



基于虚拟机的僵尸网络仿真系统可扩展性优化^{*}

阮利^{1,2}, 蔺波^{1,2}, 肖利民^{1,2}

(1.北京航空航天大学软件开发环境国家重点实验室 北京 100191;

2.北京航空航天大学计算机学院 北京 100191)

摘要:僵尸网络仿真是僵尸网络研究领域的一门新兴技术,近年来日益得到广泛的关注。现有研究中基于虚拟机的大规模僵尸网络实用仿真系统比较缺乏,现有系统缺乏对虚拟机集群的快速部署、多虚拟化(尤其是对轻量级虚拟化)、僵尸网络特性(如僵尸网络的昼夜随机关机开机)模拟、高可扩展性功能的支持。通过对僵尸网络仿真特性的分析,提出了一种基于虚拟机的僵尸网络仿真系统及适用于僵尸网络仿真系统的可扩展性优化技术。实验表明,所提出的基于内存性能优化和CPU性能优化的技术,可以使每个虚拟机的常驻内存比优化前减少77%以上,同一台物理机所能开启的最大虚拟机数量由15台增加到43台,当限制虚拟机的vCPU占用率到100 000时,主机CPU占用率在同样情况下能从100%降低到20%,优化效果显著。

关键词:僵尸网络仿真;虚拟机;可扩展性;性能优化

doi: 10.3969/j.issn.1000-0801.2013.10.013

Scalability Optimization of Virtualization Based Botnet Emulation Platform

Ruan Li^{1,2}, Lin Bo^{1,2}, Xiao Limin^{1,2}

(1.State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China;

2. School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

Abstract: Botnet emulation, a new technology to investigate Botnet characteristics, gains increasing focus. The practical virtualization based Botnet emulation platform is few and lacks support of rapid deployment, multiple virtualization, specific features of the Botnet like Botnets' circadian random shutdown and starting, and high scalability which will greatly influence the Botnet propagation emulation effect. A Botnet emulation system and its scalability improvement technologies were introduced. Experiments show that the memory performance optimization and CPU performance optimization reduce more than 77% of host memory per virtual machine, the maximum virtual machines one physical machine can starts up increase from 15 to 43; the host CPU utility is deduced from 100% to 20% when the vCPU's performance is restricted to 100 000.

Key words: Botnet emulation, virtual machine, scalability, performance optimization

^{*} 国家高技术研究发展计划(“863”计划)基金资助项目(No.2011AA01A205),国家自然科学基金资助项目(No.61003015, No.61232009),软件开发环境国家重点实验室探索性自主研究课题资助项目(No.SKLSDE-2012ZX-23),北京市自然科学基金资助项目(No.4122042),核心电子器件、高端通用芯片及基础软件产品重大专项基金资助项目(No.2010ZX01036-001-001)

1 引言

随着分布式技术和网络技术等的发展,僵尸网络(Botnet)^[1]已成为当今世界计算机安全的最大威胁之一。僵尸网络是指攻击者首先借助于网络等传播途径将僵尸程序植入受害主机内,然后通过执行恶意程序取得大量主机的控制器并达到攻击目的(如信息窃取、分布式拒绝服务攻击和垃圾邮件发送)的恶意网络行为。由于其采用一对多的命令与控制信道,相对于现有流行的恶意代码(如蠕虫、木马、病毒等),其危害性更大。

传统僵尸网络的研究方法主要可分为 3 种:对僵尸程序剖析、大范围的追踪测量研究以及 Botnet 模拟建模分析。随着 fast-flux、P2P 技术的引入,传统方法的隔离性和可扩展性等缺陷日益突出。僵尸网络仿真应运而生,主要基于虚拟化技术搭建仿真网络环境平台。Botnet 仿真研究相比于传统的研究方法,能够提供更加安全、准确的研究环境和可控制的实验参数,降低对真实环境的干扰和法律风险。

然而,现有研究中基于虚拟机的大规模僵尸网络的实用仿真系统比较缺乏,且现有系统缺乏对虚拟机集群快速部署、多虚拟化(尤其是对轻量级虚拟化)、僵尸网络特性(如僵尸网络的昼夜随机关机开机)模拟、高可扩展性功能的支持。本文通过对僵尸网络的仿真特性进行分析,提出了一种基于虚拟机的僵尸网络仿真系统及适用于僵尸网络仿真系统的可扩展性优化技术。

2 相关研究

僵尸网络的研究总体包括僵尸程序剖析、大范围追踪检测研究、僵尸网络模拟建模、僵尸网络仿真,具体见表1。基于开放环境和封闭框架研究僵尸网络是僵尸网络研究的两条重要技术路线。早在僵尸网络研究之初,就有人倡

