International Journal of Hybrid Information Technology
Vol.6, No.6 (2013), pp.333-344
http://dx.doi.org/10.14257/ijhit.2013.6.6.30

Constraint Programming based Virtual Cloud Resources Allocation Model

Long Zhang, Yi Zhuang and Wei Zhu

College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

sqzl_nuaa@126.com, zhuangyi@nuaa.edu.cn

Abstract

Cloud computing is growing in popularity among computing paradigms, and many corporate data centers are in the process of adopting a cloud computing architecture where resources are provisioned on a per-demand basis. The goal of a cloud infrastructure provider is to maximize its profit by minimizing the amount of violations of Quality-of-Service(QoS), and, at the same time, by lowering infrastructure costs. However, allocating virtual resources in cloud computing environment is very complex, and it's very difficult to manage virtual resources when the corporate data centers scale up. In this paper, we propose a dynamic constraint programming based virtual resources allocation model, which not only takes into account meeting user's QoS requirements, but also considers the cost of virtual cloud resources. We utilize constraint programming approach to solve the virtual resource provisioning and packing problem, and we validate our model through simulation experiments. We show, through simulation results, that our model is able to efficiently allocate and manage the virtual resources of the cloud platform. Compared with other traditional models, our model can significantly reduce QoS violations and achieve lower resource usage costs.

Keywords: *cloud computing, virtualization, resource allocation, constraint programming, SLA*

1. Introduction

Cloud computing is a kind of integrated network computing model, which is based on parallel computing, distributed computing and grid computing [1]. The providers of cloud computing offer their services according to several fundamental models, *i.e*., infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). And IaaS is the most basic and each higher model abstracts from the details of the lower models. Cloud computing data centers integrate various resources of the network together, which make the computing migrate from the client sides to the cloud platform. Cloud computing is provided to users through the network in the form of services, which means that the computing power can also be viewed as a kind of commodity, just like water and electricity that are out-of-the-box and low-cost [2]. By utilizing virtualization technology [3], cloud data center is able to consolidate all kinds of computing resources, storage and network resources to form a dynamic virtual resource pool. And cloud data centers use virtual resources allocation and management technology to achieve automatic deployment, dynamic scalability and on-demand allocation of cloud computing resources. The cloud users can adopt on-demand and pay-as-you-go methods to get all kinds of virtual resources. Therefore, research on allocation and management of the virtual resources is one of the hotspots and difficulties of research.

ISSN: 1738-9968 IJHIT

U.S. *Economist Special Report* shows that the current physical resource utilization of cloud data centers is generally only maintained at 5% to 20% [4]. Therefore, a large number of servers are in idle state. With the advent of server virtualization technology [5], the same physical server can run multiple virtual machines, and it can dynamically adjust the capacity of virtual machine. Server virtualization technology can also make migrations of virtual machines between different physical servers possible, which greatly improve the utilization of physical resources. However, the current study on the technology of resources allocation mainly focuses on performance improvement of applications or the costs of resources. And most research does not consider how to improve the performance while reducing the cost of resources. Cloud computing services must meet the service level agreement (SLA) [6]. SLA is negotiated between the users and the cloud resources supplier, which defines the strict performance metrics, such as response time, throughput, mean time between failures (MTBF), and so on. Therefore, under the premise of satisfying performance index, how to reduce the cost of resources is still a problem to be solved.

In view of the mentioned problem above, we propose an efficient virtual resource allocation model: constraint programming based virtual cloud resources allocation model (VCRA-CP). In this model, the virtual resources allocation problem is modeled as a constraint satisfaction and optimization problem, and using constraint programming method to solve the model. The model can make different tradeoffs between the fulfillment of the performance goals of the applications and the cost of the required resources. And it also supports for heterogeneous applications and workloads, such as CPU-intensive, memory-intensive, I/O-intensive applications, etc. The simulation shows that our model is able to efficiently allocate and manage the virtual resources of the cloud platform, and compared with other traditional models, our model can significantly reduce QoS violations and achieve lower resource usage costs.

The remainder of this paper is organized as follows. Section 2 presents the related works. Next, we present the architecture of the virtual resource allocation model in Section 3. Section 4, we show how to solve the model. We cover simulation experiment and results analysis in Section 5. Finally, Section 6 concludes and presents future work.

2. Related work

In recent years, the scholars in the field of cloud resource allocation models and algorithms have carried out a lot of research. The research is mainly on resources allocation and management algorithms to improve the performance of applications [7, 8], resources allocation and management methods to improve the utilization of resources [9, 10], economics based cloud resources allocation model [11, 14], resource allocation and management model to reduce energy consumption as well as resource costs [12, 13].

