

# 层次式仿真运行支撑环境 StarLink 中的关键技术\*

刘步权<sup>+</sup>, 王怀民, 姚益平

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

## Key Techniques of a Hierarchical Simulation Runtime Infrastructure—StarLink

LIU Bu-Quan<sup>+</sup>, WANG Huai-Min, YAO Yi-Ping

(School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4573697, E-mail: bqliu@nudt.edu.cn, <http://www.nudt.edu.cn>

Received 2002-11-27; Accepted 2002-12-31

Liu BQ, Wang HM, Yao YP. Key techniques of a hierarchical simulation runtime infrastructure—StarLink. *Journal of Software*, 2004,15(1):9-16.

<http://www.jos.org.cn/1000-9825/15/9.htm>

**Abstract:** A distributed RTI cannot be well adapted to large-scale simulations in a wide area network, especially on maintaining the consistency of the whole federation state. However a hierarchical RTI (runtime infrastructure) is very useful to reduce the computational cost of the large-scale simulations. This paper explains why the hierarchical simulation runtime infrastructure StarLink can be applied to the wide area network, and discusses three implementation techniques of StarLink, including CORBA based networking technique, multiple-threaded data transferring technique and XML based data exchanging technique. In addition, test methods for StarLink are discussed, and a new idea is put forward to test RTI's interface services using a common framework test program. StarLink can effectively provide a powerful technology support for the large-scale simulations.

**Key words:** HLA/RTI; CORBA; wide area network; large-scale simulation; framework test

**摘要:** 在广域网环境下,特别是在维护联盟状态信息的一致性方面,分布式 RTI(runtime infrastructure)难以适应大规模仿真的应用需求,而层次式 RTI 则有效地降低了大规模仿真中的计算开销.论述了面向广域网的仿真运行支撑平台 StarLink 采用层次式体系结构的理由,并从 3 个方面介绍了 StarLink 的实现技术,包括 CORBA 中间件技术、多线程技术以及基于 XML 格式的数据交互技术.最后,讨论了 StarLink 的测试方法,提出了基于“框架结构”的功能测试思想. StarLink 为面向广域网的大规模仿真应用提供了强有力的技术支持.

**关键词:** HLA/RTI;CORBA;广域网;大规模仿真;框架测试

中图法分类号: TP311 文献标识码: A

---

\* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA115127 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.G1999032703 (国家重点基础研究发展规划(973))

**作者简介:** 刘步权(1969—),男,江苏姜堰人,博士生,讲师,主要研究领域为分布式仿真,分布对象技术,操作系统;王怀民(1962—),男,博士,教授,博士生导师,主要研究领域为分布计算技术,信息安全;姚益平(1963—),男,研究员,主要研究领域为软件工程,分布式仿真,虚拟现实.

高层体系结构 HLA(high level architecture)<sup>[1-3]</sup>是美国国防部为解决美军在各个领域开发出来的多种模型、仿真系统和 C4I 系统的互联和互操作问题而提出的一种新型的分布式仿真协议。2000 年 9 月,HLA 被正式接纳为 IEEE 的分布式仿真标准,即 IEEE 1516 系列标准。在此之前,美军提出了 HLA 1.3 协议,并推出了在其国内可自由下载的 RTI 1.3 系列软件以及相应的开发工具<sup>[4]</sup>。在我国,HLA 技术同样受到了高度重视,并且已经研制成功了多个 RTI 产品,包括国防科学技术大学的 StarLink 和 KD-RTI<sup>[5]</sup>以及北京航空航天大学 DEV\_RTII<sup>[6]</sup>等。

在广域网环境下,特别是在维护与全局操作相关的状态信息时,现有的分布式 RTI 已经很难适应大规模仿真的应用需求,本文对面向广域网的仿真运行支撑平台 StarLink 的研制背景和所采用的技术进行了介绍。第 1 节描述广域网环境下大规模仿真的困难。第 2 节阐述为什么 StarLink 的层次式体系结构能够适应广域网的大规模系统仿真。第 3 节描述了 StarLink 的实现技术,包括基于 CORBA 的中间件技术、多线程技术以及遵循 IEEE 1516 标准的数据交互技术等。第 4 节则从功能测试和规模测试两方面介绍了 StarLink 的测试情况,提出了基于框架结构的功能测试思想。最后对 StarLink 所采用的软件技术进行了小结。

