维普资讯 http://www.cqvip.com

基于多代理的网格任务调度研究*)

曾万聃[1] 周绪波[2] 戴 勃[1] 游新冬[1] 常桂然[1]

(东北大学信息科学与工程学院 沈阳110004)[1](清华大学软件学院 北京100081)[2]

摘 要 随着网格技术的发展,代理技术近年来被用在网格的实现当中。多代理技术用分布式自主结构代替集中式的非自主性结构,具有更强的实时性,特别适合于动态调度,本文提出的网格任务调度系统就采用了多代理的体系结构。由于Globus已经提供了网格操作系统的大部分功能,本系统构建在Globus之上。在充分利用底层Globus提供的功能并与之结合的基础上,在代理中加入人工智能、知识学习的方法和服务质量实现策略,对网格任务进行灵活智能的调度,实现更好的负载平衡并达到一定的服务质量。采用JATLite来创建该系统,利用该系统可以根据应用背景需求来快速定制动态任务调度平台。

关键词 网格,智能代理,多代理,JATLite

Research on Grid Task Scheduling Based on Multi-Agent

ZENG Wan-Dan[1] ZHOU Xu-Bo[2] DAI Bo[1] YOU Xin-Dong[1] CHANG Gui-Ran[1]

(School of Information Science and Engineering, Northeastern University, Shenyang 110004)[1]

(Software Institute, Tinshua University, Beijing 100081)[2]

Abstract With the development of Grid technology, agent technology has been utilized in the implementation of Grids. The distributed autonomous infrastructure of multi-agent can achieve better real-time response than the centralized non-autonomous ones, and it is very suitable to dynamic scheduling. Because Globus has provided most of the functions of a Grid operating system, our system is based on Globus. This system not only makes full use of and integrates with the functions of Globus, but also utilizes artificial intelligence and machine learning methods and adopts the strategies of QoS, to achieve efficient scheduling of Grid tasks with certain levels of QoS assurance. JATLite is used to build the system. This system provides a platform with which a dynamic task scheduling system satisfying the application requirements can be created quickly and efficiently.

Keywords Grid, Agent, Multi-agent, JATLite

1 引言

网格彻底地改变了计算机和数据的访问方式,将成为下一代分布式计算的体系结构标准,它提供从孤立的系统到紧密结合的簇、企业范围内聚簇及地理上分散的计算机环境之间联系的途径[1]。网格技术使用户无论在何时何地都能透明地访问计算和存储资源,并保证一定的服务质量成为可能[2]。由于网格环境的异构性和复杂性,为网格资源提供透明的无缝的并且稳定可靠的服务成为一大难题。代理技术多年前就已经被用于解决计算机负载均衡的问题,近几年代理技术也开始被应用在网格的实现当中[3]。

在网格体系结构中,由下至上的四个管理层分别是:构造层,连接层,资源层,汇聚层。资源层包括计算资源、存储资源、网络资源、代码库等等;连接层处理简单的安全通讯,提供单点登录、代理,与不同的本地安全方案的结合以及基于用户的信任关系;资源层关注于单个资源,资源层协议的两个基本类就是信息协议和管理协议;汇聚层提供目录服务、协同分配、服务发现和调度、监控和诊断服务、数据复制服务、基于网格的编程系统,等等。

在广域分布的系统中实现汇聚资源的协调是复杂的高层任务,服务的发现、协同分配,及其监控、诊断、授权等等这些过程需要智能的、自治的和社会性的能力,智能代理都满足这些特性[4]。服务的可扩展性问题是网格的一个关键问题,资源管理、协同分配和调度等这些服务都必须考虑到系统的可扩展性问题[5]。基于代理的技术是能够提供可扩展性和自适应性服务的最有前景的一种方法。本文将多代理的体系结构用于组织和协调网格环境中的分布式资源调度,讨论了基于多代理的网格任务调度的思想、设计和实现方法。

*)基金项目:高等学校博士学科点专项科研基金(20030145017)。曾万聃 博士生。

参 考 文 献

1 汪斌强,王建东.高性能路由器技术体系、关键问题及发展趋势.电信科学,2003(10)

