

# 一种支持私有云仿真的 HLA/RTI 实现方法

黄继杰,林昌年,杨选怀,王兰香,王国平  
(北京科东电力控制系统有限责任公司,北京 100192)

**摘要:**在云仿真中 HLA/RTI 采用 WSPRC 组件实现了联邦成员间信息的交互,这些交互信息大多由联邦成员通过访问模型服务所获得,HLA 还不能直接支持对模型服务的访问。虚拟化容器 Docker 的出现以及微服务较单体式软件架构的优势,使得部署在基于微服务构建的私有云服务器上的 HLA/RTI,具有了直接访问模型微服务的能力。文中在 OMT 中将对象类的属性名定义为所调用的微服务的参数名,通过为这些属性名添加上微服务标志的前缀,使 RTI 服务端组件可以识别出此对象类的实例在更新属性时,需要它先调用此对象类所指定的微服务,将微服务计算返回的结果作为新的更新值来发布。在扩展 RTI 服务端组件时先与数据库建立连接,再利用表中所记录的联邦执行名和实验次数及对象实例号构成三元组,并将此三元组唯一映射到调用模型微服务的租户号,从而使模型微服务可为指定的对象实例连续多次地服务,并记录每次服务的状态信息。通过将扩展后的 RTI 应用到全球能源互联网电力交易云仿真实验中,验证了 RTI 在识别出 OMT 中为对象类属性名添加的特定微服务标识后,还可以调用对象类的微服务来计算对象实例属性的更新并发布,使面向对象的 HLA/RTI 能更好地支持面向服务的仿真。

**关键词:**高层体系结构;云仿真;微服务;属性更新;发布

**中图分类号:**TP391.9

**文献标识码:**A

**文章编号:**1673-629X(2019)09-0164-05

**doi:**10.3969/j.issn.1673-629X.2019.09.031

## An HLA/RTI Implementation Supporting Private Cloud Simulation

HUANG Ji-jie, LIN Chang-nian, YANG Xuan-huai, WANG Lan-xiang, WANG Guo-ping,  
(Beijing Kedong Electric Control System Co., Ltd., Beijing 100192, China)

**Abstract:** In cloud simulation, HLA/RTI adopts WSPRC component to realize the interaction of information among federate members, which is mostly obtained by federated members through accessing model services. HLA cannot directly support the access to model services. With the emergence of virtual container Docker and the advantages of microservice over monolithic software architecture, HLA/RTI deployed on private cloud server based on microservice has the ability to directly access model microservice. In the OMT, the attribute name of the object class is defined as the parameter name of the called microservice. By adding the prefix of the microservice flag to these attribute names, the RTI server component can recognize that when the instance of this object class updates the attribute, it needs to first call the microservice specified by this object class and publish the result returned by the microservice calculation as a new update value. When extending the RTI server component, it first establishes a connection with the database, and then uses the federated execution name, experiment times and object instance number recorded in the table to form a triple which is uniquely mapped to the tenant number that calls the model microservice, so that the model microservice can serve multiple times for a specified object instance and record status information for each service. By applying the extended RTI to the global energy Internet power transaction cloud simulation experiment, it is verified that after identifying the specific micro-service identifier added to the object class attribute name in the OMT, the RTI can also call the micro-service of the object class to calculate attribute-update and publish object instance properties to enable object-oriented HLA/RTI to better support service-oriented simulation.

**Key words:** HLA; cloud simulation; microservice; attribute-update; publish

## 0 引言

分布式仿真标准 IEEE1516 基于面向对象技术提出了高层体系架构 (HLA)。为了适应面向服务的仿

真环境, IEEE1516-2010 版对 HLA 进行了许多扩展<sup>[1-2]</sup>, 并采用 WSPRC 组件 (Web services provider RTI component) 技术使 HLA/RTI 能支持 Web 环境下

收稿日期:2018-09-27

修回日期:2019-01-30

网络出版时间:2019-03-27

基金项目:国家电网公司总部科技项目(5222JZ17002R)

作者简介:黄继杰(1971-),男,博士,高级工程师,从事电力培训仿真及电力系统自动化工作。

网络出版地址:<http://kns.cnki.net/kcms/detail/61.1450.TP.20190327.1629.058.html>

的仿真<sup>[3-4]</sup>。云计算技术的蓬勃发展使仿真资源得到更高层次的共享,因而“云仿真”将是未来分布仿真的主要研究方向<sup>[5-6]</sup>。文献[7]研究了云仿真技术与 HLA Evolved 技术进行综合的问题,采用的技术手段还是对 HLA 进行扩展。

