

基于 CIM 建模的统一虚拟系统管理模型

方 伟, 吴 杰, 高国钊

(复旦大学计算机科学技术学院, 上海 200433)

摘 要: 虚拟化产品的多样性增加了虚拟系统管理和调度的复杂性。为此, 构建一种基于 CIM 建模的统一虚拟系统管理模型, 该模型可解决异构虚拟化平台下的统一管理问题, 支持标准化接口的服务提供者, 具备实时监测系统状态的资源订阅模块以及自主管理调度功能。测试结果表明, 该模型具有较好的虚拟系统统一管理能力和较强的伸缩性和动态扩展性。

关键词: 虚拟系统管理; 公共信息模型; 自主管理调度; 动态扩展性

Unified Virtual Systems Management Model Based on CIM Modeling

FANG Wei, WU Jie, GAO Guo-zhao

(School of Computer Science, Fudan University, Shanghai 200433, China)

【Abstract】 The diversity of virtualization products brings the complexity of virtual systems unifying management and scheduling. This paper proposes a unified virtual systems management modeling based on Common Information Model(CIM), which solves the unified management issues of heterogeneous virtualization platform, supports the standard service provider, implements the resource subscription module and self-management ability. Meanwhile, by using and modifying of some open source projects, the model is implemented, and it shows that the model has good ability of the unified management of virtual systems as well as flexibility and dynamic extensibility.

【Key words】 virtual systems management; Common Information Model(CIM); self-management schedule; dynamic expansibility

DOI: 10.3969/j.issn.1000-3428.2011.16.091

1 概述

随着计算机软硬件技术水平的提高, 企业系统的硬件使用效率普遍偏低, 使用虚拟化技术在一台物理机上运行多个虚拟操作系统能有效地提高服务器的利用率, 降低企业 IT 运营成本, 并将各种资源整合成虚拟资源池而根据业务需要进行动态配置成为未来趋势, 可见, 虚拟化正成为企业未来绿色数据中心以及云计算平台的重要基础。

然而, 随着虚拟化技术的发展, 虚拟系统的应用越来越多, 不同厂商针对自己的产品提出不同的标准和接口, 这使得企业对不同虚拟系统产品的统一管理有困难。为解决企业信息系统中受管资源对象的统一管理问题, 标准化组织分布式管理工作组(Distributed Management Task Force, DMTF)提出以公共信息模型(Common Information Model, CIM)为基础的 WBEM(Web-based Enterprise Management)管理框架标准。它是一系列创新管理标准和基础互联网标准的集合, 旨在构建对于各种分布式异构计算环境的统一管理。

采用 WBEM 规范对网络资源管理已经有很多应用, 如 The Open Group 的 OpenPegasus 就是一个 WBEM 的开源实现, 但它侧重于设备的管理, 缺少异构平台下虚拟系统统一管理的特征。文献[1]中同样存在该问题。在虚拟系统管理领域中, 系统的实现主要针对同类虚拟系统, 很少涉及异构虚拟化平台下的统一管理。

虚拟化是未来大型数据中心和企业计算环境的重要基础, 同一个企业内部可能采用多个厂家生产的不同虚拟系统, 这就使异构虚拟平台下的虚拟系统统一管理需求随之增高。本文提出一种基于 CIM 建模的虚拟系统统一管理模型, 为各种异构虚拟化平台下的虚拟系统进行建模, 设计了服务提供者, 订阅系统相关资源信息, 从而达到自主管理调度。

2 虚拟系统管理模型设计

基于 CIM 建模的虚拟系统管理简化了不同虚拟平台下的虚拟系统管理, 能够以统一的方式对不同厂家的虚拟产品进行统一管理。将其应用到虚拟系统管理模型中能够降低系统实现的难度。

采用 CIM 建模能够使管理者以统一的方式获取不同的受管对象; 利用可扩展标记语言(eXtensible Markup Language, XML)作为其信息模型和表示方式, 使得该结构与管理平台无关, 能够方便地使用其他信息模型。

2.1 模型结构

管理模型需要解决的问题有很多, 如资源的动态调度分配^[2], 以及资源管理模型^[3]和虚拟化管理操作环境^[4]等。

在虚拟系统管理模型中各层功能如下:

(1)Self-Manager 层。该层负责发布管理命令, 同时接收来自 Server 层的反馈。主要由功能管理和策略库组成。它负责管理虚拟系统的生命周期、动态扩展等功能。不需要知道它的指令将被如何处理, 甚至可能并不知道信息提供者的存在, 只需要将命令通过接口发布出去, 对于反馈信息, 同样通过接口接收处理。它是一个自动监测和自动管理节点, 可以根据收集的输入信息发布一些执行动作。

