RESOURCE ALLOCATION METHOD FOR CLOUD COMPUTING ENVIRONMENTS WITH DIFFERENT SERVICE QUALITY TO USERS AT MULTIPLE ACCESS POINTS

Shin-ichi Kuribayashi

Department of Computer and Information Science, Seikei University, Japan

ABSTRACT

In a cloud computing environment with multiple data centers over a wide area, it is highly likely that each data center would provide the different service quality to users at different locations. It is also required to consider the nodes at the edge of the network (local cloud) which support applications such as IoTs that require low latency and location awareness. The authors proposed the joint multiple resource allocation method in a cloud computing environment that consists of multiple data centers and each data center provides the different network delay. However, the existing method does not take account of cases where requests that require a short network delay occur more than expected. Moreover, the existing method does not take account of service processing time in data centers and therefore cannot provide the optimal resource allocation when it is necessary to take the total processing time (both network delay and service processing time in a data center) into consideration in resource allocation.

This paper proposes to enhance the existing joint multiple resource allocation method, so as to provide the following two functions: (1) a function to prevent the degradation in service quality of other request types when requests that require a short network delay occur more than expected, and (2) a function to take account of the total processing time of network delay and service processing time in allocating resources. It is demonstrated by simulation evaluations that the enhanced method can handle up to twice as many requests as the existing method with the same amount of resources, and can cope with the excessive generation of requests from the specific access point.

KEYWORDS

Cloud computing, joint multiple resource allocation, different service quality, multiple access points, total processing time.

1.INTRODUCTION

Cloud computing services are allow the user to rent, only at the time when needed, only a desired amount of computing resources out of a huge mass of distributed computing resources at multiple data centers [1]-[4]. It is also necessary to allocate simultaneously a network bandwidth to access them and the necessary power capacity [5]-[7]. As cloud computing services rapidly expand their customer base, it has become important to provide them economically.

DOI : 10.5121/ijcnc.2015.7603

It is highly likely that each data center would provide the different network delay to users at different locations in a cloud computing environment. In addition, it is also required to consider the nodes at the edge of the network in local cloud (Fog computing [8],[9] or Edge computing[10]), which support applications that require low latency and location awareness. The nodes at the edge, for example, ingest the data generated by grid sensors and devices, and some of those data relates to protection and control loops that require real-time processing. The node processes the data, and could issue control commands to the actuators. Mobile applications for smartphones would be processed at the edge nodes, instead of remote servers located far away, to make up for performance lack of the smartphone.

The authors have proposed a resource allocation method that selects the optimal data center from among multiple data centers to meet requests from a specific single access point, and simultaneously allocates different types of resources, such as processing ability, bandwidth and power supply, to each request [11]. The authors have then proposed to enhance the method in Reference [11] so as to meet requests from multiple access points simultaneously, considering the fact that the network delay experienced can vary depending on the access point (referred to as "**Method B**") [12].

Although Method B focuses on ensuring the service quality for requests that require a short network delay (hence, can access only limited data centers), it cannot provide optimal resource allocation when requests that require a short network delay occur more than expected. In addition, Method B takes account of whether the response time is long or short but does not take account of whether the processing speed is high or low. Therefore, Method B could not provide optimal resource allocation for services in which the total processing time, which includes not only network delay time but also service processing time in a data center, is critical for the user.

This paper proposes to enhance the existing joint multiple resource allocation method, Method B [12], so as to provide the following two functions: (1) a function to prevent the degradation in service quality of other request types when requests that require a short network delay occur more than expected, and (2) a function to take account of the total processing time of network delay and service processing time in allocating resources.

The rest of this paper is organized as follows. Section 2 explains related works. Section 3 identifies two major issues of the existing joint multiple resource allocation, Method B, and proposes to enhance Method B so as to solve the issues. Section 4 describes simulation evaluations which confirm the effectiveness of the proposed enhancements. Finally, Section 5 gives the conclusions. This paper is an extension of the study in References [22] and [23].

2.RELATED WORK

Resource allocation for cloud computing environments has been studied very extensively in References [13]-[21]. Reference [13] has proposed the heuristic algorithm for optimal allocation of cloud resources. References [14] and [21] have proposed market-oriented allocation of resources including auction method. Reference [15] has proposed to use game-theory to solve the problem of resource allocation. References [16] and [17] have proposed automatic or autonomous resource management in cloud computing. Reference [18] has presented the system architecture to allocate resources assuming heterogeneous hardware and resource demands. Energy aware resource allocation methods for cloud environments have been proposed in Reference [20].

However, most of conventional studies on resource allocation in a cloud computing environments are treating each resource-type individually. To the best of our knowledge, the cloud resource allocation has not been fully studied which assumes that multiple resources are allocated simultaneously to each service request and each data center provides the different service quality to users at multiple locations.

3.ISSUES OF METHOD B AND SOLUTIONS

3.1 ISSUE 1 OF METHOD B

3.1.1 DETAILS OF THE ISSUE

The system model for cloud computing services assumed in this paper is illustrated in Figure 1. This model is the same as that used in Reference [12]. There are k centers at different places. Each center has servers (including virtual 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. When a service request is generated, the resource manager in the system selects one optimal center from among k centers, and it allocates both the processing ability and bandwidth in that center simultaneously to the request for a certain period. The processing ability and bandwidth of more than one center cannot be allocated to a request. When the resource holding time has elapsed, the allocated processing ability and bandwidth are released.

