JOINT MULTIPLE RESOURCE ALLOCATION METHOD FOR CLOUD COMPUTING SERVICES WITH DIFFERENT QOS TO USERS AT MULTIPLE LOCATIONS

Shin-ichi Kuribayashi[1]

[1]Department of Computer and Information Science, Seikei University, Japan
E-mail: kuribayashi@st.seikei.ac.jp

ABSTRACT

In a cloud computing environment, it is necessary to simultaneously allocate both processing ability and network bandwidth needed to access it. The authors proposed the joint multiple resource allocation method in a cloud computing environment that consists of multiple data centers with different QoS (Quality of Service).

This paper proposes to enhance the existing joint multiple resource allocation method, so that it can handle multiple heterogeneous resource-attributes. Resource-attributes of bandwidth, for example, are network delay time, packet loss probability, etc. 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 QoS for the key resource-attribute as it satisfies required Quality of Service. It is demonstrated by simulation evaluations that the enhanced method (Method A) can reduce the total amount of resources up to 30%, compared with the existing method. It is also highly likely that each data center provides the different network delay to users at multiple locations.This paper proposes the further enhancement ofMethod A in order to handle the case where each data center provides the different network delay to users at multiple locations.

KEYWORDS

Cloud computing, heterogeneous QoS, joint multiple resource allocation

1. INTRODUCTION

Cloud computing services 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

DOI : 10.5121/ijcnc.2013.5501

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, storage capacity and power capacity) simultaneously in a coordinated manner, instead of allocating each type of resource independently [6]-[9].

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 "**Existing** method ").The basic idea of Existing method 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 Existing method 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].

Existing method 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 Existing methodto handle multiple heterogeneous resource-attributes. In reality different access points can experience different network delay even if they are connected to the same center. Therefore, it is proposed thefurther enhancement of Existing method in order to handle the case where each data center provides the different network delay time to users at multiple locations.

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. For the preliminary evaluation, this paper assumes two types of resources (processing ability and bandwidth), loss-system based services and the static resource allocation.Section 4 proposes to enhance Existing method to be able to handle multiple heterogeneous resource-attributes(referred to as "**Method A**"). It also describes simulation evaluations which confirm the effectiveness of the Method A. Moreover, Section 5 proposed the further enhancement of Method A, in order to handle the case where each data center provides the difference network delay to users at multiple locations (referred to as "**Method B**").Finally, Section 6presents the conclusions. This paper is an extension of the study in References [21] and [22].

2. RELATED WORK

Resource allocation for a cloud computing environment has been studied very extensively in References [11]-[20]. References [15],[16] have proposed automatic or autonomous resource management in cloud computing. Reference [11] has proposed the heuristic algorithm for optimal allocation of cloud resources. Reference [17] has presented the system architecture to allocate resources assuming heterogeneous hardware and resource demands. References [12] and [13] have proposed the market-oriented allocation of resources including auction method.

Reference [14] has proposed to use game-theory to solve the problem of resource allocation. Energy-aware resource allocation methods have been proposed [19],[20].

However, most of conventional studies on resource allocation in a cloud computing environment 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.For example, resource-attributes of bandwidth are network delay time, packet loss probability, required electric power capacity, etc. Resource-attributes of processing ability are computation time, memory size, required electric power capacity, etc. It is also required to assume that each data center should provide the different network delay to users at multiple locations.

3. RESOURCE ALLOCATION MODEL FOR A CLOUD COMPUTING ENVIRONMENT

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. 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 different resource-attributes of processing ability and network bandwidth are provided by each center. For the preliminary evaluation, this paper assumes two types of resources (processing ability and bandwidth), loss-system based services and the static resource allocation. Moreover, this paper considers a model in which the network delay from an access point to a center differs from center to center and also from access point to access point, as shown in Figure 2. t_{Aj} in Figure 2 is the network delay time to access Center j from point A.

