[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/455 (2006.01)
G06F 9/50 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200910060752.6

[43] 公开日 2009 年 7 月 22 日

[11] 公开号 CN 101488098A

[22] 申请日 2009.2.13
[21] 申请号 200910060752.6
[71] 申请人 华中科技大学
地址 430074 湖北省武汉市洪山区珞喻路 1037 号
[72] 发明人 金 海 邵志远 李 勇 陈华才 张 德 陆晓雯 杨鹏飞 黄 健 袁旻昊

[74] 专利代理机构 华中科技大学专利中心
代理人 曹葆青

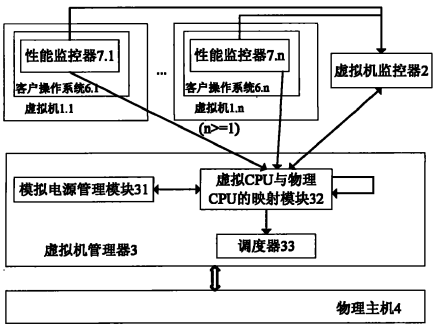
权利要求书 3 页 说明书 14 页 附图 2 页

[54] 发明名称

基于虚拟计算技术的多核计算资源管理系统

[57] 摘要

本发明基于虚拟计算技术的多核系统计算资源管理系统，包括多个虚拟机、虚拟机监控器和虚拟机管理器。 虚拟机监控器实时监控虚拟机的负载情况、运行状态；虚拟机管理器是虚拟机和物理主机通信的纽带；虚拟机运行在虚拟机管理器之上，为用户提供一个虚拟平台。 同时本发明将虚拟机分为 3 大类，对每一类的虚拟机分别采用不同的资源调整策略。 本发明为多核系统计算资源的动态调整和分配问题提供了一条切实可行的途径，实现了节能及资源利用的最大化。



1、一种基于虚拟计算技术的多核计算资源管理系统，其特征在于：该系统从上至下包括虚拟机监控器(2)、虚拟机管理器(3)、物理主机(4)和若干个虚拟机(1)，在各虚拟机(1)的客户操作系统(6)中分别设置有性能监控器(7)；

虚拟机(1)为用户提供虚拟平台，客户操作系统(6)为支持热插拔功能的操作系统，性能监控器(7)用来监测应用程序的负载变化情况，分析虚拟机(1)的负载情况，并将分析结果传递给虚拟机管理器(3)；

虚拟机监控器(2)实时监控虚拟机(1)的负载情况和运行状态，将结果及时显示反馈给用户；用户能够通过虚拟机监控器(2)手动调整虚拟机(1) 的计算资源，并将调整后的映射关系通知虚拟机管理器(3)；

虚拟机管理器(3)用于对虚拟机(1)进行管理，提供虚拟机(1)与物理主机(4)之间的访问通信。

2、根据权利要求 1 所述的多核计算资源管理系统，其特征在于：

对于运行关键应用的虚拟机，当预先知道需要的 CPU 资源量时，则性能监控器(7)直接给虚拟机管理器(3)发送其资源需求量，否则，性能监控器(7)采用乘法增加线性减少的策略得到资源需求量，并将其发送虚拟机管理器(3)；虚拟机管理器(3)按照上述资源需求量将资源分配给运行关键应用的虚拟机；

对于运行服务应用的虚拟机，虚拟机管理器(3)根据指定的服务级别协议，分配给运行服务应用的虚拟机能够满足服务要求的最少的资源量；

虚拟机管理器(3)将满足上述二种应用之外的空闲资源分配给运行后台应用的虚拟机。

3、根据权利要求1或2所述的多核计算资源管理系统，其特征在于：虚拟机管理器(3)包括模拟电源管理模块(31)、虚拟CPU与物理CPU的映射模块(32)和调度器(33)；

模拟电源管理模块(31)用于自动感知虚拟机是否进入休眠，并将这些信息及时发送给虚拟 CPU 与物理 CPU 的映射模块(32)；

虚拟 CPU 与物理 CPU 的映射模块(32)用于维护虚拟 CPU 与物理 CPU 的对应关系表，通过收集到的信息自动改变虚拟计算单元与物理计算单元的映射关系；它还将调整后的映射关系通知给实际执行映射的调度器(33)与虚拟机监控器(2)；

调度器(33)用于实现虚拟机(1)的计算资源与物理计算资源的映射，调度器(33)读取虚拟 CPU 与物理 CPU 的映射模块(32)的对应关系表，根据对应关系表进行计算资源的调整。

4、根据权利要求 2 所述的多核计算资源管理系统，其特征在于： 设定运行关键应用的虚拟机的合理 CPU 使用率范围为 N1～N2，关键应用的资源请求的阈值为 w；乘法增加线性减少的策略包括下述步骤：

①、计算采样时间内的 CPU 利用率和并行度，其中，并行度根据当前竞争计算资源的处于运行态的进程数和 CPU 的个数计算；

②、如果资源利用率大于 N2 并且并行度大于 1 连续出现 w 次，则转步骤③, 如果资源利用率小于 N1 并且并行度小于 1 连续出现 w 次，转步骤④, 否则，转入步骤①；

