用户行为仿真的部署与调度问题研究

纪一方,张国敏,邢长友
(陆军工程大学 指挥控制工程学院,江苏 南京 210007)

摘　要:近年来,网络空间对抗形势日趋严峻,网络攻击技术更是日益复杂。但对网络攻击技术的测试不能在现实的网络环境中进行,对此,网络靶场技术成为近年来关注的一个热点。为在网络靶场中高逼真地模拟现实网络,大规模用户行为仿真技术至关重要。研究了如何在有限的资源条件下,通过虚拟化容器技术,实现大规模的用户行为仿真,并实现其部署和调度问题。

关键词:用户行为仿真;大规模部署;调度
中图分类号:TP393　　**文献标识码**:A　　DOI: 10.19358/j.issn.2096-5133.2020.01.015
引用格式:纪一方,张国敏,邢长友.用户行为仿真的部署与调度问题研究[J].信息技术与网络安全,2020,39(1):78-82.

Research on local deployment and scheduling problem of user behavior simulation

Ji Yifang, Zhang Guomin, Xing Changyou
(Command & Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China)

Abstract: In recent years, the cyberspace confrontation situation has become increasingly severe, and cyberattack technology has become increasingly complex. However, the testing of network attack technology cannot be carried out in a realistic network environment. For this reason, network shooting range technology has become a hot spot in recent years. In order to simulate realistic real-world networks in the network range, large-scale user behavior simulation technology is crucial. This paper studies how to implement large-scale user behavior simulation and realize its deployment and scheduling through virtualized container technology under limited resource conditions.

Key words: user behavior simulation; large-scale deployment; scheduling

0　引言

随着网络变得日益复杂,网络开始像生物系统一样对环境变化做出响应,网络空间靶场也需要模拟各种场景变化和网络态势事件,并对事件进行响应来改变模拟用户的行为甚至是组织关系,从而实现网络空间靶场中网络流量的动态变化,使得技术人员更好地理解复杂系统中出现的异常和变化趋势[1]。

鉴于网络空间靶场的规模和事件的复杂性,以及网络用户行为的多样性[2],在对比分析各种分布式控制技术的基础上,结合在大规模网络测量系统中已经研制开发的基于策略的调度平台基础,采用基于事件驱动的策略机制构建分布式控制平台。策略机制可以将系统的管理逻辑和应用逻辑相分离,并将管理逻辑表示为控制系统行为选择的策略。基于策略的管理支持管理员直接使用与具体实现技术无关的类自然语言或描述性策略定义语言来定义管理策略,将策略从系统实现代码中提取出来,向管理人员提供一个面向整个网络空间靶场的抽象管理平台。通过策略将对仿真行为的控制和具体执行流量发生动作分离开来,通过改变策略来支持用户行为仿真的组织结构和协同行为,并且支持网络业务类型的更新和操作的改变。管理员可以动态地增加、删除、修改现有的策略,策略经过翻译和验证后分发给相应的模拟用户进行解释执行。

用户行为仿真能够模拟用户行为执行不同业务操作的软件代理,为达到预设的测试情境,在系统配置阶段将以 LXC 容器作为基本单位进行部署和复制。通过设计一个可装载用户行为模型的平台,将根据所装载的用户行为模型模拟自然人用户的终端操作,如收发电邮、Web 浏览、下载文件、访问各类业务系统等。

目前,大规模的用户行为仿真存在以下的现实

问题:(1)组织成本大、效率低、涉及范围广,难以重复高效地开展试验;(2)真实人之间存在(技能、决策等方面)的差异性,难以建立统一的量化标准,导致试验结果难以预测。因此,通过对"模拟用户"行为进行描述并开展可控的行为仿真试验成为更可行和有效的方法。基于上述技术难题,本文提出在基于虚拟化容器的环境下,利用任务文件信息,运用 LXC 容器技术模拟用户行为,解决了本地的大规模调度问题以及真实业务访问的仿真问题。

