

基于 HLA 和移动代理的负载平衡控制技术探讨

马善达, 万晓冬, 李云芳, 李 辉

(南京航空航天大学 自动化学院, 江苏 南京 210016)

摘 要:随着仿真技术的发展、仿真系统规模的不断扩大,基于 HLA 的仿真系统的负载平衡面临严峻的考验,如何保证仿真试验高效平稳的运行成为当前急需解决的一个问题。移动代理(Agent)已经被证明在平衡负载方面卓有成效。文中将移动 Agent 引入基于 HLA 的仿真系统中,主要从动态迁移方面,探讨较合理的仿真控制技术,控制负载平衡的实现过程,最终达到保证仿真运行速率的目的。根据每个仿真试验的不同情况,设置合理的迁移策略,才能提高负载平衡的效果。

关键词:高层体系结构;负载平衡;移动代理;仿真控制技术;运行速率

中图分类号:TP391.9

文献标识码:A

文章编号:1673-629X(2012)06-0163-04

Load Balancing Control Technology Based on Mobile Agent and HLA

MA Shan-da, WAN Xiao-dong, LI Yun-fang, LI Hui

(College of Automation Engineering, NUAA, Nanjing 210016, China)

Abstract: With the development of simulation technology, simulation systems have been expanding, load balancing of HLA-based simulation system is facing a severe test, it has become a urgent solution problem that how to ensure smooth and efficient operation of simulation. Mobile Agent (Agent) has been shown effective in load balancing. In this paper, combine mobile Agent with HLA based simulation system, mainly from dynamic aspects of migration, investigate a reasonable simulation control technology to control the load balancing, and ultimately ensure that the simulation running rate is under control. According to the different circumstances of each simulation, set reasonable migration rules in order to improve load balancing effect.

Key words: high level architecture; load balancing; mobile Agent; simulation control technology; running rate

0 引 言

高层体系结构(HLA)是美国国防部1995年提出的新一代仿真技术框架,它是分布式仿真DIS和聚合级仿真ALSP发展的高级阶段。HLA在解决异构、分布、协同的仿真模型和仿真系统的互操作及可重用性方面取得了重大进展。

基于HLA的仿真系统由于缺乏对仿真资源的组织、调度使得系统缺乏弹性和鲁棒性。大规模的分布式仿真中,通常将联邦成员分布在多个节点运行,以保证每个联邦成员获得更多的运算资源,提高仿真运行速率,缩短运行时间,这称为静态负载平衡。但是由于联邦成员运行的不可预料性,运行一段时间后,一些节点可能有很重的工作量,导致整个仿真联邦负载的不平衡,从而减慢联邦运行速率,增加仿真运行时间,这

是在仿真运行过程中非常不希望看到的。

为了解决这个问题,引入移动Agent技术实现系统动态负载平衡。移动Agent能迁移,并能和其他移动Agent交互,是代理技术和分布式计算技术的结合^[1]。HLA联邦的动态负载平衡利用移动Agent技术,在联邦范围内重新分配运算资源,将重负载节点上的联邦成员调度到轻负载的节点上执行,降低重负载节点上的工作量,达到平衡仿真系统负载,保证仿真运行速率的目的。

文献[2~5]从各种角度探讨了联邦成员迁移的实现方法,有很重要的借鉴意义。但是仅仅实现联邦成员迁移是不够的,同样重要的是还要控制整个联邦成员迁移过程,以最佳的方式保证仿真运行过程。

文中主要探讨HLA环境下,应采用何种控制技术控制联邦负载平衡,才能保证仿真运行的稳定、高效。

1 动态负载平衡的关键技术探讨

将联邦成员从重负载节点调度到轻负载节点的过程称为联邦成员迁移,可以迁移的联邦成员称为移动

收稿日期:2011-09-28;修回日期:2011-12-31

基金项目:武器装备预研基金(编号略)

