Proposed Joint Multiple Resource Allocation Method for Cloud Computing Services with Heterogeneous QoS

Yuuki Awano

Dept. of Computer and Information Science
Seikei University
Musashino, Tokyo, Japan
us092008@cc.seikei.ac.jp

Shin-ichi Kuribayashi

Dept. of Computer and Information Science
Seikei University
Musashino, Tokyo, Japan
kuribayashi@st.seikei.ac.jp

Abstract - ***This paper proposes to enhance the proposed joint multiple resource allocation method so that it can handle multiple heterogeneous resource-attributes. The basic idea is to identify the key resource-attribute first which has the most impact on resource allocation and to select the resources which provide the lowest Quality of Service for the key resource-attribute as it satisfies required Quality of Service. It is demonstrated by simulation evaluations that the enhanced method can reduce the total amount of resources up to 30%, compared with the conventional methods. The enhanced method could be also effective to the resource allocation in a hybrid-cloud in which either a private-cloud or a public-cloud is selected depending on the required security level.***

Keywords - *cloud computing; heterogeneous QoS; joint multiple resource allocation; hybrid cloud.*

1. Introduction

Cloud computing services are allow the user to rent, only at the time when needed, only a desired amount of computing resources (ex. processing ability, storage capacity) out of a huge mass of distributed computing resources without worrying about the locations or internal structures of these resources [1]-[5]. The popularity of cloud computing owes to the increase in the network speed, and to the fact that virtualization and grid computing technologies have become commercially available. It is anticipated that enterprises will accelerate their migration from building and owning their own systems to renting cloud computing services, because cloud computing services are easy to use and can reduce both business costs and environmental loads.

As cloud computing services rapidly expand their customer base, it has become important to provide them economically. To do so, it is essential to optimize resource allocation under the assumption that the required amount of resource can be taken from a common resource pool and rented out to the user on an hourly basis. In addition, to be able to provide processing ability and storage capacity, it is necessary to allocate simultaneously a network bandwidth to access them and the necessary power capacity. Therefore, it is necessary to allocate multiple types of resources (such as processing ability, bandwidth, and storage capacity) simultaneously in a coordinated manner, instead of allocate -ing each type of resource independently [6]-[8].

Moreover, it is necessary to consider not only the required resource size but also resource-attributes in actual resource allocation. Resource-attributes of bandwidth, for example, are network delay time, packet loss probability, etc. If it is required to respond quickly, bandwidth with a short network delay time should be selected from a group of bandwidths. Computation time is one of resource-attributes of processing ability. References [6] and [7] consider a model in which there are multiple data centers with processing ability and bandwidth to access them, and proposed the joint multiple resource allocation method (referred to as "**Method 3**").

The basic idea of Method 3 is to select a bandwidth with the longest network delay time from a group of bandwidths that satisfy the condition on service time. It is for maximizing the possibility to accept requests later, which need a short network delay time. It was demonstrated by simulation evaluations that Method 3 can handle more requests than the case where network delay time is not taken into account, and thus can reduce the required amount of resources by up to 20% [6],[7].

Method 3 takes into account only a single resource-attribute of network bandwidth (namely, network delay time). However, it is usually necessary to consider multiple heterogeneous resource-attributes in a real cloud computing environment. It is proposed to enhance the proposed method, Method 3, to handle multiple heterogeneous resource-attributes. The enhanced-Method 3 could be also effective to the efficient resource allocation in hybrid clouds [9]. In a hybrid cloud, transactions that require a critical security are executed using private clouds only and other transactions that require a normal security may be executed using more economical public clouds. For the preliminary evaluation, this paper assumes two types of resources (processing ability and bandwidth), loss-system based services and the static resource allocation.

The rest of this paper is organized as follows. Section 2 explains related works. Section 3 provides the resource allocation model for cloud computing environments. Section 4 proposes to enhance the proposed joint multiple resource allocation method, Method 3, to be able to handle multiple heterogeneous resource-attributes. Section 5 describes simulation evaluations which confirm the effectiveness of the enhanced-Method 3 (referred to as "**Method 3E**"). Finally, Section 6 gives the conclusions.