1 虚拟化技术

虚拟化技术具有多年的发展历史,现如今容器的快速发展,逐渐取代了虚拟机的存在。容器技术很好地解决了虚拟机资源开销大、运行效率低等问题。如图 1 所示,由于两种虚拟化平台的设计结构不同,虚拟机运行时 Hypervisor 要运行新的 Guest OS 来实现对宿主机硬件操作的拦截和替换,在虚拟机的资源利用率和性能开销上增加了额外的负担,导致启动速度慢。而容器没有这种忧虑。

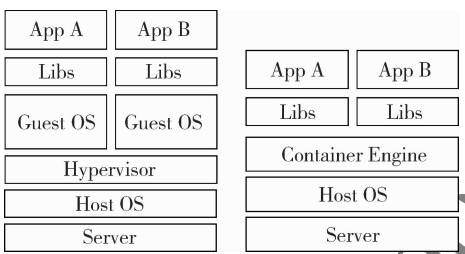

图 1 两种虚拟化平台结构对比

LXC 是基于 Linux 容器机制实现的[3],LXC 可以看做是一个在用户空间实现的容器操作工具集合,用户通过它来操作和管理一个容器。从底层出发,LXC 真正的实现是依靠 Linux 内核提供的 Cgroups[4] 和 Namespace 特性,LXC 的作用是对此作了整合与管理。Cgroups 机制是 Linux 内核提供的一个基于进程组的资源管理框架,通过结合子控制器将容器与不同的 Cgroups 子系统进行关联,可以为特定的进程组限定 CPU 时间、内存大小、I/O 速度等可使用的资源以及使用配比。Namespace 机制是用来提供容器的隔离性,通过将资源划分至不同且特性的 Namespace 中,使得诸如 PID、IPC 等系统资源不再是全局性的。

2 相关工作

目前主要采用基于策略的调度机制。分布式并行技术的核心在于设计良好的分布式并行调度算法[5-6]。在分布式实时系统领域,目前已有一些使用较为广泛的调度理论和方法:SymTA/S[7] 是基于 Holistic 调度分析方法[8] 的扩展,使用了标准事件模型[9-10] 来耦合分布式系统各个组件之间的联系,从而支持在系统级进行调度分析;实时演算(real-time calculus)是基于网络演算(network calculus)的扩展,通过建立合适的任务到达曲线和系统服务曲线来对任意模式的事件流进行建模,从而不仅仅局限于将非周期事件流近似估计为带最小到达间隔时间的事件流。

基于策略的管理(Policy Based Management, PBM)是指一种用于管理网络和分布式系统的方法[11],它把系统的管理逻辑和应用逻辑相分离,并将管理逻辑表示为控制系统行为选择的策略。策略可以动态部署、更新或删除,因此,可以在不改变软件编码或停止系统运行的前提下,通过改变策略来支持系统行为的动态适应,这意味着可以通过动态更新由分布式实体解释的策略规则来改变它们的行为。基于策略的管理目前已经在访问控制、数据备份、网络安全、资源提供、配置检查和服务规划等领域得到了应用。通过基于策略的管理系统,管理员无需再逐个地手工配置网络设备或应用资源,而是通过预先定义的策略来实施业务规则和目标,从而提高了管理效率。IETF 策略工作组(IETF Work Group on Policy-Based Management)和伦敦帝国大学是基于策略的管理领域研究的突出代表,他们都提出了完整的基于策略的管理框架,并以此为基础研究了策略领域的几个基本问题。此外,贝尔实验室等在策略语言[12] 方面进行了深入的研究。以上研究组织都或多或少地涉及了策略冲突检测问题,其中 Ponder[13] 是一种基于角色的网络策略语言,角色由一组定义管理者权限和责任的策略组成,关系由一组定义了角色之间交互协议的策略组成,支持管理者之间进行一定的协作。但策略是无状态的 ECA 规则,计算能力弱,角色和关系是分开定义的,关系定义的交互消息要携带当前交互的状态信息,使策略的定义复杂。Rei[14] 所采用的策略引擎接受一阶逻辑和 RDF(Resource Description Framework)。它把策略的内容(称为策略对象)和策略执行主体分开,然后通过算子"has"将它们关联起来。这种方法具有很强的灵活性,允许描述不同种类的策略。

