

# 云计算环境下基于模糊聚类的并行调度策略研究

张 千 梁 鸿 邢永山

(中国石油大学计算机与通信工程学院 青岛 266555)

**摘 要** 并行任务调度是分布式计算研究的核心问题之一,其结合大规模的石油地震勘探数据在处理过程中对高性能计算的需求,研究云计算环境下地震资料数据的并行调度问题。由于地震资料数据规模较大,因此通常将大作业进行分割,通过并行处理获得较高的处理效率。对任务进行并行处理的一个关键问题是如何将划分后的任务分配到合适的调度节点,最高效的情况是使云计算环境中的每一个资源节点都在进行计算,并且计算性能高的节点执行作业块大且复杂的任务,性能相对低的资源节点则运行对计算性能要求不高的任务或小任务,以达到整体上的负载平衡。因此基于模糊聚类思想,提出一种任务与资源混合聚类的调度优化策略,以作业与资源节点属性的匹配程度为基准,对并行作业进行聚类划分求解,在缩小任务调度规模的同时,为动态调度任务奠定基础。在划分完成后引入基于改进的贝叶斯分类调度算法,将资源节点依据其实时负载情况与队列中的作业进行快速的匹配。实验证实,此方案具有较高的执行效率。

**关键词** 云计算,并行调度,模糊聚类,任务资源混合聚类,贝叶斯分类算法

**中图分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.08.016

## Cloud Parallel Task Scheduling Algorithm Based on Fuzzy Clustering

ZHANG Qian LIANG Hong XING Yong-shan

(Department of Computer Science and Communication Engineering, China University of Petroleum, Qingdao 266555, China)

**Abstract** Parallel task scheduling is one of the key problems in the field of cloud computing research area, which mainly researches parallel scheduling problems in cloud computing environment by the reference to the high performance computing required by massive oil seismic exploration data processing. Because of the natural reparability of Seismic data, it can maximize the full use of computing resources to put the job file to the resource nodes, which can just meet the task computing requirements. This paper proposed scheduling optimization strategy of task and resource hybrid clustering based on fuzzy clustering. The strategy takes matching degree of task and resource nodes as reference and with the clustering partition solution of concurrent job, narrows task scheduling scale and at the same time, lays foundation for the dynamic scheduling of tasks. After the division is completed, improved Bayes classification algorithm is introduced to fast match tasks and computer according to real-time load and queue operations. In the end, the experiments verify that this scheme has higher efficiency.

**Keywords** Cloud computing, Parallel scheduling, Fuzzy clustering, Task and resource hybrid clustering, Bayes classification algorithm

## 1 引言

在目前主流的分布式计算环境中,研究作业的并行调度<sup>[1]</sup>问题成为提高作业效率的一个主要攻克方向,也是该领域未来的主要研究点之一。云计算环境中包括同构或异构的多资源、多集群环境,各个资源节点的计算能力一般不会完全相同,作业服务器处理并行作业时需考虑节点间的执行顺序,还要尽可能地在某时间周期内完成,这就需要云计算环境具有合适的资源管理方式、优良且可靠的作业调度策略。本文结合大规模的石油地震勘探数据在处理过程中对高性能计算

的需求,研究云计算环境下处理地震资料数据的并行调度问题。研究目的是提高数据处理的效率,充分高效地利用云计算资源,使任务透明、方便、高效地提交,并且使其快速匹配、高效执行在最优的计算资源中。

地震资料数据天然的数据可分性<sup>[2]</sup>,使得数据之间联系较少,相对来讲一个数据文件并发执行的效率也会较高。因此将作业队列放到能够恰好满足任务计算能力需求的资源节点上执行,能够最大限度地充分利用已有的计算资源;且由于云环境下的动态性,资源节点自身属性及任务对资源需求的描述也具有模糊性,因此本文提出一种将任务与云计算资源

到稿日期:2013-05-24 返修日期:2013-07-13 本文受中国石油天然气集团公司石油科技中青年创新基金项目(07E1024),中央高校基本科研业务费专项资金(13CX02032A)资助。

张 千(1982-),女,硕士,讲师,CCF 会员,主要研究领域为网格计算、云计算,E-mail: zhangqianupc@163.com;梁 鸿(1966-),女,博士,教授,主要研究领域为云计算、地震资料处理。

混合聚类的算法来实现任务聚类划分。该算法首先将待调度任务进行特征提取分析,然后对云计算资源节点进行特征提取,将上述两种结果执行模糊聚类划分,从而将节点中满足任务处理要求且效能发挥最好的划分到同一类,对不同的作业队列进行聚类划分求解,以该方式达到调度优化的目的。

