

# 基于 GridSim 的 A-MM 调度算法模拟

赵 健

(新疆大学 信息科学与工程学院, 新疆 乌鲁木齐 830046)

**摘 要:**对网格计算环境中的任务调度模块做了分析,介绍了两种最经典的网格任务调度算法:Min-Min 与 Max-Min。通过对两者的分析,得知 Min-Min 算法的优点在于其执行的高效性,但它存在资源负载不平衡的问题。而 Max-Min 则正好相反。经过对比,提出了一种平衡两者优缺点的 A-MM 算法。使用 GridSim 网格环境仿真包建立 A-MM 调度算法的模拟平台,对 GridSim 的内部机制及用法给予简单介绍,并通过模拟实验对 A-MM 算法进行性能评测。评测结果表明:A-MM 算法的完成时间接近 Min-Min,而负载平衡性接近 Max-Min,平衡了 Min-Min 与 Max-Min 的优缺点。

**关键词:**网格计算;任务调度;算法;GridSim

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2008)10-0096-03

## A-MM Algorithm Simulation Based on GridSim

ZHAO Jian

(School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China)

**Abstract:** Analyzes the task scheduling module in grid computing environment, and introduces two of the most classic grid task scheduling algorithm: Min-Min and Max-Min. Based on the analysis of the two, that the Min-Min algorithm advantage lies in its implementation of efficiency, but it exists a problem that resources load imbalance. Max-Min algorithm is for the very opposite. After contrast, introduces a new algorithm named A-MM which could balance the advantages and disadvantages of the two. Uses GridSim toolkit to establish simulation platform of A-MM, and introduces internal mechanisms and usage of GridSim, and uses simulation experiments to evaluate the performance of A-MM. Evaluation results show that: the completion time of A-MM algorithm close to Min-Min, and the load balancing of A-MM algorithm close to Max-Min. A-MM algorithm balances Min-Min and Max-Min's strengths and weaknesses.

**Key words:** grid computing; task scheduling; algorithm; GridSim

## 0 引 言

“网格计算”是当前互联网研究中一个出现频率很高的词汇,它代表着一种先进的技术和基础设施。同时,它也越来越成为下一代分布并行计算的规范,它聚集分布于世界各地的计算资源,使其合力解决科学、工程、商业领域的大规模问题。在网格环境中,任务管理、任务调度和资源管理是网格必须具备的三个基本功能<sup>[1]</sup>。

随着网格计算技术的飞速发展,网格计算中的任务调度问题也变得越来越重要。任务调度的目的是:在包含大量异构计算机的网格环境中,同时考虑各网格节点的计算性能、节点之间的通讯性能等参数,把不

同的任务以最合理的方式分配到相应的网格结点去完成。由于在网格环境中各处理器的运行速度、主机的负载、网络通讯的时间等参数都是动态变化的,因此任务调度问题同时也是一个非常困难的 NP 完全问题<sup>[2]</sup>。

任务调度系统性能的优劣主要取决于该系统对调度算法的选择,优秀的算法可以有效地提高网格的吞吐量,加强网格的运行效率,这样,网格的服务质量也会有所提升。目前,学者们对调度算法已经做了大量的研究,先后提出了各种调度算法。调度算法的创新与性能评估已成为当前学术界研究的热点话题。

## 1 A-MM 算法介绍

在众多的任务调度算法中,最传统也最经典的算法当属 Min-Min 与 Max-Min。下面对这两者给予简单介绍。

定义1 资源集合。假设网格环境中有  $n$  个资源  $r_1, r_2, \dots, r_n$ , 定义资源集合为  $R = \{r_1, r_2, \dots, r_n\}$ 。

收稿日期:2008-01-22

基金项目:国家自然科学基金(60563002);新疆自治区高校科研重点项目(XJEDU2004103)

作者简介:赵 健(1982-),男,新疆乌鲁木齐人,硕士研究生,研究方向为分布式计算、网格计算。

定义 2 任务集合。待处理任务队列中有  $m$  个任务  $j_1, j_2, \dots, j_m$ , 定义任务集合为  $J = \{j_1, j_2, \dots, j_m\}$ 。