为了支持以可扩展的方式开展大规模用户行为仿真,避免集中式单点仿真存在的性能瓶颈,需要建立一种分布式的加载控制架构,实现面向多点协同的仿真业务流量生成。目前,基于策略的调度模型是描述分布式加载控制架构的一种有效机制,该模型通过在上层建立统一的任务描述策略,随后根据任务需求和底层资源支持情况将策略解析分配到不同的执行点进行执行。然而,现有的策略描述模型作为一个通用框架,需要结合用户行为流量仿真的需求进行优化定制。

基于上述原因,本文设计了一种基于 XML(eXtensible Markup Language)可扩展标记语言实现对调度策略的描述。XML 文档定义方式有:文档类型定义(DTD)和 XML schema。DTD 定义了文档的整体结构以及文档的语法,应用广泛并有丰富工具支持。XML schema 用于定义管理信息等更强大、更丰富的特征。XML 能够更精确地声明内容,方便跨越多种平台得到更有意义的搜索结果。它提供了一种描述结构数据的格式,简化了网络中数据交换和表示,使得代码、数据和表示分离,并作为数据交换的标准格式。

3 实验方案设计

根据 XML 文件的基本格式,本文设计了一种可扩展的策略文件,方便用户读取、执行策略文件信息。本文以部署 200 台 LXC 容器并调度其仿真 ftp 业务流量进行实验。策略文件格式如下所示,其中 < POLICY_PARA > 项代表策略的参数信息,包括 LXC 容器的数量、IP 地址、网关、用户行为、端口号等。

策略文件(.xml)

```
<TASKLIST>
  <TASK1>
    <TASK_NUM>任务编号</TASK_NUM>
    <TASK_NAME>任务名称</TASK_NAME>
    <POLICY_PARA>
    CONTAINER_NUM=200                    <!--容器数量-->
    CONTAINER_IP_RANGE=10.0.3.1/24
                                         <!--容器的网络号-->
    CONTAINER_GATEWAY=10.0.3.1
                                         <!--容器的网关-->
    USER_PASS_LIST=ftp_1                 <!--用户行为类型-->
    SERVICE_PORT=21                      <!--端口号-->
    …
    </POLICY_PARA>
  </TASK1>
</TASKLIST>
```

实验过程中,发现在进行 LXC 容器大规模部署的时候,遇到的主要瓶颈是如何减小容器对主机物理内存的占用以及时间成本。

3.1 物理内存的占用

为实现大规模的容器部署,必须要求每一台 LXC 容器占用的物理内存非常小。而通常情况下,创建 LXC 容器模板时,均使用 ubuntu 的镜像模板,例如:sudo lxc-create -n <container-name> -t <template>。如表 1 所示,几种常见的 Linux 版本镜像大小各不相同,最小的 Alpine 镜像只有 3 MB 左右,最大的 ubuntu-core 镜像有 200 MB 以上。ubuntu 不同版本的镜像模板大约在 70 ~ 90 MB 左右,有的甚至达到了 200 MB 以上。

表 1 Linux 镜像模板

镜像模板	版本	大小/MB
Alpine	3.6	3.07
centos	7	81.74
debian	10	89.55
ubuntu	16.04	80.32
ubuntu-core	16.04	245.84

如表 2 所示,通过实验对比统计,不同的 Linux 镜像对物理内存的占用相差很大。

表 2 Alpine 与 ubuntu-core 的内存占用对比

	50 台	100 台	150 台	200 台
Alpine/MB	153.5	307	460.5	614
ubuntu-core/GB	12	24	36	48

基于上述原因,本文采用了基于 Alpine 的 LXC 镜像模板。Alpine 系统是一个面向安全的轻型 Linux 发行版。它不同于通常 Linux 发行版,Alpine 采用了 musl libc 和 busybox 以减小系统的体积和运行时资源消耗,相比于 ubuntu 的 LXC 镜像,它的容量非常小,仅仅只有 3 MB 左右,占用更少磁盘空间,且拥有非常友好的包管理机制。

3.2 时间成本

为了能够快速地部署 LXC 容器,在实验中通过调用 Python 语言中 multiprocessing 模块的进程池,采用多进程的并行方式,实现了在较短时间内对容器的大规模部署。首先,创建好一个 LXC 模板,随后创建的容器都是基于该模板复制生成的。快速