导鉴于僵尸网络的特性,需要建立封闭的框架研究僵尸网络,如 John^[2]创建的 Botlab 僵尸网络仿真系统。但由于该系统需要连接实际网络,不能称为封闭网络。此外,现有技术最大的缺点是无法提供可靠性、真实性和可扩展性。为了提供更真实的仿真僵尸网络环境,2009 年,圣迪亚国家实验室^[3]成功实现了在戴尔的雷鸟超级计算机上运行上百万个虚拟机,实现仿真。加拿大和法国的科学家^[4]通过在 98 个服务器集群上加载 3 000 个 Windows XP 副本,并基于此平台对 Waledac 僵尸网络进行仿真,充分展示了僵尸网络仿真系统的优越性。随着虚拟化和云计算技术的发展,研究者提出利用虚拟化技术构建僵尸网络。Barford P^[4]等人于 2007 年建立了基于 Emulab 网络仿真平台的僵尸网络评估平台(BEE),该平台基于 VMware 服务器,在一个主机上安装 5 个客户操作系统,完成了 500 个僵尸主机规模的僵尸网络仿真。

然而,僵尸网络仿真^[4-9]研究通常需要成百上千的虚拟机群规模,过去的仿真研究常借助虚拟机群管理平台作为仿真平台(包括 Eucalypus、OpenStack、CloudStack、XenServer、Convirt),利用集群管理工具(如 xCat 以及 xCPU 等)辅助实现仿真研究,但由于仿真研究的特殊性,现有虚拟机群管理平台并不能满足僵尸网络仿真系统的需求,存在管理平台搭建复杂、支持的 VMM 类型单一、无法实现虚拟机群的统一快速管理、可扩展性不足等问题。本文针对现有研究存在的问题,提出一种基于虚拟机的僵尸网络仿真系统及适用于僵尸网络仿真系统的可扩展性优化技术。

3 基于虚拟机的僵尸网络仿真系统

本文的僵尸网络仿真系统架构如图 1 所示,基于 Libvirt 构建的集中式僵尸网络仿真系统,由一个管理控制节点及多个代理节点构成,核心由中心服务器、代理节点程序、Libvirt 守护进程(Libvirtd) 3 个部分协调完成。基于

表 1 僵尸网络研究方法对比

研究方法	技术特征	虚拟化	适用个体	环境	性能	平台构建
僵尸程序剖析	二进制代码分析工具、逆向工程	否	僵尸机个体行为	开放	分析难度大	容易
大范围的追踪检测	蜜网	否	僵尸群及僵尸机个体	开放	追踪检测分析时间,周期长	复杂
僵尸网络建模	数学建模	否	僵尸群某一特定网络特点,具有预测能力	开放	僵尸行为研究单一	复杂
僵尸网络仿真	虚拟化	是	僵尸群及僵尸机个体行为,研究周期短,可反馈实验	封闭,完全隔离	支持 VMM 类型较单一;不支持统一快速配置管理	搭建部署繁琐;无仿真辅助功能



僵尸网络的虚拟机群具有同时操作的技术特征,僵尸网络中心服务器作为通信控制管理中心,将对网络、存储以及虚拟机配置相关管理操作统一集成到网络配置模块、存储配置模块以及模板配置模块中,并以物理节点为基本管理控制单元,僵尸网络仿真环境构建的复杂度得到了显著降低。

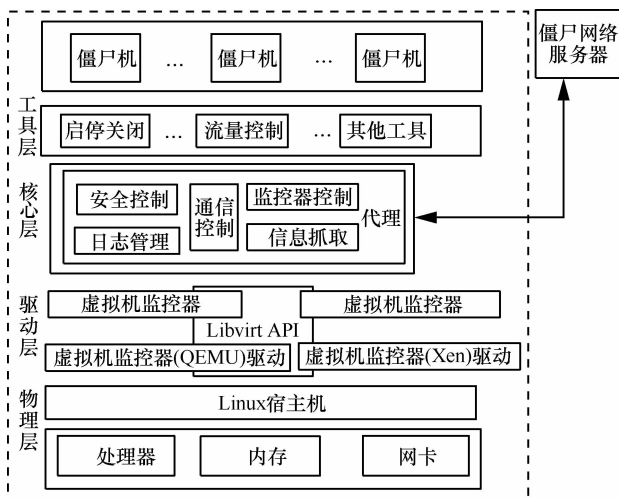


图1 僵尸网络仿真系统总体架构

其中,僵尸网络仿真系统中心服务器是控制管理中心,完成僵尸网络仿真平台初始化以及多个代理节点、存储资源池和虚拟机模板的添加功能。代理节点作为多个物理主机的代理,主要实施中心服务器发送的命令。这些命令包括虚拟机群的创建、启动、关闭、销毁、添加以及添加共享资源池,实施时区特性模拟,监控虚拟机流量等。Libvirt在每个代理节点上,接收 Libvirt API 调用,通过调用相应的 VMM 驱动,实现具体的虚拟机操作功能。如图2所示,系统模块层次主要包括驱动层、核心层和工具层。