针对划分后的任务队列,基于贝叶斯调度分类算法进行改进,将资源节点依据其实时负载情况与队列中的作业进行快速的匹配,在调度过程中通过学习作业属性来不断地改进作业调度,动态地对任务服务器上的作业进行参数调整,实现了一种适合多个作业队列并行执行,且能够正确完成作业任务的快速调度算法。

## 2 基于模糊聚类的优化划分策略

### 2.1 地震资料数据并行性特征

地震勘探数据具有自身特有的特点:(1)对抽道集的数据管理就是按照线、道数据进行管理的,如基于共叠加点(CDP)道集、共接收点(CRP)、共炮点(CSP)、共中心点(CMP)的炮集处理;(2)原始数据在采集时便以束、炮、线、道等方式存储,这种类型的数据比较适合并发执行;(3)地震的偏移、速度估计等处理需要傅里叶、F-K 等各种变换,若大规模的这类操作同时执行,并行处理是最好的选择,其能够获得非常高的处理效率。综合以上对其特点的分析可知,地震资料数据较适合分割处理,炮集内、单炮内均可对其执行并行处理。

本文主要研究的并行方式也是将地震资料数据先按炮集内炮数据(可以为一炮或多炮)进行并行划分,划分为多个作业队列后,可以同时以最大并发度执行每一炮数据的偏移成像处理,而且基于该种方式划分的各个作业队列之间不需要通信,结合 MapReduce 并行编程<sup>[3]</sup>优势,能够较好地适应云计算环境的处理。

综合上述地震数据资料的分析,考虑实验环境采用的具体云计算平台及并行编程模式,将地震资料数据进行作业划分需要解决以下问题:

(1)地震资料处理的完整流程中要进行反褶积计算、叠前(后)深度(时间)偏移处理、成像及解释等流程,整体上各个环节计算量都较大,如果直接交由作业服务主节点提交执行任务,Map 与 Reduce 节点之间,各个 Map 节点之间的通信会占据系统大部分的性能,主节点的负载情况也会较大,因此,考虑对地震资料数据进行合适的划分比较关键。本文采用基于共炮集并行划分作业的处理方式,在具体划分过程中,可以将一炮数据划分为一个子任务,数据量较大时也可以将几炮数据划为一个子任务,这样划分后各个子任务之间的耦合程度非常低,所以,该种方式的作业划分类似批作业松耦合并行调度。

(2)在云计算环境的具体规划中,考虑“物以类聚”的原则,将计算存储等性能相差不大的节点归为一类,这样实际上每个作业队列面对一个自治的节点区域,也可以认为是云计算环境中整体大环境下的一个子云环境,该种处理方式能够结合作业大小的划分,体现出高性能节点处理复杂任务、低性能节点处理简单任务的优势。

(3)在云计算具体环境中划分地震资料数据后,其存储方式可基于 HDFS 的共享分布式存储。通过该类方式,所有子任务的数据都是通过 NameNode 集中式地管理,在具体的任

务执行开始时,作业服务器会将初始的数据按队列与资源节点匹配程度进行分发,将要处理的任务分配到数据存在的节点执行,通过划分及准备,将进一步地减少云计算环境下各个 Map 与 Reduce 节点之间的数据传输及通信带来的巨大开销。

### 2.2 算法思想

由于地震资料数据的大规模特性,为了使任务调度得到优化,避免计算资源的浪费,本文考虑一种将任务与云计算资源混合聚类的任务划分算法(CFCM)。该算法首先将待调度任务进行特征提取分析,然后对云计算资源节点进行特征提取,将上述整理的结果按照某种数学方式混合,执行模糊聚类划分,从而将节点中满足任务处理要求且效能发挥最好的划分到同一类,而将处理能力较低的留给小作业,处理性能较高的留给大作业,以该方式达到调度优化的目的。

CFCM 算法将要调度处理的作业对资源的计算能力等属性的渴求程度表示为任务向量,将资源节点自身属性以自身计算能力等属性集合的形式表示为资源向量。向量相同位置的属性层次含义相同,即它们有相同的维数。通过对上述提供的属性向量进行标准化处理,将两种属性向量混合在一起进行聚类过程计算,这样聚类后获得的结果为同层次属性值较类似的被分为一类。若考虑影响划分任务的所有因素,会显得太过复杂,而且实际的任意一种分类方式都不可能处理所有细微的因素,因此本分类算法中只提取对影响分类结果强烈的主要因素。

**定义 1** 云计算环境中资源属性,如:总计算能力、节点数目、节点硬盘空间、节点 CPU 个数、主频、内存大小、网络通信性能、I/O 效率等相关参数分别用  $AC$ 、 $N$ 、 $H$ 、 $C$ 、 $CF$ 、 $M$ 、 $T$ 、 $IO$  等表示;任务属性如炮集大小、炮数、一炮数据大小、单炮数据道数分别用  $GS$ 、 $GN$ 、 $GD$ 、 $TN$  表示。

