高技术通讯 2012 年 第22 卷 第7 期:683 ~688

doi:10.3772/j.issn.1002-0470.2012.07.003

服务质量约束下的虚拟集群资源分配策略研究①

刘菲菲②*** 董小社* 田红波*

(*西安交通大学计算机科学与技术系 西安 710049)

(**武警工程大学通信工程系 西安 710086)

摘 要 为了有效实现基于虚拟机技术的高性能计算环境资源调度与分配,以满足用户不同服务质量请求,提出了一种动态可调节的资源分配策略。该策略以用户任务被放缓程度为核心指标,并允许用户根据该指标提出相应的服务质量约束,从资源提供者的角度保证了用户获得相关服务的开销最低,同时总的资源收益最大。对该分配模式下的静态资源分配问题进行了形式化的描述,并结合该策略提出混合背包算法。实验结果表明:面对大规模应用时,该算法较其它一般性算法更易得到效能较优的资源分配方案。

关键词 资源分配,虚拟集群,服务质量(QoS)

0 引 言

在物理资源有限的情况下,借助于虚拟化技术为用户搭建专属的虚拟高性能计算环境,是解决用户计算需求多样性的有效方法之一[1-3]。但由于在这种情况下分配给用户的资源相对固定或用户独占部分资源,容易造成资源闲置,而且也使得系统整体的资源利用率难于保证。因为当用户作业繁重程度不同或用户数目发生变化时,系统无法有效地在用户间对资源分配进行重新部署。目前,通过一些虚拟机监控技术[4-6],实时监测各个虚拟环境中的资源使用情况,可以部分解决资源闲置问题,但如何协调用户间对资源的争用并提高系统的资源利用率,仍是资源分配中要解决的关键问题。

伴随着网格计算、云计算的不断推广,将高性能计算能力作为一种服务提供给用户时,资源分配中会出现另一个核心问题,即如何满足特定用户的服务质量(QoS)[7]需求。一般而言,服务质量指标指时间、开销、可靠性、可信性、安全性等多个方面的指标[8],其中较为重要的通常是时间指标与开销指标。随着效能驱动(utility-drive)等概念的引入,在特定服务质量约束条件下如何达到系统效能最优,已成为协调多用户间资源分配的核心问题[9,10]。如何在虚拟集群中实现基于特定的服务质量约束的资源分配,目前已有一些相关工作。但由于资源分配形式上的变化,使得这些基于传统服务质量约束的资源分配策略,都存在明显的局限性,文献[11]提出的先依据开销最优的策略选择资源,再由这些资源动态构建虚拟集群的方法就是一个例子。这一方法虽然比较容易满足用户在开销上的约束,但并不能保证整个系统的资源效能。还有文献[12]提出的对用户分级并对高级别用户提供资源预留[13],以提高服务的可靠性的方法,但这种做法通常也会为用户带来更大的开销。针对以上问题,本文给出了一个用于评价作业被放缓程度[14]的新的服务质量指标,提出了基于该指标的新的资源分配策略。该策略允许系统在用户资源需求发生变化时对资源分配方案进行动态调节,以最大限度地缓解不同时间段内资源利用忙闲不均的情况。围绕该指标形成的资源价格机制,也能在有效降低用户开销的同时实现系统资源整体收益的最大化。

1 基于约定区间的虚拟资源分配策略描述

一般而言,用户长期使用虚拟集群时,其所需的平均计算能力可以通过一段时间的观测得到,为其分配固定数量的虚拟机实例可以相对比较容易满足

① 863 计划(2011AA01A204,2008AA01A022)和国家自然科学基金(61173039)资助项目。

② 男,1978 年生,博士生;研究方向:计算机体系结构,高性能计算;联系人,E-mail: liuff.wj@stu.xjtu.edu.cn

(收稿日期:2011-06-07)

其计算需求。从用户的角度,占有的资源越多,任务的周转时间就越短,但也会带来更大的费用开销,而当其所占用的资源达到一定值后,额外的资源带来的性能提升将非常有限。反之,即使占用的资源较少,但只要能在可以接受的时限内完成任务,其花销可能会更少,且系统的多用户整体性能效率更高。

