A Novel Resource Allocation Strategy of Virtual Cluster Based on QOS

Feifei Liu
Engineering College of Armed Police Force of China, Xi'an, China

Abstract—**How to effectively allocate computer resource to satisfy the users' different QoS requirement is one of most important issues of building virtual high performance computing environment. To address this problem, this paper presents a novel resource allocate strategy in which the slowdown degree of users' job plays the key role of resource allocation. To implement the strategy, this paper also presents a Mixed Bin Packing Algorithms which is the combination of traditional Bin-Packing Algorithm and some general Algorithms. Experimental results show that the Mixed Bin Packing Algorithm is more effective than the previous algorithms, especially in a large scale application background.**

Keywords—Resource allocation, Virtual cluster, Quality of Service (QoS)

I. INTRODUCTION

Nowadays, high performance computing environment plays an increasingly important role in science computing. With the growing demand, more users of HPC put forward the customized needs of computing environment. At the same time, by virtualization technology, building its own high performance computing environment for user is one of efficient ways to satisfy the diversity of demand, in the context of limited physical resources[1-3].

In general, resources assigned to users for building the virtual environment are relatively fixed, and users will be exclusive employ this part of resources. When the number of user and the users' job number are changed, the system can not reallocate resources belonging to different users timely and effectively. Therefore, this resource allocation strategy is fairly easy to cause waste of resources and the effective utilization of resources in the whole system is difficult to guarantee.

At present, the problem of wasting resources can be partially solved through some monitoring technologies [4-6] which can achieve real-time monitoring of resource usage of virtual machine. But how to coordinate the struggle for resources between users and improve the resource utilization, especially when users put forward some specific requirements of quality of service, are still the key problems of resource allocation which are expected to solve[9, 10].

Generally speaking, the quality of service (QoS)[7] include a variety of constraint which may involve time, cost , reliability, credibility, safety, etc[8]. Among them, the completing time and cost of jobs are usually the two important aspects. At present there are some related work is to study the problem how to realize resources allocation upon the virtual cluster based on specific service quality constraint. Because of usually based on the traditional QoS constrains, these existing strategies has obvious limitations and can not adapt to these changes accompanied with renting of virtual clusters.

Amril N. put forward two steps to built the virtual computing environment in [11]. System firstly selects the relation resource according the cost optimal strategy and then constructs the virtual organization dynamically to server the user. This method is easier to satisfy users in cost constraints, but does not guarantee that resources are efficiently utilized as a whole. In [12], users were be classified by different priorities and the resource provision[13] were provided to the high level users. The user will enjoy the high reliability at the same time paying more cost.

For the better solution of coordinate the resource using between the multiple users which has their own virtual machine independently, this paper presents a novel resource allocation strategy in which the slowdown degree[14] of user's job is regard as a new metric of QoS and plays the key role in the whole process of resources allocation. This new strategy allows the system to dynamically adjust the amount of resource belonging to the specific user. New resource price mechanism can effectively reduce the customer spending and maximize the benefit of the whole system.

The remainder of this paper is organized as follows. Section 2 gives a detail of description about the novel allocation strategy and the new metric of QoS. In Section 3, we outline detailed design of three algorithms to achieve the strategy. Section 4 presents a comprehensive set of experimental data that evaluate the performance of different algorithms. Conclusions and suggestions for future work appear in Section 5.

II. VIRTUAL RESOURCE ALLOCATION STRATEGY BASED ON SPECIFIC RANGE

When user rents a long-term virtual computer resource, the required average computing power can be acquired through a period of observation. Generally, Allocating fixed amount resources to users is relatively easy method to meet the users' computing requirement. We wish to optimize a metric that encompasses user centric notions of both performance and fairness. From the users' perspective, the more resources available, the faster the work will be. At the same time, it also can bring about greater cost. While the resources occupied reach a certain value, the improvements on the performance caused by the additional resources will be very limited. Conversely, if can complete the task within the arranged time limit, the use of less resources also means

Published by Atlantis Press, Paris, France.

lower overhead. On the system is concerned, it is also easier to improve overall system performance.

A. Problem definitions

From the description above, it is not difficult to find that computing resources available to users should meet the scope of constraints. It may make the following definition.

Definition 1: *Users Resource Requirements Agreement*. In a virtual computing environment, users can submit their resource requirements which should be a proper scope of floating. Correspondingly, system accepting the user requests also should guarantee the computation ability kept in these scopes.

