CN43-1258/TP
ISSN 1007-130X

计算机工程与科学
COMPUTER ENGINEERING & SCIENCE

2012 年第 34 卷第 2 期
Vol. 34, No. 2, 2012

文章编号:1007-130X(2012)02-0082-05

分布式计算环境中的协同分配任务调度仿真系统*

A Simulating System for Scheduling Co-Allocation Tasks in Distributed Computing Environments

李　波[1], 周恩卫[1], 沈　斌[2]

LI Bo[1], ZHOU En-wei[1], SHEN Bin[2]

(1. 云南大学信息学院, 云南 昆明 650091; 2. 武汉工程大学电气信息学院, 湖北 武汉 430073)

(1. School of Infomation Science and Engineering, Yunnan University, Kunming 650091;
2. School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430073, China)

摘　要:协同分配是在分布式计算环境中进行资源分配的一种重要技术,用于把一个应用程序分解为多个子作业,然后将其分配到多个资源上同时处理来满足特定的性能要求。本文提出了一个离散事件驱动的网格资源协同分配仿真系统,实现了对用户、调度器、协同分配器、协同预留器等协同分配相关实体的仿真,实现了 FCFS、FPFS、Backfill 等主要的协同分配调度算法和策略,可用于资源协同分配相关的分布式计算环境的资源管理和调度算法的仿真和研究。

Abstract: Resource co-allocation is one of the crucial technologies affecting the utility and quality of services of large-scale Grid-like distributed environments by simultaneously allocating the requested multiple resources to a single submitted application to meet the specific performance requirements. This paper presents a discrete-event driven simulator for studing the co-allocation-related resource management and scheduling algorithms. This tool models the main Grid components and their functions for co-allocating grid jobs, including Grid user, metascheduler, co-allocator, and co-reservator. Some main scheduling algorithms and policies, including First Come First Served, First Processor First Served and Backfilling, are implemented.

关键词:分布式计算;协同分配;协同预留;仿真

Key words: distributed computing; co-allocation; co-reservation; simulation

doi: 10.3969/j.issn.1007-130X.2012.02.016

中图分类号: TP391.9　　**文献标识码**: A

1　引言

网格计算通过 Internet 把分散在各处的硬件、软件、信息资源连结成为一个整体,形成一个巨大的资源池,供人们使用,完成各种大规模的、复杂的计算和数据处理任务。当处理网格应用程序的时候,如果一个作业必须同时使用多个分布在不同地方或者管理域的资源才能进行处理,就需要使用到协同分配(Co-Allocation)技术[1]。通过协同分配技术,可以提高任务在分布式计算环境中的性能,广泛应用于虚拟现实、虚拟仪器、大规模科学计算等领域[2~4]。为了支持资源协同分配,必须保证所需资源在同一时间段的可用性。到目前为止,已经

* **收稿日期**:2010-12-07;**修订日期**:2011-04-12
基金项目:国家自然科学基金资助项目(60663009)
通讯地址:650091 云南省昆明市云南大学信息学院
Address: School of Infomation Science and Engineering, Yunnan University, Kunming, Yunnan 650091, P. R. China

有很多本地调度器提供了对提前预留的支持。通过向多个本地调度器提交资源预留请求,网格调度器能够实现多个资源的协同预留(Co-Reservation)[1]。

随着网格技术的不断发展,开发能够仿真网格计算环境的软件越来越被重视,在网格资源管理和调度算法的研究过程中,研究者试图通过网格仿真工具对提出的算法进行分析和比较。在网格计算的研究中,研究者已经开发了一些网格仿真工具,主要有 Bricks[5]、MicroGrid[6]、SimGrid[7]、GridSim[8]、OptorSim[9] 和 ChicSim[10]。在这些工具中,Bricks、SimGrid 和 GridSim 主要用于计算网格的作业调度算法研究,OptorSim 和 ChicSim 主要用于数据网格中的作业调度与数据管理策略的研究。然而到目前为止,这些仿真工具都不能够支持网格计算环境中的协同预留和协同调度的仿真。

