PAPER Special Section on Network Virtualization, Network Softwarization and Fusion Platform of Computing and Networking

Prediction-Based Cloud Bursting Approach and Its Impact on Total Cost for Business-Critical Web Systems

Yukio OGAWA[†a)], **Go HASEGAWA**[††], ***Members**, **and*** **Masayuki MURATA**[††], ***Fellow***

SUMMARY *Cloud bursting* temporarily expands the capacity of a cloud-based service hosted in a private data center by renting public data center capacity when the demand for capacity spikes. To determine the optimal resources of a business-critical web system deployed over private and public data centers, this paper presents a cloud bursting approach based on long- and short-term predictions of requests to the system. In a private data center, a dedicated pool of virtual machines (VMs) is assigned to the web system on the basis of one-week predictions. Moreover, in both private and public data centers, VMs are activated on the basis of one-hour predictions. We formulate a problem that includes the total cost and response time constraints and conduct numerical simulations. The results indicate that our approach is tolerant of prediction errors and only slightly dependent on the processing power of a single VM. Even if the website receives bursty requests and one-hour predictions include a mean absolute percentage error (MAPE) of 0.2, the total cost decreases to half the existing cost of provisioning in the private date center alone. At the same time, 95% of response time is kept below 0.15 s.

key words: *cloud bursting, hybrid cloud, request prediction, total cost*

1. Introduction

Computing resources, e.g., physical and virtual servers, are assigned dedicatedly to an application system for achieving both high availability and desired performance without being affected by other systems. This type of resource allocation is practical for business-critical application systems in enterprise private data centers. Such systems are, however, generally built to handle peak workloads, which results in them being underutilized for most of the time [1]. An effective approach for maximizing the resource utilization to improve the cost efficiency of such existing systems is *cloud bursting* [2]. In this approach, an application system uses fixed resources in a private data center for the majority of its computing. The system further *bursts* into a public data center and temporarily combines on-demand resources when private resources are insufficient. We take this approach to provision virtual machines (VMs) for business-critical web systems. Our goal is to minimize the total cost of a computing platform while satisfying response time constraints. We thus focus on determining the right amount of VMs in both private and public data centers (i.e., in a hybrid cloud environment) in advance in order to adaptively adjust VMs to meet the current workloads.

Automatic extra resource allocation during increased demand and its termination when demand decreases is a major research topic in cloud environments [3]–[5]. Resource allocations are classified into two methods: reactive and proactive [6]. The reactive allocation method reconfigures an application system to meet the system's requirements for quality of service (QoS) after detecting changes in the workload, utilization, etc. of the system (e.g., [7]). If the requirements can be seriously violated from when the changes are detected to when the system reconfiguration is completed, the reconfiguration should be proactively triggered on the basis of estimating the future changes.

Studies on automating cloud bursting in a proactive manner are roughly divided into two categories on the basis of whether workload demand is known ahead of time or not. In the first category, the future workload is known in advance. This category includes high-performance computing for scientific applications, in which there is a trade-off between the completion time of the tasks and the amount of required resources. The number of tasks is known ahead of time and resources are adaptively scheduled for the tasks to meet deadlines [8]–[10]. In the second category, future workload is unknown. Accordingly, future demand must be estimated to adjust the trade-off between application constraints, such as response time and throughput, and computing resource economics, such as cost and configuration overhead, e.g., in the cases of enterprise applications [11], a video streaming service [12], and production systems [13]. Our target falls into the second category, in which an application platform is dynamically reconfigured to optimize the trade-off on the basis of predicting the demand for the application. Prediction errors thus can greatly affect the optimization. Although the previous studies [11]–[13] supposed different deployment scenarios and cost frameworks form ours, they, as well as autoscaling studies targeting a single cloud environment (e.g., [14], [15]), will give us a clue to consider the framework of our cost optimization problem. However, the impact of prediction accuracy on the total cost and its trade-off with application constraints has not been sufficiently discussed in the above studies. We thus focus on describing the impact for our deployment scenario.

A business-critical application system is often assigned a dedicated cluster of physical servers because availability of the system is determined at the cluster to which redundancy techniques for the VMs are applied [16]. We thus reallocate

Manuscript received December 12, 2016.
Manuscript revised March 18, 2017.
Manuscript publicized May 16, 2017.

[†] The author is with the Center for Multimedia Aided Education, Muroran Institute of Technology, Muroran-shi, 050-8585 Japan.