定义 3 预期执行完成时间 (Expected Time to Compute, ETC) 矩阵。资源  $r_j$  在没有负载的情况下执行任务  $j_i$  所需要的时间, 记为  $ETC_{ij}$ 。所有  $ETC_{ij}$  值组成 ETC 矩阵。

定义 4 A-MM 判断数组。在 ETC 矩阵中分别取出任务  $j_i$  在资源  $r_j$  的最小  $ETC_{ij}$ , 记为  $\text{Min\_Time}(i) = ETC_{ij}$ , 将所有所得值组成一维数组  $\text{Min\_Time}$ 。

Min-Min 算法的具体步骤描述如下:

- 1: 判断任务集合  $J$  是否为空, 不为空, 执行步骤 2; 否则跳到步骤 7;
- 2: 对于  $J$  中任务  $j_i$ , 预测出  $j_i$  映射到所有可用资源  $r_j$  上的  $ETC_{ij}$  值, 组成 ETC 矩阵;
- 3: 得到 ETC 矩阵的 A-MM 判断数组  $\text{Min\_Time}$ ;
- 4: 根据 3 的结果, 在  $\text{Min\_Time}$  中找出最小  $ETC_{ij}$  值, 以及该值所对应的任务  $j_i$  和资源  $r_j$ ;
- 5: 将任务  $j_i$  映射到机器  $r_j$  上, 并将该任务从任务集合  $J$  中删除;
- 6: 更新其它任务在机器  $r_j$  上的  $ETC_{ij}$  值, 回到步骤 1;
- 7: 此次映射事件结束, 退出程序。

以上算法是对 Min-Min 算法的具体描述, 而对于 Max-Min 算法, 只需修改上述算法的第 4 步, 改为在  $\text{Min\_Time}$  中找出最大  $ETC_{ij}$  值, 以及该值所对应的任务  $j_i$  和资源  $r_j$  即可。

众所周知, Min-Min 算法是一种执行速度快、实现较少时间跨度 (Makespan) 的算法, 但是, 它存在资源负载不平衡的问题。而 Max-Min 算法同 Min-Min 相反, 它在一定程度上解决了负载不平衡的问题, 但执行速度却受到了影响。

根据经验可推知, 在调度任务大小非均一的网格计算系统中, Max-Min 算法的调度性能优于 Min-Min 算法, 而在任务大小较均一的计算系统中则反之。

定义 5 相对标准偏差 (Relative Standard Deviation,

RSD)。RSD =  $\frac{s}{\bar{x}}$ , 其中  $s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$ 。s 是样本标准偏差, 表示样本参数的离散程度,  $\bar{x}$  是样本均值, 相对标准偏差能较好地说明一组数据的分散程度<sup>[3]</sup>。

A-MM 算法的基本思想表述如下:

首先, 根据网格系统的性能预测得到 ETC 矩阵, 再根据 ETC 矩阵得到  $\text{Min\_Time}$  数组, 计算出  $\text{Min\_Time}$  数组的相对标准偏差值, 记为 RSD。A-MM 算法利用 RSD 可以更好地说明数据的离散程度特性, 判

断待调度任务大小的均一性。如果 RSD 值小于临界值  $\text{constRSD}$ <sup>[3]</sup>, 说明  $\text{Min\_Time}$  数组中数值的离散程度比较小, 即待调度任务的大小较均一, 这时, 任务调度系统选择使用 Min-Min 算法; 反之则说明待调度任务的大小非均一, 此时, 选择使用 Max-Min 算法。

从 A-MM 算法的思想可看出, A-MM 算法平衡了 Min-Min 与 Max-Min 的优缺点, 力求得到较 Min-Min, Max-Min 更优的执行结果。

## 2 GridSim 工具集

由于网格环境所固有的分布性和高度的异构性, 在真实的网格系统对某个任务调度算法的性能进行评估是一项非常复杂的任务。因此, 研究者常常借助网格模拟环境来完成这一工作。文中采用 GridSim 工具集对 A-MM 算法做性能评估。

GridSim 作为网格的模拟环境, 能够模拟异构且分布于全球的资源、简单的网络、资源的查找、任务的虚拟处理等。GridSim 将复杂的网格环境分解为三层, 每层专注解决一方面的问题<sup>[1]</sup>。见图 1。