在前期研究中[11],我们已经开发了一个能够支持基于 Java 的离散事件驱动的仿真平台,实现了对支持资源预留的独立网格任务的资源管理和调度的仿真功能。在此基础上,本文进一步实现了支持网格资源协同分配的仿真功能,实现了对协同分配作业用户、协同分配器、协同预留等协同分配相关实体的仿真。本仿真软件弥补了现有仿真工具在协同预留和协同调度仿真功能方面的不足,具有仿真效率高、可扩展性强等优势,可用于资源协同预留和协同分配相关的分布式计算环境的资源管理和调度算法的仿真和研究。

本文首先介绍了支持网格资源协同分配的仿真软件的体系结构、主要组件和工作流程,然后通过一个实例介绍了该软件的仿真过程。

2　体系结构

本文在文献[12]的仿真软件的基础上实现了协同调度功能(见图 1)。

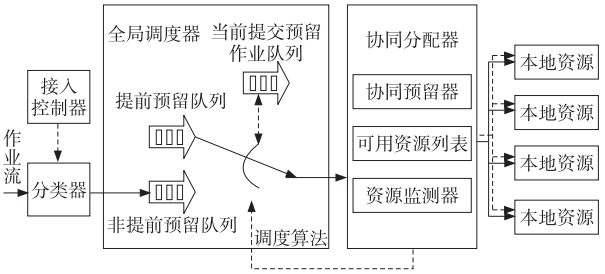


图 1　协同调度仿真框图

关于基于离散事件驱动的仿真系统的基本原理、本仿真系统所使用的离散事件驱动软件开发包、本仿真软件的整体结构,以及对支持资源预留的独立网格任务的资源管理和调度的仿真功能等信息参见文献[12]。本文主要针对如何进行网格资源协同分配仿真的相关部分进行说明。

在图 1 中,机器、处理单元 PE(Processing Element,简称 PE)以及相应的本地资源管理器构成了本地资源 LR(Local Resource,简称 LR)。本地资源用于执行和管理提交到本资源的本地作业或网格作业。本地资源管理器主要由本地分类器、调度器和监测器组成。分类器用于按照作业分类策略对到达的作业进行分类。调度器按照调度算法选择等待队列中的作业,为选中的作业分配资源,然后将其发送到处理单元。本地监测器负责监控本地资源所有 PE 的状态,并在需要的时候将信息提供给全局资源管理器。全局资源管理器主要由分类器、接入控制器、全局调度器和协同分配器组成,用于实现对网格资源和网格作业的管理。对于仅需要单个本地资源就可以执行的作业,可以在将作业提交到本地资源之前通过资源预留器为其预留资源。对于协同调度作业,通过协同调度器为其在多个本地资源进行协同预留,然后将各个子作业分别提交到本地资源。

网格资源协同分配仿真主要涉及网格用户实体和全局资源管理器实体。网格用户实体用于产生需要进行资源协同的网格作业,全局资源管理器用于实现对协同分配作业的调度和执行过程的管理。

2.1　网格用户实体

网格用户实体用于产生网格作业,可通过接口与一个或者多个全局调度器相连来提交网格作业。生成的网格作业可以是只需要单个本地资源就可以执行的作业,也可以是同时需要多个本地资源才能执行的协同分配作业。网格作业包括了子作业数量、本地处理单元数量、到达时间、估计运行时间、实际运行时间等信息。对于只需要单个本地资源就可以执行的作业,设置其子作业数量为 1,本地处理单元数量为该作业所需的处理单元数量。对于需要协同分配的作业,需要划分子作业的数量,并设置每个子作业所需本地处理单元数量。网格作业及其参数可以按照指定参数的随机分布来生成,也可以从并行作业量文档 PWA(Parallel Workload Archive,简称 PWA)[14] 或者 DAS2[13] 等实际运行环境中的作业记录中提取。