1.1 问题定义

从上文的描述中不难看出,提供给用户的计算资源应满足一个约束范围,不妨做如下定义。

定义 1 用户资源需求约定:在虚拟计算环境下,用户应根据自身任务情况提出资源需求,并确定一个可适当浮动的范围。与之相应,系统在接受该用户请求后,也应保证所提供的计算能力保持在该区间范围内。

这一资源范围可表达为 $[(1-k_i)\alpha_i,\alpha_i]$。其中,$\alpha_i$ 为每个用户所期望的最优计算能力;k_i 为每个用户所能容忍的计算能力最大下降比。显然,k_i 是用户与系统间就服务质量达成的约定,其取值应在[0,1)区间内。特定条件下该值也允许为 1,即表示用户允许系统在资源缺少的情况下将自身的任务请求暂时挂起。

定义 2 服务于特定用户的资源数限制:分布式条件下,服务于用户的通常是多种异构资源的集合。根据定义 1,此时提供给用户的资源应服从如下约束:

$$\forall i \quad \sum_{j=1}^{H}\alpha_{ij}\leqslant\alpha_i,\ \sum_{j=1}^{H}\frac{\alpha_{ij}}{\alpha_i}\geqslant(1-k_i)$$

其中,α_{ij} 为每个独立的主机资源 j 分配给用户 i 的计算资源,H 为提供计算资源的物理集群(或其它大型计算设备)的数量。

定义 3 用户任务实际被放缓(slowdown)程度:对于每个用户 i,$\sum_j(\alpha_{ij}/\alpha_i)$ 反映了其实际占用资源与理想状态的相对差距,也可以直观地认为是用户作业被放缓的程度。

需要说明的是,当系统根据对用户所占用的计算资源进行调整时,该值也随之变化。但不同于 k_i,这个比值与用户开销无关,该值越高,用户的满意度也越高。

1.2 基于 QoS 约束的资源分配与计费策略

为更好地平衡用户间对资源争用的矛盾,同时有效实现系统资源利用的效益最大化,不妨将上一小节中提出的 k 值看作用户与系统间的一种新的服务质量约定。基于该约定,我们提出如下差异化的资源使用计费方式:

(1) 差异化的资源计费方式

差异化资源计费:系统应根据用户对服务设定的 k 值,明确用户资源需求的上下限。将用户给出的需求下限内的资源作为系统应优先保证的部分,同时以较高资费标准收费;反之高于需求下限的部分,允许系统根据负载变化灵活提供,但其费用也相对较低。

此时,用户的开销除了与任务需求成正比外,也与其设定的 k_i 值紧密相关。k_i 值越低,计算任务就越有可能尽早完成,但付出的开销也越大。

(2) 基于 k 值的资源分配策略

基于以上提出的差异化的计费策略,本文提出如下资源分配策略:

a) 优先满足用户的基础需求。站在资源提供者的角度,这样做可以使系统收益最大化,在负载高峰时也能最大限度地保证更多的用户同时享有系统服务。

b) 当出现可用资源时,优先分配给 k_i 值高的用户(这些用户在系统忙时可能让出了大部分资源),使整个系统中的用户平均放缓程度降低。

c) 当有新的虚拟集群搭建请求时,由当前实际被放缓程度与 k 值差距最大的用户优先让出已占用的资源。

基于以上三个规则,用户在设定 k 值时,应该综合考虑任务的开销与所花费时间。当用户定义的 k 值较高时,其任务开销会降低,同时这也意味着该用户要接受在系统资源紧张时让出大部分资源的现实。另一方面,k 值设得较高并不一定会导致作业花费时间的显著增加,因为当负载较低时,系统也会主动为该用户分配更多相对廉价的资源。

1.3 问题的形式化描述

综合以上问题描述,多物理集群环境下虚拟集群资源共享分配问题可以归纳为线性规划问题,其表达如下:

$$\forall j \quad \sum_{i=1}^{J}\alpha_{ij}\leqslant C_j \tag{1}$$

$$\forall i \quad \sum_{j=1}^{H}\alpha_{ij}\leqslant\alpha_i \tag{2}$$

$$\forall i \quad \sum_{j=1}^{H}\frac{\alpha_{ij}}{\alpha_i}\geqslant(1-k_i) \tag{3}$$

$$\forall i \quad \sum_{j=1}^{H}\left\lceil\frac{\alpha_{ij}}{\alpha_i}\right\rceil\leqslant 1 \tag{4}$$

$$\forall i \quad \sum_{j=1}^{H}\frac{\alpha_{ij}}{\alpha_i}\geqslant Y \tag{5}$$

