

分布式协作模型及应用研究^{*})

张全海 施鹏飞

(上海交通大学图像处理与模式识别研究所 上海200030)

Study on the Distributed Collaborative Model and Application

ZHANG Quan-Hai SHI Peng-Fei

(Institute of Image Processing and Pattern Recognition, Shanghai 200030)

Abstract With the development of the Web technology, the application environment has acquired many new characters such as dynamic, openness, distribution and information uncertainty. The processing mode of application systems is more complicated than ever. For example, it requires application systems to have more community processing ability, interactive ability, distributed processing ability and collaborative ability. Accordingly the research and development of the computer application system transited from client/server information processing system into distributed collaborative processing system based on Web. Especially in the environment where the information and resources are highly distributed, the accomplishment of complicated tasks is dependent more on the resources coordination, information sharing and coordinator collaboration. The collaboration is one aspect of the group behavior and its goal is to provide a optimal method to utilize the resource through the information interaction and to solve the task which couldn't be accomplished by each coordinator alone and get the more total benefits than the sum of each benefit. The collaboration problem is the important one for distributed tasks processing. This paper surveys the research and application status of distributed collaborative models and several representative architectures of distributed collaborative processing are proposed. However, the existing problems and the future researching direction are presented.

Keywords Multi-agents, Middleware, Distributed collaborative model, Distributed artificial intelligence, Web computing

1. 引言

当前网络已经成为人们现实生活中不可或缺的一部分,基于网络的应用也迅速增加和日趋复杂,如电子商务,计算机辅助制造,智能交通系统,计算机辅助医疗诊断等,这些任务的解决需要大量的数据、知识、控制和结点的处理能力,而这些知识分布在网络系统中,没有一个节点具有足够的知识能力和能力来解决全局复杂问题,而整个问题的解决取决于多个节点的信息交互,协调(coordination)和协作(collaboration)。随着计算机网络技术,数据通信技术,多媒体技术,分布式处理系统的迅速发展,人们的工作方式更多地具有群体性、交互性、分布性和协作性,相应的计算机应用系统的开发已经从客户服务器的信息处理系统向基于网络的分布式协作系统发展,在信息和资源分布化的网络环境中,复杂任务的完成更多地依靠参与者对资源、信息的相互协调以及参与者之间行为的相互协作来完成。

20世纪90年代中期以后,计算机网络技术,多媒体技术,数据通信技术,特别是分布式处理技术的发展使计算机网络在协作过程中发挥着越来越突出的作用,网络的应用越来越多,其复杂度在增加,研究者发现网络应用具有协作性、交互性,多样性、可扩展性及相互依赖性,协作范围已经由系统内部节点间的协作扩展到系统间的协作,但是应用系统间的异构性、动态性(节点的不断更替,新节点的产生加入,旧节点的

改造及消亡)、稳定性等使得旧有的协作模式出现很多的弊端,如协作行为模式不能动态变更,协作成员的结构相对固定,而且协作者信息不能动态变化,协作者需要事先指定而不能实时寻找等,已经适应不了网络应用中动态性和分布处理的要求,需要一种新的分布式协作模式。为了适应计算方式和应用特点的变化,出现了一种新的网络计算三层模式以代替原有的C/S(客户/服务器)计算模式,这种模式的一个突出的特点就是增加了一层称作是中间件(middleware)的处理层,以提供网络分布式处理的相关服务,如资源的寻找,网络结构的屏蔽及系统异构性的消除等。中间件是一种抽象的系统软件,应用在基本系统软件如操作系统,网络协议和硬件驱动程序之上,用以屏蔽用户的界面差异、网络差异和系统差异,负责协作者的定位和资源的查找,以及底层通信,异构系统的互连形成一个分布式应用环境。而在应用中,当单个应用无法完成某个任务处理时,就需要其它应用的帮助,产生协作,协作涉及大量的问题如协作者之间如何实现信息共享和通信,如何实现不同系统和平台之间的知识理解,协作任务的分派和集成等,协作的成功在于所有协作者对知识的享用和理解以及有限资源的合理利用。协作是一个群体的行为表现,其目的是通过信息交互,在多个协作者之间提供一个资源的优化处理方式,使个体不能解决的任务得以完成,并产生大于个体单独处理总和的效益。

三层计算模式的推广加速了分布式协作模型的研究和发

^{*})本项目得到国家自然科学基金(NO. 69931010)资助,张全海 博士生,主要研究方向为图像分析,知识处理,人工智能和网络协作。施鹏飞教授,博士生导师,研究方向,模式识别、图像处理、知识发现和数据挖掘。