[††] The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

a) E-mail: y-ogawa@mmm.muroran-it.ac.jp

DOI: 10.1587/transcom.2016NNP0006

not only VMs but also physical servers in a private data center. A software-defined networking (SDN) framework make this reallocation feasible [17], although physical servers have a longer reallocation interval than VMs in practical deployment. We therefore present a two-step approach to adjust computing resources in a hybrid cloud environment: assigning physical servers in a private data center on the basis of a long-term (e.g., a week) prediction, and activating VMs in both private and public data centers on the basis of a short-term (e.g., an hour) prediction.

In this paper, we present a cloud bursting approach based on long- and short-term predictions for physical and virtual servers, respectively. Previously, we described a cost model of an application platform in a hybrid cloud environment [18]. We hence focus on evaluating the impact of prediction accuracy on our cloud bursting approach by using trace data of actual websites. The long-term prediction is, of course, not as accurate as the short-term prediction. Our main contributions are therefore to demonstrate that:

1. the prediction errors of the long-term physical server provisioning do not affect the optimized total cost much.
2. the prediction errors of the short-term VM allocation are handled by using the confidence interval for the prediction, and the allocation can enable the application system to satisfy response time constraints.

The rest of this paper is organized as follows. In Sect. 2, we introduce an operational procedure. In Sect. 3, we define a cost model. In Sect. 4, we describe a method for evaluations. Then, in Sect. 5, we evaluate our approach. Finally, in Sect. 6, we give conclusions.

2. Overview of a Cloud Busting Approach

We first give an overview of an application system in a hybrid cloud environment (called a hybrid cloud system) and explain our operational procedure. In accordance with the cost model of Weinman [19], we decrease fixed capacities to improve the utilization of application systems in a private data center and add on-demand resources in a public data center during the peak time. As shown in Fig. 1, in the private data center, a dedicated set of physical servers (i.e., a pool of VMs) is reallocated to an application system on the basis of long-term workload predictions every weekend; this interval is planned by considering long-term workload variations, as well as management aspects such as the recovery time and cost when this reallocation fails. Moreover, the amount of VMs required in the system is planned on the basis of short-term predictions every hour; this interval is set corresponding to the billing interval of the public data center. If the amount required is less than the amount available in the private data center, the minimum VMs alone are activated, and unnecessary VMs are powered off or put to sleep. In contrast, when the required number of VMs is more than the maximum number of VMs in the private data center at that time, the shortage of VMs is compensated for by additionally allocating on-demand VMs in the public data center.

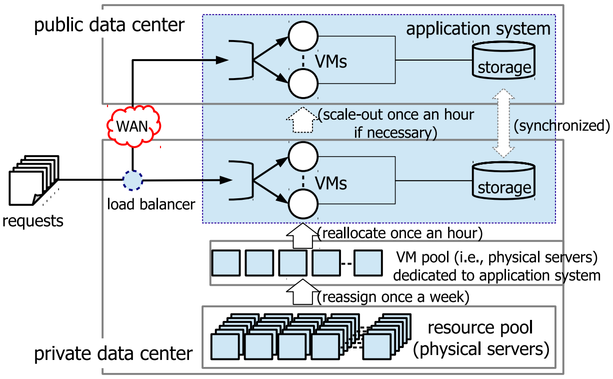

public data center
application system
VMs
storage
WAN
(scale-out once an hour if necessary)
(synchronized)
requests
load balancer
VMs
storage
(reallocate once an hour)
VM pool (i.e., physical servers) dedicated to application system
(reassign once a week)
resource pool (physical servers)
private data center

Fig. 1 Overview of cloud bursting approach.

In above deployment scenario, the total number of physical servers may be fixed during a time horizon (e.g., duration of renting the physical servers), and hence the private data center may supply a too large/small VM pool to each application system. Nevertheless, the amount of physical servers required by an application system is around the average capacity used by the system and fluctuations in the amount are much smaller than the peak-average difference. Moreover, there can be more than a hundred of application systems in an enterprise data center [20]. We therefore assume that the fluctuations in an application system are offset by those in other systems and thereby the total number of physical servers remains nearly constant in the time horizon.