作者简介:马善达(1988-),男,江苏连云港人,硕士研究生,研究方向为计算机仿真;万晓冬,硕士生导师,研究方向为计算机仿真、嵌入式系统。

Agent 联邦成员。下面就探讨在执行上述过程时,需要用到的关键技术。

1.1 迁移决策的主要技术

在仿真运行过程中,存在三个仿真运行比率:一个是设定的仿真运行比率,这个值是运行时设定的期望值;一个是当前仿真运行比率,当前仿真运行比率=时间步长/当前时间步长的联邦运行时间,它体现当前时间步长的运行比率,具有实时性,能反映仿真运行当前状态;另一个是总仿真运行比率,总仿真运行比率=总推进时间/总联邦运行时间,它反映联邦运行过程的整体运行状态。

在设定的仿真运行比率较高时,当前仿真运行比率可能无法达到设定值,但是能够维持在一个阈值之内。如果在仿真运行过程之中,当前仿真运行比率变小,变得低于阈值,并且维持一定时间,说明存在重负载节点导致仿真运行变慢,需要重新分配系统资源,平衡系统负载,提高仿真运行速率。

作出进行联邦成员迁移决策时,需要获得一些信息:

(1) 联邦范围内所有其他运行节点的资源信息。

通过比较各个运行节点的资源信息,可以得到较合适的目标节点。

(2) 待迁移联邦成员运行实例的信息。

根据联邦成员实例信息和目标节点的资源信息,估测迁移之后目标节点的负载状况,最终作出是否迁移的决定。

1.2 联邦成员迁移过程实现技术

当前,联邦成员迁移有两种方式:

(1) 一种方式是联邦范围的保存和恢复^[2]。

所有的联邦成员都受到迁移的影响,系统需要保存和恢复全部的 RTI 状态以及待迁移的联邦成员的状态。优点是避免了消息的丢失和重复运行的问题。缺点是随着联邦成员数量的增加,联邦成员迁移的开销会加大。

(2) 另一种是冻结自由的方式^[4]。

联邦成员的迁移不影响其他成员,只保存待联邦成员状态以及与其相关 RTI 的状态。无需保存所有的 RTI 状态,减小了开销。但是这种方式最大缺点就是会产生消息丢失。

文中探讨的控制技术在 DMSO-RTI 上实现,在联邦运行过程中,联邦成员包括一个用户设计的仿真模块和一个本地 RTI 组建(LRC)。DMSO-RTI 提供了支持联邦状态保存和恢复的机制和相关服务。若将此种机制应用到联邦成员迁移过程中,能够实现部分状态的自动保存和恢复,可以

减少联邦成员开发人员的工作量。

在联邦状态保存和恢复过程中,根据状态管理者不同,联邦的状态分为两个部分^[6]:

(1) 一部分由 RTI 管理。

包括相应的 FedExec 进程的状态;联邦成员的内部状态,指联邦执行中联邦成员实例的状态。

(2) 另一部分由各联邦成员自己管理。

这部分称为联邦成员外部状态,指联邦成员应用程序执行过程中其内部对象以及相关进程的状态。这部分状态的保存过程,是开发人员在联邦成员开发阶段需要进行的工作,它的实现方式由开发人员决定,给予较大自由度。在 Windows 平台下的 MFC 开发环境中,可以采用 MFC 的序列化机制^[7]进行状态的保存。

联邦状态保存阶段,RTI 将由其管理部分的状态保存在指定目录下的指定文件中,文件命名规则与联邦.fed 文件名和联邦成员名称相关;程序设计者将其负责保存的联邦成员外部状态保存至指定目录下,可以选择上述目录。

已有的代码传送机制主要采用两种传送方式^[8]:全传送和按需传送。代码源的选择方面有两种:移动代码源和固定代码源。在移动 Agent 代码传送机制中,代码源和传送方式一般是相关的。采用全代码传送方式的系统多选择移动代码源,这种方式保证在任何情况下移动 Agent 都能正常运行,稳定性高。相反,按需传送的系统多采用固定代码源,这种方式的优点是每次仅传送移动 Agent 运行所需的代码,减小了网络传输量。

2 仿真负载平衡控制的设计

为实现负载平衡,采用的联邦成员迁移框架如图 1 所示:

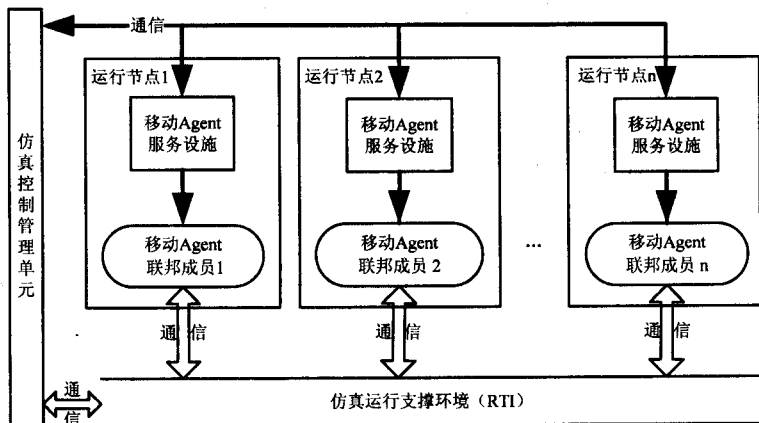


图 1 联邦成员迁移框架

如图 1 所示,主要由以下两个组件配合联邦成员和 RTI 完成联邦成员迁移过程:

(1) 仿真控制管理单元。

(2)移动 Agent 服务设施^[9]。

整个联邦成员迁移框架采用总-分结构,仿真控制管理单元运行在某一节点上,控制整个仿真运行过程,对联邦成员迁移过程作出指挥和调控,是决策产生部分;移动 Agent 服务设施驻留在联邦范围内的各个运行节点上,负责本地节点资源的采集和上传,联邦成员实例程序的启动,相关代码和状态保存文件的发送。

在仿真配置管理阶段,仿真控制管理单元与各个运行节点上的移动 Agent 服务设施建立 Socket 连接,检测联邦内各个运行节点的资源信息;结合仿真中联邦成员实例信息,将联邦成员合理分配在各个运行节点之上,我们称这个过程为静态负载均衡^[10]。

这个过程是仿真配置过程的重要步骤,传统仿真运行中,主要依靠这一步骤来保证仿真运行速率。同时,对于我们的设计,静态负载均衡能减少仿真运行过程中的联邦成员移动次数,提高联邦运行效率。

根据联邦成员配置情况和各个运行节点上存在的成员实例,自动将联邦成员运行实例发送到需要的节点上,无需手动拷贝,减少工作量,适应分布式仿真的特点。

下面主要介绍有关联邦成员迁移的实现。

2.1 联邦成员迁移决策模块设计

联邦成员迁移的决策模块在仿真控制管理单元中实现,是集中式的管理方式。这种方式与联邦成员自主决策方式相比,灵活性降低,但是管理清晰、有序,不会产生冲突和混乱。

联邦成员迁移的决策过程需要的两个信息:

(1)联邦成员实例信息是固定不变的,存储在数据库中,在决策过程中从数据库表中读取。

(2)运行节点的资源信息是实时变化的,采用“索取”方式获取:仿真控制管理单元实时监测仿真运行比率,当仿真运行比率低于阈值时,发送命令获取各个运行节点上的资源信息。

联邦成员迁移决策流程如图2所示。

图2中,目标节点的选择模块和作出迁移决策模块根据仿真运行需要,自主实现其具体内容。例如,迁移决策的作出也受制于当前的联邦执行状态(如果联邦执行很快结束的话,可能就不会移动);