③、产生资源需求增加信号，并发送给虚拟机管理器(3)，转步骤①；

④、产生资源需求减少信号，并发送给虚拟机管理器(3)，转步骤①。

5、根据权利要求 2 或 4 所述的多核计算资源管理系统，其特征在于：对于基于虚拟机的 Web 服务器，虚拟机管理器(3)给运行服务应用的虚拟机

进行分配的资源量$U_{allocate}$的计算公式为：

$$U_{allocate} = \lambda_{need} D_{cpu}^{vm} \times 120\%$$

其中，λ_{need}为指定的虚拟机服务器服务请求的到达率，D_{cpu}^{vm}为在与虚拟机相同硬件配置的虚拟机中，CPU 处理每一个 HTTP 请求服务所需要的时间。

基于虚拟计算技术的多核计算资源管理系统

技术领域

本发明属于计算机虚拟化技术领域，具体涉及一种基于虚拟计算技术的多核计算资源管理系统，该系统采用虚拟计算技术对多核计算资源进行有效的管理。

背景技术

多核技术的出现，为计算机系统结构领域，以及并行计算领域带来了翻天覆地的变化。一方面，计算机不再像以前那样只是追求更“快”，主频不断升高，而是横向发展，拥有了更“多”的计算资源；另一方面，应用软件必须提高自身的并行度，从而达到提高自身速度的目标，而不是像以前那样只能等着 CPU 变得更快。

这一系列的变化为计算机系统结构研究领域提出了新的挑战：如何管理好更“多”的计算资源？在传统的计算模型（包括新兴的集群计算）中，由于单一节点的计算资源相对较少，一般采用操作系统来管理，多任务间通过分时复用的方法，共享系统内的计算资源。然而，多核以及未来的众核计算技术的发展，带来了越来越多和越来越复杂（如异构多核）的计算资源，在这种背景下，采用单一操作系统对系统进行管理，将带来应用之间一些冲突问题，并可能导致计算资源的浪费和不合理分配。

虚拟计算技术的复兴，为这一问题的解决，提供了一把钥匙。该技术能够在一台主机上运行多个虚拟机（每个虚拟机运行一种操作系统），容纳更多的应用，从而提高系统的资源利用率。由于虚拟计算技术具备上述特点，目前该技术已经被很多企业数据中心采用，并被证明能够极大地提高数据

中心中服务器资源的利用率、减少服务器的数量，从而达到降低建设成本和节约能源的目的。虚拟计算的这一特点，能够在技术上为更好地管理多核系统提供支持。然而，作为一门复兴不久的技术，虚拟计算技术在多核资源管理方面所做的工作并不多，还远未成熟。

本发明基于 Xen 虚拟机管理器，提出一种基于虚拟计算技术的多核计算资源管理系统，能够根据虚拟机的需求，动态地对 CPU 资源进行优化配置，达到节约能源及资源有效利用的目的。其中，Xen 是一款优秀的开源虚拟机管理器，现有的 Xen 系统提供的监视和管理工具如 Xenmon、Virt-Manager 等无法对每个虚拟机内在的需求进行感知，更无法实现对资源的动态分配。

发明内容

本发明的目的在于提供一种基于虚拟计算技术的多核计算资源管理系统，该系统可以获取虚拟机的资源需求，实现了多核计算资源的动态调整和分配问题。

本发明提供的基于虚拟计算技术的多核系统计算资源管理系统，其特征在于：该系统从上至下包括虚拟机监控器、虚拟机管理器、物理主机和若干个虚拟机，在各虚拟机的客户操作系统中分别设置有性能监控器；

虚拟机为用户提供虚拟平台，客户操作系统为支持热插拔功能的操作系统，性能监控器用来监测应用程序的负载变化情况，分析虚拟机的负载情况，并将分析结果传递给虚拟机管理器；

虚拟机监控器实时监控虚拟机的负载情况和运行状态，将结果及时显示反馈给用户；用户能够通过虚拟机监控器手动调整虚拟机 的计算资源，并将调整后的映射关系通知虚拟机管理器；

虚拟机管理器用于对虚拟机进行管理，提供虚拟机与物理主机之间的

访问通信。

上述性能监控器和虚拟机管理器可以采用下述优选方式实现：

对于运行关键应用的虚拟机，当预先知道需要的 CPU 资源量时，则性能监控器直接给虚拟机管理器发送其资源需求量，否则，性能监控器采用乘法增加线性减少的策略得到资源需求量，并将其发送虚拟机管理器；虚拟机管理器按照上述资源需求量将资源分配给运行关键应用的虚拟机；

对于运行服务应用的虚拟机，虚拟机管理器根据指定的服务级别协议，分配给运行服务应用的虚拟机能够满足服务要求的最少的资源量；

虚拟机管理器将满足上述二种应用之外的空闲资源分配给运行后台应用的虚拟机。

本发明旨在采用虚拟计算技术，通过对上层应用行为的判断和对底层物理计算资源的动态调整，解决多核计算资源的动态调整和分配问题。本发明通过对虚拟机中应用行为的判断，动态调整该虚拟机中虚拟处理器（VCPU）与真实的物理 CPU 中核的对应关系。通过对虚拟机所拥有的计算资源的调整，提高了虚拟机应用的性能，优化计算机系统整体配置，并降低系统能耗的目标。本发明具有以下技术效果：