其中:C_j 为每个集群所能提供的计算资源,$j = 1, \cdots, H$;e_{ij} 代表第 i 个用户请求被分到物理集群 j 的可能性。式(1)是对每个独立的物理环境所能提供的计算资源总量的约束。式(2)、(3)表述了对虚拟集群占用资源的约束。

当限定每个虚拟计算环境请求只能分配到不多于 1 个物理集群环境时,$e_{ij} \in \{0, 1\}$;式(4)限定了同一虚拟集群内虚拟机只能部署于同一物理集群。如果允许同一虚拟集群内虚拟机部署于不同物理集群时,这一约束可以适当放宽。

该优化问题的目标是得到 Y 的最大值,其直观意义是让系统角度中所有用户被放缓的程度尽可能降低。对该问题求解,得到每个用户允许占用资源的最优方案后,可以通过调节虚拟机实例的配置来实现该方案。

2 虚拟集群静态资源分配算法

当问题规模较小、用户数与物理集群数较少时,虚拟集群资源分配问题的求解,可以借助于专用的计算工具得到最优解,如 GLPK、Lingo 等。当问题规模增大到一定程度后,精确求解很难在有限时间内完成。

2.1 一般性算法

(1) 松弛算法

为了解决问题,一种比较直接的做法是先按用户定义需求下限进行资源的预分配,然后再将剩余资源逐步分配给各个资源仍不足的任务。从直观意义上讲,这一方法能保证尽可能多地响应用户创建虚拟计算环境的请求。松弛法求解正是基于这一思想来进行资源分配。

在预分配阶段,为每一个用户分配资源时优先选择当前相对负载较低的物理集群,以保证在完成初次分配后,各物理集群的剩余资源相对均衡。在资源分配的第二阶段,应注意尝试将其全部剩余资源分配给已部署于该集群的各用户。同时,分配时优先考虑当前分配数量与需求上限相对差距较大的用户。这样一方面可保证系统的资源利用率最优,另一方面优先将资源分配给任务相对被延迟程度最大的用户,有效提高用户整体的服务满意度。

(2) 贪心算法

与松弛法从用户任务的最低需求出发,向上探索求解空间不同,贪心算法的基本思想是先按任务最大需求来预分配,只有当新的任务没有足够资源时,再从已分配的资源中收回部分资源,参与二次分配。这一方法在文献[14]中有比较详细的介绍。

2.2 混合背包算法

上一节中所提到的两种方法,都是直接从具体的用户需求出发,探索求解空间,忽略了从系统角度来观察多用户之间资源需求的相关性。为解决这一问题,本文提出了基于背包算法的改进算法。

(1) 资源相对稀缺性

为更好地刻画资源的稀缺程度,本文给出了如下定义:

$$\tau = \sum\nolimits_H C_j / \sum\nolimits_J \alpha_i$$

根据 1.2 节中所提出的资源分配策略,可以比较容易地得到如下所推论:

推论 1: 对于任意用户 i,如果 $(1 - k_i) > \tau$,系统只能按用户的最低资源需求 $(1 - k_i)\alpha_i$ 来为用户提供服务;

推论 2: 对于任意用户 i,如果 $(1 - k_i) \leqslant \tau$,该用户所能获得的资源在大于 $(1 - k_i)\alpha_i$ 的同时也将不超过 $\tau * \alpha_i$。

(2) 基于背包算法的改进

根据推论 1 与推论 2,不难看出,可以将问题的求解分为两个部分:针对固定需求部分的求解和针对可变需求的求解。其算法过程描述如图 1 所示。

1. 初始化 C_j, α_i, k_i;
2. 计算 τ;
3. 根据 k_i 值将用户需求按由小到大的顺序排序,并将其分成两个队列:队列 A ($k_i \leqslant 1 - \tau$) 和队列 B ($k_i > 1 - \tau$);
4. 计算队列 A 中每个用户的最低资源需求并按该值为每个用户在不同的物理集群分配相应资源;
5. 计算所有物理集群的剩余资源;
6. 以各集群的剩余资源和队列 B 中的用户请求构成一个新的资源分配问题(该问题是原问题的一个子问题);
7. 判断子问题的规模;
   a) 如果足够小,则使用 2.1 小节中提出的一般性算法进行资源分配问题的求解;
   b) 如果问题规模仍比较大,则将该问题作为一个新问题,重复步骤 1 到 6 求解;
8. 合并步骤 6、7 中得到的资源分配方案,得到整个问题的解。