*Va*n *et al.*, propose a dynamic virtual resource scheduling model AVRM [7, 8], which studies on how to select an appropriate virtual machine for an application and how to suitably deploy the virtual machine on the server in aspect of the two-dimensional virtual resources(CPU, RAM). In order to obtain optimal scheduling result, the scheduling problem is converted into constraint satisfaction problem, but the model only takes CPU, RAM resources into account without considering other types of physical resources. From the view of economic theory, *Gao et al.*, propose an economics based cloud computing framework [14], and design a SLA-based economic model of cloud resources management. The model provides feedback of the economic incentives for the cloud consumers and suppliers, which improves the utilization of resource, avoids reconstruction of information system, and contributes to the efficient management and configurations of cloud resources. Although, the

model can meet the requirements of QoS, the architecture and the model is too idealistic, and many implementation issues to be further discussed.

Vijayakumar et al., propose a dynamic automated resource allocation model [15], which can effectively deal with unknown data distribution and rate, and they verify the validity of the model. *Yuan* et al. propose an intelligent virtual resource allocation engine [16], which features feasibility, flexibility, reusability and configurability. The intelligent engine can handle heterogeneous physical and logical resources, and be able to provide resources according to user needs. But it lacks the capability of exception handling. *Wang* and *Cardosa* propose two energy-efficient virtual resources allocating method of POPA [17] and PC-SU [18], respectively. But the two methods are all NP-hard problem and use a large number of sorting algorithms, which result in high complexity. Li *et al.*, propose a virtual resource allocation algorithm called *EnaCloud* [19]. In spite of reducing energy consumption by deploying the application according to the dynamic energy consumption, the algorithm may lead to a large number of virtual machine migrations in the process of resource allocation.

In conclusion, research on the field of virtual resources allocation has achieved some results, but there are still some shortcomings, for instance, the diversity of cloud resources is not fully considered, the user's quality of service has not been effectively guaranteed, the utilization of various cloud resource is low, and so on. To solve these problems, we propose a multi-dimensional virtual cloud resource allocation model VCRA-CP. In this model, the virtual resources allocation problem is modeled as a constraint satisfaction and optimization problem. The model is also capable of ensuring the effectiveness of the quality of customer service while improving the utilization of various resources and minimizing the cost of resources. Simulation results show that, for different application requirements, the model can efficiently allocate virtual resources, effectively improve the utilization of resources and reduce the resource costs.

3. Virtual Resource Allocation

3.1. VCRA-CP model framework

The model we proposed is based on the literature [7], and the framework of VCRA-CP model is shown in Figure 1. The datacenter of VCRA-CP model is consist of a group of physical machines (PM), which form a cluster. Each physical machine in the cluster can run multiple virtual machines (VM), that are managed by the hypervisor residing in every PM. We assume that the number of physical machines is fixed, and they all belong to the same cluster, so that virtual machines can perform a live migration between two arbitrary PMs. A *Performance Monitor* is mainly used to monitor the performance of the current application, such as response time, load status. The expression $f_A(N_A, L_A)$ is the performance evaluation function of application A, that is used to evaluate the performance based on the current workloads L_A and the allocated virtual resources N_A. We assume that the granularity of resource allocation is performed with a VM. That is to say that a running VM is associated with one and only one application. When the workloads of an application become heavy, the system must choose VM resources from a set of pre-defined VM types to assign to the application. Any application could not request a VM with an arbitrary resource capacity in terms of CPU power, memory size and network bandwidth. Each pre-defined VM type comes with a fixed CPU power, memory capacity and network bandwidth (*e.g.*, 2000MIPS of CPU power, 2GB of memory size and 100Mbps of network bandwidth). Each type of pre-defined VM mainly differs in CPU capacity, memory capacity and network bandwidth.