This resource range can be expressed as $[(1-k_i)\alpha_i, \alpha_i]$. Among the expression, α_i represents the expected optimal computation ability of each user; let k_i be the ratio between the lowest computation ability and the most optimal computation ability. Obviously, each k_i is service quality agreement between user and system. Its value should be in (0, 1] interval. In some special environment, the ratio also can be zero, which means user allows job scheduling system suspend their jobs when it is difficult to provide the necessary resources.

Definition 2: *Distributed User Resource Restriction*. Under the distributed computing environment, the resource serving to the special user is usually a collection of multiple heterogeneous resources. From the Def. 1, the resources available to users should be subject to the following constraints:

$$\forall i \qquad \sum_{j=1}^{H} \alpha_{ij} \le \alpha_i, \sum_{j=1}^{H} \frac{\alpha_{ij}}{\alpha_i} \ge (1-k_i)$$

Among them, α_{ij} represents the resources which each independent host j allocates to user i. H is the amount of physical clusters (or mainframes). We let α_{ij} take rational values to enable fine-grain resource allocation.

Definition 3: *Slowdown Degree of User's job*. For each user *i*, $\sum_j \alpha_{ij} / \alpha_i$ is a reflection of the relative gap between its actual occupation of resources and the ideal state. It can also be considered as the actual degree of slowing of user's job.

It should be noted that, when the resource management system dynamically adjusts the computing resources occupied by the user, the ratio value also changes. But unlike k_i, this value has nothing to do with the user cost. The higher the ratio value is, the higher customer satisfaction is.

B. Resource Allocation and Accounting Strategy Under the Constrain of QoS

To better balance the resource contention among users and achieve maximum effective using of system resources, it might be proposed that take the k_i mentioned in the previous section as a new contract of QoS between the user and system. Based on this agreement, we put forward the following differentiated resource accounting strategy:

1) Differentiated Resource Accounting

After receiving the relevant request from the user, system should set the range of resource requirements according to the K value setting by user. When the resource requirement of users have a clear upper and lower limitation, the part below the lower limit should be gave higher priority to ensuring and account with the higher tariff standards; on the other hand, the part which are higher than the lower limit, can be dynamically adjusted according to the system load conditions and charged with a relatively low price.

At this time, the user's total cost will not only determined by its mission total requirements but also closely related to the k_i value setting by users themselves. Lower the value are, the more computing tasks may be completed as soon as possible, but the greater of cost to pay.

2) Resource Allocation Strategy Related to K Value

Based on this differentiation resource accounting strategy, we propose the following resource allocation strategy:

a) priority to meet user basis needs. The basis resource needs are calculated by the k_i (definited by user i). Standing the perspective of resource providers, this allows to maximize the system revenue. In the peak load, it can ensure more users to enjoy system services at the same time.

b) *When available resources arise, the user which defines the more high value of k_i have the high priority.*

When the system is busy, these users might be handed over more resources. This strategy also allows the user sharing the system resources to minimize the average slowdown degree.

c) When a new request to build a virtual cluster arrived, the resource are released by those user which have the most distance between the value of k_i and actual slowdown degree.

Based on the above three rules, even if the user-defined k_i value is high, the system still will initiative allocate more resources to user's virtual machine when the system load is low. So the user's job run-time will not necessarily be obviously prolonged. At the same time, these resources are relatively inexpensive.

C. Formalized Problem Statement

Comprehensive above problem description, resource sharing allocation problem of virtual cluster building upon multiple physical clusters environment can be can be formulated as a MILP, that is, a Linear Program (LP) with both rational and integer variables.. We can write the constrains as follows, with $i = 1,..,J$ and $j = 1..H$. Its expression is as follows:

$$\forall j \qquad \sum_{i=1}^{J} \alpha_{ij} \le C_j \qquad (1)$$

$$\forall i \qquad \sum_{j=1}^{H} \alpha_{ij} \le \alpha_i \qquad (2)$$

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

$$\forall i \qquad \sum_{j=1}^{H} \left\lceil \frac{\alpha_{ij}}{\alpha_i} \right\rceil \le 1 \qquad (4)$$

$$\forall i \qquad \sum_{j=1}^{H} \frac{\alpha_{ij}}{\alpha_i} \ge Y \qquad (5)$$

Among them, C_j represents all of the resources provided by independent host j, $j = 1..H$; e_{ij} represents the possibility which the request of user i is dispatched to host j;

If each user's request is only assigned to no more than one physical cluster, $e_{ij} \in \{0,1\}$. Equation (4) represents this constraints and it can be cancel when multiple virtual machines servicing on the same virtual cluster are allowed to be assign to different physical host.