2. Related work

Resource allocation for clouds has been studied very extensively in References [10]-[19]. References [14],[15] have proposed automatic or autonomous resource management in cloud computing. Reference [10] has proposed the heuristic algorithm for optimal allocation of cloud resources. Reference [16] has presented the system architecture to allocate resources assuming heterogeneous hardware and resource demands. References [11] and [12]

 ISBN: 978-1-61208-216-5

have proposed market-oriented allocation of resources including auction method. Reference [13] has proposed to use game-theory to solve the problem of resource allocation. Energy aware resource allocation methods for clouds have been proposed [18]-[20].

However, most of conventional studies on resource allocation in a cloud computing environments are treating each resource-type individually. To the best our knowledge, the cloud resource allocation has not been fully studied which assumes that multiple resources are allocated simultaneously to each service request and there are multiple heterogeneous resource-attributes for each resource-type.

3. Resource allocation model for cloud comput -ing environments

3.1 Resource allocation model

The resource allocation model for a cloud computing environment is such that multiple resources with heterogeneous resource-attributes taken from a common resource pool are allocated simultaneously to each request for a certain period. For the preliminary evaluation, this Section considers two resource-types: processing ability and bandwidth. It is assumed that the physical facilities for providing cloud computing services are distributed over multiple data centers, in order to make it easy to increase the number of the facilities when demand increases, to allow load balancing, and to enhance reliability.

The cloud resource allocation model that incorporates these assumptions is illustrated in Figure 1. Each center has servers which provide processing ability and network devices which provide the bandwidth to access the servers. The maximum size of processing ability and bandwidth at center j (j=1,2,..,k) is assumed to be C_{maxj} and N_{maxj} respectively. The different resource-attributes of processing ability and network bandwidth could be provided by each center.

When a service request is generated, one optimal center is selected from among k centers, and the processing ability and bandwidth in that center are allocated simultaneously to the request for a certain period. If no center has sufficient resources for a new request, the request is rejected. These

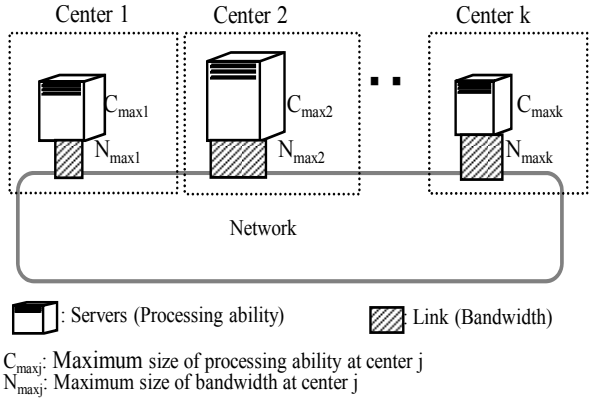


Figure 1. Resource allocation model for cloud computing environments

are the same as those in References [6]-[8].

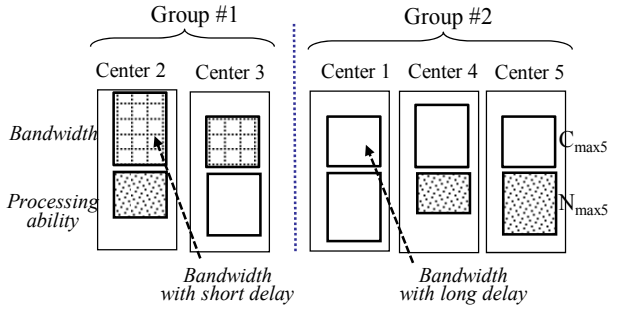
3.2 Guidelines of joint multiple resource allocation assuming multiple heterogeneous resource-attributes

In general, a cloud computing environment includes multiple resource-types and multiple resource-attributes for each resource-type. For example, resource-attributes of bandwidth are network delay time, packet loss probability, required electric power capacity, etc. If a request requires quick-response, it is needed to select one with a short network delay from a group of bandwidths. On the contrary, if a request requires a less power consumption, it is needed to select a bandwidth whose power consumption is small. Resource-attributes of processing ability are computation time, memory size, required electric power capacity, etc. In a hybrid cloud, resource-attributes may additionally include the levels of security (critical or normal) and reliability.