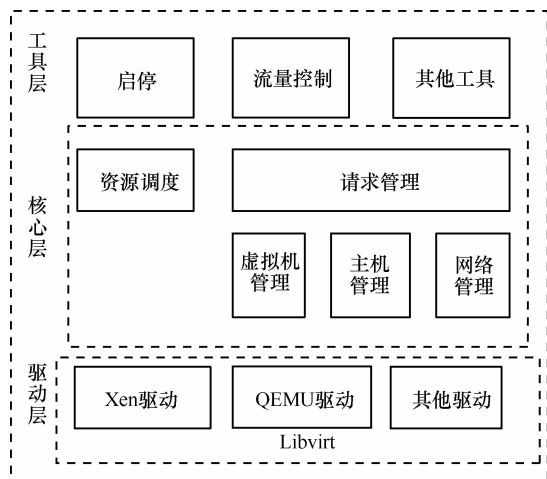


图2 僵尸网络仿真系统模块层

- 驱动层:位于核心层的代理节点以及中心服务器都安装 Libvirt 作为驱动层,可根据需要安装一款或几款 VMM。
- 核心层:共享资源池除提供上述存储功能外,还具有模板存储功能,用于存储分发虚拟机模板。代理节点上安装 VMM、Libvirt 以及代理程序。代理程序调用 Libvirt 实现虚拟机的创建、启动、关闭、销毁等功能。
- 工具层:主要实现僵尸网络仿真过程中需要的一些基本功能,如仿真网络环境中的物理主机昼夜随机启动关闭、流量控制、特性模拟等功能。

4 适用于僵尸网络仿真系统的可扩展性优化技术

本节介绍本文所提出的基于内存性能优化的集群密度可扩展性优化方法和基于 CPU 性能优化的集群密度可扩展性优化方法。

4.1 基于内存性能优化的集群密度可扩展性优化

在低负载环境下,物理内存占用率是限制虚拟机数量扩展的主要性能瓶颈,每个虚拟机创建后,为维护每台虚拟机正常运行所消耗的常驻内存是造成内存消耗的关键因素,因此提出减少每个虚拟机常驻内存的方法,进而实现虚拟机集群的密度扩展优化。

4.1.1 基于精简文件系统的集群密度可扩展性优化

基本思路为:将常规的虚拟机镜像替换为简化的文件系统,在虚拟机启动方式上以内核+文件系统的方式启动。基本实现方法如下。

算法1 基于精简文件系统的集群密度可扩展性优化

ModifyFile (initrd,busybox); //基于 busybox 工具修改 initrd 文件。

CreateSimplifiedFileSystem (ramdisk); //制作 ramdisk 精简文件系统,供虚拟机使用。

AddtoFileSystem (BotnetProgram); //制作文件系统时,在文件系统中加入僵尸程序,使得僵尸程序能在文件系统中运行。

StartupVM(kernel model); //以内核加根文件系统的启动方式启动虚拟机。

Clone(expect.script); //完成基于简化文件系统的克隆功能。由于 virt-clone 不支持对简化后的文件系统进行克隆,基于交互脚本 expect 对优化后的虚拟机文件系统进行

克隆。

ModifyVMConfig(); //修改虚拟机配置文件。

ScalabilityOptimization (SimplifiedFileSystem); //实现基于精简文件系统的集群密度可扩展性优化。

4.1.2 基于页共享的集群密度可扩展性优化

集成并实现了一种适用于僵尸网络仿真系统的 KSM (kernel shared memory) 技术,通过开启 Linux 内核的 KSM 特性对内存性能进一步优化,使得如果 VMM 发现在不同虚拟机上存在完全相同或重复的内存页面,将会在这些虚拟机之间共享页面,并建立这些内存数据块的指针,以提高性能。

算法 2 基于页共享的集群密度可扩展性优化

OpenConfig (KSM); //基于默认的内核配置,开启 Linux 操作系统 KSM 属性。

CreateControlChanel (CenterServer); //中心服务器通过建立并发送管理控制信道,控制每个代理节点 KSM 特性的开启与关闭。

RunDemonProcess (Agents,KSM); //每个代理节点中的 KSM 作为内核中的守护进程存在。

ScanPage (VMtemplate); //定期对虚拟机镜像模板执行页面扫描。

IdentifyAndMergeCopy (VM); //识别虚拟机镜像副本页面并合并虚拟机镜像副本。

MarkMemory (COW); //将合并后的内存标记上 COW 标识。

基于单一页面的僵尸网络仿真系统 KSM 方法的流程如图 3 和图 4 所示。该技术采用两个“红—黑”树管理的方法,其中一棵为不稳定红—黑树,主要存储还不能理解为稳定的虚拟机镜像页面;第二棵树为稳定红—黑树,主要存储已经被确认是稳定且合并后的虚拟机镜像页面。

