

基于负载分时分析的虚拟服务整合建模分析

敬思远 余 堃

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】介绍了一种面向下一代绿色数据中心的虚拟服务整合方案。基于服务负载高峰处于不同时段的原则,提出了基于负载分时分析的建模方法,在保证QoS的前提下能够达到资源利用率最大化。考虑了服务之间的联系性和互斥性,以及服务与服务器之间的兼容性,提出了5项整合原则,使该方案具有更强的实际应用价值。将建立的模型看作是有约束的多维装箱问题,提出了基于分组遗传算法(GGA)的智能优化算法搜索全局最优解。通过实验表明,该方案与之前的模型相比具有更高的整合率。

关键词 分组遗传算法; 多维装箱问题; 虚拟服务整合; 负载分析

中图分类号 TP302.7

文献标识码 A

doi:10.3969/j.issn.1001-0548.2012.05.025

Modeling Analysis of Virtual Service Consolidation Based on Workload Timesharing Analyzing

JING Si-yuan and SHE Kun

(School of Computer Science & Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract A solution of virtual service consolidation oriented to next generation green data center is introduced in this paper. Due to the peak-valley of the workload of services is in different time, a way to problem modeling is proposed based on workload timesharing analysis, which can achieve maximal resource utilization on the premise of keeping the required QoS. Meanwhile, the associated services and the mutual services, and the compatibility between services and servers are taken into account and five principles of consolidation are considered. By regarding the modeling as a multi-dimension bin packing problem with constraints, we propose a GGA-based heuristic to search for the global optimal solution. The experimental results show that the solution can get a better consolidation ratio than the former model.

Key words grouping generic algorithm; multi-dimension bin packing problem; virtual service consolidation; workload analysis

1 简介

IT服务提供商在提供新服务时,由于难以准确预估其实际负载,往往在预估服务负载峰值的基础上采用简单的冗余资源预留策略以保证服务质量(QoS),并且以“一机一服务”的方式进行部署。实际上,这些服务中大部分并不经常处于访问高峰,恰恰相反,它们绝大部分时间里的访问量都很小。咨询公司Gartner在其报告中指出,大部分IT数据中心中的服务器资源利用率低于20%^[1]。多数服务器资源利用率长期处于较低水平,但是它们却占据着与其利用率不符的空间以及消耗相当的电力,这种现象被IT业内称为“服务器蔓延”。随着IT数据中心规模的不断扩大,该现象不仅代表资源持续浪费,还

会由于服务器和其他基础设施(如制冷、照明设备、机架和网络设备等)的持续增加而使管理难度加大,运营成本(用于管理、维护以及电力支付等)指数式增长。随着云计算等新一代计算范式的发展,IT数据中心的规模飞速的膨胀,“服务器蔓延”已经成为制约IT数据中心发展的重要问题。

服务器虚拟化并对服务进行整合是一种解决上述问题的有效途径。IBM公司于20世纪60年代首先发明虚拟机技术,允许一台主机20运行多个操作系统。它通过一个中间层程序VMM(virtual machine monitor,也称为hypervisor)向上层操作系统、应用程序提供底层硬件的抽象,同时对虚拟机中运行的客户操作系统进行有效的协调、分配和管理。目前主流的虚拟机软件包括VMWare、Xen系列以及

收稿日期: 2010-10-29; 修回日期: 2011-06-20

基金项目: 国家863项目(2008AA04A107)

作者简介: 敬思远(1981-),男,博士生,主要从事云计算、虚拟化、绿色计算等方面的研究。

Microsoft Virtual Server等。由于数据中心中有的服务系统软件需要特定的硬件平台、特殊的操作系统版本支持,而虚拟机技术能够在一台服务器上同时运行多个异构操作系统的技术特点正好能满足这个需要,因此将以前运行在多个异构系统平台上的服务系统整合到一台服务器上变得可行。在Intel VT-x及其他类似技术出现后,这种可行性正在不断加大。