## 1 广域网大规模仿真的困难

在广域网环境下,美国国防部的目标是实现由 100 000 个仿真对象参加的分布式仿真系统,并且大部分对象都是由计算机自动生成且系统中的人机交互行为应尽可能地少<sup>[7]</sup>。在这里有必要指出,美国国防部提出的目标是针对对象而非主机而言的,所有仿真对象可能分布于广域网中的成百上千台主机中,但是对象和主机不是一对一的。

在现有条件下,要实现美国国防部提出的大规模仿真目标是很困难的,广域网环境下的大规模仿真的困难可从网络、RTI 和应用 3 个方面考虑。

基础网络设施。要将 HLA 成功地运用于大规模的系统仿真,必须要有高带宽、低延迟的先进的网络设施。而现有的网络设施没有能力承担如此巨大规模的系统仿真,下面以 100M 以太网为例说明其原因。就网络延迟而论,一个对象在 100M 以太网中接收一条消息的延迟通常以毫秒计,假设该延迟为 1ms,则 1s 内一个对象只能串行接收来自 1 000 个对象的消息。如果一台主机加载 20 个对象,则只能支持 50 台主机规模的仿真。相比于局域网而言,广域网环境下的系统仿真就更加困难,因此应该为大规模的系统仿真建设性能更高的、传输服务可靠的、专用的广域网。

运行支撑平台 RTI。RTI 要能够支持广域网环境下的大规模系统仿真,必须要有良好的体系结构和高效的内部实现技术的支持。现有 RTI 的体系结构大多为分布式,RTI 与应用程序捆绑在一起,本质上类似于分布交互仿真 DIS 的点点通信方式,分布式体系结构难以适应广域网下的大规模系统仿真。第 2 节将会介绍 StarLink 所采用的一种更加灵活的体系结构,即层次式体系结构。另外,RTI 必须具有并行处理网络消息的能力,第 3 节将会介绍 StarLink 所采用的基于 CORBA 和多线程机制的并行计算技术。

仿真应用。在广域网环境下,仿真应用程序本身对仿真的规模也有影响,例如仿真应用程序的实时性要求、是否使用与 RTI 全局操作相关的服务(所有权服务和时间管理服务)等。就仿真的实时性而言,应用程序的实时性要求越高,仿真的规模就越小;反之就越大。显然,如果仿真应用程序每 3s 更新一次数据,其规模要比每 50ms 更新一次数据更容易扩大。另外,如果应用程序使用了 HLA 的所有权服务和时间管理服务,则分布式 RTI 在处理这些服务时需要获取所有盟员的状态,会导致额外的网络延迟开销,从而影响仿真系统的规模。第 2 节对此将会有较为详细的讨论。

另外,广域网环境下的 RTI 软件还必须正确处理好网络服务质量以及可靠性等诸多问题。广域网大规模仿真困难的根本原因在于网络设施不能够满足仿真应用的实时性要求。

## 2 StarLink 的体系结构

HLA 规范只是规定盟员必须通过 RTI 接口服务进行通信,但并没有指明应该如何实现 RTI。现有 RTI 的体系结构可分为集中式<sup>[8]</sup>和分布式<sup>[4-6,9-11]</sup>两种,而 StarLink 则提出了具有集中式和分布式两类 RTI 体系结构特点

的层次式 RTI<sup>[12]</sup>.

如图 1(a)所示,集中式 RTI 只存在一个中心 RTI 服务器,盟员通过中心 RTI 服务器进行交互.图 1(b)和图 1(c)是两种典型的分布式 RTI 的体系结构.功能分布式 RTI<sup>[4,5,9,10]</sup>包含一个类似于集中式 RTI 的中心 RTI 部件 CRC,每个盟员还包含一个本地 RTI 部件 LRC,CRC 和 LRC 构成一个逻辑上的 RTI 服务器.而全分布式 RTI<sup>[6,11]</sup>则由所有盟员的 LRC 构成,不存在中心 CRC.StarLink 采用了如图 1(d)所示的层次式 RTI 的体系结构.每个本地 RTI(LRTI)负责管理多个本地盟员间的信息交互;而不同 LRTI 内部的盟员间的信息交互,则可由中心 RTI(CRTI)协调解决.StarLink 的一个重要特点是能够兼容现有的功能分布式 RTI,即当每个 LRTI 仅管理一个盟员时,StarLink 就成为一个功能分布式 RTI.

