

1-3-2019

Edge-side Simulation Method and Framework Based on Micro-services

Yanlong Zhai

1.School of Computer Science, Beijing Institute of Technology, Beijing 100081, China; ;

Wenxin Sun

1.School of Computer Science, Beijing Institute of Technology, Beijing 100081, China; ;

Tianhong Bao

1.School of Computer Science, Beijing Institute of Technology, Beijing 100081, China; ;

Yang Kai

2.Science and Technology on Special System Simulation Laboratory Beijing Simulation Center, Beijing 100854, China;

See next page for additional authors

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.

Edge-side Simulation Method and Framework Based on Micro-services

Abstract

Abstract: Aiming at the multi-devices, big data and low-latency simulation requirements in the Internet of Things, we define a concept of edge-side simulation and a simulation mode based on micro-services referring to the computational characteristics and architecture of edge computing and fog computing. For the link constraint and high-latency processing defects of "cloud simulation", a lightweight edge-side simulation platform framework combining edge computing and fog computing is designed, and the simulation speed is further accelerated by real-time simulation with LVC simulation system. Compared with the existing simulation frameworks, the edge-side simulation framework has the remarkable features of lightweight, high real-time, service, and organic integration with the Internet of Things, and can improve the collaboration, sharing, resource management and other capabilities of the simulation application from various aspects.

Keywords

LVC simulation, edge computing, fog computing, simulation and emulation

Authors

Yanlong Zhai, Wenxin Sun, Tianhong Bao, Yang Kai, and Duzheng Qing

Recommended Citation

Zhai Yanlong, Sun Wenxin, Bao Tianhong, Yang Kai, Qing Duzheng. Edge-side Simulation Method and Framework Based on Micro-services[J]. Journal of System Simulation, 2018, 30(12): 4536-4545.

基于微服务的边缘侧仿真方法及框架研究

翟岩龙¹, 孙文心¹, 包天虹¹, 杨凯², 卿杜政²

(1.北京理工大学计算机学院, 北京 100081; 2.北京仿真中心 航天系统仿真重点实验室, 北京 100854)

摘要: 针对物联网中多设备、大数据、低延迟的仿真需求, 参照边缘计算和雾计算的计算特点和架构, 定义了基于微服务的边缘侧仿真的概念及仿真模式。针对“云仿真”的链路束缚和高延迟处理缺陷, 设计了融合边缘计算和雾计算的轻量级边缘侧仿真平台框架, 并结合LVC仿真体系进一步地加快仿真速度实现实时仿真。与现有的仿真框架相比, 该边缘侧仿真框架具有轻量化、高实时性、服务化、与物联网有机融合等显著特点, 可以从多方面提高仿真应用的协作、共享、资源管理等能力。

关键词: LVC仿真; 边缘计算; 雾计算; 建模与仿真

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X(2018)12-4536-10

DOI: 10.16182/j.issn1004731x.joss.201812006

Edge-side Simulation Method and Framework Based on Micro-services

Zhai Yanlong¹, Sun Wenxin¹, Bao Tianhong¹, Yang Kai², Qing Duzheng²

(1.School of Computer Science, Beijing Institute of Technology, Beijing 100081, China;

2.Science and Technology on Special System Simulation Laboratory Beijing Simulation Center, Beijing 100854, China)

Abstract: Aiming at the multi-devices, big data and low-latency simulation requirements in the Internet of Things, we define a concept of edge-side simulation and a simulation mode based on micro-services referring to the computational characteristics and architecture of edge computing and fog computing. For the link constraint and high-latency processing defects of "cloud simulation", a lightweight edge-side simulation platform framework combining edge computing and fog computing is designed, and the simulation speed is further accelerated by real-time simulation with LVC simulation system. Compared with the existing simulation frameworks, the edge-side simulation framework has the remarkable features of lightweight, high real-time, service, and organic integration with the Internet of Things, and can improve the collaboration, sharing, resource management and other capabilities of the simulation application from various aspects.

Keywords: LVC simulation; edge computing; fog computing; simulation and emulation

引言

随着建模与仿真技术在多个学科中的不断深

入应用, 仿真平台越来越数字化、网络化、智能化、集成化、虚拟化、协同化^[1]。建模与仿真技术也一直随着信息技术的发展而不断变化, 从最初的集中式仿真, 逐渐发展了分布式仿真、并行仿真、基于Web的仿真、云仿真等不同的仿真模式。目前的系统仿真技术越来越贴近真实系统, 嵌入式仿真、并行仿真、采用实装模型的仿真往往可以提供更好的



收稿日期: 2018-05-30 修回日期: 2018-07-01;
基金项目: 国家自然科学基金(61602037), 装备预研领域基金(61400010104);
作者简介: 翟岩龙(1981-), 男, 北京, 博士, 讲师, 研究方向为分布式并行仿真等; 孙文心(1995-), 女, 河南, 硕士, 研究方向为分布式并行仿真。

<http://www.china-simulation.com>

• 4536 •

可信度和逼真度。如何将地域上分布的实装模型, 甚至是硬件装备有效的纳入到仿真系统, 并提供高实时性的系统仿真是目前面临的重要挑战。

云仿真技术为复杂系统仿真应用提供了强大的计算平台, 但是随着物联网时代的到来, 普遍存在的传感器, 如作战仿真中分布式存在的各种传感器, 以及网络边缘节点越来越多地加入到仿真中。如果将传感器和网络边缘节点的数据和计算全部发送到云仿真平台进行计算和处理, 这从网络带宽和计算实时性方面都无法满足仿真系统的要求。边缘计算和雾计算很好地弥补了云计算在数据传输和网络边缘实时计算方面的短板。

