文章编号:1673-9469(2015)01-0083-03 doi:10.3969/j.issn.1673-9469.2015.01.022

基于粒子群优化算法的 Hadoop 调度算法研究

刘盼红
(河北工程大学 信息与电气工程学院,河北 邯郸 056038)

摘要: 为提高 Hadoop 平台性能,提出一种基于粒子群优化算法的 Hadoop 调度算法。以粒子位置代表可行的资源调度方案,以任务完成时间及资源负载均衡度作为目标函数,通过粒子群优化算法,找到最优的资源调度方案。实验结果表明,该算法能够很好的平衡资源负载,减少任务完成时间,有效的提高了 Hadoop 平台的性能。

关键词: 大数据; Hadoop; 调度算法; 粒子群
中图分类号: TP3 **文献标识码:** A

Research of scheduling algorithm for Hadoop based on particle swarm optimization

LIU Pan-hong
(School of Information and Electrical Engineering, Hebei University of Engineering, Hebei Handan 056038, China)

Abstract: In order to improve the performance of Hadoop, the paper proposed a new scheduling algorithm based on particle swarm optimization. In this paper, the position of particles represent feasible resource scheduling scheme, the cloud computing task completion time and resource load balancing were taken as the objective function, the optimal resource scheduling scheme was obtained by the particle swarm optimization algorithm. The experimental results show that this algorithm can well balanced resource load and reduce the task completion time and effectively improve the performance of the Hadoop platform.

Key words: big data; Hadoop; scheduling algorithm; particle swarm optimization

近年来,随着互联网、物联网、移动网络、传感网络等各种新技术在各行各业的渗透式应用,数据呈现爆炸式的增长,对海量数据的处理与分析被称为"大数据"[1]。Hadoop,以一种可靠、高效、可伸缩的方式进行数据处理,为处理海量数据提供了一个很好的解决方案[2]。任务调度器是 Hadoop 中最核心的组件,负责整个集群资源的管理和分配。因此调度算法的好坏直接影响到 Hadoop 平台的整体性能和集群的资源利用率[3]。Hadoop 自带的调度算法在处理混合作业时没有考虑资源负载均衡的问题,在调度各节点任务时,没有达到一种最优的状态。基于以上问题,文中考虑到混合作业的特点提出了一种基于粒子群优化算法的调度算法,以便可以更高效全面的进行任务调度,平衡资源负载,减少数据处理时间,提高平台性能。

1 Hadoop 任务调度

1.1 Hadoop 原有的任务调度算法

默认调度器 FIFO。FIFO 将所有用户提交的作业都放到一个队列中,并将作业按照优先级高低和提交的先后顺序进行排序,然后按此队列的顺序执行作业。优点是实现非常简单、调度过程快;缺点是资源的利用率不高。

计算能力调度算法 Capacity Schedule。Capacity Schedule[4] 采用多个队列,每个队列分配一定的系统容量,空闲资源可以动态分配给负荷重的队列,支持作业优先级;优点是支持多作业并行执行,提高资源利用率,动态调整资源分配,提高作

收稿日期:2014-09-23
作者简介:刘盼红(1989-),女,河北栾城县人,硕士,从事大数据处理方面的研究。

业执行效率;缺点是用户需要了解大量系统信息,才能设置和选择队列。

公平调度算法 Fair Schedule。Fair Schedule[5]将作业分组形成作业池,每个作业池分配最小共享资源,然后将多余的资源平均分配给每个作业;优点是支持作业分类调度,提高服务质量,动态调整并行作业数量,充分利用资源;缺点是不考虑节点的实际负载状态,导致节点负载实际不均衡。

2 基于 PSO 的调度算法

粒子群优化算法(Particle Swarm Optimization,简称 PSO)最早是由 J. Kennedy 和电气工程师 R. Eberhart 模拟自然界的生物群体觅食而提出的一种优化方法[6],通过粒子之间的集体协作来交换信息并作出决策。算法采用迭代的方式,使每个粒子向找到的最佳位置和群体中最佳粒子靠近,从而搜索到最优解。

本文将粒子群算法引入到大数据处理框架 Hadoop 中,用于解决任务调度的问题。采用粒子群算法进行作业调度时,任务被作为粒子,资源池即为搜索空间,粒子寻找最优解的过程即为任务调度的过程[7-8],通过为任务赋予其在资源池中的初始位置和速度信息,并对其速度和位置进行迭代更新,最终形成任务和资源的映射关系。