服务质量(quality of service)和能力规划(capacity planning)一直是IT领域的研究热点。在文件及数据库系统、网络、实时系统等领域常常会对其进行研究。虚拟服务整合问题可以看作是对IT数据中心的能力规划^[2-3],其目标是在保证QoS不受影响的情况下找到一种整合解决方案,使整合后采用的服务器数量最小。文献[4]提出了一种基于组合优化的建模方法来解决这个问题。文献[5-6]认为该方法没有考虑到服务之间不相容的问题,提出了一种有冲突的非连通图模型以及一种两阶段启发式算法和一种遗传算法来寻找最优解。但是通过分析发现,该模型仍然存在不足,因为它并没考虑到不同服务其访问高峰处于不同时段,而采用历史记录中资源利用率的峰值进行简单迭加。两个不同的服务LDAP和OA于某天的资源利用率记录如图1所示。由图1可知,LDAP服务在早晨上班时间段左右处于访问高峰期,但是在其他时间的访问量并不大;而OA服务在办公时间的访问量较高,但是在上班前以及下班后则访问量不高。如果采用之前提到的方法,会需要两台服务器来分别运行这两个服务。

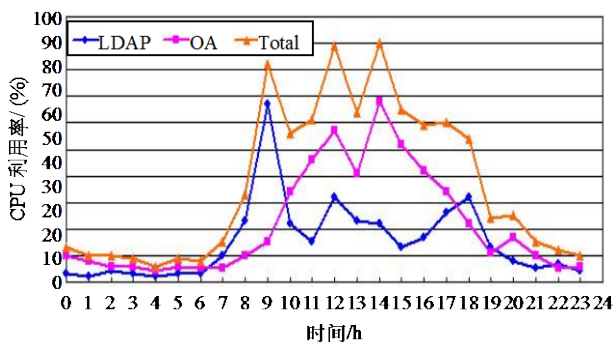


图1 不同服务负载比较

因此,本文提出了一种基于负载分时分析的建模方法。通过将服务的负载按照时间段进行划分,在每个时间段内,只要整合后的服务负载总和不超过给定风险阈值,就认为是一个有效的整合方案。同时,还考虑了服务之间的联系性和互斥性,服务与服务器之间的兼容性,提出了5项整合原则,并将其作为约束条件加入进模型,使其具有更强的实际应用价值。并提出了基于分组遗传算法(GGA)的全

局最优搜索算法。最后,本文对该算法搜索的最优解的质量、计算性能以及解决问题规模的能力进行了实验。

2 虚拟服务整合建模

2.1 问题分析与整合原则

虚拟服务整合问题即是找到一种整合方案,可以将虚拟服务分配到目标服务器上,并且满足所有的约束条件;如果使用的服务器数为最小,则该方案是最优解决方案。

在整合过程中,有一些必要因素需要考虑,如某些具有频繁的数据通信的服务(Web应用和数据库),将它们放在一起可以避免数据在网络上的传输时间,减少服务响应时间,进而提高服务QoS。本文充分分析了此类问题,并将其总结为服务之间具有的联系性和互斥性问题,以及服务与服务器之间的技术兼容性问题,并提出了具体整合过程中的5项整合原则:

1) QoS保证原则。

整合后所有服务的QoS不受影响。QoS是与为该服务提供的服务器资源相关的(如CPU、Ram、I/O等)^[7-8],因此该原则可以理解整合后每台服务器上的所有服务,在每个时段的负载(资源需求)总和均不超过该服务器的资源总量乘以风险阈值。

2) 服务增强原则。

假如某2个或多个服务之间具有较强的交互关系(如数据访问,服务调用等),那么将这些服务放在同一台服务器上能够避免数据在网络上传输的时间,减小服务响应时延(response time),可以增强服务QoS以及可靠性。