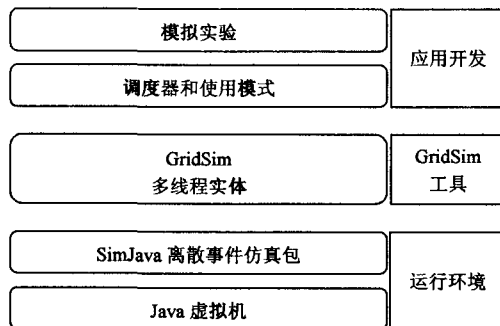


图 1 GridSim 体系结构

GridSim 工具借助底层运行环境<sup>[4]</sup>, 定义了大量的多线程实体, 这些实体可以模拟网格计算环境中的资源 (GridResource)、用户 (User)、用户资源代理 (Broker)、网络 (Input/Output)、信息服务 (GridInformationService) 等, 体现了它们的真实特性。同时, GridSim 工具也为上层的应用开发环境提供了良好的支持。GridSim 各实体之间的关系如图 2 所示。

## 3 利用 GridSim 实现 A-MM 算法

利用 GridSim 工具来模拟 A-MM 算法, 研究者需要创建网格用户 (User) 和调度器 (Broker) 的新实体<sup>[5]</sup>。

User 实体继承自 GridSim 实体, 主要负责用户应用程序的创建, 并将描述应用程序的任务 (Gridlet) 集合提交给 Broker 实体。

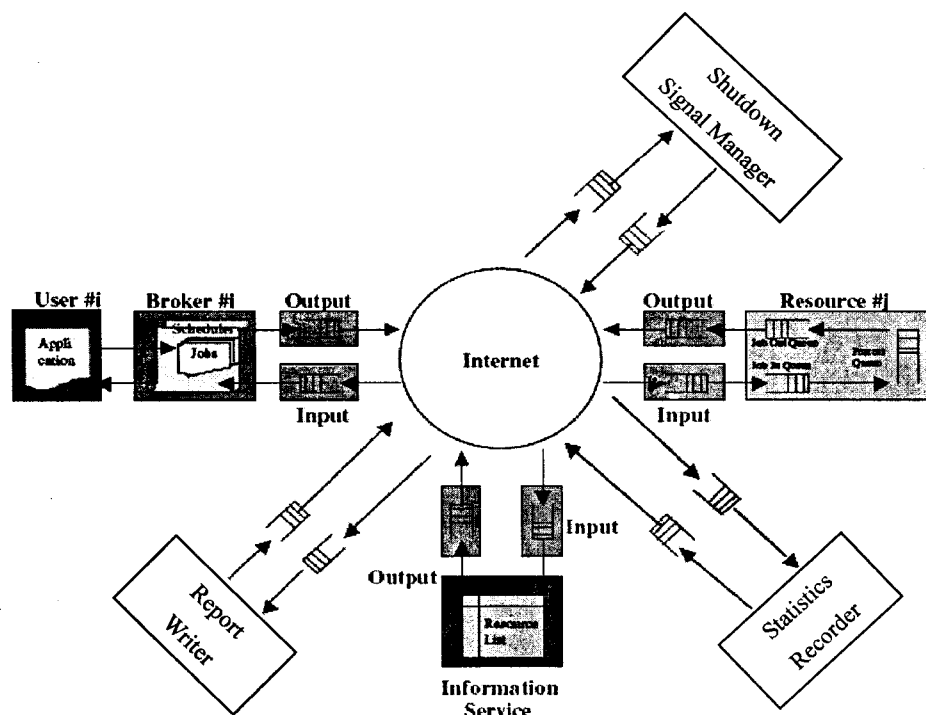


图 2 GridSim 实体

Broker 实体同样继承自 GridSim 实体,它是研究者搭建模拟网格环境的核心部件。用户的所有任务都提交给 Broker,Broker 会根据用户提出的调度原则进行任务调度。Broker 实体的功能如图 3 所示。