In Method B, resources at each data center are shared by the type of request (Type 1 request) that cannot tolerate a long network delay and the type of request (Type 2 request) that can tolerate. The allocation of resources to Type 2 requests is suppressed in Method B when the amount of available resources has dropped below a certain threshold. The objective is to maintain the service quality of Type 1 requests, which can access only limited data centers. For example, in Figure 2, the access from Point Y is suppressed to set aside the available resource for use by Type 1 requests from Point X. However, when Type 1 requests from Point X occur more than expected, requests from Point X more or less monopolizes the resource in Center 1, rejecting almost all Type 2 requests from Point Y.

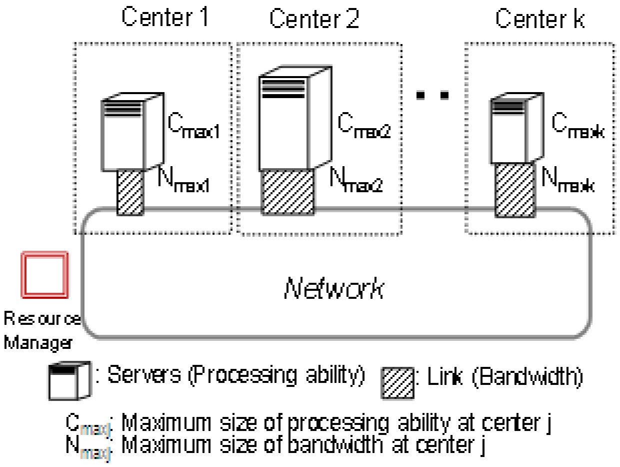

Center 1
Center 2
Center k
Cmax1
Nmax1
Cmax2
Nmax2
Cmaxk
Nmaxk
Network
Resource Manager
: Servers (Processing ability)
: Link (Bandwidth)
Cmaxj: Maximum size of processing ability at center j
Nmaxj: Maximum size of bandwidth at center j

Figure 1. Systemmodel for Cloud computing Services

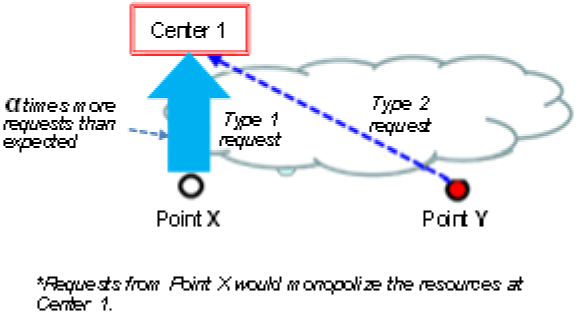

Center 1
α times more requests than expected
Type 1 request
Type 2 request
Point X
Point Y
*Requests from Point X would monopolize the resources at Center 1.

Figure 2. Issue 1 of Method B

3.1.2 PROPOSED SOLUTION

In general, Alternatives 1 and 2 in Figure 3 can be conceived for allocation of center resources to multiple access points. In Alternative 1, a dedicated center is established for each access point and the center cannot be accessed by other points. However, Alternative 1 has several problems. First, the divided resource would lead to a drop in resource efficiency. Second, when a specific center is congested, the access point associated to it cannot use other centers which have available resources. Third, it is difficult to flexibly cope with an increase or decrease in the number of access points. In Alternative 2, centers can be accessed by all access points. Alternative 2 could not accept Type 1 requests from access points located away from the center.

Therefore, a third alternative, Alternative 3, is proposed as shown in Figure 3. To deal with the problem of Alternative 2, centers are widely distributed so that requests that require a short network delay can be accepted. To solve a problem of Alternative 1 (to minimize a drop in resource efficiency), resources in each center are shared by multiple access points. In addition, minimum resources dedicated to each access point are placed in each center, in order to prevent an access point from monopolizing the resources in a center and to ensure a minimum service quality.

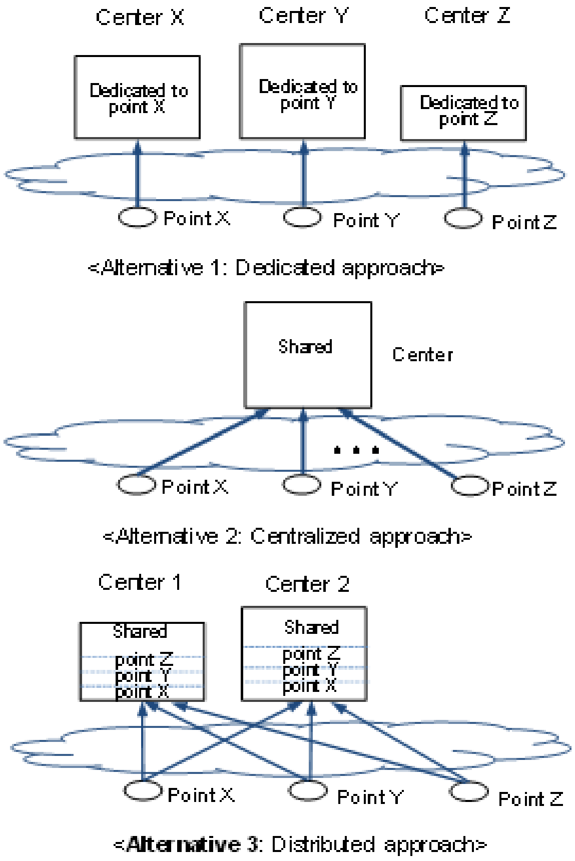


Figure 3. Resource allocation approaches

The algorithm to decide the amount of resource placed in each center for Alternative 3 is as follows:

<ASSUMPTIONS>

As shown in Figure 4, it is assumed that there are 2 access points and 2 centers. Figure 4 is illustrated with a focus on the resource type ('identified resource') that requires the largest proportionate size of resource, comparing the size of required resource with the maximum resource size for each resource type [7],[11],[12].