雾计算是将计算能力放置在网络边缘的新型计算方式, 具有云计算的优点, 也具有终端就近处理的优点^[2], 减少数据传输时延, 满足时间敏感型应用程序的计算延迟需求。边缘计算是指网络终端设备之间相互感知相互提供计算服务的计算。与传统基于云计算的仿真平台相比, 边缘侧仿真能够通过边缘侧节点之间的协作和计算减少对云仿真平台的远程访问和数据传输, 形成网络边缘侧的局部实时仿真环境, 同时也能够将越来越多的传感器和仿真设备纳入到仿真应用中。可以看出, 在物联网时代边缘侧仿真与云仿真结合的方式是实现实时、快速、大规模复杂仿真的必然趋势。

由于边缘计算和雾计算目前也处理理论研究阶段, 虽然已经建立了参考模型和理论体系, 但是还没有已经成熟的边缘计算和雾计算框架。将边缘计算和雾计算与仿真技术结合的研究目前处于起步阶段。因此很有必要研究系统仿真技术与边缘计算技术进行结合的内涵、挑战和系统架构。本文针对边缘侧仿真具体需求, 剖析了边缘计算雾计算与系统仿真融合涵义和面临的关键技术挑战, 设计了基于微服务架构的边缘侧仿真参考架构。针对边缘侧仿真面临的技术挑战, 提出了初步的解决方法。该边缘侧仿真框架具有轻量化、高实时性、服务化、与物联网有机融合等显著特点。

1 研究背景

1.1 建模与仿真

建模与仿真技术认识和改造客观世界的重要手段。随着系统仿真技术的应用不断广泛深入, 仿真规模、复杂度、数据量都在扩大。现在需要一个具有分布、异构、协同、互操作、重用等性能的新型分布建模仿真系统, 可以通过网络为人们随时随地无障碍地提供建模仿真服务^[3]。计算机技术在系统仿真中起到了决定性的支撑作用, 云仿真和网络仿真技术为实现上述要求提供的必备的支撑作用。

LVC(Live, Virtual and Constructive)仿真是一种“虚实结合”的重要仿真方法。实况(Live)仿真是指真实的人员在虚拟环境下操作真实的设备和系统。虚拟(Virtual)仿真是指真实的人员操作仿真器或者模拟器。构造(Constructive)仿真是指虚拟的人员操作计算机构造生成的虚拟的设备和系统^[4]。LVC 仿真将虚拟的和真实的仿真模型结合在一起, 这使得仿真应用的可信度大大提高, 因此很多重要领域, 如航天、军事等都越来越多的采用 LVC 仿真模式。在物联网时代 LVC 仿真将面临着种类和数量众多的传感器和真实装备, 传统 LVC 解决方案更多的是关注互操作性, 普遍采用比较重的中间件体系架构, 例如 TENA。这种架构更多的是需要像云一样的计算环境, 不适合大量传感器和轻量级设备的物联网环境。

1.2 边缘/雾计算

边缘计算与雾计算理念的产生是为了解决云计算中出现的, 大数据量高频率传输导致的网络拥塞和远距离传输导致的高网络延迟问题^[5]。雾计算在网络的边缘放置边缘节点, 为边缘设备提供有吸引力的计算能力、存储容量和网络服务^[6], 减少边缘设备的应用程序处理时间, 增强服务质量(Quality of Service, QoS)。边缘计算与雾计算类似, 但是着重强调了边缘设备间资源利用的相互协同。可以看出边缘计算和雾计算模式充分利用了网络

边缘的计算和存储能力,提高了边缘设备和传感器参与计算的程度,以及云计算在边缘侧的处理能力。将雾计算和边缘计算结合起来将是非常有前景的计算模式。

通过对 LVC 仿真和边缘计算模式的分析可以看出,边缘计算和雾计算的模式能够很好的与物联网环境融合,更好地支持大量异构的轻量级设备。将云仿真和边缘侧仿真方法结合起来能够很好的支持和实现 LVC 仿真系统。

2 边缘侧仿真的内涵和应用模式

边缘侧仿真是边缘计算和雾计算与仿真技术融合形成的新型仿真模式,是云仿真向网络边缘拓展的结果,属于系统建模与仿真支撑系统技术,为仿真实验提供高效的计算方式。边缘侧仿真因为充分利用网络边缘的资源而具备比较高的实时性,可用于实现嵌入仿真和实时仿真,这使得边缘侧仿真进一步实现普适仿真技术的目标,无缝的融合由计算和通信构成的信息空间与人们生活的物理空间^[7]。因此我们也可以将边缘侧仿真定义为融合了云计算和物联网技术的新型建模仿真技术,通过充分利用网络边缘处的网络设备、传感器、武器装备等边缘节点的资源,在网络边缘侧实现局部的高实时性仿真,减少非必要的与云中心的低效交互。

边缘节点不仅可以帮助边缘设备分担计算任务,或将计算任务上传至云中心。边缘节点还可以接收并存储边缘设备从云中心下载的计算数据^[8],按需传输至边缘设备,节省边缘设备存储资源。同时若有其他边缘设备申请同样数据时不必从云中下载,可以直接从边缘节点中获取^[9]。该情况针对于仿真应用程序与数据不在同一的设备的情况,比如作战系统仿真中,坦克请求无人机拍摄的视频资源时,可以从边缘节点(陆战车)获取,减轻无人机的存储压力和无人机通信链路的请求压力。

