体验。计算能力调度器使用时, 用户需要了解大量系统信息, 才能设置和选择队列; 公平调度器不考虑节点的实际负载状态, 导致节点负载不均匀。所以越来越多的研究者从多个方面对调度算法进行了深入研究。

为了研究资源调度策略, 研究者通过调查大量数据和不同的方向<sup>[9]</sup>从其他研究者的工作中, 将调度分成五类:

- (1)本地性感知调度(Data Locality Aware Schedulers)
- (2)可靠性感知调度(Speculative Execution Based Schedulers)
- (3)资源竞争感知调度(Resource Contention Aware Schedulers)
- (4)性能管理感知调度(Performance Management Based Schedulers)
- (5)能源与完成时间感知调度(Energy and Makespan Aware Schedulers)

MapReduce作业调度算法对集群的性能有着至关重要的影响。通过以下五个标准来比较Hadoop平台性能<sup>[10]</sup>: 平均完

成时间(是一个作业从开始到结束的时间, 同时也是衡量系统性能的最重要的标准)、公平性(调度策略给不同的作业分配的资源是否一致)、数据本地性(研究调度策略的另一重要指标, 是否在存储数据节点上处理任务)、调度时间(调度策略的开销)、调度策略是否达到客户对系统资源的配额。然而, 这些性能标准之间又存在互相冲突, 即当提高一些标准时, 会同时降低其他一些标准。在通常情况下, 作业的平均完成时间和数据本地性是每个调度策略都必须优先处理的性能标准

### 3 本地性调度算法(Local scheduling algorithm)

数据本地性是Hadoop集群平台下衡量作业调度器的重要标准。大量的数据在机架之间传输会产生较大的网络I/O, 特别是在多个不同的机架之间传输时延迟更大。这会使作业的平均完成时间降低, 同时还会产生大量的网络传输开销。Palanisamy等人提出Purlicus<sup>[11]</sup>算法, 该算法通过将任务调度和数据放置结合起来的方式, 使Reduce任务的本地性有较大幅度的提高。他指出, 如果不考虑数据的放置策略, 将会很难获得良好的本地性, 因为随机的数据放置策略可能会导致一些节点变得更加拥堵。一个有效的数据放置策略需要将这些特点考虑进去, 尽量将长作业的数据放到负载最小的节点上。但是这种算法仍然没有考虑到Reduce任务的本地性要求。

Hammoud等人提出本地化感知的Reduce任务调度算法LARTS<sup>[12]</sup>(Locality-Aware Reduce Task Scheduling for MapReduce)以解决Reduce任务数据本地性的问题。LARTS在Map任务完成到一定的阈值 $\alpha$ 后启动Early Shuffle机制。这种调度策略利用Early Shuffle的优点并且兼顾了Reduce任务的数据本地性。但是阈值 $\alpha$ 的设定需要根据不同类型的作业设定, 而且存在一定的误差。

### 4 集成静态优先级抢占的本地性调度算法(Local scheduling algorithm with integrated static priority preemption)

对于本地性调度算法来说, 优先强调的是数据的本地性。但是, 无论是单机调度还是集群式调度都会涉及到任务的优先性问题。尤其是在集群环境下的作业调度, 当前作业的Map任务个数多, 需要系统利用大量时间进行处理计算。而后面进入的重要任务一直没办法分配到资源, 使得任务无响应, 严重时甚至会引发系统崩溃。这样正常的本地性调度并不能处理这些问题, 本文提出静态优先级抢占式本地性调度算法。

集成静态优先级抢占的本地性调度策略, 为每一个提交的作业都设置一个静态优先级, 而被设置的静态优先级意味

着作业的紧急程度。按照优先级抢占策略,紧急程度高的作业有着较高的优先级,它可以抢占紧急程度低的且优先级低的作业的处理资源。使得调度策略更加的有针对性,提高调度策略对高优先级任务的关注,使计算资源优先分配。确保紧急任务紧急处理,减少高优先级任务的响应时间。

一般的本地性调度算法都是使用FIFO算法的调度方式,也就是先到的作业先进行处理,这样的调度算法缺少对任务紧急程度的关注。所以集成静态优先级抢占的本地性调度策略首先要对优先级进行定义。每个作业在提交时设置独立的参数staticpriority,用来表示作业的紧要程度。作业越紧要越优先staticpriority值越高。