随着虚拟化技术的快速发展,出现了虚拟化容器技术(如 Docker<sup>[8-9]</sup>),并且使基于单体式软件架构的 SaaS 层云服务可拆分为许多微服务<sup>[10]</sup>。基于 Docker 和微服务来部署私有云<sup>[11]</sup>就可采用一台高性能的服务器,根据仿真任务的需要启动 Docker 容器,并将仿真模型微服务部署其中<sup>[12]</sup>。这种虚拟化和微服务技术使得 HLA 与面向服务技术能更好地结合。

HLA 的对象模板类(OMT)是其支持联邦成员交互的基础。文中先利用 OMT 的对象属性来定义仿真模型微服务的特殊标志及相关信息,并在 RTI 实现中,当遇到对象属性名为仿真模型微服务的特殊标志时,由 RTI 来负责完成对模型微服务的访问。

### 1 私有仿真云中部署微服务

随着虚拟化容器技术的兴起,产生了微服务,比如现在非常通用的 Docker 容器。Docker 与虚拟机不同,Docker 容器并不是一套硬件虚拟化方案,而是一个操作系统级别的虚拟化技术,仅为应用提供一套隔离的运行时虚拟环境,而不是整个虚拟的硬件资源。Docker 是以容器为资源分割和调度的基本单元和调度单位,封装了整个服务运行时的环境,用于构建、发布和运行分布式应用的平台。它使用 Linux 内核提供的 LXC 技术实现类似虚拟机的功能,以更节省的方式利用硬件资源,提供用户更多更高效的计算资源服务。

在 Docker 内部署的微服务改变了软件架构模式,使软件架构从单体式转向了微服务的架构模式。单体式软件架构也称为集中式软件体系架构,是指所有的功能在一个应用中,各个模块之间通过相互调用的方式交互。单体式软件架构具有以下不足:维护和扩展困难;难于持续部署;应用伸缩性差和可靠性低。因此,在云平台中这种单体式架构如今正开始被最新的微服务架构逐渐取代。

微服务软件架构被定义为:“开发单个服务组件作为一系列应用的套件,其中每个服务可以使用不同的编程语言实现,以及不同的存储技术独立运行在自己的进程空间中,围绕着业务功能而构建,可通过完全独立的自动化部署机制进行构建和部署,实现去中心化服务、集中式管理最小化,各个服务之间通过轻量级机制实现通信”。微服务具有以下特点:彼此独立;原子化;组合和重构。基于微服务开发的系统其所有的个体由于在概念上是对等的,结构相对模块化、简单

化,因此具有松散耦合的特点,构成的系统具有更强的可扩展性和鲁棒性。

### 2 OMT 支持微服务的策略

HLA 基于 OMT 来实现联邦成员间的数据交互,其中对象属性更新是最常用的方式。而对象属性的计算常常通过调用模型的接口来获得。由于微服务软件架构相比单体架构具有优势,用于计算对象属性的模型接口越来越多地采用微服务的接口。在私有云内的微服务环境下进行分布交互仿真一般采用图 1 左侧的方式,即联邦成员先调用私有云内的模型微服务计算对象属性,再将对象属性的更新值通过私有云内的 RTI 服务发布到其他联邦成员。

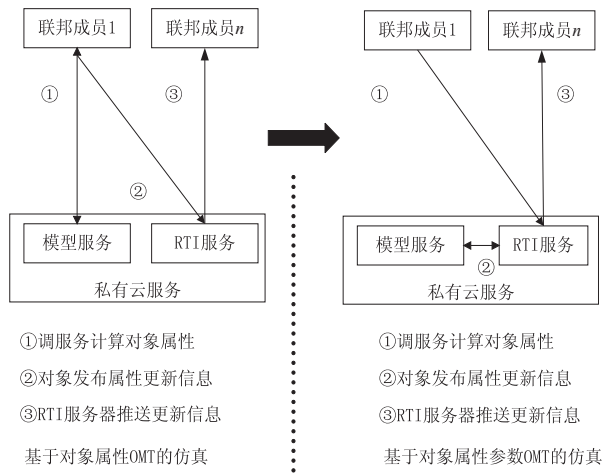


图 1 RTI 支持微服务的通信架构

当设计 FOM/SOM 时,将联邦成员计算对象属性的输入参数和输出参数加入到 OMT 中,以此代替该原对象需要更新的属性,其发布过程如图 1 右所示<sup>[13]</sup>。由于模型服务和 RTI 服务都位于私有云服务器,而服务器内部进程间的通信速度高于局域网中机器间的进程通信速度,故图 1 右所示的数据交互速度比左图所表示的更快。