Parameters are defined as follows:

- x_1, y_1, xy_1: Amount of resources placed in Center 1 dedicated to Point X, Point Y and Points X&Y, respectively.
- max_1: $x_1+y_1+xy_1$
- x_2, y_2, xy_2: Amount of resources placed in Center 2 dedicated to Point X, Point Y and Points X&Y, respectively.
- max_2: $x_2+y_2+xy_2$
- dx_1,dy_1: Network delay time to center 1 from point X and point Y, respectively
- dx_2,dy_2: Network delay time to center 2 from point X and point Y, respectively
- λx_1: Expected rate of Type1-request occurrence from Point X which requires a short network delay.
- λx_2: Expected rate of Type2-request occurrence from Point X which tolerates a long delay time.
- λy_1: Expected rate of Type1-request occurrence from Point Y which tolerates a long delay time.
- λy_2: Expected rate of Type2-request occurrence from Point Y which requires a short network delay.
- Hx：Average resource holding time of requests from point X （The value is supposed to be constant regardless of the selected center）
- Hy：Average resource holding time of requests from point Y （The value is supposed to be constant regardless of the selected center）
- Sx: Average resource size of request occurred from Point X.
- Sy: Average resource size of request occurred from Point Y.

<ALGORITHM TO DECIDE THE AMOUNT OF RESOURCE>

1. The amount of resource dedicated to a specific access point (x_1, y_1, x_2, y_2)

- The required amount of resource is calculated assuming that Type 1 requests would select a center that can be accessed with a short delay. According to this assumption, Point X accesses Center 1, and Point Y accesses Center 2.
- The required amount of resource is calculated assuming that Type 2 requests would select a center that can be accessed with a long delay time. According to this assumption, Point X accesses Center 2, and Point Y accesses Center 1.
- According to the above assumption, the amounts of resources placed in Centers 1 and 2 for Type 1 requests and for Type 2 requests are calculated using the following equation:
- Required amount of resource =‘Expected rate of request occurrence’ ×‘Average resource holding time’ (1)

2. The amount of resource shared by both access points (xy_1, xy_2)

The size of shared resources in each center is determined in such a way that the expected request loss probability at each center is less than a certain value (e.g., 1%) when the shared resource is added to the dedicated resource calculated in 1) above.

3. Other

Even in the case where only a single access point uses resources in a center, the amounts of resources as calculated in 1) and 2) are installed in the center.

3.1.3 PROPOSED RESOURCE ALLOCATION METHOD

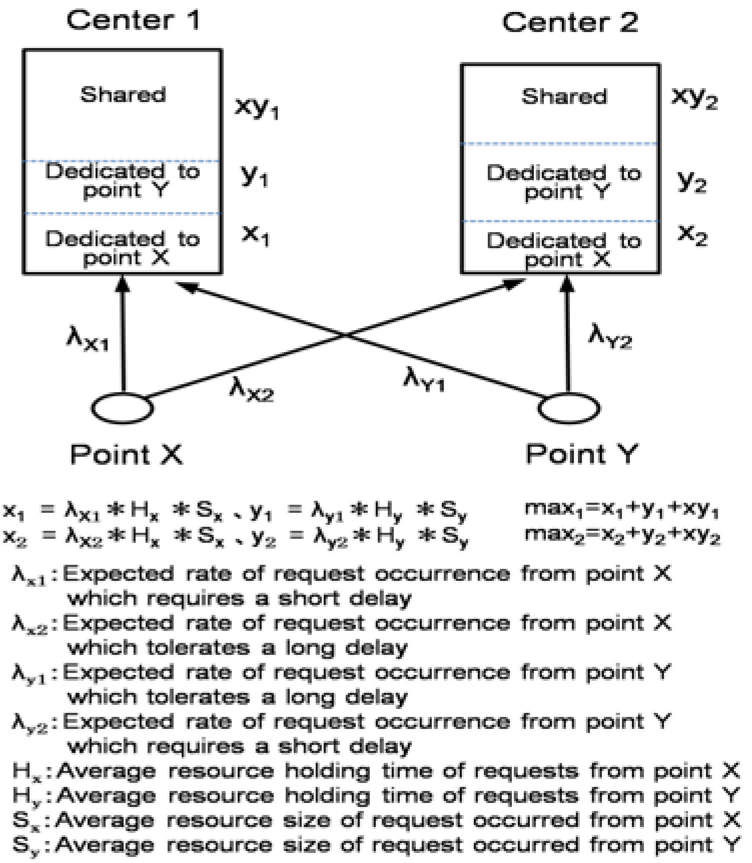
To solve the issue 1 of Method B, it is proposed to add the following function to Method B (referred to as “**Method C**”), which is based on alternative 3 in Figure 3. As in Figure 4, the algorithm of Method C is described below with a focus on the identified resource. Two access points, X and Y are assumed.