3) 服务冲突避免原则。

假如某2个或多个服务之间具有相同的特殊资源占用要求,或者其安全策略产生冲突,那么需要将这类服务分开部署,以避免产生冲突。

4) 预先分配原则。

假如某些服务有特殊的技术要求,或者某些服务器具有的某些技术特征可以为这些服务提供有益的帮助,那么需要将这类服务预先分配到这类服务器上。

5) 禁止分配原则。

假如某些服务器具有的某些技术特征与服务产生冲突,使其不能正常运行或者产生其他一些负面的影响,那么必须禁止将这类服务分配到这类服务器上。

2.2 数学建模

假设给定需要整合的服务集合 $S = \{s_1, s_2, \dots, s_n\}$ 和用于服务分配部署的服务器集合 $P = \{p_1, p_2, \dots, p_m\}$ 。其中任意服务 s_i 在 t 时间段对资源 k 的需求为 u_{ik} , 假设所有服务器 p_j 的资源 k 的总量是 r_{jk} , 资源的集合表示为 K , $k \in K$ 。根据以上假设, 构造基础的数学模型为:

$$1) \text{ minimize } \sum_{j=1}^m y_j .$$

Subject to:

$$2) y_j, x_{ij} \in \{0, 1\}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m .$$

$$3) \text{ if } s_i \rightarrow p_j, \text{ then } x_{ij} = 1, \quad \forall i, \forall j .$$

$$4) \text{ if } \sum_{i=1}^n x_{ij} \neq 0, \text{ then } y_j = 1, \text{ else } y_j = 0, \forall j .$$

$$5) \sum_{j=1}^m x_{ij} = 1, \forall i .$$

$$6) \sum_{i=1}^n u_{ik} x_{ij} \leq \alpha r_{jk}, \quad \forall j, \forall t, \forall k .$$

约束条件1)为目标函数, 代表用于整合的服务器的数量最小化; 约束条件2)表示 y_j, x_{ij} 的取值范围; 约束条件3)表示如果有服务 s_i 被分配给服务器 p_j , 则 x_{ij} 取值为1; 约束条件4)表示如果服务器 p_j 上的服务数不为0, 则 y_j 取值为1, 否则取值为0; 约束条件5)表示一个服务只能被分配到一台服务器上; 约束条件6)代表整合原则1), 即部署在服务器 p_j 上的服务在任意时段 t , 对任意资源 k 的需求总和应小于服务器 p_j 的资源 k 的总量, 其中 α 是风险阈值系数, 该系数可以用于控制风险。

更进一步, 假设满足整合原则2)的服务集合 $E_l \in E$, $l \in \{1, 2, \dots, |E|\}$, 其中 E 表示满足原则2)的所有服务集合的集合, 并且需满足 $E_l \cap E_{l_2} = \emptyset$, 则有:

$$\sum_{i \in C} x_{ij} = |E_l| \quad C = \{i | s_i \in E_l, \forall i\}, \forall j, \forall l \quad (1)$$

式(1)表示对于任意服务器 p_j , 其上属于 E_l 的服务 s_i 的个数等于 E_l 中服务的个数。对于 E_l 中服务的资源需求总和大于服务器资源总量的情况, 可以通过分析将 E_l 进一步细分, 以满足约束条件6)。

假设符合原则3)的服务集合 $T_l \in T$, $l \in \{1, 2, \dots, |T|\}$, 其中 T 表示符合原则3)的所有服务集合的集合, 则有:

$$\sum_{i \in C} x_{ij} \leq 1 \quad C = \{i | s_i \in T_l, \forall i\}, \forall j, \forall l \quad (2)$$

式(2)表示对于任意服务器 p_j , 其上属于 T_l 的服务 s_i 的个数小于1, 保证了最多只有一个属于 T_l 的服务被分配到一台服务器上。

假设服务器具有的技术特征表示为 $A(p_j)$, 服务 s_i 有技术要求 $A(s_i^+)$ 和技术冲突 $A(s_i^-)$, $A(s_i^+) \cap A(s_i^-) = \emptyset$, 那么整合原则4)、5)可以转化为下面的约束条件:

$$\sum_{j \in J_i^+} x_{ij} = 1 \quad J_i^+ = \{j | A^+(s_i) \subseteq A(p_j), \forall j\}, \forall i \quad (3)$$

$$\sum_{j \in J_i^-} x_{ij} = 0 \quad J_i^- = \{j | A^-(s_i) \cap A(p_j) \neq \emptyset, \forall j\}, \forall i \quad (4)$$

