

6-23-2021

Design of LVC Simulation Test Middleware for SaaS

Du Nan

Military Exercise and Training Center, Army Academy of Armored Forces, Beijing 100072, China;

Yaxin Tan

Military Exercise and Training Center, Army Academy of Armored Forces, Beijing 100072, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Design of LVC Simulation Test Middleware for SaaS

Abstract

Abstract: LVC (Live-Virtual-Constructive) technology provides a new technical means for equipment test. In view of the large number of heterogeneous simulation resource objects and complex interaction in the LVC system test, which can not meet the test task requirements of quick response of the weapon system, *the middleware and application model framework is reformed according to the remote method invocation based on TENA. The functional modules of LVC simulation middleware including model running component, publishing and ordering component, message processing component and real-time running component are designed to provide efficient real-time communication mechanism for LVC simulation and support the application mode of simulation as a service (SaaS).* The middleware can realize operation management services such as rapid scheduling, monitoring, distributing, deploying, operation controlling, data collecting and other complex resources of LVC test. It also provides technical supports with test task analysis, process monitoring, exception handling, combat weapons or various platforms cooperating for LVC test.

Keywords

Live-Virtual-Constructive (LVC), Simulation as a Service (SaaS), equipment test, middleware

Recommended Citation

Du Nan, Tan Yaxin. Design of LVC Simulation Test Middleware for SaaS[J]. Journal of System Simulation, 2021, 33(6): 1268-1276.

面向 SaaS 的 LVC 仿真试验中间件设计研究

杜楠, 谭亚新

(陆军装甲兵学院 演训中心, 北京 100072)

摘要: 真实-虚拟-构造 (Live-Virtual-Constructive, LVC) 仿真技术为装备试验提供了全新的技术手段。针对 LVC 体系试验中异构仿真资源对象数量多、交互复杂导致无法满足武器体系快速响应的试验任务需求, 以训练使能体系架构为基础, 按照远程方法调用为主的模式对中间件、应用模型框架进行改造。设计 LVC 仿真中间件的模型运行组件、发布订购组件、消息处理组件、实时运行组件各个功能模块, 为 LVC 仿真提供高效的实时通信机制, 支撑仿真即服务 (Simulation as a Service, SaaS) 的运用模式。该中间件可实现 LVC 试验复杂资源的快速调度、监控、分发、部署、运行控制、数据采集等运行管理服务, 为 LVC 试验任务分析、过程监控、异常处理及作战武器之间、各类平台之间的协同提供技术支撑。

关键词: 真实-虚拟-构造; 仿真即服务; 装备试验; 中间件

中图分类号: E919; TP391

文献标志码: A

文章编号: 1004-731X (2021) 06-1268-09

DOI: 10.16182/j.issn1004731x.joss.21-0399

Design of LVC Simulation Test Middleware for SaaS

Du Nan, Tan Yaxin

(Military Exercise and Training Center, Army Academy of Armored Forces, Beijing 100072, China)

Abstract: LVC (Live-Virtual-Constructive) technology provides a new technical means for equipment test. In view of the large number of heterogeneous simulation resource objects and complex interaction in the LVC system test, which can not meet the test task requirements of quick response of the weapon system, *the middleware and application model framework is reformed according to the remote method invocation based on TENA. The functional modules of LVC simulation middleware including model running component, publishing and ordering component, message processing component and real-time running component are designed to provide efficient real-time communication mechanism for LVC simulation and support the application mode of simulation as a service (SaaS).* The middleware can realize operation management services such as rapid scheduling, monitoring, distributing, deploying, operation controlling, data collecting and other complex resources of LVC test. It also provides technical supports with test task analysis, process monitoring, exception handling, combat weapons or various platforms cooperating for LVC test.

Keywords: Live-Virtual-Constructive (LVC); Simulation as a Service (SaaS); equipment test; middleware

引言

真实-虚拟-构造 (Live-Virtual-Constructive, LVC) 仿真技术的出现为装备试验提供了全新的技术手段。中间件作为一种在试验运行期间为试验对

象、试验资源提供实时通信的系统, 用以解决异构系统间的互联、互通、互操作问题, 是 LVC 试验支撑框架的核心子系统^[1-3]。目前, 国内外对 LVC 仿真技术展开了一些研究, 国外试验模式正在由以实物试验为主的模式向虚实结合的联合试验方向