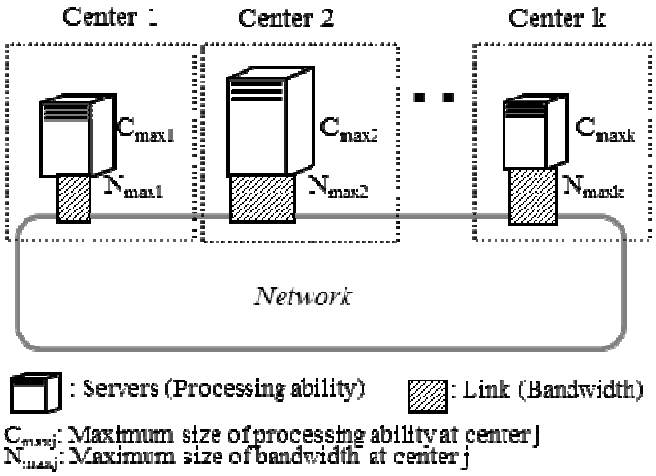


Figure 1. Resource allocation model for a cloud computing environment

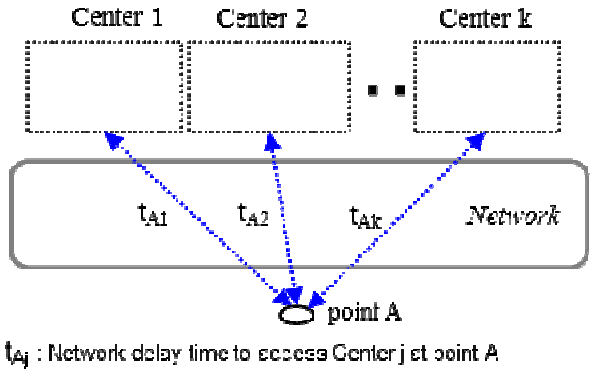


Figure 2. Network delay model

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 are the same as those in References [6]-[9].

3.2 Necessity of resource allocation guideline 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 Existing method proposed in References [6] and [7] is explained with Figure 3. There are five centers in different locations, and that each center has two resource-types: bandwidth and processing ability. In Figure 3(1), centers are divided to two 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, Existing method 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. On the other hand, Figure 3(2) considers center groups taking a resource-attribute (computation time) of processingability into consideration. If a request has no stringentrequirement on computation time, Existing methodfirst 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.

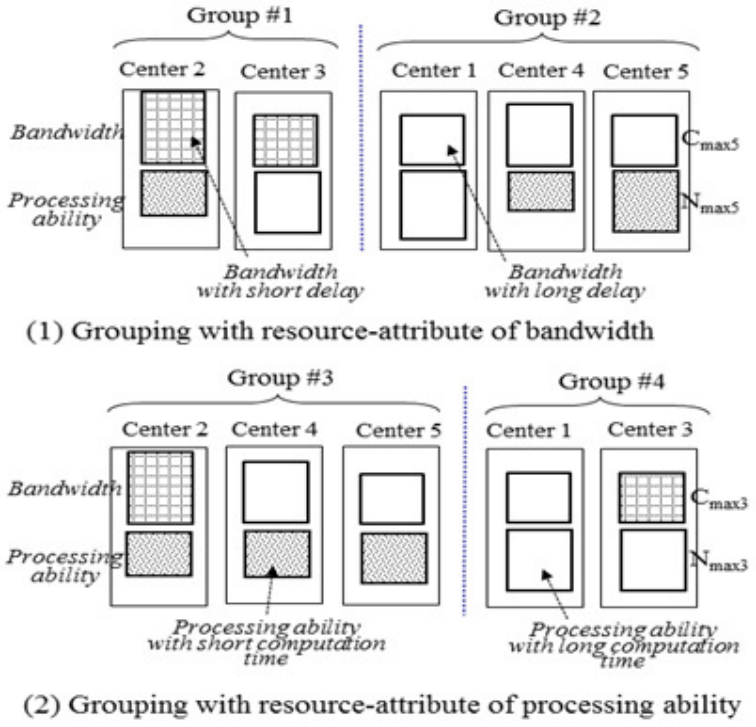


Figure 3. Example of resource allocation assuming heterogeneous resource-attributes

In this way, the priority with which a center group is selected differs between Figure 3(1) and Figure 3(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 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. The proposed guidelines are explained in Section 4.

The guideline could also be applicable 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 4. 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.