边缘侧仿真的内涵实际上是融合了云仿真和

边缘节点仿真的完整概念。这正如计算机系统技术由 20 世纪四五十年代的集中式服务器,发展到 21 世纪初分散式的互联网,然后到集中式的云计算,再到现在集中和分散结合在一起的“云+边缘”模式,系统仿真技术也经历了十分相似的发展历程。随着计算机技术的发展规律可以看出,“云+边缘”的仿真模式也将是接下来一段时间最有发展的仿真模式。尤其是随着大数据、人工智能、深度学习等也逐渐从云端开始下移到网络边缘,边缘侧复杂系统仿真未来可期。

依照上述对边缘侧仿真的内涵分析,设计了如图 1 所示的边缘侧仿真模式。边缘侧仿真的体系架构整体上可以分为三个层次,顶层的云仿真环境,中间层的雾仿真环境以及底层的边缘仿真环境。其中云仿真环境由于具有足够的存储和计算资源,适合运行大规模复杂计算,例如一些由计算机生成的复杂装备计算模型(Constructive)。雾仿真环境具备比一般边缘节点更多的资源和功能,可以用于处理不同仿真域的模式计算和仿真任务。边缘仿真环境直接连接传感器和各种实际装备或者模拟器(Live and Virtual),可以对传感器和设备的数据进行初步的处理和计算,完成本地仿真。如果边缘设备无法完成本地的仿真任务,它可以通过设备感知服务找到临近的具备能力的边缘设备(边缘计算)或是边缘节点(雾计算),然后通过协商机制建立协作模式。最后通过工作负载共享服务完成计算任务的转移。在需要的时候,边缘设备和边缘节点可以向云中心发送必须的数据或者服务请求。

从图 1 中可以看出,在边缘设备和边缘节点两层之间形成了局部的高实时性仿真环境,由云中心和边缘侧节点构成了分布式的 LVC 仿真。系统中产生的数据从边缘层经过处理和过滤后逐层传输到云中心,然后经过云中心计算分析后产生智能信息和具体的操作指令再传递到边缘层。

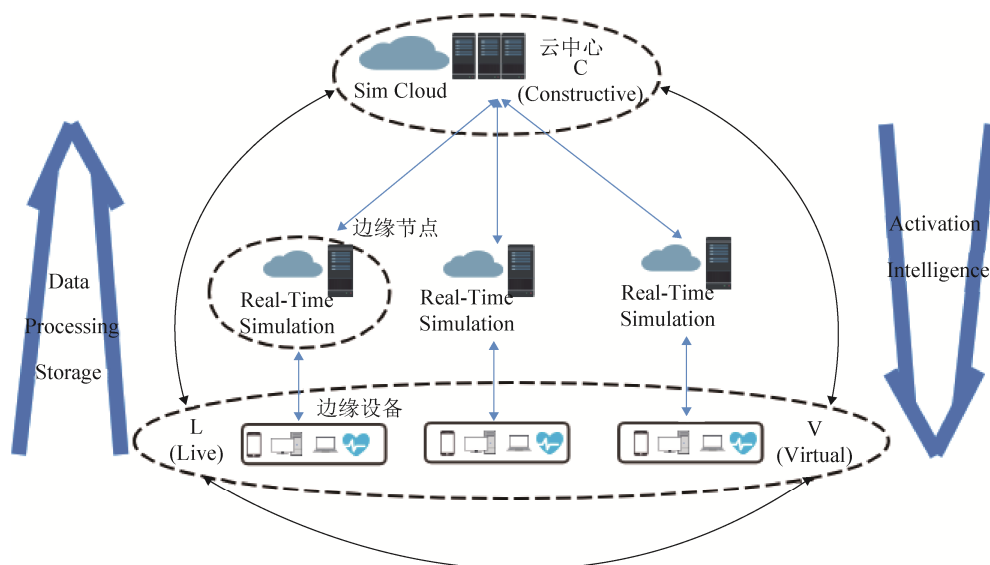


图 1 边缘侧 LVC 仿真模式
Fig. 1 Edge-side simulation method of LVC

边缘侧仿真平台是一种利用网络、边缘计算和云计算方式的仿真系统,其仿真操作流程如下:1)在边缘节点连接部署边缘设备或模拟的数字边缘设备;2)在对应边缘设备上部署相应的应用程序;3)仿真开始,监控服务监控应用程序运行情况,若直至程序运行完成,都没有违反规定的 QoS,则跳转至 1 进行下一任务的处理。反之,跳转到 4;4)当违反规定的 QoS 时,使用策略服务和设备感知服务寻找最合适的运行节点。5)工作负载共享服务将该程序转移至 4 中得出的运行节点。6)程序运行结束后,跳转至 1)进行下一任务的处理。

3 典型边缘侧仿真场景

本节将以智能战场仿真应用为典型案例,分析如何采用边缘侧仿真方法进行建模和仿真,通过该典型仿真场景说明边缘侧仿真在实时性和物联网设备支持方面的优势。

军事技术是一个国家实力的体现,提高军事技术可以在自然灾害前减少人员伤亡,可以在恐怖主义前捍卫和平,而不仅仅是与另一个国家开展战争,争夺资源。智能战场的仿真模型采用网络中心战^[10],通过作战设备的联网实现信息全面感知,