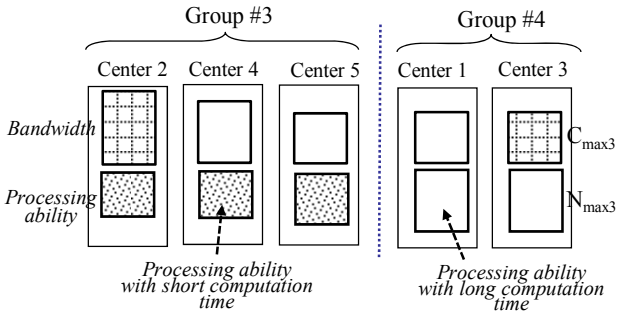
The center selection algorithm with Method 3 proposed in References [6] and [7] is explained with Figure 2. Figure 2 is just an example. There are five centers in different locations, and that each center has two resource-types: bandwidth and processing ability. In Figure 2(1), centers are divided to multiple groups according a resource-attribute (network delay time) of bandwidth. That is, centers in Group #1 can provide bandwidth with short delay and centers in Group #2 provide bandwidth with long delay. If a request's requirement on response is not so stringent, Method 3 first tries to select a center from Group #2, and only when there is no center with appropriate resources available in this group, it selects a center from Group #1. This approach makes it possible to meet more future requests later, which need a short delay. We next consider center groups taking a resource-attribute (computation time) of processing ability into consideration, as shown in Figure 2(2). If a request has no stringent requirement on computation time, Method 3 first attempts to select a center from Group #4, and only when there is no center with appropriate resources available in this group, it selects a center from Group #3.

In this way, the priority with which a center group is selected differs between Figure 2(1) and Figure 2(2). If a request with no strong requirement is allocated to a center 4 or center 5 taking only one resource-type into consideration, for example, then fewer resources are likely to be available later when requests with a stringent requirement on processing ability are generated. Therefore, it is necessary to take both multiple resource-types and multiple resource-attributes into consideration simultaneously in selecting a center. Moreover, it would be necessary to consider a new center group if requests with a stringent requirement on both bandwidth and processing ability are generated. Even if center groups are created taking all the resource-types and resource-attributes into consideration, the combinations of different requirements can be too numerous to be manageable, and it would not be easy to develop a guideline as to the sequence of priority in which center groups are to be selected.

 ISBN: 978-1-61208-216-5



(1) Grouping with resource-attribute of bandwidth



(2) Grouping with resource-attribute of processing ability

Figure 2. Example of resource allocation assuming heterogeneous resource-attributes

Therefore, the simplified algorithm adopted by the authors in References [6] and [7] would be also applicable here.

The above guidelines could also be effective to the resource allocation in a hybrid-cloud. In hybrid-cloud, either a private or a public cloud will be selected depending on the required levels of security or reliability, as shown in Figure 3. Requests that require a normal security should be allocated to the public cloud first, and then to the private cloud so that the resources in the private cloud can be kept available for future requests that require a critical security. It turns out that security level or reliability level need to be considered as one of resource-attributes.