展,目的是减少系统间协作结构的复杂性,增强协作关系的一致性 & 可控制性,不同模型的区别在于中间件软件的选择及使用,如何表示出协作任务间及不同协作者之间错综复杂的关系如同步、通信、并发控制等。本文在查阅国内外有关文献的基础上,对分布式协作模型近年来的研究发展和应用进行了综述,列举几个典型的分布式协作模型:工作流(workflow)协作模型,多用户柔性(multi-users flexible)协作模型,多智能体协作模型(Multi agents, MAS)和协作中间件模型(CLF协作模型及 ORBIT 模型),并对当前协作模型的发展趋势和存在的问题进行了总结。

2. 协作模型基础

协作被认为是活动实体相互合作以完成一个共同的目标,基本理解是首先活动实体愿意协作,其次活动实体知道与哪些实体协作,以及活动实体知道该如何协作,这个对协作的初始理解模型如图1所示。

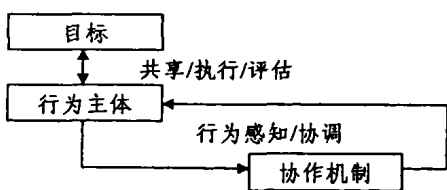


图1 协作认知模型

1986年,Carriero 和 Gelernter 在协作语言 Linda 的基础上提出了协作模式的概念^[1],认为协作模式是一个将杂乱无序的行为活动进行有机规划,使之成为一个结合紧密的行为序列的粘合剂(glue),即一个协作模型定义了行为活动是怎

样进行交互及怎样被控制的,交互包括行为处理过程的感知、建立和消除,信息的通信及活动执行状态的同步。对协作模型描述实际上成为协作模型开发的设计标准,即不同的协作模式采用不同的机制和结构去表示或实现这些独立的行为实体(协作体)之间的交互。计算机协作的本质就是利用计算机和网络建立一个协同工作的环境,在此环境中各种协作实体可以互相合作,共同处理任务。1996年 Ciancarni 提出了三元组协作数学模型,认为协作模型 CM 由三元组构成^[2]:协作者 C(coordinables),协作媒介 M(coordination media)和协作规则 L(coordination laws)构成,表示为:CM=(C,M,L)。协作者指模型中彼此间交互行为由模型控制,完成共同目标的行为实体,可以是用户、事件、程序等等,是协作模型的基本元素;协作媒介用于提供场所将协作者组合到一起,为其信息传递和信息交互提供方式,如通道、共享变量、共享空间、黑板结构(blackboard)等;协作规则定义了协作者间交互,以及协作媒介对交互事件的反映行为,对实体间的协作行为进行控制,它们可以用协作语言来进行描述,即用交互原语(primitives)和语义(semantics)来表示和交换数据结构及规则。Malone 和 Crowston 结合计算机科学、管理科学和组织理论提出了协作理论模型^[3]以揭示不同系统间的协作规则,如图2所示,认为协作规则管理的是行为间的依赖关系而不是行为本身,因此协作行为是独立于协作者自身行为之外的,协作模型的作用是建立这些相互关系的时间和空间逻辑,如进行目标识别,行为和目标匹配,行为的规划,及选择协作者处理目标行为,资源分配等。将目标和行为联系在一起是因为目标只有在行为中体现,只有在行为环境中才有意义,而不依赖于其它的概念。

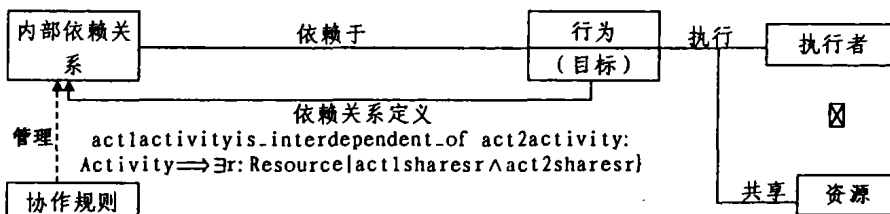


图2 协作理论模型

3. 分布式协作模型

3.1 工作流协作模型

工作流协作模型^[4]是分布式协作的一个大的研究方向,工作流是一种计算机辅助任务完成的自动化行为,辅助不同处理过程的实体完成多个任务间的协作执行,即辅助建立工作计划,使得在一个工作流程中定义的多个任务以合理的秩序协作执行,以使最终目标得以完成。工作流是一个复杂的行为,涉及多个级别上的多个事务的处理,如在系统级别上,处理过程的定义和执行;事物处理级别上处理过程的建模等。工作流协作模型实质上是从管理的角度上根据现实中事物处理的过程,增加一个工作流中间件,提供工作流程的建立,控制数据的传输和交换,工作流程执行的监控和变更等服务,使用的主要概念是角色(role),实施者(actor),交互(interaction)和工具(artifact),角色定义了任务的目标和达到的结果,实施者是具体任务的执行对象,工具是实施者进行交互时采用的方式。工作流协作模式研究主要是参照了 WfMC 组织发布的