2 Wang Jun, Klara Nahrstedt Parallel IP Packet Forwarding for Tomorrow's IP Routers. In: Proc. of 2001 IEEE Workshop on High Performance Switching and Routing (HPSR'01), Dallas, TX, May 2001. 353～357

3 肖萍萍,周芳.SDH原理与技术.北邮出版社,2002

4 Simpson W. PPP in HDLC-like Framing. STD 51, RFC 1662, July 1994

5 Xilinx. Virtex-II Pro Platform FPGA Handbook. 2003

6 Altera. Stratix Device Handbook. 2003

7 Cypress. Ayama 10000 Network Search Engine. April 2003

8 Netlogic. Nse5000 Network Search Engine Device. Nov. 2002

9 Waldvogel M, Varghese G, et al. Scalable High-Speed IP Routing Lookups. Procedures ACM SIGCOMM, 1997. 25～36

10 徐恪,吴建平,吴剑.基于TCAM的高速路由查找.小型微型计算机系统增刊(CERNET 2001学术年会)

11 Gupta P. Algorithms for routing lookups and packet classification: [PHD thesis]. http://klamath.stanford.edu/～pankaj/thesis/thesis 1sided.pdf, Dec. 2000

维普资讯 http://www.cqvip.com

2 相关研究工作

近年来已经有一些研究实例将代理用在网格中。CoABS 代理网格提供了一个统一的、异构的分布式计算环境，在这个环境中计算资源可以无缝的连接；CoABS 网格体系结构提供广域的异构架构的集成[6]。

Rana 和 Walker 提出 PSEs (problem solving environments) 支持不同工具和特定算法之间的互操作，每一个代理代表集成服务和资源的不同的 PSE。代理包含的行为规则可以根据他们与其他代理或环境的交互进行修改[7]。

ACG 将网格环境中的所有实体，包括计算资源和服务，都作为代理，它实现了网格计算资源和服务的高层统一管理，为用户提供一个一致的透明的访问服务的接口[3]。

采用代理技术可以用来构建更好的可靠性、可扩展性、生存性、进化性和自适应性的系统，这些恰恰都是构建网格应用系统的需求。

3 面向代理的方法和多代理技术

面向代理的技术作为一种新的分析和设计复杂软件系统的方法，极大地推动了现有软件工程实践，是适合于构建复杂系统的解决方案；面向代理的方法可以适应复杂系统对动态组织关系和结构进行管理的需求[8]。

面向代理的系统分解是对复杂系统的问题空间进行划分的有效途径，采用面向代理的方法进行软件工程的设计，意味着将问题分解为具有实现特定目标的多个相互作用的自治实体，即代理。软件代理是一个在给定的环境下封装的软件实体，它能够采取灵活、自治的行为，适应环境，并且完成自身的功能；它具有一定的智能，能被动地响应信息处理的需求，除了完成事先制定的任务外，还能够主动地预测、适应乃至积极地寻找途径以支持用户完成任务[4]。多代理调度系统用自主模块构成的分布式结构代替传统的集中式非自主性结构，实际的调度执行更多的是通过多个代理协商来完成，而不是完全的预先计划，因此具有更强的实时性，特别适合于网格环境中的任务调度。

4 系统设计

多是采用代理的思想和方法建立一个全新的网格平台，例如：ACG，NetSolve，CoABS 等等；但是考虑到我们现在的实际情况，并且借鉴欧洲数据网格 DataGrid、分布式异构计算环境 Cactus 等一些成功的网格平台的设计思想，将系统架构在 Globus 平台之上。因为 Globus 作为一个网格基础设施平台，已经实现了本文简介中所提到的网格体系结构中基本的功能，相当于网格操作系统。当然 Globus 与具体的应用还有一定距离，所以本文中提出的多代理调度系统架构在 Globus 之上，在其基础上扩展具体应用中实现复杂调度的具体解决办法。在代理中加入人工智能、知识学习的方法和服务质量实现策略，对网格任务进行灵活智能的调度，实现更好的负载平衡并达到一定的服务质量，并在系统设计和实现过程中重视该系统与 Globus 所提供的逻辑功能与物理模块的更完美的结合。我们采用 JATLite 来实现该系统，JATLite 是一个能够快速创建新的代理系统的工具，可以根据应用的需求灵活定制和创建所需的代理，大大缩短开发周期，提高软件的灵活性。

