

一种基于先验知识动态分配资源的作业调度策略

汤灿恩^{①②}, 刘定生^①, 马艳^{①②}

(^① 中科院对地观测与数字地球科学中心, 北京 100086; ^② 中国科学院研究生院, 北京 100190)

摘要: 针对目前遥感图像并行处理系统的调度策略中资源分配盲目性, 导致作业平均带权周转时间过长和系统利用率低的问题, 提出了一种新的基于先验知识的动态分配资源调度策略。这种策略一方面利用先验数据库来存储各种算法单位资源单位数据量的平均运行时间, 结合待处理图像的数据量来估算作业单位资源的运行时间, 从而为系统作业调度提供准确的并行作业运行时间估计; 另一方面利用模糊数学知识动态地计算出集群的整体负载值。最终, 本文设计了一种自适应的资源动态分配方法, 可根据作业单位资源的运行时间和当前集群整体系统负载值动态决定作业所需分配的资源数。这种资源分配方式可解决传统资源分配策略在资源分配上的盲目性, 缩短作业平均带权周转时间, 从而使系统达到负载均衡。通过对比分析和实验结果, 本调度策略能较好地解决了目前并行遥感图像处理系统中传统作业调度存在的问题, 缩短了系统的作业平均带权周转时间, 提高了系统资源利用率, 使得整体系统的处理性能得到大幅优化。

关键词: 遥感图像并行处理系统; 动态调度策略; 负载均衡

doi: 10.3969/j.issn.1000-3177.2011.03.001

中图分类号: TP79 **文献标识码:** A **文章编号:** 1000-3177(2011)115-0003-06

A Dynamical Scheduling Strategy of Resources Allocation Based on Priory Knowledge

TANG Can en^{①②}, LIU Ding sheng^①, MA Yan^①

(^① Center for Earth Observation and Digital Earth, Chinese Academy of Sciences (CAS), Beijing 100086;

^② Graduate University of Chinese Academy of Sciences, Beijing 100190)

Abstract: The effective job scheduling schemes play a critically important role in the cluster-based parallel processing of remote sensing image. However, the most job scheduling strategies commonly treat various parallel algorithms as undifferentiated jobs without estimating accurate run time of algorithms. Such methods could cause blindness in resources allocation, and always lead to low system utilization and long average weighted turnaround time. To overcome the problems above, a dynamic allocation of resources strategy is proposed in this paper. The unit time (average run time of processing per unit data) of various parallel algorithms that recorded in the knowledge database in advance is used for estimating the accurate run time of real parallel jobs with self learning ability. Through the accurate estimation of job run time and system load that dynamically computed by fuzzy model, the resources allocation is dynamically decided in order to shorten the average weighted turnaround time, and finally to achieve the load balance of the entire system. Through experimental and comparative analysis, the outstanding scheduling efficiency of the strategy is showed in this paper.

Key words: parallel pprocessing system for remote sensing image; dynamic scheduling strategy; load balancing

收稿日期: 2010-04-14 修订日期: 2010-06-20

基金项目: 国家 863 重点项目支持。

作者简介: 汤灿恩, 男, 硕士研究生, 主要研究领域遥感图像处理系统作业调度与系统负载均衡。

E mail: gaolaot@139.com

1 引言

遥感图像并行处理系统的调度策略好坏将直接影响系统的性能。遥感图像处理算法往往涉及海量的遥感数据,算法复杂且计算量大。高性能遥感图像处理平台采用有效的调度策略和资源分配策略,优化调度效率^[1],从而大大提高系统性能。

遥感图像并行处理系统的作业调度一般由运行管理子系统^[2]根据一定的调度策略,依据特定的调度算法或规程向系统提交作业。目前遥感图像并行处理系统中常用的调度策略包括,先来先服务调度算法(FIFO)、首次适应算法(First fit)、最佳适应算法(Best fit)、贪心算法(Greedy)、优先级调度策略(Priority)、轮转调度(Round robin)、加权轮转调度(Weighted round robin)、资源预约算法(Reservation)、装填策略(Backfilling)等。这些调度策略都比较成熟,且一定程度上优化了系统的调度性能。但是这些调度策略都假设处理作业是随机的,且调度前都无法对待调度作业进行准确的运行时间估计。因此,这些调度策略在资源分配上存在一定盲目性,在不知道作业的运行时间情况下,只能根据当前集群服务的实时负载情况,采用平均资源分配策略,盲目地根据一定的算法为不同作业分配相同的资源数目。为此,本文提出一种基于先验知识的动态分配资源算法,核心思想是依据各种遥感图像处理算法在单位资源和单位数据量下的平均运行时间,结合处理图像的大小来估算实际作业单位资源的运行时间,同时动态地计算出集群的整体负载值,进而根据作业单位资源的运行时间和集群整体负载值来分配作业的资源数。该策略的目标是通过合理地分配系统资源,最终达到减少作业平均带权周转时间,提高集群的节点利用率。