2.1 相关定义

定义 1:集合 $T=\{t_1,t_2,\cdots,t_n\}$ 表示待分配的 n 个任务。

定义 2:集合 $R=\{r_1,r_2,\cdots,r_m\}$ 表示 YARN 资源管理系统中 m 个 Container 资源。

定义 3:矩阵 $ET=\begin{bmatrix} et_{11} & \Lambda & et_{1m} \\ M & O & M \\ et_{n1} & \Lambda & et_{nm} \end{bmatrix}$ 表示任务的执行时间。其中元素 $e_{ij}(i=1,2,\cdots,m;j=1,2,\cdots,n)$ 表示任务 t_i 在 r_j 个 Container 资源上的执行时间。

定义 4:矩阵 $X_{ij}=\begin{bmatrix} x_{11} & \Lambda & x_{1j} \\ M & O & M \\ x_{j1} & \Lambda & x_{ij} \end{bmatrix}$ 表示粒子 i 的位置。其中 $x_{ij}=\{0,1\}$。1 表示任务 t_i 在 Container 资源 r_j 上执行,0 表示不在 Container 资源 r_j 上执行。

定义 5:矩阵 $V_{ij}=\begin{bmatrix} V_{11} & \Lambda & V_{1j} \\ M & O & M \\ V_{j1} & \Lambda & V_{ij} \end{bmatrix}$ 表示粒子 i 的速度。其中 $V_{ij}\in[-V_{\max},V_{\max}]$。

2.2 位置和速度的更新

在粒子群算法中,为了使粒子的位置向量 X_{ij}^k 取值为 0 或 1,引入模糊函数 $\mathrm{sig}(V_{ij}^k)$,它具备把数限制在 [0,1] 的特点。其公式为 $\mathrm{sig}(V_{ij}^k)=\frac{1}{1+\exp(-V_{ij}^k)}$。位置和速度更新公式为

$$V_{ij}^{k+1}=w^*V_{ij}^k+c_1^*\mathrm{rand}_1^*(pb-x_{ij}^k)+c_2^*\mathrm{rand}_2^*(gb-x_{ij}^k)$$

$$X_{ij}^{k+1}=\begin{cases}1, R(0,1)>=\mathrm{sig}(v_{ij}^k)\\ 0, \text{其它}\end{cases}$$

其中,c_1、c_2 为学习因子,w 为权重参数,rand()、$R(0,1)$ 是出于 0 和 1 之间的随机数,pb 和 gb 分别表示个体极值和全局极值所对应的位置,X_{ij}^{k+1} 表示粒子当前位置。

2.3 确定目标函数

Container 资源节点 r_j 上执行的任务时间为 $H(r_j)=\sum_{j=1}^{n}e_{ij}S_{ij}$,任务的完成时间计算公式为 $T=\max(H(r_j))$。

2.4 算法的实现

采用 PSO 进行调度的具体步骤如下:

1)接收用户提交的任务,并把每个任务划分为若干个子任务。

2)初始化算法参数(c_1、c_2、w 等),随机产生初始种群,设定每个粒子的位置和速度初始值,设定最大迭代数。

3)通过目标函数计算每个粒子的适应度函数值,比较粒子当前适应值与自身历史最优值,若当前粒子优于历史最优值,则当前粒子替代历史最优值作为个体最优 pb;否则 pb 取历史最优。

