

基于性价比改进的网格调度算法

杜玉霞, 刘方爱

(山东师范大学 信息科学与工程学院, 山东 济南 250014)

摘要:在动态、异构的复杂网格环境中,任务调度算法已被证明是一个NP难问题。Min-Min调度算法是研究其它调度算法的基础之一。在分析Min-Min调度算法的基础上,指出该算法的缺陷:负载不均衡并且调度的过程中没有考虑费用的问题。针对这两个方面,提出了一种基于性价比改进的调度算法,通过分析表明,改进后的算法在费用、负载平衡度方面有了很大的提高,并且节省了很大一部分费用,说明改进后的算法在一定程度上提高了算法的效率,提高了网络的整体性能和总体吞吐量。

关键词:网格;调度;性价比;Min-Min

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2010)12-0099-04

Scheduling Algorithm of Modified Min-Min Based on Cost Performance

DU Yu-xia, LIU Fang-ai

(School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

Abstract: Task scheduling algorithm has been proved to be a NP hard problem in the dynamic, heterogeneous and complicated grid environment. Min-Min scheduling algorithm is one of the foundations of other scheduling algorithms. Its deficiencies: load imbalance and scheduling process does not consider the costs are pointed out in the analysis of Min-Min scheduling algorithms. For these two aspects, a scheduling algorithm based on cost performance is proposed through the analysis of the Min-Min scheduling algorithm. The analysis shows that the algorithm improved the cost and the load balance has considerably increased, indicating improved algorithm improve the efficiency of the algorithm and improve the overall network performance and overall throughput to some extent.

Key words: grid; schedule; performance-cost; Min-Min

0 引言

网格的概念产生于20世纪90年代中期,网格这个词是从电力网的概念发展而来的,它实际上是一个集成的计算与资源环境,它就好比一台巨大的超级计算机,能够实现计算资源、数据资源、专家资源等各种各样的资源在全球范围内的共享。

网格技术是当今世界发展的重点问题,任务调度作为网格技术发展的重要组成部分,越来越受到人们的关注。任务调度被证实为一个NP难问题^[1],因此引起了广大科研工作人员的广泛关注。在以往的研究中,已经产生了很多的任务调度算法,比较经典的算法有:Min-Min算法,Max-Min算法,遗传算法,模拟

退火算法,贪心算法等等。这些算法在一定程度上提高了网格任务调度的效率,其中Min-Min调度算法是一个简单、快速、有效的算法,调度时优先调度具有最小任务调度时间的任务-资源对进行调度,在一定程度上提高了资源的利用率,但是它有一个很大的缺点:负载不均衡。

1 任务调度

网格作为Internet的下一代,具有下述三个特点:

1)分布与共享性。分布性是指网格资源是遍布全球的;共享性是指网格是实现全球范围内的各种各样的资源的共享。

2)动态与多样性。网格资源分布在全球并不是恒定不变的,而是时刻处在变化之中的;并且网格资源的种类也是非常多的,即网格资源的多样性。

3)自治性。网格上的资源首先是属于某个人或某个组织,其次才是属于网格,这就决定了网格资源的所

收稿日期:2010-04-28;修回日期:2010-07-04

基金项目:山东省自然科学基金(Y2007G11)

作者简介:杜玉霞(1984-),女,山东莒南人,硕士研究生,主要研究领域为网格计算、互联网络;刘方爱,博士,教授,博导,主要研究领域为并行处理、互联网络、网络计算等。

有者对网格的管理具有最高的权限,这就是网格资源管理的自治性。

目前,重要的网格体系结构主要有两个,一个是五层沙漏结构,再一个就是开放网格服务结构 OGSA (Open Grid Services Architecture)。五层沙漏结构是一种影响十分广泛的结构,它不是具体的协议,而主要侧重于定性的描述,因此很容易理解。开放网格服务结构 OGSA 被称之为下一代网格结构。所以说五层沙漏结构是以协议为中心,而 OGSA 就是以服务为中心。

由于网格资源具有广域分布性,具有异构、动态和不稳定等特点,任务的分配和调度成为网格环境中最关键的一部分,它直接影响到各用户的任务是否能在指定的时间限度内完成。网格环境中的任务调度是面向异构平台的,在网格系统中,分布着各种各样的资源,如服务器、客户机等,它们是异构的,可运行在各种各样的操作系统上。并且任务调度必须具有可扩展性,因为现在的计算机规模是比较大的,并且随着技术的发展,网格环境系统毕竟要进一步扩大,所以任务调度必须具有可扩展性,才能保证整个网格系统的整体性能。

在网格环境中,网格资源分布全球,而且数量非常之巨大,且分布的位置不均,同样,网格任务也具有如此特点,这样就决定了任务调度的复杂性和未知性。由于以上原因,网格任务调度具有以下特点:任务调度面向异构平台且任务调度是大规模的、非集中式的;任务在调度的过程中不干涉网格节点内部的调度策略;任务调度必须具有可扩展性;任务调度能够动态自适应。