图 1 混合背包算法过程描述

在图 1 所描述的过程中,步骤 4 中所面临的是针对固定数量需求的资源分配问题,与传统的背包

问题[15]相似，故可以采用一些比较传统的针对背包问题的求解算法。步骤 7 中关于问题规模大小的判断也可以根据针对实际问题的解决效果进行动态调整。

不难看出，算法中两个队列中各自的任务数取决于用户为任务设定的 k_i 值。当 $\tau > 1$ 时，算法直接演化为背包问题的求解，而当所有的 $k_i < (1-\tau)$ 时，系统将无法找到满足所有用户最低需求的资源分配方案。

2.3 标准化技术

上文中所提到的三种算法都能在有限时间内得到资源分配的可行解，但如果求解得到的结果需要通过改变虚拟机实例的配置来落实，无疑也将增加资源管理与作业调度的复杂性。为更好地解决这一问题，可以按照物理资源的处理能力，运用标准化的定制技术，将跨集群的资源统一规划为性能一致的虚拟机实例，保证每个虚拟机实例所占用的资源与其计算能力均大致相同。通过动态自适应的集群资源管理技术，可使虚拟集群自动适应这种虚拟机实例数的变化，也降低资源管理的复杂度。这一做法的另一优势是可以将原问题转化为整形规划问题，以进一步简化问题的求解。

3 实验

为便于数据分析和不同算法比较，本文相关实验均以简化的资源分配模型为基础进行设计。资源分配的主体简化为标准化虚拟机实例，它同时也是物理集群计算能力的度量标准。作为实验的初始条件，需要明确物理集群的个数及每个集群所能容纳的标准虚拟机实例数。为在资源分配中保持一致，用户的需求也通过所需的虚拟机实例数来表述。

数据随机生成中，用户虚拟机实例最大需求服从均值为 30，方差为 10 的正态分布，该均值约为实验中单一物理集群环境所能提供全部计算能力的一半。用户所能允许的最大被延迟程序 k_i 实验的另一个关键因素，出于一般性考虑，本文选择让该值服从范围为 0 -0.9 的均匀分布。

实验过程中，以增加用户数量的方式增加系统负载。考虑到用户资源需求小于各物理集群提供的计算能力之和时，每个用户都能比较容易地分配到所需的最大资源，实验将主要针对用户最大需求之和接近或超过系统实有资源总数后的资源分配情况。对比不同算法的效果，主要通过观察以下二个数据：(1)算法所达到 Y 值；(2)完成分配后，系统剩余资源总量。

为了充分检测算法的适用性，实验中给出了两组物理集群环境。其中较小规模的一组包含 5 个不同容量的物理集群，所能容纳的标准虚拟机实例总数为 320，其单个物理集群容量不小于 50；而较大规模的一组数据由 8 个计算能力不少于支持 90 个标准虚拟机实例的物理集群组成，其总的资源容量可以支持 1000 个标准虚拟机实例。

3.1 较小问题规模的实验数据

图 2 显示了当用户需求总量增加时 Y 值的变化。图 3 显示了用户需求总量增加时剩余资源数的变化。Y 值代表了完成资源分配后，任务延迟最大的用户所拥有资源与其最大资源需求的比值。Y 值为 1，说明所有用户按其最高需求分配到了相应资源。而算法会在保证所有用户的最低需求后，为获得资源相对较少的用户优先分配资源，所以 Y 值越低，可能表明用户获得的资源普遍较少。对比图 2 中数据可以看出，该 Y 值随用户数与用户需求总量的增加而明显降低。