(2)Server 层。该层就像处理器, 是命令真正的处理者, 负责命令的解释、转换和执行, 可以进行指示、订阅等功能。

基金项目: 上海市科委科研项目计划基金资助项目(10511501504); 上海市重点学科建设基金资助项目(B114)

作者简介: 方 伟(1987—), 男, 硕士研究生, 主研方向: 虚拟现实, 云计算; 吴 杰, 副教授、博士; 高国钊, 硕士研究生

收稿日期: 2011-03-04 **E-mail:** flywingsfang@gmail.com

它向上需要接收 Self-manager 层发出的管理信息，经过处理后进行执行，同时需要将结果返回。向下需要对 Resource，即被管理的资源进行实际的操作。它通过 Provider Manager 与 Provider 进行通信获得被管理资源的信息。

(3)Provider 层。该层为了让 Server 层的管理操作能够暴露在一个特定的资源上。它为一个特定资源的接口和 CIM 接口之间提供相关映射。简单地说,该层是与被管理对象通信的接口,用以存取各种数据,或是在某一被管理对象被触发事件时,返回适当地响应。它是 Server 层和 Resource 层之间的桥梁。

(4)Resource 层,即被管理的资源。这里指特定的虚拟系统,如 xen 或 kvm 虚拟平台下的虚拟系统,通常一台物理机上有多个虚拟系统,用于相同或不同服务。这些虚拟系统需要通过建模被统一管理。

虚拟系统管理模型结构如图 1 所示。

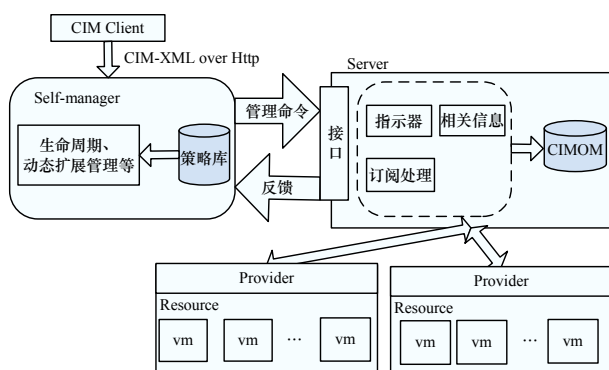


图 1 虚拟系统管理模型结构

2.2 模型基础设计

2.2.1 虚拟系统 CIM 建模

CIM 开发类似于面向对象开发,通过使用继承扩展展现有的标准类,首先设计开发类图,创建类的语言描述以及类的关联,实现类,最后部署并使用类实例。虚拟系统建模^[5]过程如下:用 CIM 建模技术为虚拟系统建模;为类和类之间的关联产生托管对象格式(Managed Object Format, MOF);为要求动态数据更新的类构建动态提供者;将 MOF 导入公共信息模型管理器(CIM Object Manager, CIMOM)并植入类实例要求的任意静态数据;通过 CIMOM 的 API 使用受管对象。其中,虚拟系统和物理系统均由 CIM_ComputerSystem 类表示,继承自 CIM_ManagedElement 核心类,两者通过 CIM_HostedDependency 关联,对物理系统和虚拟系统的操作管理都以 CIM_ComputerSystem 类对象为基础直接或间接实现。此外,通过对某个已存在的虚拟系统 CIM_ComputerSystem 实例调用 RequestStateChange()方法,可以使虚拟系统处于初始、活动、暂停、休眠等状态。

2.2.2 Provider 设计

系统中的提供者主要指方法和实例提供者。前者需要实现 `invokeMethod` 接口，然后传入方法名、CIM 对象路径等参数以调用在 MOF 中声明并在提供者中具体实现的私有方法，例如启动、暂停虚拟系统等；后者定义了一些抽象方法，如枚举、创建实例等操作，可以直接调用。对于虚拟系统动态信息，可以通过 `Virt_ComputerSystem Provider` 获取，此外，Libvirt 开源项目的 `Libvirt-CIM Provider` 也实现了部分功能，同样可以用来获取相关信息。部分 Provider 代码如下：

```
Provider( "cmpi::Virt_ComputerSystem" )
//Provider
```

```

Class Virt_ComputerSystem: CIM_System {
    [Description( “唯一标识一个虚拟系统” )]
    string NameFormat;
    [Description( “系统允许状态” )]
    uint16 EnabledState
    [Description( “系统状态转换情况” ),
    ValueMap{ “1”、“2”、“3”、“4” },
    Values{ “Enabled”、“Disabled”、“Paused”、“Suspended” }]
    uint16 RequestedState;
    [ Description( “改变状态方法” )]
    uint32 RequestStateChange(
        [IN]uint16 RequestedState,
        [IN]datetime TimeoutPeriod);
    [Description ( “设置电源状态方法” )]
    uint32 SetPowerState(
        [IN]uint32 PowerState,
        [IN]datetime Time);    ...};

