



基于容器技术的电力设备仿真云平台设计与开发

赵宇峰, 雷 晟, 张国钢, 耿英三

(西安交通大学 电力设备电气绝缘国家重点实验室, 西安 710049)

摘 要: 针对电力设备设计仿真流程中成本高、计算资源利用效率低等问题, 利用 Docker 容器化技术和 Kubernetes 容器编排技术, 构建基于容器的电力设备仿真云平台。结合仿真云平台的需求, 从功能实现、镜像构建、云平台架构等方面设计云平台各模块, 并对云平台运行中的关键处理流程进行分析。在此基础上, 通过动态构建技术从特性匹配、调度算法等方面对云平台的运行进行优化。应用实例结果表明, 基于容器的电力设备仿真云平台能充分发挥云平台的优势, 正确地根据用户提供的参数应用模板进行仿真。

关键词: 仿真云平台; 电力设备; 云计算; Docker 技术; Kubernetes 技术

开放科学(资源服务)标志码(OSID):



中文引用格式: 赵宇峰, 雷晟, 张国钢, 等. 基于容器技术的电力设备仿真云平台设计与开发[J]. 计算机工程, 2021, 47(9): 171-177, 184.

英文引用格式: ZHAO Y F, LEI S, ZHANG G G, et al. Design and development of container-based cloud platform for power equipment simulation[J]. Computer Engineering, 2021, 47(9): 171-177, 184.

Design and Development of Container-based Cloud Platform for Power Equipment Simulation

ZHAO Yufeng, LEI Sheng, ZHANG Guogang, GENG Yingsan

(State Key Laboratory of Electrical Insulation and Power Equipment, Xi'an Jiaotong University, Xi'an 710049, China)

[Abstract] The existing methods for power equipment simulation are costly, and have a low utilization rate of computing resources. To address the problem, a container-based cloud platform for power equipment simulation is constructed using Docker and Kubernetes. Based on the actual requirements, the modules of the cloud platform are designed from the aspects of function implementation, image construction, and cloud platform architecture. In addition, the procedures that are key to the cloud platform operation are analyzed. On this basis, the dynamic building technology is used to optimize the operation of the cloud platform from the aspects of feature matching and the scheduling algorithm. The simulation result shows that the proposed platform uses cloud computing to maximum advantage, and can provide simulation correctly according to the parameters given by users.

[Key words] simulation cloud platform; power equipment; cloud computing; Docker technology; Kubernetes technology
DOI: 10.19678/j.issn.1000-3428.0058324

0 概述

随着电力系统的规模不断扩大, 电网智能化程度不断提高, 对电力设备产品的创新也有了更高的要求^[1]。新形势下, 传统电力设备设计仿真面临诸多方面的挑战, 而云计算技术和虚拟化技术能较好地解决传统单机模式下进行仿真所存在的各种问题^[2-4]。基于云计算技术, 云仿真将仿真过程中使用的计算资源、存储资源进行虚拟化并放入云端^[5], 用

户可以通过多种设备接入云仿真平台进行各项操作, 仿真用户按照计算时间来付费, 避免产生高性能计算机的购置费用。同时, 云计算平台可以充分发挥多核处理器的优势, 并根据集群中的负载情况自动扩容和缩容, 提高主机计算资源的利用效率^[6-7]。

国内外学者对云计算进行仿真的实现方法做了很多研究。文献[8]以云计算理念为基础, 提出一种网络化建模与仿真平台——云仿真平台。文献[9]研究虚拟化技术下云仿真运行环境动态构建技术,

基金项目: 国家重点研发计划“智能配电网复杂用电工况下的计量传感和校准技术研究”(2016YFF0201205)。

作者简介: 赵宇峰(1996—), 男, 硕士研究生, 主研方向为云平台架构设计; 雷 晟, 硕士研究生; 张国钢(通信作者), 耿英三, 教授、博士。

收稿日期: 2020-05-14 **修回日期:** 2020-08-06 **E-mail:** zyf1996@stu.xjtu.edu.cn

提出动态构建的三层映射算法。文献[10]提出一种基于高级架构(High-Level Architecture, HLA)的仿真框架,并通过软件即服务(Software-as-a-Service, SaaS)向用户提供复杂系统仿真的支持。文献[11]研究多租户架构(Multi-Tenancy Architecture, MTA)的云仿真相关问题,并构建一个可以满足租户特定要求灵活的云仿真架构。文献[12]将云计算应用到电力仿真之上,设计并构建高性能的电力仿真平台——CloudPSS。