2 基于先验知识的动态分配资源算法

基于先验知识^[3]的动态分配资源调度策略能够根据不断变化作业信息而快速合理地分配该作业的资源个数,调度策略的结构如图1所示:

在具体实现上可分成4个步骤:

①分析作业的请求信息,由先验知识数据库查询算法单位资源单位数据量的平均运行时间和计算处理遥感图像的大小,从而估算出作业单位资源的运行时间;

②确定集群节点的负载指标,例如:CPU利用率、内存的使用情况、磁盘的I/O速度、系统的响应

速度等;

③计算出集群各个执行服务器的负载值,加权计算出集群整体的负载值;

④最后根据作业单位资源的运行时间和集群整体的负载信息来合理分配作业资源,并且把这个作业提交给集群作业管理系统。

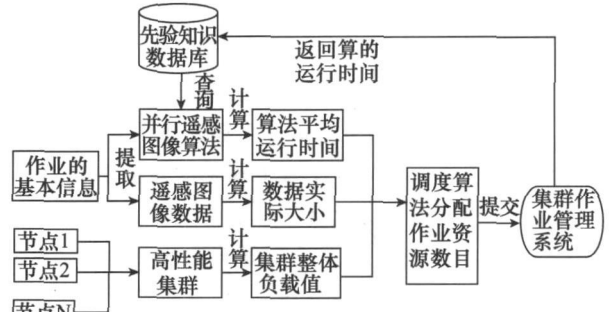


图1 基于先验知识的动态分配资源调度策略

2.1 作业单位资源运行时间的估算

资源分配策略第一大任务是估算作业单位资源运行时间,它由并行遥感算法的平均运行时间与作业所处理的遥感图像数据的大小来决定。当有新的作业请求时,系统根据作业的基本信息计算出作业处理图像的数据量 m 和确定作业所对应的并行遥感图像处理算法,接着通过先验知识数据库查询出该算法单位资源单位数据量的平均运行时间 T_{ave} ,显然作业单位资源的运行时间 $T = T_{ave} \times m$ 。

这里处理的遥感图像数据是复杂多变的,有些数据是多个波段,有些数据是单个波段,并且有些并行遥感图像处理算法只支持多个数据处理的,而有些只支持单个数据处理的。计算数据大小的过程如下:

①计算出第 i 个数据中第 j 个波段大小:

$$m_{bandj} = row_j \times col_j \tag{1}$$

②计算出第 i 个数据的大小:

$$m_{datai} = \sum_{j=1}^M m_{bandj} \tag{2}$$

③最后计算出图像的总数据量:

$$m = \sum_{i=1}^N m_{datai} \tag{3}$$

其中 N 代表所处理图像的个数, M 代表该图像的波段数目。

先验知识数据库是用来存储各种并行遥感图像处理算法单位资源单位数据量的平均运行时间 T_{ave} 和算法运行的总次数 N_{total} 。每当算法模块处理完后,根据算法总运行时间 t_r ,计算出该算法单位资源单位数据量的平均运行时间 T_{now} :

$$T_{now} = t_r \times n \div m \tag{4}$$

并及时更新先验知识数据库并行遥感图像算法的 T_{ave} 和 N_{total} :