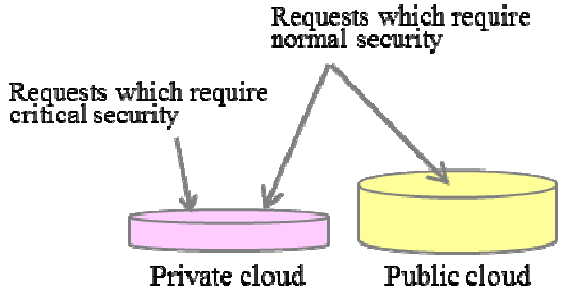


Figure 4. Services with both private and public cloud

4. ENHANCEMENT OF EXSITING METHOD TO SUPPORT MULTIPLE RESOURCE-ATTRIBUTES

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 by the authors in References [6] and [7]. That is, it is proposed to allocate resources focusing on the most impact 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 new resource allocation algorithm(referred to as **Method A**) which enhanced Existing methodis explained in the next Section 4.2, which adopts the concept of key resource-attribute above.

4.2 Resource allocation algorithm of Method A

4.2.1 Identification of key resource-attribute

An attribute with the maximum 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%). In this case, the resource-attribute j with the next largest value of M_j is selected as key resource-attribute.

4.2.2 Identification of a center group

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

4.2.3 Selection of a center

(i) A center that can provide multiple resources required by the request is selected. If there is no center that can satisfy the requirement, the request is rejected.

(ii) A center is selected as follows depending on the QoS required by the request:

- If the request requires lower QoS than that associated with key resource-attribute, it is tried to select a center in Center Group X. If there are several selectable centers, the center with the least available amount of resources is selected. 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.

- If the request requires the QoS associated with key resource-attribute, a center is selected in Center Group Y. If there are several selectable centers, the center with the least available amount of resources is selected.If there is no selectable center there, a center in Center Group Z is selected.

- If the request requires higher QoS than that associated with key resource-attribute, it is tried to select a center in Center Group Z. If there are several selectable centers, the center with the least available amount of resources is selected.

4.2.4 Resource allocation

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 are released.

4.3 Simulation Evaluation

4.3.1 Evaluation model

1) Method A proposed in Section 4.2 is evaluatedusing 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) We consider three centers, centers 1, 2 and 3 and each center provides resources with different resource-attributes. Moreover, two cases in Figure 5 are assumed. Case 1 assumes that the maximum amount of resources at each center is uneven although Case 2 assumes uniform. In both Cases, center 3 only provides a high quality of computation time(high_1) and center 2 only provides a high quality of network delay time (high_2).Any attribute other than high_1 or high_2 is referred to as 'normal'.
4) As for requests, three types in Table 1are considered. Type_1 requests require normal quality for both computation time and network delay time. Type_2 requests require a high quality (high_1) for computation time.Type_3 requests require a high quality (high_2) for network delay time. q_1, q_2 and q_3 are the generation ratio of type_1, type_2, and type_3 respectively. The value of q_2 and $q_3$is set to $(1-q_1)/2$.

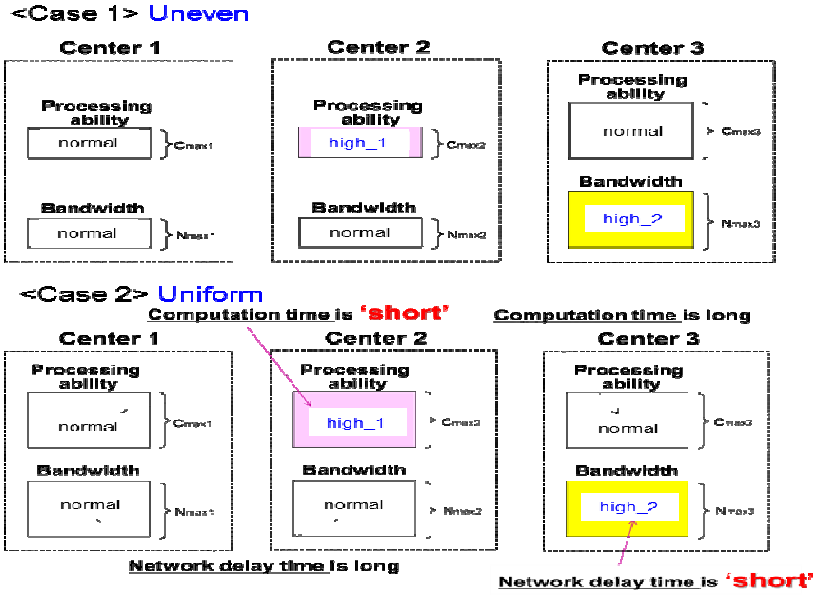


Figure 5. Center resources for simulation model

Table 1. Request types for simulation evaluation

Request type	Computation Time	Network delay time	Generation ratio
type_1	normal	normal	q_1
type_2	high 1	normal	q_2
type_3	normal	high_2	q_3

- Average required size of processing ability C : Gaussian distribution
- Average required size of bandwidth N : Gaussian distribution

- $q_2=q_3=(1-q_1)/2$
- $q_1+q_2+q_3=1$

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

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; …; 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.