加快决策速度提高作战计划的准确性加大杀伤力提高作战能力。典型作战仿真模型需要实现物理域、信息域和认知域的建模和仿真。作战仿真模型使用基于为服务的边缘侧仿真平台仿真时,物理域对应于仿真战场中的无人机、坦克、陆战车和战士,其中陆战车为边缘节点其他硬件设备则为边缘设备。信息域和认知域主要负责数据处理和资源智能化管理,使用边缘侧计算形式实现,进一步加快信息共享和作战指挥速度。

智能战场作战仿真应用的场景如图 2 所示。无人机侦察战场,仿真时由模拟器模拟。无人机接收陆战车发来的航线信息和飞行动作信息侦察战场,通过摄像机收集战场视频信息,同时收集自身的位置信息和飞行状态信息。坦克识别敌方发动攻击,通过传感器收集到的附近作战设备发出的频率,判别敌我信息,并配合无人机的视频信息找到合适的角度攻击敌方。陆战车接收各个设备的数据,分析数据发出指令指挥作战。陆战车是距离战场较近的移动边缘节点,收集各个边缘武器设备的信息,定制合适的战略并将策略信息发送回边缘作战设备。同时监控战士的生命特征、位置和请求等信息,使人员伤亡降到最低。

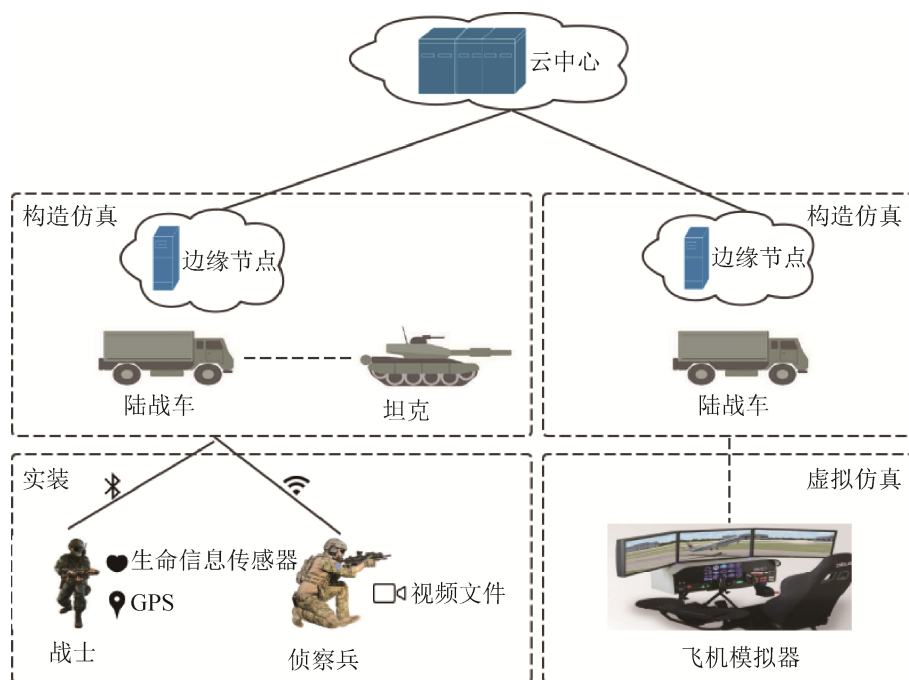


图2 基于边缘侧仿真的智能作战仿真场景

Fig. 2 Intelligent combat simulation scenario with edge-side simulation

战士佩戴多种传感器收集数据接收指令。战士佩戴的传感器分为两种，一种监控生命特征，一种监控战场信息和位置信息，确保伤员可以尽快发现，尽早接受治疗。同时，也可以使陆战车内的操作人员更加了解战争形势，做出正确的战略决定。

该仿真场景采用了基于云中心和边缘侧节点的仿真环境，包含实装传感器采集战场上实时的信息，一些模型则通过模拟生成在边缘节点运行，还有一些是通过模拟器以边缘设备的方式接入仿真。与典型 LVC 仿真的区别在于，仿真模型和设备彼此可以直接通信，并不需要通过中心节点进行中转。此外很多轻量级的设备也可以以仿真节点的形式直接参与到仿真中，例如场景里的生命信息传感器。边缘侧仿真并不需要安装重量级的中间件系统，而是可以再有限资源的边缘节点上直接通过微服务进行仿真。

4 边缘侧仿真的系统需求和挑战

边缘侧仿真平台的实现首先需要满足基本的仿真平台功能：边缘侧仿真应用部署、边缘侧仿真

模型互操作方法、数据分发和时间管理。其次需要满足边缘计算理念和雾计算理念：工作负载共享。最后，根据文献[11]，边缘侧仿真平台还需实现架构可扩展、具有基础软硬件仿真环境以及实现 LVC 仿真。

4.1 边缘侧仿真应用部署方法

应用程序部署作为仿真平台的输入端口，被要求为用户留下部署程序的接口。作为边缘侧仿真平台的输入层，应用程序部署端口还需要应用程序分割工具和工作流转换工具。

区别于云仿真平台的应用程序集中式传输集中式处理，边缘侧仿真平台需要将同一应用程序在不同的计算节点运行，计算节点包括边缘设备本地和与边缘设备连接的任一边缘节点。所以边缘侧仿真平台需要为用户提供应用程序分割工具，同时解决程序分割时线程的数据相关、指令相关和控制相关。在目前的研究中，致力于实现程序的自动分割，例如文献[12]将程序按照数据依赖程度进行程序分块，或是文献[13]将程序按照对硬件的依赖分块，但是研究还处于理论阶段，使用人工分割程序