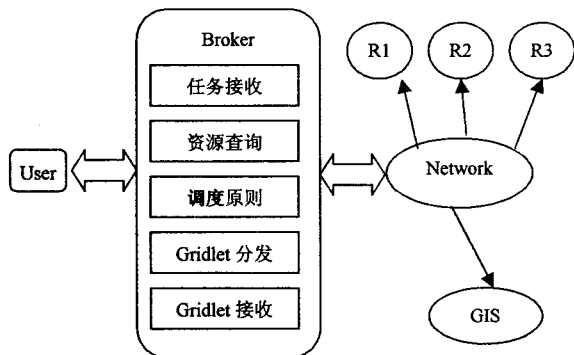


图 3 Broker 的功能

要将 A-MM 算法融入 GridSim 并对其进行性能评估,研究者应着重关注 Broker 实体的开发,具体过程描述如下:

1:创建 User 实体,通过 User 实体创建描述用户应用程序的 Gridlet 列表,并将此 Gridlet 列表和用户需求传递给 Broker 实体;

2:Broker 接收用户 Gridlet 列表;

3:Broker 的资源查询模块联系 GIS 实体,获得各资源的相关信息,组成一张资源属性列表;

4:Broker 判断 Gridlet 列表是否为空,若空,跳至步骤 7;否则,Broker 的调度原则模块根据各 Gridlet 的属性和各资源的属性预测 ETC 矩阵,再根据 ETC 矩

阵得到 Min\_Time 数组,通过 Min\_Time 数组计算 RSD。如果 RSD 小于 constRSD,执行步骤 5;否则跳至步骤 6;

5:Broker 的 Gridlet 分发模块按 Min-Min 的原则将 Gridlet 映射到相应的资源上,从 Gridlet 列表中删除该 Gridlet,回到步骤 4;

6:将步骤 5 的 Min-Min 改为 Max-Min;

7:Broker 将所有 Gridlet 提交至相应资源的 Gridlet 等待队列;

8:Gridlet 处理完成后,资源把它返回给 Broker 的 Gridlet 接收模块,接收模块

对返回的 Gridlet 进行相应的处理,再将它返回给 User 实体;

9:研究者根据 Gridlet 的返回结果对 A-MM 算法进行性能评估。

## 4 模拟实验

这里定义一个模拟场景来对 Min-Min,Max-Min,A-MM 三者的性能做出评估。该场景定义 50 个网格资源,并定义 100 个长度按照正态分布的任务,正态分布的均值为 77500 单位处理时间,均方差为 39500,constRSD 取 0.75。按照上述场景分别使用 Min-Min,Max-Min,A-MM 做一次实验,实验重复 50 次,取 50 次实验的平均结果作为模拟实验结果,统计平均 Makespan 值。并对其中的一个处理能力较强的资源进行监控,根据它执行任务的时间和数目统计出它的负载指数。实验结果如表 1 所示。

表 1 实验结果

	Min-Min	Max-Min	A-MM
Makespan	135	170	133
资源负载	6.3	2.8	3.9

由结果可得出,A-MM 算法的完成时间接近 Min-Min,而负载平衡性接近 Max-Min,平衡了 Min-Min 与 Max-Min 的优缺点。当然,对调度算法的评估是一项非常复杂的工作,文中的实验还比较单薄,这是有待改进的地方。

(下转第 102 页)

应的独立集报告类型。然而没有其它有利的非诚实揭示。这样,直接证明如果代理 1 总是报告策略上最优的类型,则这个机制的结果总是与最优诚实机制一致。当然,为了代理 1 总是报告策略上最优的类型,当他的类型是  $\theta_Y$  时,则他需要构建一个独立集。因为这个问题是 NPC 问题,猜想代理 1 将不总能构建这样的集合,是合理的。如果代理 1 确实不能构建此情况下的独立集,则结果将是  $x \in Y$ 。因此,在计算能力是无限的假设下,严格大于代理 1 群体利益。

因此,它也比最优诚实机制的社会福利更大。

### 3 结束语

显示原理在机制设计中是一个基础工具。然而,文中表明了,在计算上增加一个合理的约束就可能使显示原理无效。首先,研究了最优诚实机制的情况,说明中心处理这个机制的算法是 NPC 的。通过变换到非诚实机制,它能够从中心到代理转移解决 NPC 问题的负担;其次,提出了一种新的数据库模型,它处理效用

