

## 一种基于 CORBA 及互操作协议的分布仿真基础支撑平台实现方法\*

姚益平, 郭敏, 卢锡城

(国防科技大学计算机学院, 湖南长沙 410073)

**摘要:**为解决现有 RTI 服务器在大规模仿真中容易成为影响系统效率的瓶颈、仿真规模受到限制的问题,提出了一种基于 CORBA 及互操作协议的层次式分布仿真运行支撑环境(RTI)实现方法,阐述了盟员、局部 RTI 服务器及中心 RTI 服务器的部署方式,给出了基于 CORBA 及内部互操作协议的层次式 RTI 的工作流程。既解决了大规模仿真中集中式及功能分布式 RTI 服务器的瓶颈问题,减少了全局操作的延迟,也为各仿真盟员的时空一致性提供了保障。

关键词:HLA; RTI; CORBA; 互操作

中图分类号:TP391.9 文献标识码:A

## An Implementation Method of RTI Based on CORBA and Interoperability Protocol

YAO Yi-ping, GUO Min, LU Xi-cheng

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** To solve the performance bottleneck of the existing RTI servers in the large-scale simulation, an implementation method based on CORBA and interoperability protocol of hierarchical RTI server is presented. The deployment of federates, local RTI servers and central RTI server is described. The workflow of hierarchical RTI based on CORBA and interoperability protocol is introduced. This method not only solves the bottleneck problem of centralized and distributed RTI servers in the large-scale simulation, and decreases the latency of operation, but also ensures time and space consistency for all federates.

Key words: HLA; RTI; CORBA; interoperability

高层体系结构 HLA(High Level Architecture)已于 2000 年 9 月被定为国际分布仿真通用标准 IEEE 1516<sup>[1]</sup>,它为建模与仿真提供了一个通用的技术框架和开放的标准。RTI(RunTime Infrastructure)是 HLA 框架的核心,它实现了接口规范中定义的服务,其目的是将仿真应用和底层通信等基本功能相分离。由 RTI 提供对底层通信等基本功能的支持,即在同一联盟执行过程中,所有的盟员按照 HLA 接口规范说明要求同 RTI 进行数据交换,实现盟员之间的互操作。同时遵循共同的 RTI 接口的仿真应用可以灵活地组成功能各异的联盟,有利于构件的重用以满足不同需要。RTI 相当于一个分布式操作系统,HLA 规定了 RTI 应提供哪些服务功能,但并未具体规定应如何部署及实现 RTI<sup>[2]</sup>。

## 1 现有的几种 RTI 实现方法简介

## 1.1 集中式 RTI

所谓集中式 RTI 即是在一个联盟中只部署一个 RTI 服务器,HLA 的所有功能都集中在该 RTI 服务器上实现(见图 1)。由于 HLA 规定盟员之间不能直接通信,盟员之间的所

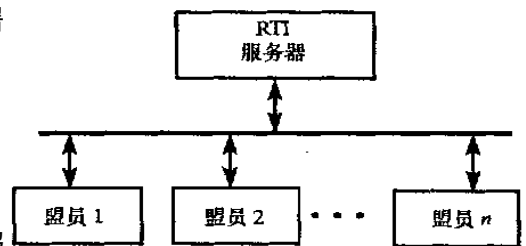


图 1 集中式 RTI 结构模型

Fig. 1 The structure model of centralized RTI server

\* 收稿日期:2003- 1- 11

基金项目:国家 863 高技术资助项目(2001AA115127)

作者简介:姚益平(1963-),男,教授,硕士。

有通信都必须经过 RTI, 因此, 集中式的 RTI 服务器在大规模仿真中容易成为影响系统效率的瓶颈, 不利于仿真规模的扩充, 不适合大规模仿真的需要。

## 1.2 功能分布式 RTI

为了解决集中式 RTI 的效率瓶颈, 美国国防部建模与仿真办公室 DMSO 提出了功能分布式 RTI 的实现方法(见图 2)。所谓功能分布式的 RTI 即是把 RTI 的实现分成两部分: 一部分称为中心 RTI 部件 CRC(Central RTI Component), 主要负责联盟管理、对象管理中的全局名及全局标识管理、所有权管理以及时间管理中的时间计算等工作; 另一部分称为局部 RTI 部件 LRC(Local RTI Component), 主要负责声明管理、对象管理中的对象发现、数据更新、交互等、时间管理中的逻辑时间、消息排序等、以及数据分发管理等功能。在一次分布式仿真中, 系统部署一个中心 RTI 部件 CRC, 同时每个盟员中部署一个局部 RTI 部件 LRC, CRC 部件、每个 LRC 部件之间都相互连接。使用功能分布式 RTI 仿真, 要求在 CRC 和不同的 LRC 之间维护数据的一致性。随着仿真规模的增大, 数据一致性维护十分困难; 同时由于在功能分布式 RTI 的实现中要求 CRC、每个 LRC 之间相互通信, 因此以 CRC、LRC 为节点将组成一个全连通的通信拓扑结构, 造成网络中大量的信息交互, 导致节点数目受到网络通信的限制, 因而仿真规模也受到限制。功能分布式的 RTI 产品有 DMSO 的 RTI 1.3NG-v5、瑞典 Pitch 公司研制的 pRTI<sup>[3]</sup> 等。