**定义 2** 作业向量  $X=(x_1, x_2, \dots, x_s)$ ,其中  $s$  为炮数值,  $x_i=\{x_{i1}, x_{i2}, \dots, x_{im}\}$  为  $X$  中每一炮的数据向量表示,  $x_{i1}, \dots, x_{im}$  为每一炮数据  $x_i$  的  $m$  个指标特征,向量中  $x_{im}$  表示作业集的第  $i$  个要进行分类的对象的第  $m$  个属性指标的原始数据。具体表示参数如定义 1 所示。

**定义 3** 资源向量  $R=(r_1, r_2, \dots, r_n)$ ,其中  $n$  为节点个数,  $r_i=\{r_{i1}, r_{i2}, \dots, r_{im}\}$  为每一个节点的数据向量,  $r_{i1}, \dots, r_{im}$  为  $r_i$  的  $m$  个指标特征,具体表示参数如定义 1 所示。

**定义 4** 权重向量  $w=(w_1, w_2, \dots, w_m)$ ,其中 
$$\sum_{i=0}^m w_i = 1 \quad (1)$$
  $w_i$  为不同云计算环境中各权重向量参数,取值情况不同。

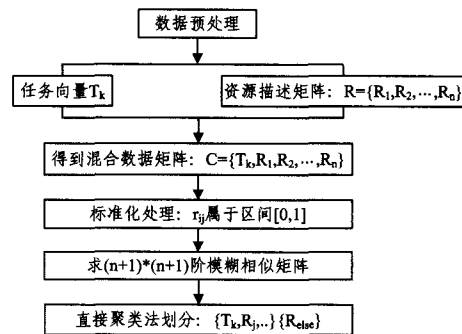


图 1 任务与资源一次混合聚类流程图

如图 1 所示,任务与资源混合聚类优化算法主要包括以

下4个步骤:(1)描述参数预处理;(2)标准化处理;(3)任务模块数据混合向量建立模糊相似矩阵;(4)执行直接聚类分类划分。

## 2.3 算法步骤

### 2.3.1 描述参数预处理

该步骤主要考虑如何分类一个新加入的任务。在环境确定情况下,选择节点的计算性能、任务所需计算性能、可用空间大小、任务数据大小等作为作业划分阶段的主要影响参数,然后按照模糊矩阵划分理论,将各个关联度数值转换到实值区间 $[0,1]$ 。

(1)单个节点计算性能表示为:

$$x_{1j} = \begin{cases} a \times C \times CF + \beta \times M + o(x_{1j}), & j > 1 \\ w_1 \times TN, & j = 1 \end{cases} \quad (2)$$

其中, $j > 1$ 为面向节点的计算性能表示, $j = 1$ 为面向作业的计算性能表示。 $a, \beta$ 为性能参数, $o(x_{1j})$ 为其它影响部分(可忽略),这里仅考虑 $C, M, CF$ ,分别对应定义1中的CPU个数、内存大小值、主频, $TN$ 为一炮数据的道数, $w_1$ 为权重向量值。

(2)为避免节点因多个副本而导致空间不足的情况发生,本文定义数据大小表示形式为:

$$x_{2j} = \begin{cases} H, & j > 1 \\ w_2 \times GD_k, & j = 1 \end{cases} \quad (3)$$

其中, $H$ 为第 $j$ 个节点硬盘大小, $GD_k$ 为该炮(第 $k$ 炮)任务对应炮数据大小, $w_2$ 为权重向量调整参数,若数据交互较多,可适当调大该参数值,交互较少时可适当取较小的参数值。

(3)节点数目对应炮数表示形式:

$$x_{3j} = \begin{cases} N, & j > 1 \\ w_3 \times GN, & j = 1 \end{cases} \quad (4)$$

其中, $N$ 为云计算环境中节点数目, $GN$ 为炮集中炮的数目, $w_3$ 为权重向量调整参数。

(4)总计算能力对应炮集大小表示形式:

$$x_{4j} = \begin{cases} \sum_{n=2}^{N+1} x_{1n}, & j > 1 \\ w_4 \times \sum_{k=1}^{GN} GD_k, & j = 1 \end{cases} \quad (5)$$

其中, $N$ 为云计算环境中节点的数目, $GN$ 为炮集中炮的数目, $w_4$ 为权重向量调整参数。

以上获得的计算数据仅作为在某一单独属性内的计算性能表示,经过后续标准化处理后才能表示其计算能力的具体含义。

### 2.3.2 标准化处理

将 $x_{ij}$ 转换为 $[0,1]$ 区间每个参数带权值的数值,组成 $X^{m \times (n+1)}$ 矩阵,通常有以下几种变换:平移·标准差变换、对数变换及平移·极差变换,这里取第一种标准化处理。

$$x'_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}, i=1,2,\dots,m; j=1,2,\dots,n+1 \quad (6)$$