设计 OMT 时可以像文献[14]一样,在 OMT 具体的 FED 文件中增加关键词来标示此对象需要 RTI 访问的微服务。考虑到与 IEEE1516 现有标准的一致性,利用对象类名加“\_microService”字段来通知 RTI,由它先调用微服务更新对象属性,再查阅订购者并将更新值发给它。以学生对象类(Class Student)为例<sup>[15]</sup>,假设它包含三个属性(LunchMoney, AmmoAmount 和 Cleanliness),并通过 CalculateStudent 函数来重新计算(LunchMoney, AmmoAmount 和 Cleanliness)这三个属性。现将 CalculateStudent 函数封装为微服务的接口,接口参数包括代表这三个属性的输入和输出,此外还包括微服务的地址。为此,将学生类对象(新类名为 Student\_microService)重定义为七个属性(addr\_Calcu-

lateStudent, in \_ LunchMoney, in \_ AmmoAmount, in \_ Cleanliness, out \_ LunchMoney, out \_ AmmoAmount, out \_ Cleanliness)。RTI 在调完微服务后将原来数据分发的处理量由 ( LunchMoney, AmmoAmount 和 Cleanliness ) 转向其中的 ( out \_ LunchMoney, out \_ AmmoAmount 和 out \_ Cleanliness ) 这三个属性,并可用这三个输出属性过滤基于 HLA 的区域。OMT 中命名统一规定为:前缀“addr\_”表示接口地址,“in\_”表示输入参数属性,“out\_”表示输出参数属性。

3 支持微服务的 RTI 实现方法

在开源 CERTI 源码的基础上进行 RTI 服务端的修改<sup>[16]</sup>,涉及到对象声明管理和对象管理两部分。RTI 服务端在处理对象声明管理时,需要嵌入对类名称的判断,若类名字后部包含“\_microService”,则该对象类的实例需要由 RTI 服务端调用微服务来计算各对象的属性并完成发布和订阅功能。处理流程如图 2 所示。

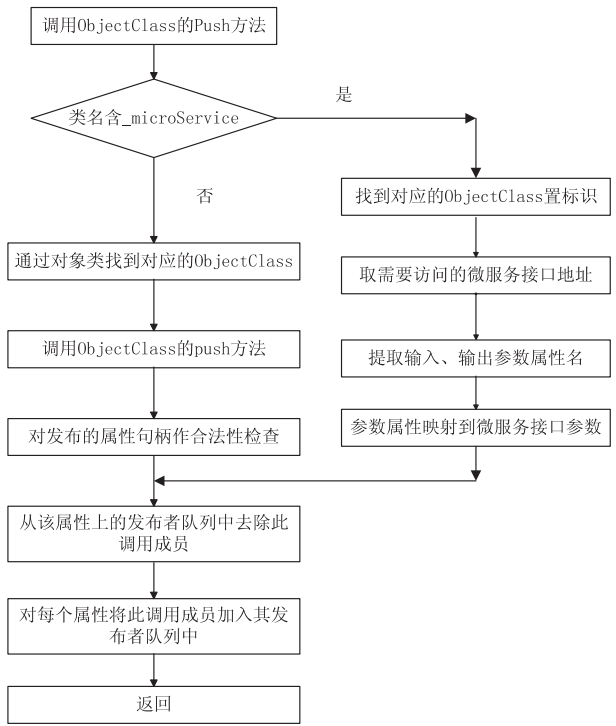


图 2 RTI 支持微服务的处理流程

图 2 中,当判断为“否”时,程序流程走的是原 RTI 服务处理流程;当判断为“是”时,先设置需调用微服务接口的标志,然后查看属性名的前缀是否包含“in\_”和“out\_”,若包含“in\_”为该属性的输入参数,若包含“out\_”为该属性的输出参数。对微服务接口地址参数,通过查看是否包含前缀“addr\_”来获得,如学生对对象类所调的微服务接口,其属性为:“addr \_ CalculateStudent”。

在后面的学生对象更新其属性时,属性“addr \_

CalculateStudent”所对应的值由微服务的配置文件和此接口的名构成,如“exchange. inf @ startPowerExchange”,其中 exchange. inf 是配置文件, startPowerExchange 是接口名。