近年来,云计算中容器化技术和容器编排技术得到了突飞猛进的发展。以Docker为代表的容器化已逐渐取代传统使用虚拟机进行操作系统级别的虚拟,成为虚拟化技术中新的代表技术^[13-14]。与容器化技术相对应的容器编排技术也得到逐步提高,特别是谷歌公司于2014年开源Kubernetes以来,该应用具有强大的容器管理能力,已成为容器管理领域的事实标准^[15]。Docker与Kubernetes的结合在各领域中得到了应用,如气象^[16]、通信^[17]等。

随着相关技术的普及,近年来基于容器化技术和容器编排技术的仿真平台研究发展迅速,但在发展过程中也同样存在着许多问题。例如,由于底层的物理资源性能条件差异、用户需求不定,仿真平台需要合理分配现有资源,做好资源管理、调度和动态构建的工作。文献[18]提出基于聚类和改进共生演化算法的任务调度策略,相比传统的启发式调度算法,其调度成本更低,但用户满意度更高。文献[19]提出改进的协同优化文化基因任务调度算法,在循环中加入协同进化操作,有效地提高算法的运行效率。

除此之外,由于不同行业中使用的专业软件不同,需进行仿真的仿真任务特性也各有差异,云仿真平台在具体行业领域的应用过程中存在问题。研究者将云仿真平台与已有的TMSR-SF1工程仿真机相结合,实现了适用于核电仿真的云仿真平台^[20]。

为解决传统单机模式下仿真中存在的问题,本文基于Docker容器技术和Kubernetes容器编排技术,设计并开发专用于电力设备仿真的电力仿真云平台,研究该云平台的运行流程和优化调度策略,提高云平台的整体性能和集群主机计算资源的利用效率。

1 需求分析

电力设备仿真云平台的功能需求主要有以下5个方面:

1) 用户管理。用户可以进行注册、登录等操作,并由云平台负责进行身份验证。用户在登录账户成功后,即可保持登录状态。除非清除浏览器缓存或更换浏览器,用户无需再次登录。

2) 仿真任务提交流程管理。云平台应保证用户

提交的仿真参数可以准确无误地被应用到模板中,并生成用于执行仿真任务的代码,最终启动仿真进程运行仿真。

3) 仿真进程生命周期管理。从仿真进程的创建、运行和实时输出进度日志,再到最终仿真进程的结束并输出结果信息或报错信息,云平台都应采取合理有效的措施进行相关处理。在整个流程中,保证可以脱离用户的参与,完全自主地进行生命周期管理,做好随时可以供用户查看仿真进度的准备。

4) 仿真日志和结果文件持久化。仿真云平台中应存在可以持久化保存数据的模块,用于仿真进程运行中和之后保存相关数据。

5) 用户交互。用户通过用户界面进行各种操作,是用户与电力设备仿真云平台进行交互的唯一接口。

此外,在确保实现各项功能不受影响的情况下,应该充分地运用合理的调度算法,采用合适的架构设计,保证云平台可以高效、稳定地实现功能。

2 模块设计与构建

2.1 模块设计

电力设备仿真云平台在设计过程中采用模块化的思想,将具有不同功能的组件单独划分出来构建模块,并以Docker容器镜像的形式应用到Kubernetes所构建的云平台上。各模块之间交互主要通过Kubernetes提供的Service资源对象进行通信,并通过XML-RPC的数据交互协议对数据交互格式进行约定。

经过对整体功能需求的分析,为保证各模块之间的高聚合、低耦合关系,将整个云平台分为4个功能模块,各模块的功能及交互关系如图1所示。

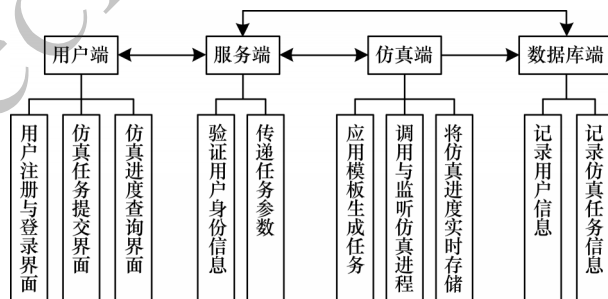


图1 云平台模块功能及交互关系

Fig.1 Module functions and interaction relationship of cloud platform

云平台模块主要有:

1) 服务端模块。服务端模块中运行Web Service的服务端程序,该模块向通过认证的用户生成并提供用户界面。此外,服务端模块还可以将仿真任务的参数传递给电力设备仿真云平台的仿真端模块,并向用户提供仿真进度的查看和仿真结果的传回。

2) 仿真端模块。仿真端模块可以从电力设备仿真云平台的服务端模块中接收新建仿真任务的参

数,并根据参数生成仿真任务。同时,该模块还可以监听所有正在被运行的仿真进程,实时获取其仿真状态并存入数据库中。

3)数据库端模块。数据库端模块中运行用于存储数据的数据库程序。该模块对云平台运行中一些必要信息进行存储,包括用户的认证信息,仿真任务所处的Pod、状态、结果等仿真信息,以供其他模块的