考虑到网格资源分布范围广且动态变化等特点，我们在这里采用多代理的体系结构。将地理位置或者逻辑关系紧密的一定数量的网格节点作为一个网格区域进行管理。在每一个网格应用区域范围的接入点设置一个管理代理(MA)，在每个网格节点安装一个本地代理(LA)。

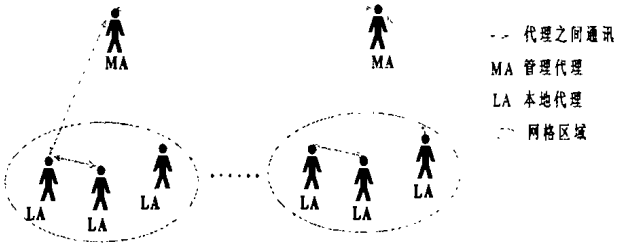

图1 系统体系结构视图

管理代理主要负责处理不同网格区域范围内的任务调度问题，也可以对本区域的代理之间的任务分配进行协调。当有本区域范围之外的任务请求到达本区域时，管理代理负责处理该请求，并与本地代理进行交互，实现外来任务在本区域的任务调度；本地代理既可以与本区域内的其他本地代理进行通信，也可以与本区域的管理代理进行通信。

本地代理需要对其所在的网格节点的资源可用性和负载进行管理，并负责该节点与本区域内部其他网格节点的本地代理和管理代理进行交互通信。当其他网格区域向本区域的管理代理发出服务请求时，管理代理把需求发给本地代理，本地代理给管理代理做出相应的应答，如果可能，为管理代理提供满足其要求的服务。当任务需要从一个节点转移到网格中的另外节点或者同一个任务分解到不同的网格节点时，本地代理还要与本区域内的其他本地代理进行通信，处理与之相关的一些问题和操作。本地代理还可根据任务需求设定相关的限制条件和服务质量策略，并对其进行实现。本地代理还利用下层的网格操作系统 Globus 提供的服务来实现负载平衡、管理服务质量、分配计算任务等功能。由于系统底层是 Globus 网格平台，对于每一个网格节点，本地代理的功能是建立在 Globus 提供的功能的基础之上，网格节点的功能逻辑结构如图2所示。

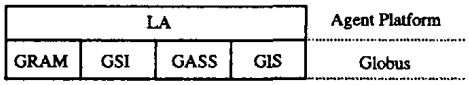

图2 网格节点逻辑结构图

5 系统实现

JATLite 是一个使用户能够快速创建新的代理系统的工具，它使用代理通信语言 KQML 作为底层的通信语言，它使用消息路由器 AMR 来保证代理间消息的传递，可以使每个代理通信时不一定要知道其他代理的 IP 地址。代理消息路由器(AMR)服务可以允许代理失败、重新启动、移动，甚至连 Java 的 Applet 都可以成为代理。JATLite 利用 Java 面向对象的特点，可以在它的基本模板上根据需要构建特定的类，对类进行扩展。JATLite 的层次结构如图3所示，由底向上逐步由通用向专用变化，不同的层次是逐层特殊化的关系。每一层都可以独立地使用可行的技术按照需求进行开发扩展，而不影响其他层软件的运行，可以根据需求在代理中加入对不同层次功能的扩展。

我们可以根据需要在 JATLite 的各个层次上构建系统所需的功能，JATLite 的这个特点极大地提高了系统的智能性和可扩展性；根据所需可以在任意层次加入任务调度和服务质量控制策略和方法。