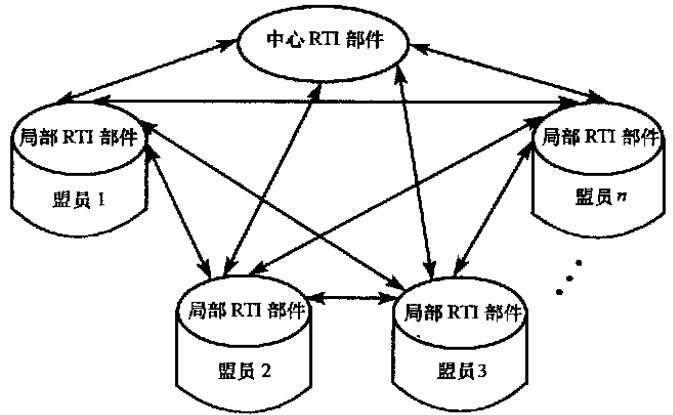


图 2 功能分布式 RTI 结构模型

Fig. 2 The structure model of distributed RTI server

系统部署一个中心 RTI 部件 CRC, 同时每个盟员中部署一个局部 RTI 部件 LRC, CRC 部件、每个 LRC 部件之间都相互连接。使用功能分布式 RTI 仿真, 要求在 CRC 和不同的 LRC 之间维护数据的一致性。随着仿真规模的增大, 数据一致性维护十分困难; 同时由于在功能分布式 RTI 的实现中要求 CRC、每个 LRC 之间相互通信, 因此以 CRC、LRC 为节点将组成一个全连通的通信拓扑结构, 造成网络中大量的信息交互, 导致节点数目受到网络通信的限制, 因而仿真规模也受到限制。功能分布式的 RTI 产品有 DMSO 的 RTI 1.3NG-v5、瑞典 Pitch 公司研制的 pRTI<sup>[3]</sup> 等。

## 2 基于 CORBA 及互操作协议的层次式 RTI

### 2.1 结构模型

针对集中式 RTI 服务器和功能分布式 RTI 服务器在大规模仿真中容易成为影响系统效率的瓶颈, 根据 Myjak 等人提出的 RTI 互操作思路<sup>[4]</sup>, 在这里提出一种基于 CORBA 及内部互操作协议的层次式 RTI 实现方法。

所谓层次式 RTI 是根据仿真规模的不同, 在局域网或广域网上部署多个 RTI 服务器, 其中有一个负责全局操作的中心 RTI 服务器, 在中心 RTI 服务器下设置一组局部 RTI 服务器, 各局部 RTI 服务器负责一组盟员的服务请求, 涉及到全局操作的请求则由中心 RTI 服务器协调完成(见图 3)。各局部 RTI 服务器与中心 RTI 服务器构成一个逻辑整体, 对盟员来讲, 这个逻辑整体如同一个集中式 RTI 服务器。盟员只需向管理自己的 RTI 服务器请求服务, 不必关心 RTI 服务器之间的内部通信。每个盟员只看到一个 RTI 服务器, 这个 RTI 服务器通常称作该盟员的“本地 RTI 服务器”。如果一个盟员需要与不同的 RTI 服务器上的盟员进行信息交换, 则由这两个盟员的本地 RTI 服务器进行协商; 与全局相关的状态和控制信息则与中心 RTI 服务器协商解决。

### 2.2 实现方法

CORBA 是位于操作系统和应用程序之间的软件, 通过“Internet 互操作协议 IIOP”实现基于不同操作系统平台以及不同程序设计语言的应用程序之间的互操作。大多数 RTI 系统需要实现网络环境下的多个仿真盟员和多个 RTI 服务器之间的互操作, 这些仿真盟员以及 RTI 服务器可能位于不同的操作系统平台(如 Windows 或 Unix), 并且可能基于不同的程序设计语言(如 C++ 或 Java), 通过 CORBA 中间件, 盟员和 RTI 服务器之间的调用关系如同在一台主机上的同一个应用程序中的多个子程序或函数之间的相互调用, 因为网络之间具体的通信细节已经被 CORBA 中间件很好地解决了。CORBA 技术的这些优势是传统的 TCP/IP 技术所无法比拟的。采用 CORBA 技术设计层次式 RTI 服务器系统, 无需考虑如何