工作流协作参考模型,定义了工作流建模语言强化不同开发商提供的系统间的协作,设计了一组工具以支持工作流建立的管理服务包括工作流程定义、执行、管理和监控。Zhang 则在此模型中增加了一个分布式处理总线(ORB)和工作任务管理者(worklist manager)^[5],以增加其网络应用,可以通过 HTTP 获取用户任务请求,可以在工作任务表空间中对工作任务进行修改,如增加新的任务,消除已执行过的任务及任务的悬挂等。

3.2 多用户柔性(multi-users flexible)协作模型

柔性的动态支持网络分布式协作模型 COCA(collaborative object coordination architecture)^[6],主要解决协作过程中的动态性和可扩展性。由于没有统一的协议和协作策略适用于各种系统,因此不同的系统往往采用适合自己的方式和策略进行协作和计算处理,当多个异构的系统连接在一起时,就会出现不兼容的情况,使工作无法进行,解决方法则为协作者通过增加一个推理功能模块,内部数据库和策略定义功能模块来实现协作的可扩展性。推理模块实现信息通信的监控,协

作规则的激活,根据策略实现执行协作;内部数据库为策略的制定和行为的监控提供状态信息和规则信息;策略定义模块用于对协作策略的定义。该模型提供了一个双总线结构(协作总线和会晤总线,该模型认为协作是通过一个个会议进行的)实现监测和控制,按照协作策略控制协作信息流的流向,前者用于连接协作者,后者用于连接各种协作工具和应用系统。同时定义了一种协作语言用于在协作总线和虚拟组件之间实现信息的传递和协作行为的执行。虚拟组件再加上双总线结构则构成了协作中间件,这样,在协作总线上可以对协作者进行管理,加入和离开都很自由,每个协作者可以采用不同的系统,因为在协作总线上实现了信息的均匀化,并且由于虚拟组件在每个协作者方,可以根据变化方便地改变自己的协作策略。Begole 则在1999年提出了另一种通用模型实现柔性的透明协作^[7]。该模型采用了一种复制体系结构使得协作者和应用系统透明地安置在网络中,运行时选择的单个用户的组件则被一个通过协作感知的多用户使用组件代替,并给出了使用该组件的用户的信息。当某个用户使用某个组件时候,通过广播的方式将信息传递给其它协作者,这样每个协作者都有一个其它协作者的应用备份,以达到柔性和透明的协作。T. Mansfield 利用了一个成为本地化服务(Locale)的机制及 ORBIT 总线^[8],存储信息系统,提供协作者与资源利用的一个局部视图,为协作者的交互提供一个更为方便的空间。本地化可以认为是一个场所,协作者根据自己的意图在此场所中利用资源和此空间进行协作和信息交互,不用考虑其它协作者的位置,就像在本地工作一样,而这些辅助工作都由本地化提供的机制完成,包括工具的提供,协作者的定位,协议的转化和信息格式的标准化等,实现协作者的分布式工作。在网络环境中,协作者和网络的异构性要求协作信息能够智能地传送以至能匹配协作者的能力和所拥有的资源,但是现有的协作模型无法满足分布式协作处理的需要,Pravin Bhandarker 提出了可适应的分布式协作模型^[9],在此模型下,分布式客户可以按照自己的意图加入,离开协作环境,实现信息的有效共享,并且协作者的数目、位置、能力和协作意图都是动态变化的。该模型基于信息协作模型,使用了“推(push)”机制进行分布式信息管理,另外应用面向对象设计模式将信息协作和知识共享都进行了封装,应用 JAVA 和 CORBA 技术来实现分布式处理。

3.3 多智能体(Multi agents systems, MAS)协作模型

计算机应用系统之间协作的复杂性和组织机构的异构性,使软件协作系统的设计越来越向智能化方向发展,智能行为取决于所获取的信息或者知识的数量及相关度。但是如何利用所获取的知识进行推理,预测及判断知识的相关或无关性,则逐渐成为分布式协作的主要问题。Durfee 认为实现分布式协作最重要一点就是协作者必须拥有足够的知识处理和推理能力,使之能有效地利用协作环境中动态变化的协作信息作出符合全局观点的决策以完成协作目标^[10]。特别是在动态分布的网络环境中,事物之间的关系具有的模糊性导致了所获得信息的不确定性,这也是在进行分布式协作时候所直接面临的问题,在此情况下协作更应该运用知识处理的机制来进行推理,思维及求解问题,增加协作者的“智能性”。

多智能体系统(Multi agents system, MAS)是由具有自治能力的 Agent 通过协作完成某些任务或达到某些目标的计算系统,是分布式人工智能(Distributed Artificial Intelligence, DAI)一个重要的分支。MAS 系统本质上是一个拥有