1．本发明实现了计算资源的优化管理，提供了自定义配置接口。通过对客户操作系统的实时性能监控，分析出虚拟机的资源需求，动态地对物理资源进行合理分配。

2．本发明根据应用对 CPU 资源的需求度，将虚拟机应用分为 3 大类：关键应用、服务应用、后台应用，对不同的虚拟机应用采用不同的 CPU 资源调整策略。进一步增强资源管理能力，动态资源调整的粒度更细，资源调整策略更具针对性。提高了系统整体的 CPU 利用率和服务质量，更能最大化地提高系统资源的利用率和减少能源的损耗。

3. 本发明提供管理员高效实时的虚拟机性能监控，提供的虚拟机性能监控器能够实时监控多个虚拟机的运行状态，具体还可以监控虚拟机VCPU的资源利用情况、VCPU与CPU映射关系等。

4. 最后，本发明以单机多核为平台，提供对 ACPI 电源管理的支持。可以根据虚拟机平台的资源消耗状态，动态关闭或者打开物理 CPU 资源。

附图说明

图 1 为本发明基于虚拟计算技术的多核系统计算资源管理系统的体系结构；

图 2 为缺省情况下计算资源映射关系图；

图 3 为图 2 中虚拟机进入休眠后的计算资源映射关系图；

图 4 为图 3 中虚拟机繁忙后的计算资源映射关系图；

图 5 为多类虚拟机运行在同一虚拟平台上的计算资源映射关系图。

具体实施方式

下面通过借助以下实施例将更加详细说明本发明，且以下实施例仅是说明性的，本发明并不受这些实施例的限制。

本发明所采用的硬件具有 Intel VT 或者 AMD Pacifica 技术，采用 XEN3.2 版本搭建虚拟平台，各类虚拟机运行在该虚拟平台，通过对不同类别的虚拟机采用不同的资源调整策略，实现最大化地利用资源。

如图 1 所示，从体系结构的层次上来看，本发明所应用的计算机系统从上至下包括虚拟机 1.1，1.2，……，1.n、虚拟机监控器 2、虚拟机管理器 3 和物理主机 4，本发明可以容纳多个虚拟机，n 为正整数。各虚拟机 1.1，1.2，……，1.n 的客户操作系统 6.1，6.2，……，6.n 中分别设置有性能监控器 7.1，7.2，……，7.n。为表述方便，下文中将虚拟机 1.1，1.2，……，1.n、客户操作系统 6.1，6.2，……，6.n 和性能监控器 7.1，7.2，……，7.n 分别统称为虚拟机 1、客户操作系统 6 和性能监控器 7。

虚拟机 1 为用户提供虚拟平台，在其客户操作系统 6 内包含有性能监控器 7。客户操作系统 6 为支持热插拔功能的操作系统，能够增加或者减少其 CPU 个数。

性能监控器 7 运行在客户操作系统 6 中，用来监测应用程序的负载变化情况，如 CPU 使用率、内存使用率等。判断是否需要更多（或者更少）的计算资源，分析虚拟机 1 中的 CPU 使用状态，CPU 是否满负荷或空闲，从而得出是否需要增加或者减少分配给虚拟机 1 的 CPU 资源，并通过相应的算法预测出 CPU 资源需求量。此需求量为保证虚拟机 1 正常服务运行的情况下，最小的 CPU 资源消耗量，并将判断结果传递给虚拟机管理器 3 中的虚拟 CPU 与物理 CPU 的映射模块 32。

正如上述，为了更好地更细化地实现资源分配，本发明将虚拟机应用分为 3 大类：关键应用、服务应用、后台应用。

对于关键应用虚拟机，如果可以预先知道需要多少 CPU 资源，那么就直接给虚拟 CPU 与物理 CPU 的映射模块 32 发送其资源需求量；如果不能预先知道需要多少 CPU 资源，采用乘法增加线性减少的策略，即成倍增加 CPU 资源，依次减少 CPU 资源。具体预测算法如下所述：

设定运行关键应用的虚拟机的合理 CPU 使用率范围 N1～N2，关键应用的资源请求的阈值为 w。通过性能监控器 7 实时监控虚拟机的 CPU 利用率，如果该类虚拟机的资源占用率很高且有多个“饥渴”进程在竞争同一个 CPU，这就需要产生增加 CPU 资源的信号。以 N1=50%，N2=70%，w=5 为例，CPU 资源的增加和减少信号的产生是按照如下过程产生：

1、计算预先设计好的采样时间内的 CPU 利用率和并行度，并行度的计算是根据当前竞争计算资源的处于运行态的进程数和 CPU 的个数得出的；

2、如果资源利用率大于 N2 并且并行度大于 1 连续出现 w 次，则转步骤 5；

3、如果资源利用率小于N1并且并行度小于1连续出现w次，转步骤6;

4、如果资源利用率介于N1 和N2之间，转步骤1；

5、产生资源需求增加信号，并发送给虚拟机管理器3，转步骤1；

6、产生资源需求减少信号，并发送给虚拟机管理器3，转步骤1。

由于动态调整资源分配会消耗一定的资源，如果资源调整过于频繁，将会影响系统的服务性能。因此本实例为关键应用的资源请求设定了一个阈值为5，当资源请求次数等于5，则产生资源调整信号。

服务应用虚拟机主要给用户提供服务应用，包括：网络应用、数据库应用等等。对于这类的虚拟机，需要保证服务质量，响应时间是评价其服务质量的主要指标之一。针对服务应用虚拟机的资源调整,其主要思想是在指定的服务级别协议（SLA）下，分配给虚拟机最少的资源来满足服务要求。