网格任务调度的主要目标就是要实现任务在所有资源上的利用率,并设法提高网格系统的总体吞吐量。主要目标^[2]包含以下四个:最优跨度、负载均衡、服务质量 QoS、经济原则。

(1) 最优跨度。

跨度是一个最基本的目标,是指从首个任务调度开始到最后一个任务调度完成所经历的时间。可见,跨度越小说明调度过程越快,效率也就越高。在任务调度的过程中,我们是希望跨度越短越好。因此,最优跨度是大多数研究工作者追求的目标之一。

(2) 服务质量 QoS^[3]。

服务质量是在任务调度的过程中体现出来的,在没有服务质量的任务调度中,调度的效率是很低的,对于网格任务调度来说,所谓的服务质量主要是指资源的价格、性能、优先级等一系列内容,不同的结构有不同的服务质量要求。为了提高任务调度的有效性,在调度的过程中保障服务质量是完全有必要的。

(3) 负载均衡。

负载均衡是指在任务调度过程中,资源要尽可能平均分配,使得每个资源都要尽可能的有任务在运行,这样不仅缩短了任务调度总执行时间,而且效率大大提高。负载均衡问题在并行任务调度的过程中运用得最为广泛,在一定程度上能够充分利用资源,提高了资源的利用率。文中主要在这一方面考虑了算法的有效性。

(4) 经济原则。

网格资源是分布全球的,并且每个资源所属的管理者不同,因此,网格资源的价格也是不一样的,在任务调度的过程中,不仅仅要考虑资源调度的有效性,并且还要充分考虑经济效益。这就好比在现实生活中,我们购买商品一样,我们想买一件功能很强大的商品,但是又得考虑自己的经济状况,最后选择一个我们最满意价格的商品。这样,网格资源的生产者和消费者才能长久地相处下去。

2 Min - Min 调度算法简介

Min - Min 算法是最经典的调度算法,是许多算法的研究基础之一。该算法的主要思想^[4]:

设网格环境是由 n 个任务 $T = \{t_1, t_2, \dots, t_n\}$ 和 m 个资源 $R = \{r_1, r_2, \dots, r_m\}$ 组成。当需要调度的任务不为空时,执行如下操作直至所有的任务都被调度完毕:1) 计算每一个任务 t_i 在所有资源上的最小执行时间(资源数为 n)。假设任务 t_i 在第 m 个资源上的完成时间为最小,记为 $\text{MinTime}(i) = \text{MCT}(i, m)$,可得到一个数组: MinTime ; 2) 设数组 MinTime 中最小的元素为第 i 个,且对应的主机为 host ,把任务 t_i 分配到资源 host 上; 3) 从所有任务中把任务 t_i 去掉,再返回第一步。若所有的任务都被调度完毕,则退出整个调度过程。Min - Min 算法总是尽可能先分配小任务却不能保证网格资源负载均衡。

表 1 任务在资源上调度时的最小执行时间

	h1	h2
t1	6	X
t2	1	4
t3	10	X
t4	2	8

表 1 所示的数据为网格环境中的四个任务在主机 h1, h2 上的预期执行时间值,其中 X 代表该任务在相应的主机上不能执行,即可以认为该任务在该主机上的与其执行时间为无穷大。由表 1 可知,任务 t1 只能在主机 h1 上执行,任务 t3 只能在主机 h1 上执行,其它任务没有限制。由 Min - Min 调度算法可知,调度的顺序应该为: t2(h1), t4(h1), t1(h1), t3(h1)。即

网格环境中的四个任务都在主机 h1 上执行,而主机 h2 上没有任何任务的执行,这就造成主机 h1 上的负载非常严重,而主机 h2 则是很清闲。这就是 Min-Min 调度算法的缺陷所在,即负载严重不均衡,并且在算法的调度过程中,没有考虑调度过程中的花费问题,导致在整个过程中的花费也是不少的,这是该算法的另一个缺陷所在,文中就是从这两个方面来改进 Min-Min 调度算法的。

3 基于性价比改进的 Min-Min 调度算法

针对 Min-Min 算法的缺点,人们在后来的研究中对 Min-Min 调度算法进行了改进,改进的方式有很多,文献[5]是从优先级的角度考虑任务的调度,从一定程度上提高了调度的效率。文献[6]是从服务质量的高低的角度来考虑任务调度的,任务调度优先考虑服务质量要求高的任务-资源对,这在一定程度上提高了网络的整体性能,提高了算法的效率。文献[7]把网格资源分为:特熟资源和普通资源,并且将任务和资源分别作标记,调度的时候先考虑调度特熟资源,再调度普通资源。还有很多的算法都是从一定程度上改进了 Min-Min 算法,文中也是要改进 Min-Min 调度算法,但是改进的角度和其他的算法不一样,文中是从性价比的角度来考虑改进算法的。