发展,并形成了一系列标准规范包括高层体系结构(HLA, High Level Architecture)、基本对象模型(BOM)、试验与训练使能体系结构(TENA, Test and Training Enabling Architecture)、模型驱动体系结构(MDA, Model Driven Architecture)等^[4]。这些标准规范应用在 LVC 仿真通信上使用的都是发布/订阅模式,可用于在实现单元和订阅单元之间分发数据^[5-7]。LVC 体系试验中存在大量交互复杂的异构仿真资源对象,当模型之间通信量大时会占用较大的网络带宽,而且当某个模型的实现逻辑被多次调用时,增加了代码冗余性,因此无法满足武器体系快速响应的试验任务需求^[8-9]。本文以训练使能体系架构为基础开发了 LVC 仿真试验中间件,按照远程方法调用为主的模式对中间件、应用模型框架进行改造,体现仿真即服务(Simulation as a Service, SaaS)理念,在即插即用的仿真环境中实现异构资源最优分配^[10-12]。

1 总体架构

中间件总体架构如图 1 所示,包括模型运行组件、发布/订阅组件、消息处理组件、实时运行组件,提供资源动态监控、时间管理等服务。模型运行组件负责维护事件通道、通知服务及命名服务,维护模型间的名字映射关系及通信链路,保证模型的正常运行;发布/订阅组件负责数据的发布和订阅;消息处理组件负责消息的发送和回调;实时运行组件负责维护实时的时钟,保证通信的实时性。

2 模型运行组件

模型运行组件分为试验域管理模块和命名服务模块,如图 2 所示。试验域管理模块主要为整个通信提供支持,需要绑定到命名服务,然后创建通知通道,应用程序加入到试验域后,可以获得通知通道的发布者代理、订阅者代理。命名服务模块将模型仓库和通知通道的对象绑定,建立名字与对象引用之间的映射关系,应用程序可直接通过命名服务实现各应用间的通信。

2.1 命名服务模块

为了区别不同的试验域,命名服务模块提供了不同的试验域名称。维护命名树子模块的功能是维护运行中间件各类实体层次关系及其对应对象。当应用程序加入某个试验域时,可以通过试验域的名称来定位该试验域,从而加入该试验域。在某个试验域内部,可以通过试验对象名称来定位和解析试验对象。绑定名字子模块解析获取根命名,并使用对象请求代理提供的命名上下文的方法将各类对象的名字加入到命名服务中。解析名字子模块即使用命名服务,获取名字对应的对象。