有限资源的知识获取,处理和采取相应行动对周围环境的作用做出反应,并能改变环境的智能体社会^[11]。多智能体系统由多个具有独立的问题解决能力和社会交互能力的异构计算实体(智能体)构成的分布式处理软件系统,以解决超越单个智能体能力的复杂任务。在此系统中,为了获得共同的目标,智能体之间必须进行协作和通信。多智能体协作模型研究的是一组自治智能体(Agent)之间智能行为协作,即如何协调他们的知识、目的、技巧和规划,从而完成问题的协作求解。MAS 强调分布式问题处理时控制上的分散性和处理方法上的柔韧性,研究重点则是各智能体之间目标,规划,知识和行为的协作以使他们能完成各自的任务或者协作完成复杂任务,而任务的完成取决于智能体知识处理和推理能力的好坏以及协作机制的有效性,协作(一种特殊的协调方式)是 MAS 的核心研究问题,没有协作 MAS 将会退化为纯粹的智能体的组合,在共同工作时也达不到预期的目的,甚至会严重影响系统的性能。比较普遍的 MAS 协作模型是组织协作模型、基于合同网的市场协作模型、多方规划协作模型和基于社会规则的协作模型。

3.3.1 基于合同网的市场协作模型 从现实的市场结构中得到启发,每个智能体可以承担两种功能,管理者和协作者,即首先智能体被分到一个任务,他成为协作者,如果他认为该任务利用本地资源和自身的能力不能完成,则他负责将此任务分解并尽力寻找协作者完成这些子任务,此时他成为管理者,这些子问题的解决基于合同网协议(contract net protocol),即将现实生活中的合同过程引入到分布式处理中,通过在智能体间建立和执行特定目标和任务的合同来保证任务的完成,从而从合同双方的关系来建立交互和协作方式。比较典型的系统如 Schodhom 将该模型应用于分布式车辆派货领域,R. Davis 采用的招标-投标-中标的任务分配等,Parunak 将该模型应用于制造控制,以及 Huhns 的利用自投标方式的动态任务分配系统^[12]。该模型的缺点是不能检测和解决有关的恶性冲突。

3.3.2 多方规划协作模式 多智能体的规划模型主要用于为多个智能体制定协调一致的问题求解规划。智能体具有自己的求解目标,并且每个智能体都考虑其它智能体的行动和约束,并进行独立的规划,成为部分规划。Duree 提出的部分全局规划 PGP(Partial Global Planning)模型能实现各智能体的动态协作。多方规划协作模式将多个智能体间的协作问题简化为一种规划问题,即强调避免不一致和冲突情形的出现,为了形成协作规划,智能体在行动前应先决定其计划和行动方案,交互策略,一旦制定完毕后不允许改变,但在制定过程中通过信息通信来消除方案中的冲突问题,这个要求参与协作的智能体共享和处理大量的信息,因此通信和计算工作量非常大,如 Lesser 采用该模型制定的 FA/C 协议^[13]。

3.3.3 基于社会规则的协作模型 基于社会规则的协作模型强调协作的规则性,即把不熟悉的事务处理方式,通过知识的共享和智能体的学习,改变为其熟悉的处理方式。这种模式要求更高的分析和推理能力,而这都依赖于智能体的知识获取。智能体的行为由预先存储的处理模式所决定,由该模式的例程和预编译的规则来完成协作处理,如果某个模式事先未存储,则通过自学习,智能体确认该模式为新模式,则将该模式存储起来。

3.3.4 组织协作模型 组织协作模型是最简单和直接的协作模型,组织机构间有组织的信息流动,可以弥补个体知

识的有限;精确划分的组织机构和明确的个体任务,可以弥补个体处理信息并运用信息作出有限决策能力的缺陷。组织协作模型的基本体系是 Master/Slave 或者是 Client/Server 结构,模型中存在一个或者多个 Master 对资源和任务在 Slaves 中进行分配,另外 Master 也负责收集参与协作的智能体的信息,建立协作方案,指派任务,监督任务的执行,这样做可以保证任务处理的一致性,通常这种模式都是利用一个共享空间来实现这种交互和协作,如黑板结构(blackboard),组织模型中比较典型的系统是 Werkman 的 DFI 系统^[14]和 Kearney 的 SMAK 系统^[15],该模型的缺点是不容易实现和维护,而且中心 Master 的信息通信量巨大,系统健壮性差。