We also note that, while we dynamically reconfigure computing resources, we fix the configurations of a wide-area network (WAN) between private and public data centers and local-area networks (LANs) in them. The WAN, which is managed by another service provider, can not be dynamically reconfigured from the private data center and thus should provide sufficient bandwidth to handle peak workloads of business-critical application systems. The LANs usually supply sufficient bandwidth as well, because the LANs are much cheaper than computing resources [21].

3. Model of a Hybrid Cloud System

In this section, we describe a cost model and response time constraints of a hybrid cloud system. We sometimes call VMs in private and public data centers private VMs and public VMs.

3.1 Cost Model

The VMs in both private and public data centers are controlled at fixed intervals called time slots, each of which is indexed by t ($t = 1, \cdots, T$). A hybrid cloud system has parameters that change with time slots t as depicted in Fig. 2 and summarized in Table 1. Here, the size (N_t) and processing rate (Λ_t) of a dedicated VM pool are altered at the end of each interval of w time slots.

Our objective is to minimize the total cost of an appli-

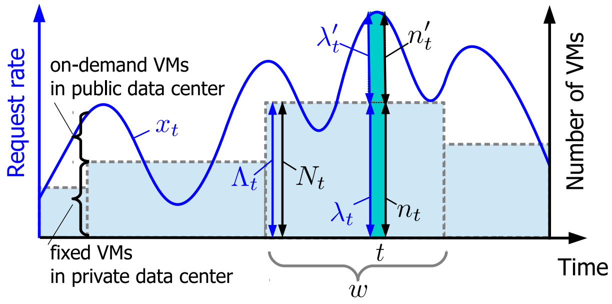


Fig. 2 Main parameters used for explaining cloud bursting approach.

Table 1 Parameters changing with time slot t.

x_t	Average rate of requests to application system
n_t, n'_t	Numbers of VMs allocated and turned on for the application system deployed over private and public data centers
λ_t, λ'_t	Average rates of requests sent to VMs in a private data center and a public data center, respectively ($\lambda_t + \lambda'_t = x_t$)
N_t, Λ_t	Capacity of the VM pool dedicated for the application system in the private data center and its average processing rate ($n_t \leq N_t, \lambda_t \leq \Lambda_t$))

Table 2 Constants for describing cost related to servers.

(a) Constants related to VMs assigned in private data center

c_{ps}	Cost of renting a physical server per time slot
n_{vm}	Number of VMs per physical server
c_{ec}	Energy charge rate
p_{ps}	Energy consumed per physical server
e	Energy-proportional coefficient [22]

(b) Constants related to VMs assigned in public data center

c_{vm}	Cost of an on-demand VM per time slot
c_{tr}	Cost of forwarding requests per unit size
d	Average amount of transferred data per request campus website and 4100 bytes for a consumer website)

(c) Constants for defining cost for operation and management

c_{st}	Personnel cost per time slot per staff member
n_{st}	Number of VMs managed by a staff member
α	Constant for specifying economics of scale ($\alpha \leq 1$) [23]

cation hosting platform, C, defined as the sum of the cost related to the fixed private VMs, F, that are relevant to the on-demand public VMs, U, and that for the operation and management, O, over a time horizon.

Objective: minimize

$$C = \sum_{t=1}^{T} \left(aF(N_t, n_t) + a'U(n'_t, \lambda'_t) + O(N_t, n'_t) \right), \quad (1)$$

where a is a constant for determining the total cost including all of the servers, networks, storages, etc. from the cost related to the servers alone in the private data center, and a' is that in the public data center.

First, the cost related to the private VMs is defined by using the constants in Table 2(a) as

$$F(N_t, n_t) = c_{\mathrm{ps}} \left\lceil \frac{N_t}{n_{\mathrm{vm}}} \right\rceil + c_{\mathrm{ec}} p_{\mathrm{ps}} \left((1-e) \left\lceil \frac{n_t}{n_{\mathrm{vm}}} \right\rceil + e \frac{n_t}{n_{\mathrm{vm}}} \right), \quad (2)$$

where, on the right side, the first term is the cost of renting $\left\lceil \frac{N_t}{n_{\mathrm{vm}}} \right\rceil$ physical servers. The second term is the cost for powering the physical servers [22], where $\left\lceil \frac{n_t}{n_{\mathrm{vm}}} \right\rceil$ physical servers are needed for allocating and turning on n_t VMs. Here, the second term assumes that each physical server has energy proportionality [24].