4)比较粒子种群的适应度函数值,从所有粒子中选取最小的作为最优粒子,最优粒子值作为群体最优值 gb。

5)利用速度更新公式,更新每个粒子的速度及位置信息。

6)判断是否达到最大迭代次数,若没有,则重复步骤(3)(4)(5)。达到迭代次数则输出最优结果,算法结束。采用 PSO 进行调度的流程图如图 1 所示。

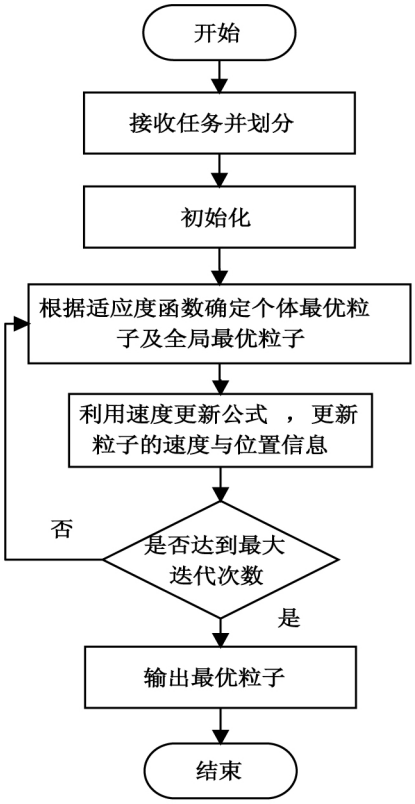

图1 流程图

Fig. 1 Flow chart

3 实验结果

3.1 实验环境

集群环境配置如表 1 所示。

表 1 实验环境配置

Tab. 1 The experimental environment configuration

集群大小	6 台 PC 机组成的小型 Hadoop 集群
主节点	集群中的一台 PC 机
节点	集群中其余的 5 台 PC 机
操作系统	Ubuntu

3.2 实验结果及性能分析

1) 处理相同作业的性能。本实验在 Hadoop 系统中将本算法与 Hadoop 现有的 FIFO 算法、Capacity Schedule 计算能力调度算法和 Fair Schedule 公平调度算法进行对比验证，将此四种算法分别作为调度方法，提交 1 ~ 10 个相同的 WordCount 作业，每个 WordCount 作业处理相同的 128MB 文本文件，记录其作业。其中，运算时间通过代码获取运算开始及结束的计算机系统时间相减得到。为减少误差，同样的实验进行 5 次运算取平均值，效果对比图如图 2 所示。

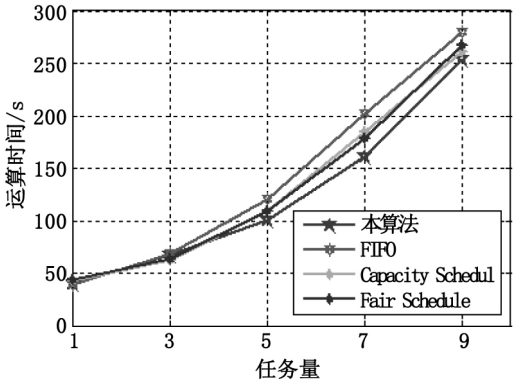

图2 相同作业类型下实验结果对比图

Fig. 2 The comparison figure of the experimental results under the same job types

2) 处理不同类型作业的性能。实验将一个 WordCount 作业、一个 Teragen 作业与一个 Terasort 作业作为一个作业组。WordCount 作业属于统计词频的 CPU 资源密集型作业; TeraGen 作业属于随机生成数据的磁盘 I/O 密集型作业; TeraSort 作业是对输入数据进行排序的作业，它既是 CPU 密集型的也是磁盘密集型的。其中 WordCount 作业处理 128 MB 的文本文件，Teragen 作业生成一个 500 M 的数据集，Terasort 则对一个事先已经生成好的 500 MB 的数据进行排序。同时运行 1 ~ 5 组这样的作业组，记录其平均运行时间。效果对比图如图 3 所示。

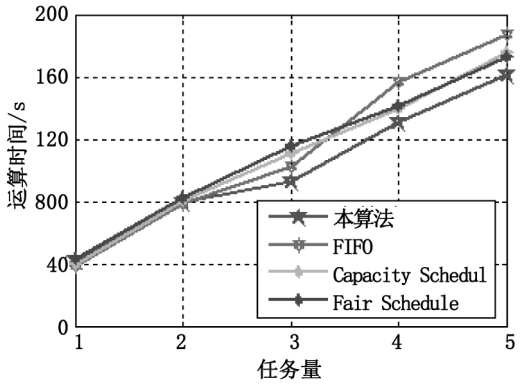

图3 不同作业类型下实验结果对比图

Fig. 3 The comparison figure of the experimental results under the different job types

从图 2、图 3 中可以看出使用本文提出的调度算法平均运行时间是最少的。实验证明本算法针对不论是相同类型的作业还是面对负载不均衡的不同类型的作业都能较好的完成调度任务，使得整体运算效率较高。

4 结语

采用粒子寻优策略，找到调(下转第 95 页)

$R_{C1} = (qL_1/2)\ [1-(a_1/L_1)^2] = (2.15\times 3\,200\div 2)\times[1-(\frac{1}{4})^2] = 3\,225$ N

$P_2 = R_{C1} = 3\,225$ N

$P_n = P_2(1-a_i/L_i) = 3\,225\times(1-\frac{550}{3\,450}) = 2\,710.87$ N

$M_{B2} = -(P_2a_2 + qa_2^2/2) = -(3\,225\times 550 + 2.15\times 550^2/2) = -2\,098\,937.5$ N·mm