在仿真开始之前,需要为网格用户实体设置用户类型、作业类型、仿真作业数量、作业到达率、提

前预留参数、协同分配参数等控制参数,然后网格用户实体根据这些参数来产生相应的作业队列,并在仿真过程中根据作业的到达时间,将作业提交到资源管理器。对于协同分配作业,根据作业提交时是否为子任务指定了所用的资源,可以分为静态协同作业和动态协同作业。其中,静态协同作业指定了子作业的划分、每个子作业所需的处理单元数量以及执行的本地资源,而动态协同分配作业仅指定了子作业的划分和每个子作业所需的处理单元数量,执行子作业的本地资源可以动态选择。现有的并行作业量文档 PWA 或者 DAS2 中的作业记录都仅仅是各个多处理机服务器上运行的作业记录,为了将这些作业记录用于协同调度仿真,网格用户实体会根据作业所用的处理机数量,将这些记录中的单个作业划分成多个子作业,这些子作业就构成了一个协同分配作业,在多个服务器上同步运行。在网格用户实体中可以修改子作业的划分方法。

2.2 全局资源管理器

全局资源管理器是支持协同分配仿真的核心部分,包括了作业分类器、全局调度器、协同分配器等实体,用于实现对协同分配作业的调度和执行过程的管理。作业分类器用于对协同分配作业按照静态协同和动态协同进行分类,协同分配器用于将协同分配作业的各个子作业分配到合适的本地资源,而全局调度器按照协同分配器的资源协同分配结果将协同分配作业的各个子作业调度到对应的本地资源并监测执行过程。

协同分配器包括了资源监测器、可用资源列表和协同预留器三个模块,用于根据协同分配作业的子作业的属性和可用资源状况进行子作业到本地资源的映射,是网格资源协同分配的核心部分。资源监测器用于监测和本协同分配器相连的本地资源的状态,用于查询或者接收本地资源的状态信息,包括空闲处理单元数量、资源在某个时间段的分配情况等。然后协同分配器根据协同分配子作业的到达时间、所需的处理单元数量、估计运行时间等信息选择合适的本地资源和处理单元。如果能够为一个协同分配作业的所有子作业找到可以同时运行的可用资源,就完成了该协同分配作业的资源映射过程。为了保证协同分配作业所需的多个本地资源的可用性,协同预留器通过为每个协同分配任务的子任务寻找能够让这些子任务同步运行的可用资源并进行预留,实现了在同一个时间段内在多个本地资源为协同分配作业预留资源的功能。

2.3 协同分配算法

在网格之类的分布式运行环境中,有可能同时有多个协同分配作业需要处理,因此需要通过适当的协同分配调度算法来对这些作业进行调度。本文的仿真器主要实现了 FCFS(First Come First Served,简称 FCFS)算法、FPFS(First Processor First Served,简称 FPFS)算法和 Backfill 算法。

FCFS:在到达的作业队列中,第一个作业的优先级是最高的。在每次调度中,都是检测可用资源是否适合队列中的第一个作业。如果适合,就马上调度它;如果不适合,就一直等待,直到下次调度。

FPFS:在每次调度中,调度器给予到达作业队列中第一个适合可用资源的作业最高的优先级,并不要求必须是到达作业队列的第一个作业。如果在队列中存在这样的作业,就立即进行处理;如果没有这样的作业就一直等到,直到下次调度。

Backfill 算法:Backfill 策略与 FPFS 算法相似。如果当前作业不能被执行,则检查是否可以对其进行资源预留,如果可以预留,则为其预留资源,然后再依次检查后面的作业是否可以在不影响正在运行的作业和已经预留作业的前提下能够在当前时刻运行。常用的 Backfill 调度算法是 EASY Backfill 和 Conservative Backfill,前者的预留作业数为 1,后者的预留作业数可以不限。在选择回填作业时,可以采用 First Fit、Best Fit 等策略。

在有多个协同分配任务在等待处理时,本仿真器可采用上述算法来对多个任务进行调度。而对于不需要协同预留的普通任务,在本仿真器中主要采用了 FCFS、EASY Backfill 和 Conservative Backfill 三种调度算法。相对于只由单个任务组成的不需要协同预留的普通任务,这些算法在用于协同分配任务时,需要同时考虑协同分配任务的多个子任务。只有可用资源能够满足当前协同分配任务的所有子任务的要求时才能进行调度。除了上述算法之外,还可以通过扩展本仿真软件中的调度算法,从而实现对其他算法的仿真和研究工作。