3.4 协作中间件模型

D. Arregui 提出的异构组件协作的方法即使用协作语言设施的方法 CLF(coordination language facility)^[16],提供了一套基本工具实现分布式环境中协作者和资源管理者处理复杂任务及资源的协作处理机制。CLF 协作组件能执行由协作语言描述的与其它组件的交互动作,包括查询协作组件的 URL,通过 URL 建立协作组件的连接,与该组件的信息通信,以制定协作计划和原子操作。在 CLF 方法中强化对象封装技术,组件所拥有的资源不能直接获取而必须通过与该组件的交互来获取。CLF 方法主要应用于分布式环境中不同平台上组件间的协作,通过协作者和协作脚本语言实现异构组件调用的一致性界面,并发地执行多个通信协议及实现通信协议的相互转换,保证系统和描述语言的独立性。资源管理器则实现资源的管理如加入和消除资源,协作组件则是这些资源的操纵者,当使用某个资源时,需要协作组件来激活这些资源。其它的协作中间件模型如孙艳春等结合 CORBA(一种分布式中间件技术)的协作模型, M. Y. Sung 等提出的 CoMed^[17],史殿习等提出的利用 CORBA 的分布式协作模型, N. Bricon-Souf 提出的动态工作流协作模型^[18], A. Sleth 结合 CORBA 和工作流技术的协作方法,这类协作模型从技术上解决了分布式处理中的异构性、动态性和可扩展性。

4. 分布式协作应用领域

网络应用的复杂性和分布性,以及网络计算方式的兴起,使得分布式协作技术的应用具有非常广阔的应用范围,特别是面向智能体的应用不断增加,智能体能力的不断增强,能模拟越来越多的人的思想和行为,使得分布式协作系统,智能体系统代表了一种新的方式和途径,用于描述和实现复杂、庞大的系统,涉及工业控制、制造业,金融业、医学领域、网络信息提取及处理和交通控制行业等多个领域。

4.1 金融管理,商业活动中的应用

分布式协作技术特别是 MAS 近年来在商务管理、金融市场、电子商务和项目管理等商业行为活动中有着比较突出的应用,借助 MAS 技术,特别是各个智能体之间的协作,使交易过程有着更高的自动化程度,用智能体协作来完成交易过程中各个环节,具有更高的市场反应能力,并且可以快速得到最新的信息,拓宽业务范围,提高效率。周志中将 MAS 应用于系统评估和管理金融市场中的信用风险,能够处理复杂模型的计算,这样传统的分析模型中许多假设得以放松使之更加符合,通过建立基于多 Agent 的决策支持系统,其最大的好处是实现了人机协同工作,它能够把结构化信息处理和非结构化的信息处理的优点结合起来,最大程度地从数据中挖掘有用的信息,王坚等通过将 CSCW 和 Agent 技术结合,

以在系统内部和外部都建立起一种协作的环境,提出了并行项目协调和管理系统原型, T. P. Liang 提出了基于 Agent 的电子市场中介系统模型^[19], J. Teich 将 MAS 技术应用于网上商业谈判支持系统, Takayuki 提出了支持用户参与、监控和拍卖多个交易的在线拍卖系统 Biddingpot, R. Guttman 则将 MAS 协作技术应用到零售电子商务和股票市场^[20],随着网络支持技术和协作技术的研究逐渐深入,智能化的网络协作将渗入到商业活动的各个角落。

4.2 工业控制中的应用

工业控制具有复杂性和大型性等特点,传统的集中式控制方式难以满足控制系统实时性、经济性和可靠性等要求,这就要求引入分布式协作智能控制系统,提高系统的灵活性和自适应性,特别是把一个复杂的问题分解成相对独立的小问题交给智能体来完成,维护和调试,这样使系统的可靠性增加,提高问题的解决效率,这方面的系统有 Musliner 提出的协调智能实时控制的框架 CIRCA, Jennings 提出的基于智能体的过程控制模型 ARCHON,用于电力调度管理和离子加速器的控制。在工业过程中也利用分布式协作系统进行故障诊断与监控,通过设计诊断智能体,将其分散在控制现场的不同节点上,利用智能体之间的通讯,协作得出整个系统的工作状况和故障诊断,这方面成功的系统如 Iyer 利用移动智能体(mobile agent)设计的可适应性故障控制系统(fault control system)^[21], Alonso 提出的基于知识的专家系统 AEROLID 用于过程监控和故障诊断^[22]。在工业控制中,分布式协作系统应用最广的是智能机器人的研制,多机器人的协作是智能机器人研究的一个重要分支,研究的是在单个机器人由于缺乏全局信息、技能而不能单独完成大的作业任务的情况下,如何相互合作完成作业,如智能机器人足球比赛,机器人协作装配系统 MRCAS^[23]。