其中:

$$\begin{cases} \bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij} \\ s_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2} \end{cases} \quad (7)$$

经过上述的标准化处理,转换后的变量的标准差为1,均值为0,并且在一定程度上减弱了量纲带来的影响。但该处

理得到的 $x'_{ij}$ 还不能保证在区间 $[0,1]$ 上,需要经过如下处理:

$$x''_{ij} = \frac{x'_{ij} - \min_{1 \leq i \leq n} \{x'_{ij}\}}{\max_{1 \leq i \leq n} \{x'_{ij}\} - \min_{1 \leq i \leq n} \{x'_{ij}\}}, j=1,2,\dots,n+1 \quad (8)$$

由此可得, $\forall i \in [1,m], \forall j \in [1,n+1], 0 \leq x''_{ij} \leq 1$ ,量纲层面影响被进一步降低。

### 2.3.3 任务模块数据混合向量建立模糊相似矩阵

(1)将定义2与定义3中定义的需求描述向量进行混合,可以得到混合向量集 $\{T, R\} \rightarrow \{C_1, C_2, \dots, C_{n+1}\}$ ,其中每个向量包含 $m$ 个影响参数因子,即 $C_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}, i=1,2,\dots,n+1$ ,这样得到任务与资源混合的描述矩阵。

在模糊聚类算法中<sup>[4]</sup>,模糊矩阵中元素之间的相关系数常用的有夹角余弦法、最大最小法、算术平均最小法等,本文取算术平均最小法,即:

$$r_{ij} = \frac{2 \sum_{k=1}^m (x_{ik} \wedge x_{jk})}{\sum_{k=1}^m (x_{ik} + x_{jk})} \quad (9)$$

由此得到 $r_{ij} \in [0,1]$ 。

(2)以矩阵形式表示各个混合向量间的模糊相似关系:

$$R = [r_{ij}]_{(n+1) \times (n+1)} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1(n+1)} \\ r_{21} & r_{22} & \dots & r_{2(n+1)} \\ \vdots & \vdots & \dots & \vdots \\ r_{(n+1)1} & r_{(n+1)2} & \dots & r_{(n+1)(n+1)} \end{bmatrix}_{(n+1) \times (n+1)} \quad (10)$$

聚类数目 $c$ 一般由专家指出,但对大量数据处理时缺乏必要的信服性,可以采用文献[5]所示方法,动态自适应获得分类数目 $c$ 。权指数 $m$ 的最佳取值范围为 $[1.5, 2.5]$ ,一般选 $m=2$ 即可。

### 2.3.4 执行直接聚类分类划分

本文依据直接聚类划分算法求解方式,为进一步提高计算速度,不再通过大量的计算过程求矩阵的传递闭包 $t(R)$ ,同时也摒弃布尔矩阵法等复杂的求取方式,建立模糊相似矩阵后,直接从其本身出发,以设置 $\lambda$ 值方式求取任务的所属分类,即完成一次聚类。

本文为了更方便地求作业的聚类结果,且避免一些特殊情况发生,如单纯地将任务划分为一类而该类中却没有任何资源节点,可以考虑将每一个单炮作业属性向量依次作为要构建的模糊矩阵的第一列,然后根据基于数学上等价关系模糊聚类划分计算方法,可以以单次遍历的计算量得到第一列的最大值,以此设置为截取水平 $\lambda$ 的值并进行聚类划分方案求解。通过反复地将新作业以属性向量形式加入,能够将地震资料炮集数据与合适的资源节点分为一类,得到新的描述作业划分情况的聚类结果集 $R_s'$ 。

## 2.4 算法流程

记作业 $JobID \in [1,5]$ ,节点 $RcID \in [1,n]$ ,且均为连续数值,由图1可知,混合聚类算法步骤为:

- (1)对数据进行预处理;
- (2)作业向量化 $\{job_1, job_2, \dots, job_s\}$ 与资源向量化 $\{re_1, re_2, \dots, re_n\}$ ;
- (3)将原始矩阵标准化处理,初始化 $k=1$ ;
- (4)初始化 $q=0$ ;
- (5)组合欲调度作业获得原始数据矩阵 $U_k = \{job_k, re_1, re_2, \dots, re_n\}, k \in [1,s]$ ;