Web服务是一个I/O密集型任务，本发明中以基于虚拟机的Web服务器为例来研究服务虚拟机的CPU资源需求。下面首先介绍计算中常用的参数：

T：表示整个测试过程的时间。

K：表示系统中资源的数目。

B_i：在整个观察过程T中，资源i的忙碌时间。

A_i：在观察过程T中，资源i的请求数量。

C_i：观察过程T中，资源i的完成的数量。C表示整个系统完成的请求数。

通过以上几个已知量，可以得到几个更有用的参数：

λ_i：到达率λ_i表示单位时间内工作请求的数量。

$$\lambda_i = \frac{A_i}{T} \qquad (1.1)$$

U_i：资源i的利用率；

$$U_i = \frac{B_i}{T} \tag{1.2}$$

X_i：资源 i 的吞吐率，表示单位时间完成的请求数量；

$$X_i = \frac{C_i}{T} \tag{1.3}$$

D_i 表示处理一种给定类型的请求消耗资源 i 的时间。由定义得，

$$D_i = \frac{B_i}{C} = \frac{U_i \times T}{C} = \frac{U_i}{C \div T} = \frac{U_i}{X} \tag{1.4}$$

在进出平衡的系统中，

$$X = \lambda \tag{1.5}$$

因此，

$$D_i = \frac{U_i}{\lambda} \tag{1.6}$$

由于每一种资源利用率都不会超过 100%，由上式得，

$$U_i = \lambda \times D_i \le 1 \tag{1.7}$$

因此，

$$\lambda \le \frac{1}{D_i} \tag{1.8}$$

所以有

$$\lambda \le \frac{1}{\max_{i=1}^{K} D_i} \tag{1.9}$$

响应时间是一个重要的性能衡量指标。下面两个公式给出了计算响应时间的方法。

$$R = \sum_{i=1}^{K} R_i \tag{1.10}$$

$$R_i = \frac{D_i}{1 - U_i} \tag{1.11}$$

本实例定义一个减速因子 S_v，表示在虚拟机和本地操作系统执行一个

相同操作时，CPU 资源忙碌时间的比例：

$$S_v = \frac{B_{CPU}^{HVM}}{B_{CPU}^{Linux}} \tag{1.12}$$

在虚拟机和本地系统测试时间段都为 T，由式（1.2）和式（1.12）得，

$$S_v = \frac{B_{CPU}^{HVM}}{B_{CPU}^{Linux}} = \frac{U_{CPU}^{HVM} \times T}{U_{CPU}^{Linux} \times T} = \frac{U_{CPU}^{HVM}}{U_{CPU}^{Linux}} \tag{1.13}$$

如果在虚拟机和本地操作系统中用相同的到达率进行测试，由式(1.13）和式（1.6）可以推得，

$$S_v = \frac{U_{CPU}^{HVM}}{U_{CPU}^{Linux}} = \frac{D_{CPU}^{HVM} \times \lambda}{D_{CPU}^{Linux} \times \lambda} = \frac{D_{CPU}^{HVM}}{D_{CPU}^{Linux}} \tag{1.14}$$

在虚拟机中处理一个和本地操作系统相同的事务，不仅虚拟机需要消耗资源，同时 Dom0 也要消耗资源，其中 Dom0 在 Xen 中表示一个特权的虚拟机，所有虚拟机的 I/O 操作都需要通过 Dom0 来处理。为了计算 Dom0 的性能开销，定义一个因数 $Cost_{HVM}^{Dom0}$：

$$Cost_{HVM}^{Dom0} = \frac{B_{CPU}^{Dom0}}{B_{CPU}^{HVM}} \tag{1.15}$$

在测试中 Dom0 和虚拟机具有相同的测试时间段 T，则由式（1.2）和式（1.15）可以推得，

$$Cost_{HVM}^{Dom0} = \frac{B_{CPU}^{Dom0}}{B_{CPU}^{HVM}} = \frac{U_{CPU}^{Dom0} \times T}{U_{CPU}^{HVM} \times T} = \frac{U_{CPU}^{Dom0}}{U_{CPU}^{HVM}} \tag{1.16}$$

由式（1.16）和式（1.6）可以推得，

$$Cost_{HVM}^{Dom0} = \frac{U_{CPU}^{Dom0}}{U_{CPU}^{HVM}} = \frac{D_{CPU}^{Dom0} \times \lambda}{D_{CPU}^{HVM} \times \lambda} = \frac{D_{CPU}^{Dom0}}{D_{CPU}^{HVM}} \tag{1.17}$$

为了计算基于服务应用虚拟机的最大吞吐率需要以下几个参数：

D_{CPU}^{Linux}：在与虚拟机相同硬件配置的本地 Linux 操作系统中，处理每一个 HTTP 请求的 CPU 服务需要。该参数通过在本地 Linux 系统进行测试可以计算出来。

S_v：Linux 系统到硬件虚拟机的减速因子。计算该参数不仅需要在 Linux

系统中进行测试，还需要在配置相同的硬件虚拟机进行相同的测试，通过式（1.13）计算出该参数。

$Cost_{HVM}^{Dom0}$：根据硬件虚拟机下的测试数据，利用式（1.16）可以计算出该参数。

有了以上几个参数，使用式（1.14）和式（1.17）可以得出 D_{CPU}^{Dom0} 和 D_{CPU}^{HVM}。虚拟机在 I/O 密集型负载下最大能够使用的 CPU 资源由参数 *Ceil* 给出，因此可以得到：

$$\lambda_{\max} \approx \min(\frac{1}{D_{cpu}^{Dom0}}, \frac{Ceil}{D_{cpu}^{vm}}) \tag{2.18}$$