4.3 制造业应用

信息技术的发展使传统的制造业面临了许多新的问题,如市场变化快使得无法预知它的变化方向,产品生产周期大为缩短,生产方式也转变为面向客户要求为主,而将 DAI 技术引入到制造业中将实现制造业的扩展和重构,产生新的应用,将 DAI 中的知识推理,智能协作及智能体方法应用到制造业,近年来被广为重视,学者们也对此进行了多种角度的研究,如在制造业中,已经存在许多成熟的单项技术,形成许多“自动化孤岛”,通过利用 DAI 中 MAS 结构既可以保持这些系统的相对独立,而且又可以借助于智能体间的协作,冲突消解等策略分析它们之间的复杂交互关系,达到信息的集成,产生新的控制方式,如 Bussmann 提出了面向智能体的制造系统体系结构,进行制造系统中的物流控制^[24]; Butler 提出了用分布式动态调度的 ADDYMS 体系结构^[25],李斌等提出的用 CORBA 和智能体结合的分布式网络化制造模式。

4.4 医学应用

医学诊断是一个复杂的过程,主要包含:信息的智能收集,即收集大量相关的静态和动态数据;诊断方案设计;最佳诊断方案的选择和诊断方案的实施。因此为了实现一个良好的医学诊断系统需要解决以下几个问题:能及时有效地发现并获取信息;具有标准的数据或信息表示方法,使之在互异的系统中能相互理解;系统间能有多种方式进行交互和通信;而在实际执行中需要其它实体的辅助和应用服务支持,更需要有大量的经验知识和历史数据,而这些实体和数据知识库处于不同的地点和系统中,因此需要建立一个基于网络处理的

分布式处理系统来获取和应用这些资源和数据。Enrigue 利用宽带技术和多媒体协作技术对远程医疗和医学影像协作诊断系统进行了研究,Roberto 利用虚拟协作者的概念,建立了 Internet 上的远程协作诊断系统^[26],Mee 等建立了实时协作医疗系统 CoMed^[17],David 等建立的医学协作系统 TeleMed, Marios 等设计的分布式医学图像处理系统 DIPE, G. R. Rao 设计的支持医学协作决策的超文本组决策支持系统, Huang 等人提出的一个利用智能体的分布式诊疗系统的原型系统^[27]。医学信息处理需要大量的信息和工具,因此协作系统在医学治疗、诊断领域具有巨大的应用空间。

4.5 交通控制,远程教育和电信管理的应用

分布式协作在远程教育,电信行业,交通控制业上也具有广泛的应用价值,俞峥等通过将 Agent 引入到城市交通控制系统中,统一了控制系统中人和其他控制单元的行为,以及实现了统一描述,加强了之间的协作,便于对复杂路面进行优化控制,Wangermann 针对空中飞行器的大量出现,提出了将各飞行器和交通控制单元设计成智能体,通过利用 MAS 来提高整体系统的性能,提高对飞行器行为的有效控制^[28],这方面的应用还有铁路交通控制 RTC,海洋交通控制 MTC。在远程教育上,应用比较多的是协作式学习系统,如刘贵全提出的基于多智能体的协作学习系统 CML^[29],Capuano 提出的利用分布式人工智能技术和 MAS 结构的远程指导(tutoring)系统,在电信行业中,主要是引入移动代理(mobile agent)和多智能体结构来进行网络管理。

结束语 协作技术是计算机科学中为解决大规模的分布和并行处理问题而出现的,协作技术强调的是计算和协调的正交性,通过抽象出协作模型和协作语言来组织协作者的通信和交互。对于分布式协作来说,关键是需要有一个中间层,来屏蔽掉参与协作的各个实体之间的软件和硬件差异,使从外部看整个协作系统不仅是体系结构的问题,另外还是如何适当地组织各个协作实体的行为,以发挥其个体的能力,得到全局最佳的效益,这是一个规划的内容,具体包括协作机制和策略的制定,各个协作实体信息交互以及协作知识的管理和组织。随着网络技术和人工智能技术的发展,使得基于 Web 的应用正快速增长,分布式对象处理技术将会向复杂化、智能化、协作化和标准化方向发展。分布式计算技术和传统的面向对象技术的最大不同在于,各种应用组件可以分布在网络的任何位置,而且只关心组件的应用接口,并不关心其内部的具体实现,而且更加强调分布式应用间的适应性,网络获得性,互操作性和应用协作性。将多智能体技术和分布式对象计算技术相互融合,以形成一个以统一的方式集成不同的、异构的组件,并且实现组件的动态重构和动态协作。统一的良好分布式组件应用界面的设计,以及分布式处理机制和智能体技术的结合,加强知识交互协议和语言等的完善和制定,特别是对于协作的实时性和协作质量如 QoS 参数的研究,实现分布式计算中组件的“即插即用”将是未来分布式技术发展的方向,也是以后研究工作的重点。

参考文献

- 1 Carriero N, Gelernter D. Linda in Context. *Communications of the ACM*, 1989, 32(4): 444~458
- 2 Ciancarini P, Omicini A, Zambonelli F. Coordination Technologies for Internet Agents. *Nordic Journal of Computing*, 1999, 6(3): 215~240
- 3 Malone T W, Crowston K. The interdisciplinary study of coordina-

