# 网格资源调度算法研究

徐慧慧,石 磊,陈 信
(山东师范大学 信息科学与工程学院,山东 济南 250014)

**摘 要**:网格资源调度算法是影响网格成功与否的关键技术之一。首先对网格资源调度方法从不同的视角进行了分类,从三个方面阐述了网格资源调度的性能指标,并着重比较分析了几种典型的网格资源调度算法,包括Min-min算法、Max-min算法、基于经济模型的调度算法、基于遗传算法以及基于模拟退火算法的网格资源调度算法等,指出各种算法性能上尚存的不足之处并对下一步如何改进给出建议,最后给出资源调度的研究展望。文中为网格资源调度算法的研究提供了很好的参考。

**关键词**:网格;资源调度;性能指标
**中图分类号**:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)09-0076-03

# Research on Grid Resource Scheduling Algorithm

XU Hui-hui, SHI Lei, CHEN Xin
(College of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

**Abstract**: Algorithm for grid resource scheduling is one of the key technologies which influence grid success. Firstly classifies the grid resource scheduling methods from different angles, addresses the performance metric of grid resource scheduling from three aspects, and focuses on comparing and analysing several typical grid resource scheduling algorithms, including the Min-min algorithm, Max-min algorithm, grid resource scheduling algorithm based on economic models, genetic algorithms and simulated annealing, etc, points out that the remaining deficiencies of the performance and to improve the next step forward, finally makes an expectation about its future research directions. This essay provides very good references for the study of grid resource scheduling algorithm.

**Key words**: grid; resource scheduling; performance metric

## 0 引 言

网格计算[1](Grid Computing)是当前互联网研究中的一个热点,也是并行和分布处理技术的一个发展方向。网格资源调度是网格计算领域中的关键研究方向之一,在网格计算中,通过采取适合于网格任务特征和资源特点的调度策略,将网格计算中的资源分配给匹配的网格任务,从而使网格资源利用率最大化。

文中对几种典型的网格资源调度算法进行了深入探讨和研究,指出性能上尚存的不足之处及下一步改进的方向,为今后网格资源调度算法的研究提供了很好的参考。

**收稿日期**:2008-12-23;**修回日期**:2009-03-05
**基金项目**:国家自然科学基金资助项目(60373063);山东省自然科学基金项目(Y2006G19)
**作者简介**:徐慧慧(1984-),女,山东临沂人,硕士研究生,研究方向为网格调度;石 磊,副教授,硕士生导师,研究方向为网格资源管理、网络信息安全。

## 1 网格资源调度概述

### 1.1 网格资源调度分类

网格资源调度具体包括三个问题,即映射任务或任务集到哪个/哪些资源;给出应用的任务调度顺序;给出资源上任务的执行顺序。现有的网格资源调度方法多种多样,文中从不同的视角对网格资源调度方法进行了归纳,如表1所示。

依据调度目标,网格资源调度算法可以分为任务调度、资源调度和应用调度;依据调度算法所应用于调度的阶段,可分为起始调度算法、重调度算法及元调度算法。同时网格调度算法还可以分为性能调度与经济调度。经济调度可设置预算、截止期限等,性能调度方法包括精确调度、元启发式和多准则[2]等。

### 1.2 网格资源调度性能指标

网格资源调度的主要目标就是要对用户提交的任务实现最优调度,并设法提高网格系统的总体吞吐率。具体的说,网格资源调度方案性能的优劣可以从以下几个方面来衡量:

(1)任务总执行时间(Makespan)。一个好的资源

调度方案应该能够充分合理地利用网格环境中各种可用的计算资源和存储资源,实现在整个系统内网格应用任务的完成时间最小,Makespan 越小说明调度策略越好。

(2)负载平衡特性。网格资源调度策略应该能够充分使用网格环境中性能各异的各种资源,让不同的资源都能发挥它的优势,高性能的机器分配的任务多一些,低性能的机器分配的任务少一些,这样才能最大限度地利用网格资源,让任务尽快地完成。