4.1.3 基于 Vmalloc 空间优化的集群密度可扩展性优化

同一个物理节点上,各个虚拟机以独有方式占用 Vmalloc 线性地址空间,此外,Vmalloc 每次得到的 VADDR 空间中要保留一个页面大小的空间,这导致当 Botnet 中创建的虚拟机数量显著增加时,Vmalloc 的线性地址空间严重限制虚拟机的规模扩展。

采用基于精简文件系统和基于页共享的优化技术后,能够将每个虚拟机的常驻内存缩减到 34 MB,这样理论上能够在 3 GB 内存的单物理节点上,开启 80 台左右虚拟机的效果。为了逼近理论值,进行的测试实验如图 5 所示。

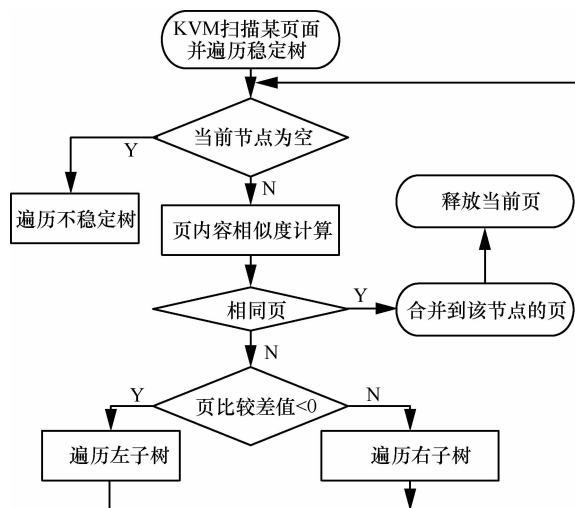


图 3 基于 KSM 的虚拟机镜像页面扫描流程 (遍历稳定子树)

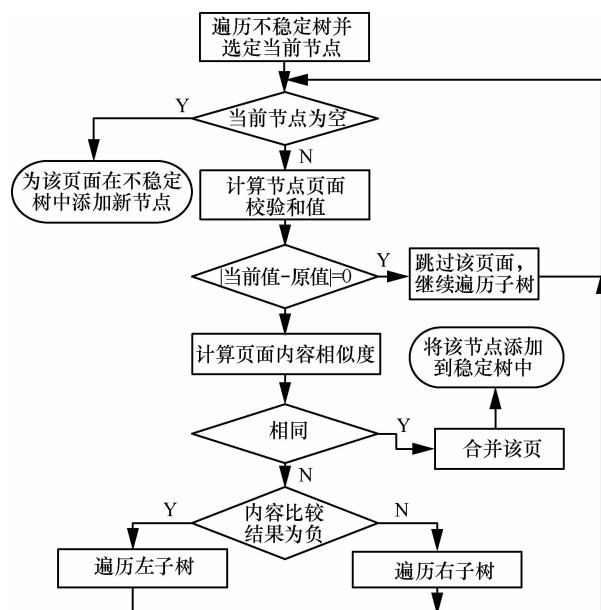


图 4 基于 KSM 的虚拟机镜像页面扫描流程 (遍历不稳定子树)

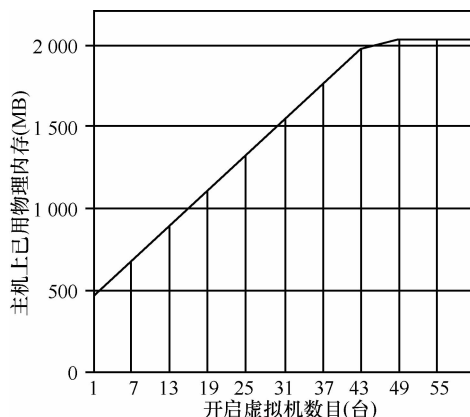


图 5 基于精简文件系统与 KSM 优化后虚拟机群物理内存使用情况



结果显示,当单物理节点上创建 43 台虚拟机时,会报创建虚拟机失败错误,但当前情况下的物理内存还远未达到 3 GB 的理论上限,因此仍然存在优化空间。

虚拟机群 Vmalloc 内存使用情况如图 6 所示。每台虚拟机固定占用 1 464 KB 的 Vmalloc 空间,在虚拟机数目达到 43 台时,由于 Vmalloc 已没有实际可用空间分配,从而无法继续创建虚拟机。基于对虚拟机创建失败的错误信息以及 KVM 的内存虚拟化机制的分析发现,基本实施流程介绍如下。