值难以计算的情况,即使在所有相关信息都可以获得的时候下,用这种模型也能处理。通过转移到非诚实机制,它能够转移指数数量询问的负担。在这两种情况里,非诚实机制与无限计算中最优诚实机制一样好。

#### 参考文献:

- [1] 谢识予. 经济博弈论[M]. 上海: 复旦大学出版社, 2002.
- [2] 迈尔森 R B. 博弈论[M]. 于 寅, 费剑平译. 北京: 中国经济出版社, 2001.
- [3] Rothkopf M H, Pekec A, Harstad R M. Computationally manageable combinatorial auctions[J]. Management Science, 1998, 44(8): 1131 - 1147.
- [4] Sandholm T. Algorithm for optimal winner determination in combinatorial auctions[J]. Artificial Intelligence, 2002, 135: 1 - 54.
- [5] Wurman P R, Wellman M P. AkBA: A progressive, anonymous - price combinatorial auction[C]//In Proceedings of the ACM Conference on Electronic Commerce (ACM - EC). Minneapolis, MN: [s. n.], 2000: 21 - 29.

(上接第 95 页)

- [3] Kao Yi - Tung, Zahara E, Kao I - Wei. A hybridized approach to data clustering[J]. Expert Systems with Applications, 2007 (34): 1754 - 1762.
- [4] Kennedy J, Eberhart R. Particle Swarm Optimization[C]//Proc. of IEEE International Conference on Neural Networks (ICNN). Perth, Australia: [s. n.], 1995: 1942 - 1948.
- [5] Shi Y, Eberhart R C. A modified particle swarm optimizer

[C]//Proceeding of IEEE International Conference on Evolutionary Computation. New York, NY, USA: IEEE, 1998: 69 - 73.

- [6] 李 兵, 蒋慰孙. 混沌优化方法及其应用[J]. 控制理论及其应用, 1997, 14(4): 613 - 615.
- [7] 刘华莹, 林玉娥, 张君施. 基于混沌搜索解决早熟收敛的混合粒子群算法[J]. 计算机工程与应用, 2006(13): 77 - 79.

(上接第 98 页)

### 5 结束语

由于网格调度算法对网格系统性能的巨大影响,促使人们对调度算法进行深入研究。文中针对传统的 Min - Min 算法的高效特性和 Max - Min 算法的负载均衡特性,介绍了平衡二者的优缺点的 A - MM 算法,说明了它的优势。在网格调度算法研究中,需要对算法进行深入的性能评估,然而由于受到真实网格环境的高度复杂性的限制,促使人们采用模拟工具来分析调度算法的性能。文中对 GridSim 进行了简单分析,并介绍了如何将 A - MM 算法融入到 GridSim 中去,最后通过模拟实验搭建网格仿真平台对 A - MM 算法进行性能评估。笔者认为,GridSim 仍然是网格仿真技术中最具有创造力的工具。

#### 参考文献:

- [1] Buyya R, Murshed M. GridSim: A Toolkit for the Modeling

and Simulation of Distributed Resource Management and Scheduling for Grid Computing[J]. Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002, 14 (13 - 15): 1175 - 1200.

- [2] Sulistio A, Buyya R. A Grid Simulation Infrastructure Supporting Advance Reservation[C]// Proceedings of the 16th International Conference on Parallel and Distributed Computing and Systems ( PDCS 2004 ). Cambridge, USA: MIT, 2004: 1 - 7.
- [3] 侯 勇, 于 炯. 基于非贡献网格的自适应任务调度算法研究[J]. 微电子学与计算机, 2007(10): 190 - 192.
- [4] Howell F, McNab R. Simjava: a discrete event simulation package for Java with applications in computer systems modelling[C]//in proc. First International Conference on Web - based Modeling and Simulation. San Diego, CA: Society for Computer Simulation, 1998.
- [5] 刘祥瑞, 朱建勇, 樊孝忠. 基于 GridSim 的网格调度模拟[J]. 计算机工程, 2006, 32(2): 42 - 44.