3 仿真实验过程

由于本仿真平台使用了 SimJava[12] 的开发包完成底层离散事件驱动功能,因此在使用时按照 SimJava 开发包指定的使用方法即可。仿真是通过主函数 MetaSchedule 实现的,通过本论文介绍的仿真平台,可以对网格资源的协同分配和协同预

留进行仿真,可以实现不同的协同分配算法。仿真的基本流程如下:

(1)设置仿真控制参数,进行仿真系统初始化,包括仿真停止条件、事件跟踪细节等。

(2)设置仿真实体参数并初始化,如作业数量、作业类型、用户数量、协同分配算法类型、仿真月数等等。

(3)运行仿真并分析仿真结果。通过实验可以得到平均扩展因子 AXF(Average eXpansion Factor,简称 AXF)和加权平均响应时间 AWRT(Average Weighted Response Time,简称 AWRT)两个性能参数。其中 AXF 为所有作业的扩展因子的平均值,一个作业的扩展因子 XF(eXpansion Factor),也称为减缓(Slowdown),定义为:

$$Xfactor = \frac{\text{响应时间}}{\text{执行时间}} = \frac{endTime_j - submitTime_j}{endTime_j - startTime_j}$$

$AWRT$ 考虑了作业的处理器数量,定义为:

$$AWRT = \frac{\sum_{j \in Jobs} PE_j \times (endTime_j - submitTime_j)}{\sum_{j \in Jobs} PE_j \times (endTime_j - startTime_j)}$$

通过本仿真平台进行仿真实验,利用上述的两个性能参数,可以对本仿真平台所实现的协同分配算法的性能进行比较,同时可以分析设置不同参数对系统性能的影响。

4　实验举例

在仿真过程中,采用了 DAS2[13] 的实际运行作业记录。在作业记录中,由于各个记录时间单位长度是独立的,我们就对这些不同时间单位长度进行了归一化处理。在进行实验仿真的时候,首先设置作业的到达率、仿真月数、预留比率、协同调度算法、Backfill 原则、是否是 Ordered 作业、能否准确估计作业执行时间,然后对机器、网格用户、网格调度器、网格调度器包含的机器代理进行创建和初始化,并将各个实体通过物理接口连接起来,最后再进行实验报告的填写。实验结果表明,作业类型分别为静态协同作业和动态协同时作业,作业估计运行时间不等于实际运行时间。FCFS 和 EASY backfill 的性能如表 1 所示。可以看出,动态协同分配比静态协同分配能够更有效地改善作业性能,而 EASY backfill 的性能优于 FCFS。当作业运行时间分别为准确估计和不准确估计时,EASY backfill 算法的性能如表 2 所示。可以看出,对于本仿真作业量,作业估计运行时间准确时的性能优于不准确时的性能。

在实验过程中,可以通过参数的选择和设置,得出不同参数值和不同算法的性能参数值,从而研究参数对系统性能的影响和比较算法的性能。

表 1　作业运行时间不准确估计时静态协同分配和动态协同分配时的性能

	FCFS		EASY	
	AXF	*AWRT*	*AXF*	*AWRT*
静态协同分配	2.526	1.171	1.596	1.117
动态协同分配	1.698	1.141	1.585	1.117

表 2　作业估计运行时间对 EASY backfill 算法性能的影响

	不准确估计		准确估计	
	AXF	*AWRT*	*AXF*	*AWRT*
静态协同分配	1.596	1.117	1.555	1.115
动态协同分配	1.585	1.117	1.464	1.113

5　结束语

本文提出了支持网格作业协同分配和协同预留的仿真平台,介绍了这个平台的体系结构、各个实体以及实体的交互过程。通过本仿真平台,可以实现网格资源的协同分配和协同预留的仿真。该仿真平台不仅可用于网格计算环境,还可以用于其它需要进行资源协同分配的分布式系统中。在未来的研究就中,将通过更多的实验,分析各个参数对系统性能的影响,并通过实验进行协同调度和协同预留算法的改进,以及对提出的新算法通过试验进行评估。

参考文献:

[1] Czajkowski K, Foster I, Kesselman C. Resource Co-Allocation in Computational Grids[C]// Proc of the 8th International Symposium on High Performance Distributed Computing, 1999:219-228.