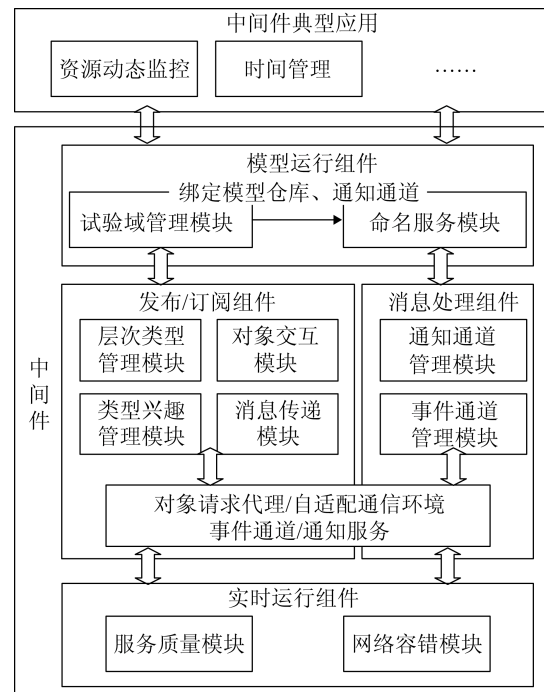


图 1 中间件架构

Fig. 1 Architecture of middleware

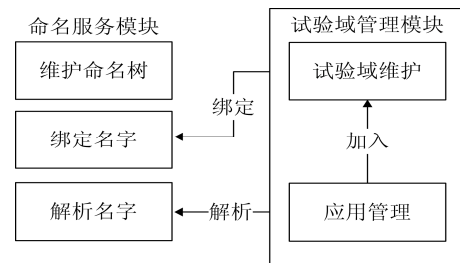


图 2 模型运行组件

Fig. 2 Model running component

2.2 试验域管理模块

试验域管理模块的功能包含试验域创建和销毁, 以及试验域内应用加入和退出。随着试验域的增加, 需要合理的管理试验域, 试验域管理模块针对不同的场景使用系统配置模块中的域管理配置模块配置不同的试验域, 从而创建不同的试验域。当需要运行多个试验域时, 只需要配置不同的参数启动多个域管理即可。每个域管理与相应的试验域对应, 每个域管理会创建自己的通知通道, 该试验域中事件的订阅和发布都在该试验域中进行, 而不会在其他域管理的试验域中进行。试验域维护包括试验域的创建、销毁。试验域的创建即创建试验域, 在每个试验域中创建本试验域的通知通道供试验应用加入。试验域的销毁即通过调用试验域的析构函数来实现试验域的销毁, 并调用通知通道的销毁操作, 使得属于本试验域的试验应用全部退出, 并且试验应用不能再加入该试验域。应用管理包括应用加入、退出。应用加入主要通过查找试验域集合来完成, 即将某应用加入到对应的试验域中, 以试验域名称以及应用程序需要设置的服务质量(Quality of Service, QoS)为参数调用加入试验域函数。应用退出即调用退出(exit)方法使得试验域应用退出对应的试验域, 最后一个退出的应用负责将该试验域清除出试验域集合。

3 发布/订阅组件

发布/订阅组件主要负责维护整个分布式系统中的对象交互, 保证对象的正确交互和基于类型等形式的兴趣管理, 并提供远程方法调用功能, 分为层次类型管理模块、类型兴趣管理模块、对象交互模块、消息传递子模块^[13-14]。如图 3 所示, 层次类型管理模块包括类型仓库、对象仓库; 类型兴趣管理模块包括过滤器工厂、过滤器、过滤器表达式集; 对象交互模块包括状态更新、对象订阅、代理及回调队列管理。消息传递模块包括消息更新、消息订阅、代理及回调队列管理。层次类型

管理首先查询想定文件(XML 格式), 初始化类型仓库, 对象交互需要查询层次类型管理模块以获取标识(Identity, ID)。

3.1 层次类型管理模块

层次类型管理模块负责管理发布者、订阅者间的类层次关系, 为试验系统中的各类订阅提供支持。为了实现发布者和订阅者类型和对象的一致性, 层次类型管理模块采用仓库的思想, 通过维护类型仓库和对象仓库来实现发布者和订阅者的类型和对象的关联, 保证类型和对象在试验域内的全局唯一性。在一个试验域内, 类型仓库和对象仓库供试验域全局。类型仓库将类型标识符映射成一个全局唯一的数字作为相应的 ID, 并维护类型层次关系, 同时支持反查。使用 2 张表分别来记录虚拟化状态对象类型和消息类型的名字到 ID 的映射, 其中类型标识符作为字符串主要用于接口参数的传递, 而类型 ID 则在实现内部使用。为实现类型标识符到类型 ID 的一一映射, 每一种类型都需要事先被注册到类型仓库中。注册一种类型, 就意味着对这种类型分配一个库中唯一的 ID。类型仓库同时还提供通过名字查询 ID 的方法, 用于返回类型仓库中已注册的类型的 ID。另外它维护一个继承关系表, 使用邻接链表的方式保存。表中每个元素代表一种类型, 元素的内容是一张记录该类型所有子类类型的表。继承关系需要通过解析配置文件写入模型仓库。对象仓库将对象标识符映射成一个全局唯一的数字作为相应的 ID, 并支持反查。使用 2 张表分别来记录虚拟化状态对象和消息对象的名字到 ID 的映射, 其中对象标识符作为字符串主要用于接口参数的传递, 而对象 ID 则在实现内部使用。为实现对象标识符到对象 ID 的一一映射, 每一种对象都需要事先被注册到对象仓库中。注册一个对象, 就意味着对这个对象分配一个库中唯一的 ID。对象仓库同时还提供通过名字查询 ID 的方法, 用于返回对象仓库中已注册的对象 ID。