(3)网格服务提供者和网格服务使用者之间互惠互利。对使用者来讲,网格调度的方案应该充分考虑使用者的需求和预算,应最大限度地减少应用网格服务的开销;对服务提供者而言,调度系统在满足任务按时完成的同时,最大限度地使服务价值得到最大化。

表 1　网格资源调度分类

| 视角 | 分　类 | 描述 |
|---|---|---|
| 调度目标 | 任务调度 | 优化吞吐量,即单位时间完成的任务数,又称高吞吐量调度器 |
| | 资源调度 | 优化公平标准,或优化资源利用率 |
| | 应用调度 | 以应用的目标为中心,通常是优化应用的性能 |
| 调度阶段 | 起始调度 | 在应用执行前匹配应用需求和可用资源 |
| | 重调度 | 在动态系统或应用变化时修改初始匹配。通过监视应用的执行,并作出是否重调度的决策,包括应用的迁移和动态的负载平衡 |
| | 元调度 | 考虑应用和系统整体的性能,协调同一网格中多个应用的调度 |
| 调度手段 | 性能调度 | 优化应用或系统的性能 |
| | 经济调度 | 考虑预算或截止期限制,优化经济因素 |

## 2　网格资源调度算法研究

围绕着网格资源调度,国内外已做了许多研究工作,先后提出了各种调度算法。Min－min,Max－min,Max－int 等算法是解决网格调度的经典算法。另外,Buyya 提出了一种基于应用经济模型的优化调度模型[3],其目的是在资源的拥有者和使用者之间建立一种“交易”,以尽可能低的费用满足资源使用者进行计算任务的最低要求;Vincenzo 介绍了一种基于遗传算法的资源调度算法[4],其目的是为了尽可能地提高资源的使用率和吞吐量;Abraham 等人介绍了模拟退火等进化算法[5]在网格资源调度中的应用。

### 2.1　Min－min 算法

该算法的主要思想如下:当需要调度的任务集合非空时,反复执行如下操作直至集合为空:

(1) 对集合中每一个等待分配的任务 $T_i$,分别计算出把该任务分配到 $n$ 台机器上的最小完成时间。假设任务在第 $k$ 台机器上的完成时间为最小,记为 Min $T_i(I)$ = MCT$(I,k)$,可得到一个含有 $m$ 个元素的一维数组 Min－Time;

(2) 设第 $a$ 个元素是 Min－Time 数组中最小的,对应的主机为 $b$,把任务 $a$ 分配到机器 $b$ 上;

(3) 从任务集合中把任务 $a$ 删除,同时更新 MCT 矩阵。

Min－min 算法一个最大的不足就是负载不平衡,已经有很多改进方案提出来改进该算法的性能。文献[3]提出一种 QoS Guided Min－min 算法,该算法先对高 QoS 作业使用 Min－min 算法进行调度,将其分配到高 QoS 资源上执行;然后再对低 QoS 作业使用 Min－min 算法进行调度,将其分配到所有网格资源上执行。这种算法将高 QoS 作业优先调度,解决了高 QoS 资源被低 QoS 作业占据的问题。

### 2.2　Max－min 算法

Max－min 算法的实现思路和 Min－min 算法很相似,只是把上面第(2)步找 Min－Time 数组中最小的元素改为找最大的元素。在同构非均一的计算系统中,Max－min 算法的调度性能优于 Min－min 算法;而在全异构的计算系统上,Min－min 算法的调度性能优于 Max－min 算法。

### 2.3　基于经济模型的调度算法

在基于经济模型的调度算法中,把网格环境和市场环境进行类比,用户作为买方,而资源的拥有者作为卖方,把诸如 CPU、存储器、带宽等不同类型资源的使用情况都转变成单一的成本。在基于经济模型的资源调度算法中,每一个任务都有一个最迟完成时间(deadline),同时整个任务集的运行费用不能超过用户定义的预算值(budget)。因此,资源调度器的目标是在 deadline 和 budget 约束的前提下,有着较优的调度性能。

文献[6]提出一种基于效益最优的网格资源调度算法,效益最优化算法是通过一个效益函数来评价任务分配到资源上的性能指标,以达到性能指标的最优性。

该算法中涉及的一些实体和参数有:

1) 用户集 $U = \{U_1, U_2, \cdots, U_n\}$,是 $n$ 个用户的集合;

2) 任务集 $T = \{T_1, T_2, \cdots, T_n\}$,每一个任务 $T_i$ 都包含一些任务信息任务长度(MI)、完成期限和费用预算;

3) 资源集 $R = \{R_1, R_2, \cdots, R_m\}$,是 $m$ 个计算资源的集合。

定义的效益函数的一般形式为:

$$\text{Benefit}(T_i, R_j) = \alpha \times \text{Benefit\_ETC}(T_i, R_j) + \beta \times \text{Benefit\_Cost}(T_i, R_j) \quad (1)$$

这里，Benefit($T_i$,$R_j$) 表示将任务 $T_i$ 分配到资源 $R$ 上所获得的效益，$\alpha+\beta=1$，$0\leqslant\alpha$，$\beta\leqslant1$。该效益函数是由完成时间的效益 Benefit_ ETC($T_i$,$R_j$) 和运行费用 Benefit_ Cost($T_i$,$R_j$) 的效益两部分加权平均得到。在调度过程中优先安排效益最大的任务和资源对（任务 $T_i$，资源 $R_j$），如果 $R_j$ 能够满足 $T_i$ 的完成时间期限 deadline 和费用预算 budget，那么将任务 $T_i$ 提交到资源 $R_j$ 上运行。否则，将 $T_i$ 提交到效益次优的资源上进行相同的操作，直到任务列表中的每个任务都被调度或调度失败。

在实际网格环境中，资源的供给和需求是一直处在变化之中的，价格和时间期限也会有浮动，所以为了适应这种动态性的变化，下一步设计算法时应该考虑到资源供需变化的动态自适应性这个问题，以此改进使用效益意义上的资源负载平衡的性能。

### 2.4 基于遗传算法的网格资源调度算法

遗传算法（GA，Genetic Algorithms）是将问题的求解表示成染色体，从而构成一群染色体。将它们置于问题的环境中，根据适者生存的原则，从中选择出适应环境的染色体进行复制，通过交叉、变异产生出新一代更适应环境的染色体群，这样一代一代地不断进化，最后收敛到一个最适合环境的个体上，求得问题的最优解。

文献[7]提出了一种基于遗传算法的资源调度算法，该算法采用资源－任务的间接编码方式，通过 DAG 图获取子任务的层次关系如图 1 所示，并将子任务按照层次深度排序，解决了种群中的非法问题。用如下的集合表示：$sb=\{sb_i \mid sb_1,\cdots,sb_n,1\leqslant i\leqslant N\}$，$N$ 是计算程序分解后子任务的数量。子任务 $i$ 在资源 $j$ 上执行的时间用 $\boldsymbol{E}[i][j]$ 表示；资源 $i$ 和资源 $j$ 之间的传输延迟用 $\boldsymbol{Tr}[i][j]$ 表示。算法流程如下：

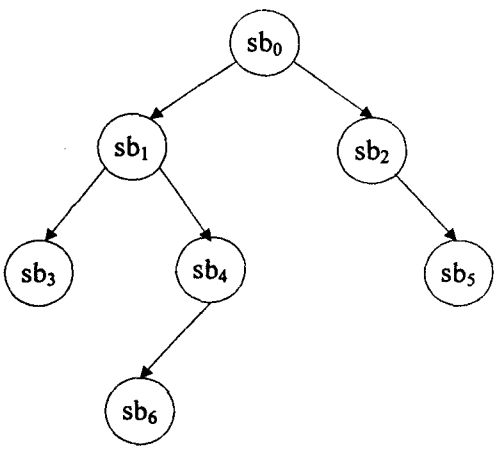



图 1 子任务的 DAG 图

(1) 初始化矩阵 $\boldsymbol{E}$ 和矩阵 $\boldsymbol{Tr}$，并根据 DAG 图生成每个子任务之间的逻辑关系，计算每个子任务的深度值；所有子任务之间的逻辑关系可以用一张 DAG 图表示，可以对 DAG 图进行分层，每层有一个深度值，深度值越小，代表优先级越高。深度值的计算公式如式(2)：

$$\text{level}(sb_i)=\begin{cases}0, sb_i \text{ 无父节点}\\ 1+\max(\text{level}(\text{parent}(sb_i))),\\ \text{其他 } sb_i\end{cases} \quad (2)$$