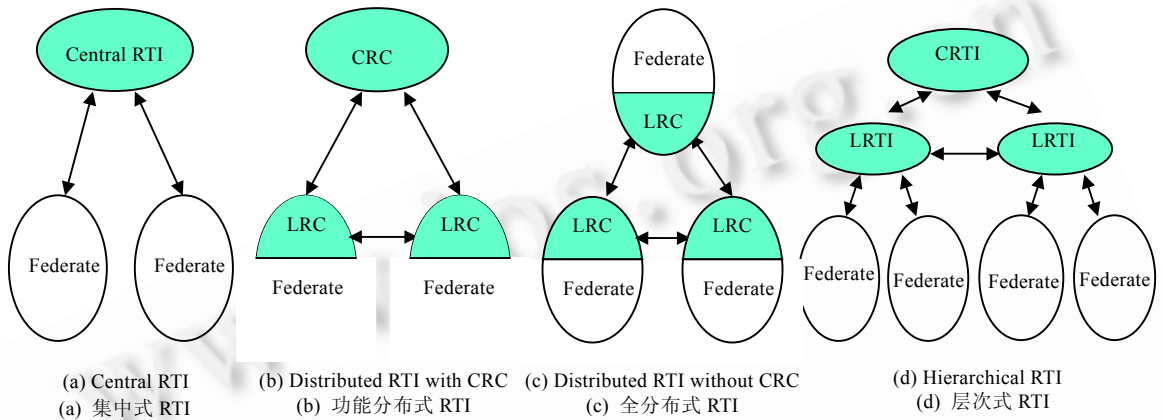


Fig.1 RTI architectures

图 1 RTI 体系结构

RTI 包含两方面的重要功能,其一是作为数据的“传送带”,将公布盟员的更新数据传送给一个或多个定购盟员;其二是作为状态的“控制器”,对全局状态进行控制并维护全局数据的一致性,保证仿真能够正确地进行(例如所有权控制和时间同步等).下面对各类 RTI 在网络环境下的开销进行计算并加以比较.

对于集中式 RTI 而言,数据从发送盟员 sender 到达接收盟员 receiver 所花费的时间  $T(central)$  可以粗略地用下式表示

$$T(central) = T(sender\_RTI) + T(globalComputing) + T(RTI\_receiver) \quad (1)$$

$T(sender\_RTI)$  表示数据从发送盟员 sender 到达 RTI 所花费的时间,  $T(globalComputing)$  表示 RTI 维护全局数据一致性所花费的时间,  $T(RTI\_receiver)$  表示数据从 RTI 到达接收盟员所花费的时间.假设发送者、RTI 和接收者位于不同的主机上,则可简单地认为  $T(sender\_RTI) = T(RTI\_receiver) = T(network)$ .  $T(network)$  表示数据在网络环境下从一个主机传送到另外一个主机所花费的时间.因为对于集中式 RTI 而言,全局数据的一致性维护在 RTI 本地进行,不涉及网络开销,所以  $T(globalComputing) \approx 0$ .再由式(1)可得

$$T(central) = T(network) + T(globalComputing) + T(network) \approx 2T(network) \quad (2)$$

对两种分布式 RTI 而言,数据从发送盟员 sender 到达接收盟员 receiver 所花费的时间  $T(distributed)$  可以粗略地用下式表示

$$T(distributed) = T(sender\_localLRC) + T(globalComputing) + T(localLRC\_remoteLRC) + T(remoteLRC\_receiver) \quad (3)$$

$T(sender\_localLRC)$  表示数据从发送盟员传递给本地 LRC 所花费的时间,  $T(localLRC\_remoteLRC)$  表示数据从本地 LRC 到达远程 LRC 所花费的时间,  $T(remoteLRC\_receiver)$  表示数据从接收盟员 receiver 的 LRC 传递给自己所花费的时间.与集中式 RTI 类似,  $T(localLRC\_remoteLRC) = T(network)$ ,  $T(sender\_localLRC) \ll T(network)$ ,  $T(remoteLRC\_receiver) \ll T(network)$ , 所以由式(3)可得

$$T(distributed) \approx T(globalComputing) + T(network) \quad (4)$$

在式(4)中,  $T(globalComputing)$  依赖于 RTI 所采用的具体算法以及仿真系统的规模等因素.在不需要维护全

局数据一致性的仿真中, $T(\text{globalComputing}) \approx 0$ ,比较式(2)和式(4)可知,分布式 RTI 要优越于集中式 RTI;但是在需要维护全局数据一致性的仿真中,则计算  $T(\text{globalComputing})$ 时需要多个 LRC 进行协商解决,其开销远大于  $T(\text{network})$ .这样,集中式 RTI 的性能将远大于分布式 RTI.