4. Enhanced joint multiple resource allocation supporting multiple resource-attributes

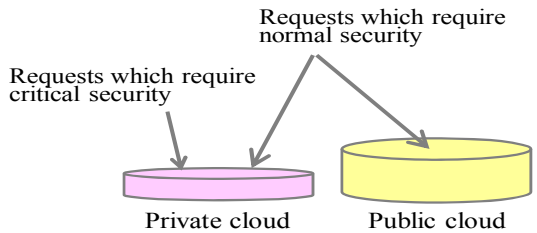


Figure 3. Services with both private and public cloud

4.1 Principle

As discussed in Section 3.2, it is difficult to take multiple resource-types and multiple resource-attributes for each resource-type into consideration simultaneously. It is proposed to apply the same principle adopted in References [6] and [7]. That is, it is proposed to allocate resources focusing on the most important resource-attribute (hereafter referred to as the "**key resource-attribute**"). The key resource-attribute is decided by the system (not by the user), and can be different for each request.

The resource allocation algorithm of enhanced Method 3 (Method 3E) is explained in the next Section 4.2, which adopts the concept of key resource-attribute above.

4.2 Resource allocation algorithm of Method 3E

4.2.1 Identification of key resource-attribute

An attribute with the lowest **relative amount of resource** is selected as key resource-attribute from among multiple resource-attributes for all resource-types. The relative amount of resource, M_g, for resource-attribute g is calculated by

$$M_g = d_{2g}/d_{1g} \quad (1)$$

where d_{1g} is the sum of resources which offer resource-attribute g and all the resources which offer higher quality of service (QoS) than resource-attribute g. d_{2g} is the expected amount of resources with resource-attribute g required by all requests.

For example, if there are bandwidths with network delay time of 50ms and those with network delay time of 200ms, d_{1g} for network delay time of 200ms includes not only the amount of bandwidths with network delay time of 200ms but also the amount of bandwidths with network delay time of 50ms.

It is also proposed that resource-attribute g is not selected as key resource-attribute when the ratio of the number of requests requiring resource-attribute g to the total number of requests is lower than a certain value (e.g., 10%).

4.2.2 Identification of a center group

Here we focus on the resource-type associated with the key resource-attribute, and classify center groups into three categories: Center Group X, which contains resources that provide lower QoS than that provided by the key resource-attribute, Center Group Y, which contains resources that provide QoS equal to that provided by the key resource-attribute, and Center Group Z, which contains resources that provide higher QoS than that provided by the key resource-attribute. In some cases, Center Group X or Center Group Z may not exist.

4.2.3 Selection of a center

- A center that can provide multiple resources required by the request is selected. If there is no center that can satisfy the requirements, the request is rejected.
- If there are several selectable centers in the center group, one is selected either at random or sequentially.
- A center is selected as follows depending on the QoS required by the request.

i) If the request requires lower QoS than that associated with the key resource-attribute, it is tried to select a center in Center Group X. If there is no selectable center in the group,

a selectable center in Center Group Y or in Center Group Z is selected in this order.

ii) If the request requires the QoS associated with the key resource-attribute, a center is selected in Center Group Y. If there is no selectable center there, a center in Center Group Z is selected.

iii) If the request requires higher QoS than that associated with the key resource-attribute, it is tried to select a center in Center Group Z.

- The multiple resources with required resource-attribute in the selected center are allocated to the request simultaneously.
- When the service time to the request has expired, all the resources allocated in Section 4.2.4 are released.

5. Simulation evaluation

5.1 Evaluation model

1) Method 3E proposed in Section 4.2 is evaluated using a (self-made) simulator written in the C language.

2) For the preliminary evaluation, we consider only two resource-types: processing ability and bandwidth. 'Computation time' is used as a resource-attribute of processing ability and 'network delay time' as that of bandwidth here.

3) Figure 1 with k=3 is assumed as the resource allocation model. That is, there are three centers, Centers 1, 2 and 3, which provide resources with different resource-attributes as follows:

<Attribute: Computation time>
- long for Centers 1 and 3
- short (referred to as 'high_1') for Center 2

< Attribute: Network delay time>
- long for Centers 1 and 2
- short (referred to as 'high_2') for Center 3

Any attribute other than high_1 or high_2 is referred to as 'normal' in this Section.

4) Three types of requests are considered here:

<Type_1> Requests that can be satisfied with attribute 'normal' for both computation time and network delay time. Selectable resources exist in any center. The probability at which type_1 request occurs is designated as q_1.