部署 LXC 容器的伪代码如下：

Create Container-template

```
1;def create(i):
2:   sudo lxc-create -t download -n m0 --server mirror. tuna. tsinghua.
edu. cn/lxc-images -d alpine -r 3.6 -a amd64
3;def clone(i):
4:   sudo lxc-clone -n m{} -o m0
5;pool = Pool(40)
                    //进程池大小可以自定义,在本实验中设置为 40
6;for i in range(1,201):                    //创建 200 台 LXC 容器
7:   pool. apply_async(clone,args = (i,))
8;pool. close()
9;pool. join()
```

经过实验测试,发现如果采用传统的 ubuntu 镜像进行部署 200 台 LXC,不管是串行方式还是并行方式,都至少要花费半小时以上,如表 3 所示。

表 3　ubuntu 镜像的部署时间对比　(s)

	20 台	50 台	100 台	150 台	200 台
串行	97	590	1 298	2 110	3 786
并行	74	357	928	1 264	3 091

而当采用较小的 Alpine 镜像进行部署容器时,如表 4 所示,时间会大大减少,尤其是采用并行方式,大大降低了部署的时间成本。

表 4　Alpine 镜像的部署时间对比　(s)

	20 台	50 台	100 台	150 台	200 台
串行	46.6	115	230	346	464.4
并行	4.5	8.4	16.3	24.9	35.1

由图 2 可以更加直观地看出采用 Alpine 镜像并行方式部署的优势,用户进程是完全建立在用户控件的进程库,用户进程的创建、调度、同步和销毁全都在用户空间完成,不需要内核的帮助。因此这种多进程的调度机制是极其低损耗和高效的。

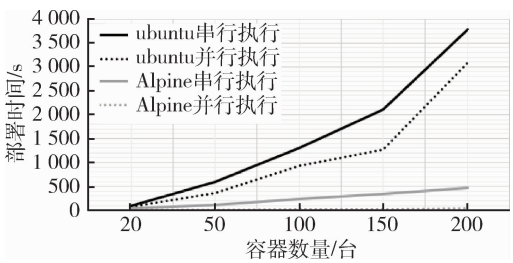

图 2　不同部署方式的时间成本

3.3　部署调度实现

根据以上的实验分析,通过 Python 语言编写程序,将策略文件中的参数进行读取、解析,最终实现仿真目标,如图 3、图 4 所示。

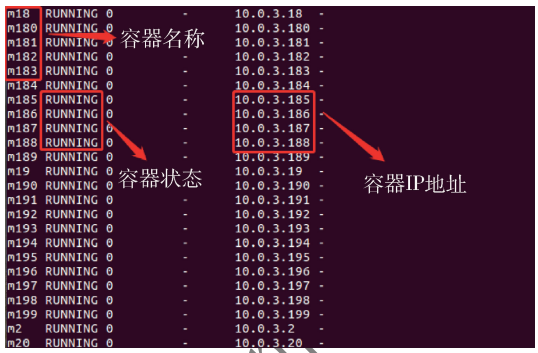

图 3　部署 200 台 LXC 容器

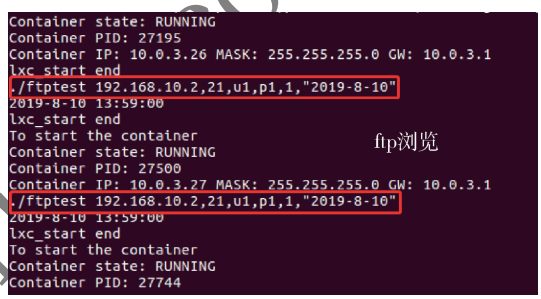

图 4　实现用户行为仿真

虽然在本实验中,仅仅讨论了 ftp 的用户行为仿真,但可以通过对任务文件中 < POLICY_PARA > 项参数的设定,实现诸如 Web 访问等对多种用户行为的仿真。

4　结论

随着未来网络的不断发展,大规模的“人机交互”行为仿真的需求也越来越大。如何在现有的条件下实现用户行为仿真的部署以及调度问题,已经成为了目前研究的热点。本文提出了一种基于策略的部署调度方式,并在部署规模和时间成本等问题上提出了自己的解决方案。

参考文献

[1] 方滨兴,贾焰,李爱平,等. 网络空间靶场技术研究[J]. 信息安全学报,2016,1(3):1-9.