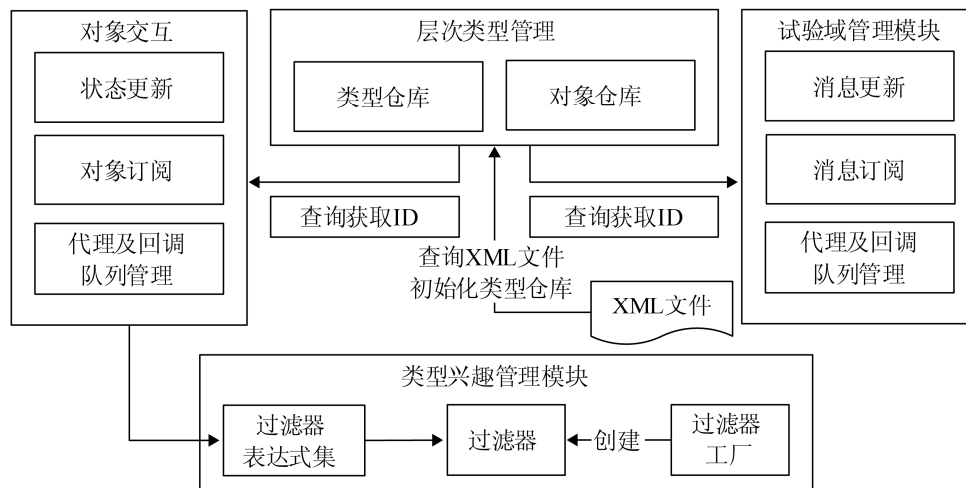


图 3 发布/订阅组件

Fig. 3 Publish/Subscribe component

进行类型订阅时, 将 QoS 的 ID 设置成为任意类型, 也就是订阅该类型的所有实例。当运行中间件接收到数据时, 对数据实例反查, 并将该实例名封装到原始数据类中, 最终将原始数据类数据提交给虚拟化对象模型。原始数据类中封装了实例名以及实例的数据。在发布/订阅中, 如果发布/订阅的类型存在继承关系, 那么当订阅父类类型时, 其子类数据也应该订阅到。发布/订阅的继承关系首先在想定文件中写好, 然后启动试验域管理服务时, 域管理会解析想定文件, 将类型继承关系记录在模型仓库中。当订阅端订阅父类数据时, 首先查看模型仓库该类型是否有继承关系, 如果有, 则修改 QoS 将子类的类型添加到 QoS 中; 否则不修改 QoS, 最后重新连接通知通道。

3.2 类型兴趣管理模块

类型兴趣管理模块管理应用程序的发布、订阅兴趣。应用程序通过通知通道或事件通道, 可以发布虚拟化状态对象消息实例以及订阅虚拟化状态对象消息类型或实例。通知服务支持 OMG 定义的过滤语言(Extended Trader Constraint Language, ETCL)。ETCL 允许应用程序创建复杂的表达式来描述事件, 因此它可以使接收者发布精确的订阅兴趣, 提高数据处理效率。兴趣管理模块通过过滤器

工厂、过滤器以及过滤表达式集实现。

过滤器工厂通过解析通知通道获取此通知通道的默认过滤器工厂并用于创建过滤器。过滤器是实现事件过滤的关键。通过在对象请求代理通知服务的各个层次对象添加过滤器实现对事件的层次过滤, 使过滤功能更强大。例如, 当应用程序订阅一种类型或发布一种类型实例时, 通过过滤器工厂为此类型创建相应的过滤器, 并将此过滤器添加至各层次对象。过滤表达式集是具体描述哪些事件可以通过过滤器的约束集, 将其添加至过滤器, 过滤器通过对比过滤表达式与事件的本体决定此事件是否满足应用程序的发布或订阅兴趣。当应用程序订阅或发布一种实例时, 首先判断此实例的类型, 查询此类型的过滤器是否创建, 若已创建, 则根据此实例名创建过滤表达式, 并添加至过滤表达式集; 若没有创建, 则首先根据此实例的类型创建过滤器, 并将过滤器添加至对象, 然后根据此实例创建过滤表达式, 并将其添加至此过滤器。

3.3 对象交互模块

该模块主要提供虚拟化状态对象的发布和订阅功能, 负责对象的发布/订阅、更新和维护。对象交互通过交互模块中的状态更新、虚拟化状态对象订阅及虚拟化状态对象队列管理完成。状态更新

子模块主要用于发布者发布更新虚拟化状态对象，其功能为连接通知通道或事件通道、将对象发布状态封装成结构事件后推送到通知通道或事件通道中，进而完成整个状态更新的全过程。

连接通知通道过程使用自适应通信环境/对象请求代理(ACE/TAO)通用的连接通知通道方法封装在生产伺服器类的构造中。获取通知通道或事件通道的订阅者代理后，该类就拥有了向通知通道发布结构事件(Structured Event)的能力，通过构造该类实例并调用其更新方法，就可以完成对象的更新。对象订阅主要针对订阅者，用于向通知通道或事件通道表达订阅虚拟化状态对象兴趣。

对象订阅模块的功能为连接通知通道或事件通道、将用户的订阅兴趣转化为通知通道或事件通道中标准格式，进而完成整个对象订阅兴趣提交的过程。连接事件通道过程使用自适应通信环境/对象请求代理通用的连接事件通道方法封装在接收者代理类的构造中。获取通知通道或事件通道的生产者代理后，该类就拥有了向通知通道或事件通道表达订阅兴趣的能力，通过构造该类实例并调用该类订阅虚拟化状态对象兴趣的方法，就可以完成向通知通道或事件通道表达订阅兴趣的过程。