通过对虚拟机的 CPU 资源进行限制可以改变 Web 服务器的最大吞吐率。当为虚拟机分配 CPU 资源数量为 U（$U < Ceil$）时，则有

$$\lambda_{\max}^{U} \approx \min(\frac{1}{D_{CPU}^{Dom0}}, \frac{U}{D_{CPU}^{HVM}}) \tag{2.19}$$

在指定的虚拟机服务器服务请求的到达率 λ_{need} 下，由公式（2.2）可以计算出硬件虚拟机所需的 CPU 资源数量：

$$U_{need} = D_{CPU}^{HVM} \times \lambda_{need} \tag{2.20}$$

式（2.18）给出了在指定分配 CPU 资源下最大吞吐量的计算方法，式（2.20）给出指定到达率计算所需要的 CPU 资源数量的方法。例如指定服务器需要的到达率为 λ_{need}，由（2.20）可以计算出所需要的 CPU 资源数量。但是在服务器实际运行时 CPU 利用率并不是恒定不变的，它在平均利用率上下浮动。因此在调整过程中，分配给硬件虚拟机的资源需要略大于硬件虚拟机所需要的 CPU 资源数量。如果只为硬件虚拟机分配计算出的 CPU 资源数，就会出现在运行时，有时 CPU 资源不足的情况。这会导致服务器的性能不稳定，从而不能提供正常的服务。同时由公式（1.11），可以看出 CPU 利用率在接近上限时，响应时间会急剧上升，尤其是超过 80%以后，响应时间会成倍增长。因此在分配资源时，为硬件虚拟机多分配 20%的资源，这样既可以保证服务器的稳定，同时还可以保证响应时间在理想的范

围内。计算方法如下：

$$U_{allocate} = \lambda_{need} D_{cpu}^{vm} \times 120\% \qquad (2.21)$$

其中 D_{cpu}^{vm} 为在与虚拟机相同硬件配置的虚拟机中，CPU 处理每一个 HTTP 请求服务所需要的时间。

对于后台应用虚拟机，本发明不为其进行资源需求预测分析，因为后台应用虚拟机上运行的主要是批处理作业、后台任务等，对于这类的虚拟机，其运行的任务和作业什么时候完成并不是很重要，因此本发明并不为其进行 CPU 资源需求预测，只要有空闲的资源，就分配给后台应用虚拟机使用。

虚拟机监控器 2 和虚拟机 1 一样运行在虚拟机管理器 3 上。虚拟机监控器 2 不仅实时监控虚拟机的性能状态，而且提供了图形化显示和用户控制接口。虚拟机监控器 2 实时监控虚拟机 1 的负载情况和运行状态，将结果及时地图形化显示反馈给用户。用户可以通过虚拟机监控器 2 手动调整虚拟机 1 的计算资源，并将调整后的映射关系通知虚拟 CPU 与物理 CPU 的映射模块 32。

虚拟机1很多情况下是通过虚拟机管理器3来与物理主机4进行通信,由虚拟机管理器3来决定其对物理主机4上所有资源的访问。虚拟机管理器3包括模拟电源管理模块31、虚拟CPU与物理CPU的映射模块32和调度器33。

模拟电源管理模块 31 可以自动感知虚拟机是否进入休眠，并将这些信息及时发送给虚拟 CPU 与物理 CPU 的映射模块 32。如果模拟电源管理模块 31 得知虚拟机 1 进入休眠的信息，则将该休眠信息发送给虚拟 CPU 与物理 CPU 的映射模块 32，由后者对虚拟机 1 的计算资源进行调整，达到节约能源及减少计算机损耗的目的。虚拟 CPU 与物理 CPU 的映射模块 32 根据接收到的休眠信息对其维护对应关系表进行调整，删除需要关闭的 CPU、VCPU 相应的对应关系。

虚拟 CPU 与物理 CPU 的映射模块 32 维护虚拟 CPU 与物理 CPU 的对应关系表，虚拟 CPU 与物理 CPU 的映射模块 32 通过收集到的信息可以自动改变虚拟计算单元与物理计算单元的映射关系，正如上所述，本发明将虚拟机应用根据 CPU 需求度划分为 3 类，不同类别的应用采用不同的资源调整策略，并将调整后的映射关系通知给实际执行映射的调度器 33 与虚拟机监控器 2。

关键应用一般对时间比较敏感，通过完成时间来衡量其性能，因此需要大量的计算资源。这类虚拟机应用的资源调整策略比较简单，为了实现性能与真实机器上执行时间的接近，本发明为该类虚拟机应用尽可能地提供其所要求的 CPU 资源，最大限度的满足关键应用的 CPU 资源需求。该类虚拟应用采用资源强占及优先级策略，即优先给该类虚拟机分配 CPU 资源，并将虚拟机 CPU 资源与物理 CPU 资源进行一一对应。如果剩余的 CPU 资源不能满足该类应用的要求，实行强占剥夺式策略，强占已经分配给其他虚拟机的 CPU 资源，从而最大限度的保证该类虚拟机应用所请求的 CPU 资源。