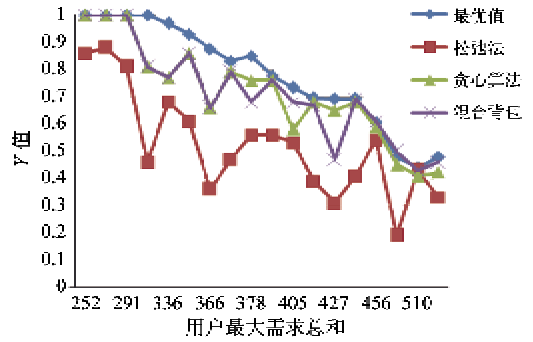

图 2 用户需求总量增加时 Y 值的变化

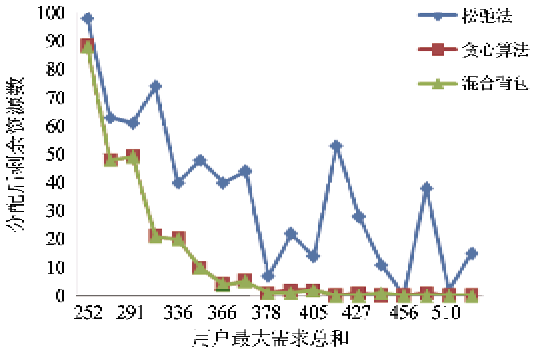

图 3 用户需求总量增加时剩余资源数的变化

对于小规模的问题，工具软件得到的线性规划最优解，是算法间进行比较的一个重要参考。从

图2中不难看出,贪心算法或是混合背包算法相对于松弛法在 Y 值上通常有 20% ~40% 的提升。结合图3中对分配完成后剩余资源的统计可以看出,同等环境下相对于其它两种算法,松弛法在用户满意度不高的同时其资源利用率也不高。

3.2 较大问题规模的实验数据

对于较大规模问题,图4与图5中数据表明,三种算法表现得都更为稳定,而松弛法的性能明显低于另外二种方法,仍有 20% ~50% 的差距。混合背包算法与贪心算法都能保证资源的最大化利用,但前者在 Y 值方面略有优势。

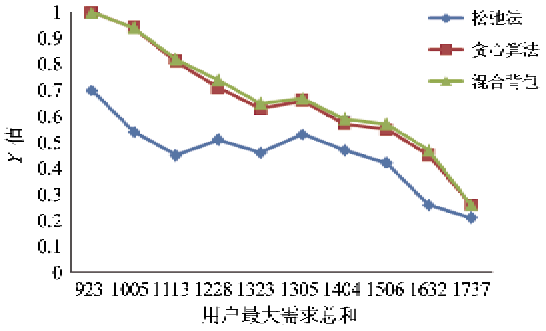

图4 用户需求总量增加时 Y 值的变化

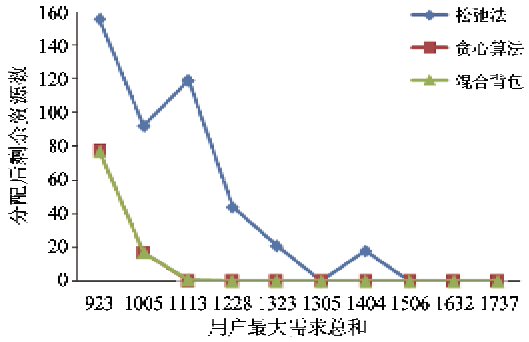

图5 用户需求总量增加时剩余资源数的变化

3.3 结果分析

综合以上实验数据不难看出,在两类问题规模下,混合背包算法与贪心算法在性能上均能达到近似最优,较松弛法有明显优势,资源利用率也更高。混合背包算法与贪心算法相比,在问题规模较小时前者稳定性较差,但在问题规模较大时前者性能相对略好于后者,说明混合背包算法更适用于大规模问题的求解,而贪心算法则更适用于小规模数据的求解。

当前资源分配问题中,由于限制了同一用户所属的虚拟机只能部署于单个物理集群,这种做法保证了用户虚拟环境不会因为地域上的分散而影响整个虚拟集群的性能。但问题域中所做的假设,使得单个用户任务最大需求与单个集群的计算能力的比值约为1:2 ~1:4,所以每个物理集群在任务分配时仅会为其部署3 ~5个虚拟计算环境。这一因素制约了算法性能的进一步提高,在资源规模较小时更体现为算法性能针对不同实例的较大波动。