- tion. *ACM Computing Surveys*, 1994, 26(1): 87~119
- 4 Hawryszkiewycza I T, de Moor A. A workflow approach to designing cooperative systems. In: *Proc. of the third Intl. Conf. on the Design of Cooperative Systems (COOP'98)*, Cannes, France, May, 1998. 165~174
- 5 Kaya A. Workflow interoperability: The WFCM Reference Model and an Implementation. [Master thesis]. Software Systems Department, Technical University Hamburg-Hamburg, Germany, 2001
- 6 Li Du. COCA: A Framework for Modeling and Supporting Flexible and Adaptable Synchronous Collaborations. [PHD. Dissertation]. University of California, Los Angeles, 2000
- 7 Begole J, Rosson M B, et al. Flexible Collaboration Transparency? supporting worker independence in replicated application-sharing systems. *ACM Transactions on Computer-Human Interaction*, 1999, 6(2): 95~132
- 8 Mansfield T, Kaplan S, et al. Towards Locales Supporting Collaboration with Orbit. *Journal of Information and Software Technology*, Elsevier, 1998, 41(6): 367~382
- 9 Bhandarkar P. An Adaptive Framework for Collaboration in Heterogeneous Environments. [Masters Degree Thesis]. Rutgers University, May 1999
- 10 Durfee E H, Lesser V R, Corkill D. Trends in Cooperative Distributed Problem Solving. *IEEE Tran. On Knowledge and Data Engineering*, 1989, 1(1): 63~83
- 11 胡代平, 王浣尘. 基于多 Agent 的模型系统研究. *系统工程理论方法应用*, 2001, 10(2): 89~92
- 12 Huhns M N, Singh M P. Distributed Artificial Intelligence for Information Systems. CKBS-94 Tutorial, June 15, University of Keele, UK
- 13 Lesser V, Corkill D. Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, 1981, 11(1): 81~96
- 14 Werkman K. Knowledge-based model of negotiation using shareable perspectives. In: *Proc. of the 10th Intl. Workshop on DAI*, Texas, 1990
- 15 Kearney P, Sehmi A, Smith R. Emergent behaviour in a multi-agent economics simulation. In: Cohn A G, ed. *Proc. of the 11th European Conf. on Artificial Intelligence*, London, John Wiley, 1994
- 16 Arregui D, Pacull F, et al. Heterogeneous Component Coordination: the CLF approach. In: *Proc. of EDOC'00*, Tokyo, Japan, Sept. 2000
- 17 Mee Y S, Moon S K, et al. CoMed: a real time collaborative medicine system. *International Journal of Medical Informatics*, 2000, 57: 117~126
- 18 Souf N B, et al. Dynamic workflow model for complex activity in intensive care unit. *International Journal of Medical Informatics*, 1999, 53: 143~150
- 19 Liang T P, Huang J S. A framework for applying intelligent agents to support electronic trading. *Decision Support System*, 2000, 28: 305~317
- 20 Guttman R, Maes P. Cooperative vs. Competitive Multi-agent Negotiations in Retail Electronic Commerce. In: *Proc. of the Second Intl. Workshop on Cooperative Information Agents (CIA'98)*, Paris, France, July, 1998
- 21 Iyer R K, Kalbarciyk, et al. Adaptive fault tolerance using reliable, mobile agents. In: *Proc. of the IEEE Symposium on Reliable Distributed Systems*, Durham, NC, USA, 1997. 61~62
- 22 Gonzalez C A, Acosta G, Mira J. Knowledge based process control supervision and diagnosis: the AEROLID approach. *Expert System with Application*, 1998, 14: 371~383
- 23 Wu Z W, et al. A Multi-robot coordination system. In: *Proc. of the fourth IFAC Symposium on Robot Control*, Sep. Capri, Italy,

- 24 Bussmann S. Agent-Oriented Programming of Manufacturing Control Tasks. In: Proc. of the 3rd Intl. Conf. on Multi-Agent Systems (ICMAS'98), Paris, France, 1998. 57~63
- 25 Butler J, Ohtsubo H. ADDYMS: architecture for distributed dynamic manufacturing scheduling. Artificial Intelligence Application in Manufacturing, MIT Press, 1992. 199~214
- 26 Sacile R. Telemedicine Systems for Collaborative Diagnosis over the Internet: Towards virtual "collaborators". In: Proc. of the Acadma/Industry working Conf. on Research Challenges

- 27 Huang J, Jennings N R, Fox J. An agent architecture for distributed medical care. In: M. J. Wooldridge, N. R. Jennings, eds. Intelligent Agents lecture notes in Artificial Intelligence, Vol. 890, 1995. 219~232
- 28 Wangermann J P, Stengel R F. Principal negotiation between intelligent agents: a model for air traffic management. Artificial Intelligence in Engineering, 1998, 12: 177~187
- 29 刘贵全, 陈小平. 一个基于 Agent 的协作式学习系统. 中国科学技术大学学报, 2000, 30(1): 113~118