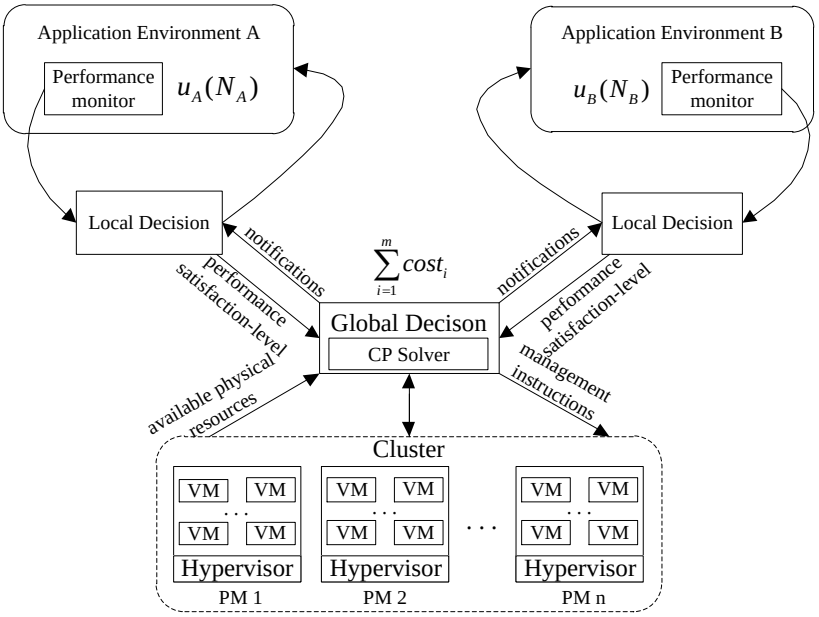

Figure 1. VCRA-CP model framework

In VCRA-CP model an *Application Environment* is the running environment of an application, which provides the necessary resources for the application. And an *Application Environment* is also associated with specific performance goals specified in the SLA contract. For example, for a strict real-time requirements application, the *Application Environment* demanding for virtual resources correlates closely with the response time T and throughput D defined by SLA. The topology of an *Application Environment* can span one or more VMs. Figure 2 is the schematic of virtual resources allocation. In this figure we can see that *Application Environment A* is consist of *Web* and *DB* (database application) two parts, which spans two VMs running on two physical servers. If the resources assigned to application *A* does not meet its performance requirements, the system will allocate more resources to application *A*. When the workloads of application *A* decreases and the resources assigned to application *A* are sufficient to meet its performance requirements, the system will recycle excess resources allocated to the application *A*.

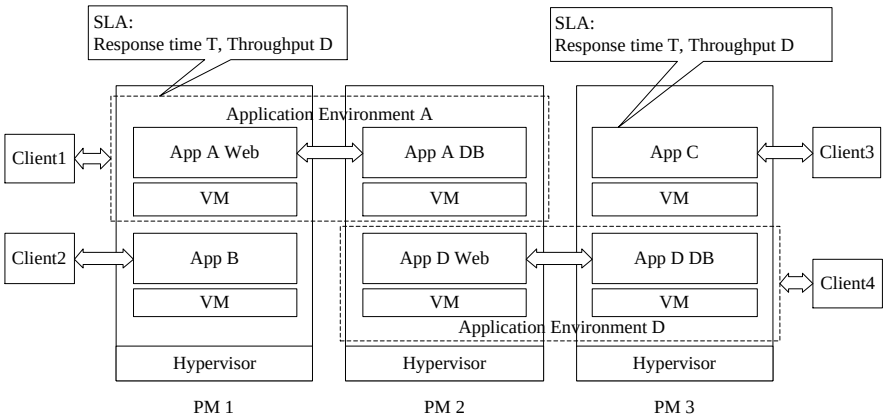

Figure 2. Virtual resource allocation schematic

3.2. Virtual resource allocation process

Our proposed model is based on a two-level structure. The first level structure is the *global decision*, which corresponds to VM provisioning phase and packing phase. And the second is the *local decision*. Each *Application Environment* includes a *local decision*, which is used to evaluate the performance of applications on the basis of the current workloads and determine whether the system need to allocate more VMs or release the idle VMs. For example, if the SLA of application a_i defines that the application shall not exceed the average response time T(milliseconds). Then we can treat T as the performance index. Within a period of time, the *Local Decision* of application a_i uses performance evaluation function f_i to assess the response time of application a_i based on the current workloads and resources allocated to the application. If the current response time of application a_i exceeds the performance index T, then the application a_i can't satisfy SLA requirement. The system need assign more resources to application a_i to satisfy SLA requirement.