由于服务应用虚拟机主要的资源需求 I/O 资源，不像关键应用那样需要消耗大量的计算资源，本发明对服务应用虚拟机采用了资源共享的调整策略。为了更好地利用 CPU 资源，本发明对服务虚拟机采用细粒度的计算资源调整方式。本发明将单个 CPU 资源分为十份，通过划分 CPU 时间来实现。通过在一段时间内，限制服务应用虚拟机消耗 CPU 使用时间实现，保证了服务应用虚拟机在保证 CPU 资源的前提下，限制其对 CPU 资源的消耗。例如通过资源预测算法得出，在指定到达率 λ_{need} 下，虚拟机所需的 CPU 资源数量为 2.4 个。其中的小数部分表示其可以使用分配的 CPU 资源的份额，即在一段时间 t 秒内，该虚拟机最多消耗 2.4t 秒的 CPU 时间。通过细粒度的资源调整可以更好更优地实现资源利用最大化。

本发明对后台应用虚拟机采用了共享的资源调整策略，其分配资源的

优先级最低，在优先保证前面两类虚拟机正常提供服务的基础上，只要系统中有多余的 CPU 资源，就分配给该类虚拟机使用。

调度器 33 实现虚拟机 1 的计算资源与物理计算资源的映射。调度器 33 读取虚拟 CPU 与物理 CPU 的映射模块 32 的对应关系表，根据对应关系表进行计算资源的调整。调度器将会根据对应关系表，将虚拟 CPU 调度到相对应的物理 CPU 的运行队列，等待物理 CPU 的调度执行。

如图 2 所示，在虚拟管理器 3 上启动了两个虚拟机，此处只为举例说明，本系统可以同时启动多个虚拟机，该图是采用虚拟计算技术时，两个虚拟机中虚拟处理单元和物理处理单元的缺省对应关系。其中，物理主机具有 4 个处理单元，虚拟机 1.1 具有两个虚拟处理单元，全部被激活，并各自对应了一个物理处理单元；虚拟机 1.2 具有 4 个虚拟处理单元，但由于本身负载不重（可以由虚拟机 1.2 的性能监控器得到），所以只激活了其中 2 个并分别对应了一个物理处理单元。

为了实现这一缺省情况下的计算资源映射，本发明通过在虚拟机管理器 3 中实现图 2 的虚拟计算单元到物理计算单元的映射关系（由模块虚拟 CPU 与物理 CPU 的映射模块 32 实现），以及关闭虚拟机 1.2 中的部分虚拟处理单元（由支持热插拔功能的客户操作系统 2 实现）来达到 CPU 资源优化配置的目的。

如图 3 所示，在系统运行一段时间后，虚拟机 1.1 由于很长时间没有任务，而进入了休眠状态，出于节省能耗的考虑，需要释放它以前占用的两个物理处理单元。这时，由于虚拟机 1.2 的负载没有变化，而虚拟机 1.1 所释放的物理计算资源无事可做，因而被虚拟机管理器 3 关闭（或降低主频），从而节省电力资源。

为实现图 3 中的映射关系，本发明需要实现以下的功能：

1. 实时监视虚拟机的状态，通过模拟电源管理模块 31 得到虚拟机 1.1 进入休眠的信息；

2. 模拟电源管理模块31将虚拟机1.1休眠的信息传递给虚拟CPU与物理CPU的映射模块32，由后者动态调整虚拟CPU与物理CPU的对应关系；

3. 调度器33读取模块虚拟CPU与物理CPU的映射模块32的调整结果，由其关闭物理主机中相应的处理单元。

如图4所示。在图3的基础上，如果虚拟机1.2中的负载增加，需要更多的计算资源，本发明将该执行环境中的休眠的虚拟处理单元激活，并通知虚拟机管理器3中的虚拟CPU与物理CPU的映射模块32，让后者将物理主机上关闭的CPU或核打开，并将激活的虚拟计算单元定向（或调度）至新打开的CPU或核上。

为了实现这一功能，本发明实现流程如下：

1. 由模块虚拟机监控器2实时监控虚拟机1.2中的负载情况，以确定该环境需要更多的计算资源；

2. 虚拟CPU与物理CPU的映射模块32动态实现映射关系自动调整；

3. 虚拟机1.2的性能监控器激活该环境中的虚拟计算单元。

4. 由调度器打开主机上的物理计算单元，并将虚拟计算单元映射到新打开的物理计算单元上。

为了更好地更细化地实现资源分配，本发明将虚拟机应用分为3大类：关键应用、服务应用、后台应用。对于不同的虚拟机应用，采用不同的资源调整策略，实现资源的灵活调整。

如图5所示，物理主机上共有4个CPU，虚拟机管理器3上运行了3个虚拟机，在虚拟机1.1上运行着关键应用，虚拟机1.2上运行服务应用，虚拟机1.3上运行了后台应用，多类虚拟机共同运行在同一个虚拟平台上。本发明规定关键应用具有最高的资源分配优先级，其CPU分配策略为一一对应强占式。如图5所示，为了保证虚拟机1.1的性能，需要分配2个CPU的计算资源。根据资源调整策略，给虚拟机1.1分配两个虚拟CPU并分别

与物理 CPU 一一对应，此处只是举例说明，如果随着虚拟机 1.1 的任务加重需要更多的 CPU 资源，可以强行抢占虚拟机 1.2 和虚拟机 1.3 的资源来保证其性能，如果任务减轻，需要释放 CPU 资源，所释放的 CPU 资源可以分配给虚拟机 1.2 和 1.3 使用。