网格的任务调度是要消耗一定的费用的,而且调度的性能也是一个必须要考虑的问题,如何能从费用和性能两方面来提高 Min-Min 算法,这是文中关注的焦点,这里采用经济学的角度来考虑问题。性价比用性能/价格的方式来衡量,得到的比值越高,说明性价比越高,应该优先调度性价比高的任务-资源对来进行调度。

进行以下假设:

1)所有的任务都是原子的,即任务之间不存在相互依赖的关系;如果任务之间存在一定的依赖或通信关系,就用 DAG 图来实现。

2)在分配任务之前,每个任务在所有机器上的最小执行时间是已知的,即 ETC 矩阵是预先知道的。

3)每个机器是独占的,即一个任务被分配到一台机器上时,直到这个任务被执行完,才能进行其它任务的分配。

基于性价比改进的 Min-Min 调度算法的思想是:

网格环境是由 n 个任务 $T = \{t_1, t_2, \dots, t_n\}$ 和 m 个资源 $R = \{r_1, r_2, \dots, r_m\}$ 组成。当还有需要调度的任务时,循环执行如下操作直至所有任务都执行完毕:

(1)对于所有任务中的每一个任务 t_i ,分别计算出

把该任务分配到所有资源上的最小完成时间。

(2)计算每个任务在各个资源上的性价比(即由任务在资源上性价比和单位时间内花费的比值得出),即为 $P/C(i, j)$,表示任务 t_i 在资源 r_j 上的性价比,形成一个性价比矩阵 $P/C[i, j]$ 。

(3)将步骤(2)中计算出来的性价比比值按照由大到小的顺序排列。

(4)设矩阵 $P/C[i, j]$ 中值最大的是第 i 个元素,且对应的主机为 $host$,把任务 t_i 分配到资源 $host$ 上。

(5)将已经调度的任务 t_i 从所有任务中删除。

若需要调度的任务非空,重复以上步骤,直到所有任务分配完毕。

从以上来看,改进后的算法在一定程度上提高了资源利用率,并且在一定程度上提高了网络的整体性能,是一种行之有效的算法。原有的算法最大的缺点是负载不均衡,改进后的算法在一定程度上平衡了主机负载,提高了效率。而且还从一定程度上节省了费用。

假设网格环境由任务 (t_1, t_2, t_3, t_4, t_5) 和资源 (h_1, h_2) 组成,并且每个任务在主机上的最小执行时间(毫秒:ms)和执行时单位时间的花费(美元:dollar)已经给出(见表 2)。

表 2 任务在资源上调度时的时间,花费和性能数值

	主机 h1			主机 h2		
	最小执行时间	执行时的花费	任务调度的性能	最小执行时间	执行时的花费	任务调度的性能
t_1	2	8	10	X	0	12
t_2	5	6	20	9	2	15
t_3	7	3	30	10	4	20
t_4	8	10	40	12	2	25
t_5	15	1	50	20	2	32

从以上分析知,运用原始的 Min-Min 调度算法来执行,所有的任务都在主机 h1 上运行,导致主机 h2 没有任何资源运行,处于空闲状态,主机 h1 负载严重,资源严重失衡。而使用改进后的算法可知,主机 h1, h2 得到充分利用,改进后的算法在资源负载方面明显占了优势。任务的性价比分别为: $t_1(h_1) = 1.25$, $t_1(h_2) = \text{无穷大}$; $t_2(h_1) = 3.3$, $t_2(h_2) = 7.5$; $t_3(h_1) = 10$, $t_3(h_2) = 5$; $t_4(h_1) = 4$, $t_4(h_2) = 12.5$; $t_5(h_1) = 50$, $t_5(h_2) = 16$ 。

所以调度的顺序应该是: $t_1(h_2)$, $t_2(h_2)$, $t_3(h_1)$, $t_4(h_2)$, $t_5(h_1)$ 。从以上的调度情况来看,调度的时候所有的任务并不是全部在主机 h1 上运行,而是分别运行在主机 h1 和 h2 上,改进的算法从一定程度上提高了算法的平衡度,从而弥补了原始算法的缺陷。

经典的 Min-Min 调度算法的缺陷是负载不均衡,并且调度的过程中不能考虑经济效益,文中从性价

比的角度来改进改算法,从一定程度上满足了这两方面的要求。通过建立数学模型的方法验证了改进后的算法比改进前经济效益有所提高,并且能平衡一定的负载。

4 Gridsim 模拟工具

我们知道,建立一个网格试验台是昂贵的,且是耗费资源和耗时的;而且使用真实的环境进行实验会产生很多开销,耗费的不仅仅是人力还有物力,而且任务调度的一些指标难以计算,算法的有效性在这里也得不到体现,所以采用网格模拟器来模拟真实的环境。在模拟环境下,所有的资源会工作在一个真实的经济模型下,进行模拟实验,不仅仅均衡负载可以衡量,最优跨度也能得到很好的体现。