<Normal situation>

The resource in center i is divided into the one (x_i) dedicated to access from Point X, the one (y_i) dedicated to access from Point Y, and the one (xy_i) that can be accessed from both Points X and Y. The sizes of x_i, y_i, and xy_i are calculated as described in Section 3.1.2.

<Congested situation>

In the case where requests from either point occur more than expected, no resources are shared, and the total resource in each center is divided into x and y, in proportion to the expected number of requests from each Point (designated by x_{10}, y_{10}, x_{20}, y_{20}, respectively).



$x_1 = \lambda_{X1} * H_x * S_x$ 、$y_1 = \lambda_{y1} * H_y * S_y$ $max_1 = x_1 + y_1 + xy_1$

$x_2 = \lambda_{X2} * H_x * S_x$ 、$y_2 = \lambda_{y2} * H_y * S_y$ $max_2 = x_2 + y_2 + xy_2$

λ_{x1}: Expected rate of request occurrence from point X which requires a short delay

λ_{x2}: Expected rate of request occurrence from point X which tolerates a long delay

λ_{y1}: Expected rate of request occurrence from point Y which tolerates a long delay

λ_{y2}: Expected rate of request occurrence from point Y which requires a short delay

H_x: Average resource holding time of requests from point X

H_y: Average resource holding time of requests from point Y

S_x: Average resource size of request occurred from point X

S_y: Average resource size of request occurred from point Y

Figure 4.Resource allocation method for Method C

<BOUNDARY BETWEEN NORMAL SITUATION AND CONGESTED SITUATION>

There is no clear boundary. That is, the action for the congested situation will be taken only if the average request loss probability in the congested situation is smaller than that in the normal.

Note that it is better to bundle access points into groups based on geographic areas or types of request and to select a center by group rather than by access point, when the number of access points is more than ten.

3.2 ISSUE 2 OF METHOD B

3.2.1 DETAILS OF THE ISSUE

The existing methods including Method B assumed services which would ignore service processing time at a center compared with network delay. When a new service which can't ignore a service processing time at a center is assumed, it is necessary to take account of a total processing time (both network delay and service processing time) for resource allocations and select the center that satisfies the maximum permissible total processing time. Figure 5 illustrates the difference a supposed service for Method B and a new service. The service processing time in a Center is designated by H and the network delay to access a Center is designated by d.

A network model consisting of two points (Points X and Y) and two centers (Centers 1 and 2) is illustrated in Figure 6. The service processing times in Center 1 and Center2 are designated by h_1 and h_2, and the network delay to access Center 1 and Center 2 are designated by d_1 and d_2. It is assumed here that the permitted total processing time for the request is 1050ms. In this example, Center 1 should be selected in order to reserve as much resources as possible for future requests which requires 650m second total processing time and can select only Center 2. However, Method B will select Center 2 as it takes account of only network delay. This results in a high loss probability of requests that cannot tolerate a long total processing time. It is noted that Figure 6 is illustrated with a focus on ‘identified resource’ as with Method C in Section 3.1.

3.2.2 PROPOSED SOLUTION

This section proposes to enhance Method C, so as to take account of the total processing time in allocating resources and to prevent the degradation in service quality of requests from other access points when requests from a specific access point occur more than expected (referred to as “**Method D**”). As with Method C in Section 3.1, Method D classifies center resources into those dedicated to each access point and those shared by access points. Two types of requests are considered: Priority request that requires a short total processing time and Normal request that can tolerate a long total processing time.

The algorithm for determining the amount of resources is described below using the parameters in Figure 7. The parameter definition except the following parameters is the same as in Figure 4:

- hx_1, hy_1 : Service processing time in Center 1 for request from point X and point Y, respectively
- hx_2、hy_2 : Service processing time in Center 2 for request from point X and point Y, respectively

- $\lambda x_1 D$: Expected rate of request occurrence from point X which selects Center1
- $\lambda x_2 D$: Expected rate of request occurrence from point X which selects Center2
- $\lambda y_1 D$: Expected rate of request occurrence from point Y which selects Center1
- $\lambda y_2 D$: Expected rate of request occurrence from point Y which selects Center 2