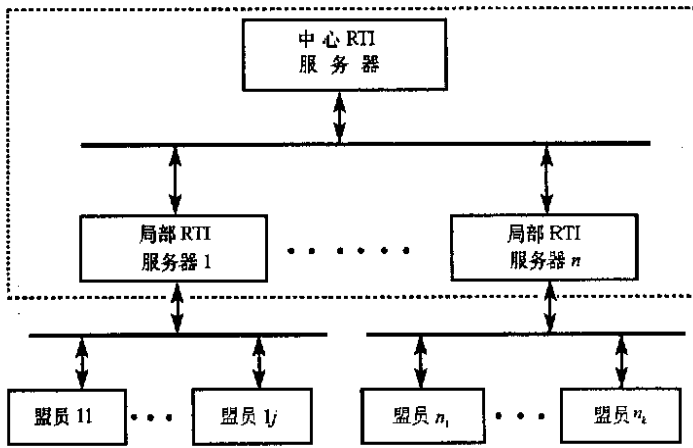


图 3 层次式 RTI 部署结构模型

Fig. 3 The deployment model of hierarchical RTI server

管理和通过网络访问异地对象,可以集中精力设计和实现系统的功能,从而大大减少了分布式应用系统开发的工作量;且使整个系统具有很好的可重用性、移植性和互操作性,提高了分布式应用系统的开发效率。具体地说,层次式 RTI 运用 CORBA 的代理技术实现盟员和 RTI 服务器之间的相互访问过程如下:在盟员方设置一个 RTI 服务器的代理,盟员首先将请求交给 RTI 服务器代理,再由 RTI 服务器代理将盟员的请求转交给 RTI 服务器,这样就实现了盟员对 RTI 服务器的请求;在 RTI 服务器代理与 RTI 服务器之间的网络操作细节由 CORBA 技术自动实现;同样,在 RTI 服务器方设置一个盟员代理,RTI 服务器首先将结果转交给盟员代理,然后再由盟员代理将结果回调给盟员,这样就实现了 RTI 服务器对盟员的回调,盟员代理和盟员之间的网络操作细节由 CORBA 技术自动实现,不需要盟员编写额外的实现代码。在层次式 RTI 系统中,各 RTI 服务器实现 IEEE 1516 标准规定的管理功能,为盟员提供一致的接口;RTI 服务器之间通过 CORBA 内部的互操作协议进行数据和状态信息交换。其实现结构如图 4 所示。

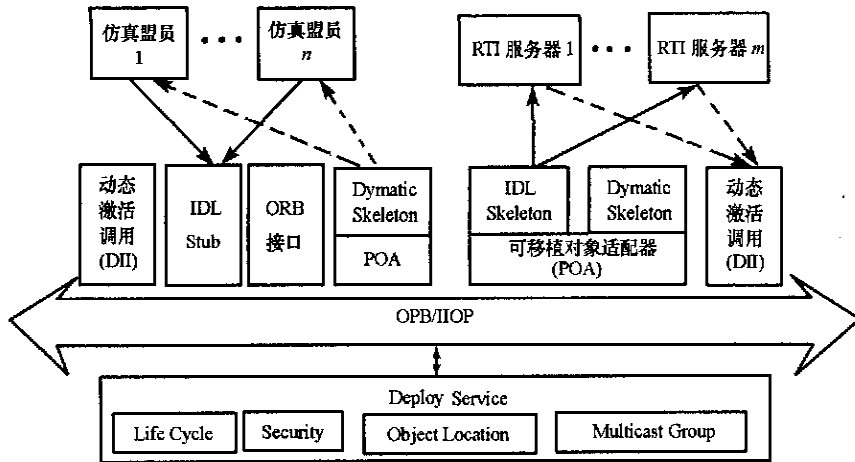


图 4 基于 CORBA 的层次式 RTI 系统结构

Fig. 4 Architecture of CORBA-based hierarchical RTI

层次式 RTI 内部互操作协议可从三个方面来说明,即盟员和 RTI 服务器之间的互操作、盟员之间的互操作以及 RTI 服务器之间的互操作。盟员和 RTI 服务器之间的互操作通过 RTI 服务器代理和盟员代理来协同完成。根据 HLA 规范的要求,盟员之间并不直接通信,而是通过 RTI 服务器来完成。为了减少盟员开发的复杂性,规定每个盟员只能与一个 RTI 服务器直接通信,使用同一个 RTI 服务器的所有盟员位于同一个“组”。同组内的盟员之间的数据交互由本地 RTI 服务器处理并完成;不同组内的任意两个盟员之间的数据交互则通过各自的本地 RTI 服务器来完成。盟员不与中心 RTI 服务器直接通信,所