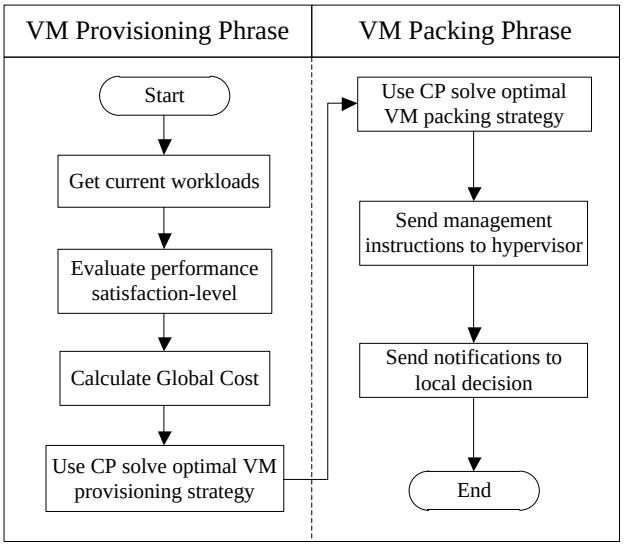

Figure 3. Resource allocation flowchart

Global decision is the control center of virtual resources allocation, and interacts with all *local decision*. *Global decision* is capable of arbitrating resources requirements coming from every application without knowing the nature of the application. The performance satisfaction level of each *local decision* and physical resources are the input of *Global decision*, and it uses constraint programming solver to solve the input. Based on the output, *global decision* sends management instructions of resources allocation to the hypervisor and resources allocation notifications to the *local decision* of each application. Management instructions of resources allocation include starting VMs, stopping VMs and migrating VMs. The resources allocation notifications are consist of new virtual machine being assigned to an application, an existing VM being upgraded or downgraded (*e.g.*, virtual machine type has been changed) and VMs belonging to an application being preempted (*e.g.*, release idle virtual machines). Figure 3 shows the resources allocation flowchart of VCRA-CP model. For VM provisioning and packing problems both are instances of the NP-hard problems which are difficult to solve. To solve both problems easily, we select *Constraint Programming* (CP) method that solve a problem by stating relations between different variables in the form of constraints which must be satisfied by the solution.

4. VCRA-CP model

Constraint Satisfaction Problems (CSPs) are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations. A large number of problems in *Artificial Intelligence* (AI) and other areas of computer science can be viewed as special cases of the constraint-satisfaction problem. Some examples are map coloring, machine vision, *Sudoku*, temporal reasoning, belief maintenance, floor plan design, scheduling, graph problems, the planning of genetic experiments, and eight queens puzzle problem. A number of different approaches have been developed for solving these problems. Some of them use constraint propagation to simplify the original problem. Others use backtracking to directly search for possible solutions. Some are a combination of these two techniques. Generally, solving CSPs can be divided into two steps. The first step is to use a set of variables and a set of constraints expression to represent the problem. The second is to utilize constraint propagation and search algorithms to solve the problem. Considering the limited physical resources and the resources cost of cloud computing data centers, we can view virtual resource allocation problem as a CSP.

4.1. Related concepts and definitions

Constraint programming allows solving combinatorial problems modeled by a Constraint Satisfaction. Formally speaking, a CSP is defined by a triplet (X, D, C).

Variables: $X=\{X_1, X_2, \ldots, X_n\}$ is the set of variables of the problem to be solved.

Domains: D is a function which associates to each variable X_i its domain $D(X_i)$, *i.e.*, the set of possible values that can be assigned to X_i. The domain of a variable is usually a finite set of integers or real numbers, and a domain can also be continuous or made of discrete set values.

Constraints: $C=\{C_1, C_2, \ldots, C_m\}$ is the set of constraints. A constraint C_j is a relation defined on a subset $X^j = \{X_1^j, X_2^j, \ldots, X_q^j\} \subseteq X$ of variables which restricts the possible tuples of values $(v_1, v_2, \ldots, v_q)$ for these variables: $(v_1, v_2, \ldots, v_q) \in (D(X_1^j) \times D(X_2^j) \times \ldots \times D(X_q^j))$ and must satisfy constraint C_j. Such a relation can be defined explicitly (ex: $(X_1, X_2) \in \{(0, 1), (1, 0)\}$) or implicitly (ex: $X_1 + X_2 \leq 1$).

Solving a CSP consists in finding a tuple $v = (v_1, v_2, \ldots, v_n) \in D(X)$ on the set of variables which satisfies all the constraints. Constraint satisfaction optimization problem (CSOP) can be seen as a CSP with an objective, which requires all constraints being satisfied while minimizing or maximizing the objective. For CSOP, one needs to define an objective function f : D(X)→$\boldsymbol{R}$(real numbers). An optimal solution is a solution tuple of the CSP that minimizes (or maximizes) function f.

In VCRA-CP model, we define vector $\boldsymbol{A} = (a_1, a_2, a_i, \ldots, a_m)$ represents the set of applications and $\boldsymbol{P} = (p_1, p_2, p_j, \ldots, p_q)$ denotes the set of PMs in the cloud datacenter. Vector $\boldsymbol{S} = (s_1, s_2, s_k, \ldots, s_c)$ denotes that there are c types of VM available in the data center, where $s_k = \left(s_k^{cpu}, s_k^{ram}, s_k^{bandwidth}\right)$ specifies the CPU capacity of VM expressed in MIPS, the memory size expressed in Mb and the bandwidth of network expressed in Mbps, respectively. In the following two subsections, we describe the constraints and objective functions of VCRA-CP model detailed.

4.2. Local decision

Based on the allocated resource capacity (CPU, RAM, Bandwidth) and the current workloads, *Local Decision* use performance evaluation function to assess the level of service

achieved. And the level of service achieved is mapped to a value of performance satisfaction-level, which is used to interact with the *Global Decision*. Let yt_{a_i} denotes the performance index of application a_i and $yr_i = f_i(N_i, L_i)$ denotes the value of performance evaluation of application a_i, where f_i is the evaluation function. L_i is the current workloads. N_i is the virtual resources allocated to application a_i: $N_i = (n_{i1}, n_{i2}, \dots, n_{ik}, \dots, n_{ic})$ where n_{ik} is the number of VMs of type s_k allocated to application a_i. We define $u_i = f_i'(yr_i, yt_{a_i})$ as the performance satisfaction-level of application a_i, where function f_i' calculate the application performance satisfaction-level based on the performance index yt_{a_i} as well as yr_i. If the current application performance yr_i has reached the required performance metric yt_{a_i}, the application performance satisfaction-level $u_i = 1$, otherwise $u_i < 1$.