(2) 随机产生大小为 $M$ 的初始种群；根据每个资源上的子任务的执行序列，计算每条染色体的适应值；计算适应值必须计算每个子任务的完成时间，假设子任务 $i$ 在资源 $j$ 上的完成时间为 fin[$i$][$j$]，则 fin[$i$][$j$] = start[$i$][$j$] + $\boldsymbol{E}$[$i$][$j$]，start[$i$][$j$] 为子任务 $i$ 在资源 $j$ 上的开始执行时间。start[$i$][$j$] 的计算如公式(3)：

$$\text{start}[i][j]=\max\{\text{rs}-\text{spare}[j],\ \max(\text{fin}(\text{parent}(i)))+\boldsymbol{Tr}[m][j]\} \quad (3)$$

其中 rs－spare[$j$] 是资源 $j$ 上最近的一次空闲时刻；max(fin(parent($i$))) 返回值是子任务 $i$ 的所有父节点任务的完成时间的最大值；$\boldsymbol{Tr}$[$m$][$j$] 是该父节点任务所在资源 $m$ 和子任务所在资源 $j$ 之间的通信延迟。

(3) 选择染色体进行交叉操作和变异操作，计算新生成染色体的适应值，生成新的种群。

(4) 判断是否满足遗传算法的终止条件，如果满足，则停止计算，输出最小时间和对应的染色体；如果不满足，则返回(3)。

文献[8]中只考虑了子任务在资源上的执行时间和资源之间的传输延迟这两个关键因素，实际的网格环境中的任务特性包括更多的因素，因此下一步可以考虑更多网络因素，例如资源负载的动态变化以及资源的稳定性等方面来改进算法。

### 2.5 基于模拟退火算法的资源调度算法

模拟退火算法（SA，Simulated Annealing）源于对固体退火过程的模拟，其物理背景是固体退火的物理现象和统计力学模型，是一种每次都考虑一个可能的解决方案的重复技术，是广泛使用的解决优化组合问题的算法之一。它的最大优点是能避免问题的解落入局部最小，但其参数难以控制。

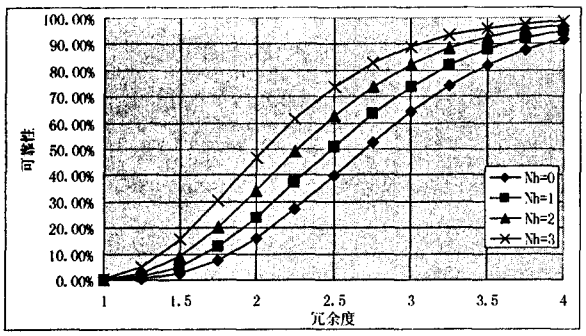
## 3 结束语

到目前为止，人们提出了很多网格系统的资源调度技术和算法。但是，不少调度算法还没有完整的理论依据，一些调度技术的结论还只是来自仿真结果，至今还没有形成网格计算的任务调度理论。因此，该领域的研究还有待进一步发展和完善，例如调度算法粒度的细化、网格调度系统安全机制的建立等。文中对