程序进行查询与记录。

4)用户端模块。仿真用户在用户端模块进行注册和登录操作,已通过身份验证的仿真用户进行仿真任务的新建、进度查询和结果查看。

2.2 模块构建

服务端模块、仿真端模块和数据库端模块的构建流程如图2所示。

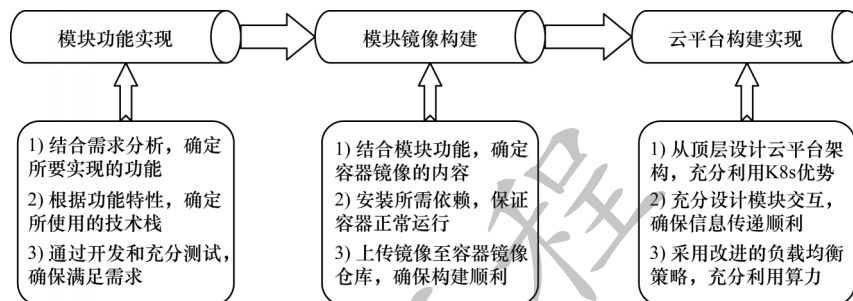


图2 云平台模块构建流程

Fig.2 Construction procedure of cloud platform module

2.2.1 模块功能实现与镜像构建

在模块功能实现的步骤中,主要是结合前文中所叙述的需求分析,对每个需求条目进行开发,并进行充分测试以保证功能可以正常实现。在模块镜像构建的步骤中,通过分析模块程序运行过程中所需要的依赖、与其他模块交互方式等,设计模块需要加载的外部卷、需要暴露的端口等,并编写Dockerfile以生成容器,并通过容器镜像仓库对容器镜像进行托管。

考虑到电力设备仿真过程中独有的特点,在模块功能实现与镜像构建的步骤中也做了各种优化措施。以仿真端模块为例,由于在日常的电力设备仿真中,Ansys、Matlab等仿真软件被广泛地应用,因此有必要在仿真端模块中集成此类软件。在集成过程中,云平台采用将软件主体作为系统服务,仅在容器镜像中安装软件运行所需依赖的方法,如图3所示。

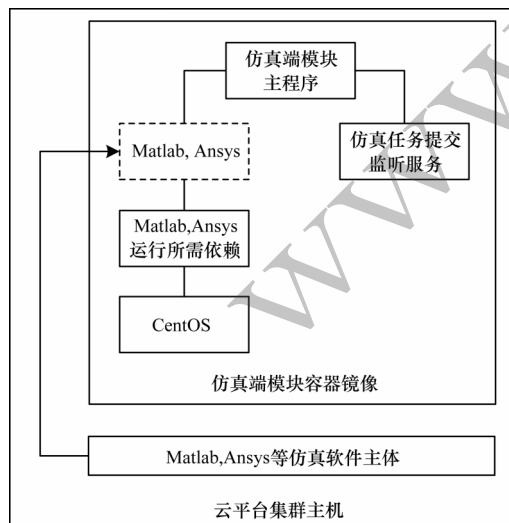


图3 仿真端模块中仿真软件的解决方案

Fig.3 The solution of simulation software in simulation module

通过此方法构建仿真端镜像,保证仿真软件在仿真端模块中可用的同时大幅减小容器镜像体积,进而加快云平台的构建速度。

2.2.2 云平台架构实现

对于服务端模块、仿真端模块和数据库端模块的云平台架构,均可使用如图4所示的架构设计。

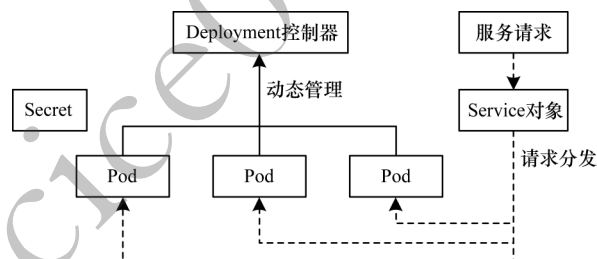


图4 模块的云平台架构实现

Fig.4 Cloud platform architecture implementation of module

单个模块的云平台架构主要由Pod、Deployment、Service、Secret 4类资源对象组成。

最底层的是承载有容器镜像的Pod对象。作为Kubernetes中执行功能的最基本单位,Pod使用前一节中所构建的镜像文件,并运行着对应的程序。与仅有一个的Deployment控制器和Service对象不同,为了充分利用云平台的集群优势,并充分发挥集群主机的性能,Pod通常会有多个副本,每个副本中都运行着相同的程序,并由上层的Deployment控制器进行统一调度和负载均衡。