RTI 实现中将属性映射为属性句柄,通过 ( 句柄值和属性值 ) 来打包传输数据;对于接口的参数名也需要与属性的句柄值相关联。为此,输入参数属性在 RTI 解析 FOM/SOM 时会将属性名映射为属性句柄,且 RTI 在分配属性句柄时是按自动递增顺序来计算句柄值的。故按微服务接口参数的顺序依次定义 FOM/SOM 中的访问该微服务的对象类的属性,则参数值、属性句柄值和属性值是一一对应的关系。RTI 在调用该接口时,先获得该接口的全部参数,并按对应的属性句柄大小进行属性值排队,从而按此顺序将属性值作为参数来调用对应的微服务接口。

同样,对于输出参数属性,同样需要此微服务接口在计算完后按输入参数的顺序进行数据打包。包格式为:4 字节参数个数,第一个参数字节长度,第一个参数内容,第二个参数长度,第二个参数内容,以此类推。当 RTI 服务端调用此微服务接口后,对获得的返回值进行上述格式的解包,并按输出参数属性句柄的大小依次赋值给各输出参数属性。

由于只有输出参数属性是需要在联邦间交互的值,故对于对象类的订购处理流程按 CERTI 的原代码处理,包括数据分发的区域过滤也按 CERTI 的原代码处理。

4 RTI 对多租户微服务的访问

当私有云中的微服务支持同一用户多次访问时,微服务可采用 SaaS 的多租户技术。多租户技术的实现重点在于不同租户间应用程序环境的隔离 ( application context isolation ) 以及数据的隔离 ( data isolation ),以维持不同租户间应用程序不会相互干扰,同时数据的保密性也够强。

与访问微服务的租户所对应的是 RTI 中的联邦执行号、联邦成员号和对象实例句柄号<sup>[17]</sup>。当 RTI 服务重启后,RTI 服务端对联邦执行号、联邦成员号和对象实例句柄号的赋值又会从 1 开始递增编号,故与微服务所用的租户号不能一一对应。但联邦执行的名字是唯一的,用户可多次运行此名字的联邦执行。用户每次创建该联邦执行,表示其要进行一次仿真实验,用实验次数与联邦执行的名字来唯一确定数据表中的一条记录。对于此次实验中的每一对象实例,用其实例句柄与上述记录号而得的二元信息,可唯一标识对象实例对微服务的访问,也即与租户号相对应了。所以可用三元组 [ 联邦执行的名字,实验次数号,对象实例

句柄号]来查询租户表中所对应的租户号。

为此,RTI 服务端在启动时需要连接上数据库,在创建联邦时以联邦执行的名字为索引到库表中查找记录,获得最大的联邦执行实验次数号,并加 1 作为新记录写入库表中,同时返回并得该记录的记录号 FID。当创建对象实例时,将记录号 FID 和该实例的句柄号及实例名一起写入另一库表,同时返回并得该记录的

记录号 TID,也即是租户 ID。

## 5 私有云中构建基于 RTI 扩展的实验

在构建跨国和跨洲的全球能源互联网云仿真实验中采用文中扩展的 RTI 功能<sup>[18]</sup>。全球能源互联网由跨洲、跨国骨干网架和各国各电压等级电网(输电网、配电网)构成,具体的硬件结构如图 3 所示。