是目前较为可行的方法。用户将分割好的应用程序进行工作流转换, 按照执行顺序构成有向无环图^[14], 端点表示应用程序边表示依赖关系。

应用程序部署由仿真系统用户进行, 平台真正需要实现的是应用部署接口的预留和识别用户部署的工作流, 实现难点为应用程序的分割工具的实现和用户构造的有向无环图的识别。

4.2 边缘侧仿真模型互操作方法

在边缘侧仿真平台中进行仿真操作时, 需要在设备间建立通信进行数据传输。设备间的通信建立属于设备感知需求, 传输内容为交流信息组, 建立连接后即可传输计算数据以及分析感知周围设备的计算资源和存储资源, 找到合适的程序工作负载转移位置, 使计算延迟最小并减少能耗。为了满足边缘侧仿真平台的可扩展性, 设备通信中设备感知功能还需要满足架构可扩展。

由于边缘侧仿真平台建立在无线物联网中, 边缘设备多为资源受限型——内存小计算资源有限^[15], 所以数据传输功能和设备感知功能需要寻找合适的协议, 完成设备间互操作功能的定制, 例如数据的打包、通信的建立等。设备通信的实现难点在于协议内容的定制, 如何利用协议完成设备的互操作。

4.3 数据分发和时间管理

边缘侧仿真的数据分发需要考虑仿真任务的目的和要求, 根据边缘节点和边缘设备的计算资源和存储资源、参与计算的数据量和仿真关系, 确定数据的分片、转移位置和数据类型。与传统仿真平台相比, 边缘侧仿真平台的数据分发可以借用边缘节点的计算力和存储资源, 使数据可以在仿真环境中快速的流动处理。

但是, 随之也带来了更多的挑战: 因为边缘节点的加入, 在确定仿真成员之间的关系时, 需要寻找已经被分发至边缘节点的应用程序; 在平衡仿真成员之间的计算能力和存储能力时, 需要将仿真设

备周围的边缘节点的属性考虑在内; 为了使仿真更贴近真实情况, 还需考虑并克服边缘设备的移动性。总之, 边缘侧仿真平台的数据分发面对了一个更复杂的网络。

边缘侧仿真的时间管理是解决加入了工作负载共享带来的边缘节点和边缘设备之间的时间差。此外边缘侧仿真可能造成依赖性低的任务在不同的边缘节点同时执行, 这将导致物理时间仿真时间和执行时间之间的复杂关系。因此, 确保各个边缘节点之间存在等待延迟和执行延迟的同时正确完成时间同步完成仿真任务, 是时间管理功能实现的难点。

4.4 边缘侧仿真模型工作负载共享

工作负载共享是多个边缘节点共同完成一个仿真操作的必要功能, 也是边缘侧仿真平台的特征体现。工作负载共享考虑仿真任务的最大延迟需求, 结合边缘设备的计算能力和边缘节点的计算资源, 计算出可以满足低时延计算要求的边缘节点, 然后进行工作负载共享。

工作负载共享需要多模块的协助, 例如监控服务、数据分发服务、通信互操作等等, 所以轻量易启动的工作负载共享功能是实现的挑战之一。此外, 工作负载共享涉及多个硬件设备, 设备具有异构性, 涉及的传输协议也是多样化的, 这些都是对边缘侧仿真平台实现的挑战。

在 iFogSim 仿真雾环境工具中还提供了能耗监控^[16]。如何使仿真满足最小的计算时间的同时, 尽可能的减少计算能耗, 也是实现边缘侧仿真平台的解决的重点和挑战。

5 边缘侧仿真框架设计

为使边缘侧计算与仿真技术结合, 实现上文中提到的设计需求与挑战, 设计了边缘侧仿真平台框架, 如图 3 所示, 分为应用程序部署层、仿真层、微服务层、通信层、硬件层和物联网层。该框架可以快速轻量地部署在任何设备中, 通过通信层进行交流, 满足设备的可扩展性。



图3 基于微服务的边缘侧仿真框架

Fig. 3 Edge-side simulation framework with micro-services

5.1 应用程序部署层

应用程序部署层作为应用程序的输入层,接收工作流。该层作为仿真平台的用户界面层,为用户提供了程序输入接口,用户可以使用仿真平台提供的程序分割工具,将仿真任务按照仿真平台的要求进行拆分标记,并按照执行顺序依赖构成有向无环图。

该层提供的两项服务为:程序分割工具和工作流转换工具。实现方式与 Node-RED 软件类似,用户拖拽节点,每一个节点代表应用程序块——用户使用分割工具分割程序构成的程序块。再将节点按照执行依赖连接起来构成有向无环图,在之后工作中将会为每条边设置权重,使依赖关系不强的节点可以进一步的并行工作,加快仿真进度。