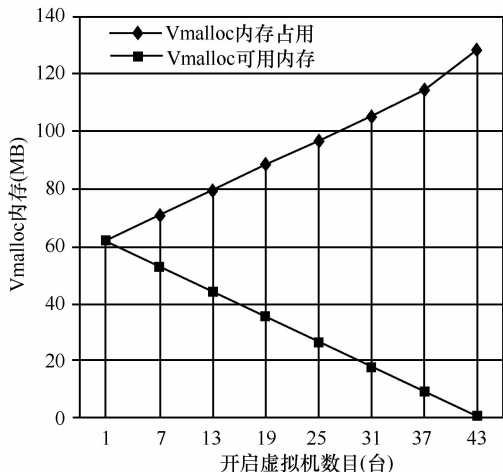


图 6 虚拟机群 Vmalloc 内存使用情况

RecordAddressMap (kvm_memory_slot); //KVM 的内存虚拟化使用 kvm_memory_slot 数据结构,记录每一个地址区间的映射关系。

cpu_register_physical_memory(); //cpu_register_physical_memory 函数实现内存注册。

cpu_register_physical_memory(); //cpu_register_physical_memory 函数调用系统调用函数 kvm_vm_ioctl_set_memory_region 进入内核。

在内核态下由 kvm_set_memory_region 函数实现 kvm_memory_slot 数据结构的实例化。

上述过程中,实际分配相应数据结构完成该虚拟机的物理内存地址到宿主虚拟地址的转化,由内核根据设置的虚拟机内存的大小实现。可见,每个虚拟机的 Vmalloc 空间占用情况与虚拟机在创建时设置的虚拟内存值密切相关。在 x86 架构下,Vmalloc 空间由系统预留了 128 MB 的空间用于调用,根据 Vmalloc 空间与虚拟机内存值的变化关系,得出 x86 系统所能支持的虚拟机理论最大数量与虚拟机内存值的变化关系,如图 7 所示。在保证虚拟机创建

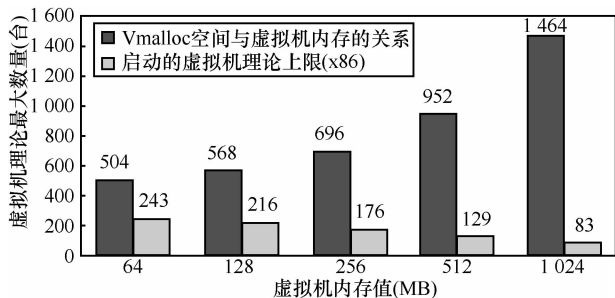


图 7 Vmalloc 空间占用、虚拟机数量与虚拟机内存设置的变化关系

成功的前提下,可将虚拟机内存设置到最大 1 024 MB,理论上将获得最多支持 83 台的优化效果。

4.1.4 基于 ballooning 优化的集群密度可扩展性优化

内存 ballooning 技术是当前虚拟机的常用技术,主要借助于驱动的帮助,通过将未使用的内存从当前非活跃的虚拟机中进行回收,得到物理内存利用率提高的效果。但 ballooning 还存在不足,如基于客户机操作系统实现对 virtio_balloon 驱动的加载,缺乏自动化的 ballooning 管理、执行等机制,这给基于虚拟机的僵尸网络仿真平台的部署带来了巨大的工作量。此外,由于僵尸网络群还具有类似于僵尸网络的长期运行和等待、瞬时性、突发性等待等特性,可以基于 ballooning 技术研究虚拟机群密度扩展性优化。基于 QEMU/KVM 的僵尸网络仿真平台的 ballooning 优化技术,实现的关键是加载 virtio_balloon 驱动。利用 Libvirt 提供 ballooning 支持的优势,通过在配置虚拟机模板中添加代码<devices> <memballoon model='virtio'/></devices>的方式实现。

另一条技术路线是基于 Linux 中最为成熟的基于内存交换的内存扩容技术,这种技术可以实现希望得到的过载利用内存效果,即通过将虚拟机的内存数据进行交换,充分利用交换空间技术弥补内存自身空间的不足。

4.2 基于 CPU 性能优化的集群密度可扩展性优化

在高负载环境下,CPU 是限制虚拟机数量扩展的主要瓶颈,而网络负载是影响仿真效果的主要瓶颈,下面分别从每个虚拟机的网络负载及 CPU 占用率角度,介绍本文基于 CPU 性能优化的集群密度可扩展性优化方法,具体如下。

(1) 基于单虚拟机 vCPU 数目限制的可扩展性优化方法

当前的虚拟机监控器(KVM、Xen 等)都能提供对多个 vCPU 的支持,因此单个虚拟机被指派的 CPU 数目可以不用等于或小于物理 CPU 数量,突破多余物理主机的物理