(6) 计算模糊相似矩阵

$$R_k = [r_{ij}]_{(n+1) \times (n+1)};$$

(7) 聚类过程: 首先确定截取水平, 令  $\lambda = r_{21}, p[k][q] = 0;$

(7.1) 竖向寻找最大  $\lambda$

for ( $i=2; i \leq n+1; i++$ ) { 遍历提取  $r_{i1}$  中大于  $\lambda$  的值, 并记录  $i$  的位置  $p[k][q]=i$  }

$q++;$

(7.2) 在最大  $\lambda$  的横向寻找可运行该作业的节点

for ( $i=2; i \leq p[k][q]; i++$ ) {

如果横向存在大于  $\lambda$ , 作为分类可选节点号, 并记录位置  $p[k][q]=i, q++$  }

$k++;$

(8)  $f(k) \geq s$  则已经完成所有任务与资源划分, 转向步骤(9), 否则转向步骤(4), 继续执行下一个任务划分;

(9) 遍历  $p[k][q]$ , 输出所有任务队列与最优匹配节点集合。

### 3 基于改进贝叶斯分类的调度算法

#### 3.1 算法思想

在云计算环境下通过 MapReduce 并行处理计算任务的过程中, 一个作业被划分为多个任务并行分发到多个节点同时执行 Map 与 Reduce 操作, 作业调度器的角色就是将一个任务合理地安排到一个 slave 节点上运行, 尽可能地保证完成时间, 同时不出现过载情况。

一般的云调度算法对任务进行调度时需要预先设置每个 slave 节点的 Map 任务个数及 Reduce 个数, 即使在已知任务规模及计算量大小的前提下, 没有经验也很难定义优良的参数, 保证系统以较高的效率完成任务, 设置参数过大或过小都不利于系统并行程序获得较短的完成时间。尤其对于并行任务, FIFO 并行性不高, 明显不能满足基本处理要求, 公平调度算法及计算能力需要设置资源分配及作业队列划分情况, 在执行过程中出现失败或任务运行时间超过 deadline 不能保证作业尽可能完成<sup>[6]</sup>。可通过贝叶斯调度算法解决出现的这些问题, 不预先设置参数, 而是通过后续的学习来捕获经验值, 实现对任务服务器上作业的动态调整。

贝叶斯调度算法<sup>[7]</sup>按某时刻负载情况将作业划分为好作业坏作业后, 好作业队列中的作业永远会被调度, 坏作业则被丢弃, 这种方式未考虑系统的实时负载平衡, 可能出现确定好的好作业调度任务在暂缓调度后再被调度时导致系统过载, 即未考虑 Map 与 Reduce 节点实时负载变化情况, 同时某时刻的坏作业可能是影响后续所有作业的关键作业, 终止该作业可能导致整个作业运行失败。

改进的贝叶斯调度算法需要结合并行作业队列和上述混合聚类的调度优化算法, 如图 2 所示, 将并发粒度较好的作业划分为多个队列形式, 同时通过混合聚类的优化算法, 分配与作业队列计算量相匹配的计算资源。当资源节点出现空闲状态时, 作业服务器通过心跳信息会发现其闲置, 通过改进的贝叶斯调度算法从作业队列长度最长的队列中选择一个作业进行分类, 如果该作业按概率密度被认定为好作业, 则开始调度执行; 若被认定为等待作业, 则继续计算下一个作业的好作业与等待作业概率值, 直至在该队列中找到合适的作业, 具体执行时可能遇到本队列没有可调度作业的情况, 则需要转移到

队列长度次大值的队列中继续查找。通过该改进的贝叶斯作业调度算法, 能较快速地获得最适合在其上运行的作业任务。

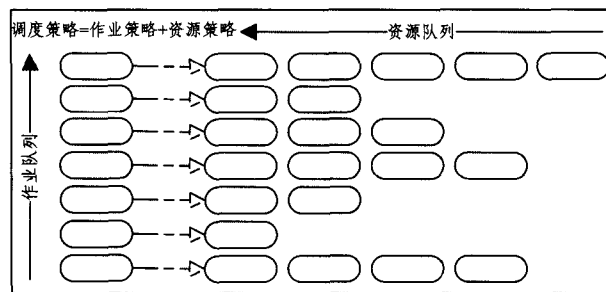


图2 作业结合资源动态分配图

在具体实施中, 需要结合混合聚类调度优化算法执行, 因此可首先考虑调度资源节点本身是否存在慢任务的情况, 如果空闲节点本身就存在其它慢任务, 则不需要执行改进的贝叶斯调度算法, 从而可以减少远程数据复制带来的开销, 只需在该空闲节点本身对慢任务执行数据备份并执行, 在资源节点本地协助完成慢任务比将该慢任务备份到其它节点再并列执行取最先完成至任务结束的情况具有更快的速度。将以上调度算法应用于地震资料数据处理, 在一定程度上减少了大容量数据在不同节点间备份的传输与通信开销, 并能够优先选择能最快速完成对应计算任务的节点来承担该任务, 提高了算法的效率。