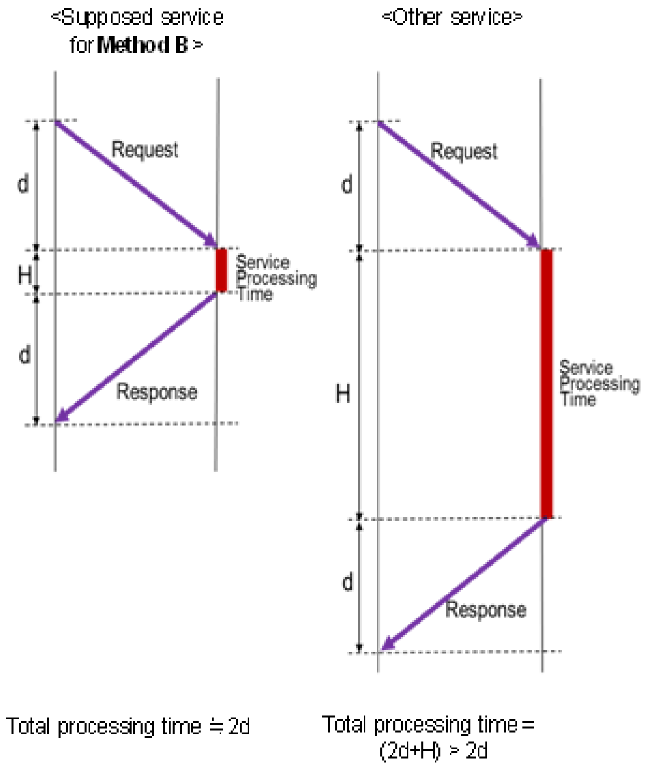


Figure 5.Difference between supposed services for Method B and other service

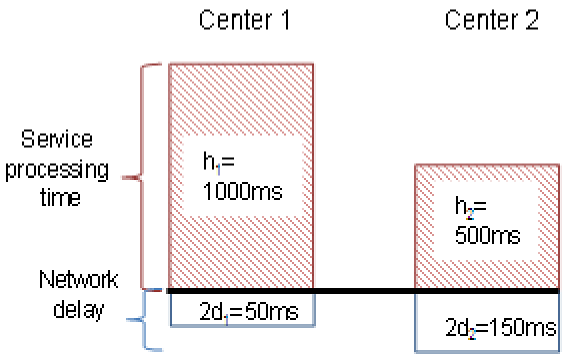
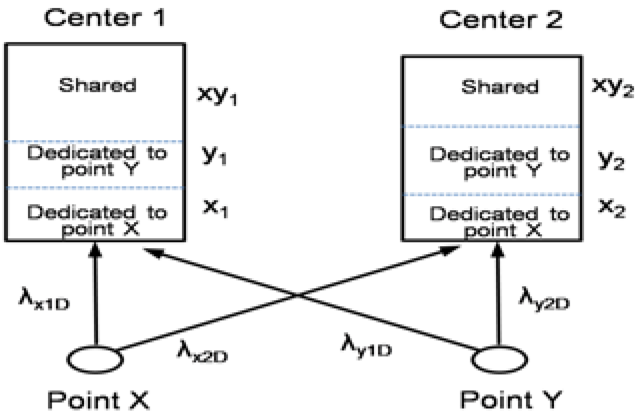


Figure 6.Issue 2 of Method B



$x_1 = \lambda_{x1D} * H_x * S_x$ 、 $y_1 = \lambda_{y1D} * H_y * S_y$ $max_1 = x_1 + y_1 + xy_1$
$x_2 = \lambda_{x2D} * H_x * S_x$ 、 $y_2 = \lambda_{y2D} * H_y * S_y$ $max_2 = x_2 + y_2 + xy_2$

d_{x1}, d_{y1}: Network delay time to center 1 for request from point X and point Y, respectively
d_{x2}, d_{y2}: Network delay time to center 2 for request from point X and point Y, respectively
h_{x1}, h_{y1} : Service processing time in Center 1 for request from point X and point Y, respectively
h_{x2}、 h_{y2} : Service processing time in Center 2 for request from point X and point Y, respectively
λ_{x1D}: Expected rate of request occurrence from point X which selects Center1
λ_{x2D}: Expected rate of request occurrence from point X which selects Center2
λ_{y1D}: Expected rate of request occurrence from point Y which selects Center1
λ_{y2D}: Expected rate of request occurrence from point Y which selects Center 2
H_x: Average resource holding time of requests from point X
H_y: Average resource holding time of requests from point Y
S_x: Average resource size of request occurred from point X
S_y: Average resource size of request occurred from point Y

Figure 7. Resource allocation for Method D

<Algorithm to decide the amount of resources placed in Centers 1 and 2>

The same algorithm for Method C in Section 3 can be applied to Method D, except considering the total processing time instead of network delay.

4.SIMULATION EVALUATIONS

4.1 EVALUATION OF METHOD C

4.1.1 SIMULATION MODEL

Method C is evaluated using a computer simulator written in C++ language. The resource allocation model shown in Figure 4 is assumed. The model has two centers, Centers 1 and 2. The resource types considered are processing ability and bandwidths. It is assumed that the maximum sizes of these two types of resources are the same in both centers. Two access point, Points X and Y, are assumed.

4.1.2 SIMULATION CONDITIONS

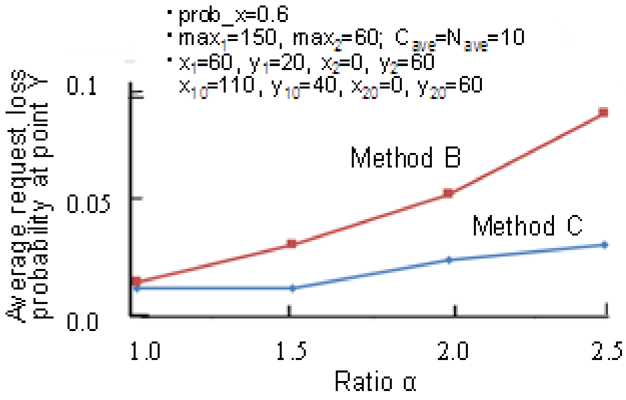
- The following network delay is assumed: 2dx1=50, 2dx2=150, 2dy1=150, 2dy2=50
- Requests from Point X can select only Center 1 while those from Point Y can select both Centers 1 and 2. Prob_x is the ratio of the number of requests from Point X to that from both Point X and Point Y.
- The size of required processing ability and bandwidth by each request is assumed to follow a Gaussian distribution (dispersion is 5). The average size of processing ability and that of bandwidth are designated by Cave and Nave. These are the same for both centers.
- The intervals between requests follow an exponential distribution with the average, r. The length of resource holding time, T, is constant. All allocated resources are released simultaneously after the resource holding time expires.
- The amount of resources is calculated with the algorithm described in Section 3.1.2 as follows, supposing that prob_x is 0.6: x_1=60, y_1=20, x_2=0, y_2=60