[2] Sonmez O, Mohamed H, Epema D. On the Benefit of Processor Co-Allocation in Multicluster Grid Systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2009, 21(6):778-789.

[3] Chang Y-S, Zou G-J, Chang C-L. RARS: A Resource-Aware Replica Selection and Co-Allocation Scheme for Mobile Grid[J]. International Journal of Ad Hoc and Ubiquitous Computing, 2010, 6(2):99-113.

[4] 肖鹏, 胡志刚. 面向实时网格任务的多策略资源协同分配模型[J]. 吉林大学学报(工学版), 2010, 40(1):218-223.

[5] Takefusa A M S, Nakada H. Overview of a Performance Evaluation System for Global Computing Scheduling Algorithms[C]// Proc of the 8th IEEE International Symposium on High Performance Distributing Computing, 1999:97-104.

[6] Song H J, Liu X, Jakobsen D, et al. The Microgrid: A Scientific Tool for Modelling Computational Grids[J]. Scientific Programming, 2000, 8(3): 127-141.

[7] Legrand A M L, Casanova H. Scheduling Distributed Applications: The Simgrid Simulation Framework[C]// Proc of the Third IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003: 138-145.

[8] Buyya R M M. Gridsim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing[J]. Concurrency and Computation: Practice and Experience, 2002, 14(13-15): 1175-1220.

[9] Cameron D G, Carvajal-Schiaffino, Millar A P, et al. Evaluating Scheduling and Replica Optimisation Strategies in Optorsim[C]// Proc of the 4th International Workshop on Grid Computing, 2003: 52-59.

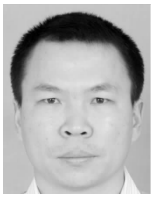
[10] Ranganathan K F I. Simulation Sstudies of Computation and Data Scheduling Algorithms for Data Grids[J]. Journal of Grid Computing, 2003, 1(1): 53-62.

[11] 李波, 赵东风, 沈斌. 支持资源预留的网格计算仿真平台[J]. 系统仿真学报, 2006, 18(Suppl 2): 373-376.

[12] Howell F, McNab R. SimJava: A Discrete Event Simulation Package for Java With Applications In Computer Systems Modelling[C]// Proc of the First International Conference on Web-Based Modelling and Simulation, 1998: 483-488.

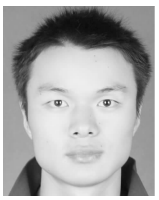
[13] Bucur A I D, Epema D H J. Trace-Based Simulations of Processor Co-Allocation Policies in Multiclusters[C]// Proc of the 12th IEEE International Symposium on High Performance Distributed Computing, 2003: 70-79.

[14] Feitelson D. Parallel Workloads Archive[EB/OL]. [2011-03-02]. http://www.cs.huji.ac.il/labs/parallel/workload.



李波(1976-),男,云南曲靖人,博士,副教授,研究方向为并行和分布式计算。**E-mail**: liboynu@163.com

LI Bo, born in 1976, PhD, associate professor, his research interests include parallel and distributed computing.



周恩卫(1984-),男,陕西咸阳人,硕士生,研究方向为并行和分布式计算。**E-mail**: 281980512@qq.com

ZHOU En-wei, born in 1984, MS candidate, his research interests include parallel and distributed computing.

沈斌(1973-),男,湖北武汉人,博士,副教授,研究方向为网络和分布式计算。**E-mail**: shenbingwhict@126.com

SHEN Bin, born in 1973, PhD, associate professor, his research interests include network and distributed computing.