4.3.2 Simulation results and evaluation

The simulation results are illustrated in Figures 6, 7 and 8. The horizontal axis shows the probability $q_1$at which type_1 request occurs. The vertical axis of Figures 6 and 7 shows the average request loss probability. The vertical axis of Figure8 shows the ratio of required amount of resources by Method Aand those by Existing method, on the condition of keeping the same average request loss probability.Figure 6(1) shows evaluation results for the case where the request generation pattern is uniform. Figure 6(2) shows the case where it is uneven (i.e., rise and fall in anti-phase). Figure 7 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 7, the total resource amount of center 3 is twice that ofcenter 1 or center 2 in Figure 6. Figure 7(1) and 7(2) show the total average request loss probability and the requestloss probability for each request-type respectively.The parenthesis following Existing method or Method A in Figure 7indicates the request-type.The following points are clear from these Figures:

i) Except for the area where almost all requests are type_1, the request loss probabilities of Method A and Existing method 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 methodtends to select center 2 or center 3 more often compared with Method A or Existing method. The reason why there is not much difference in results between Methods A and Existing method is that type_1 requests use almost all resources in centers 1, 2 and 3 when q_1 comes close to 1.0.

ii) The request loss probability of Method A is smaller than that of Existing methodwhen the total resource amount used by each request-type is different. The differences in the request loss probability between different request-types can also be made smaller by Method A.

<Reason>Existing method, 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 A, 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 resourcesare kept available in center 2 than in the case of using Existing method, 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 A.

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

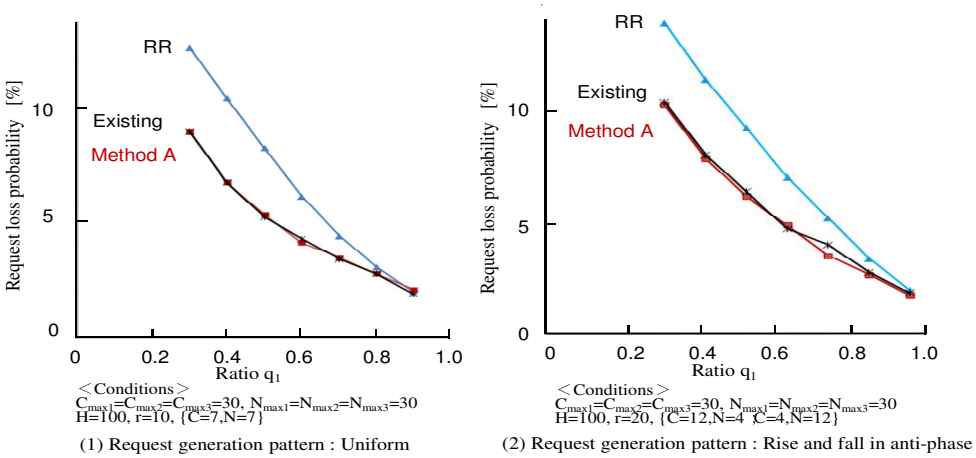


Figure 6. Comparative evaluation of RR method, Existing method and method A (Case 1 in Figure 5)