此外,由于一些联邦成员与邻近的资源通信非常多,编程人员要考虑到这些联邦可能不适合移动,在联邦成员实例信息中需要标明。

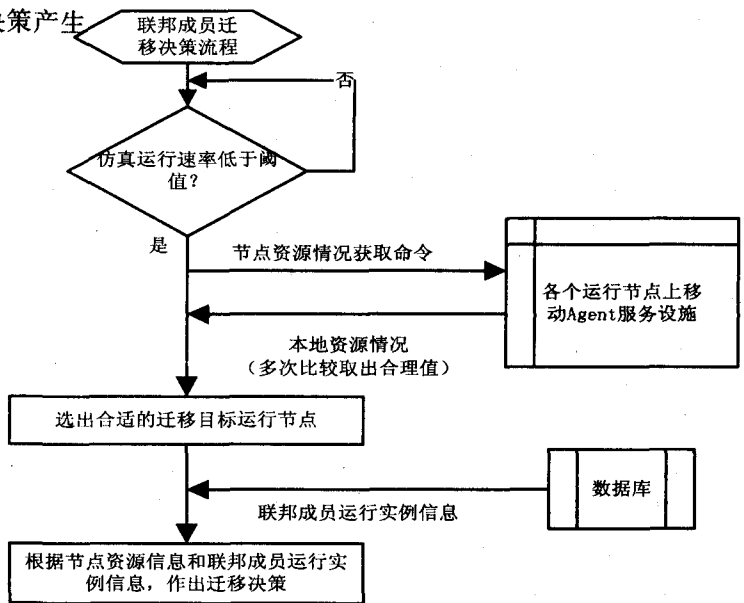


图2 联邦成员迁移决策流程

2.2 联邦成员迁移过程设计

考虑到文中所探讨的仿真控制技术的应用背景,冻结自由方式会产生消息丢失,这样会带来不可预料的结果,影响仿真运行的准确性,所以选择联邦范围的保存和恢复方式。

采用移动代码源的全代码传送方式具有高稳定的特点,能保证仿真运行不会出现意外,是一个较理想的方式,故采取此种方式进行联邦成员实例代码的传递。

根据迁移决策,完成联邦状态保存过程;将源节点上的联邦成员实例代码和保存的联邦成员状态文件发送到目标节点,并保存在指定目录下;在仿真恢复阶段就可以根据保存的文件重现早前的联邦。

采用的联邦成员迁移的控制过程^[11]如图3所示:

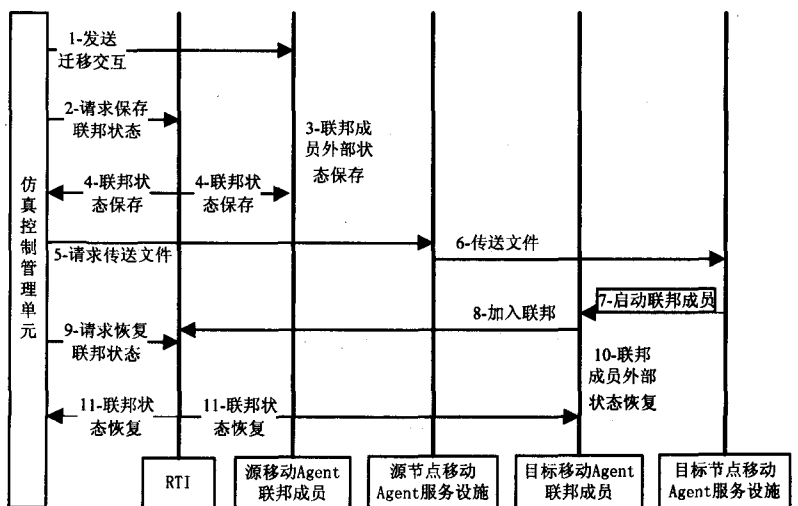


图3 联邦成员迁移控制过程

各步骤解释如下:

(1) 仿真控制管理单元在作出迁移决策之后, 首先向待迁移的联邦成员发送“迁移”交互。接收到此交互的联邦成员将布尔型的联邦成员外部状态保存标志设置为 true, 这样就保证在第三步中, 只有待迁移的联邦成员保存外部状态, 减小状态保存的工作量。

(2) 迁移交互发送完毕, 仿真控制管理单元请求联邦状态保存, 开始联邦状态保存过程。

(3) 联邦成员在收到 RTI 的 `initiateFederateSave()` 回调函数时, 根据联邦成员外部状态保存标志决定是否保存其外部状态。在 RTI 1.3 版本中, 不需要调用 `federateSaveBegin()` 服务, 但是考虑到兼容性, 联邦成员调用此服务通知 RTI, 其开始进行联邦成员外部状态保存。

(4) 待迁移联邦成员保存其联邦成员外部状态之后向 RTI 发送 `federateSaveComplete()` 服务, 其余联邦成员直接向 RTI 发送此服务。RTI 在收到所有联邦成员的 `federateSaveComplete()` 服务之后, 通过回调函数 `federationSaved()` 通知各个联邦成员状态保存完毕。此时, 要暂停整个联邦执行, 待迁移联邦成员退出联邦执行。