Second, we define the cost related to the public VMs by referring the constants in Table 2(b) as

$$U(n'_t, \lambda'_t) = c_{\mathrm{vm}} n'_t + c_{\mathrm{tr}} d \lambda'_t, \quad (3)$$

where, on the right side, the first term is the cost for using on-demand VMs, and the second term is the cost for transferring requests to/from the VMs and synchronizing data storages. Note that we do not count the cost for traversing a WAN, e.g., (virtual) dedicated network or the internet, between private and public data centers because we assume that the hybrid cloud system shares the WAN with other application systems and that the WAN is charged at a flat rate.

Finally, the cost for operation and management, i.e., the cost of the IT staff members who manage and operate the application hosting platform across private and public data centers, is defined by using the constants in Table 2(c) as

$$O(N_t, n'_t) = c_{\mathrm{st}} \left(\frac{1}{n_{\mathrm{st}}} (N_t + n'_t) \right)^{\alpha}, \quad (4)$$

where the IT staff members are prepared to support the sum of the maximum number of private VMs and the average number of public VMs. We also assume economics of scale [23].

3.2 Constraints on Response Time Performance

There is a trade-off between application latency and resource amount given to the application system. We thus pose constraints on response time: in both private and public data centers, qth percentiles of response time distribution for each time slot (r^q and $r^{q\prime}$) are not more than a threshold r_c. Here, q is the target probability. When we define the cumulative distribution function of response time (R defined in Sect. 3.4), the above relationship for the private data center is replaced with an alternative relationship: the probability determined by the number of private VMs (n_t), the request rate processed by these VMs (λ_t), and the threshold time (r_c) is not less than the target probability (q), as shown in Constraint (5). The same relationship is also given to the public data center by Constraint (6). Here, we add the notation ^ to the parameter of a predicted value.

Subject to:

$$r^q \leq r_c \quad \left(R(n_t, \hat{\lambda}_t, r_c) \geq \frac{q}{100} \right) \quad (\forall t) \quad (5)$$

$$r^{q\prime} \leq r_c \quad \left(R(n'_t, \hat{\lambda}'_t, r_c) \geq \frac{q}{100} \right) \quad (\forall t) \quad (6)$$

In these constraints, the numbers of private VMs (n_t) and public VMs (n'_t) are determined by using the predicted values of request rates ($\hat{\lambda}_t$ and $\hat{\lambda}'_t$). The actual qth percentiles (r^q and $r^{q\prime}$) can exceed r_c due to prediction errors.

3.3 Request Rate Prediction

We adopt the autoregressive integrated moving average (ARIMA) model [25] to predict the request rates. When defining the backward shift operator B by $Bx_t = x_{t-1}$, the original time series, x_t, is transformed into a stationary time series $y_t = (1-B)^d(1-B^s)^D x_t$ by applying the dth-order non-periodic differencing and the Dth-order periodic differencing. This y_t is then expressed as a function of its past values and/or past errors, as follows.

$$y_t = \sum_{i=1}^{p} \phi_i B^i y_t + \left(1 + \sum_{j=1}^{q} \theta_j B^j\right) \epsilon_t \tag{7}$$

where ϕ_i, θ_j are the parameters, and ϵ_t is the error term that follows $\epsilon_t \sim N(0, \sigma^2)$. The confidence interval of the one-time-slot-ahead prediction is the standard deviation of the errors (σ), which means $y_{t+1} \sim N(\hat{y}_{t+1}, \sigma^2)$. Moreover, when y_{t+h} is expressed as $y_{t+h} = \sum_{\tau=0}^{\infty} \psi_\tau \epsilon_{t+h-\tau}$ (where ψ_τ is the parameter calculated from the observed values and $\psi_0 = 1$), y_{t+h} follows $y_{t+h} \sim N(\hat{y}_{t+h}, \sigma^2 \sum_{\tau=0}^{h-1} \psi_\tau^2)$.