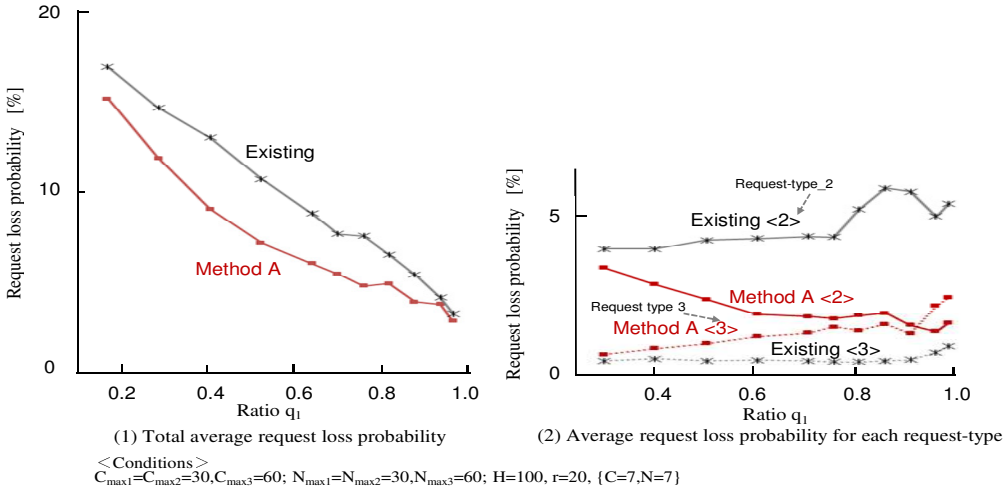


Figure 7. Comparative evaluation of Existing method and method A (Case 2 in Figure 5)

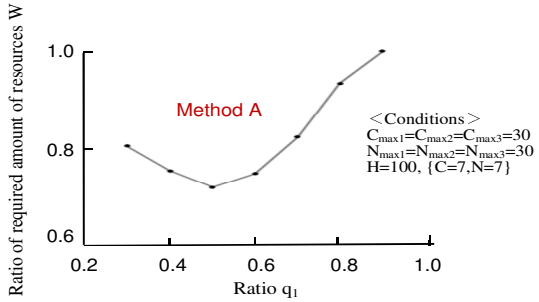


Figure 8. Ratio of required amount of resources

5. ISSUE OF METHOD A AND PROPOSED SOLUTION

5.1 Issue of Method A

Method A in Section 4 considers only a specific access point. In reality different access points can experience different network delay even if they are connected to the same center. If resources are allocated without assuming that a network delay differs from access point to access point, a service competition will arise from differences in the locations of access points, as shown in Figure 9. In Method A, if center 1 provides short-delay access for point A, for example, point A selects center 2 with high priority for requests that can tolerate a long delay, in order to ensure that the resources of center 1 will be available for future requests that cannot accept a long delay. However, the delay situation is reversed for point B. Point B selects center 1 with high priority, resulting in service competition between points A and B in the preferred use of resources.

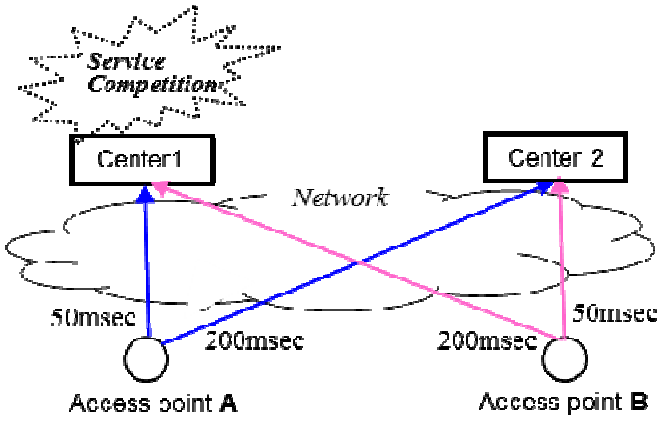


Figure 9. An example of service competition by different access points

5. 2 Proposed solution

(1) Objective
It aims to minimize the average request loss probability for all access points and for all levels of request quality.

(2) Proposed solution
<Solution 1> When a center finds that the amount of available resources has dropped below a certain threshold (referred as 'alevel'), it accepts only those requests that require the highest quality. A conceptual diagram of Solution 1 is shown in Figure 10.In Figure 10, center 1 provides short network delay for point A and long network delay for point B. Center 1 restricts requests from point B if the amount of available resource is less than MAX*alevel. Here, MAX is the total amount of resources at center 1.
<Solution 2> Each center reserves a certain amount of resources for each access point, and allows the remaining amount of resources to be shared.
Although solution 2 is more impartial, its logic is more complex. Therefore, this paper adopts the enhancement of Method A with solution 1 mechanism (referred to as "**Method B**") for the joint multiple resource allocation.