CPU 数量的限制,以提高 CPU 的利用率。由于僵尸网络仿真系统的 agent 需要在各自封闭的环境下运行支持,仿真平台需要尽可能多地开启虚拟机,如果允许每个客户机运行的 vCPU 数量过多,将使得该物理主机的 CPU 性能明显下降,因此本系统限制单虚拟机的 vCPU 数量为 1,即<VM, vCPU>间以 1:1 的映射关系进行设置。

(2)基于虚拟机网络流量限制的可扩展性优化方法

在高负载情况下,CPU 占用率高的核心影响因素之一为:僵尸网络仿真系统中的每个虚拟机都会最大限度地发挥处理能力,即用最大的处理能力向网络发送相应的数据分组,这使得网络和 CPU 负载更高,虚拟机数量无法有效扩展。因此本文的仿真系统集成了现有的网络流量控制工具 TC,通过实时监控虚拟机网络流量,并设定相应的阈值,限制虚拟机网络流量,进而提高虚拟机群密度的扩展性。

(3)基于虚拟机 vCPU 性能限制的可扩展性优化方法

应用基于虚拟机网络流量限制的可扩展性优化方法后,整体网络流量已经取得较大幅度的降低,但由于虚拟机的 CPU 性能和数据分组的发送速率基本不变,各宿主机的物理 CPU 占用率不能较大幅度地降低。本文提出一种基于进程管理组(如 cGroups)的虚拟机 CPU 性能约束方法,以降低中断次数,进而提高可扩展性。具体利用 Libvirt 对 cGroups 资源管理控制策略的支持性能,在虚拟机模板的配置文件中对 cGroups 资源管理控制策略进行设置,需设置的宿主机 CPU 占用率参数如下所示:

```
<cpupune>
<period>1000000</period>
<quota>-1</quota>
<emulator_period>1000000</emulator_period>
<emulator_quota>-1</emulator_quota>
</cpupune>
```

各命令的具体含义见表 2。

表 2 cGroups 资源管理控制策略命令含义

命令	含义
period 与 quota	限制虚拟机 CPU 的资源占用
period	虚拟机 CPU 运行间隔
quota	虚拟机在一次运行时所能持续的最长时间
Emulator_quota 与 emulator_quota	限制虚拟机除 CPU 外其他线程的资源占用

本文主要用 period 与 quota 命令配合管理控制虚拟机的 vCPU 资源,实现单物理节点上虚拟机群的密度扩

展优化。

5 系统测试与分析

5.1 测试环境

测试环境的硬件配置(代理节点)和软件配置见表 3。软件测试环境由底层宿主系统、Libvirt 抽象层、中间宿主 QEMU-KVM 进程以及代理节点程序(或者中心服务器程序)构成。

表 3 软硬件环境

硬件环境		软件环境	
CPU	Intel Xeon 5160 3 GHz 双核 4 线程	宿主 OS	Ubuntu Server 12.04 LTS
		宿主 OS 内核	Linux 3.2.0-rc3
内存	DDR2 667 MHz,3×1 GB	Libvirt	Libvirt 0.9.8
网卡	Intel 80003ES2LAN Gigabit (e1000e)	Botserver/ Botclient	V1.0

5.2 测试结果及分析

5.2.1 基于内存性能优化的集群密度可扩展性优化测试

基于精简文件系统的集群密度可扩展性优化方法、基于精简文件系统和页共享结合的集群密度可扩展性优化方法与优化前方法的 Ubuntu 标准单虚拟机常驻内存占用对比如图 8 所示。对比优化前,分别减少了 53.3%以上和 77%以上。

基于内存优化技术所开启的最大虚拟机数对比如图 9

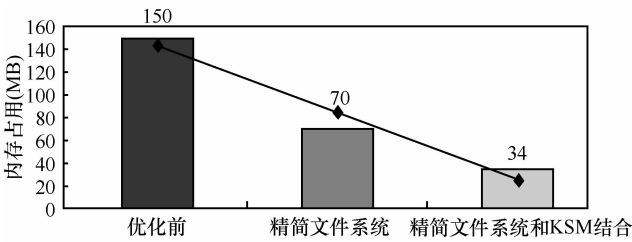


图 8 基于内存优化的可扩展性优化技术比较

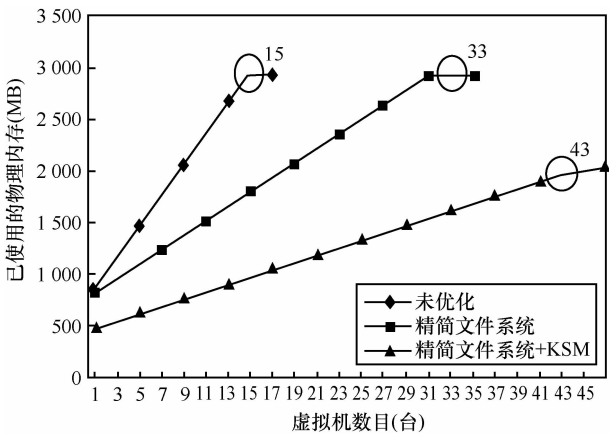


图 9 内存优化技术与开启最大虚拟机数量关系的对比



所示。使用基于精简文件系统的集群密度可扩展性优化方法、基于精简文件系统和页共享结合的集群密度可扩展性优化方法后,同样一台物理机所能开启的最大虚拟机数量由 15 台增加到 43 台。