```

2.3 自主管理特征设计

若要实现自主管理调度的特征，首先则要有自我监测机制，能够实时自我监测系统的运行情况，比如 CPU、内存占用率等。这里可以通过指示器模块实现，通过事件订阅(Subscribe)将信息传送给 CIMLister，这样就能实时地监测系统情况。例如对虚拟系统的 CPU 的监测过程为：系统管理端向某台虚拟系统的 CPU 资源订阅一个阈值，由 CIM_IndicationFilter 记录；订阅管理器调用 CIMOM 对 CPU 资源进行订阅，并维护订阅列表；如果超过阈值，则 CIM_IndicationHandler 会将事件发送至 CIMLister，发出警告。

当得到系统运行情况时,需要进行智能判断处理相关问题。此时需要定义一些策略用来指导和确定如何管理、控制和分配资源。系统策略库存放于 Self-Manager 层,根据输入输出结果。其中,输入主要包括接收各虚拟系统的性能度量(CPU、内存)以及负载情况等;输出包括向虚拟系统管理平台发出的管理动作和事件通知,通知包括虚拟系统被开启或是休眠关闭,虚拟系统上的负荷情况等。

当传来输入时,策略中的动作将被执行。例如当一个虚拟系统的负载达到 60% 时,将会动态扩展,新增一个虚拟系统,达到系统间动态平衡;反之,若负载长时间低于 10%,则将该系统休眠。策略执行过程如图 2 所示。

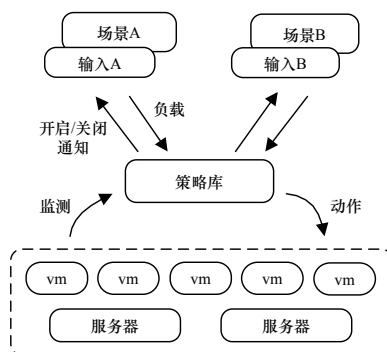


图2 策略执行过程

3 虚拟系统管理模型实现

模型底层采用 Libvirt-CIM Provider 作为虚拟系统 Provider 提供者, Libvirt-CIM Provider 是 RedHat 主导的开源项目。目前, 部分实现虚拟资源 CIM Provider 功能, 它通过 Libvirt API 实现对虚拟机 Hypervisor 的操作, Xend 进程用于控制物理机上所有虚拟系统相关的管理功能, 管理员可通过

Xm 管理工具获得虚拟机的管理操作。

对于模型上层方面,采用 CIM-XML 通过 XML over Http^[6] 与 CIMOM 进行 CIM 信息交换。目前,主流的 CIMOM 产品有 SFCB、OpenPegasus、OpenWbem、WMI 等,它们支持不同的 Provider 接口标准,例如 CMPI 和 CIMPLe。本文采用 OpenPegasus(支持 CMPI)作为 CIMOM 容器。图 3 展示了在 Linux 和 xen 虚拟化技术平台下,采用标准化通信协议的虚拟系统管理模型实现。其中,xend 服务是 xen 中用来管理所有虚拟机的一个服务。

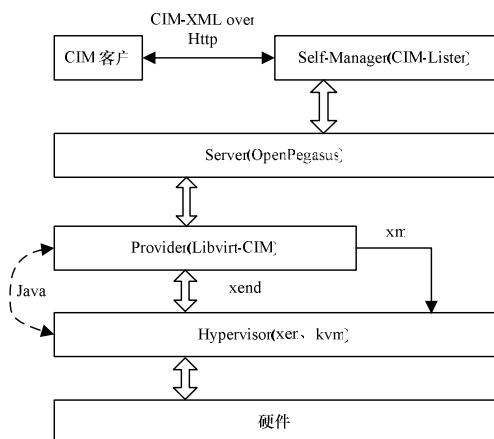


图3 虚拟系统管理模型的实现

基于以上条件实现测试环境,实现虚拟系统 CIM 建模,在 CentOS 和 xen、kvm 组成的虚拟平台下对该模型进行测试。

4 系统测试与结果

在系统原型测试中管理的 Web 请求服务是圆周率计算服务,可以供访问者计算不同位数的圆周率值。系统由 1 台 Linux 虚拟服务器(LVS)和 2 台 Web 访问服务器组成。Web 访问服务器上各有一些虚拟系统,分别基于 xen、kvm 虚拟平台,操作系统为 CentOS 5.3。拓扑结构如图 4 所示。