维普资讯 http://www.cqvip.com

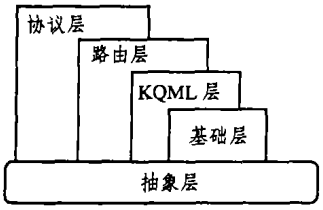

图3 JATLite 层次结构

由于网格应用具有地理分布广、结构复杂等特点，使用一个 AMR 容易造成系统瓶颈，我们这里采用多个 AMR 相互连接来为代理提供更快更好的服务。AMR 设置在网格中的什么位置，要根据具体网格的地理分布及其任务负载来确定。由于多个 AMR 和单个 AMR 系统逻辑结构和功能是相同的，在这里只示意性地给出只有一个 AMR 的系统结构示意图，如图4所示。

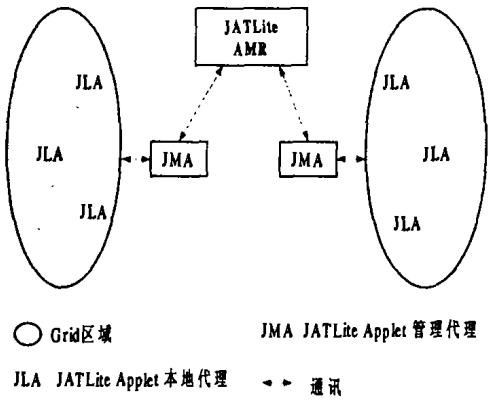

图4 基于 JATLite 的系统示意图

在这里，本地代理和区域管理代理都是完成相应功能的 JATLite Applet，可以在其中加入智能学习、服务质量等功能。本地代理的 JATLite Applet 之间可以相互通信协商，进行任务分解调度。不同区域的任务请求由 AMR 发送到本地管理代理，管理代理与本地代理进行通信，找到可以完成任务的可用节点。然后，管理代理将得到的结果发送回 AMR，AMR 再将这个结果返回给发出请求的管理代理，管理代理再将结果发回给发出请求的本地代理，这样就实现了不同区域网格任务的调度。对于网格节点来说，调用不同区域的网格节点执行任务对它来说是完全透明的，就像在本地执行任务调度一样执行相同的操作，返回的结果也和本地的任务返回结果形式完全相同。

网格中的一个任务请求在该系统中的执行顺序图，如图5所示。

首先，当一个网格节点的本地代理向它所在区域的管理代理发送任务请求时，管理代理经过处理后，如果发现该区域的其他网格节点可以完成该任务，则把该结果返回给发送请求的本地代理，由该代理与能完成该任务的节点的代理直接通信，进行区域范围内的任务调度。否则，管理代理向 AMR 发送任务请求，AMR 得到该请求之后向其他网格区域的管理代理发送任务请求，该管理代理再向它所在的区域的本地代理发送任务请求，然后将结果返回给 AMR，AMR 再将结果返回给发送请求的管理代理，由它将结果发给发送任务请求的代理。