对层次式 RTI 而言,数据从发送盟员 sender 到达另外一个 LRTI 中的接收盟员 receiver 所花费的时间  $T(\text{hierarchical})$ 可以粗略地用下式表示

$$T(\text{hierarchical}) = T(\text{sender\_localLRTI}) + T(\text{globalComputing}) + T(\text{localLRTI\_remoteLRTI}) + T(\text{remoteLRTI\_receiver}) \quad (5)$$

在分布式 RTI 中, $T(\text{sender\_localLRC})$ 和  $T(\text{remoteLRC\_receiver})$ 是本地操作,与数据在网络上传输所花费的时间  $T(\text{network})$ 相比可以忽略不计,但是在层次式 RTI 中却不能被简单地忽略掉.对于局域网而言, $T(\text{sender\_localLRTI}) = T(\text{network})$ , $T(\text{remoteLRTI\_receiver}) = T(\text{network})$ .由式(5)可得

$$T(\text{hierarchicalEthernet}) = T(\text{network}) + T(\text{globalComputing}) + T(\text{network}) + T(\text{network}) = 3T(\text{network}) + T(\text{globalComputing}) \quad (6)$$

对于广域网而言, $T(\text{sender\_localLRTI})$ 和  $T(\text{remoteLRTI\_receiver})$ 相对于数据在不同 LRTI 之间的传输时间而言可以忽略不计(假设局域网中的数据传输时间远小于广域网中的数据传输时间).用  $T(\text{network})$ 表示广域网上数据的传输时间,由式(5)可得

$$T(\text{hierarchicalWAN}) \approx T(\text{globalComputing}) + T(\text{localLRTI\_remoteLRTI}) = T(\text{globalComputing}) + T(\text{network}) \quad (7)$$

式(4)中的  $T(\text{globalComputing})$ 是盟员个数的复杂度,而式(6)和式(7)中的  $T(\text{globalComputing})$ 仅仅是 LRTI 个数的复杂度.比较式(4)和式(7)可知,在广域网环境下,层次式 RTI 要优越于分布式 RTI;比较式(4)和式(6)可知,在局域网环境下则难以对两种体系结构的开销进行比较.

由以上讨论可知,StarLink 适合广域网环境下的大规模系统仿真,其体系结构比现有 RTI 的体系结构更科学,具有集中式 RTI 和分布式 RTI 的双重特点,并与功能分布式 RTI 兼容.

### 3 StarLink 的实现技术

#### 3.1 基于CORBA的实现技术

能够实现基于不同操作系统、网络位置和编程语言的仿真应用之间的互操作是 HLA 所要追求的重要目标,在目前能够见到的 RTI 软件中,大致有 3 类实现方式:TCP/IP, JAVA 和 CORBA<sup>[13]</sup>.这 3 类方式各有其特点, TCP/IP 是网络传输数据的最基本方式, JAVA 和 CORBA 技术从本质上讲都是以 TCP/IP 技术来传输网络数据,但 CORBA 已日渐成为 RTI 实现的主流技术,主要基于下列几点理由:

从技术特点来看,网络环境中的分布计算技术具有十分鲜明的特点:不存在统一的硬件平台、不存在统一的操作系统、不存在统一的网络协议、不存在确定的规模、不存在统一的应用模式.这就要求分布计算技术必须在互操作上达成一致,而这正是 CORBA 技术所要着力解决的问题.作为分布计算技术的一个重要分支,基于 HLA 的分布仿真技术也面临同样的问题. CORBA 是 RTI 实现的天然技术框架, JAVA 只是实现 RTI 编程语言方面的一种选择,还不足以解决分布计算的诸多问题.

从历史渊源来看, CORBA 和 HLA 属于同一时期的技术.两类技术都起源于 20 世纪 80 年代后期,并发展于 20 世纪 90 年代. HLA 本身是为了解决仿真的互操作和可重用问题而提出的分布仿真标准,但是 HLA 标准并没有说明互操作的语法和语义. HLA 是在基于 CORBA 的可重用、可移植和互操作问题已经解决的背景下而提出的分布仿真技术.