$$T_{ave} = (T_{ave} \times N_{total} + T_{mv}) \div (N_{total} + 1) \quad (5)$$

$$N_{total} = N_{total} + 1 \quad (6)$$

其中 t_r 是作业所对应算法的运行时间, m 是作业处理图像的数据量, n 是分配给作业的资源数目。

2.2 集群整体负载估算模型

资源分配策略第二大任务是估算集群系统的整体负载值。集群中各节点的性能可能有很大差异, 因此不能简单地用计算的连接数来确定节点的负载, 节点的负载应该归结于它本身的各项指标, 典型的有: CPU 利用率、内存利用率、磁盘的 I/O 占有率、响应速度等^[4]。采取适当的算法得出当前时刻各节点负载值, 最后根据各节点的负载值推算出集群的整体负载值。根据模糊子集的定义^[5], 定义论域: CPU 使用率, $fL(u)$, $fM(u)$, $fH(u)$ 分别称为 u 对于 L, M, H 的隶属函数, 其中模糊子集 L, M, H 所描述的 3 个阶段(同样也适用于其他属性的情况), 即低、中、高 3 个阶段。将经过模糊统计方法得到的隶属函数进行简化。其数学表达式为:

$$fL(u) = \begin{cases} 1 & 0 \leq u \leq 0.16 \\ 1.6 - 4u & 0.16 < u \leq 0.4 \\ 0 & 0.4 < u \leq 1 \end{cases} \quad (7)$$

$$fM(u) = \begin{cases} 0 & 0 \leq u \leq 0.3 \\ \frac{20}{3}u - 2 & 0.3 < u < 0.45 \\ 1 & 0.45 \leq u \leq 0.55 \\ \frac{14}{3} - \frac{20}{3}u & 0.55 \leq u \leq 0.7 \\ 0 & 0.7 \leq u \leq 1 \end{cases} \quad (8)$$

$$fH(u) = \begin{cases} 0 & 0 \leq u \leq 0.65 \\ 4u - 2.4 & 0.65 < u < 0.85 \\ 1 & 0.85 \leq u \leq 1 \end{cases} \quad (9)$$

这里 CPU 利用率被划分成 3 个语言变量: {Low(简称为 L), Middle(简称为 M), High(简称为 H)}, 其划分的具体意义分别为: CPU 使用率低、中、高。当给出任意时刻 CPU 利用率时, 都能根据上面的隶属函数计算出这一时刻 CPU 利用率属于

L, M, H 的程度: $fL(u), fM(u), fH(u)$ 。

接着考察各种负载指标的状态, 输入模糊变量 CPU 的利用率 u_1 、内存的使用率 u_2 、磁盘 I/O 的占有率 u_3 、响应速度 u_4 , 可得到它们相应的隶属函数, 隶属函数一般可由经验决定, 各属性的隶属函数都采取与 CPU 利用率相似的隶属函数。特别指出的是磁盘 I/O, 响应速度是与用户所选择的硬件紧密联系的, 所以隶属函数的系数是由用户的输入来调整。用因素集 U 表示: $U = \{u_1, u_2, u_3, u_4\}$, 那么, 因素集 U 到决策集 $V = \{L, M, H\}$ 存在一个模糊关系, 用模糊隶属矩阵 $R_{3 \times 4}$ 表示, 它的元素 r_{ij} 表示属性 j 隶属于状态 L, M, H 的程度 ($i = 1, 2, 3; j = 1, 2, 3, 4$):

$$R \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \end{bmatrix}$$

显然, r_{1j} 越大, 表明隶属于 L 的程度越高, 同理 r_{2j} 越大, 表明隶属于 M 的程度越高, r_{3j} 越大, 表明隶属于 H 的程度越高。考虑到 CPU 利用率高, 表明服务器繁忙、负载重, 反过来 CPU 利用率低, 表明执行节点空闲、负载轻。所以, 对 CPU 负载来说, 当前时刻负载等于 $r_{11} \times 2 + r_{21} \times 0 + r_{31} \times (-2)$ 是合理的。当然这里的常数 2 可根据不同的服务以改变。对于 Memory 利用率、I/O 占有率两属性与 CPU 利用率情况一样, 而响应速度与其相反。表 1 是 4 个属性的 Load 的算法:

表 1 负载中各属性的计算算法

	CPU	Memory	I/O	Reponses
L	$r_{11} \times (-2)$	$r_{12} \times (-2)$	$r_{13} \times (-4)$	$r_{14} \times 2$
M	$r_{21} \times 0$	$r_{22} \times 0$	$r_{23} \times 0$	$r_{24} \times 0$
H	$r_{31} \times 2$	$r_{32} \times 2$	$r_{33} \times 4$	$r_{34} \times (-2)$
Load				