The objective of optimization is to maximize Y. To maximize the minimum yield, which meaning that all users in the system is to minimize the degree of slowdown. Solving the problem and obtaining the optimal solution or the feasible solutions, we can adjust the configuration of virtual machine instances to implement the solution of resource allocation problem.

III. RESOURCE ALLOCATION ALGORITHM FOR VIRTUAL CLUSTER

In general, solving a MILP requires exponential time. When the number of users and physical clusters are small, solution of the resource allocation problem can be calculated by means of special tools for the optimal solution, such as the Gnu Linear Programming Toolkit (GLPK), Lingo and so on. When problem scale increases to a certain degree, the exact solution is often difficultly acquired in a limited time.

A. General algorithms

1) Gradually Relaxed Algorithm

In order to solve the problem, a more direct approach is to firstly pre-allocate the resources based on the user-defined minimum requirements and then gradually allocate remaining part to each task which resources is insufficient. Intuitive sense, this method can guarantee as much as possible to response to user requests. Gradually relaxed algorithm is based on this thought to solve the resources distribution.

In the pre-allocation phase, for each user request it should first select the physical cluster which burden is relative low in order to guarantee that the remaining resources of every physical clusters is roughly equivalent. In the second stage, priority should be given these user requests which still exist a relatively big distance to reach the upper limit of resource requirement. To prior allocate resources to the user which possess the largest Slowdown Degree can effectively improve the overall service satisfaction.

2) Greedy Algorithm

In contrast with the Gradually Relaxed Algorithm, the basic idea of Greedy Algorithm are to try to meet the maximum demand of each user in the pre-allocation phase and only when new user's tasks request cannot be satisfied, the system then withdraws part of the allocated resources for secondary distribution. In order to more equal treatment of all kinds of users, priority should be given user request with greater resources demand in the pre-allocation phase.

There is a more detailed introduction about this approach in [14].

B. Mixed Bin Packing Algorithm

The algorithms mentioned in Section III.A explore the solution space directly from specific user needs. Because of ignoring the correlation between the resource requirements of multiple users, both methods have their obvious shortcomings. To solve this problem, this paper proposes the improved algorithm based on the knapsack algorithm.

1）Relative Scarcity of Resources

From the system perspective, to better characterize the degree of scarcity of resources, we gives the following definition:

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

For request of any user i, according to the resource allocation strategy proposed in section II.B it can be easier to get the following inference.

Inference 1: If $(1-k_i) > \tau$, then $\sum_{j=1}^{H} \alpha_{ij} = (1-k_i)\alpha_i$.

Inference 2: If $(1-k_i) \le \tau$, then $\sum_{j=1}^{H} \alpha_{ij} \ge (1-k_i)\alpha_i$ and $\sum_{j=1}^{H} \alpha_{ij} \le \tau * \alpha_i$.

2）Algorithm Improvement

According to Inference 1 and 2, it is not difficult to discover that solving the problem can be divided into two parts: the solution for the fixed demand part and the solution for variable demand. The MBPA algorithm is illustrated in Fig 1.

Among the process, the solution of resource allocation problem in step 4 is similar with the traditional bin packing problem[15]. At present, many mature approaches to solve such problems have already been expressed. In step 7, the judgement about the scale of problem should be adjusted according to the results for the solution of practical problem.

```
1. Initiate $C_j, \alpha_i, k_i$;
2. Calculate $\tau$;
3. Sort user's request queue by $k_i$ in ascending order and put them into two groups, i.e. queue A ($k_i \le 1-\tau$) and queue B ($k_i > 1-\tau$);
4. Calculate the lowest resource requirement of each user's request in queue A and according to this value allocate user's requests in queue A to different physical cluster;
5. Calculate the remaining resource of each physical cluster;
6. Construct a new problem B which includes the remaining resource of each physical cluster and all of user's requests in queue B.
7. If the scale of problem B is enough small
       Use the Gradually Relaxed or Greedy algorithm to solve problem B;
   Else
       Repeat the procedure from step 1 to 6 to solve problem B;
8. Merge the result obtained in step 6 and 7 and get the final allocation scheme.
```

It is obvious that both the number of tasks in queue A and B depend on k_i setting by each user. If $\tau > 1$, the original problem will directly degenerate into the knapsack problem. Otherwise, if all of k_i are smaller than $(1-\tau)$, the system will not be able to find the resource allocation scheme to meet the needs of all users. To complete all tasks, some of request must be hang firstly.