约束条件3)确保了服务 s_i 一定被分配在满足其技术要求的服务器上; 约束条件4)确保了服务 s_i 一定不会被分配在与其有技术冲突的服务器上。

3 基于分组遗传算法的最优化方法

本文提出的分时模型可以看作是一种有约束的多维装箱问题(multi-dimension bin packing problem)的扩展, 并且在该问题中bin的方向是确定的, 即不可旋转。该问题已经被证明是一个NP-hard问题^[9-10]。解决这类问题的常用方法有分支界限法^[11](branch and bound, B&B)和一些智能算法^[12]。

遗传算法(GA)是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法。GA本身的鲁棒性、并行性以及NP完全问题求解中的广泛应用表明它应该非常适合解决上述问题。但是实际上, 在求解此类分组问题上, 经典的Holland GA的表现并不尽如人意^[13]。文献[14]分析了问题的原因, 并设计了一种混合的分组遗传算法(GGA)来求解分组问题, 取得了令人满意的效果。本文采用该方法来求解问题模型。

3.1 编码方法

正确地对染色体进行编码来表示问题的解是GA的基础工作, 也是最重要的工作。经典Holland GA提出的0-1编码、实数编码和整数编码等3种编码方式中, 可以采用整数编码表示本文模型, 即:

$$X = (ABACB)$$

染色体中每一位的位置代表一个服务, 每一位的值代表该服务被分配到的服务器编号。上式代表服务1和3被分配到服务器A, 服务2和5被分配到服务器B, 服务4被分配到服务器C。但是, 该表示方法难以体现服务器的数量和进行交叉遗传。

文献[14]提出了一种新的编码方式。编码分为两部分, 分别表示服务和服务器。表示服务的部分仍

采用上述方式，并且长度固定；表示服务器的部分长度不固定，为当前采用的服务器，即：

$$X = (ABACB : ABC)$$

3.2 遗传算子

1) 交叉算子。

本文采用注入方式的交叉遗传。注入仅限于编码中的可变长度部分，即服务器编码。注入过程首先对于两个选定的染色体 X_1 和 X_2 ，随机选取两个切点，并将 X_2 中的DNA片段注入 X_1 ，即：

$$\begin{aligned} X_1 &= (\dots AB|ED|C) \\ X_2 &= (\dots A'|C'D'|E'B') \\ X_1' &= (\dots|ABC'D'EDC) \\ \Rightarrow X_2' &= (\dots|A'EDC'D'E'B') \end{aligned}$$

在 X_1' 中，如果原有的分组中某些服务器上的服务和新插入的 $C'D'$ 上的服务有重叠，则将其从染色体中删去。如 AC 重叠，则 $X_1' = (\dots|BC'D'ED)$ ，释放的服务再通过FFD(first fit descending)算法将这些服务进行重新分配。

染色体中的不变长度部分，即服务编码则根据交叉遗传的结果进行变换。

2) 选择算子。

本文采取择优选择的原则。根据染色体适宜度函数的值对父代染色体和后代染色体进行统一排序，取前面一半作为下一次的遗传计算的输入。

3) 变异算子。

本文选择如下的变异操作，以较小的概率 p ，随机选择少量的已分配服务的服务器进行释放。由于释放后这些服务器上的服务需要被再次分配。本文采用FFD算法将这些服务进行重新分配。

3.3 适宜度函数

遗传算法对一个解的好坏用适宜度函数的大小来评价。本文从资源利用率的角度进行评价，有：

$$f(x) = \frac{\sum_{j=1}^m y_j \left(\frac{1}{|K|} \sum_{k=1}^{|K|} \left(\max_{i=1}^{c_i} \left(\sum_{i=1}^{n_j} u_{ijk} x_{ij} \right) / r_k \right) \right)}{\sum_{j=1}^m y_j} \quad (5)$$

3.4 种群初始化方法

本文首先对待分配服务进行随机选择，然后通过FFD算法进行分配。如果分配过程中出现违反约束的情况，则通过一种简单的策略将其与之前已分配的服务进行交换和修复，直到所有服务都被分配完毕。

3.5 停止策略

算法停止条件是当采用的服务器数等于预先计

算的下界^[15](lower bound, LB)或者达到约定的遗传代数。LB为：

$$LB = \left\lceil \max_{t \in T, k \in K} \left(\sum_{i=1}^n u_{tik} / r_k \right) \right\rceil \quad (6)$$