表 1 中的系数: 2, 2, 4, 2 实际上给出了各属性在负载中所占的比例, 即权重。也就是说可以认为 CPU 利用率、Memory 利用率、I/O 速度、Reponses 时间表明服务器负载的权重分别为: 0.2, 0.2, 0.4, 0.2。这种权重关系一般根据并行系统所处理的作业特定来决定的, 而遥感图像并行处理系统所处理的遥感图像处理算法由于一般要处理大规模的遥感图像数据, 因此该类型算法处理计算密集外, 还是一种 I/O 密集型并行算法, 而且当系统的 I/O 性能较低时, 很有可能因为 CPU 与 I/O 较大的性能差距, 而产生 I/O 性能瓶颈。因此, 本系统的系统负载估

算模型中, I/O 占有率占有较高的比例系数。

由表 1 可以得到节点当前时刻的负载情况:

$$Load = [r_{11} \times (-2) + r_{21} \times 0 + r_{31} \times 2 + r_{12} \times (-2) + r_{22} \times 0 + r_{32} \times 2 + r_{13} \times (-4) + r_{23} \times 0 + r_{33} \times 4 + r_{14} \times 2 + r_{24} \times 0 + r_{34} \times (-2)] \quad (10)$$

$$\text{简化后: } Load = [r_{11} \times (-2) + r_{31} \times 2 + r_{12} \times (-2) + r_{32} \times 2 + r_{13} \times (-4) + r_{33} \times 4 + r_{14} \times 2 + r_{34} \times (-2)] \quad (11)$$

最后, 按照上述的模型计算出各节点的负载 Load 值, 集群整体的负载值计算公式如下:

$$Load_{ave} = \sum_{i=1}^N \omega_i Load_i \quad (12)$$

其中 $Load_{ave}$ 代表集群整体的负载值, N 代表集群执行节点的数目, $Load_i$ 代表执行节点 i 的负载值, ω_i 代表节点 i 的加权系数且满足 $\omega_1 + \omega_2 + \dots + \omega_N = 1$, 由于集群各节点的硬件不一致, 执行节点的性能越高, 其加权系数越大, 执行节点的性能越低, 其加权系数越小。

2.3 自适应动态资源分配策略

为了对本系统的并行计算资源进行合理有效分配, 提出了一种自适应动态资源分配策略。该策略分别对并行遥感图像处理算法、算法处理所需分配的系统资源数目进行分类, 并在算法分类和算法处理所需分配的系统资源数目之间建立一种映射关系, 如公式 (13)。其中并行算法根据算法单位资源的平均运行时间 T (单位时间 s) 划分等级, 每个等级的时间间隔为 800s; 算法处理所需分配的系统资源数目 N 分为 4 个分区, 分别为 2, 4, 6, 8。本系统调度中心根据算法处理所需分配的系统资源数目 N 建立相应的算法等待调度队列。算法与资源数为 N 的调度队列的映射关系如下^[6]:

$$N = \begin{cases} 2, & (0 < T \leq 800) \\ 4, & (800 < T \leq 1600) \\ 6, & (1600 < T \leq 2400) \\ 8, & (2400 < T) \end{cases} \quad (13)$$

接着根据集群实时的整体负载情况 $Load_{ave}$ 进行独立分区, 本文把集群负载值划分为 4 个分区, 分区索引 λ 分别为 0.5, 1, 1.5, 2, 分区索引 λ 与集群负载值的映射关系如下:

$$\lambda = \begin{cases} 0.5, & (Load_{ave} > 15) \\ 1, & (10 < Load_{ave} \leq 15) \\ 1.5, & (5 < Load_{ave} \leq 10) \\ 2, & (0 < Load_{ave} \leq 5) \end{cases} \quad (14)$$

最后分配给作业的资源数目 n 是根据算法单位资源的平均运行时间和集群实时的整体负载值来决定, 因此分配给作业的资源数目 $n, n = N \times \lambda$ 。事实上, 在估算作业单位资源的运行时间过程中, 利用先验知识数据库, 当系统所处理的作业越多, 其估算的结果越准确, 系统分配给作业的资源越合理。

3 性能估计与对比

下面利用作业平均带权周转时间和系统利用率分别对资源静态平均分配策略和基于先验知识的动态分配资源调度策略进行量化的对比分析。

3.1 作业平均带权周转时间性能比较

假设用户提交的作业是随机的, 提交作业的数量为 $n, T_{process}, T_{wait}, T_{weight}$ 分别为作业实际运行时间、作业等待时间、作业带权周转时间。