但是如果仅仅考虑到作业的优先性问题,有可能导致作业优先级低且数据量很小的作业一直被优先级高数据量很大的作业抢占,导致优先级低的作业一直无法执行。所以本文在定义优先级的时候加入新的参数作业的等待时间waittime。

$$\text{waittime} = \text{nowtime} - \text{submittime} \quad (1)$$

在公式(1)中nowtime和submittime分别表示系统的现在时间和作业的提交时间,通过两者做差的方式得出作业在作业池中的等待时间。

为了防止上文中提到的优先级低的作业无法执行的问题为作业的优先级加入等待时间这个影响因素。但是即使加上了作业的等待时间也会出现等待时长过长的的问题。比如优先级较高的作业数据非常大,Map任务数量也较多。系统在通过原本地性调度策略后,作业的处理时间也非常大。在处理的过程中可能会有优先级相同且数据小,Map任务个数少的作业等待时间变长。在Hadoop集群环境下不能像其他系统一样直接抢占运算资源,因为其中涉到了Map任务完成后的中间值问题,和Reduce任务的中间拷贝等问题。无法直接抢占原有作业的运算资源。所以作业池中的优先级定义就特别重要。在加入等待时间的基础上再加入作业的估计执行时间estimatetime如公式(2)。

$$\text{priority} = \alpha \times \text{staticpriority} + \beta \times \text{waittime} + \gamma \times \text{estimatetime} \quad (2)$$

$$\alpha + \beta + \gamma = 1 (\alpha > 0, \beta > 0, \gamma > 0) \quad (3)$$

priority是调度算法的最终优先级。 $\alpha$ 、 $\beta$ 、 $\gamma$ 表示其中各项参数所占比例,对于不同种的数据类型和作业将取不同的数值,以达到对作业优先性能的标准。其中estimatetime的确定要根据不同的本地性调度算法出发,针对算法的本地性调度得出估计时间。

## 5 实验结果及性能分析(Experimental results and performance analysis)

本文通过虚拟机的方式搭建异构测试环境。定义两个机

架,每个机架5台虚拟机,每个虚拟机分配512MB内存。测试作业为WordCount。通过给不同大小的作业设置不同的静态优先级实验比较算法间作业的响应时间。

提交五个大小为5G的WordCount作业,静态优先级分别设置为1、2、3、4、5,作业编号为1、2、3、4、5。提交五个大小为10GB的WordCount作业,静态优先级分别设置为1、2、3、4、5,作业编号为6、7、8、9、10。

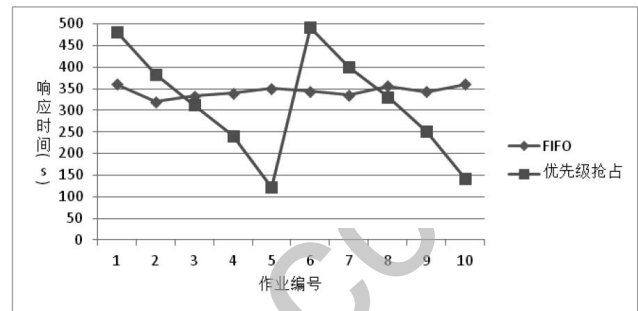


图2 比较不同算法的响应时间

Fig.2 Compare the response times of different algorithms

通过图2可以观察到编号为5、10的响应最快,其次是编号4、9,响应时间最长的是编号1、6。这样的结果可以证明前面的想法,作业的响应时间和作业的静态优先级设置有关,通过实验可以发现编号5、10的作业优先级最高,调度策略将优先处理这些作业,使得调度算法在实际上将资源倾斜。而两种作业之间比较5GB的作业处理估计时间短,所以响应时间要比10GB的作业短。这也证明了之前的想法,相同情况下执行时间小的作业优先处理。

## 6 结论(Conclusion)

本文分析了Hadoop集群下对数据本地性调度的改进,在保证原算法的数据本地性的前提下,指出可以通过集成静态优先级抢占的方式提高优先级高的作业响应时间。通过获得静态优先级,计算等待时间等参数,得到作业的优先级。通过优先级分配资源给各个作业,使得作业按照优先级响应。避免高优先级的作业无法执行。通过实验可以发现这种调度策略,基本上达到了要求,即优先级高的作业的响应时间要小于优先级低的,等待时间长的作业对应的等待时间权值也会增加,而执行时间小的作业在相同优先级情况下优先执行。这个算法的设计使紧急程度较高的作业能优先执行,且尽量小的去影响其他作业。

这种集成了静态优先级抢占的本地性调度算法依然存在一些不足,例如添加了抢占机制后增加了系统开销。实验的作业功能和数据类型不全面,在大数据情况下的性能测试还不是很多,实验在普遍性上还有不足。接下来的工作重点可以研究如何降低系统开销,当开销为何值时可被接受等问