Pod的上层是一个Deployment控制器,用于管理其下属的Pod资源对象。Deployment控制器通过重启因错误而停止的Pod,在配置文件被修改时对Pod进行滚动更新等方法来确保其所管理的Pod处于被期望的状态。

Service对象用于将外部请求进行负载均衡并分发至不同Pod对象中。对于服务端模块,由于其需要对集群外部的用户提供服务,因此使用可被集群外部访问的NodePort类型的Service对象。而仿真端模块和数据库端模块由于只需要对集群内部提供服务,所以只使用默认的ClusterIP类型的Service对象。

除具有层次结构的Pod、Deployment、Service资源对象之外,云平台架构还使用Secret对象来保存敏感数据。由于服务端模块和仿真端模块在运行时都需要密码对数据库进行访问,数据库端模块在运行时也需要密码进行数据库启动,因此使用Secret对象用于储存访问数据库的密码。

云平台架构中集群的每个主机都会被抽象为node资源对象。每个Pod都会在某一个node上面运行,使用node资源对象底层的物理机计算资源。

根据以上云平台架构设计思路,保证各主机在同一网络下编写yaml配置文件并将其应用至Kubernetes中,即可成功启动电力设备仿真云平台。

3 电力设备仿真云平台的运行

3.1 仿真任务的提交流程

在电力设备仿真云平台启动、身份认证成功后,用户可选择模板、填入参数进行仿真任务提交。仿真任务提交流程如图5所示。

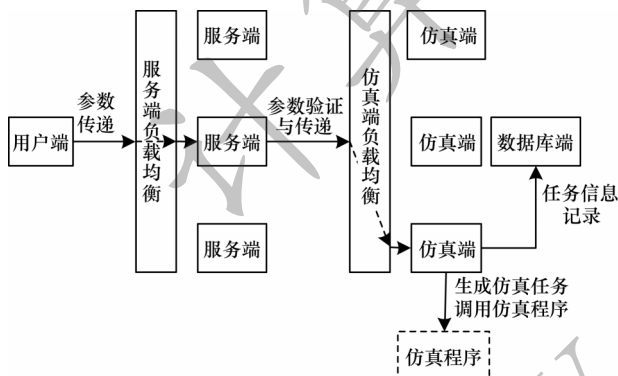


图5 仿真任务提交流程

Fig.5 Submission procedure of simulation task

在仿真任务的提交流程中,用户提交的参数共经历两次负载均衡,并根据实时负载情况分别被分配到较适合的服务端和仿真端进行处理。服务端对用户端发送的仿真参数进行进一步验证,并在验证通过后将参数发送到仿真端。而在仿真端中,则会根据服务端发送的参数生成对应的仿真任务,调用一个新的仿真进程运行仿真,同时将相关的仿真信息记录到数据库中。

在整个流程中,服务端会将用户的身份信息以session的形式予以记录,从而使用户的登录状态得以维持。同时,也采用粘滞会话的策略,尽力保证用户在整个会话周期内所有请求都在负载均衡中被转发到同一个服务端Pod上。在session和粘滞会话的

共同作用下,用户可以在一定时间内无需再次登录,用户体验也得以提升。

3.2 仿真任务的运行与轮询

由于电力设备的仿真任务所需时间不同,很多情况下并不能实时返回计算结果,因此在仿真任务的提交流程后,服务端和仿真端的连接将会被中断。仿真端会将仿真任务的仿真信息和结果实时更新到数据库端中,等待新的服务端直接从数据库端获取仿真数据。

在单个仿真端内部,将采用一个仿真任务队列来对当前所有的仿真任务进行管理。仿真任务队列中的每项均代表一个仿真任务,其背后均有一个仿真进程在对其进行仿真。仿真端主进程会通过不断轮询该仿真任务队列获得仿真任务运行的最新信息,轮询的流程如图6所示。

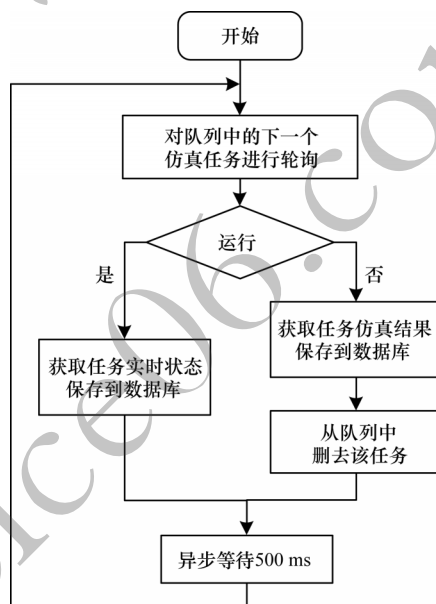


图6 仿真任务队列轮询流程

Fig.6 Polling procedure of simulation task queue

在仿真任务队列的轮询过程中,会逐个查询队列中仿真任务的运行状态。对于正在运行的仿真任务,获取其实时输出、运行状态等信息,并记录于数据库中。对于已经结束运行的仿真任务,则根据其退出码判断进程是否正常结束,并读取仿真结果,记录于数据库,最后从仿真任务队列中将该仿真任务移除。