作业带权周转时间:

$$T_{weight} = \frac{(T_{process} + T_{wait})}{T_{process}} \quad (15)$$

使用静态分配策略时, 作业的平均带权周转时间 T_{static_ave} 如下:

$$T_{static_ave} = \frac{\sum_{i=0}^n T_i}{n} \quad (16)$$

使用动态分配策略时, 作业的平均带权周转时间 $T_{dynamic_ave}$ 如下:

$$T_{dynamic_ave} = \frac{\sum_{j=0}^n T_j}{n} \quad (17)$$

其中, T_i 和 T_j 分别代表静态分配策略时第 i 个作业的带权周转时间和动态分配策略时第 j 个作业的带权周转时间。

为了比较两者的大小, (16) 式与 (17) 式相减得 (18) 式:

$$\left[\frac{\sum_{i=0}^n \frac{(T_{static_process_i} + T_{static_wait_i})}{T_{static_process_i}}}{n} - \frac{\sum_{j=0}^n \frac{(T_{dynamic_process_j} + T_{dynamic_wait_j})}{T_{dynamic_process_j}}}{n} \right] \quad (18)$$

其中, $T_{static_process_i}, T_{static_wait_i}$ 分别代表静态分配策略时第 i 个作业的实际运行时间和等待时间, $T_{dynamic_process_j}, T_{dynamic_wait_j}$ 分别代表动态分配策略时第 j 个作业的实际运行时间和等待时间。

由于作业带权周转时间: $T_{weight} = (T_{process} + T_{wait}) / T_{process} = 1 + T_{wait} / T_{process}$, 并且比较与分母无

关,故(18)式化简成(19)式如下:

$$\sum_{i=0}^n \frac{T_{static_wait_i}}{T_{static_process_j}} - \sum_{j=0}^n \frac{T_{dynamic_wait_j}}{T_{dynamic_process_j}} \quad (19)$$

假设用户对各种类型作业的请求是随机时,两种分配策略作业的平均等待时间是近似相等的,故3.5式进一步简化如下:

$$\sum_{i=0}^n \frac{1}{T_{static_process_j}} - \sum_{j=0}^n \frac{1}{T_{dynamic_process_j}} \quad (20)$$

$$\text{由数学公式: } \frac{1}{a} + \frac{1}{b} \geq \frac{2}{(a+b)} \quad (21)$$

其中 $a, b > 0$, 当且仅当 $a = b$ 时取等号。因此可以推导,当采用静态分配策略时,作业的运行时间长短各不一致,存在较大的差距;而采用动态分配策略时,作业的运行时间都比较一致,假设作业的总运行时间近似相等时,由(21)式可知,当各作业运行时间越集中,其作业平均带权周转时间最小,即:

$$\sum_{i=0}^n \frac{1}{T_{static_process_j}} > \sum_{j=0}^n \frac{1}{T_{dynamic_process_j}} \quad (22)$$

显然,当系统作业的单位资源运行时间越不一样时,动态分配策略在作业平均带权周转时间得到了非常好的改进。

3.2 系统利用率性能比较

系统利用率是指系统负载与可承受负载的比率。调度策略在进行节点选择时要充分考虑到节点间的负载平衡,要尽量使它满负荷地运转,不能使节点过载,也不能使节点空闲。假设集群可用的节点资源数 N_{total} , 在时刻 t 系统正在工作的节点数为 N_{nn} , 所以时刻 t 系统的利用率 η 如下:

$$\eta = \frac{N_{nn}}{N_{total}} \times 100\% \quad (23)$$

假设系统在时间段 T 内 (T 足够大) 处理作业, 集群系统整体负载肯定有负载高、负载中和负载低3个阶段, 所以时间段 T 内系统的利用率:

$$\eta_T = \frac{T_{high}}{T} \times \eta_{high} + \frac{T_{middle}}{T} \times \eta_{middle} + \frac{T_{low}}{T} \times \eta_{low} \quad (24)$$

其中, T_{high} 、 T_{middle} 和 T_{low} 分别代表负载高、负载中和负载低3个阶段的时间, 且 $T_{high} + T_{middle} + T_{low} = T$ 。