常见的模拟器有 Simgrid^[8,9], Bricks^[10], Microgrid^[11], Gridsim^[12,13]等。SimGrid 的模拟环境是分布式计算平台,能模拟网络拓扑结构、资源和任务的处理等。SimGrid 有很多函数,但是它有很大的缺陷是:模拟的过程非常复杂化。一般的用户难以使用此模拟器。Bricks 模拟器使用 Java 语言书写的,其结构是由全球计算环境和调度单元组成的。其中,全球计算环境包括三个部分:客户机,网络和服务端。调度单元主要有:网络监控器,服务监控器,资源数据库和调度者等。它能够模拟资源的发现等操作,最重要的特点就是集成了 NWS 系统。Microgrid 是建立在 Globus 之上,仿真环境是用 Globus 建立起来的网格环境,能够仿真资源的行为、任务的处理等。仿真结果接近实际,并且仿真的结果可以精确,适用于实际系统开发过程中的模拟。GridSim 的模拟环境是网格计算平台,主要是基于经济的网格资源调度模型,它能够模拟异构资源,主要应用于网格计算。并且 GridSim 中提供了很多例子,可以随时调用其中的类来进行模拟,非常方便。

选择 Gridsim 作为文中的模拟工具,是因为它是一个基于 Java 语言开发的仿真工具,用于对集群、对等计算和网格等分布式计算环境机器调度算法进行建模和仿真。它的目标是通过模拟来研究资源的最优分配方法。而且 Gridsim 主要针对网格计算,它提供丰富的函数库用来支持网格中各种各样的资源,通过消息事件来进行通信,而且 Gridsim 提供一套图形界面工具 Visual Modeler,简称 VM,使用 VM 可以帮助用户并产生相应资源和用户的代码,就不需要用户从头开始写代码了。Gridsim 里提供了很多类,使用起来很方便,模拟完成后,用户可以从 GridSim 的库函数中来收集各种数据。

5 结束语

从性价比的角度改进算法的效率,通过分析表明,改进后的算法在负载平衡度及费用方面有了很大的提高。下一步的工作将构建网格仿真的硬件平台,利用构建的仿真平台对所提出的算法进行模拟试验。并采用与其它算法进行比较的方法分析其性能优劣性,对算法进行优化。继续对计算任务调度算法进行研究,制定出性能更高、更容易实现、更适合计算网格特有环境的新计算任务调度策略。同时,对现有算法进行进一步的验证,使用网格模拟器 Gridsim 进行模拟。

参考文献:

- [1] 徐洪智,李仁发,张彬连,等.一种快速平衡任务的网格调度算法[J].系统仿真学报,2007,19(11):2437-2443.
- [2] 马少静.网格计算中任务调度算法的研究与实现[D].太原:太原理工大学,2007
- [3] 刘宴兵,陈杰,熊仕勇.基于 QoS 相似度的网格任务调度算法[J].重庆邮电大学学报,2009,21(3):416-420.
- [4] 马景奕,隋兵,舒万能.基于 Min-Min 遗传算法的网格任务调度方法[J].计算机工程与应用,2008,44(23):102-104.
- [5] 耀钦,赵霜,宋建社.军用网格环境下基于优先权的 Min-Min 任务调度算法[J].微电子学与计算机,2006,23(12):8-11.
- [6] 吴豪,吴光文.基于 QoS 约束与 Min-Min 算法的网格资源调度研究[J].信息科学,2009(11):8-11.
- [7] 吴高峰,蒋玉明,杨林,等.基于 QoS 改进的 Min-Min 网格调度算法[J].微计算机信息,2009,25(9-3):110-112.
- [8] 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,200,32(2):42-44
- [9] Casanova H, Simgrid. A Toolkit for the Simulation of Application Scheduling[C]//proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001). Brisbane, Australia: [s. n.], 2001.
- [10] Aida K, Takefusa A, Nakada H, et al. Performance evaluation model for scheduling in a global computing system[J]. The International Journal of High Performance Computing Applications, 2000, 14(3): 268-279.
- [11] Song H, Liu X, Jakobsen D, et al. The MicroGrid: A Scientific Tool for Modeling Computation Grids[C]// Proceedings of the IEEE Supercomputing (SC 2000). Dallas, USA: [s. n.], 2000.
- [12] Buyya R, Murshed M. Gridsim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing[J]. The Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002, 14(13-15): 1175-1220.
- [13] 高强,刘波.关于网格模拟器的研究[J].计算机技术与发展,2010,20(1):100-103.