从软件开发来看,提高质量与生产率是软件工程要解决的核心问题.高质量的程序设计是非常重要的环节,需要考虑软件的正确性、健壮性、可靠性、效率、易用性、可读性、可扩展性、可复用性、兼容性、可移植性等质量属性,而基于 CORBA 技术开发的软件质量显然要比直接使用 TCP/IP 技术开发的软件质量高.

从发展趋势来看,使用 CORBA 开发 RTI 已经成为一种时尚.美国国防部建模与仿真办公室 DMSO 在使用 TCP/IP 技术开发了多个 RTI 版本之后,已经开始使用 Ace Tao<sup>[14]</sup>的 CORBA 技术来开发新一代的 RTI 软件,即

RTI-NG 1.3 系列软件.

从执行效率来看,已经很难对基于 CORBA 技术和 TCP/IP 技术开发的两类 RTI 软件进行比较.该问题类似于比较用 C++编程语言和汇编语言编写的两类应用程序的效率问题,程序设计语言本身效率的高低并不表示所编写的复杂软件代码的效率的高低.若仅仅以一个消息的传输时间来衡量两类 RTI 软件的执行效率,则这种方法是不科学的.因为在设计 RTI 软件时,需要考虑很多因素.在综合考虑各种因素的基础上,使用 CORBA 技术设计的 RTI 的执行效率有可能还要高些.例如,由 DMSO 使用 CORBA 技术开发的 RTI-NG 1.3v3.2 的执行效率要比 DMSO 使用 TCP/IP 技术开发的 RTI 1.3v7 的效率高<sup>[15]</sup>.

StarBus<sup>[16]</sup>是具有自主知识产权的国产 CORBA 产品,StarLink 就是在 StarBus 的基础上研制开发的.StarBus 队伍中的不少专家和技术人员都参与了 StarLink 的开发.良好的开发环境和科研队伍以及强有力的技术支持是我们采用 CORBA 技术开发 StarLink 的重要原因.

### 3.2 多线程技术

单线程技术不论在分布式 RTI 还是在其他体系结构模式的 RTI 中都具有很大的局限性,并且有可能行不通.HLA 1.3 提供了一种称为“轮询”的进程模式<sup>[17]</sup>.在该模式中,应用程序和本地 RTI 部件(即 LRC)绑定在一起,并共享单个线程,线程控制权的切换由应用程序通过调用 tick()服务进行.在这种方式中,应用程序并不知道 RTI 何时处于空闲状态或忙状态,因此应用程序通常需要为一次 RTI 服务调用足够多的 tick()服务,浪费了大量的处理器时间.

在基于 Client/Server 方式的 StarLink 中,RTI 服务器负责调度多个盟员程序,盟员不可能与 RTI 绑定在一起,因此没有必要使用 HLA 1.3 中的 tick()服务;另外,IEEE 1516 标准也没有提供 tick()服务.StarLink 在设计时必须考虑好两个问题.其一是死锁问题,如图 2(a)所示,盟员和 RTI 均采用单线程方式,盟员首先调用 RTI,并等待 RTI 的应答;如果 RTI 需要回调盟员,按照通常的调用方式,RTI 也需要等待来自盟员的应答,这样盟员和 RTI 处于相互等待状态而发生死锁.因此,StarLink 要求盟员方必须使用多线程技术,一个线程调用 RTI,另外一个线程用来响应 RTI 对盟员的回调.根据 IEEE 1516 标准,消息按照传输类型分为可靠消息和非可靠消息.可靠消息要求 RTI 必须将一个消息从发送方可靠地传送给接收方;非可靠消息则要求 RTI 尽力地将一个消息从发送方传送给接收方.与之对应,CORBA 规范同样提供了这两种语义,即可靠语义和基于 oneway 语义的非可靠语义.StarLink 分别采用了如图 2(b)所示的基于 CORBA 的可靠消息传输方式和如图 2(c)所示的基于 oneway 语义的非可靠消息传输方式.其二是同时请求问题,在网络环境下,存在多个盟员同时调用 RTI 的情形,如果采用串行方式响应盟员的请求,则整个仿真的推进将会非常缓慢.为此,StarLink 必须解决好多个盟员同时请求 RTI 的技术难题.