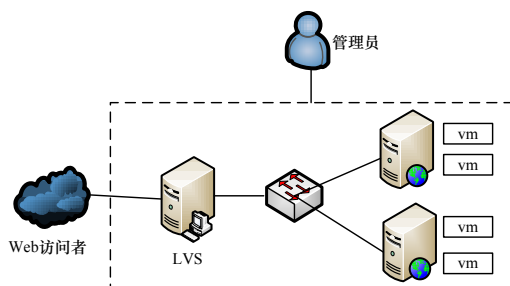


图4 系统测试的拓扑结构

利用 APACHE 自带的压力测试工具 ab(apache benchmark)模拟用户大量访问。给虚拟系统订阅的 CPU 占用率为 60%,用 ab 模拟 1 000 个用户同时发出 1 000 个请求,可以发现订阅的 CPU 信息发出警报,此时将会动态扩展,新增一个虚拟系统。此时虚拟系统间的运行情况截图如图 5 所示。



图5 虚拟系统间的运行情况截图

当一段时间后,访问量降低,新开启的虚拟系统 CPU 占用率持续下降,当低于 10%时,订阅信息会警示将休眠虚拟系统 vm2,此时 vm2 的 CPU 利用率将变为 0,但仍享有内存。

该管理模型不仅能够统一管理异构平台下的虚拟系统,同时也大大提高了访问质量。在 ab 模拟大量访问测试中,对于上面的测试,如果是对单一的虚拟系统进行访问,请求时间为 6 667.312 ms,而使用该系统的请求时间为 3 049.088 ms,具体速率比较如图 6 所示。

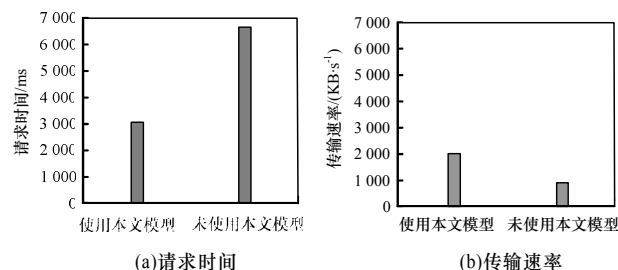


图6 使用和未使用本文模型的性能比较

根据以上测试对虚拟系统进行 CIM 建模,并通过管理模型对虚拟对象进行实例操作,实现了对虚拟系统的统一管理,且模型本身具有良好的伸缩性与扩展性。

5 结束语

随着虚拟化技术的发展,多样化的计算环境已经被普遍应用,包括虚拟和群集系统等。传统管理手段已经达不到虚拟系统的管理要求。另外,对于不同厂商的产品,如何统一地进行部署和管理,也变得较重要。WBEM 提供了一个标准化的建模环境,对不同资源可以进行统一建模管理。

在此基础上,本文提出一种虚拟系统自主管理调度的模型,对异构虚拟平台下的虚拟系统能够进行统一管理,并通过策略方式进行实时监控、动态扩展等功能。该模型已应用于实验室日常项目中,验证了其可行性和先进性。但测试时采取了局域网环境,对于大型集群的资源管理有所欠缺,有待进一步研究优化。

参考文献

- [1] 肖政,韦卫,侯紫峰. 基于 WBEM 的统一管理系统[J]. 计算机工程, 2005, 31(1): 47-49.
- [2] 王春娟,董丽丽,贾丽. Web 集群系统的负载均衡算法[J]. 计算机工程, 2010, 36(2): 102-104.
- [3] Foster S K. Overhead Matters: A Model for Virtual Resource Management[C]//Proc. of the 1st International Workshop on Virtualization Technology in Distributed Computing. Tampa, USA: [s. n.] 2006: 6-7.
- [4] 何涛. 高可用性虚拟化管理操作环境的研究与开发[D]. 西安: 西北工业大学, 2007.
- [5] Distributed Management Task Force, Inc.. CIM System Virtualization Model White Paper v1.0.0[EB/OL]. (2007-11-11). http://www.dmtf.org/sites/default/files/standards/.../DSP2013_1.0.0.pdf.
- [6] Distributed Management Task Force, Inc.. CIM Operations over HTTP v1.3.1[EB/OL]. (2009-07-29). http://www.dmtf.org/standards/published_documents.

编辑 陆燕菲