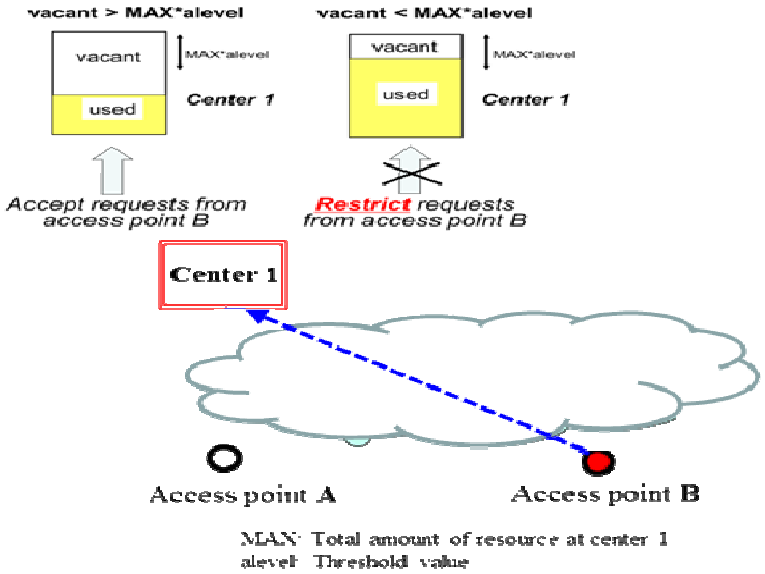


Figure 10. Image of Solution 1

5.3 Simulation evaluation

5.3.1Evaluation model

1) Method B proposed in Section 5.2 is evaluated using a (self-made) simulator written in the C language.
2) For the preliminary evaluation, we consider two access points (points A and B) and use a network delay time as a resource-attribute. Two levels of network delay are considered: “normal quality” (long delay) and “high quality” (short delay).
3) There are two centers (centers 1 and 2), each providing the following network delay:
<center 1>high quality for point A and normal quality for point B.

<center 2> normal quality for point A and high quality for point B.
4)
-Request generation ratio between points A and B
point A : point B = prob_a : (1-prob_a) (2)
-Request generation ratio at point A between requests requiring ‘high quality’ and requests requiring ‘normal quality’
‘high quality’ : ‘normal quality’ = prob_10 : (1.0 - prob_10) (3)
-Request generation ratio at point B between requests requiring ‘high quality’ and requests requiring ‘normal quality’
‘high quality’ : ‘normal quality’ = prob_20 : (1.0 - prob_20) (4)
5) When a new request is generated, one appropriate center is selected according to Method B in Section 5.2 and then both processing ability and bandwidth from that center is allocated

to the request simultaneously. Requests requiring high quality from point A can select only center 1. Requests requiring normal quality from point A select center 2 with high priority and select center 1 only when center 2 does not have a sufficient amount of available resources. On the other

hand, requests requiring normal quality from point B select center 1 with high priority and select center 2 only when center 1 does not have a sufficient amount of available resources.

6) As for the size of required processing ability and bandwidth by each request, the intervals between requests, the length of resource holding time and the pattern in which requests occur, the same conditions as in section 4.3.1 are applied.

5. 3.2 Simulation Results And Evaluation

Figure 11 illustrates two traffic conditions for evaluations, and Figures 12 to 14 illustrate the simulation results. It is assumed that 60% of all requests occur from point A (i.e., prob_a is 0.6) and the value of alevel at center 2 is 0. As for traffic condition #1 in Figure 11, it is assumed that 90% of requests occurring from point A require short delay (i.e., prob_10 is 0.9) while 10% require long delay. It is also assumed that 10% of requests occurring from point B require short delay (i.e., prob_20 is 0.1) while 90% require long delay.As for traffic condition #2, it is assumed that 50% of requests occurring from point A require short delay (i.e., prob_10 is 0.5), and 50% of requests occurring from point B require short delay (i.e., prob_20 is 0.5). Figure 12and Figure 13 illustrate the simulation results assuming traffic condition #1 and #2 respectively. The vertical axis of both Figure 12(1) and Figure 13(1) shows the average request loss probability. The vertical axis of both Figure 12(2) and Figure 13(2) shows the minimum amount of resources in the case of Method B, which is required to keep the average request loss probability of 5% or less. The value is normalized relative to the minimum amount of resources similarly required in the case of Method A. For example, 0.8 means that Method B can decrease the required resources by 20%, compared with Method A. The horizontal axis of both Figures 11 and 12 shows the value of ‘alevel of Center 1’. Figure 14 shows how the average request loss probability for all requests changes when the value of prob_10 changes. Prob_10 is the ratio of requests from point A which require short delay. The horizontal axis of Figures 14 shows the value of prob_10.