题,在更大的实验数据集上进行改进和验证。

## 参考文献(References)

- [1] Thakkar, Shraddha<sup>1</sup>, Patel, Sanjay. Scheduling in Big Data Heterogeneous Distributed System Using Hadoop[C]. Proceedings of International Conference on ICT for Sustainable Development, Gujarat, 2016: 119–131.
- [2] Khan, et al. Data Locality in Hadoop Cluster Systems[C]. Proceedings of 11th International Conference on Fuzzy Systems and Knowledge Discovery, 2014: 720–724.
- [3] Xiong, et al. Optimizing Data Placement in Heterogeneous Hadoop Clusters[J]. Cluster Computing, 2015(18): 1465–1480.
- [4] Hadoop[EB/OL]. <http://hadoop.apache.org>.
- [5] Shvachko K, et al. The hadoop distributed file system[C]. Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologies, IEEE, 2010: 1–10.
- [6] 董西成. Hadoop技术内幕: 深入解析MapReduce架构设计与实现原理[M]. 北京: 机械工业出版社, 2013.
- [7] 胡丹, 于炯. Hadoop平台下改进的LATE调度算法[J]. 计算机工程与应用, 2014, 50(4): 86–89.
- [8] 何文峰. 基于任务特征与公平策略的Hadoop作业调度算法研究[D]. 湖北: 华中科技大学, 2013.
- [9] 燕明磊. Hadoop集群中作业调度研究[J]. 软件导刊, 2015, 14(4): 1–2.
- [10] 储雅, 马廷淮. 云计算资源调度: 策略与算法[J]. 计算机科学, 2013, 40(11): 8–13.
- [11] 陶昌俊. Hadoop平台的作业调度算法[D]. 安徽: 中国科学技术大学, 2015.
- [12] B. Palanisamy, A. Singh, L. Liu, B. Jain. "Purlicious: Locality-Aware Resource Allocation for MapReduce in a Cloud"[C]. Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, 2011.
- [13] Mohammad Hammoud, Majd F. Sakr. Locality-Aware Reduce Task Scheduling for MapReduce[C]. Proceedings of International Conference on Cloud Computing Technology & Science, Beijing, 2011: 570–576.

## 作者简介:

王越峰(1990–), 男, 研究生. 研究领域: 嵌入式系统.

陈福洪(1992–), 男, 研究生. 研究领域: 数据挖掘.

(上接第52页)

效补充。通过开展探究学习、任务驱动、项目竞标、项目开发、验收答辩、成果展示等训练活动,实现“学中做”和“做中学”,达到全面提升学生专业技能、自学能力、团队合作能力、组织管理能力、创新创业能力的目的,促进软件专业教学水平的整体提高。

## 参考文献(References)

- [1] 陈冀东, 范新灿. 软件行业人才培养现状与职业发展策略[J]. 中国高校科技, 2014, 11: 57–59.
- [2] J. Bergmann, A. Sams. Remixing Chemistry Class: Two Colorado Teachers Make Vodcasts of Their Lectures to Free up Class Time for Hands-on Activities[J]. Learning & Leading with Technology, 2013, 36(4): 22–27.
- [3] Bensing Karen McNally. A Guide to Information Resources on Aging and Gerontology[J]. Medical Reference Services Quarterly, 2006(2): 726.
- [4] 郝军涛, 张伟刚. 软件外包人才培养模式研究[J]. 软件工程师, 2010, 04: 56–58.
- [5] 王志华, 孔金生. 提升学生实践能力的应用型软件人才培养[J]. 计算机教育, 2014, 02: 1–5.
- [6] 朱香卫, 等. 高校软件教育的问题分析与对策研究[J]. 计算机时代, 2013, 1: 47–49.
- [7] 谢科, 黄成兵. 创新软件人才培养模式研究[J]. 时代教育, 2013, 10: 5–6.
- [8] 高春静. 用人单位人才需求分析与用人标准研究[J]. 天津职业院校联合学报, 2013, 3: 96–98.
- [9] 刘浩, 唐培和, 朱亚超. 工程型软件人才职业素质培养的研究与实践[J]. 广西民族大学学报(自然科学版), 2012, 2: 98–100.
- [10] 安俊秀, 李超, 卢军. 软件人才职业素质教育模式的探索与实践[J]. 计算机教育, 2009, 8: 17–19.
- [11] S. W. Bonham, D. L. Deardorff, R. J. Beichner. Comparison of Student Performance Using Web and Paper Based Homework in College-Level Physics[J]. Journal of Research in Science Teaching, 2013, 40(10): 1050–1071.

## 作者简介:

黄成兵(1980–), 男, 硕士, 副教授. 研究领域: 移动自组织网络.