有的本地 RTI 服务器相对于中心 RTI 服务器而言都是“局部 RTI 服务器”。每个 RTI 服务器在实现时都定义两套接口, 一套接口遵循 IEEE 1516 标准, 用于盟员和 RTI 服务器的交互(中心 RTI 服务器可不提供该接口); 另一套接口为完成 RTI 服务器之间的通信(包括局部 RTI 服务器与局部 RTI 服务器之间以及局部 RTI 服务器与中心 RTI 服务器之间的通信)提供服务, 主要包括对应联盟管理的相关接口、对对象管理中的全局名及全局标识管理的相关接口、对应所有权管理的相关接口、对应时间管理中的时间计算的相关接口等。不同 RTI 服务器可以并发执行, 这样大大提高了整个仿真系统的效率。

### 2.3 工作流程

从仿真的过程来看, 层次式 RTI 工作流程可大致分为启动、执行和结束三个阶段。在启动阶段主要完成联盟的创建、盟员加入联盟、初始化数据、盟员能够公布的信息以及盟员需要定购的信息等工作。启动阶段完成后, 就进入仿真的执行阶段: 盟员向 RTI 服务器请求时间的推进, 如果 RTI 同意其推进, 则盟员推进到相应的时间; 否则盟员的推进请求被挂起, 直到 RTI 服务器认为该盟员可以推进时才允许其推进。在时间推进被允许后, 盟员能够做各类操作, 除了完成本地的计算外, 可以进一步要求“公布/定购”信息、可以登记或删除自己创建的对象、可以更新对象的属性值、可以发送特定的交互(例如“开火”等事件)。本地 RTI 服务器在接收到盟员的请求后, 如果认为该请求能够自行解决, 则把相应结果回调给本地盟员; 如果本地 RTI 服务器认为该请求涉及到与全局状态相关的信息, 则把请求传送给中心 RTI 服务器, 由中心 RTI 服务器与局部 RTI 服务器协同解决; 如果本地 RTI 服务器认为不需要中心 RTI 服务器参与, 则可以请求相关的远地 RTI 服务器协同解决。当仿真任务完成后, 联盟就进入结束阶段, 盟员向本地 RTI 服务器请求退出联盟执行, 当局部 RTI 服务器上的所有盟员都退出联盟执行后, 局部 RTI 服务器从中心 RTI 服务器中退出, 同时删除局部 RTI 服务器上的联盟执行, 当所有局部 RTI 服务器都从中心 RTI 服务器中退出后, 删除中心服务器上的联盟执行。

## 3 结束语

采用基于 CORBA 及内部互操作协议的层次式 RTI 可达到以下效果:

- (1) 由于层次结构中的不同 RTI 服务器可以并发执行, 提高了仿真系统的效率, 解决了大规模仿真中集中式及功能分布式 RTI 服务器的瓶颈问题;
- (2) 由于设置了中心 RTI 服务器, 时间管理中的时间计算等工作由中心 RTI 服务器统一完成, 为广域网上各仿真盟员的时间一致性提供了保障;
- (3) 由于采用 CORBA 及内部互操作协议, 无须考虑网络通信细节, 可以集中精力设计和实现系统的功能, 大大减少了分布式应用系统开发的工作量, 而且整个系统具有很好的可重用性、移植性和互操作性, 适合于大规模、多层次的分布式仿真需要。

## 参考文献:

- [1] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture(HLA)—IEEE Std 1516- 2000, 1516. 1- 2000, 1516. 2- 2000[S]. New York: The Institute of Electrical and Electronics Engineers Inc., 2000.
- [2] 姚益平, 等. HLA/RTI 的研究与实现[J]. 系统仿真学报, 2000, 12(4): 364- 366.
- [3] Karlsson M, Olsson L. pRTI 1516 Rationale and Design[R]. Fall Simulation Interoperability Workshop, 01F- SIW- 038, 2001.
- [4] Myjak M D, Clark D, Lake T. RTI Interoperability Study Group Final Report[R]. Fall Simulation Interoperability Workshop, 99F- SIW- 001, 1999.