4.3. Global decision

The main two responsibilities of *Global Decision* are allocating VMs for each application a_i, and placing the allocated VMs on PMs to minimize the number of physical machines. In this paper, the main two tasks are called VM provisioning and VM packing, respectively. During VM provisioning phase, it will take the resource request as the input and the output is the virtual machine allocation vector N_i for each application. During VM packing phase, it will take vector N_i as the input and the output is the VMs packing vector H_j.

4.3.1. VM provisioning: VM provisioning is used solve the vector N_i for each application a_i and minimize the global cost U_{cost}. For each application, let $N_i^{max} = (n_{i1}^{max}, n_{i2}^{max}, \dots, n_{ik}^{max}, \dots, n_{ic}^{max})$ denotes the maximum number of VMs of each type that the system can provide, and the total demand for all the number of VMs can't exceed T_i^{max}. Then the VM provisioning problem can be modeled as CSOP:

$$U_{cost} = Minimize \ \sum_{i=1}^{m}\left(\alpha_i.(u_i - 1)^2 + \varepsilon.cost(N_i)\right) \quad (1)$$

s.t.

$$n_{ik} \le n_{ik}^{max}, 1 \le i \le m \text{ and } 1 \le k \le c \quad (2)$$

$$1 \le \sum_{k=1}^{c} n_{ik} \le T_i^{max}, 1 \le i \le m \quad (3)$$

$$\sum_{i=1}^{m}\sum_{k=1}^{c} n_{ik}.s_k^{cpu} \le \sum_{j=1}^{q} C_j^{cpu} \quad (4)$$

$$\sum_{i=1}^{m}\sum_{k=1}^{c} n_{ik}.s_k^{ram} \le \sum_{j=1}^{q} C_j^{ram} \quad (5)$$

$$\sum_{i=1}^{m}\sum_{k=1}^{c} n_{ik}.s_k^{bandwidth} \le \sum_{j=1}^{q} C_j^{bandwidth} \quad (6)$$

where formula(1) is the objective function, which is a weighted sum of the performance satisfaction-level function u_i and resources cost function $cost(N_i)$. The variable α_i is the weight of each application a_i, where $0 < \alpha_i < 1$, and $\sum_{i=1}^{m} \alpha_i = 1$. The higher the weight of an application is, the higher priority of resources allocation is. The coefficient ε allows the system to make different tradeoffs between the performance of applications and resources cost. We can divide formula(1) into two parts: $J_{p_i} = \alpha_i.(u_i - 1)^2$, $J_{c_i} = \varepsilon.cost(N_i)$. If the performance satisfaction-level $u_i = 1$,then $J_{p_i} = 0$, otherwise $J_{p_i} > 0$, which can be viewed as the performance loss cost due to not satisfying the performance index. We can treat J_{c_i} as the resources cost, and the more resources the system allocates, the higher the cost is.

In order to facilitate the management of VMs, formula (2) and (3) are used to limit the number of each type of VMs and the total number of VMs assigned to each application a_i.

Formula (4), (5) and (6) indicate that all the resources allocated should be less than all the physical resources owned by the system, where C_j^{cpu}, C_j^{ram} and $C_j^{bandwidth}$ are the CPU, RAM and Bandwidth capacity of PM p_j. In VCRA-CP model, we define the expression of cost function *cost*(N_i) as:

$$Cost(N_i) = \frac{\sum_{k=1}^{c} n_{ik}.\left(0.7\times s_k^{cpu}+0.2\times s_k^{ram}+0.1\times s_k^{bandwidth}\right)}{\sum_{j=1}^{q}\left(0.7\times C_j^{cpu}+0.2\times C_j^{ram}+0.1\times C_j^{bandwidth}\right)} \quad (7)$$

where the coefficient (0.7,0.2,0.1) denote the cost-weight of CPU, RAM and Bandwidth. The output of VM provisioning phase is a group of vectors N_i that satisfy the constraints (2), (3), (4), (5), (6) and minimize U_{cost}. By comparing the vectors N_i with the those vectors N_i' calculated during the previous cycle, the global decision is able to determine which applications need to create VMs, destroy VMs and change the type of VMs. The placement of the newly created VMs and the migrations of VMs is mainly handled by the VM packing phase described following.