其余两类应用的优先级较低，服务应用虚拟机的优先级要高于后台应用虚拟机，这两者在资源分配策略上都采用资源共享策略，即两者共享使用 CPU 资源，优先考虑虚拟机 1.2 的资源请求，在最大化满足虚拟机 1.2 的请求下，才将多余的 CPU 资源分配给虚拟机 1.3 使用。

本发明不仅局限于上述具体实施方式，本领域一般技术人员根据本发明公开的内容，可以采用其它多种具体实施方式实施本发明，因此，凡是采用本发明的设计结构和思路，做一些简单的变化或更改的设计，都落入本发明保护的范围。

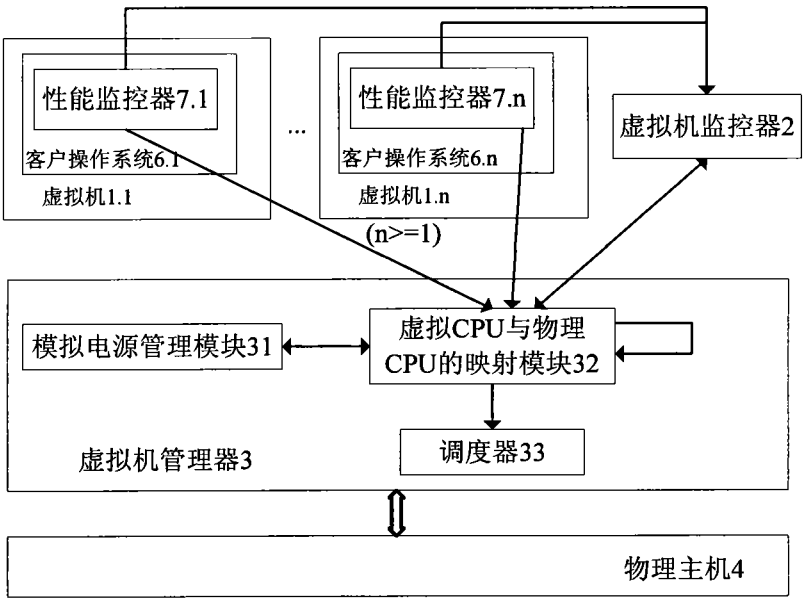

性能监控器7.1
客户操作系统6.1
虚拟机1.1
...
性能监控器7.n
客户操作系统6.n
虚拟机1.n
(n>=1)
虚拟机监控器2
模拟电源管理模块31
虚拟CPU与物理CPU的映射模块32
调度器33
虚拟机管理器3
物理主机4

图 1

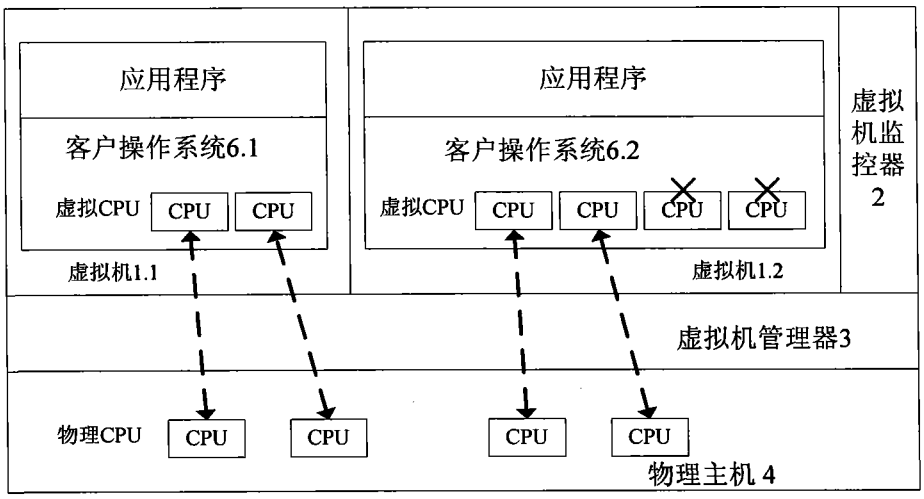

应用程序
客户操作系统6.1
虚拟CPU
CPU
CPU
虚拟机1.1
应用程序
客户操作系统6.2
虚拟CPU
CPU
CPU
CPU
CPU
虚拟机1.2
虚拟机监控器2
虚拟机管理器3
物理CPU
CPU
CPU
CPU
CPU
物理主机 4