4 电力设备仿真云平台的优化

4.1 云平台动态构建

电力设备仿真云平台的核心在于“云计算”,即使用主机集群对大量仿真任务进行处理,从而完成传统的单机环境下无法完成或者需要很长时间才能完成的工作。因此,合理规划主机集群中的计算单位,以及将不同的仿真任务分发到不同的计算单位上,是设计和实现云平台的关键步骤。在云平台上,主机集群中各部分关系如图7所示。

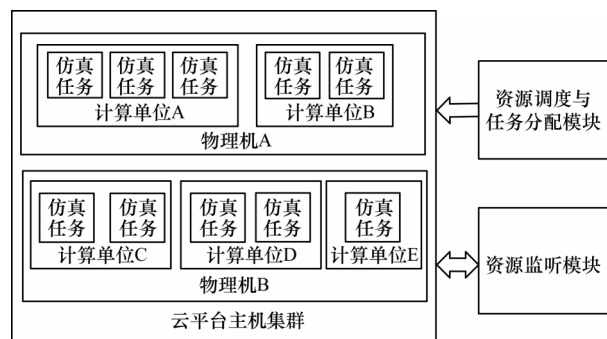


图7 主机集群中各部分关系

Fig.7 Relationship between the parts in the host cluster

在云平台的主机集群,一般会存在多个物理机,用于提供进行仿真任务所需要的计算资源。而这些物理机上的计算资源常通过划分为多个虚拟机或采用容器化的方式进行虚拟化,从而分割成一个或多个计算单位。计算单位是直接对仿真任务进行承载的单位,其对内部运行的仿真任务表现为一个具有完整CPU、内存、I/O、网络等资源的实体,对外则接受来自于管理系统的资源调度,与同一个物理机内的不同计算单位共同分配并使用物理机的计算资源。

此外,为实时获取到各单位的资源空闲程度,进而合理分配调度已有资源,管理系统会通过资源监听模块对每个计算单位和物理机上的CPU、内存等计算资源使用情况进行监听。资源调度与任务分配模块则会根据资源监听模块所监听到的资源使用情况,再根据仿真任务自身的特点,采用特定的算法将仿真任务进行分配。资源监听模块还会在某些需要的条件下增加或删除物理机中的计算单位,从而保

证计算资源最高效使用。

本文使用Kubernetes管理的电力设备仿真云平台系统,上述的物理机被抽象为Kubernetes中的node资源对象,而计算单位为node之下的Pod资源对象。Kubernetes作为容器管理工具,其内部进行了很多优化,从而可以适应较通用的云平台构建,例如Deployment中Pod的自动扩容。

4.2 仿真任务特性与运行主机特性相匹配

对于电力设备仿真任务,不同的仿真类型、网格划分方式或划分密度都可能影响其在仿真过程中所需要的计算资源。同样,集群中不同的主机也可能具有不同的配置,如有的主机CPU性能相较于其内存容量更占优势,或者有的主机CPU性能较为低下,但是具有相对比较大的内存。因此,若在仿真任务的提交流程中,将合适的仿真任务提交到合适的集群主机中,则可以充分利用集群主机中CPU和内存资源,进而提高整体运行效率。

为实现仿真任务特性与运行主机特性相匹配,首先对云平台架构进行改造。将架构中的node资源对象分为内存优势型主机和CPU优势型主机。与node的改动相适应,在Deployment资源对象的设计中,同样将原本单个Deployment改造为内存优势型Deployment和CPU优势型Deployment,并通过亲和调度策略分别分配到对应的node主机上。最终是Service资源对象的改造,分为内存型、CPU型和自动分配型,其中内存型Service将请求发送至内存优势型Deployment,CPU型Service将请求发送至CPU优势型Deployment,自动分配型Service则根据仿真系统中整体的负载情况进行自动分配。经过如上改造,最终云平台架构中各资源对象的对应关系如图8所示。