3.4 Estimation of Response Time Distribution

To introduce the response time constraints explained in Sect. 3.2, we define the cumulative distribution function of response time at time slot t by applying the M/M/m queuing model [26]. Since a web system is supposed to be implemented asynchronously so that it can respond quickly to a request without waiting for the request to be completed, we adopt the waiting time distribution, not the sojourn time distribution. Let r, r_0, and μ be the response time from the application system at t, a constant network latency, and average processing rate of requests per VM, respectively. The cumulative distribution function R is defined as

$$R(n_t, \lambda_t, r) = 1 - \pi(n_t, \lambda_t) e^{-(n_t\mu - \lambda_t)(r - r_0)} \quad (r \geq r_0), \tag{8}$$

where $\pi(n_t, \lambda_t)$ is the probability of requests to be queued at t. This probability is defined as

$$\pi(n_t, \lambda_t) = \frac{n_t \rho_t^{n_t}}{n_t!\,(n_t - \rho_t)} \left[\frac{n_t \rho_t^{n_t}}{n_t!\,(n_t - \rho_t)} + \sum_{l=0}^{n_t - 1} \frac{\rho_t^l}{l!} \right]^{-1}$$
$$\rho_t = \frac{\lambda_t}{\mu}. \tag{9}$$

Note that the above function is in the case of the private data center, but this function is also applied for the public data center. We also note that the network latency, r_0, corresponds the sums of latencies of a WAN and LANs, indeed. We regard r_0 as a constant, because we neglect the buffering delay and consider propagation and transmission delays in the networks which have sufficient bandwidth.

Algorithm 1 Resource allocation in hybrid cloud system

```
for each time slot $t$ ($t = 1, \cdots, T$) do
    if $t$ mod $w = 0$ then
        Predict $\{\hat{x}_{t+h} \mid h = 1, 2, \cdots, w\}$ according to Eq.(7).
        $N_{t+1} \leftarrow$ VMPoolSize($\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+w}$).
        The number of dedicated physical servers in the next week
            is given by $\left\lceil \frac{N_{t+1}}{n_{\mathrm{vm}}} \right\rceil$.
    end if
    Predict $\hat{x}_{t+1}$ according to Eq. (7).
    $\{n_{t+1}, n'_{t+1}, \lambda'_{t+1}\} \leftarrow$ VMAllocSize($\hat{x}_{t+1}, N_{t+1}$).
end for
```

Algorithm 2 Sizing of VM pool in private data center

```
function VMPoolSize($\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+w}$)
    $N_{t+1} \leftarrow 0$ and $\Lambda_{t+1} \leftarrow 0$.
    while $\Lambda_{t+1} < \max(\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+w})$ do
        Calculate $\Lambda_{t+1}$ so as to satisfy Constraint (5) with substituting
            $N_{t+1}$.
        for each $t + h$ ($h = 1, 2, \cdots, w$) do
            $\{n_{t+h}, n'_{t+h}, \lambda'_{t+h}\} \leftarrow$ VMAllocSize($\hat{x}_{t+h}, N_{t+1}$).
            Calculate $C_{t+h}$ in accordance with Objective (1).
        end for
        The cost of the VM pool in this interval (denoted by $C^*(N_{t+1})$)
            is given by $\sum_{h=1,2,\cdots,w} C_{t+h}$.
        $N_{t+1} \leftarrow N_{t+1} + 1$.
    end while
    return $N_{t+1} \leftarrow \arg\min_{N_{t+1}} C^*(N_{t+1})$.
end function
```

4. Method for Dynamically Allocating Resources

As explained in Sect. 2, we use long and short-term VM provisioning. The size of a VM pool in the private data center over the next w-time-slot interval ($\{N_{t+h} \mid h = 1, \cdots, w\}(N_{t+1} = \cdots = N_{t+w})$) is determined on the basis of the predictions from one-time-slot-ahead to w-time-slot ahead ($\{\hat{x}_{t+h} \mid h = 1, \cdots, w\}$) at the end of each w-time-slot interval, while the numbers of private and public VMs at the next time slot (n_{t+1} and n'_{t+1}) are recalculated by using the one-time-slot-ahead prediction $\hat{x}_{t+1}$ and N_{t+1} at each time slot, as given in Algorithm (1).

The VM pool size in the next interval (N_{t+1}) is determined so as to minimize Objective (1), which is counted up with increasing N_{t+1} from 0 (i.e., the case of a pure public data center) to more than the maximum of $\{\hat{x}_{t+h} \mid h = 1, \cdots, w\}$ (i.e., the case of a pure private data center), as described in Algorithm (2). Moreover, the numbers of private and public VMs in the next time slot (n_{t+1} and n'_{t+1}) are determined by comparing the processing ability of the VM pool in the private data center (Λ_{t+1}) with the predicted request rate to an application system ($\hat{x}_{t+1}$), as shown in Algorithm (3).