#### 3.2 算法流程

根据上节对基于改进贝叶斯调度算法的思想描述, 本文对朴素贝叶斯分类算法<sup>[8]</sup>进行改进, 得到仅选择某一时刻针对具体节点的最优作业, 通过简单的学习过程或默认概率设置, 以过载与心跳信息中包含的作业状态监控值为调节因子, 系统可自适应地调节不同属性值在不同任务分类前提下的先验概率大小, 同时, 集合调度优化的全局调度策略, 可得基于改进后的贝叶斯调度优化算法流程如图 3 所示。从图 3 可以看出, 基于改进的贝叶斯调度算法主要分为如下几步:

(1) JobTracker 定时从 TaskTracker 获取心跳信息, 通过心跳信息实时地得到任务及节点执行属性的相关参数, 如节点剩余 CPU 利用率、剩余内存大小;

(2) 通过(1)获取的参数判断某个节点是否被判定为空闲节点, 若不是空闲节点, 则不需执行下一步操作; 若为空闲节点, 则需要调入作业, 执行步骤(3);

(3) JobTracker 周期性地对集群中的所有正在执行的任务按照最大估计剩余时间进行一次计算, 然后将计算所得的节点的任务状态与慢任务判定临界值 SlowTaskPoint 进行比较, 如果存在慢任务, 则首先在该空闲节点执行本节点的慢任务; 如果不存在慢任务, 则执行步骤(4);

(4) 按照调度优化策略, 从队列的二维数组中优先筛选本地节点数据相关的任务执行; 若不存在该划分后的数据相关性作业, 则执行步骤(5);

(5) 按照改进的贝叶斯调度算法, 以最大概率估计找到最适合执行的作业, 具体分为以下步骤:

(5.1) JobTracker 接收来自 TaskTracker 的心跳信息中包含剩余 CPU 利用率、剩余内存大小、I/O 读写速率等属性, 结合作业大小、道数值, 构造成对应该作业实时状态的属性特征变量;

(5.2) 根据设置的过载条件(HighLoad 的值), 基于学习

阶段各属性的概率,比较上次通过贝叶斯分类调度策略分配给该 TaskTracker 的任务完成情况,以重新生成各属性在最新状态的概率值;

(5.3)从最长等待队列中选择一个作业,并计算最大概率估计,对作业确定分类;

(5.4)若上述计算的作业为好作业,则将该作业分配到资源节点执行,完成该阶段的调度,否则继续步骤(5.3)选择作业判断;

(6)任务在 TaskTracker 完成之后,该 TaskTracker 将通过心跳信息将任务状态发送到 JobTracker,重复步骤(1)~(6)。

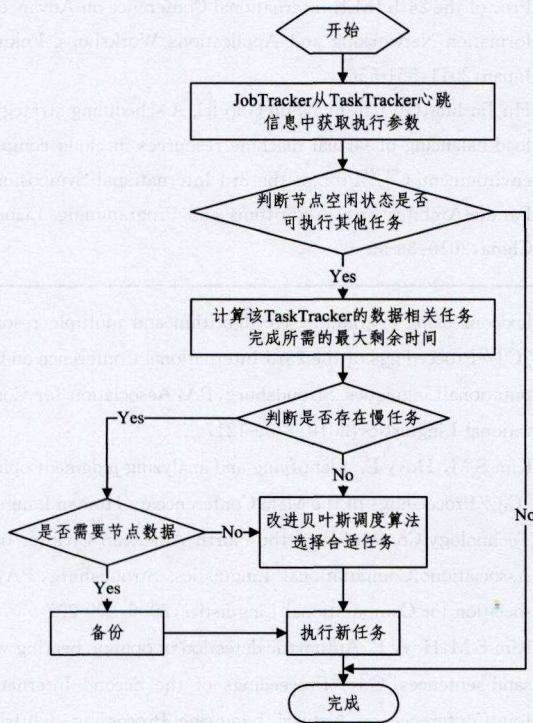


图3 基于改进的贝叶斯算法并行调度流程图

#### 4 实验测试及分析

本文用来测试结果使用的云计算环境包含 8 个节点。将一个节点作为主节点,即 HDFS 的目录服务器节点(Name-Node)与 MapReduce 作业服务器节点(JobTracker),剩余节点都作为工作节点(DataNode 与 TaskTracker)。各个节点间以千兆交换机连接。4 个 Dell PowerEdge R710 机架式服务器均为 4 核 2 个 CPU(Xeon E5520),硬盘大小 500G,4M 高速缓存,PC 节点均为双核 1 个 CPU(Pentium E5 系列),所有节点均安装 RedHat Enterprise Linux 5.1,云计算系统用开源 Hadoop 0.20.0 进行实验测试,操作系统采用 Redhat Enterprise Linux 5.1。