C. *Standardizing Technology*

The algorithms proposed above can be obtained a feasible solution of the resources allocation problem within a limited time. However, if the results need to be implemented by changing the configuration of each virtual machine instances, no doubt, it will increase the complexity of resource management and job scheduling.

To better address this issue, we can customize a standardized virtual machine instance in the range of resources across multiple physical clusters. This method will ensure that each virtual machine instance is broadly similar at the resources amount and its computing power. By dynamically adaptive management techniques, virtual cluster can automatically adapt to the changes of number of the virtual machine instances. It also reduces the complexity of resource management. Another advantage of this approach is that the original problem can degenerate into integer programming problem and solving the problem can be further simplified.

IV. Experimental Test

To facilitate data analysis and comparison of different algorithms, this paper experiments are based on standardized virtual machine instances. The main body of resource allocation is simplified to the standardized virtual machine instance. It is also a measure standard of computing ability for physical cluster. As the experimental initial conditions, it needs to clear the number of physical clusters and the number of the standard virtual machine instances which each physical cluster can accommodate. To maintain consistency in the allocation of resources, the user's demand also required to express the number of instances the virtual machine.

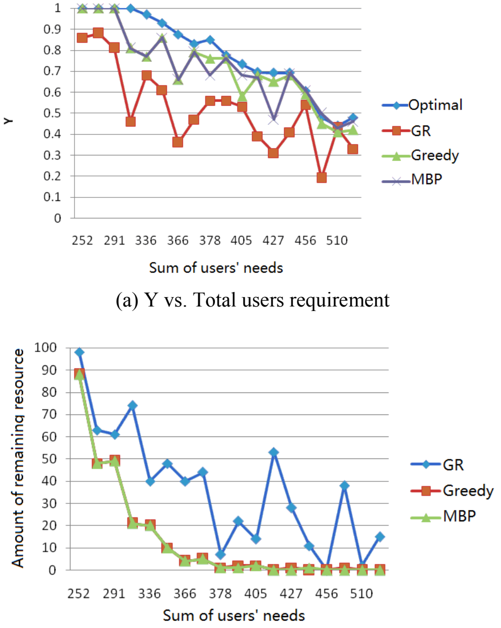
During the process of experiments, we gradually increase the system load through increasing the number of users. In general, when the sum of users' needs of resource is far less than the amount of resources provided by multiple physical clusters, most of user's maximum requests can be satisfied easily. So, Experiments will only focus on the situation in which the sum of maximum demand for all user is close or exceed to the amount of resources. We compare the effects of different algorithms mainly by observing the following two data: 1) Y value which algorithm can reach; 2) after completion of distribution, the amount of remaining resources.

In order to fully test the applicability of the algorithm, two groups of physical cluster experiments environment are given. First group with smaller size in which contains five different physical clusters can accommodate no more than 320 standard virtual machines. The capacity of each individual physical cluster is not less than 50 standard virtual machine instances. Second group with larger size is composed by eight physical clusters which can support not less than 90 standard virtual machines. The total computing power reaches to support 1000 standard virtual machine.

A. *Small Problems*

Fig. 2(a) shows changes the value of Y when the user total demand increases. Value of Y represents the minimum value of slowdown degree among all users' request. Solution of problem should guarantee to get the largest possible value of Y. When value of Y reach 1, it indicates that every user get enough computing resource as their highest demand. Conversely, the lower value of Y means that users generally had less access to resources. Under the same conditions, different algorithms will be different Y values. The higher the value of Y, the higher the overall satisfaction of users. Compared data in Fig. 2(a), it can be seen that the value of Y decrease significantly when the total number of users and user needs continue to increase.

For small-scale problems, the optimal solution obtained through linear programming tools is an important reference in comparing the data obtained by different algorithms. The data of in Fig. 2(a) indicate that the Y value of the Greedy Algorithm or Mixed Bin Packing Algorithms is usually 20%~40% increase than the value of the Gradually Relaxed Algorithm. Fig. 2(b) shows the change of amount of total remaining resource accompanying with the increase of users' needs. The higher remaining resource, the lower overall resource utilization. According to Fig. 2(b), the Gradually Relaxed Algorithm also has a wide gap in customer satisfaction than the other two algorithms.