使用静态分配策略时, 系统在时间 T 内的利用率 η_{T_static} 如下:

$$\eta_{T_static} = \frac{T_{high}}{T} \times \eta_{high_static} + \frac{T_{middle}}{T} \times \eta_{middle_static} + \frac{T_{low}}{T} \times \eta_{low_static} \quad (25)$$

使用动态分配策略时, 系统在时间 T 内的利用率 $\eta_{T_dynamic}$ 如下:

$$\eta_{T_dynamic} = \frac{T_{high}}{T} \times \eta_{high_dynamic} + \frac{T_{middle}}{T} \times \eta_{middle_dynamic} + \frac{T_{low}}{T} \times \eta_{low_dynamic} \quad (26)$$

当系统属于负载高阶段, 集群系统各节点负载普遍高而静态分配策略仍然分配同样的资源数目给作业, 导致集群各节点过载, 故 $\eta_{high_static} < \eta_{high_dynamic}$; 当系统属于负载中阶段, 两种分配策略分配同样的资源数目, 故 $\eta_{middle_static} \approx \eta_{middle_dynamic}$; 当系统属于负载低阶段, 集群系统各节点负载普遍低而静态分配策略才分配同样的节点执行任务, 导致局部某些节点比较空闲, 故 $\eta_{low_static} < \eta_{low_dynamic}$ 。因此, 比较公式(25)和(26), 显然动态分配策略在系统利用率得到了非常好的改进。

4 实验比较与结论

为了验证基于先验知识的动态分配资源策略的高效性, 本文设计了一组实验, 对基于先验知识的动态分配资源调度策略和资源静态平均分配策略作了模拟对比实验, 作业负载样本来源高性能遥感图像处理平台各种并行图像处理算法混合作业, 通过对系统总运行时间、作业平均带权周转时间和利用率3个指标进行分析, 来衡量基于先验知识的动态分配资源调度策略的效率。

4.1 作业负载样本设计

测试所用的环境为联想深腾 1800 高性能服务器, 系统由 16 个节点组成, 每个节点配置均一样, 具体情况见表 2。

表 2 作业处理测试环境

机群类型	联想深腾 1800 高性能服务器
计算节点数目	16
单节点 CPU 数目	2
单 CPU	Intel(R) Xeon(TM) CPU 3.00GHZ
单节点内存	1GB DDR ECC
操作系统	Cent OS 4.0
交换网络	1000M 以太网(联想交换机 3224)
编译器	Intel 编译器 8.1

实验中假设每一个作业对处理器的数量需求是随机的, 资源静态平均分配策略提供给作业 4 个资源数, 而基于先验知识的动态分配资源调度策略是根据不同作业动态地分配资源数。实验提供 100 多个遥感图像数据, 总共设计了 50, 100, 200, 300 个作业, 作业产生的次序是随机的, 各种类型作业所占比例近似相等。