<Type_2> Requests that can be satisfied only with attribute 'high_1' for computation time, but can be satisfied with attribute 'normal' for network delay time. Selectable resources exist only in Center 2. The probability at which type_2 request occurs is designated as q_2.

<Type_3> Requests that can be satisfied only with attribute 'high_2' for network delay time, but can be satisfied with attribute 'normal' for computation time. Selectable resources exist only in Center 3. The probability at which type_3 request occurs is designated as q_3 ($q_1+q_2+q_3=1$).

5) When a new request is generated, one appropriate center is selected according to the resource allocation algorithm (Method 3E) in Section 4.2 and then both processing ability and bandwidth from that center is allocated to the request simultaneously. For the purpose of comparison, the proposed method, Method 3, and Round Robin method (referred to as "RR Method") in which a center is selected in sequence, are also evaluated in the simulation. Method 3, which does not have the concept of key resource-attribute, considers only network delay time here.

6) The size of required processing ability and bandwidth by each request is assumed to follow a Gaussian distribution (dispersion is 5). Let C and N be the averages of the distributions of processing ability and bandwidth respectively.

7) The intervals between requests follow an exponential distribution with the average, r. The length of resource holding time, H, is constant. All allocated resources are released simultaneously after the resource holding time expires.

8) The pattern in which requests occur is a repetition of $\{C=a_1, N=b_1; C=a_2, N=b_2; \ldots; C=a_w, N=b_w\}$, where w is the number of requests that occur within one cycle of repetition, a_u (u=1~w) is the size of C of the u-th request, and b_u (u=1~w) is the size of N of the u-th request.

5. 2 Simulation results and evaluation

The simulation results are shown in Figures 4, 5 and 6. The horizontal axis shows the probability q_1 at which type_1 request occurs. The value of q_2 and q_3 is set to $(1- q_1)/2$ respectively. The vertical axis of Figures 4 and 5 shows the average request loss probability. The vertical axis of Figure 6 shows the ratio of required amount of resources by Method 3E and those by RR method, on the condition of keeping the same average request loss probability. Figure 4(1) shows evaluation results for the case where the request generation pattern is uniform. Figure 4(2) shows the case where it is uneven (i.e., rise and fall in anti-phase). Figure 5 is intended to evaluate the impact of the unevenness of the total amount of resources between centers. While the total amount of resources in each center is the same in Figure 4, the total resource amount of Center 3 is twice that of Center 1 or Center 2 in Figure 5. Figure 5(1) and 5(2) show the total average request loss probability and the request loss probability for each request-type respectively. The parenthesis following Method 3 or Method 3E in Figure 5 indicates the request-type.

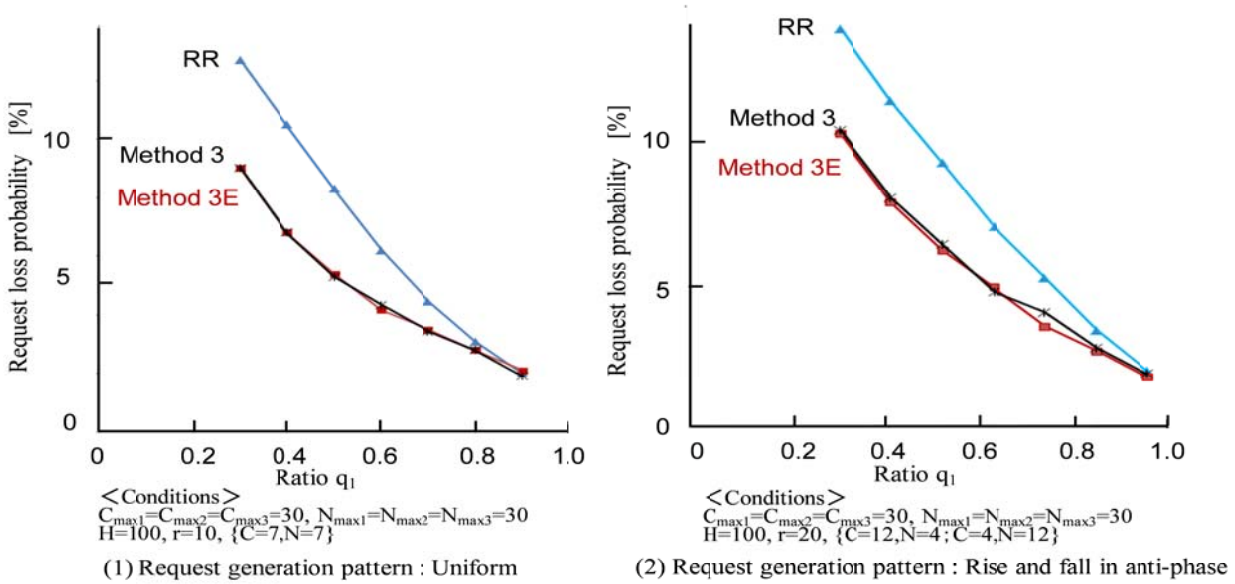
The following points are clear from these Figures:

i) Except for the area near $q_1=1.0$ (i.e., the area where almost all requests are type_1), the request loss probabilities of Method 3E and Method 3 are smaller than that of the RR method by up to 30%. This tendency is effective regardless of the request generation pattern.

<Reason> Even when requests are type_1, RR method tends to select Center 2 or Center 3 more often compared with Method 3E or Method 3. The reason why there is not much difference in results between Methods 3E and 3 is that type_1 requests use almost all resources in Centers 1, 2 and 3 when q_1 comes close to 1.0.

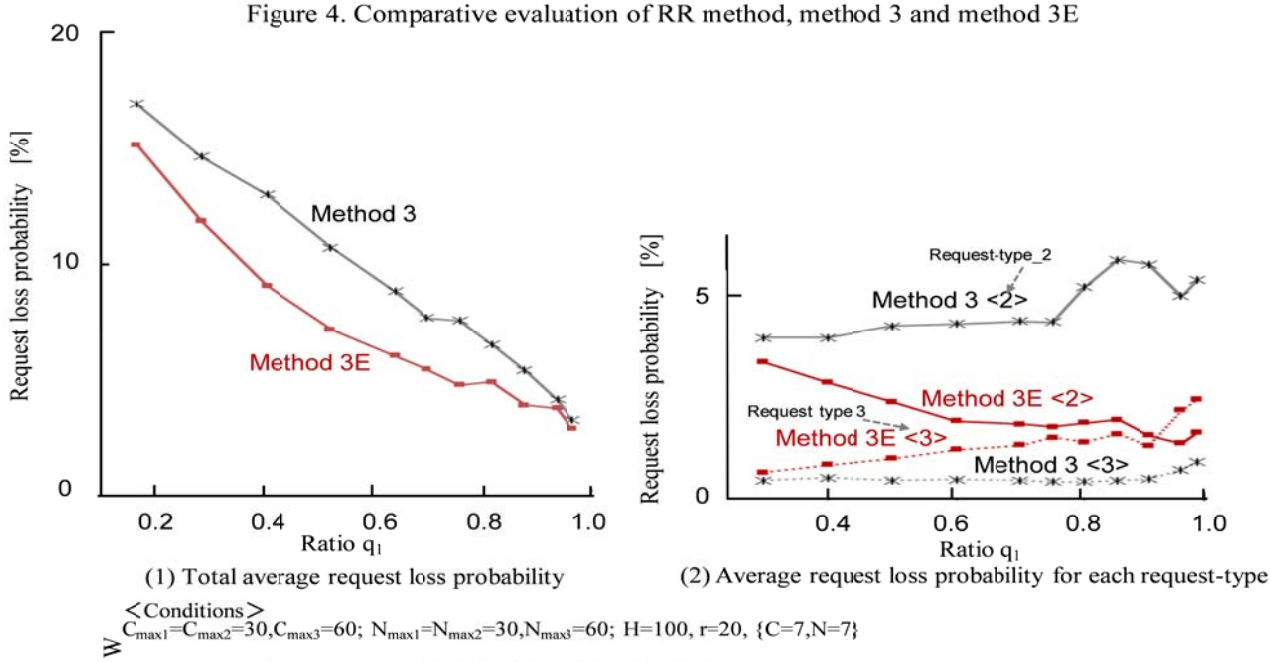
ii) Except for the area near $q_1=1.0$, the request loss probability of Method 3E is smaller than that of Method 3 when the total resource amount used by each request-type is