5. Evaluation

In this section, we evaluate the total cost and response time of web systems and analyze the effect of prediction errors on

Algorithm 3 Sizing of VMs allocated in both data centers

```
function VMAllocSize($\hat{x}_{t+h}$, $N_{t+1}$)
    Calculate $\Lambda_{t+1}$ so as to satisfy Constraint (5) with substituting $N_{t+1}$.
    if $\hat{x}_{t+h} \leq \Lambda_{t+1}$ then
        $\hat{\lambda}_{t+h} \leftarrow \hat{x}_{t+h}$.
        Calculate $n_{t+h}$ so as to satisfy Constraint (5) with substituting $\hat{\lambda}_{t+h}$.
        $\hat{\lambda}'_{t+h} \leftarrow 0$ and $n'_{t+h} \leftarrow 0$.
    else
        $\hat{\lambda}_{t+h} \leftarrow \Lambda_{t+1}$ and $n_{t+h} \leftarrow N_{t+1}$.
        $\hat{\lambda}'_{t+h} \leftarrow \hat{x}_{t+h} - \Lambda_{t+1}$.
        Calculate $n'_{t+h}$ so as to satisfy Constraint (6) with substituting $\hat{\lambda}'_{t+h}$.
    end if
    return $\{n_{t+h}, n'_{t+h}, \hat{\lambda}'_{t+h}\}$.
end function
```

them through numerical simulations based on trace data of actual web systems. In the evaluations, each time slot is set to one-hour long in accordance with the billing interval of a prominent public data center [27].

5.1 Simulation Settings

5.1.1 Datasets

We used the arrival traces collected from two web application systems.

- 5-month access log (from April 1 to August 26, 2014) for a campus website of a university with about 30,000 students and staff members (called a campus web).
- 2.5-month access log (from April 30 to July 16, 1998) for the 1998 FIFA World Cup website [28] (called a consumer web).

5.1.2 Cost Model

The description of physical servers in the private data center was given as follows. All physical servers, which have 8 CPU cores and a 32-GB memory each [29], were assumed to be used on a three-year lease. The price of a single physical server was set to ¥600,000. Thus, the cost of renting a physical server per time slot (c_{ps}) was ¥$600,000/(3 \times 365 \times 24)$ =¥22.8 per physical server per hour. Each physical server had up to 2 VMs (i.e., $n_{\text{vm}} = 2$). The energy charge rate (c_{ec}) was ¥16 per kWh [30], and the power consumption of a single physical server (p_{ps}) was set to 550 W. In addition, the energy-proportional coefficient (e) of the physical servers was set to 0.6 [22].

On the other hand, each public VM was assumed to be a m4.2xlarge instance at Amazon EC2 [27], which performed similarly to a single VM when $n_{\text{vm}} = 2$ in the private data center. The price of a single on-demand VM (c_{vm}) was \$0.732 per hour. The price for transferring data to/from the public data center (c_{tr}) was set to \$0/\$0.14 per GB, respectively, as well [27]. We converted dollars into yen at an exchange rate of ¥120 to \$1. In addition, the average amount of transferred data per request (d) was 7800 bytes for a campus website and 4100 bytes for a consumer website; these values were calculated on the basis of the figures recorded in the trace data.

Moreover, the personnel cost for operation and management (c_{st}) was set to ¥900,000 per month $/(30 \times 24) =$ ¥1250 per time slot per staff member. A single staff member is assumed to be able to operate and manage up to 100 VMs (i.e., $n_{\text{st}} = 100$) [21]. Furthermore, the constant for specifying economics of scale (α) was set to 0.6 [23].

The processing rate of each private and public VM (μ) was set to 5.5 requests/s for the campus web and 275 requests/s for the consumer web. The μ of the consumer web was set so that the maximum number of VMs for the consumer web was similar to that for the campus web. In addition, the cost related to the servers was assumed to be 50% of the total cost including all of the devices and equipments in a private data center [21]. The cost related to VMs was similarly supposed to be 80% of the total cost in a public data center [27]. We thereby set $a = 2$ and $a' = 1.25$ in Objective (1).

5.1.3 Response Time Constraints

In Constraints (5) and (6), target probability q was defined as 95%. The threshold of response time (r_c) was set to 0.15 s because users can notice the response time when the response delay exceeds this threshold [31]. Furthermore, the sums of latency of a WAN and LANs (r_0 in Eq. (8)) were set to 0.001 s for the private data center and 0.14 s for the public data center.