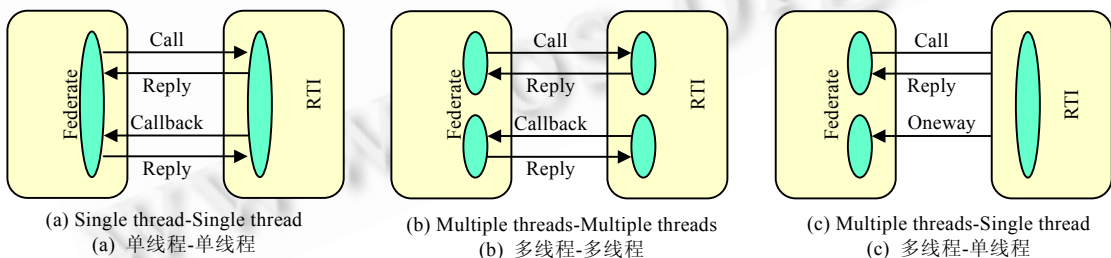


Fig.2 Communication deadlock resolution of YH-SRTI

图 2 YH-SRTI 中的通信死锁

为了解决多个盟员同时请求 RTI 的技术难题,StarLink 采用了如图 3 所示的多线程技术.主要包含 3 方面的内容:

- (1) 并行接收:RTI 使用多线程技术并行收集多个盟员所发送的数据.这样可以减少盟员等待接收的时间,从而为盟员处理其他事务赢得了时间.
- (2) 集中处理:使用单线程串行处理收集器所接收到的数据.使用单线程处理技术可以减少不必要的并发控制,简化了编程,相应地提高了系统的效率.

(3) 异步回调:StarLink 采用 StarBus 中间件所提供的异步回调方式将一个消息同时回调给多个感兴趣的盟员.在 StarBus 内部,异步回调方式是基于多线程技术的,但对 RTI 设计者而言是透明的,可看做是单线程方式.

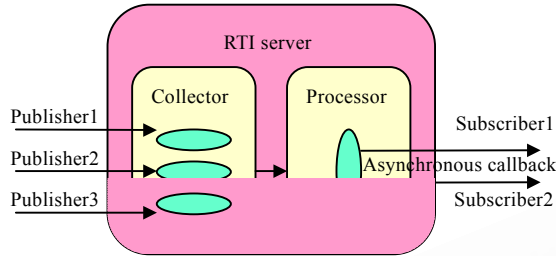


Fig.3 Concurrent model of StarLink

图3 StarLink 的并发模型

### 3.3 数据交互技术

按照 HLA 标准,在仿真中需要通过网络传输的所有数据都必须按照对象模型模板<sup>[3]</sup>的格式进行定义.传统的 RTI 几乎都是按照巴科斯范式 BNF(Backus-Naur form)描述的数据交互格式来定义 FED 文件中的仿真数据的,我们强调数据交互技术,是因为 StarLink 采用了一种完全不同的方式,即按照最新的 HLA 规范 IEEE 1516.2 中的要求,使用可扩展置标语言 XML(extensible markup language)来描述数据交互格式.在此基础上,StarLink 提供了专门的仿真数据开发工具,为用户建模和生成 StarLink 能够自动识别的 XML 格式的 FDD 文件提供了极大的方便.这些文件能够通过 IE 浏览器查看,为用户通过网络共享和传递信息提供了便利.

## 4 测试技术

一个软件系统可以从多方面、多层次的角度进行测试,我们从功能测试和规模测试两方面介绍 StarLink 的测试情况.

### 4.1 功能测试

所谓“功能测试”是指测试 StarLink 所提供的服务是否符合 IEEE 1516 标准.IEEE 1516 标准详细地描述了盟员能够调用的所有服务,以及每个服务的调用参数、返回值、使用该服务的前提条件、调用该服务之后应该满足的条件、调用该服务可能引起的各类异常等.

StarLink 需要按照 IEEE 1516 标准对盟员能够调用的每个服务进行测试(对回调服务不需要测试,因为回调服务是由盟员方编程提供的),那么是否应该为每次测试编写一个测试程序呢?为了节约测试时间,我们提出了基于“框架”结构的功能测试思想.具体方法是,使用一个通用的测试程序来测试 StarLink 的功能,支持任意的 XML 数据交互文件,支持任意多的测试盟员,支持任意的 RTI 服务的调度序列.