 ISBN: 978-1-61208-216-5



(1) Request generation pattern : Uniform (2) Request generation pattern : Rise and fall in anti-phase

Figure 4. Comparative evaluation of RR method, method 3 and method 3E



(1) Total average request loss probability (2) Average request loss probability for each request-type

Figure 5. Comparative evaluation of method 3E and method 3

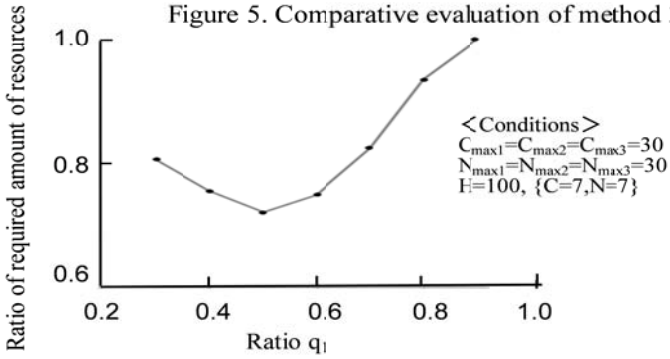


Figure 6. Ratio of required amount of resources

different. The differences in the request loss probability between different request-types can also be made smaller by Method 3E.

< Reason> Method 3, which does not have the concept of key resource-attribute and takes attribute high_2 into consideration, goes on to select Center 2 for type_1 requests if the appropriate resources are not available in Center 1. Therefore, the amount of resources available in Center 2 decreases rather than that in Center 3. As a result, the request loss probability of type_2 requests increases, which require

resources with attribute high_1 (key resource-attribute here).

In Method 3E, the key resource attribute is set to attribute high_1, and when type_1 requests cannot use Center 1, they attempt to select Center 3, which has more resources. As a result, more resources are kept available in Center 2 than in the case of using Method 3, and it is possible to reduce the request loss probability of type_2 requests. As the value of q_1 becomes small, the number of type_2 requests to handle increases and the request loss probability of type_2 will increase also by Method 3E.

iii) The total amount of resources required for keeping the same request loss probability could be smaller with Method 3E than with RR method by up to 30%.

6. Conclusion and Future Work

This paper has enhanced the proposed joint multiple resource allocation method (Method 3) so that it can handle multiple heterogeneous resource-attributes. The basic idea of the enhanced Method 3 (**Method 3E**) is to identify the key resource-attribute first which has the most impact on resource allocation and to select the resources which provide the lowest QoS for the key resource-attribute as it satisfies required QoS, so that future requests with more stringent requirement can still find available resources.

It has been demonstrated by simulation evaluations that Method 3E can reduce the total amount of resources up to 30%, compared with the conventional methods. Method 3E could be also effective to the resource allocation in a hybrid-cloud in which either a private-cloud or a public-cloud is used depending on the required level of security.

For the preliminary evaluation, we have limited the numbers of request types, centers, resource-types, and resource-attributes to small numbers in our simulation evaluation. We will make an evaluation with larger numbers of these to confirm the effectiveness of the proposed method and to identify the conditions in which the proposed method is effective. Moreover, the value of resource-attribute related to bandwidth may change with the location where a request occurs. For example, the procedure to regulate the access from a distant location temporarily when the amount of available resources are less than the threshold value is required to be studied.

Acknowledgement

This work was supported in part by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C) (21500041).