4 仿真及性能评价

4.1 实验设计

本文采用文献[4-6]的方法，通过生成伪随机数的方式获取实验数据，但是具体方法又有不同。由于对服务的访问是服从泊松过程的，并且其形态受到参数 λ 的控制，因此本文采用了以下方法生成伪随机数：

1) 通过0~1均匀分布随机生成参数 λ 。

2) 根据生成的随机参数 λ ，采用泊松过程生成服务24 h的负载数据，数据以小时为单位，处于0~1之间；为了简便，本文仅生成了1维资源数据。

3) 随机给服务器加入约束条件。

4) 生成服务器参数。服务器容量为1，风险阈值 α 取值为0.9。服务器技术特征随机生成。

4.1.1 验过程及目的

本文的实验主要是针对整合质量以及算法性能进行分析。整合质量通过整合率 τ 进行判断，整合率越高，代表平均资源利用率越高，采用的服务器数量越小。算法性能主要通过两方面进行测试：

1) 测试问题规模对性能的影响，即在时段粒度和问题维度确定的情况下，通过改变服务数量规模来对算法进行测试；

2) 测试参数 t 对性能的影响，即在问题维度和问题规模确定的情况下，通过改变时段划分的粒度进行测试，有：

$$\tau = \frac{n}{\sum_{j=1}^m y_j} \quad (7)$$

式中， n 是参加整合的服务数； $\sum_{j=1}^m y_j$ 是用于整合的服务器数量； τ 为整合率，代表整合后每台服务器平均部署的服务数。

4.1.2 遗传算法参数

种群规模为200，最大代数为120，交叉概率为1.0，变异概率为0.01。

4.1.3 实验环境

本文采用了Matlab 2009a作为实验工具。硬件平台为AMD Phenom II×4 945 (3 000 MHz)和2 GB DDR内存。

4.2 整合结果比较

将模型的约束条件分为第一类约束和第二类约束两类: 第一类约束条件包括式(1)~(4), 第二类约束条件包括式(5)。图2和图3分别显示了分时模型与文献[6]中的模型, 在无第一类约束条件和有第一类约束条件下, 采用GGA算法计算得到的整合率比较。其中在第二项实验中, 随机添加了大约20%的第一类约束(数据是由约束数除以服务数获得)。问题规模从100开始, 以100为单位递增, 问题最大规模为600。

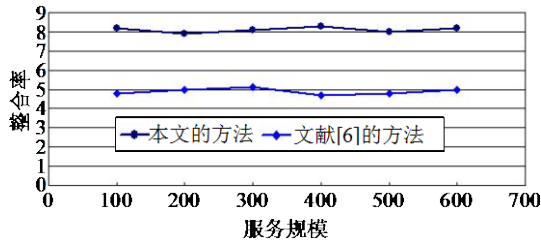


图2 无第一类约束条件下的整合结果比较

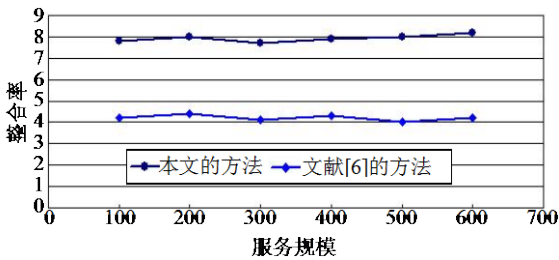


图3 有第一类约束条件下的整合结果比较

由图可以看出, 在无第一类约束条件下, 分时模型的整合率大约在8:1, 而文献[6]中的模型仅有约5:1, 分时模型在整合率上提高了3左右, 即同一服务器上部署的服务数量增加了3个。在有第一类约束条件下, 分时模型的整合率变化不大, 而文献[6]中的模型下降了大约0.5。该试验说明分时模型在整合率上明显优于文献[6]中的模型, 并且针对有第一类约束条件的情况, 具有较好的稳定性。