因为 HLA 本身就是基于“框架”结构的思想,与具体应用无关,可以集成各类仿真;而一个 RTI 用户应用程序总是通过调用一定的 RTI 服务序列来体现;对于测试程序而言,并不需要进行复杂的逻辑运算,所以设计一个通用的 RTI 测试程序是完全可能的.该程序的结构如图 4 所示,程序有两个分支,表示使用两个线程,一个用于调用 RTI,一个用于接收来自 RTI 的回调.对于每个调用服务,需要由测试者提供参数,当所有的参数都提供完之后,点击“调用 RTI 服务”按钮完成对 RTI 的调用;如果有返回值,则显示.通用测试程序对 RTI 回调服务的处理相当简单,只是把 RTI 的每个回调结果显示出来即可,因为我们关心的是对 RTI 调用服务的测试,而不是回调服务的测试.根据回调结果及 RTI 的测试目标,可进行下一次的服务调用.整个测试过程是一个不断循环的事件处理过程,通过通用测试程序能够实现对所有调用服务的功能测试.

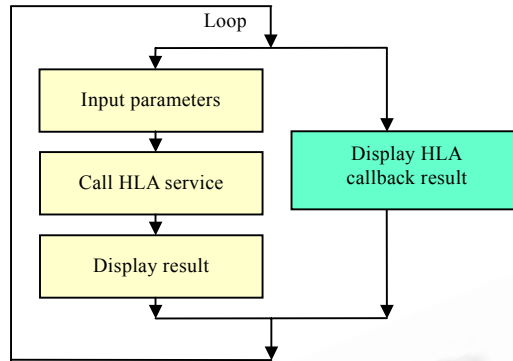


Fig.4 Framework test  
图4 功能测试框架

## 4.2 规模测试

所谓“规模测试”是指对 StarLink 能够支持多少个盟员的仿真进行测试.我们认为规模测试是一种含糊的、不确定的测试方法,因为一个 RTI 能够支持多大规模的仿真除了与使用的网络带宽、CPU 的性能、内存等客观因素有关以外,还与盟员推进的方式、推进的速度以及具体的仿真应用等因素相关.

StarLink 采用这样的规模测试方法:对  $n$  个盟员而言(盟员按  $1, 2, \dots, n$  编号),每个盟员向 RTI 公布一个不同的类(盟员  $i$  公布的类为  $C(i)$ ),则  $n$  个盟员将公布  $n$  个类( $C(1), C(2), \dots, C(n)$ );所有编号  $\leq n/2$  的盟员都要订购编号  $> n/2$  的盟员所公布的类,所有编号  $> n/2$  的盟员都要订购编号  $\leq n/2$  的盟员所公布的类.以  $n$  等于 100 为例,一共有 100 个公布类,编号为  $1 \sim 50$  的盟员需要订购 50 个类  $C(51), C(52), \dots, C(100)$ ;编号为  $51 \sim 100$  的盟员需要订购 50 个类  $C(1), C(2), \dots, C(50)$ .应当说,这种公布/订购方式需要传输的数据量是非常大的,对 StarLink 是个考验.

测试环境由 10M 以太网互联的 5 台主机组成,配置为 Windows 2000, Intel P4 1.8G, 256M 的主机 1 台; Windows 2000, Intel P4 1.5G, 256M 的主机两台; Windows 2000, Intel P4 2G, 1024M 的主机两台.所测得的结果如图 5 所示,图中横坐标表示盟员数,纵坐标表示 RTI 对盟员请求的响应时间.结果表明,由单个 RTI 服务器组成的 StarLink 适合中小规模的系统仿真,从而说明了由多个 RTI 服务器组成的层次式 RTI 一定能够支持更大规模的系统仿真.