4.3.2. VM packing: The VMs placement issues can be seen as a classic *Bin Packing Problem*, where different computing nodes represent different bins. And VMs are the items to be packed and the size of the bin can be seen as the capacity of physical node. The objective of VMs placement is to minimize the number of physical nodes. The VM packing phase takes the VM allocation vectors N_i as input and converts the vector N_i into a single vector $V = (vm_1, vm_2, \dots, vm_l, \dots, vm_v)$ that denotes all the currently running VMs. For each PM $p_j \in P$, the bit vector $H_j = (h_{j1}, h_{j2}, \dots, h_{jl}, \dots, h_{jv})$ denotes which VMs assigned to PM p_j, that is, if $h_{jl} = 1$, then PM p_j is responsible for running vm_l. Let $R = (r_1, r_2, \dots, r_l, \dots, r_v)$ be the resource capacity(CPU, RAM, Bandwidth) of all VMs, where $r_l = (r_l^{cpu}, r_l^{ram}, r_l^{bandwidth})$. Then the VM packing problem can be modeled as CSOP:

$$M = Minimize \sum_{j=1}^{q} m_j, where\ m_j = \begin{cases} 1, \exists vm_l \in V\ and\ h_{jl} = 1 \\ 0, otherwise \end{cases} \quad (8)$$

s.t.

$$\sum_{l=1}^{v} r_l^{cpu}.h_{jl} \le C_j^{cpu}, 1 \le j \le q \quad (9)$$

$$\sum_{l=1}^{v} r_l^{ram}.h_{jl} \le C_j^{ram}, 1 \le j \le q \quad (10)$$

$$\sum_{l=1}^{v} r_l^{bandwidth}.h_{jl} \le C_j^{bandwidth}, 1 \le j \le q \quad (11)$$

$$\sum_{j=1}^{q} h_{jl} = 1,\ 1 \le l \le v \quad (12)$$

Formula (8) is the objective function of VM packing problem, which is to minimize the amount of PMs. Formula (9), (10) and (11) indicate that the total resources capacity of all VMs running on PM p_j should be less than the total physical resources capacity of PM p_j, where C_j^{cpu}, C_j^{ram} and $C_j^{bandwidth}$ are the CPU, RAM and Bandwidth capacity of PM p_j. Formula (12) means that each VM vm_l must be allocated to a single physical machine. For the formula (8), as long as there is a VM running on PM p_j, then $m_j = 1$. Only when there is none VM running on PM p_j, then $m_j = 0$. The solving of the VM packing problem produces the placement vector H_j that indicates all the VMs should be placed on which physical machines.

5. Simulation Experiment and Result Analysis

In order to illustrate and validate the proposed model, we conduct some simulation experiments which show how the system allocates resources to different applications with different performance satisfaction-level. Our simulation experiments rely on the platform of *CloudSim* [20] and use *JaCop* [21] constraint solver to solve the VM provisioning and packing problems. Our simulation environment consists of five physical machines of the same capacity. The hardware configuration of each physical machine is shown in Table 1. We run two applications A and B in the simulated environment, which represent the online game and web application, respectively. The application A has high real-time requirements, and the performance index is the average response time of requests. Only when the average response time of application A is less than or equal to the performance index τ_{SLA}, the performance satisfaction-level of application A is equal to 1, otherwise 0. The performance index of application B is also the average response time of requests and the workload is measured by the number of requests per second. The performance satisfaction-level function of application A and B are shown in figure 4(a) and 4(b), respectively. Table 2 shows the capacity of different VMs types. When system allocates resources to different applications, it can only select the pre-defined VM type from Table 2. The coefficient ε is set to 0.06 and the performance index τ_{SLA} is set to 100ms, other experiment parameters shown in Table 3.

Table 1. Physical machine configuration

PM types	CPU(MIPS)	RAM(MB)	Bandwidth(Mbps)
P_1	4000	4000	1000
P_2	4000	4000	1000
P_3	4000	4000	1000
P_4	4000	4000	1000
P_5	4000	4000	1000

Table 2. Virtual machine configuration

VM types	CPU(MIPS)	RAM(MB)	Bandwidth (Mbps)
S_1	500	500	100
S_2	1000	1000	100
S_3	2000	2000	100
S_4	4000	4000	100

Table 3. Experiment parameter setting

App	α	T^{max}	n_1^{max}	n_2^{max}	n_3^{max}	n_4^{max}
A	0.7	18	5	5	5	5
B	0.3	18	5	5	5	5