图 2

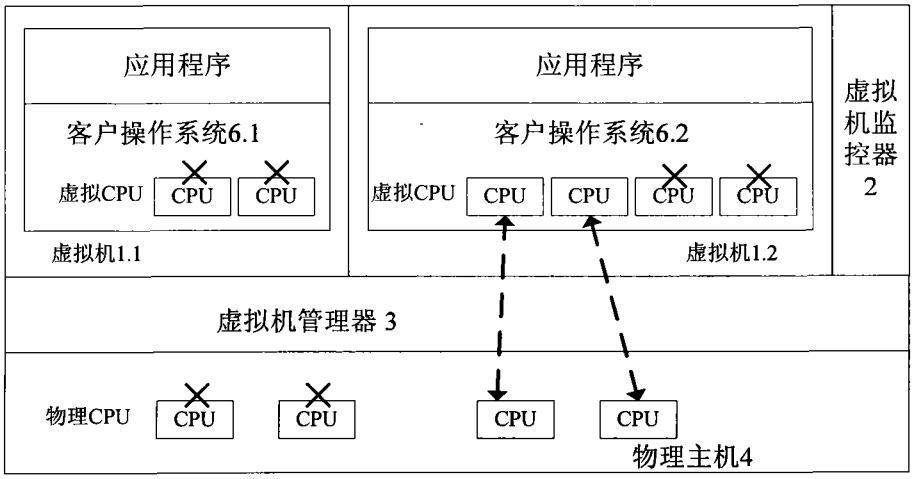

应用程序
客户操作系统6.1
虚拟CPU
CPU
CPU
虚拟机1.1
应用程序
客户操作系统6.2
虚拟CPU
CPU
CPU
CPU
CPU
虚拟机1.2
虚拟机监控器2
虚拟机管理器 3
物理CPU
CPU
CPU
CPU
CPU
物理主机4

图 3

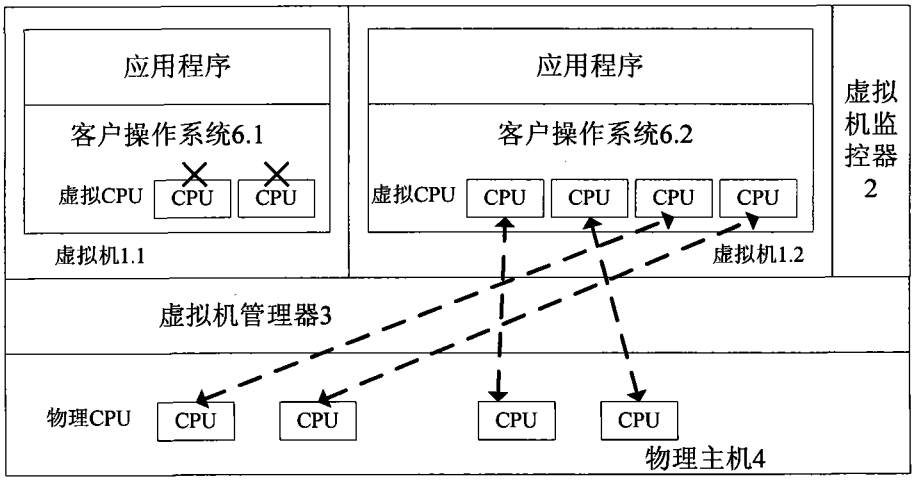

应用程序
客户操作系统6.1
虚拟CPU
CPU
CPU
虚拟机1.1
应用程序
客户操作系统6.2
虚拟CPU
CPU
CPU
CPU
CPU
虚拟机1.2
虚拟机监控器2
虚拟机管理器3
物理CPU
CPU
CPU
CPU
CPU
物理主机4

图 4

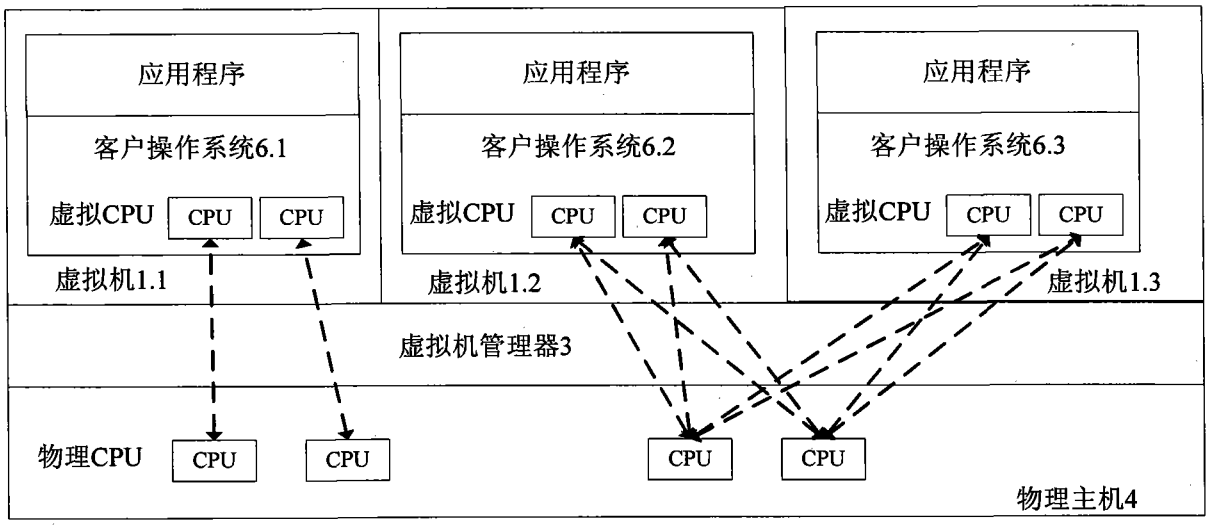

应用程序
客户操作系统6.1
虚拟CPU
CPU
CPU
虚拟机1.1
应用程序
客户操作系统6.2
虚拟CPU
CPU
CPU
虚拟机1.2
应用程序
客户操作系统6.3
虚拟CPU
CPU
CPU
虚拟机1.3
虚拟机管理器3
物理CPU
CPU
CPU
CPU
CPU
物理主机4

图 5