4.2 实验结果及分析

在本实验中, 分别对以上两种分配策略的作业的总运行时间、作业平均带权周转时间以及系统利用率。

4.2.1 总运行时间、作业平均带权周转时间对比分析

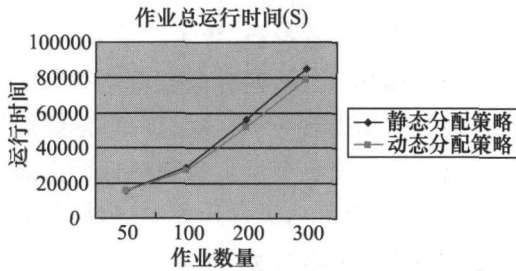


图2 总运行时间性能比较

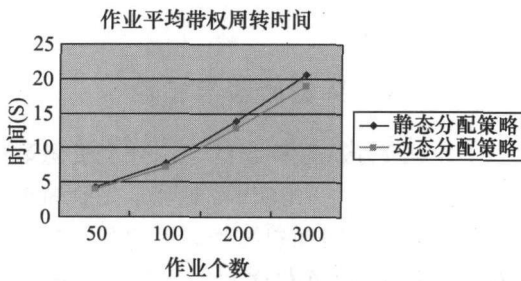


图3 作业平均带权周转时间性能比较

从图 2、3 中可以看出, 在各种作业样本中, 基于先验知识的动态分配资源调度策略有较小的总运行时间和平均带权周转时间。

4.2.2 系统利用率在不同作业样本对比

为了结合性能理论估计对系统利用率进行进一步分析, 实验中对系统在各作业样本进行统计, 实验结果如图 4、5、6 和 7 所示。

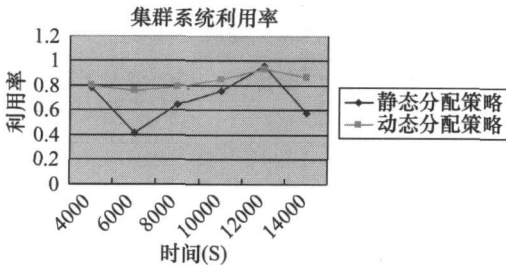


图4 作业样本 50 个

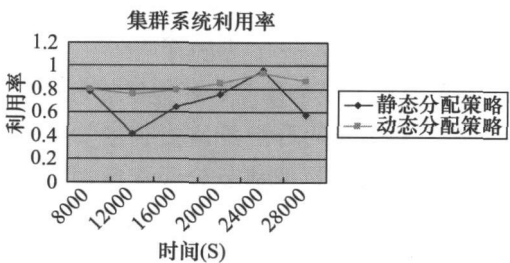


图5 作业样本 100 个

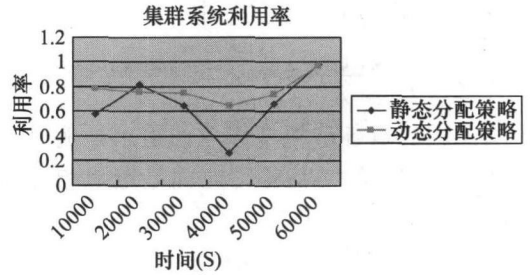


图6 作业样本 200 个

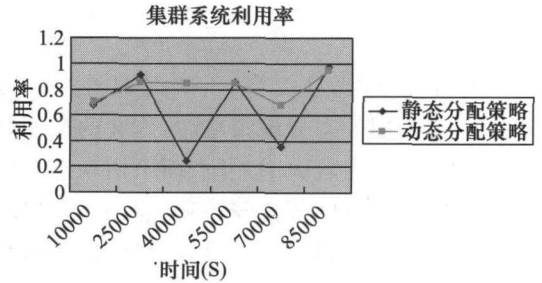


图7 作业样本 300 个

由图 4、5、6 和 7 知, 在各种作业样本中, 基于先验知识的动态分配资源调度策略的系统利用率比静态分配策略高。当作业样本越多时, 动态分配策略的系统利用率越稳定; 相反, 静态策略的系统利用率越不稳定。

5 结束语

通过前面的分析, 基于先验知识的动态分配资源调度策略本质不同于其他并行遥感图像处理系统资源分配策略可归结于以下几点:

①该资源分配策略是基于先验知识数据库根据自身算法与图像大小来分配资源数对作业所要运行的并行算法进行精确的运行时间估算, 并根据运行时间和当前集群综合负载情况进行自适应的动态资源分配, 真正实现了资源的合理分配; 而一般的分配策略是盲目固定不变地分配作业资源数, 很难真正做到合理分配资源;

②该资源分配策略动态实时地获取集群的整体负载值, 真正地实现了动态反馈调度; 而一般的调度策略很难真正实现动态反馈调度;

③该资源分配策略中作业运行时间的估算模型使用了先验知识数据库, 当系统处理的作业越多, 对作业运行时间的估算越准确。

实验结果表明, 基于先验知识的动态分配资源调度策略在作业的总运行时间和作业平均带权周转时间都比资源静态平均分配调度策略要短, 而且系统利用率都得到一定的提高。

(下转第 43 页)

4 结束语

去云处理是重要的遥感图像预处理环节,傅里叶变换对空间域和频率域不能兼顾的缺陷导致以其为基础的图像处理往往会有信息丢失。而小波变换将图像分解为一系列具有不同分辨率、频率和方向特征的子带信号,可以支持在综合考虑图像空间域与频率域特征的情况下进行处理操作。综合去云效果与信息保留两方面考虑,基于小波变换的处理结果要优于传统基于傅里叶变换的同态滤波方法。尽管引入的方法得到了较好的去云效果,但是在进行