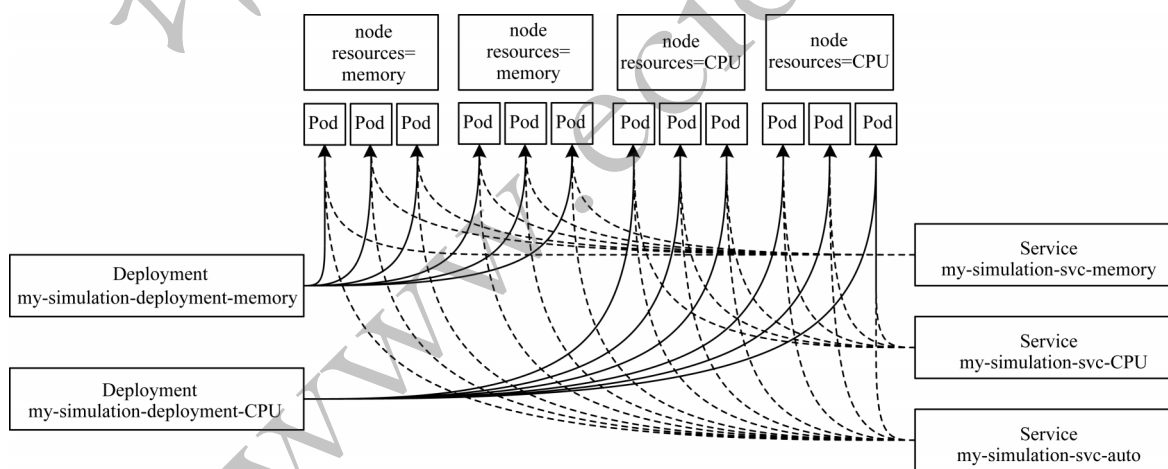


图8 不同资源对象的对应关系

Fig.8 Correspondence between different resource objects

从图8可以看出,每个Deployment均与其管理的Pod用实线相连,每个Service与可能被分配仿真任务的Pod用虚线相连。用户在新建仿真任务时,根据其选择的参数,估算出仿真任务的资源使用特性,并选择合适的Service来提交仿真任务。对于难以估算资源特性的仿真任务,使用默认auto类型的

Service,云平台根据所有Pod的负载情况进行仿真任务分发。

4.3 云平台中仿真任务与Pod之间的映射

相比传统的单机平台,云平台利用灵活的负载均衡算法,将多个仿真任务分配到不同的Pod上,从而使云平台整体的计算资源得到充分利用。在仿真

任务和 Pod 资源对象之间的映射算法中,采用预选、优选、选定来确定最终要执行仿真任务的 Pod。

1) 预选

从多个角度对 Service 所相连的 Pod 资源对象进行评价,并排除硬性条件上无法进行仿真任务新建的 Pod 资源对象。在预选步骤中,程序将进行如下评价:当前 Pod 是否可被连接;当前 Pod 中内存使用量与其内存限制之间的差值是否小于阈值;当前 Pod 中内存使用率和 CPU 使用率是否大于阈值等。只有上述条件全部评价通过,才能通过预选环节,进入优选阶段。

2) 优选

将对 Pod 中各计算资源的使用率进行排名。每个 Pod 有 4 个资源指标,分别为 CPU、内存、网络和 I/O。对于电力仿真任务,使用的资源主要是 CPU 和内存。因此,在进行资源使用率的排名中,对 CPU 和内存设置了相对比较大的权重,而对网络和 I/O 设置了相对比较小的权重。

假设第 d 号 Pod 资源对象,其 CPU 的使用限制为 N_{CPU} ,内存的使用限制为 N_{memory} ,网络的使用限制为 N_{Net} ,I/O 的使用限制为 $N_{I/O}$,其对应的使用量分别为 n_{CPU} 、 n_{memory} 、 n_{Net} 、 $n_{I/O}$,则该 Pod 的评分如式(1)所示:

$$V_d = K_1 \frac{n_{CPU}}{N_{CPU}} + K_2 \frac{n_{memory}}{N_{memory}} + K_3 \frac{n_{Net}}{N_{Net}} + K_4 \frac{n_{I/O}}{N_{I/O}} \quad (1)$$

其中: K_1 、 K_2 、 K_3 、 K_4 分别为 CPU、内存、网络和 I/O 资源对象的权重,且 K_1 和 K_2 大于 K_3 和 K_4 。

将所有通过预选的 Pod 资源对象按照式(1)计算,即可得出每个 Pod 对应的评分。至此,优选部分结束。

3) 选定

使用优选步骤中得出的评分序列,根据式(2)计算最终要使用的 Pod。

$$V_x = \min(V_x), x = 1, 2, \dots, n \quad (2)$$

其中: x^* 为采用映射算法得出的最优 Pod,其通过选取资源使用率最低的 Pod 而得出。

4.4 扩容机制与扩容映射算法

若所有 Service 对应 Pod 均达到内存使用量的阈值等条件而无法通过预选,则说明当前云平台内的仿真任务较繁重,需要进行 Pod 扩容。在扩容过程中,新增加的 Pod 通过一个映射算法来决定其被分配到哪台主机上,即需要确定新增 Pod 对象与 node 对象的对应关系。