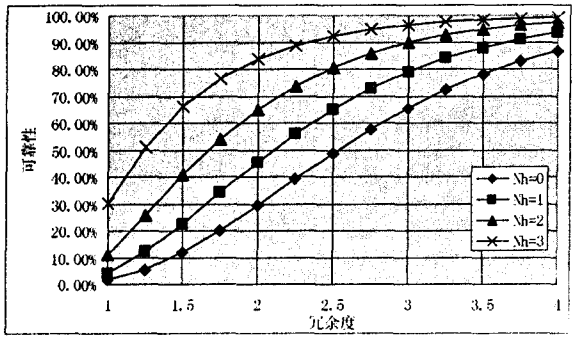
（下转第 82 页）

$$R = \sum_{j=0}^{j=mr-m} \binom{mr}{j} p_L^{mr-j}(1-p_L)^j$$

考察 $p_H = 0.95, p_L = 0.4, m = 8$,不同 $n_H$ 的取值对系统可靠性的影响。




m=8




m=4

图 4 系统可靠性分析

从图 4 的实验分析结果曲线可以得到结论,采用文中提出的基于节点可靠度的分层存储策略,高可靠度节点的利用率得到最大限度的利用,在相同的冗余条件下,随着编码块落在高可靠性节点上的数目的增加,系统可靠性也得到了进一步的提高。特别的,在冗余度为 1.5~3.4 范围内提高更为明显,这说明,当系统遭受一定的损坏后,节点数较少,系统经过一定的冗余恢复后,若处在相对较低的冗余情况,系统可靠性也能得到有效的保证。

## 4 结束语

目前,如何管理好日益增长的气象数据已经成为顺利有效开展气象业务的关键点,针对机房内部较稳定节点及系统设计足够简单的目标,提出的基于节点可靠度的数据存储策略,在一定程度上提高了系统的可靠性。文中研究可以作为探索中的试验田,通过实际应用充分体现 P2P 应用于存储的优势,发现并研究该方向建设过程中面临的问题,为其他气象台站建立统一的海量数据存储系统探索一条有效的解决途径。

**参考文献:**

[1] 罗杰文. Peer - To - Peer 综述[DB/OL]. 2006. http://www.intsci.ac.cn/users/luojw/P2P/ch01.html.

[2] 江武汉,叶从欢,孙世新. P2P - Grid 结构模型研究与设计[J]. 计算机技术与发展,2006,16(2):135 - 138.

[3] Wells C. The oceanstore archive: Goals, structures, and self - repair[R]. UC Berkeley Masters Report. California, USA:[s. n.],2002.

[4] 田 敬,代亚非. P2P 持久存储研究综述[J]. 软件学报,2007,18:2481 - 2494.

[5] Dabek F, Kaashoek M F, Karger D, et al. Wide - Area cooperative storage with CFS[C]//In: Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001). Banff: Chateau Lake Louise, 2001.

[6] Zheng W, Hu J, Li M. Granary: architecture of object oriented Internet storage service. E - Commerce Technology for Dynamic E - Business, 2004[C]//IEEE International Conference. Shanghai, China:[s. n.],2004:294 - 297.

[7] 袁卫东,战守义. 一种分组 P2P 网络模型[J]. 微机发展(现更名:计算机与技术发展),2005,15(8):50 - 52.

[8] 杨文俊. P2P 网络系统中节点自组织管理机制[J]. 计算机技术与发展,2006,16(7):57 - 60.

(上接第 78 页)

网格资源调度算法的研究提供了很好的参考。

**参考文献:**

[1] 陈宇寒. 网格计算技术研究[J]. 计算机技术与发展,2008,18(5):82 - 85.

[2] Moreno R A. Job scheduling and Resource Management Techniques in Dynamic Grid Environments[C]//in: 2003 annual Crossgrid Project Workshop & 1st European Across Grids Conference. Santiago de Compostela, Spain:[s. n.],2003.

[3] Buyya R, Abramson D, Giddy J. An economy driven resource management architecture for global computational power grids [C]//Int'l Conf on Parallel and Distributed Processing Techniques and Applications. Las Vegas:[s. n.],2000.

[4] Di Martino V. Scheduling in a grid computing environment using genetic algorithms[C]//Mililotti M. the 16th Int'l Parallel and Distributed Processing Symp (IPDPS2002). Florida, USA:[s. n.],2002.

[5] Abraham A, Buyya R. Nature's heuristics for scheduling jobs on computational grids[C]//The 8th Int'l Conf on Advanced Computing and Communications(ADCOM 2000). Cochin, India:[s. n.],2000.

[6] 胡自林,徐 云,毛 涛. 基于效益最优的网格资源调度[J]. 计算机工程与应用,2005(7):69 - 71.

[7] 林剑柠,吴慧中. 基于遗传算法的网格资源调度算法[J]. 计算机研究与发展,2004,41(12):2195 - 2199.