4 结 论

本文针对虚拟集群条件下的资源分配问题,提出了一种以用户任务被放缓的程度为核心指标的资源分配机制。突出了不同负载条件下,用户占用资源的弹性变化。在不同用户服务质量约束下,保证了系统资源的效能优化与用户开销最低。同时针对不同问题规模,对所提出的算法进行了验证。相关实验数据表明了混合背包算法在效能上的优越性及近似最优的特性。

出于降低策略实现的复杂度考虑,本文当前的研究保留了同一虚拟集群中各虚拟机实例应处于同一物理集群的约束。但在单个物理集群计算能力受限时,放松该约束条件也将提高系统的资源利用率和用户满意度。如何将现有的算法针对该约束条件进行扩展,并逐步实现动态负载条件下的资源分配与调度,将是下一步研究的重点方向。

参考文献

[1] Foster I, Freeman T, Keahey K, et al. Virtual clusters for Grid communities. In: Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), Singapore, 2006. 513-520

[2] Lizhe W, von Laszewski G, Jie T, et al. Grid virtualization engine: design, implementation, and evaluation. *Systems Journal, IEEE*, 2009,3: 477-488

[3] Murphy M, Abraham L, Fenn M, et al. Autonomic Clouds on the Grid. *Journal of Grid Computing*, 2010, 8:1-18

[4] Jeffrey S, David C, Aravind M, et al. Concurrent direct network access for virtual machine monitors. In: Proceedings of the International Symposium on High-Performance Computer Architecture, Scottsdale, USA, 2007. 306-317

[5] Nesbit K J, Moreto M, Cazorla F J, et al. Multicore resource management. *Micro, IEEE*, 2008, 28:6-16

[6] Cherkasova L, Gardner R. Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In: Proceedings of the annual conference on USENIX Annual Technical Conference, ed. Anaheim, CA: USENIX As-

sociation, 2005. 24-24

[7] Menasce D A, Casalicchio E. QoS in grid computing. *IEEE Internet Computing*, 2004, 8:85-87

[8] Yeo C S, Buyya R. A taxonomy of market-based resource management systems for utility-driven cluster computing. *Software: Practice and Experience*, 2006, 36:1381-1419

[9] Kumar S, Dutta K, Mookerjee V. Maximizing business value by optimal assignment of jobs to resources in grid computing. *European Journal of Operational Research*, 2009, 194:856-872

[10] Li C, Li L. Multi-level scheduling for global optimization in grid computing. *Computers & Electrical Engineering*, 2008, 34:202-221

[11] Amril N. A cost efficient framework for managing distributed resources in a cluster environment. In: Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Dalian, China, 2009. 29-35

[12] Yang C, Tianyu W, Jianxin L. An efficient resource management system for on-line virtual cluster provision. In: Proceedings of the IEEE International Conference on Cloud Computing, Bangalore, India, 2009. 72-79

[13] Elmroth E, Tordsson J. Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions. *Future Generation Computer Systems*, 2008, 24:585-593

[14] Stillwell M, Schanzenbach D, Vivien F, et al. Resource Allocation Using Virtual Clusters. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai, China, 2009. 260-267

[15] Leinberger W, Karypis G, Kumar V. Multi-capacity bin packing algorithms with applications to job scheduling under multiple constraints. In: Proceedings of International Conference on Parallel Processing, Aizu-Wakamatsu City, Japan, 1999. 404-412

Research on resource allocation of virtual cluster based on QoS

Liu Feifei[*,**], Dong Xiaoshe[*], Tian Hongbo[*]

([*]School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049)

([**]Engineering University of Armed Police Force of China, Xi'an 710086)

Abstract

To effectively allocate computer resources under virtual cluster-based high performance computing environments to satisfy users' different QoS requirements, this paper presents a novel dynamically-adjustable resources allocate strategy. The strategy regards the slowdown degree of users' job as a new metric of QoS and the key role of resources allocation. From the perspective of resource providers, this strategy guarantees low expenditure of users and high resource profit of users service. To implement the strategy, a mixed bin packing algorithm is also presented, which is the combination of the traditional bin-packing algorithm and some general algorithms. The experimental results show that the mixed bin packing algorithm is more effective than the previous algorithms, especially in a large application scale background.

Key words: resource allocation, virtual cluster, quality of service (QoS)