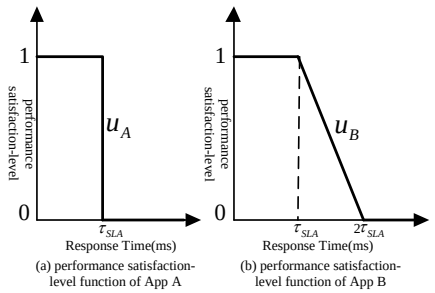

Figure 4. Performance satisfaction-level function

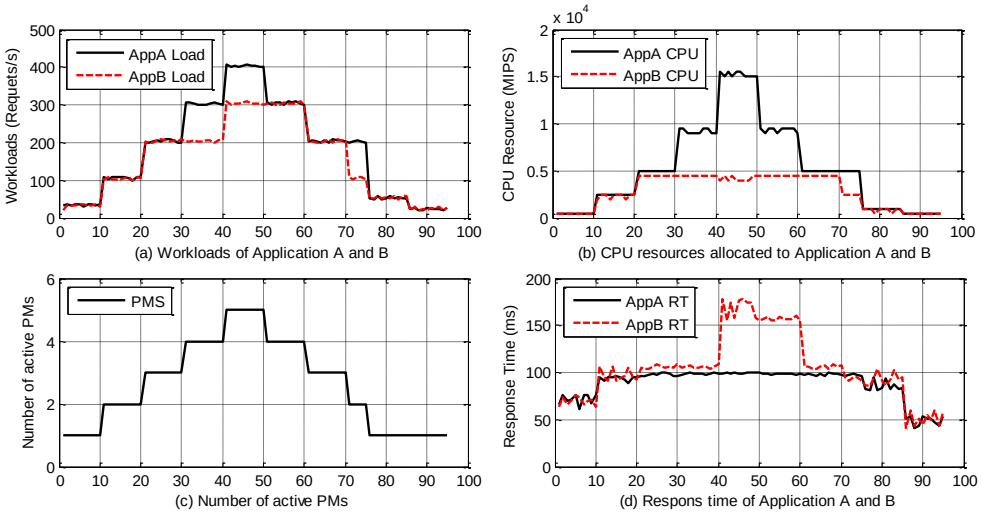

Figure 5. Simulation results

Figure 5 is the simulation results, where the horizontal coordinate denotes the time (measured by minutes). Figure 5(a) shows the workloads (measured by user requests per second) of application A and B in a period of time. Between time 0 and 10, as workloads of application A and B are both low, the resources demands are negligible(shown in Figure 5(b)), hence there is only on PM launched, and the average response time of application A and B both are less than 100ms(as shown in Figure 5(c), 5(d)). Between time 10 and 40, the workloads increase, so the system allocates more resources to the two applications. However, the workloads are not too heavy, the system still has sufficient resources to be allocated and the response time of application A and B are about 100ms. But between time 40 and 60, the workloads are relatively heavy, all the five physical machines are working. From Figure 5(b) and d (d), we can see that resources allocated to the application A are more than application B and the response time of application A is still not greater than 100ms, but the response time of application B exceeds 100ms. This is mainly due to application A has the higher resource allocating priority than application B, so when there are not enough system resources, the system allocates resources to application A firstly.

We also compare our model with the dynamic virtual resource scheduling model AVRM and the virtual machine allocation model VMPA-SP [22] based on stochastic programming. Figure 6 is the comparison of the three different models. The results show that our proposed model VCRA-CP can achieve lower resources cost during the resources allocation phase.

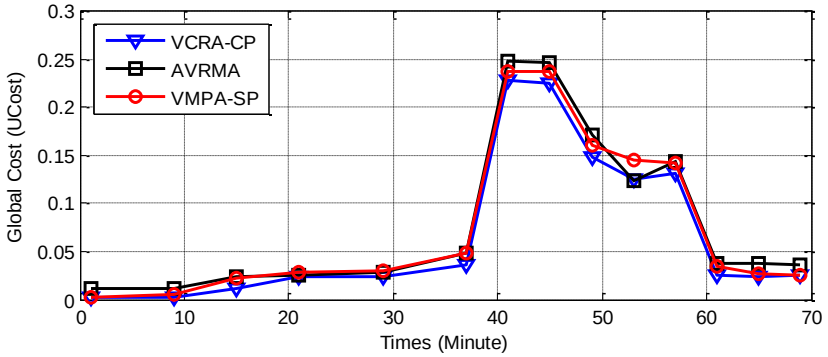

Figure 6. Comparison of different models

6. Conclusion

This paper does research on the problem of virtual cloud resource allocation model. We propose a constraint programming based virtual resource allocation model VCRA-CP, which takes into account both user's QoS requirements and the cost of virtual cloud resources. We utilize constraint programming approach to solve the VM provisioning and packing problems, and we validate our model through simulation experiments. Simulation results show that the model can efficiently allocate and manage the virtual resources of the cloud platform. And compared with other traditional models our model can significantly reduce QoS violations and achieve lower resource usage costs. But with the types of VMs, the number of PMs in datacenters and the size of applications increasing, the constraint solver will search larger space to obtain the best solution. Hence, in this situation it's inefficient to solve the problems. In order to manage and allocate cloud resources more efficiently and accurately, in the future work, we will further optimize the model and use better algorithm to solve the proposed model.