由于 Pod 对象与 node 对象之间的关系和仿真任务与 Pod 对象之间的关系十分相似,因此采用对 node 进行预选、优选和选定,并最终确定新的 Pod 所属 node 对象。当 node 中负载过大时,集群主机的扩容无法保证实时性。因此,当云平台处于满负荷运行时,需要将此时新加入的仿真任务加入一个队列

中,等待资源空闲时再开始执行仿真。

5 应用实例

5.1 测试环境

使用由 3 台主机构成的集群进行测试,集群主机的软硬件配置采用 CPU Intel i7-9700K、内存 64 GB、硬盘 500 GB SSD,操作系统 Ubuntu 18.04。

将其中 1 台主机设置为 Master 节点,另外 2 台设置为工作节点。对主机进行相关配置,并进行 Docker、Kubernetes 等必备软件安装。在配置过程中,将 Ansys、Matlab 作为系统服务供仿真端 Docker 调用,从而大幅减小仿真端镜像体积。同时,使用容器镜像服务对服务端模块、仿真端模块和数据库端模块的容器镜像进行托管,使 Pod 资源对象被首次构建时可自动从镜像托管服务处下载并应用容器镜像。

5.2 功能性测试

以变压器铁芯内部磁场分布的仿真为例,对电力设备仿真云平台进行功能性测试。云平台正常启动后,仿真用户可以在浏览器中输入 Master 节点的 IP 地址以及服务端模块的 Service 对象所指定的 NodePort 端口来接受服务。用户可以在云平台所提供的用户界面中登录或注册,如图 9 所示。



图 9 云平台的用户登录界面

Fig.9 User login interface of cloud platform

注册成功并登录账户后,选取模板并输入参数进行仿真任务提交。仿真任务的参数设置和提交界面如图 10 所示。使用空载变压器的 Ansys 仿真模型如图 11 所示。



图 10 云平台的仿真任务提交界面

Fig.10 Simulation task submission interface of cloud platform

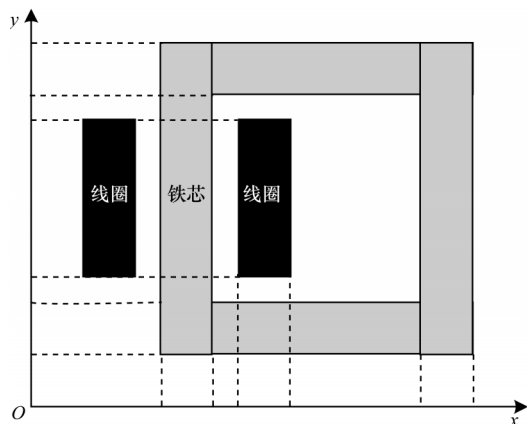


图11 空载变压器仿真模型

Fig.11 Simulation model of no-load transformer

设置铁芯相对磁导率为2 000 H/m,电流密度为1 000 A/m²。由于该仿真任务的模型结构比较简单,仿真平台能很快得出并返回仿真结果,返回的结果如图12所示。从图12可以看出,云平台已成功应用参数在模板之中,调用仿真进程执行仿真,并返回正确的仿真结果。



图12 变压器铁芯内部磁场分布的仿真结果

Fig.12 Simulation results of magnetic field distribution inside transformer core

为验证云平台的仿真结果持久化功能,退出目前云平台的登录并将设备更换为其他设备。重新登录后发现仿真结果仍然可以被查看,由此验证云平台的数据持久化功能。

在以上实例中,云平台按照预期实现了需求分析中的用户管理、提交流程管理、仿真任务管理、仿真日志和结果文件持久化、用户交互等功能,完成电力设备仿真云平台的功能性测试。

6 结束语

本文采用Docker容器化技术和Kubernetes容器编排技术,结合模块化思想,搭建基于容器的电力设备仿真云平台。通过仿真特性匹配和两个映射算法对电力设备仿真云平台的运行进行优化。应用实例结果表明,该电力设备仿真云平台利用云平台优势