References

[1] G.Reese: “Cloud Application Architecture”, O’Reilly& Associates, Inc., Apr. 2009.
[2] J.W.Rittinghouse and J.F.Ransone: “Cloud Computing: Impremention, Management, and Security”, CRC Press LLC, Aug. 2009.
[3] P.Mell and T.Grance, “Effectively and securely Using the Cloud Computing Paradigm”, NIST, Information Technology Lab., July 2009.
[4] P.Mell and T.Grance : “The NIST Definition of Cloud Computing” Version 15, 2009.
[5]Z.Hang, L.Cheng, and R.Boutaba, “Cloud compuing: state-of-the-art and research challenges”, J Internet Serv Apl, Jan. 2010.
[6] S.Kuribayashi, “Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments”, International Journal of Research and Reviews in Computer Science (IJRRCS), Vol. 2, No.1, Feb. 2011.
[7] S.Tsumura and S.Kuribayashi: “Simultaneous allocation of multiple resources for computer communications networks”, In Proceeding of 12th Asia-Pacific Conference on Communications (APCC2006), 2F-4, Aug. 2006.
[8] K.Mochizuki and S.Kuribayashi, “Evaluation of optimal resource allocation method for cloud computing environments with limited electric power capacity”, Proceeding of the 14-th International Conference on Network-Based Information Systems (NBiS-2011), Sep. 2011.
[9] H.Zhang, G.Jiang, K.Yoshihira, H.Chen, and A.Saxena, “Intelligent workload factoring for a hybrid cloud computing model”, Proceedings of the 2009 IEEE Congress on Services (Services’09), July 2009.
[10] B. Soumya, M. Indrajit, and P. Mahanti, “Cloud computing initiative using modified ant colony framework,” in In the World Academy of Science, Engineering and Technology 56, 2009.
[11]R.Buyya, C.S. Yeo, and S.Venugopal, “Market-Oriented Cloud Computing:Vision, Hype, and Reality for Delivering IT Services as Computing Utilities”, Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08), Sep. 2008
[12] W.Y. Lin, G.Y. Lin, and H.Y.Wei, “Dynamic Auction Mechanism for Cloud Resource Allocation”, 10th IEEEACM International Conference on Cluster Cloud and Grid Computing (2010)
[13] G.Wei, A.V. Vasilakos, Y.Zheng, and N.Xiong, “A game-theoretic method of fair resource allocation for cloud computing services”, The journal of supercomputing, Vol.54, No.2.
[14] Yazir, Y.O., Matthews, C., Farahbod, R., Neville, S., Guitouni, A., Ganti, S., and Coady, Y., “Dynamic Resource Allocation in Computing Clouds through Distributed Multiple Criteria Decision Analysis”, 2010 IEEE 3rd Internatiuonal Conference on Cloud Computing (CLOUD 2010), July 2010.
[15] B.Malet and P.Pietzuch, “Resource Allocation across Multiple Cloud Data Centres”, 8th International workshop on Middleware for Grids, Clouds and e-Science. (MGC'10), Nov. 2010.
[16] G.Leey, B.G.Chunz, and R.H.Katz, “Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud”, HotCloud '11 June. 2011.
[17] B. Rajkumar, B. Anton, and A. Jemal, “Energy efficient management of data center resources for computing: Vision, architectural elements and open challenges,” in International Conference on Parallel and Distributed Processing Techniques and Applications, Jul. 2010.
[18] M. Mazzucco, D. Dyachuk, and R. Deters, “Maximizing Cloud Providers’ Revenues via Energy Aware Allocation Policies,” in 2010 IEEE 3rd International Conference on Cloud Computing. IEEE, 2010.
[19] K.Mochizuki and S.Kuribayashi, “Evaluation of optimal resource allocation method for cloud computing environments with limited electric power capacity”, Proceeding of the 14-th International Conference on Network-Based Information Systems (NBiS-2011), Sep. 2011.

 ISBN: 978-1-61208-216-5