当虚拟机内存设置为 128 MB、256 MB、512 MB、1 024 MB 时,Vmalloc 内存优化和所能开启的最大虚拟机数量测试结果如图 10 所示。在内存为 512 MB 时,基于 Vmalloc 空间优化能维持最大的虚拟机开启数目,所取得的规模扩展性优化效果最为显著,单物理节点上所能开启的最大虚拟机数量可扩展至 170 台左右。

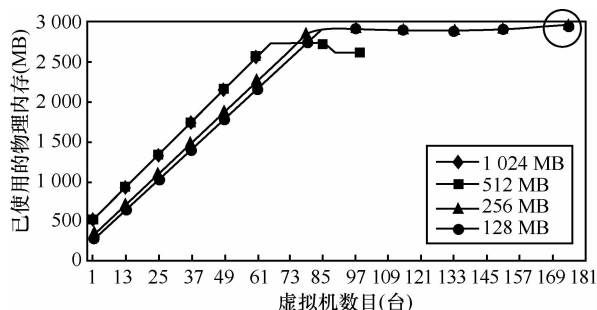


图 10 Vmalloc 优化技术对开启最大虚拟机数量的影响

5.2.2 基于 CPU 性能优化的集群密度可扩展性优化测试

本实验立足于降低虚拟机的分子发送速率,基于虚拟机内部的流量限制方法进行测试,KVM 进行 1 min DDoS 攻击性能指标测试的对比如图 11 所示。实施基于流量控制的性能优化后,网络负载及虚拟机退出次数大大下降,但 CPU 性能改进效果并不理想。

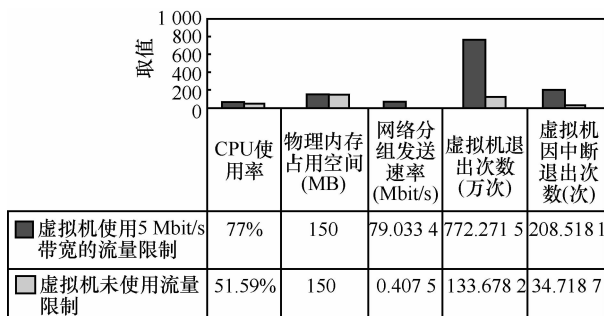


图 11 基于流量限制的物理 CPU 占用率

基于 cGroups 的性能优化结果如图 12 所示。 $vCPU_quota$ 的取值范围在 (1 000, 18 446 744 073 709 551) 或者负数,默认情况下是 -1,表示虚拟机使用无限制的资源,单位为 s。分别测试 $vCPU_quota$ 值从 1 000 000 到 100 000 的效果。如图 12 所示,当限制虚拟机的 $vCPU$ 性能 (即设置 $vCPU_quota$ 到 100 000) 时,性能改进效果相对显著,即物理机 CPU 占用率

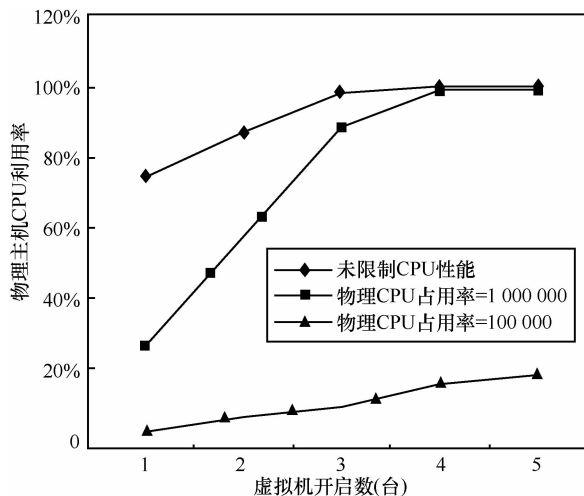


图 12 基于 cGroups 资源控制的物理 CPU 占用率

能在同样情况下降低到 20%。当 $vCPU_quota$ 值继续下降时,虚拟机可能因得不到 CPU 资源而宕机。

5.3 僵尸网络仿真系统对比

对比现有僵尸网络与基于 Libvirt 的僵尸网络仿真系统 (BAEP),对比结果见表 4。本文的仿真系统能有效解决基于虚拟机的僵尸网络仿真系统存在的缺乏,对虚拟机集群的快速部署功能,对多虚拟化 (尤其是对轻量级虚拟化)、对僵尸网络特性 (如僵尸网络的昼夜随机关机开机) 模拟的支持等问题,具有非常好的可扩展性。此外,由于本文所提出的优化方法总体针对虚拟化技术和仿真平台本身,在可扩展性上,对其他基于虚拟机的僵尸网络仿真系统具有实用性和可推广性。