References

[1] M. Armbrust, A. Fox and R. Griffith, "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, **(2010)**, pp. 50-58.
[2] R. Buyya, C. S. Yeo and S. Venugopal, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation computer systems, vol. 25, no. 6, **(2009)**, pp. 599-616.
[3] G. Xiao-ming, H. Jie and C. Long, "Principle and implementation of virtualization technology", Electronic Industry Press, Beijing, **(2012)**.
[4] L. Siegele, "Let It Rise: A Special Report on Corporate IT", The Economist, **(2008)**.
[5] C. Ying, W. Qin-bo and J. Xing, "Virtualization and cloud computing", Electronic Industry Press, Beijing, **(2009)**.
[6] H. Ludwig, A. Keller and A. Dan, "Web service level agreement(WSLA) language specification", IBM Corporation, **(2003)**, pp. 815-824.
[7] H. N. Van, F. D. Tran and J. M. Menaud, "SLA-aware virtual resource management for cloud infrastructures", Proceedings of the 9th International Conference on Computer and Information Technology, **(2009)**, pp. 357-362.
[8] H. N. Van, F. D. Tran and J. M. Menaud, "Autonomic virtual resource management for service hosting platforms", Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, **(2009)**, pp. 1-8.
[9] W. Gui-yi, A. Vasilakos A and Z. Yao, "A game-theoretic method of fair resource allocation for cloud computing services", The Journal of Supercomputing, vol. 54, no. 2, **(2010)**, pp. 252-269.
[10] L. Wei-Wei, J. Z. Wang and L. Chen, "A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing", Procedia Engineering, vol. 23, **(2011)**, pp. 695-703.
[11] Í. Goiri, J. Guitart and J. Torres, "Economic model of a Cloud provider operating in a federated Cloud", Information Systems Frontiers, vol. 14, no. 4, **(2012)**, pp. 827-843.
[12] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", Future Generation Computer Systems, vol. 28, no. 5, **(2012)**, pp. 755-768.
[13] M. Mezmaz, N. Melab and Y. Kessaci, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems", Journal of Parallel and Distributed Computing(JPDC), vol. 71, no. 11, **(2011)**, pp. 1497-1508.
[14] G. Hong-qing, X. Ying, "Research on cloud resource management model based on economics", Computer Engineering and Design, vol. 31, no. 19, **(2010)**, pp. 4139-4212.
[15] S. Vijayakumar, Q. Zhu and G. Agrawal, "Automated and dynamic application accuracy management and resource provisioning in a cloud environment", Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (GRID), **(2012)**, pp. 33-40.
[16] J. B. Yuan, Y. C. Lee and W. Wu, "Building an intelligent provisioning engine for laaS cloud computing services", Proceedings of the 13th Asia-Pacific Network Operations and Management Symposium(APNOMS), **(2011)**, pp. 1-6.
[17] Y. WANG, X. WANG, "Power optimization with performance assurance for multi-tier applications in

virtualized data centers", Proceedings of the 39th International Conference on Parallel Processing Workshops, **(2010)**, pp. 1-8.
[18] M. CARDOSA, M. KORUPOLU and A. SINGH, "Shares and utilities based power consolidation in virtualized server environments", Proceedings of IFIP/IEEE International Symposium on Integrated Network Management, **(2009)**, pp. 327-334.
[19] B. LI, J. X. LI and J. P. HUAI, "Enacloud: an energy-saving application live placement approach for cloud computing environments", Proceedings of the 9th IEEE International Conference on Cloud Computing, **(2009)**, pp. 17-24.
[20] R. BUYYA，R. RANJAN and R. N. CALHEROS, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: challenges and opportunities", Proceedings of the 7th high performance computing and simulation conference, Leipzig: IEEE, **(2009),** pp. 1-11.
[21] K. Kuchcinski, R. Szymanek, "Jacop-java constraint programming solver", JaCop Library User's Guid, **(2012)**.
[22] Xie Wen-jing, Tang Zhuo and Yang Liu, "Research on the virtual machine placement algorithm in cloud computing based on stochastic programming", Computer Engineering and Science, vol. 34, no. 5, **(2012)**, pp. 95-100.

Authors

Long Zhang

He received the B.S. degree from Jinling Institute of Technology in 2007. Now he is studying for the master degree at Nanjing University of Aeronautics & Astronautics. His research interest is Cloud and Distributed Computing.

Yi Zhuang

She was born in 1957. She is a professor and Ph. D. supervisor of Nanjing University of Aeronautics & Astronautics. Her research interests include Distributed Computing and Information Security.

Wei Zhu

He received the B.S. degree from Northeastern University in 1996 and his M.S. degree in Computer Applications Technology from Huazhong University of Science & Technology in 2004. Currently, He is the research professor of Jiangsu Automation Research Institute, Lianyungang, China. His research interests include distribute systems, embedded systems, cloud computing and real-time control systems.