可完成用户提出的仿真要求,并满足需求分析中的各项功能要求。后续将在云平台上开发基于遗传算法等启发式算法的仿真模型参数优化功能,进一步发挥云平台的优势。

参考文献

- [1] 王建华,张国钢,宋政湘,等. 物联网+大数据+智能电器——电力设备发展的未来[J]. 高压电器,2018,54(7):7-15,25. WANG J H, ZHANG G G, SONG Z X, et al. Internet of things, big data, intelligent electrical apparatus—the future development of power equipment [J]. High Voltage Apparatus, 2018, 54(7): 7-15, 25. (in Chinese)
- [2] ARMBRUST M, FOX A, GRIFFITH R. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50-58.
- [3] 全力,傅明. 云计算中任务调度优化策略的研究[J]. 计算机工程, 2018, 44(8): 14-18. QUAN L, FU M. Research on task scheduling optimization strategy in cloud computing [J]. Computer Engineering, 2018, 44(8): 14-18. (in Chinese)
- [4] JAIN N, CHOUDHARY S. Overview of virtualization in cloud computing[C]//Proceedings of 2016 Symposium on Colossal Data Analysis and Networking. Washington D. C., USA: IEEE Press, 2016: 1-10.
- [5] MAGNUSSON P S, CHRISTENSSON M, ESKILSON J, et al. Simics: a full system simulation platform [J]. Computer, 2002, 35(2): 50-58.
- [6] PAHL C. Containerization and the paas cloud [J]. IEEE Cloud Computing, 2015, 2(3): 24-31.
- [7] NOOR T H, SHENG Q Z, NGU A H H, et al. Analysis of web-scale cloud services [J]. IEEE Internet Computing, 2014, 18(4): 55-61.
- [8] 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 B H, CHAI X D, HOU B C, 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. (in Chinese)
- [9] 张雅彬,李伯虎,柴旭东,等. 基于虚拟化技术的云仿真运行环境动态构建技术[J]. 系统工程与电子技术, 2012, 34(3): 619-624. ZHANG Y B, LI B H, CHAI X D, et al. Research on virtualization-based cloud simulation running environment dynamic building technology [J]. Journal of Systems Engineering and Electronics, 2012, 34(3): 619-624. (in Chinese)
- [10] XIONG W, TSAI W T. HLA-based SaaS-oriented simulation frameworks [C]//Proceedings of the 8th International Symposium on Service Oriented System Engineering. Washington D. C., USA: IEEE Press, 2014: 376-383.
- [11] TSAI W T, LI W, BAI X, et al. P4-simsaas: policy specification for multi-tendency simulation software-as-a-service model [C]//Proceedings of Winter Simulation Conference. Washington D. C., USA: IEEE Press, 2011: 3067-3081.

(下转第184页)

(上接第 177 页)

- [12] SONG Y, CHEN Y, YU Z, et al. Cloudpss: a high-performance power system simulator based on cloud computing[EB/OL]. [2020-04-10]. <https://arxiv.org/abs/1903.01081>.
- [13] 毕小红, 刘渊, 陈飞. 微服务应用平台的网络性能研究与优化[J]. 计算机工程, 2018, 44(5): 53-59.
- BI X H, LIU Y, CHEN F. Research and optimization of network performance for micro-service application platform[J]. Computer Engineering, 2018, 44(5): 53-59. (in Chinese)
- [14] SINGH S, SINGH N. Containers & docker: emerging roles & future of cloud technology[C]//Proceedings of the 2nd International Conference on Applied and Theoretical Computing and Communication Technology. Washington D. C., USA: IEEE Press, 2017: 804-809.
- [15] BERNSTEIN D. Containers and cloud: from LXC to Docker to Kubernetes[J]. IEEE Cloud Computing, 2014, 1(3): 81-84.
- [16] 关兴民, 张兆伟, 王祝先, 等. 基于 Kubernetes 的气象私有云自动化部署与优化[J]. 信息技术, 2019, 43(5): 76-80.
- GUAN X M, ZHANG Z W, WANG Z X, et al. Automatic deployment and optimization of meteorological private cloud based on Kubernetes[J]. Information Technology, 2019, 43(5): 76-80. (in Chinese)
- [17] 王志军. 中国联通容器化大数据云平台的探索与实践[J]. 信息技术与标准化, 2019(5): 66-69.
- WANG Z J. Exploration and practice of china unicom containerized big data cloud platform [J]. Information Technology and Standardization, 2019(5): 66-69. (in Chinese)
- [18] 李昆仑, 关立伟, 郭昌隆. 基于聚类和改进共生演算法的云任务调度策略[J]. 计算机应用, 2018, 38(3): 707-714.
- LI K L, GUAN L W, GUO C L. Cloud task scheduling strategy based on clustering and improved symbiotic organisms search algorithm [J]. Journal of Computer Applications, 2018, 38(3): 707-714. (in Chinese)
- [19] XIE X, YUAN T, ZHOU X, et al. Research on trust model in container-based cloud service[J]. Computers Materials & Continua, 2018, 56(2): 273-283.
- [20] 何越, 程懋松, 戴志敏. TMSR 云仿真平台初步设计与实现[J]. 核技术, 2018, 41(7): 79-86.
- HE Y, CHENG M S, DAI Z M. Preliminary design and implementation of TMSR cloud simulation platform[J]. Nuclear Techniques, 2018, 41(7): 79-86. (in Chinese)

编辑 薛晋栋