6 结束语

本文针对僵尸网络的仿真特性,提出一种基于虚拟机的僵尸网络仿真系统及其可扩展性优化新技术。该系统的整体通信架构采用集中式的管理控制,包括驱动层、核心层和工具层。针对现有研究中僵尸网络仿真系统在可扩展性方面存在的问题,提出了基于内存性能优化和 CPU 性能优化的集群密度可扩展性优化方法,实验表明,每台虚拟机的常驻内存相比优化前减少 77% 以上,同样一台物理机所能开启的最大虚拟机数量由 15 台增加到 43 台,当限制虚拟机的 $vCPU$ 占用率到 100 000 时,效果相对显著,物理机 CPU 占用率在同样情况下能降低到 20%。通过同现有系统的综合对比可见,本文系统能有效解决传统的基于虚拟机的僵尸网络仿真系统存在的缺乏,如对虚拟机集群的快速部署功能,对多虚拟化 (尤其是对轻

表 4 僵尸网络仿真系统对比

名称	硬件配置	控制系统	支持 VMM	虚拟机数目 (台)	实验与否	仿真真实网络昼夜特点	工具
BEE	Emulab 全球测试平台中 15 个节点	Emulab	VMware Server	上百	是	无	Tcpdump
Sandia Library	Thunderbird 4 460 个节点	vMatic	Lguest, KVM	百万	否	无	无
Joan.etc	机群(98 个 blade)	xCat	VMware ESXi, KVM	上千	是	无	手工编写
BAEP	普通硬件资源 (几个至十 几个)	BAEP	VMware ESX and GSX, KVM, Xen, OpenVZ,UML, VirtualBox	几百到上千	是	有	有

量级虚拟化)、对僵尸网络特性(如僵尸网络的昼夜随机关机开机)模拟的支持等。下一步的工作包括优化现有仿真系统并集成到 Libvirt 工具集中以及将一些检测分析工具集成到现有系统中,进一步提高整个僵尸网络仿真系统的可用性。

参考文献

1 Zhu Z S, Lu G H, Chen Y. Botnet research survey. Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference, Turku, 2008: 967~972

2 Damballa G O. Botnet communication topologies: understanding the intricacies of Botnet command-and-control. <http://www.technicalinfo.net/papers/BotnetCommunicationTopologies.html>

3 Kanich C, Kreibich C, Levchenko K, *et al.* An empirical analysis of spam marketing conversion. Proceedings of the 15th ACM Conference on Computer and Communications Security, ACM, New York, NY, USA, 2008: 3~14

4 Calvet J, Davis C R, Fernandez J M, *et al.* The case for in-the-lab Botnet experimentation: creating and taking down a 3000-node Botnet. Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC'10), New York, NY, USA, 2010:141~150

5 Benzel T, Braden R, KIM D, *et al.* Testbeds and research infrastructures for the development of networks and communities. Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Barcelona, 2006:10~30

6 Dagon D, Gu G F, Lee C P, *et al.* A taxonomy of botnet structures. Proceedings of Twenty-Third Annual Computer Security Applications Conference, Seoul, Korea, 2007: 325~339

7 Ruan L, Xiao L M, Zhu M F. Content addressable storage optimization for desktop virtualization based disaster backup

storage system. China Communications, 2012,9(7):1~13

8 Jackson A W, Lapsley D, Jones C, *et al.* SLINGbot: a system for live investigation of next generation botnets. Proceedings of IEEE Conference for Homeland Security, Cybersecurity Applications and Technology, Washington, DC, USA, Mar 2009: 313~318

9 Ruan L, Wang H X, Xiao L M, *et al.* Memory virtualization for MIPS processor based cloud server. Advances in Grid and Pervasive Computing, 2012(7296):54~63

10 蔺波. 基于虚拟机的僵尸网络仿真平台的研究与实现. 北京航空航天大学硕士学位论文, 2012

[作者简介]



阮利,女,博士,北京航空航天大学计算机学院讲师,主要研究方向为计算机系统软件、系统虚拟化、分布式与并行存储、高性能计算机。



蔺波,男,北京航空航天大学硕士研究生,主要研究方向为虚拟化、僵尸网络仿真。



肖利民,男,博士,北京航空航天大学计算机学院教授,主要研究方向为计算机系统结构、虚拟化与云计算、高效能计算机与高端服务器。

(收稿日期:2013-06-28)