(上接第7页)

- 7) 调用 integrateResult() 函数集成结果;
- 8) 将与集成结果相关的进度条标志位置为 FALSE;
- 9) 置 Finish [i] 为 TRUE, 等待界面线程 i 结束标记置为 TRUE;
- 10) 判断 Table 中是否还有没有读完的数据块, 如果有, 转移到2;

如果没有, Again [i] 置为 FALSE, Finish [i] 置标志位;

算法6 界面线程 i 的(描述)算法

- 1) 判断 Again [i] 的初值是否为 TRUE, 如果不为 TRUE, 则结束。
- 2) Finish [i] 置为 FALSE, Continue [i][1] 置为 TRUE, continue [i][3] 置为 TRUE。
- 3) 用进度条显示数据发送的过程。
- 4) 如果 Continue [i][1] 为 FALSE, 转移到6)。
- 5) 与 Continue [i][1] 对应的进度条开始反馈运行信息, 转移到4)。
- 6) 与 Continue [i][1] 对应的进度条显示完毕。
- 7) 用进度条显示接收数据的过程。
- 8) 如果 Continue [i][3] 不为 TRUE, 则转移到10)。
- 9) 与 Continue [i][3] 对应的进度条开始反馈信息, 转移到8)。
- 10) 与 Continue [i][3] 对应的进度条显示结束. 与本线程对应的线程结束标记置 TURE, 等待 Finish [i] 标记置 TRU-E, 转到1)。

4. 可视化分布式并行矩阵乘法的实现

本文的可视化分布式并行计算模型已在一个由9台异构 PC 机组成的局域网上实现, 操作系统使用 Windows 2000 Server, CORBA 采用 IONA 公司的 ORBacus 4. 1. 3^[6-8], 编程语言使用 C++。

实验采用的机器如下: 5台128M 内存 PIII800机器, 2台256M 内存 PIII500机器, 2台128M 内存 AMD 300机器. 其中一台 PIII800机器做为客户机, 其它机器全部用作节点机. 图1是在三个节点的情况下, 维数为512的两个方阵相乘时的可视化程序的运行情况。

从图1中可以看到矩阵的整个运算过程和各节点机的运行状态. 如节点2、节点3正处于计算状态; 节点1已经完成了一个计算任务, 并已将计算结果传给客户机, 此时客户机正在集

成节点1回传的结果, 同时节点1正准备接收新的计算任务. 由此可以看出, 可视化技术可以使用户了解计算节点在某一时刻的状态。

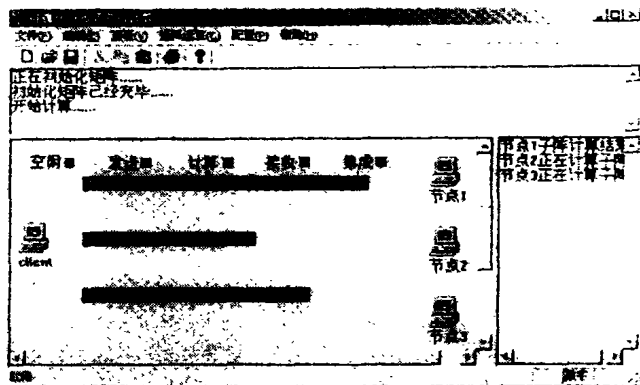


图1

结束语 在分布式并行计算中引入可视化的方法, 可以将整个计算的过程直观地呈现给用户, 使用户了解计算节点在某一时刻的状态, 便于用户对其所设计的并行算法进行分析、调整和评价, 并在此基础上设计出效率更高的算法. 同时可视化技术还可以使用户方便地进行并行程序的调试, 并在一定程度上降低并行算法的设计和调试难度。

参 考 文 献

- 1 Aleksy M, Korhous A. A CORBA-Based Object Group Service and a Join Service Providing a Transparent Solution for Parallel Programming. 2000 IEEE
- 2 陈国良. 并行计算. 高等教育出版社
- 3 (美)Hwang K 著, 王鼎兴等译. 高等计算机系统结构. 清华大学出版社
- 4 肖依, 等. 面向21世纪的广域高性能元计算技术. 计算机科学, 1999, 26(10)
- 5 王育峰, 杨寿保. 网络计算系统研究及发展方向. 计算机科学, 2002, 29(6)
- 6 CORBA/C++ Programming with ORBacus. http://www.ooc.com
- 7 Jthreads/C++ (Java-like Threads for C++). http://www.ooc.com
- 8 Henning M, Vinoski S. 基于C++的CORBA高级编程. 清华大学出版社, 2000. 7