代理及回调队列管理模块主要负责表达兴趣后，将收到的对象数据缓存并解析出来。订阅不同的虚拟化状态对象需要构造不同的代理类的实例，在表达完虚拟化状态对象的订阅兴趣后，需要向中间件注册与该虚拟化状态对象关联的代理对象，等待中间件提交订阅到的虚拟化状态对象。由于订阅的虚拟化状态对象与回调代理对象具有对应关系，代理可以将收到的虚拟化状态对象正确的解析出来并添加到各自的队列中去。时间对象作为一个特殊的订阅，只保存当前最新的。用户在业务逻辑中需要何种对象，可以到这种类型对应的代理队列中取出对象处理。

3.4 消息传递模块

消息传递模块提供消息发现与消息传递功能，

包括消息更新、消息订阅、代理及回调队列管理等功能。消息更新为发布者功能，主要用于发布更新消息。

消息更新模块的功能为连接通知通道或事件通道、将对象消息封装成结构事件后推送到通知通道或事件通道中，进而完成整个消息更新的全过程。连接事件通道过程使用自适应通信环境/对象请求代理通用的连接事件通道方法封装在生产伺服器类的构造中。获取通知通道的订阅者代理后，该类就拥有了向通知通道或事件通道发布结构事件的能力，通过构造该类实例并调用其更新方法完成消息更新。

消息订阅主要为订阅者功能，主要用于向通知通道或事件通道表达订阅消息兴趣。消息订阅模块的功能为连接通知通道或事件通道，将用户的消息订阅兴趣转化为通知通道中标准格式，进而完成整个消息订阅兴趣提交的过程。连接事件通道过程使用自适应通信环境/对象请求代理通用的连接事件通道方法封装在接收者代理类的构造中。获取通知通道或事件通道的生产者代理后，该类就拥有了向通知通道或事件通道表达订阅兴趣的能力，通过构造该类实例并调用表达订阅消息兴趣方法完成向通知通道表达消息订阅兴趣。

代理及回调队列管理模块主要负责将表达消息兴趣后，将收到的对象数据解析出来。订阅不同的消息对象需要构造不同的代理类的实例，在表达完消息的订阅兴趣后，需要向中间件注册与该消息关联的代理对象。中间件将不同的消息转交给相应的回调代理对象，因为订阅的消息与回调代理对象一一对应，所以代理可以将收到的消息正确的解析出来并添加到各自的队列中去。特别是消息作为一种限定性的对象交互，代理中的缓存队列长度为1，即只保存最新的消息。远程方法调用区别于对象交互、消息传递的处理，提供及时、可控的远程通信。

4 消息处理组件

消息处理组件负责试验对象进程、代理进程、

监控进程之间的信息传递、交互、管理, 包括通知通道管理模块和事件通道管理模块, 分别负责创建与维护通知通道、事件通道, 供试验应用程序传输数据。

4.1 通知通道管理模块

通知通道管理的模块功能包括通知通道维护、订阅者管理者获取、发布者管理者获取三部分。通知通道维护主要包括创建、销毁等操作。通知通道创建中间件运行的通信平台, 供试验域应用间交互, 既是信息提供者又是信息订阅者的插入对象, 支持多个“发布者”与“订阅者”之间异步通信, 且允许多个信息提供者和多个信息订阅者异步通信, 实现了发布者和订阅者之间的解耦合。订阅者管理者获取调用“通知服务的管理获取函数”获取负责事件订阅者连接通知的通道, 从而实现结构事件的订阅。发布者管理者获取使用“通知服务的发布者管理获取函数”获取负责事件发布者连接通知通道, 从而实现结构事件的发布。

4.2 事件通道管理模块

事件通道管理模块负责实现事件通道间的相互通信, 每个节点创建一个发送者子模块、一个接收者子模块、一个事件头子模块, 通过组播的方式实现。发送者子模块作为订阅者接收本地事件并组播发送, 接收者子模块通过事件头子模块接收对组播中感兴趣的事件。多事件通道间的数据传输也是通过组播实现的, 并按试验域进行分组, 不同试验域内的事件通道无法实现数据传输。

5 实时运行组件

LVC 仿真中间件系统可提供通知通道和多事件通道两种数据传输方式, 为确保数据传输的实时性、可靠性, 实时组件为通知通道、多事件通道提供可靠 QoS 传输服务。

5.1 服务质量模块

图 4 为 QoS 层次模型, 根据此层次模型, 设

计实现 QoS 需要实现 QoS 参数映射机制等。对于具有 QoS 驱动的应用, QoS 交互过程为: 用户通过通知通道 QoS 设置的接口函数设置应用的 QoS。可设置的 QoS 参数包括网络服务类型、数据超时时间、数据传输优先级、数据排队策略和丢弃策略、通知通道最大发布者和订阅者个数等。