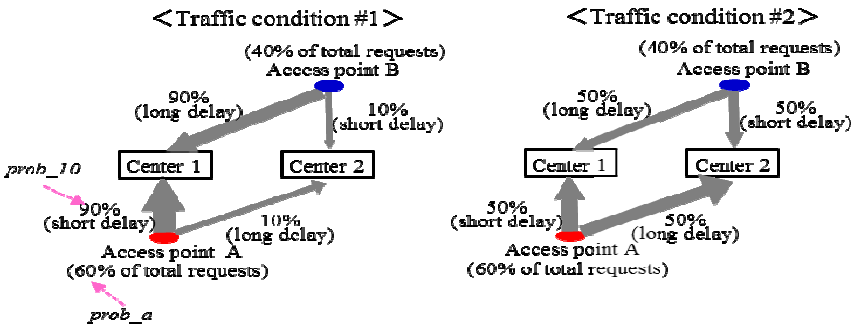
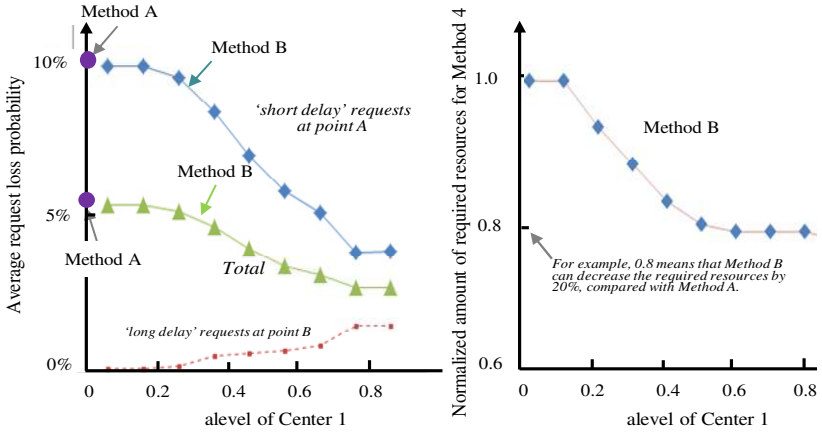
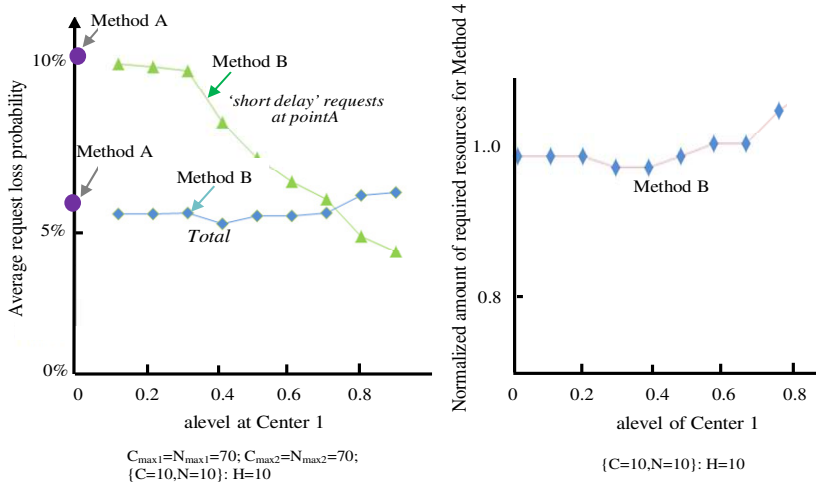


Figure 11. Traffic conditions for evaluation



(1) Request loss probability (2) Resource efficiency

Figure 12. Comparative evaluation of Method A and Method B (traffic condition #1)



(1) Request loss probability (2) Resource efficiency

Figure 13. Comparative evaluation of Method A and Method B (traffic condition #2)

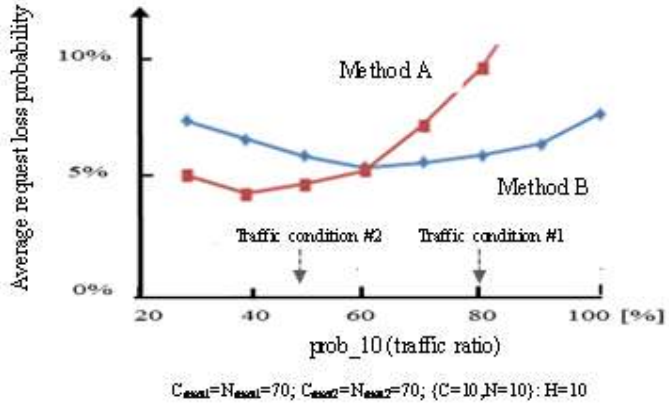


Figure 14. Impact of traffic ratio prob_10

The following points are clear from these figures:

i) Method B can reduce the average request loss probability greatly and the amount of required resources by up to 20%, compared with Method A. Method B could make the average request loss probability low when prob_10 is high (more than 60%).The effectbecomes large especially when the value of alevel is around 0.8.
[Reason] In Method A, requests requiring long delay from point B select center 1 with high priority and without any restriction, in spite of the fact that it can also use center 2 as well. Requests requiring short delay from point A can select only center 1 and this makes the request loss probability of requests requiring short delay high.

On the other hand, as Method B restricts some of requests requiring long delay from point B, more resources would be allocated for requests requiring short delay from point A and this makes the request loss probability of these requests lower. When the value of alevel is more than 0.8, most of requests from point B will be restricted and it makes the total average request loss probability high.Itis lso noted that the request loss probability of requests from point B will not become high, because requestrestricted atcenter 1 could select center 2 which has more available resources.

ii) It should not restrict requests from point B with Method B when prob_10 is small.
[Reason] Method B will restrict many requests from point B, even though there are available resources in Center 1. In other words, Method Bwould reserve resources for access point A more than needed when prob_10 is small..

6. CONCLUSIONS

This paper has proposed to enhance the existing joint multiple resource allocation method so that it can handle multiple heterogeneous resource-attributes. The basic idea of the enhanced method, Method A, 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 A can reduce the total amount of resources up to 30%, compared with theexisting method.

This paper has proposed the further enhancement ofMethod A, in order to handle the case where each data center provides the different network delay to users at multiple locations. It is demonstrated by simulation evaluations that the enhancement of Method Acan greatly make the request loss probability lower and reduce the amount of required resources by up to about 20%, compared with Method A.

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. It is required to make an evaluation with larger numbers of these to confirm the effectiveness of the proposed methods and to identify the conditions in which the proposed methodsare effective. Moreover, Method B could not process other requests at all if a lot of requests that require the highest quality occur more than the expected number. It is for further study to solve this issue.

REFERENCES

[1] G.Reese: "Cloud Application Architecture", O'Reilly& Associates, Inc., Apr. 2009.

[2] J.W.Rittinghouse and J.F.Ransone: "Cloud Computing: Impementation, 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] S.Kuribayashi, "Reducing Total Power Consumption Method in Cloud Computing Environments", International journal of Computer Networks & Communications (IJCNC), Vol.4, No.2, pp.69-84, March 2012.

[10] 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.

[11] 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.

[12] 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

[13] 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)

[14] 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.

[15] 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.

[16] 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.

[17] G.Leey, B.G.Chunz, and R.H.Katz, "Heterogeneity-Aware Resource Allocation and Scheduling in the

Cloud", HotCloud '11 June. 2011.
[18] 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.
[19] 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.
[20] 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.
[21] Yuuki Awano and Shin-ichi Kuribayashi, "Proposed Joint Multiple Resource Allocation Method for Cloud Computing Services with Heterogeneous QoS", The Third International Conference on Cloud Computing, GRIDs,and Virtualization(CLOUD COMPUTING 2012) , July 2012.
[22] Yuuki Awano and Shin-ichi Kuribayashi, "A joint multiple resource allocation method for cloud computing environments with different QoS to users at multiple locations" Proceeding of 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (Pacrim13), Aug. 2013.

AUTHORS

Shin-ichi Kuribayashi received the B.E., M.E., and D.E. degrees from Tohoku University, Japan, in 1978, 1980, and 1988 respectively. He joined NTT Electrical Communications Labs in 1980. He has been engaged in the design and development of DDX and ISDN packet switching, ATM, PHS, and IMT 2000 and IP-VPN systems. He researched distributed communication systems at Stanford University from December 1988 through December 1989. He participated in international standardization on ATM signaling and IMT2000 signaling protocols at ITU-T SG11 from 1990 through 2000. Since April 2004, he has been a Professor in the Department of Computer and Information Science, Faculty of Science and Technology, Seikei University. His research interests include optimal resource management, QoS control, traffic control for cloud computing environments and green network. He is a member of IEEE, IEICE and IPSJ.