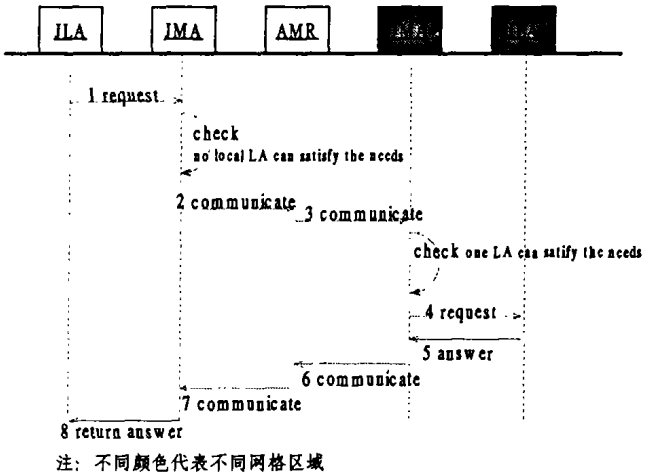

图5 网格任务请求在系统中的执行顺序图

6 系统的 Petri 网性能模型

根据以上实现原理，我们对多代理网格调度系统进行了 Petri 网建模，以便更好地研究其详细流程和性能。多代理网格调度系统的 Petri 网模型如图6所示。

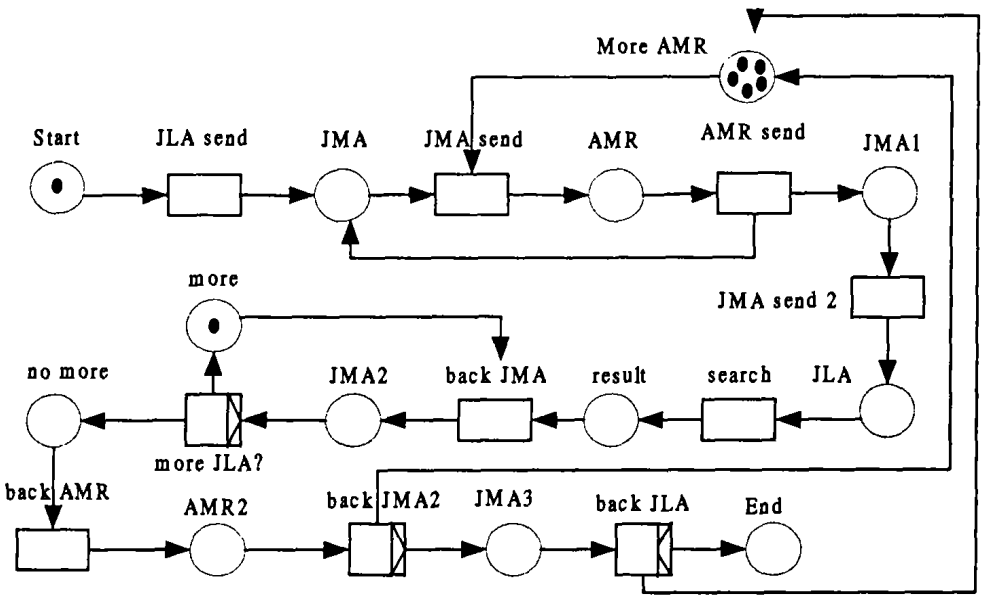

图6 多代理网格调度系统的 Petri 网模型

维普资讯 http://www.cqvip.com

如图中所示，本地代理的管理代理在收到本地代理发出的 JLA send 资源请求消息后，将此消息转发给其对应的5个 AMR(为说明方便，我们在这里假定是5个)。每个 AMR 在收到资源请求消息后，再转发给其他区域的多代理 JMA1，然后由 JMA1转发给 JLA。JLA 在收到请求后，进行相应的 search 操作，得到相应的结果，此时 back JMA 事件触发。管理代理 JMA2在得到所有相应的 JLA 的返回结果后，执行 back AMR 事件，将结果返回给 AMR2，再返回给 JMA3和发送资源请求的 JLA。当所有 AMR 都返回结果时，系统流程结束。

结束语 下一步我们将对原型系统进行实现。对系统的功能做出更详细的设计，对各个代理的功能和代理之间的接口进行更进一步的设计实现；在充分考虑到该系统与 Globus 所提供的各逻辑功能的结合的同时，将人工智能和知识学习的方法嵌入到代理当中。该系统力求实现对网格任务的更加灵活高效地调度，并具有良好的可扩展性和可移植性。

参 考 文 献

1 Foster I. The anatomy of the Grid: Enabling scalable virtual organizations [J]. Concurrency and Computation: Practice and Experience. 2001,13

2 Bruneo D,et al. Communication Paradigms for Mobile Grid Users [C]. In:Proc. of the 3rd IEEE/ACM Intl. Symposium on Cluster Computing and the Grid. 2003 IEEE

3 Li Chunlin,Li Layuan. Agent framework to support the computational grid [J]. The Journal of Systems and Software. 2004,70: 177～187

4 Wooldridge M J, Jennings N R. Intelligent agents: Theory and practice [J]. The Knowledge Engineering Review, 1995, 10(2): 115～152

5 Wijngaards N J E,Overeinder B J,van Steen M,Brazier F M T. Supporting Internet-scale multi-agent systems [J]. Data and Knowledge Engineering,2002