##### 4.1 模糊聚类调度优化实验

为了测试静态模糊聚类调度优化的性能,本文取计算能力与数据大小为主要参考因子,对计算能力占主导部分的任务调度,定义权重向量  $w=(0.7,0.3)$ ,选用 200 炮数据的划分作为测试用例,测试作业划分块数及数据具体处理的运行时间。通过观测模糊聚类划分法中途输出结果,可知其将作业分为 6 类(即 6 个队列),为了与其进行对比,人为地将数据

设定多个分类方式,作业划分完后用 MapReduce 默认的 FIFO 调度方式进行处理,在各种划分方式下任务执行的时间如图 4 所示。

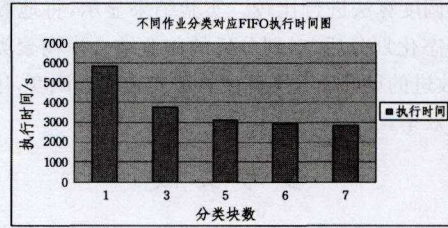


图4 模糊聚类作业块划分运行时间

由图中可以看出,模糊聚类作业划分算法能依据任务和环境层次对应特性找到较适合的粗粒度作业划分方式,为 MapReduce 根据负载情况动态调度任务做好准备,是提高云计算环境处理大规模数据整体执行效率的基础。

##### 4.2 改进贝叶斯调度算法实验

将本文提出的改进贝叶斯调度算法与 MapReduce 常用的 3 种调度算法进行性能比较实验。其中,FIFO 调度算法较为容易,而计算能力调度算法与公平份额调度算法则需要设置任务服务器上最多同时运行的任务数目,同时需要逐个增加对资源的描述,这不仅增加任务提交者的工作量及难度,若参数设置不当也会影响整体运行性能。因此在具体测试实验中需要设置适合占用 CPU 较多的参数设置方式,以减少对计算能力调度算法和公平调度算法的性能影响。

具体实验中,采用全局数组存储资源与节点的静态分配,通过简单的学习,设置属性在好作业与等待作业下的概率,分别计算 600、400、300 及 200 炮地震资料数据,并对各炮集处理结果进行比较。由图 5 可以看出,FIFO 调度算法整体近似直线,相对运行同类型任务时间比较稳定,但该调度算法的效率较低。对于计算能力调度算法与公平调度算法,它们的运行时间因任务服务器同时运行任务数及作业队列设置不同,负载均衡情况不会相同,故处理不同数据时,各显示出不同的优势。本文改进的调度算法由于事先设置了简单的学习过程,整体概率并不合理,故对炮集数据较少且本机节点未出现慢任务时,学习过程占据了部分时间,运行时间相对不占优,但随着数据量的增加,运行时间相对于其它 3 种调度算法表现出一定的优势。

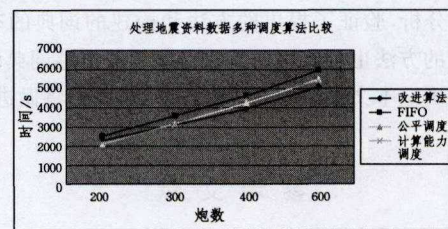


图5 处理炮数据的不同调度算法比较

**结束语** 本文是在云计算环境下以地震资料数据处理为基础进行的,在具体的调度决策过程中,将节点资源信息与作业属性动态结合,提出了一种基于模糊聚类思想的调度优化策略,将满足任务需求的资源分配给相应任务,避免将任务调度到相差悬殊的资源节点上,所有资源节点均具有被调度的机会,从而达到了较好的负载均衡效果。以贝叶斯分类策略

的概率估计为基础,引入改进的调度策略,建立作业与实时节点负载之间的协作关系,自适应调节属性概率,调度最适合执行的优先任务。最后通过实验对改进的调度算法进行测试,并与经典调度算法进行比较。实验结果显示,将地震资料数据进行静态化划分后,对划分后的作业结合模糊聚类优化策略,通过改进的贝叶斯分类调度算法实施动态调度,达到了较高的计算效率。

### 参考文献

[1] 赵春燕. 云环境下作业调度算法研究与实现[D]. 北京:北京交通大学,2009

[2] 罗银河,刘江平,俞国柱. 叠前深度偏移述评[J]. 物探与化探,2007,28(6):540-546

[3] Dean J, Ghemawat S. Mapreduce: Simplified Data Processing on Large Clusters[J]. Communications of the ACM, 2008, 1: 107-109