5.2 仿真层

仿真层是应用程序部署层的逻辑实现层,提供时间管理工具、数据分布工具和协同控制工具,实现基本的仿真逻辑,屏蔽边缘节点之间的传输延迟和执行等待。

时间管理工具是仿真系统必不可少的控制工具,平衡仿真时间与物理时间的关系,协同多个边缘节点共同执行一个仿真任务,这与 HLA 分布交

互式仿真体系结构中的时间管理需求类似。因此时间管理工具的实现参考 HLA 仿真系统中的时间管理服务,包括消息分发顺序、逻辑时间推进协议和传输服务三项^[17]。

数据分布工具通过过滤机制对数据预处理,将参与计算的数据仅分发到对应的边缘节点中,减少发送数据的冗余度和重复度。协同控制工具包括仿真运行管理工具和仿真成员的登记工具^[18],前者包括了仿真的开始结束暂停撤销等基本运行功能控制,后者则是多仿真设备的加入退出的登记操作,二者搭配完成多个边缘节点同时仿真的协同工作。

5.3 微服务层

微服务层提供满足雾计算的服务模块:资源管理服务、监控服务、工作负载转移服务、设备感知服务、边缘节点交流服务、用户服务质量服务和策略设置服务。该层的实现充分利用了边缘节点的计算资源和存储资源,减少仿真时间,满足所仿真的程序的 QoS 要求,同时也满足仿真的 QoS 要求。

资源管理服务管理边缘节点的存储资源,在文献[19]中通过设置雾设备的资源调度策略,按照资源被移动设备请求的热度,对资源进行移除和加载,可以有效地提高资源的利用率,降低传输消耗。其次在仿真任务进行时,资源管理服务为仿真任务分配计算资源和存储计算数据的硬件资源。当计算任务从边缘设备转移至边缘节点时,边缘节点为程序建立轻量级的线程启动。

监控服务监控应用程序执行情况,当指标数据超过规定阈值时,使用设备感知工具寻找可以使用的边缘节点,策略服务和用户服务质量服务找到最适合的边缘节点,使用工作负载转移服务将应用程序转移到边缘节点工作,由于转移的应用程序个数可能为多个,所以需要边缘节点交流工具帮助边缘节点之间协同工作,共同完成仿真任务。

协同服务为节点之间的互操作行为提供方法模块。利用通信层提供的服务,建立感知通信的接口和数据传输通信接口,以及数据分发、打包、解析等设备间的交互性行为。

5.4 通信层

通信层为边缘节点和边缘设备提供了通信服务。边缘设备与边缘节点之间的通信建立是已知的, 在部署应用程序之前已经确定的, 主要负责计算数据的传输(传感器到边缘节点), 或者是计算结果的反馈(边缘节点到执行设备)。通信层为实现边缘设备和边缘节点的连接提供了 LoWAN 服务, 为实现数据的传送提供了 MQTT 服务。

此外, 通信层为边缘节点之间的通信提供了服务。边缘节点与边缘节点之间的通信是未建立的, 需要设备感知功能找寻周围的边缘节点, 为该功能提供了 CoAP 服务, 边缘节点之间的数据传输同边缘设备, 使用 MQTT 服务。

5.5 硬件层

硬件层是设备的操作系统层, 将该框架与真实设备的操作系统融合, 使用设备本身的操作系统提供基本的运算、联网、存储、加速功能。这层从真正意义上屏蔽了设备之间的异构性, 利用不同的操作系统实现相同的服务提供给框架上层, 实现仿真功能。同时该层的实现也决定了该框架是一个轻量级易部署的框架。

5.6 物联网层

物联网层是指传感器和执行器的连接接口, 包括虚拟设备的接口和真实设备的接口, 即连接产生数据的硬件(传感器)和表达结果的机器(执行器),

比如电脑的鼠标和屏幕。硬件层可以使用 LVC 仿真技术中的实况仿真技术, 连接实际的边缘设备, 也可以使用 LVC 仿真技术中的虚拟仿真技术, 连接模拟器。

6 框架实现

基于设计的边缘侧仿真方法, 我们实现了一个轻量级的边缘侧仿真框架。仿真框架采用 Node-RED 系统实现可视化的边缘侧仿真流程编辑。Node-RED 是由 IBM Emerging Technologies 团队开发的图形化可拖拽式数据流编程软件, 根据服务中数据的处理流程编写 flow 实现服务程序^[20]。Node-RED 基于 Node.js 实现, 通过高性能的 JavaScript 语言来完成对系统不同模块的调用。这带来的显著优势就是一些业务逻辑不在需要经过编译就可以直接运行。实验数据表明, 采用谷歌 V8 引擎实现的 JS 引擎具有非常高的解释性能, 与 C 语言实现的业务逻辑性能没有明显差异。在应用程序部署层之下, 我们通过 Node-RED 流程实现了一系列的仿真服务, 主要包括时间管理服务、数据分发服务和模型协作服务。这一层的仿真服务是通过下一层的微服务实现的。在微服务层, 我们实现了协同服务、设备感知服务、数据通信服务、资源管理服务等一系列微服务。图 4 所示为实现的协同微服务, 仿真模型通过协商服务于邻近的节点进行协商, 协商一致的节点可以直接进行交互。