4.1.3 RESULTS AND EVALUATION

Figures 8 to 10 illustrate simulation results. Figure 8 shows how an increase in the number of requests from Point X above the expected number affects the request loss probability at Point Y. The number of requests from Point Y is as expected. The horizontal axis indicates that the rate of occurrence at Point X is α times more than expected. The vertical axis shows the average request loss probability at Point Y. Figure 9 shows how the average request loss probability at each access point varies with the value of y_1. The horizontal axis indicates that the value of y_1 is β times the value calculated using the formula given in Section 3.1.2. The vertical axis shows the average request loss probability at each access point. Figure 10 shows how the average request loss probability at Point Y varies with the value of x_1. The horizontal axis indicates that the value of x_1 is γ times the value calculated using the formula given in Section 3.1.2. The vertical axis shows the average request loss probability at Point Y.

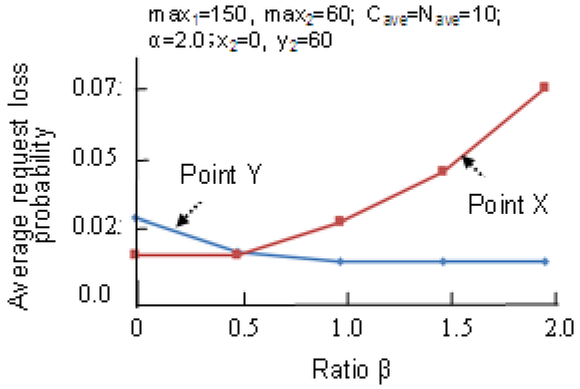
The following points are clear from these Figures:

- With Method C, the request loss probability at Point Y hardly increases even if requests from Point X occur more than expected (i.e., even if the value of α on the horizontal axis is large). However, with Method B in which the number of requests from Point X is not restricted, the request loss probability at Point Y increases dramatically if the value of α on the horizontal axis is large.
- It is reasonable to decide the sizes of xi and yi (i=1,2) with the formulae given in Section 3.1.2. If one of them is set to a large value, the request loss probability of the other increases.
- With additional evaluations, it was confirmed that Method C becomes particularly more advantageous in the following conditions:
  - I. prob_x is more than 0.8
  - II. Ratio α is more than 1.5
- It was also clarified that Method C can cope with the excessive generation of requests from Point Y.



Ratio α= {Rate of request occurrence at point X} / {Expected rate of request occurrence at point X}

Figure 8. Impact of rate of request occurrence at point X



Ratio β= {size of y$_1$}/ {size of y$_1$ defined in Sec. 3.1.2}

Figure 9.Impact of size of Y$_1$

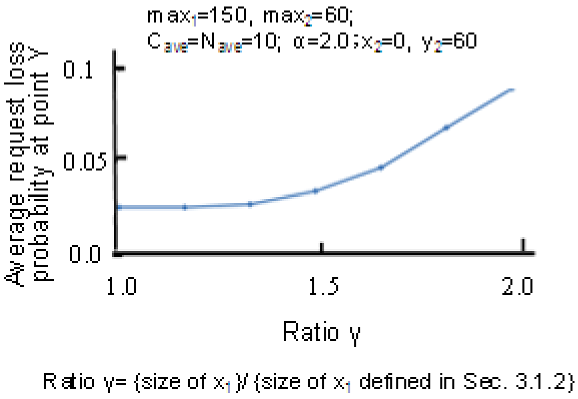


Figure 10.Impact of size of X_1

4.2 EVALUATION OF METHOD D

4.2.1 SIMULATION MODEL

As in Section 4.1, Method D is evaluated using a computer simulator written in C++ language. The resource allocation model shown in Figure 7 is assumed. The same simulation conditions except the following conditions are the same as those in Sec 4.1.2:

- As the total processing time is not depend on the access point, two types of requests are assumed:
  i. 'priority request' which requires a short total processing time
  ii. 'normal request' which can tolerate a long total processing time
- The network delay and service processing time are as follows: $2dx_1$=50, $2dx_2$=150, $2dy_1$=150, $2dy_2$=50; hx_1=1000, hx_2=500,hy_1=1000, hy_2=500
- The amount of resources is calculated with the algorithm described in Section 3.2 as follows, supposing that the percentage of normal requests is 50% (at both Points X and Y). x_1=40, y_1=40, xy_1=10, x_2=20, y_2=20, xy_2=10

4.2.2 EVALUATION RESULTS AND DISCUSSIONS

Figures 11 to 14 illustrate simulation results. Figure 11 compares Method D and Method C in terms of the maximum rate of request occurrence at which the average request loss probability is 1% or lower. The horizontal axis shows the ratio of normal requests. The vertical axis shows the ratio, **δ**, of the maximum rate of request occurrence of Method D to that of Method C. For example, "δ is equal to 2" means that Method D can handle two times as many requests as Method C. In addition, the results for a round-robin method (RR Method), which selects centers 1 and 2 in turn, are presented for comparison.

Figure 12 compares Method D and Method C in terms of the average request loss probability with a varying ratio of normal requests. The horizontal axis shows the ratio of normal requests. The vertical axis in the upper part of Figure 12 shows the average request loss probability of total requests but the vertical axis in the lower part of Figure 12 shows that of priority requests. The amount of resources in each center is the same as those in Figure 11, and the rate of request occurrence is fixed.