6 Manola F, Thompson C. Characterizing the agent grid. http://www. objs. com/agility/techreports/990623-characterizing-the-agentgrid. html, June 1999

7 Rana O F, Walker D W. The Agent Grid: Agent-based resource integration in PSEs [A]. In: Proc. of the 16th IMACS World Congress on Scientific Computing, Applied Mathematics and Simulation. Lausanne, Switzerland, Aug. 2000

8 Yushun F, Junwei C. Multi-Agent Systems: Theory, Method and Applications [M]. German: Springer Press, 2002, 5

9 van der Aalst W, van Hee K M. Workflow Management: Models, Methods, and Systems [M]. 2002

(上接第13页)

当应用有新的需求向动态绑定层注册时，将激发又一次的需求请求复制和查找过程。当 perTrader 中注册的 service type 或 service offer 发生变化时(通过 storage factor 了解)，同样也激活又一次的查询过程，并将相应的结果返回给动态绑定层，动态绑定层依然是经过仲裁后，填写 Map table。

应用需要获得服务引用时，只需要查询 Map table，而不用关心下层环境的变化。动态绑定层自动完成需求到相应服务的动态更新映射。

如图5所示，环境中可用的服务发生了变化，原来的服务①不能使用了，同时出现了新的服务②，动态绑定层和发现联邦层结合实现需求和提供之间的动态绑定，应用始终能获得可利用的服务。

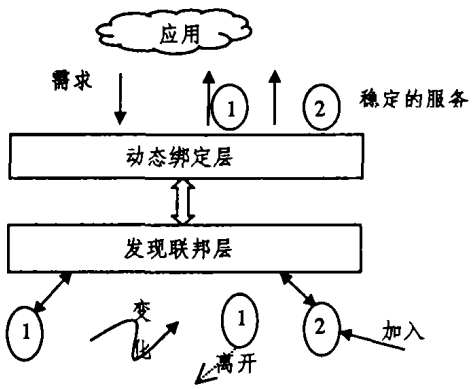

图5 服务的动态绑定

5 试验验证

为验证 CWDB 对动态绑定功能的支持，我们在 PDA 上实现了一个位置显示应用，PDA 具有 GPS 定位功能，整个系统包含室内部分和室外部分，室内部分装配了室内定位系统 Active bat。当 PDA 在室外的时候，位置显示应用使用 GPS 向用户显示位置信息，当 PDA 进入室内的时候，位置显示应用自动使用 Active bat 获取位置信息(室内状态下，GPS 失效)。从而验证了 CWDB 能够在动态环境中实现动态绑定，并保持了应用程序的稳定性和简单化，同时也验证了发现服务的高性能。

小结 普适计算环境是一个动态的环境，如何向应用程序屏蔽环境的动态性?这个问题传统的中间件没有解决。文中我们通过在中间件的基础上引入动态绑定机制来完成这个功能，并开发了相应的原型 CWDB，通过实验验证了 CWDB 对动态绑定的支持，及其高性能。

参 考 文 献

1 ISO/IEC and ITU. Reference Model of ODP Part 1 Overview. May 1995

2 Otway D. The ANSA Binding Model. ANSA Phase 3 Project Report APM. 1392. 01, Jan. 1995

3 Universal Plug, Play Forum. Universal Plug and Play Web Site. 2000. http://www. upnp. org/

4 Edwards W K. Core JINI. Upper Saddle River, NJ: Prentice Hall, 1999

5 Guttman E, Perkins C, Veizades J, Day M. Service Location Protocol. Version 2. RFC 2608, June 1999

6 Nigel D, Mitchell K, Cheverst K, Blair G. Developing a context-sensitive tour guide. In the 1st Workshop on Human Computer Interaction for Mobile Devices, Glasgow, Scotland. May, 1998

7 Gray P, Salber D. Modeling and Using Sensed Information in the Design of Interactive Applications. 8th IFIP Intl. Conf. EHCI 2001, Toronto, Canada, 2001

8 Object Management Group. CORBAservices: Common Object Services Specification. Object Management Group 1999