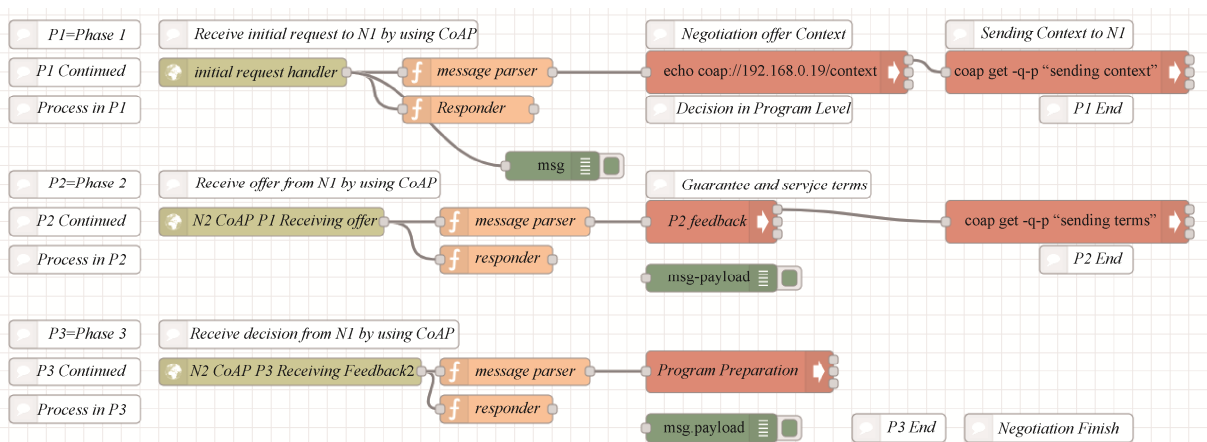


图 4 边缘节点协商微服务

Fig. 4 Negotiation micro-service for edge nodes

<http://www.china-simulation.com>

• 4543 •

边缘侧仿真的通信层充分借鉴了物联网通信层的技术,采用 MQTT 协议来支持 Pub/Sub 的数据分发模式,采用 CoAP 协议实现轻量级的模型交互。MQTT 是一种轻量级的消息传递协议,适用于物联网中低功率设备或移动设备的消息传输,在本平台中用于实现计算负载的转移和数据传输。CoAP 协议被用作网络受限的设备间通过互联网通信的协议,该协议通过类似 JSON 的方式进行通信,虽然信息自解释能力变弱,但是通信数据量小,实时性高,适合设备间的通信。

我们将实现的边缘侧仿真框架部署到了 3 个轻量级的边缘节点上,构建了简单的仿真环境,用于验证框架的有效性,如图 5 所示。在该仿真环境中采用了 Orange Pi、Raspberry Pi 和 Intel 的 x86 开发板,部署了包含摄像头的不同传感器。仿真任务可以在任意边缘节点部署,结合 LVC 仿真技术,边缘设备可以做为真实的参与仿真的物联网结点,也可以运行构造仿真模型,通过计算完成仿真模型功能。下一步将进一步完善该仿真应用并进行性能试验,验证边缘侧仿真应用在实时性和数据通信方面带来的提高。

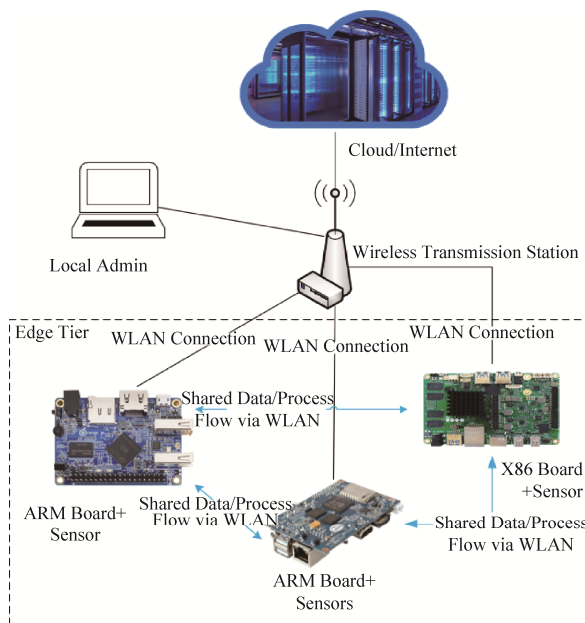


图 5 基于 SoC 系统的边缘侧仿真应用

Fig. 5 Edge-side simulation application for based on system

7 结论

本文提出的边缘侧仿真平台实现框架,将边缘计算雾计算与仿真系统技术进行了融合。边缘侧仿真平台是云仿真平台的进阶,享有云仿真平台的高计算容量优势,同时解决了大数据量带来的链路束缚,提高了仿真的效率和速度。结合所参照的 LVC 技术,使仿真计算与真实世界的融合成为可能,人们可以随时随地地获取仿真服务。

本文目前完成边缘侧仿真框架的设计和原型实现,在接下来的研究中,将使用该框架进行典型仿真应用、进行性能测试。并逐步实现框架的容错性,完善边缘侧仿真的微服务,进一步提高仿真效率,同时考虑多仿真任务的并行处理和对其他仿真模型的兼容性,使平台更加的人性化易操作。

参考文献:

- [1] 李伯虎. 现代建模与仿真技术的新发展[C]// 中国自动化学会控制理论专业委员会. 第二十一届中国控制会议论文集中国自动化学会控制理论专业委员会:, 2002: 4.
Li Bohu. New development of modern modeling and simulation technology[C]// China Automation Society Control Theory Committee. Proceedings of the 21st China Control Conference. China Automation Society Control Theory Committee, 2002 : 4.
- [2] 杨志和. 物联网的边界计算模型: 雾计算[J]. 物联网技术, 2014, 4(12): 65-67, 70.
Yang Zhihe. A boundary calculation model of IoT fog computing: Fog Calculation[J]. Internet of Things Technology, 2014, 4(12): 65-67, 70.
- [3] 李伯虎, 柴旭东, 侯宝存, 等. 一种基于云计算理念的网络化建模与仿真平台——“云仿真平台”[J]. 系统仿真学报, 2009, 21(17): 5292-5299.
LI Bohu, CHAI Xudong, HOU Baocun, et al. Networked Modeling & Simulation Platform Based on Concept of Cloud Computing—Cloud Simulation Platform[J]. Journal of System Simulation, 2009, 21(17): 5292-5299.
- [4] 周玉芳, 余云智, 翟永翠. LVC 仿真技术综述[J]. 指挥控制与仿真, 2010, 32(4): 1-7.
ZHOU Yufang, YU Yunzhi, ZHAI Yongcui. Review on LVC Simulation Technology[J]. Command and Simulation, 2010, 32(4): 1-7.
- [5] 李世超. 基于雾计算的物联网架构研究[J]. 电信工程

- 技术与标准化, 2017, 30(11): 63-66.
- Li Shichao. Research on internet of things architecture based on fog computing[J]. Telecommunications Engineering Technology and Standardization, 2017, 30(11): 63-66.
- [6] 方巍. 从云计算到雾计算的范式转变[J]. 南京信息工程大学学报(自然科学版), 2016, 8(5): 404-414.
- Fang Wei. A paradigm shift from cloud computing to fog computing and edge computing[J]. Journal of Nanjing University of Information Science and Technology (Natural Science Edition), 2016, 8(5): 404-414.
- [7] 李伯虎, 柴旭东, 朱文海, 等. 现代建模与仿真技术发展中的几个焦点[J]. 系统仿真学报, 2004, 16(9): 1871-1878.
- LI Bohu, CHAI Xudong, ZHU Wenhai, et al. Some Focusing Points in Development of Modern Modeling and Simulation Technology[J]. Journal of System Simulation, 2004, 16(9): 1871-1878.
- [8] P Varshney, Y. Simmhan. Demystifying Fog Computing: Characterizing Architectures, Applications and Abstractions[C]//, 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, 2017: 115-124.
- [9] P M P Silva, J Rodrigues, J Silva, et al. Using Edge-Clouds to Reduce Load on Traditional WiFi Infrastructures and Improve Quality of Experience[C]//, 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, 2017: 61-67.
- [10] 李伯虎, 柴旭东, 张霖, 等. 面向新型人工智能系统的建模与仿真技术初步研究[J]. 系统仿真学报, 2018, 30(2): 349-362.
- Li Bohu, Chai Xudong, Zhang Ling, et al. Preliminary Study of Modeling and Simulation Technology Oriented to Neo-type Artificial Intelligent Systems[J]. Journal of System Simulation, 2018, 30(2): 349-362.
- [11] S Guo, B Xiao, Y Yang, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C]//, in Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on. IEEE, 2016: 1-9. doi: 10.1109/INFOCOM.2016.7524497
- [12] 汪洲, 彭晓源, 李宁. 基于网络中心战概念的虚拟战场建模/仿真技术[J]. 系统仿真学报, 2005, 17(6): 1294-1298.
- WANG Zhou, PENG Xiaoyuan, LI Ning. Key Technologies in Construction of Network-Centric Warfare Concept Based Virtual Battlefield[J]. Journal of System Simulation, 2005, 17(6): 1294-1298.
- [13] 汤春明, 张荧, 吴宇平. 无线物联网中 CoAP 协议的研究与实现[J]. 现代电子技术, 2013, 36(1): 40-44.
- TANG Chunming, ZHANG Ying, WU Yuping. Research and Study and implementation of CoAP protocol in wireless Internet of Things[J]. Modern Electronics Technique, 2013, 36(1): 40-44.
- [14] S H Seo, J Straub. Comparative analysis of graph partitioning algorithms in context of computation offloading[C]//, 2017 IEEE International Conference on Electro Information Technology (EIT), Lincoln, NE, 2017: 1-7. doi: 10.1109/EIT.2017.8053425
- [15] X. Wei, MVR: An Architecture for Computation Offloading in Mobile Edge Computing[C]//, 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, 2017: 232-235. doi: 10.1109/IEEE.EDGE.2017.42
- [16] H Gupta, A V Dastjerdi, S K Ghosh, et al. Ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things edge and fog computing environments[C]//, ar Xiv preprint arXiv:1606.02007, 2016.
- [17] 黄健, 黄柯棣. HLA 中的时间管理[J]. 计算机仿真, 2000(4): 46-78.
- Huang Jian, Huang Kedi. Time Management in the High Level Architecture[J]. Computer Simulation, 2000(4): 46-78.
- [18] 陈晓波, 熊光楞, 郭斌, 等. 基于 HLA 的协同仿真运行研究[J]. 系统仿真学报, 2003, 15(12): 1707-1711, 1723.
- CHEN Xiabo, XIONG Guangleng, GUO Bin, et al. Research on Co-simulation Running Based on HLA[J]. Journal of System Simulation, 2003, 15(12): 1707-1711, 1723.
- [19] S Guo, B Xiao, Y Yang, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C]//, IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, 2016: 1-9.
- [20] 王江波. 基于 Node-red 与 Redis 的实时流数据处理模型的设计与应用[D]. 成都: 电子科技大学, 2017.
- Wang Jiangbo. The design and application of real-time stream data pcomputing model based on Node-red and Redis[D]. Chengdu: University of Electronic Science and Technology of China, 2017.