Figures 13 and 14 make the same evaluation as in Figure 11 except that the amount of resources in each center. Figure 13 shows evaluation results in a case where x_1=20, x_2=20, xy_1=10; x_2=20, y_2=20, and xy_2=10, which is for a case where the amount of resources in center 1 is the same as that of center 2. Figure 14 shows results in a case where x_1=20, x_2=20, xy_1=10; x_2=40, y_2=40, and xy_2=10, which is for a case where the amount of resources in center 1 and that in center 2 is reversed.

The following points are clear from these figures:

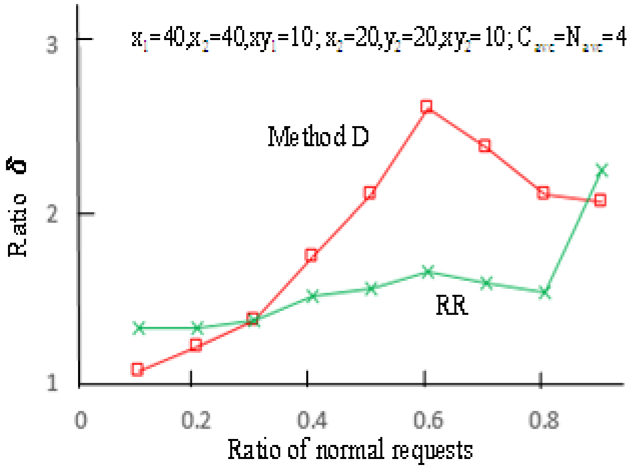


Figure 11.Comparison of maximum rate of request occurrence between Method D and Method C

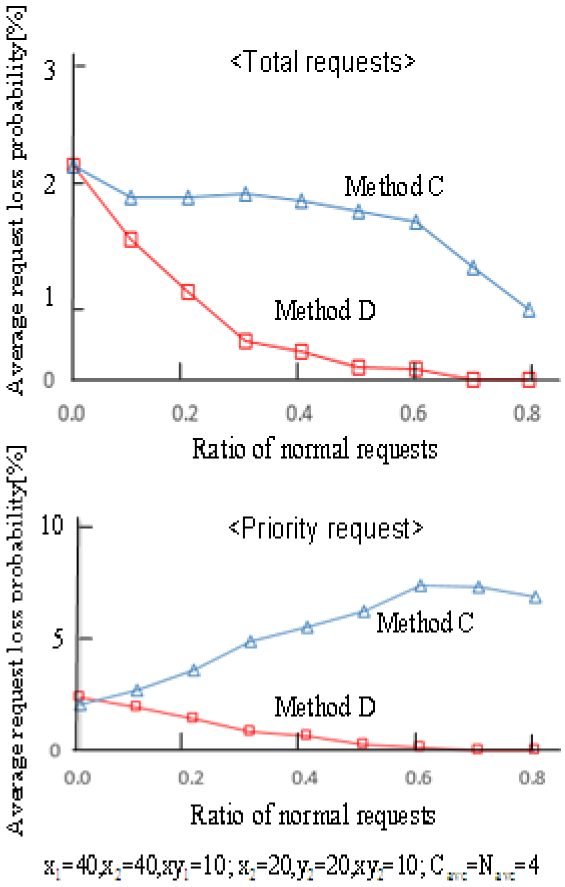


Figure 12. Comparison of average request loss probability between Method D and Method C

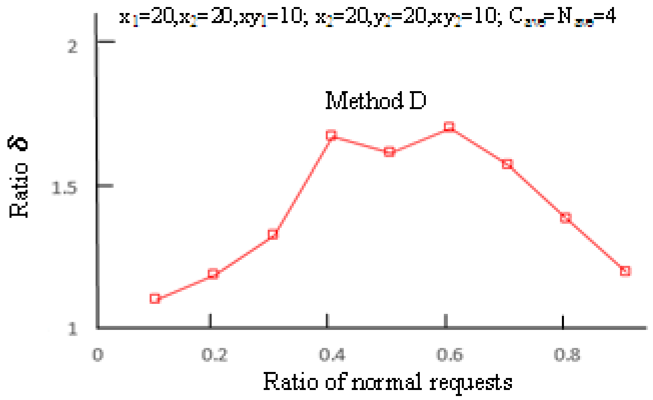


Figure 13.Comparison of rate of request occurrence between Method D and Method C

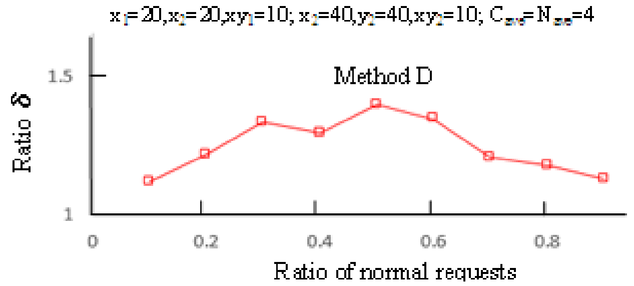


Figure 14. Comparison of rate of request occurrence between Method D and Method C