(5) 仿真控制管理单元向源节点移动 Agent 服务设施发送传送文件命令, 并告知待迁移的联邦成员实例程序名称、目标节点 IP 地址等信息。

(6) 源节点移动 Agent 服务设施收到上述命令之后, 建立与目标节点移动 Agent 服务设施的 Socket 连接, 将运行实例、状态保存文件等相关内容发送到目标节点上。

(7) 目标节点移动 Agent 服务设施启动接收到的联邦成员运行实例。

(8) 新联邦成员完成加入联邦在内的前期准备工作。

(9) 仿真控制管理单元在发现新的联邦成员加入联邦之后, 请求联邦状态恢复。

(10) 联邦成员根据目录下是否存在外部状态文件, 选择恢复或者不恢复联邦成员外部状态。新的联邦成员外部状态恢复完毕或者无恢复要求的联邦成员, 调用 `federateRestoreComplete()` 服务通知 RTI。

(11) RTI 收到所有成员的 `federateRestoreComplete()` 服务, 通过 `federationRestored()` 回调函数通知各个联邦成员, 联邦成员状态恢复完毕, 可以继续仿真运行。

联邦成员内部状态和相应 FedExec 进程的状态保存和恢复在上述过程中已经由 RTI 执行, 无需程序开发人员实现。

3 结束语

论文探讨了在 HLA 框架内, 结合移动 Agent 技术, 实现系统负载平衡的控制技术。采用集中式管理方式, 结合静态负载平衡和动态负载平衡, 保证仿真运行速率。在其中, 联邦的设计者要对存储和恢复执行状态编码负责, 这可能是一项艰巨的任务。

在一个大型分布式仿真系统中, 进行负载平衡, 只是联邦成员移动的一个基本的范例。联邦成员迁移过程的原因不止是需要进行负载平衡, 还有诸如出现联邦成员需要调用较大的异地数据资源等情况, 都需要进行联邦成员迁移。出现这些情况只需要将文中探讨的控制技术的迁移条件进行重新编码, 就可以实现。

以后工作中, 会考虑扩大此控制技术的应用范围。

参考文献:

- [1] 王卫青, 华祖耀. 基于 Agent 体系结构的 HLA 联邦成员设计与实现[J]. 计算机仿真, 2006, 23(7): 142-145.
- [2] Zajac K, Buba M, Malawski M, et al. Towards a Grid Management System for HLA-based Interactive Simulations[C]//Proceedings of the Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications. [s. l.]: [s. n.], 2003.
- [3] Eklöf M, Sparf M, Moradi F, et al. Peer-to-peer-based Resource Management in Support of HLA-based Distributed Simulations[J]. Simulation, 2004, 80(4-5): 181-190.
- [4] Yuan Zijing, Cai Wentong, Low M Y H. Federate Migration in HLA-based Distributed Simulation[C]//International Conference on Computational Science (ICCS2004). Poland: [s. n.], 2004.
- [5] Cai Wentong, Turner S J, Zhao Hanfeng. A Load Management System for Running HLA-based Simulation over the Grid[C]//Proceedings of the Sixth IEEE International Symposium on Distributed Simulation and Real Time Applications. [s. l.]: [s. n.], 2002.
- [6] 周彦, 戴剑伟. HLA 仿真应用程序设计[M]. 北京: 电子工业出版社, 2002.
- [7] 张柯, 张新宇. 基于 HLA 的分布仿真系统数据采集解决方案[J]. 系统仿真学报, 2004(12): 2725-2728.
- [8] 王红. 移动 Agent 关键技术研究[D]. 北京: 中科院, 2002.
- [9] 张云勇, 刘锦德. 移动 Agent 技术[M]. 北京: 清华大学出版社, 2003.
- [10] 王庆俊, 李晋惠, 梁向阳. 基于 HLA/RTI 的客户/服务员模型的实现[J]. 微机发展(现更名: 计算机技术与发展), 2004, 14(4): 40-42.
- [11] 王琼, 艾丽蓉, 龚爱珍. 基于 HLA 的分布式仿真中负载平衡的研究[J]. 计算机技术与发展, 2011, 21(6): 119-122.