[2] 刘鹏. 网络用户行为分析的若干问题研究[D]. 北京:北京邮电大学,2010.

[3] Linux containers - LXC - introduction [EB/OL]. (2017-02-16). https://linuxcontainers. org/lxc/introduction/.

[4] 周明耀. CGroup 介绍、应用实例及原理描述[DB/OL]. (2015-06-10)[2019-10-02]. http://www. ibm. com/de-

veloperworks/cn/linux/1506_cgroup/.
[5] 张辉宜,赵海军,周秀丽. 基于 Pfair 的分布式实时调度策略 Linux 下实现[J]. 计算机技术与发展,2008,18(2):31-33.
[6] HAMANN A,HENIA R,RACU R,et al. SymTA/S-Symbolic timing analysis for systems[C]. Proceedings of the 16th ECRTS. Catania:IEEE Press,2004:17-20.
[7] DROZDOWICZ M,GANZHA M,PAPRZYCKI M. Semantic policy information point-preliminary considerations[C]. ICT Innovations 2015. Springer International Publishing,2016:11-19.
[8] LÊ L S,WEGMANN A. Hierarchy-oriented modeling of enterprise architecture using reference-model of open distributed processing[J]. Computer Standards & Interfaces,2013,35(3):277-293.
[9] HAMANN A,JERSAK M,KAI R,et al. A framework for modular analysis and exploration of heterogeneous embedded systems[J]. Real-Time Systems,2006,33(1-3):101-137.
[10] BINI E,BUTTAZZO G,LIPARI G. Minimizing CPU energy in real-time systems with discrete speed management[J]. ACM Transactions on Embedded Computing Systems,2009,8(4):167-176.
[11] PRIETO A G,STADLER R. Adaptive real-time monitoring for large-scale networked systems[C]. IFIP/IEEE International Conference on Symposium on Integrated Network Management. IEEE Press,2009:790-795.
[12] LAZOUSKI A,MARTINELLI F,MORI P. Usage control in computer security: a survey[J]. Computer Science Review,2010,4(2):81-99.
[13] DAMIANOU N, DULAY N, LUPU E, et al. Ponder: a language for specifying security and management policies for distributed systems[R]. Imperial College Research Report,2000.
[14] DIMMOCK N. BELOKOSZTOLSZKI A. EYERS D, et al. Using trust and risk in role-based access control policies[C]. Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies,2004.

(收稿日期:2019-10-23)

作者简介:

纪一方(1992 -),男,硕士研究生,主要研究方向:计算机网络、虚拟化技术。

张国敏(1979 -),男,博士,副教授,主要研究方向:网络体系结构、网络空间安全。

邢长友(1982 -),男,博士,副教授,主要研究方向:网络体系结构、网络空间安全。

(上接第 77 页)

[6] 赵慧冬,黑勇,乔树山. OFDM 电力线通信系统的自动增益控制方法[J]. 科学技术与工程,2012,12(19):4559-4662.
[7] 何世彪,杨迷,张青,等. OFDM 电力线载波通信系统的 AGC 实现[J]. 世界科技研究与发展,2013,35(6):720-722.
[8] 鲍飞鸿,车珊,秦风,等. 基于 PID 方法的自动增益控制[J]. 太赫兹科学与电子信息学报,2016,14(1):112-116.
[9] 周春良,周芝梅,樊文杰,等. 电力专用宽带电力线载波通信芯片的设计与应用[J]. 微型机与应用,2017,36(11):34-36,43.
[10] 周春良,周芝梅,杨晓平,等. 宽带电力线通信芯片的低功耗设计[J]. 电子技术应用,2017,43(10):16-19.

(收稿日期:2019-09-25)

作者简介:

周春良(1977 -),男,硕士,高级工程师,主要研究方向:通信芯片与系统实现。

王连成(1968 -),男,硕士,高级工程师,主要研究方向:集成电路设计。

迟海明(1983 -),男,硕士,工程师,主要研究方向:电力线载波通信算法。

版权声明