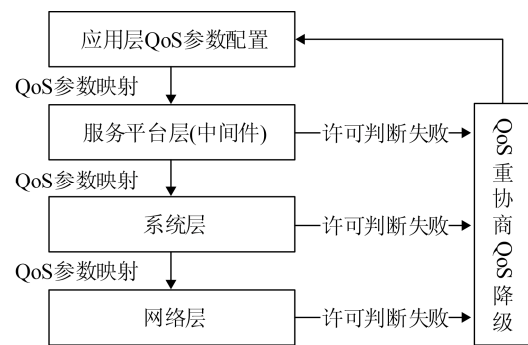


图 4 服务质量模块

Fig. 4 Quality of Service component

5.2 网络容错模块

网络容错模块通过对对象请求代理服务质量的参数选项、包括 TCP/IP 协议服务质量的研究, 提供短、中、长 3 种断线情况的处理。短和中通过对相应服务质量的支持提供重传处理机制, 即在可靠的服务质量中增加断点重传数量、在发布订阅端增加缓冲数量, 达到网络容错的目的。长的情况通过系统监控和预警方式处理, 即通过仿真模型和系统在控制流和数据流方面解耦, 心跳检测与处理机制, 使得节点断线、异常退出等现象能够被监控并得到很好的处理, 不影响整个系统仿真进程。

在系统使用过程中, 可能会出现通讯失败、对象无效、对象不存在、瞬态故障、超时等异常。通讯失败, 指客户的请求被接受, 但是在处理请求时发生了故障; 对象无效, 指客户端试图调用一个无效的对象引用; 对象不存在, 指客户端尝试对不存在的对象执行调用操作; 瞬态故障, 指请求无法到达目的地, 因为执行请求的关键资源(服务器、链接)不可用; 超时, 指请求不能在规定时间内完成。

以上类型故障将以事件形式反馈给开发者, 通

过设置事件通道的相关控制策略进行断线重连操作及错误反馈。在事件通道中,通过设置参数选择断线时采取的控制策略(无响应,被动响应,重连)。在采取重连措施时,事件通道将依照之前配置重新尝试连接。同时可以通过更改轮询周期以选择轮询的频率,通过设置往返超时来确定超时异常的触发条件。若多次重连未取得效果,系统长期处于断线状态,则上报错误由用户进行判断。对象请求代理通过保持一个公共数据表示缓冲区的链表来最小化不必要的复制。通过优化该操作参数以实现内存开销与复制操作的平衡。同时,缓冲区的大小会影响断线时的数据存储。

6 典型应用

6.1 资源动态监控

资源动态监控为整个虚拟试验场景提供了一个实时监控的功能,包括监控数据获取和监控界面,如图 5 所示。虚拟化监控对象(Virtualization Monitor Object, VMO)是资源动态监控模块的重要组成部分,它用于监测试验应用的信息,并将监测的所有信息发布给试验监控管理系统。VMO 包括对象静态信息收集与发布、模型继承信息收集与发布、对象动态信息收集与发布、程序状态信息收集与发布。监控数据获取模块负责收集 VMO 的信息,包括对象静态信息、模型继承信息、对象动态信息、程序状态信息。对象静态信息收集与发布试验对象静态信息收集与发布,负责收集试验域应用的相关模型信息。模型继承信息收集与发布负责收集模型的继承信息并通过继承信息发布者发送。对象动态信息收集与发布负责收集试验运行动态信息并统计。

6.2 时间管理

时间管理用来保证整个试验能够正确有序的进行。时间推进机制分为时间管理客户端和时间管理服务端 2 个部分,如图 6 所示。时间管理客户端分为时间推进、时间策略管理、消息队列管理、数

据传输服务 4 个子模块。时间管理服务端分为时间推进、时间策略管理、全局时间管理 3 个子模块,用于协调管理整个系统的时间推进,负责对时间管理客户端发来的设置时间受限请求、时间控制请求进行响应^[15]。