4.3 性能分析

本文针对问题规模对计算性能的影响进行分析。时间划分粒度为1 h, 问题维度(即资源种类)为1。图4显示了在无第一类约束条件和有第一类约束条件下, 问题规模分别为100~800时, 在120次遗传迭代内算法的计算出的结果与最终最优结果之间的比。可以看出, 当问题规模超过400时, 计算结果与最优结果的比开始快速下降, 当问题规模达到800时, 该比值仅仅大约达到70%和40%。事实上, 当问题规模达到800时, 搜索达到理想结果的时间超过3 h。

本文还针对时间划分粒度对计算性能的影响进行分析。问题规模维持在400, 问题维度为1。实验

中时间划分粒度分别24、6、3和1。不同的时间划分粒度需要对数据进行预处理, 本文采用划分时间段内负载的峰值作为该时间段的负载数据。当时间划分粒度为24时, 在无第一类约束条件下, 分时模型退化为文献[6]中的模型。图5~8显示了实验结果, 图中纵坐标Q代表搜索过程中目标函数与算法下界的比值, 横坐标代表遗传代数。由于LB与时间划分粒度相关, 所以图中的LB值并不相等。随着时间划分粒度越细, 算法的收敛速度会变得较慢。该结果很好地解释了图4中的问题规模对计算性能的影响。

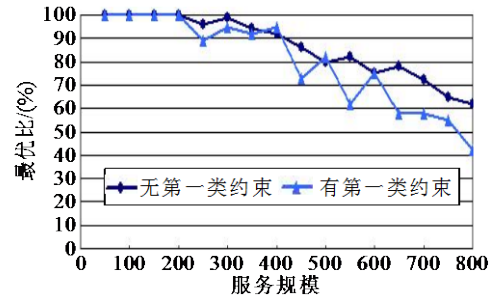


图4 120次遗传迭代内算法的计算结果

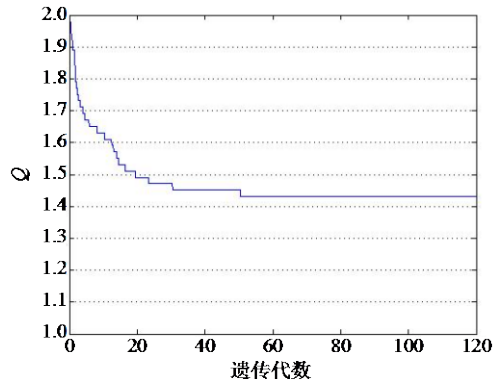


图5 时间划分粒度为24 h

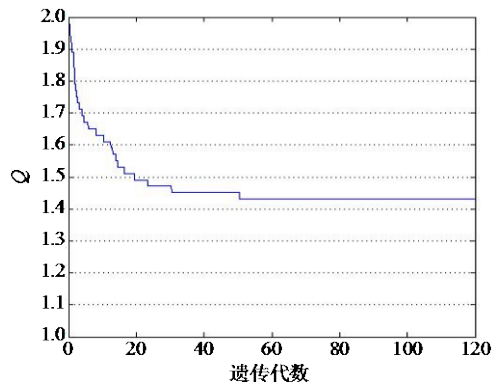


图6 时间划分粒度为6 h

分时模型及本文的GGA算法在时间划分粒度大于1 h, 问题规模小于400时, 能获得较好的时间性能。事实上, 时间划分粒度为1已较细, 并且能在可接受的时间内, 解决400个服务的整合问题和很多数据中心的整合问题; 如果服务数量大于400, 可以通