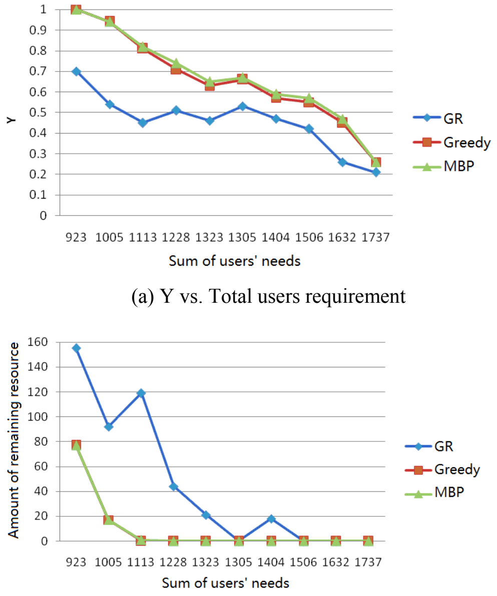


(a) Y vs. Total users requirement

(b) Remaining resources vs. Total users requirement

Figure 2 Y value and Remaining resources in small problems

B. Large Problems



(a) Y vs. Total users requirement

(b) Remaining resources vs. Total users requirement

Figure 3 Y value and Remaining resources in small problems

The data in Fig. 3 shows that the performances of three algorithms are more stable for large-scale problems. The performance of Gradually Relaxed Algorithm is significantly lower than the other two methods. Both the Mixed Bin Packing Algorithm and the Greedy Algorithm can guarantee to maximize the utilization of resources, but the former has more value in the Y value.

C. Result Analysis

Based on the above experimental data, it is easy to see that the Mixed Bin Packing Algorithm and the Greedy Algorithm have obvious advantages than the Gradually Relaxed Algorithm in resource utilization. Compared with the Greedy Algorithm, the Mixed Bin Packing Algorithm is less stable in the small scale of the problem, but in the larger problem its relative performance is better, that is, the former is more suitable for solving small-scale data and the latter is more suitable for solving large scale problems.

In the current discussion, we retain the limitation which the virtual machine instances of same virtual cluster locate in the same physical cluster. In practice, this approach will ensures that the user virtual environment will not affect by the geographic dispersion of entire virtual cluster. But the factor also restricts the improvement of performance further. Especially when the overall size of resources is small, this factor will lead to greater fluctuations in algorithm performance for different instances.

V. CONCLUSION AND FUTURE WORK

In order to solve resource allocation problem under the conditions of virtual cluster, this paper proposes a novel mechanism which takes the slowdown degree of user's job as the core indicators of QoS. Under different user constraints of QoS, the mechanism ensures optimal performance of system and users the lowest cost. Based on the novel mechanism，a novel algorithm Mixed Bin Packing Algorithm possesses is presented. The Mixed Bin Packing Algorithm possesses was verified in different scale of the problem. Experimental data show that the Mixed Bin Packing Algorithm possesses superiority in performance and the approximate optimal features.

For reducing implementation complexity, current study retains the constraint which the virtual machine instances of same virtual cluster locate in the same physical cluster. However, when the computing ability of single physical cluster is limited, relaxing this constraints will also improve resource utilization and user satisfaction. How to expand existing algorithms and progressively realize the resource allocation under the dynamic load conditions will be the focus of future research.

ACKNOWLEDGMENT

This work is supported by 863 Project of China (Grant No. 2009AA01A135 and 2008AA01A202), National Natural Science Foundation of China (Grant No. 60773118), the Fundamental Research Funds for the Central Universities (Grant No.08142007).

REFERENCES

[1] Foster I, Freeman T, Keahey K, et al. Virtual clusters for Grid communities. In: Sixth 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: International Symposium on High-Performance Computer Architecture, Scottsdale, AZ, 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 and 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 Association, 2005, pp. 24-24.

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

[8] Yeo C S and 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 and 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 and 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 10th IEEE International Conference on High Performance Computing and Communications, Dalian,China, 2009. 29-35

[12] Yang C, Tianyu W and Jianxin L. An Efficient Resource Management System for On-Line Virtual Cluster Provision. In: Proceedings of IEEE International Conference on Cloud Computing, CLOUD '09., Bangalore, 2009. 72-79

[13] Elmroth E and 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, IEEE Computer Society, 2009. 260-267

[15] Leinberger W, Karypis G and 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, 1999. 404-412