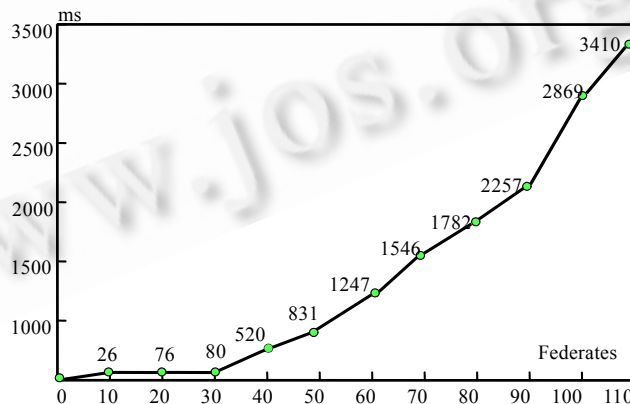


Fig.5 Scale test  
图5 规模测试

## 5 结 语

国防科学技术大学计算机学院在国家九五“863”<sup>[18]</sup>和十五“863”计划的支持下,于 1997 年即开始了

HLA/RTI 的研制工作.StarLink 是基于 CORBA 技术的国产 RTI 软件,具有许多独到之处.StarLink 采用层次式 RTI 体系结构,其目标是面向广域网环境下的大规模系统仿真.该软件遵循最新的 HLA 规范即 IEEE 1516 标准,采用了基于 CORBA 技术的集中控制和并行数据传输相结合的新型结构模式,既方便了 StarLink 的程序设计,减少了不必要的并发控制,又提高了系统的传输效率,加快了仿真推进的速度.为了便于仿真应用程序的开发,StarLink 采用了基于 XML 格式的数据交互技术并提供了相应的数据文件开发工具.另外还介绍了 StarLink 的功能测试和规模测试情况,提出了基于“框架”结构的功能测试思想.测试结果表明,StarLink 是一个功能全面的、实用的,能够适应广域网仿真的 RTI 软件.StarLink 为我们理解 HLA 技术和进一步开发应用功能更加强大的 RTI 软件积累了非常宝贵的经验,对于推动我国的 HLA 仿真技术的发展以及基于 HLA 的教学和科研工作也必将具有一定的重要意义.

#### References:

- [1] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules (IEEE Std 1516-2000). Institute of Electrical and Electronics Engineers, Inc., 2000.
- [2] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Federate Interface Specification (IEEE Std 1516.1-2000). Institute of Electrical and Electronics Engineers, Inc., 2001.
- [3] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Object Model Template (OMT) Specification (IEEE Std 1516.2-2000). Institute of Electrical and Electronics Engineers, Inc., 2001.
- [4] Software distribution center. 2002. <http://sdc.dmo.mil/>
- [5] Hao JG, Huang J. Implementation architecture of KD-RTI. *System Modeling & Simulation*, 2002,1(1):48~52.
- [6] Zhao QP. DVENET Distributed Virtual Environment. Beijing: Science Press, 2002. 39~87 (in Chinese).
- [7] Pullen M, Myjak M, Bouwens C. Limitations of Internet protocol suite for distributed simulation in the large multicast environment. 1999. <http://www.faqs.org/rfcs/rfc2502.html>
- [8] Furuichi M, Mizuno M, Miyata H. Performance evaluation model of HLA-RTI and evaluation result of eRTI. In: Proc. of the 1997 Fall Simulation Interoperability Workshop. 1997. 1099~1109.
- [9] pRTI runtime infrastructure for HLA simulations. 2002. <http://www.pitch.se/prti/default.asp>
- [10] MAK technologies. 2001. <http://www.mak.com/>
- [11] FDK—Federated simulations development kit. 2000. <http://www.cc.gatech.edu/computing/pads/fdk.html>
- [12] Yao YP, Lu XC, Liu BQ, Lu G, Han L. Design of data structure of a hierarchical RTI server. *Journal of Computer Research and Development*, 2002,39(Supplement):274~278 (in Chinese with English abstract).
- [13] CORBA. 2003. <http://www.corba.org/>
- [14] ACE and TAO success stories. 2002. <http://www.cs.wustl.edu/~schmidt/TAO-users.html>
- [15] Burks T, Alexander T, Lessmann K. Latency performance of various HLA RTI implementations. 2002. [http://www.mak.com/white\\_papers.htm](http://www.mak.com/white_papers.htm)
- [16] 2002. [http://www.863.org.cn/863\\_95/indust/ind24.html](http://www.863.org.cn/863_95/indust/ind24.html)
- [17] Zhou Y, Dai JW. HLA Simulation Programming. Beijing: Publishing House of Electronics Industry, 2002. 175~177 (in Chinese).
- [18] Yao YP, Shi XQ, Wan JH. Research and implementation of high level architecture. *Journal of System Simulation*, 2000,12(4): 364~366 (in Chinese with English abstract).

#### 附中文参考文献:

- [6] 赵沁平. DVENET 分布式虚拟环境. 北京: 科学出版社, 2002. 39~87.
- [12] 姚益平, 卢锡城, 刘步权, 卢刚, 韩林. 层次式 RTI 中数据结构的设计. *计算机研究与发展*, 2002, 39(增刊): 274~278.
- [17] 周彦, 戴剑伟. HLA 仿真程序设计. 北京: 电子工业出版社, 2002. 175~177.
- [18] 姚益平, 时向泉, 万江华. HLA/RTI 的研究与实现. *系统仿真学报*, 2000, 12(4): 364~366.