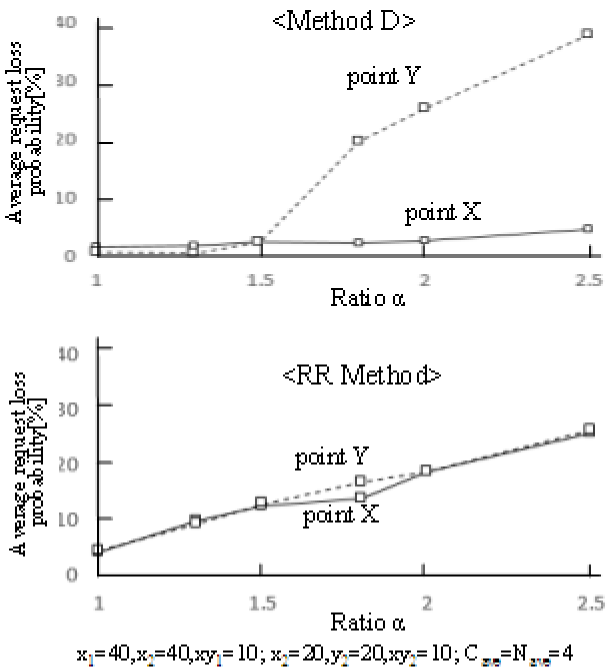


Figure 15. Impact of rate of request occurrence at point Y

Method D can handle up to more than twice as many requests as Method C. This is because Method D can make the request loss probability for priority requests lower than Method C (i.e.,resources are allocated well for priority requests). In Method C, the request loss probability becomes high when the ratio of normal requests is high, because normal requests tend to use the resources that priority requests can only use, instead of those that priority requests cannot use. Likewise, Method D can handle up to 1.5 times as many requests as the RR Method.

Except for cases where the ratio of normal requests is 20% or lower or alternatively 80% or higher, Method D remains superior to the other methods. In cases where the ratio of normal request is 20% or lower or alternatively 80% or higher, most requests are priority requests or alternatively most are normal requests. Therefore, the efficiency of resource usage is less dependent on the selection of centers.

Even when the amount of resources in each center is not determined in the way proposed in Section 3.2, Method D remains superior to Method C, although Method D becomes less superior to Method C as the amount of resources to priority requests is small.

Figure 15 compares Method D and RR method in terms of the impacts on the request loss probability in a case where requests from point Y occur α times more than expected. The horizontal axis and the vertical axis show ratio α and average request loss probability, respectively. It is clear that Method D can prevent an increase in the request loss probability from point X even when requests from point Y occur more than expected, although RR method cannot prevent.

5. CONCLUTIONS

This paper has proposed to enhance the existing joint multiple resource allocation method, Method B, so as to solve the issues of Method B. First, Method C was proposed to cope with the excessive generation of requests from specific access point. It was confirmed by simulation evaluations that that Method C can prevent the degradation in service quality of other request types even if specific requests occur more than expected. Next, Method D was proposed in order to provide the optimal resource allocation for services which requires to take account of total processing time (instead of network delay) in allocating resource, by enhancing Method C. It was demonstrated by simulation evaluations that Method D can serve up to twice as many requests as the existing methods (Methods B and D) with the same amount of resources and cope with the excessive generation of requests from the specific access point.

Since the model used for evaluation contained only limited numbers of access points and centers, it is required to evaluate the effectiveness of the proposed method and to identify the conditions under which the method are effective, assuming more access points and centers.

ACKNOWLEDGMENT

We would like to thank Mr. Yutaro Magome and Mr. Masatoshi Uriu for their help with the simulation.

REFERENCES

[1] G.Reese: “Cloud Application Architecture”, O’Reilly& Associates, Inc., Apr. 2009.

[2] J.W.Rittinghouse and J.F.Ransone: “Cloud Computing: Imprementation, 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] V.Vinothina, R.Sridaran, and P. Ganapathi, “A Survey on Resource Allocation Strategies in Cloud Computing”, International Journal of Advanced Computer Science and Applications,Vol. 3, No.6, 2012.

[5] 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.

[6] 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.

[7] 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.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ser. MCC’12. ACM,2012, pp. 13–16.

[9] I.Stojmenovic and S.Wen, "The Fog Computing Paradigm: Scenarios and Security Issues", Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 1–8.

[10] NTT press releases, “Announcing the Edge computing concept and the Edge accelerated Web platform prototype to improve response time of cloud applications," Jan. 2014. http://www.ntt.co.jp/news2014/1401e/140123a.html

[11] Y.Awano and S.Kuribayashi, “Proposed Joint Multiple Resource Allocation Method for Cloud Computing Services with Heterogeneous QoS”, Cloud Computing 2012, July 2012.

[12] S.Kuribayashi,“Joint Multiple Resource Allocation Method for Cloud Computing Services with different QoS to users at multiple locations”, International journal of Computer Networks & Communications (IJCNC), Vol.5, No.5, pp.1-18, Sep. 2013.

[13] 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.

[14] 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

[15] 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.

[16] 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.

[17] 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.

[18] G.Leey, B.G.Chunz, and R.H.Katz, “Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud”, HotCloud '11 June. 2011.

[19] 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.

[20] 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.

[21] 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).

[22] Y.Magome and S.Kuribayashi, “Resource allocation method for cloud computing environments with different service quality to users at multiple access points”, Proceeding of the 17-th International Conference on Network-Based Information Systems (NBiS-2014), Sep. 2014.

[23] M.Uriu and S.Kuribayashi, “Resource allocation method in cloud computing environments with multiple data centers over a wide area”, Proceeding of 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (Pacrim2015), C1-1, Aug. 2015.

AUTHOR

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.