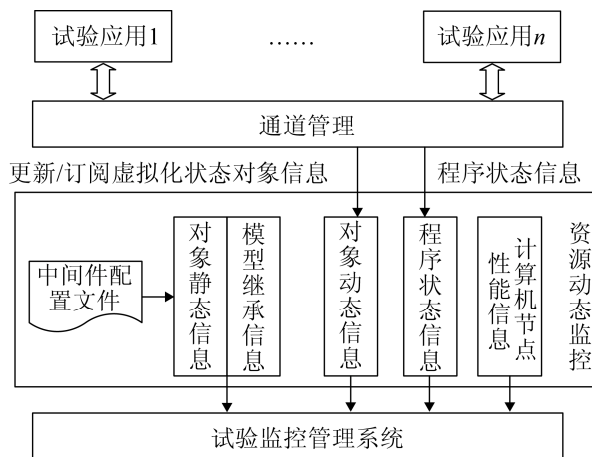


图 5 资源动态监控
Fig. 5 Dynamic resource monitoring

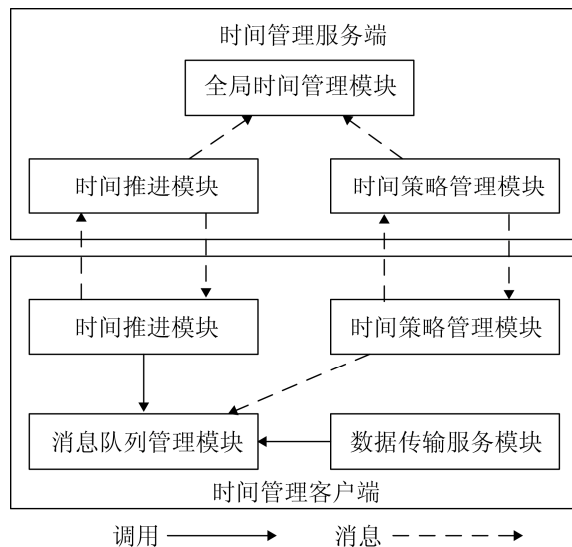


图 6 时间管理
Fig. 6 Time management

时间推进模块响应应用程序时间推进请求。时间管理客户端循环接收消息,判断消息类型,将其他虚拟化对象模型运行节点发过来的消息直接存入消息队列供应用程序获取。时间策略管理模块设置时间管理客户端的时间策略。消息队列管理模块为每一个虚拟化对象模型节点增加消息队列、消息

队列里的消息按照时戳从小到大排列, 同时提供对消息队列的维护和访问功能。存储消息回调, 使消息可以按一定的顺序发送给应用程序, 保证消息的正确处理。数据传输服务模块接收消息, 根据消息的类型做出相应的处理。为了使蓝方仿真平台支持带时戳的消息处理, 需在设计消息结构时, 考虑时戳信息。

时间推进模块统一管理所有时间管理客户端运行节点的时间推进。蓝方仿真平台可以实现与异构系统的互联、互通、互操作, 这就需要时间管理实现蓝方仿真平台与异构系统的时间推进同步。时间管理服务端循环等待时间管理客户端发送的请求消息, 判断蓝方仿真平台是否需要与 HLA 同步, 并响应时间管理客户端的时间推进请求, 并向异构网关发送蓝方仿真平台的时间推进请求。

7 结论

针对 LVC 试验中存在大量异构的仿真资源对象, 传统对象模型无法满足信息化战争对武器体系快速响应的试验任务需求问题, 以训练使能体系架构为基础, 以仿真即服务为理念, 按照远程方法调用为主的模式对中间件、应用模型框架进行改造, 设计了 LVC 仿真中间件的模型运行组件、发布订购组件、消息处理组件、实时运行组件各个功能模块的设计方案, 通过远程方法通信实现 LVC 仿模型之间的互联互通, 为 LVC 仿真提供高效的实时通信机制。该方法可实现 LVC 试验复杂资源的快速调度、监控、分发、部署、运行控制、数据采集等运行管理服务, 为 LVC 试验任务分析、过程监控、异常处理及作战武器之间、各类平台之间的协同提供技术支撑。

参考文献:

- [1] 吴金平, 陆铭华, 薛昌友. 潜艇作战系统 LVC 一体化仿真设计与引擎实现[J/OL]. [2021-04-25] 系统仿真学报, <https://doi.org/10.16182/j.issn1004731x.joss.20-0231>. Wu Jinping, Lu Minghua, Xue Cangyou. Design and Engine Implementation of Submarine Combat System Simulation Based on LVC[J/OL]. [2021-04-25] Journal of System Simulation. <https://doi.org/10.16182/j.issn1004-731x.joss.20-0231>.
- [2] 董志华, 张小伟, 朱元昌. 基于网关的 HLA-TENA 互联研究[J]. 火炮发射与控制学报, 2014, 35(4): 83-86. Dong Zhihua, Zhang Xiaowei, Zhu Yuanchang. Research on the Inter-connection Between HLA-TENA Based on Gateway[J]. Journal of Gun Launch & Control, 2014, 35(4): 83-86.
- [3] 蔡继红, 卿杜政, 谢宝娣. 支持 LVC 互操作的分布式联合仿真技术研究[J]. 系统仿真学报, 2015, 27(1): 93-97. Cai Jihong, Qing Duzheng, Xie Baodi. Research of Joint Simulation Platform Supporting Interoperability of LVC[J]. Journal of System Simulation, 2015, 27(1): 93-97.
- [4] 彭彰, 李钦富. 一种基于 TENA 的应急指挥模拟演练平台架构设计[J]. 系统仿真学报, 2021, 33(1): 231-238. Peng Zhang, Li Qinfu. Architecture Design of an Emergency Command Simulation Exercise Platform Based on TENA[J]. Journal of System Simulation, 2021, 33(1): 231-238.
- [5] 彭健, 赵雯, 章乐平, 等. 虚拟试验支撑框架 VITA 研究与实现[J]. 计算机测量与控制, 2017, 25(8): 289-293. Peng Jian, Zhao Wen, Zhang Leping, et al. Research and Implementation of Virtual Test and Evaluation Enabling Architecture[J]. Computer Measurement & Control, 2017, 25(8): 289-293.
- [6] 董志明, 郭齐胜. 试验训练一体化仿真支撑技术研究[J]. 计算机仿真, 2021, 38(2): 1-3, 23. Dong Zhiming, Guo Qisheng. Research on Support Technology of Test and Training Integrated Simulation[J]. Computer Simulation, 2021, 38(2): 1-3, 23.
- [7] 郭齐胜, 董志明. 试验训练一体化仿真支撑平台设计研究_郭齐胜[J]. 计算机仿真, 2020, 37(9): 1-5. Guo Qisheng, Dong Zhiming. Research on Design of Test and Training Integrated Simulation Supporting Platform[J]. Computer Simulation, 2020, 37(9): 1-5.
- [8] 涂亿彬. LVC 联合试验体系结构及关键技术研究[D]. 长沙: 国防科技大学, 2016. Tu Yibin. Research on LVC Joint Test Architecture and Key Technology[D]. Changsha: National University of Defense Technology, 2016.
- [9] 王西宝. 武器装备联合仿真试验关键技术研究[D]. 长沙: 国防科技大学, 2017. Wang Xibao. Research on the Key Technologies of Joint Simulation and Experiment on Weapon Equipment[D].

- Changsha: National University of Defense Technology, 2017.
- [10] 张国强, 刘小荷, 蒋方婷, 等. 反导体系试训一体化环境开发集成方法研究[J]. 系统仿真学报, 2016, 28(4): 898-906.
- Zhang Guoqiang, Liu Xiaohe, Jiang Fangting, et al. Research on Development and Integration Method of BMDS Integrated Test and Training Environment[J]. Journal of System Simulation, 2016, 28(4): 898-906.
- [11] 罗永亮, 张琚, 熊玉平, 等. 支持LVC仿真的航空指挥和保障异构系统集成技术[J]. 系统仿真学报, 2017, 29(10): 2538-2541.
- Luo Yongliang, Zhang Jun, Xiong Yuping, et al. Air Command and Support Heterogeneous Systems Integration Technology Supporting LVC Simulation[J]. Journal of System Simulation, 2017, 29(10): 2538-2541.
- [12] 张筠彬, 董志明. 基于仿真即服务的作战试验仿真系统设计[J]. 兵器装备工程学报, 2020, 41(2): 72-74.
- Zhang Yubin, Dong Zhiming. Design of Operational Test Simulation System Based on SaaS[J]. Journal of Ordnance Equipment Engineering, 2020, 41(2): 72-74.
- [13] 王晓路, 贾长伟, 刘闻, 等. 体系级LVC仿真集成技术研究[C]//2020中国仿真大会论文集. 北京: 中国仿真学会, 2020: 358-364.
- Wang Xiaolu, Jia Changwei, Liu Wen, et al. Research on System Level LVC Simulation Integration Technology[C]//2020 China Simulation Conference Proceedings. Beijing: China Simulation Federation, 2020: 358-364.
- [14] 冯润明, 王国玉, 黄柯棣. TENA中间件的设计与实现[J]. 系统仿真学报, 2004, 16(11): 2373-2377.
- Feng Runming, Wang Guoyu, Huang Kedi. The Design and Implementation of TENA Middleware[J]. Journal of System Simulation, 2004, 16(11): 2373-2377.
- [15] 冯琦琦, 董志明, 贾长伟, 等. 面向LVC仿真的多层分级时间管理方法研究[J]. 计算机仿真, 2020, 37(12): 1-4, 11.
- Feng Qiqi, Dong Zhiming, Jia Changwei, et al. Research on Multi-level Time Management Method for LVC Simulation[J]. Computer Simulation, 2020, 37(12): 1-4, 11.