$M_n = (qL_n^2/8)\ [1-(a_n/L_n)^2]^2 - P_na_n\ [1-(1+a_n/L_n)^2/2 + a_n/L_n] = 3\,450^2\times 2.15\div 8\times[1-(\frac{550}{3\,450})^2]^2 - 2\,710.87\times 550\times[1-(1+\frac{550}{3\,450})^2/2+\frac{550}{3\,450}] = 2\,311\,730.087$ N·mm

$M_1 = 9L_1^2/8\ [1-(a_1/L_1)^2]^2 = [2.15\times 3\,200^2/8]\ [1-(800/3\,200)^2]^2 = 2\,418\,750$ N·mm

通过单支座不等跨梁多跨铰接力学模型与简支梁力学模型计算结果相比较，选用前者可节约: $\frac{4.3-2.42}{4.3}\times 100\% \approx 43.7\%$，比单支座等跨梁多跨铰接力学模型更为节约材料。因此，不等跨多跨铰接梁力学模型是一种节材、降低支座安装数、降低人工费用的立柱力学模型设计方法。建议结合混凝土梁实际情况并根据本文的分析计算方法及结论进行优化选择。因此，采用不等跨多跨铰接梁力学模型进行立柱设计可以达到节材、降低支座安装数、降低人工费用等效果。建议在幕墙设计计算过程中，结合混凝土梁实际情况并根据本文的分析计算方法及结论进行优化设计。

3 结语

现阶段大部分工程为了方便仍直接选择最简单的力学模型进行立柱设计计算，并不深入的考虑实情从而大量浪费材料，建议根据本文分析结果，根据主体结构尽量选择多跨铰接梁力学模型进行立柱设计计算，并根据实际混凝土结构合理安装节点位置以达到节材、节能的效果。

参考文献:

[1] 黄 圻. 把握机遇调结构，提高质量增效益 [R]. 广州: 中国建筑结构金属协会铝门窗幕墙编委会，2013.

[2] 刘 伟，刘 斌. 青岛城市高层建筑楼顶形式特点的解析 [J]. 河北工程大学学报: 自然科学版，2011，28(4): 51－54.

[3] JGJ102－2003，玻璃幕墙工程技术规范 [S].

[4] GB50009－2001，建筑结构荷载规范 [S].

[5] 张 蕾，田 炯，李树娜，等. 幕墙中双跨梁模型的计算方法 [J]. 门窗，2011 (01): 16－19.

[6] 张 芹. 建筑幕墙立柱优化设计 [A] //2007 年全国铝门窗幕墙行业年会论文 [C]. 广州: 中国建筑结构金属协会铝门窗幕墙编委会，2007: 280－283.

[7] 陆新晓，谢云飞. 玻璃幕墙建筑的火灾特性 [J]. 黑龙江科技学院学报，2011，21(4): 317－320.

(责任编辑　刘存英)

(上接第 85 页) 度层面最优的粒子进行寻优，以减少调度计算所用的时间。通过在实际 Hadoop 平台的测试和分析表明，本文提出的调度算法能够很好的平衡不同类型作业之间的负载，减少作业运行时间，提高了平台性能。

参考文献:

[1] MANYIKA J，CHUI M，BROWN B，et al. Big data: The next frontier for innovation，competition，and productivity [J]. Communications of the ACM，2011，56(2): 100－105.

[2] SHVACHKO K，KUANG H，RADIA S，et al. The hadoop distributed file system [C] //Mass Storage Systems and Technologies (MSST)，2010 IEEE 26th Symposium on. IEEE，2010: 1－10.

[3] 曹 英. 大数据环境下 Hadoop 性能优化的研究 [D]. 大连: 大连海事大学，2013.

[4] Capacity Scheduler for Hadoop [EB/OL]. http://hadoop.apache.org/docs/current/hadoop－yarn/hadoop－yarn－site/CapacityScheduler.html，2014－09－05.

[5] Fair Scheduler for Hadoop [EB/OL]. http://hadoop.apache.org/docs/current/hadoop－yarn/hadoop－yarn－site/FairScheduler.html，2014－09－05.

[6] 李爱国，覃 征. 粒子群优化算法 [J]. 计算机工程与应用，2002，38(21): 1－3.

[7] 刘素芹，王 婧，李兴盛，等. 基于粒子群优化算法的集群调度策略 [J]. 郑州大学学报: 理学版，2010，42(2): 43－46.

[8] 张京军，刘文娟，刘光远. 基于改进免疫遗传算法的网格任务调度 [J]. 河北工程大学: 自然科学版，2013，30(2): 80－83.

(责任编辑　刘存英)