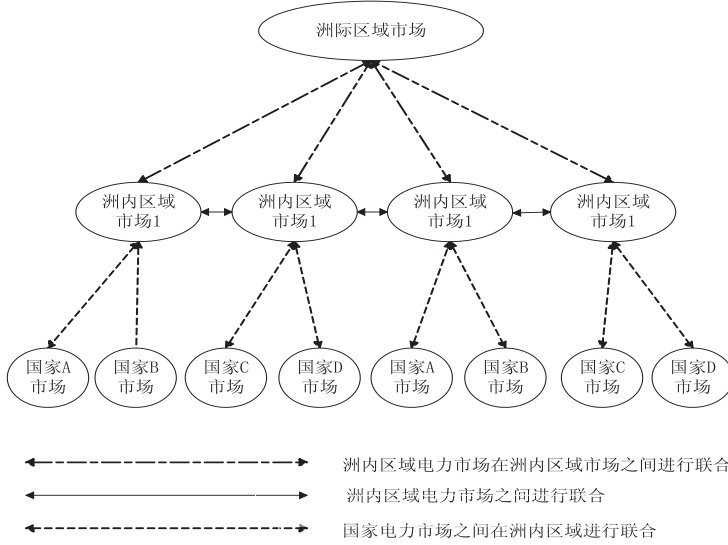


图 3 全球能源互联网电力交易图

为了保证系统实时运行时的电力平衡,需要进行实时平衡市场的交易。在实时平衡市场中,需求侧和供给侧交易主体根据自身交易需求情况,向所在辖区国的系统运营机构提交竞标的价格和数量,系统运营机构把每小时的报价按照价格排序,然后根据交易需求对每小时的系统进行匹配。在对各国进行电力交易仿真时,需要根据电网内所有负荷和所有发电机组的出力及其报价,以电网内所有用户购电费用最小为最优化目标,通过线性规划算法,最终给出每个发电厂的发电量和上网报价。

此实验采用微服务的设计架构,为此需要构建电力交易微服务。该服务需要设置每个节点的负荷  $L$ , 机组个数  $N_g$  及报价  $P_g$  和容量  $C_g$ , 支路  $(i, j)$  电抗  $X_{ij}$ , 最大传输有功功率  $PM_{ij}$ 。设决策变量机组  $g$  实际出力为  $U_g$ , 其线性规划模型为:

最优目标函数是购电费用最小:

$$\text{Min} \sum_{g=1}^{NG} P_g U_g \quad (1)$$

约束条件包括:

能量平衡约束:

$$\sum_{g=1}^{NG} U_g = \sum_{i=1}^N L_i \quad (2)$$

机组容量约束:

$$0 \leq U_g \leq C_g, g = 1, 2, \dots, NG \quad (3)$$

支路容量约束:

$$-P_{ij}^M \leq PL \leq P_{ij}^M, (i, j) \in G^L \quad (4)$$

其中支路容量约束又需要通过计算网络直流潮流的微服务来获得,电网直流潮流微服务计算在发电模块产生的能量(功率)、负荷模块消耗的能量(功率)及各条线路的电抗已知的情况下,可以计算节点模型间能量的流动。

$$P_i^{SP} = \sum_{j \in i, j \neq i} P_{ij} = \sum_{j \in i, j \neq i} \frac{\theta_i - \theta_j}{X_{ij}}, i = 1, 2, \dots, N \quad (5)$$

其中,  $\theta_i$  和  $\theta_j$  分别为线路两端的直流电位;  $X_{ij}$  (或电纳为  $B_{ij}$ ) 是线路的直流电阻。

$$P_{ij} = B_{ij}(\theta_i - \theta_j) = \frac{\theta_i - \theta_j}{X_{ij}} \quad (6)$$

用矩阵形式表示为:

$$[B_0][\theta] = [P] \quad (7)$$

其中,  $B_0$  为正常运行时的网络节点电纳矩阵;  $\theta$  为节点电压相位角的向量;  $P$  为节点注入的有功功率向量。当  $B_0$  和  $P$  已知时,用上式可以容易地求得各节点电压相位角,进一步可算出各支路的有功功率潮流。

电力交易微服务<sup>[19]</sup>和直流潮流计算微服务基于 ZeroC 公司的 ICE 实现。RTI 服务端组件在启动时连接 MySQL 数据库,在对象实例访问微服务时,创建 ICE 的连接,如下:

Ice::CommunicatorHolder



```

    ich( argcEnv,argvEnv," exchange. inf" );
    auto twoway=Ice::checkedCast<ExchangPrx>( communicator
->propertyToProxy( " Exchang. Proxy" )->ice_
secure( false) );

```

```

    UgStr = twoway -> startPowerExchange ( L , Ng , Pg , Cg ,
Xij , PMij , TenantID );

```

由于电力交易微服务需要获得参与仿真的各国的输入参数,包括线路阻抗和最大容量,电力网络各节点的申报负荷量和申报报价,各节点机组的申报发电量和申报价格,而后才能计算。所以各国需要在两个仿真步长内调用两次交易对象的属性更新接口。在第一步长内,将参数送电力交易微服务,由微服务将参数值写入库表,并将返回各国对象实例的参数置为空;在第二步长内,各国交易对象实例更新参数时,电力交易微服务通过查库表可知该值已存在,则表明前面第一次的参数已全部获得,故启动最优化计算,并将计算结果写入数据库表中。