[4] 李文娟,张启飞. 基于模糊聚类的云任务调度算法[J]. 通信学报,2012,33(3):146-153

[5] 李柏年. 一种改进的模糊 C-均值算法[J]. 计算机应用与软件,

2009,25(6):21-25

[6] 夏祎. Hadoop 平台下的作业调度算法研究与改进[D]. 广州:华南理工大学,2010

[7] 余正祥. 基于学习方式对 Hadoop 作业调度的改进研究[J]. 计算机科学,2012,39(6):220-224

[8] Hand D J. 机器学习十大算法:朴素贝叶斯[M]. Taylor & Francis Group, LLC,2009:163-178

[9] Zhang Shu-fen, Zhang Shuai, Chen Xue-bin, et al. Analysis and Research of Cloud Computing System Instance[J]. 2010 Second International Conference on Future Networks,2010,60:88-92

[10] Randles M, Lamb D, Taleb-Bendiab A. A comparative study into distributed load balancing algorithms for cloud computing[C]// Proc of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, Japan,2011:551-556

[11] Hu Jin-hua, Gu Jian-hua, Sun Guo-fei. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment[C]// Proc of the 3rd International Symposium on Parallel Architectures, Algorithms and Programming, Liaoning, China,2010:89-96

(上接第 69 页)

论,“正面”2100 条,“负面”2400 条,采用 k-fold( $k=10$ )交叉验证的方法来验证,同时使用不同的情感词典进行对比实验,实验结果如表 4 所列。

表 4 新闻评论情感分类实验结果

使用的情感词典	正面评论 准确率	负面评论 准确率	平均准确率
本文构建的情感词典	80.95%	87.50%	84.44%
情感词汇本体	76.19%	81.25%	78.89%
NTUSD	69.04%	72.91%	71.10%
HowNet 情感词典	56.51%	68.13%	66.73%

从表 4 可以看出,本文所构建的情感词典分类准确率较其它情感词典高,表明本文提出的方法具有有效性;同时,本文构建的情感词典在训练分类器时所需时间最短。

**结束语** 本文针对网络新闻评论,提出了新闻评论专题领域情感词典构建方法。这种方法利用语料库和基础情感词典产生专题领域情感词集,在情感词集的基础上,获取种子情感词典,然后利用图排序模型原理,通过迭代运算,获得专题领域情感词的极性和强度,进而构建情感词典。文章通过实验和结果分析,验证了提出的方法和构建的词典的有效性。本文提出的方法也可以用于其它专题领域情感词典的构建。如何自动确定本文提出的方法的各参数,我们将做进一步探讨。

### 参考文献

[1] 杜伟夫. 文本倾向性分析中的情感词典构建技术研究[D]. 哈尔滨:哈尔滨工业大学,2010

[2] 许静芳,李星,李粤. 信息检索中主题式词典的构建方法[J]. 计算机工程,2005(21):153-155

[3] 柳位平,朱艳辉,栗春亮,等. 中文基础情感词词典构建方法研究[J]. 计算机应用,2009(10):2875-2877

[4] Xu G E, Meng Xin-fan, Wang Hou-feng. Build Chinese emotion-

lexicons using a graph-based algorithm and multiple resources [C]// Proceedings of the 23rd International Conference on Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics,2010:1209-1217

[5] Kim S M, Hovy E. Identifying and analyzing judgment opinions [C]// Proceedings of the Main Conference on Human Language-Technology Conference of the North American Chapter of the Association of Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics,2006:200-207

[6] Kim S M, Hovy E. Automatic detection of opinion bearing words and sentences [C]// Proceedings of the Second International Joint Conference on Natural Language Processing. Jeju Island: [s. n],2005:61-66

[7] Hatzivassiloglou V, Mckeown K. Predicting the semantic orientation of adjectives [C]// ACL-97: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. Madrid, Spain: [s. n],1997:174-181

[8] Velikovich L, Blair-Goldensohn S, Hannan K, et al. The viability of Web-derived polarity lexicons [C]// Proceedings of the North American Chapter of the Association for Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics,2010:777-785

[9] Turney P, Littman M L. Measuring praise and criticism: Inference of semantic orientation from association [J]. ACM Transactions on Information Systems,2003,21(4):315-346

[10] Brin S, Page L, Motwami R, et al. The PageRank Citation Ranking: Bringing Order to the Web [R]. Stanford: Stanford University,1999

[11] 徐琳宏,林鸿飞,潘宇,等. 情感词汇本体的构造[J]. 情报学报,2008,27(2):180-185

[12] 杨鼎,阳爱民. 一种基于情感词典和朴素贝叶斯的中文文本情感分类方法[J]. 计算机应用研究,2010,27(10):3737-3739