过分组的方法进行处理。

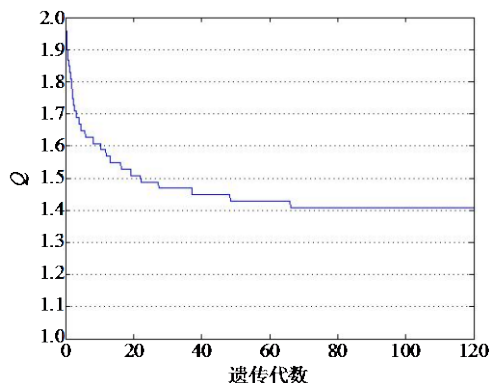


图7 时间划分粒度为3 h

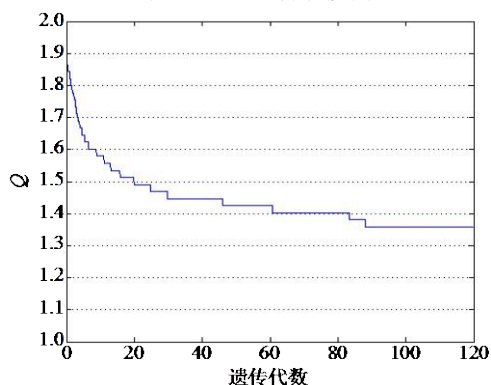


图8 时间划分粒度为1 h

5 结 论

本文介绍了一种新的虚拟服务整合方案。通过实验证明了本文的模型与现有的模型比较具有更高的整合率，平均高出3个服务/服务器，并且在有第一类约束条件时表现得比现有模型更好，这对IT服务提供商进一步提高资源利用率和降低运营成本具有很好的指导作用。此外，GGA算法在本文的模型上的计算性能也得到了进一步的分析。实验表明，该算法在时间划分粒度大于1 h、问题规模小于400时，能获得较好的计算性能。

下一步将研究虚拟数据中心服务的动态调度问题，使其在性能不受影响的情况下达到节能的目的。

参 考 文 献

[1] PARENT K. Server consolidation improves IT's capacity utilization[R]. Court Square Data Group, 2005.

- [2] SPELLMANN A, ERICKSON K, REYNOLDS J. Server consolidation using performance modeling[J]. IT Professional, 2003, 5(5): 31-36.
- [3] BICHLER M. Capacity planning for virtualized servers[C]//16th Workshop on Information Technologies and System. Milwaukee, USA: [s.n.], 2006.
- [4] AIJIRO Y, TANAKA A. A combinational optimization algorithm for server consolidation[C]//The 21st Annual Conference of Japanese Society for Artificial Intelligence. Kyoto, Japan: [s.n.], 2007.
- [5] GUPTA R, BOSE S K, SUNDARRAJAN S, et al. A two stage heuristic algorithm for solving the server consolidation problem with item-item and bin-item incompatibility constraints[C]//Proceedings of IEEE International Conference on Service Computing. Hawaii, USA: IEEE, 2008.
- [6] AGRAWAL S, BOSE S K, SUNDARRAJAN S. Grouping genetic algorithm for solving the server consolidation problem with conflicts[C]//Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation. Shanghai, China: ACM, 2009.
- [7] BENEVENUTO F, FERNANDES C. Performance models for virtualized applications[C]//ISPA 2006 Workshops. [S.l.]: [s.n.], 2006.
- [8] GTSI White Paper. Reducing data center's power and energy consumption: saving money and go green[EB/OL]. [2009-10-21]. <http://www.gtsi.com/cms>.
- [9] CHEKURI C, KHANNA S. On multi-dimensional packing problems[C]//Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms. Baltimore, Maryland, USA: ACM, 1999.
- [10] GAREY M R, JOHNSON D S. Computer and intractability: a guide to the theory of NP-completeness[M]. New York, USA: W H Freeman and Company, 1979.
- [11] 陈宝林. 最优化理论与算法[M]. 北京: 清华大学出版社, 2005.
CHEN Bao-lin. Optimization theory and algorithm[M]. Beijing: Tsinghua University Press, 2005.
- [12] 汪定伟. 智能优化方法[M]. 北京: 高等教育出版社, 2007.
WANG Ding-wei. Intelligent optimization methods[M]. Beijing: Academic Press, 2007.
- [13] FALKEKENUER E. Generic algorithm and grouping problems[M]. New Jersey, UEA: John Wiley & Sons Ltd, 1998.
- [14] FALKEKENUER E. A hybrid group generic algorithm for bin packing[J]. Journal of Heuristic, 2004, 2: 5-30.
- [15] GENDREAU M, LAPORTE G, SEMET F. Heuristic and lower bounds for the bin-packing problem with conflicts[J]. Computer and Operations Research, 2004, 31: 347-358.

编辑 黄 莘