RTI 服务端组件调用电力交易微服务接口,该接口直接从数据库中读出该对象的信息并返回,RTI 服务端组件通过解析返回的数据流,获得对象实例的实际出力  $U_g$  和最优报价  $P_g$  并发布到各国。发出更新请求的对象实例也将收到自己的最优报价和最优发电量。而之前对象实例更新的输入参数是各国期望的价格和发电量,还不能保证实时运行时的电力平衡,只有 RTI 服务端组件通过调用微服务而得到的最优报价和最优发电量才能满足实时运行时的电力平衡。

## 6 结束语

RTI 的服务端组件可与微服务共同部署到私有云服务器上,使得联邦成员在更新对象实例前访问私有云中计算微服务的方式,转变为由 RTI 服务端来访问云中的微服务,减少了数据在局域网内的传输次数。RTI 服务端组件通过与数据库相连,将联邦执行名、实验次数和对象实例与微服务的租户号相对应,使 RTI 能调用微服务计算出对象实例的属性更新值并发布出去,从而拓展了 RTI 与模型服务的结合方式。文中利用 IEEE1516 的 OMT 来定义对象类属性与微服务接口参数的对应关系,若在 OMT 中直接定义与微服务接口相关的信息会使 HLA 与服务化更深度融合,这将是 IEEE1516 继续发展的一个方向。

## 参考文献:

- [1] 钟 蔚,龚建兴,郝建国,等. HLA Evolved 规范研究分析[J]. 系统仿真学报,2011,23(4):691-696.
- [2] RUKIYE C, VEYSI I. Applying RESTful web services architecture to HLA based simulation applications[C]//Fall sim-

ulation interoperability workshop. Orlando, USA: [ s. n. ], 2012:187-196.

- [3] 张 涛,史璐莎,张 斌,等. 一种快速扩展 HLA /RTI Web 服务的方法[J]. 计算机技术与发展,2016,26(12):49-52.
- [4] MICHAEL E, JOHNNY J G. Next generation modeling and simulation architecture 11F-SIW-068 [ C ]//Fall simulation interoperability workshop. Orlando, USA: [ s. n. ], 2011:448-456.
- [5] ZHANG Z H, CHAI X D, HOU B C. System security approach for web-enabled HLA/RTI in the cloud simulation environment [ C ]//6th IEEE conference on industrial electronics and applications. Beijing, China: IEEE, 2011:245-248.
- [6] ANTHONY C, TOM V D B, WIM H. Data farming in support of HLA performance assessment [ C ]//Fall simulation interoperability workshop. Orlando, USA: [ s. n. ], 2014:317-328.
- [7] 高武奇,康凤举,钟联炯,等. 一种基于 HLA Evolved 的云仿真技术研究[J]. 系统仿真学报,2011,23(8):1643-1647.
- [8] 浙江大学 SEL 实验室. Docker:容器与容器云[M]. 北京:人民邮电出版社,2015.
- [9] 赵乐乐,黄 刚,马 越. 基于 Docker 的 Hadoop 平台架构研究[J]. 计算机技术与发展,2016,26(9):99-103.
- [10] 郝庭毅,吴 恒,吴国全,等. 面向微服务架构的容器级弹性资源供给方法[J]. 计算机研究与发展,2017,54(3):597-608.
- [11] 王纪军,张 斌,顾永生,等. 云环境中 Web 应用的微服务架构评估[J]. 计算机系统应用,2017,26(5):9-15.
- [12] 张 晶,黄小锋,李春阳. 微服务框架的设计与实现[J]. 计算机系统应用,2017,26(6):259-262.
- [13] 张智慧,李伯虎,柴旭东,等. 一种面向共享内存环境的高性能 HLA/RTI 原型系统[J]. 系统仿真学报,2014,26(2):315-322.
- [14] 岳英超,肖田元,范文慧,等. 基于多视图—嵌套本体映射的高层体系结构语义互操作[J]. 计算机集成制造系统,2012,18(5):932-942.
- [15] 周 彦. HLA 仿真程序设计[M]. 北京:电子工业出版社,2002.
- [16] 张德刚,王达达,李泽河,等. 基于计算机协同工作的高层体系结构组播扩展[J]. 计算机技术与发展,2018,28(6):188-191.
- [17] 王 琼,艾丽蓉,龚爱珍. 基于 HLA 的分布式仿真中负载均衡的研究[J]. 计算机技术与发展,2011,21(6):119-121.
- [18] 杨选怀,林昌年,王国平,等. 基于高层体系结构的电力系统综合仿真平台[J]. 电网技术,2009,33(11):98-103.
- [19] 承 林,王海宁,高春成. 微服务在电力交易系统中的应用研究[J]. 电网技术,2018,42(2):441-446.