高通滤波时,滤波器截止频率的确定仍是通过人工多次试验的方式获得。因此,对滤波器仍需进一步的研究改造,以期能通过对图像特征的综合分析自动获得这个参数,使该方法更加科学、有效。同时,当地物大面积特征相近、变化不明显时,也是一种低频信息,处理过程中难免导致这类信息的丢失,所以要取得效果较好的去云结果图像,除了提高所用算法的科学性与精确性外,更重要的是应该综合利用各种方法来进行处理,可以用单波段/单幅遥感图像处理方法对薄云或云层覆盖进行识别,然后利用数据融合方法对云区像元值进行恢复。

参考文献

- [1] 曹爽. 高分辨率遥感影像去云方法研究[D]. 南京: 河海大学, 2006.
- [2] 贺辉, 彭望球, 匡锦瑜. 自适应滤波的高分辨率遥感图像薄云去除算法[J]. 地球信息科学学报, 2009, 11(3): 305-311.
- [3] Lillesand, Thomas, M. Kiefer, Ralph W. . 遥感与图像解译[M]. 彭望球, 徐先川, 周涛, 等译. 北京: 电子工业出版社, 2003.
- [4] 刘洋, 白俊武. 遥感图像中薄云的去除方法研究[J]. 测绘与空间地理信息, 2008(313): 120-122, 125.
- [5] 王恒进. 基于小波的遥感图像薄云去除的研究与实现[D]. 西安: 西北工业大学, 2002.
- [6] 汪兴方. 遥感图像去云方法的研究及其应用[D]. 南京: 南京理工大学, 2007.
- [7] Tseng DC, Tseng HT, Chien CL. Automatic cloud removal from multi temporal SPOT images [J]. Applied Mathematics and Computation, 2008(205): 584-600.
- [8] Cihlar J., Howarth J. Detection removal of cloud contamination from AVHRR images [J]. IEEE. Transactions on Geoscience and Remote Sensing, 1994, 32(3): 583-589.
- [9] 梅安新, 彭望球, 秦其明, 刘慧平. 遥感导论[M]. 北京: 高等教育出版社, 2001.
- [10] 曾好德, 张韧. 基于小波分解的薄云薄雾图像的恢复与增强[J]. 解放军理工大学学报(自然科学版), 2005, 6(1): 76-80.
- [11] 周春艳, 王萍, 郑学芬. 基于 MATLAB 的遥感图像薄云去除 [J]. 西部探矿工程, 2006(S1): 224-225, 227.
- [12] 张汗灵. MATLAB 在图像处理中的应用[M]. 北京: 清华大学出版社, 2008.
- [13] 陈武凡. 小波分析及其在图像处理中的应用[M]. 北京: 科学出版社, 2002.
- [14] 张金泉, 杨进华, 卢珊. 基于同态滤波的图像去烟雾方法研究[J]. 机械与电子, 2009(7).

(上接第8页)

参考文献

- [1] 向彪. 卫星地面预处理系统控制单元设计与控制技术研究[D]. 北京: 中国科学院对地观测与数字地球科学中心, 2008. 6
- [2] James W. Cutler, Christopher A. Kitts Mercury. A satellite ground station control system [C]//Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, 2004.
- [3] 李闻, 吴秀清. 基于先验知识的小目标快速随机识别方法[J]. 计算机工程与应用, 2003(12).
- [4] 唐丹, 金海, 张永坤. 集群动态负载均衡系统性能的评价[J]. 计算机学报, 2004, 27(6): 80-81.
- [5] QI Zhaohui, SUN Jizhou, LI Xiaotu. A fuzzy congestion control algorithm based on queue [J]. Journal of Software, 2005, 16(2): 286-294.
- [6] 赵宗弟, 胡凯, 胡建平. 基于 OpenPBS 的集群作业调度策略的设计与实现[J]. 计算机与数字工程, 2006, 34(11).
- [7] 林闯. Web 服务器机群请求分配和选择的性能分析[J]. 计算机学报, 2000, 23(5): 500-508.
- [8] David B. Jackson, Heather L. Jackson, Quinn O. Snell. Simulation based HPC workload analysis [J]. IEEE, 2001.
- [9] Xiao Qin. Performance comparisons of load balancing algorithms for I/O intensive workloads on clusters [J]. Journal of Network and Computer Applications, 2008, 31(1): 32-46.
- [10] Qin X, Jiang H. Improving effective bandwidth of networks on clusters using load balancing for communication intensive applications [C]//Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference. 2005: 27-34.