5.1.4 Request Rate Prediction

Based on the observation of the trace data, the datasets had weekly, i.e., $24 \times 7 = 168$ time slots, periodicity. We convert the time series into a logarithmic scale for counteracting the effect of the rapid increase and decrease. We then applied the transformation of $y_t = (1-B)(1-B^{168}) \log_{10} x_t$ in order to make the original time series x_t stationary. At each time slot, we extracted the last three weeks, i.e., $24 \times 21 = 504$ time slots, of data and identified the values of p and q of $ARIMA(p, 1, q)$ by changing p $(0 \leq p \leq 5)$ and q $(0 \leq q \leq 5)$ until no lower AIC (Akaike Information Criterion) could be found [25].

5.2 Evaluation Results

5.2.1 Prediction Error of Request Rate

We performed the allocation process shown in Algorithm (1) 48 times by changing the starting time slot of a time horizon. Figure 3 shows an example of bursty requests and their predictions. Figure 4 shows the prediction accuracy, where we analyzed the mean absolute percentage error (MAPE) defined as $\frac{1}{T}\sum_{t=1}^{T}\frac{|x_t-\hat{x}_t|}{x_t}$. For the 168-time-slot, i.e., one-week, predictions, the campus web showed relatively small

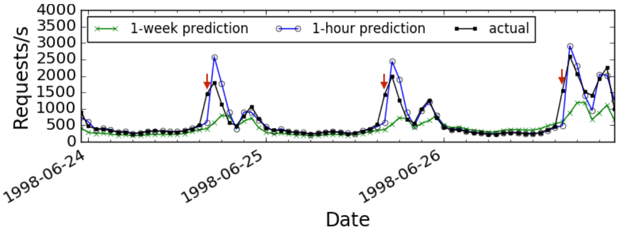


Fig. 3 Example of bursty requests and predictions (for consumer web).

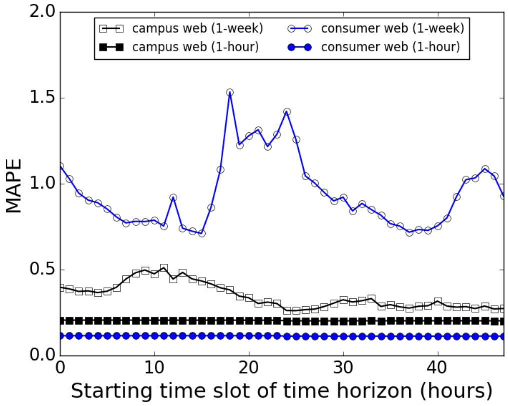


Fig. 4 Prediction errors.

error (0.34 on average) because it had regular predictable patterns, while the consumer web showed a large error (0.94 on average) because it sometimes received unexpected request spikes. In contrast, the one-time-slot-ahead, i.e., one-hour, predictions indicated relatively small errors in both webs (0.2 and 0.1 on average).

5.2.2 Sizing of VM Pool in Private Data Center

In the private data center, the dedicated VM pool was resized once a week to minimize Objective (1). For example, Fig. 5 shows the Objective (1) value of a certain week as a function of the size of the private VM pool (N_t) in the case of the consumer web, where the Objective (1) value is expressed in terms of the cost relative to that when the application is deployed by using an existing provisioning approach (denoted by C_{existing}). This C_{existing} is calculated for when the system is assigned private VMs able to handle the maximum request rate of the time horizon and all VMs always stay active in the private data center. In Fig. 5, the consumer web was processed in the private data center alone when N_t was 11 and in the public data center alone when N_t was 0.

The cost associated with the physical servers in the private data center decreased linearly with N_t, where most of the cost went to rent the physical servers under our evaluation settings. The cost of energy consumed by the physical servers did not significantly change when N_t was large because the number of active VMs was almost the same in that case. On the other hand, the cost for using on-demand VMs in the public data center and that for transferring data from them increased rapidly when N_t was close to 0 because the number of VMs in the public data center greatly expanded at that time. Furthermore, the cost for VM operation and management was reduced nearly linearly with N_t. As shown in Fig. 5, Objective (1) was minimized when N_t was 2 and was almost unchanged until $N_t = 4$. In this case, $\lceil \frac{2}{2} \rceil$(= 1) physical server was reassigned to the consumer web in the next week.

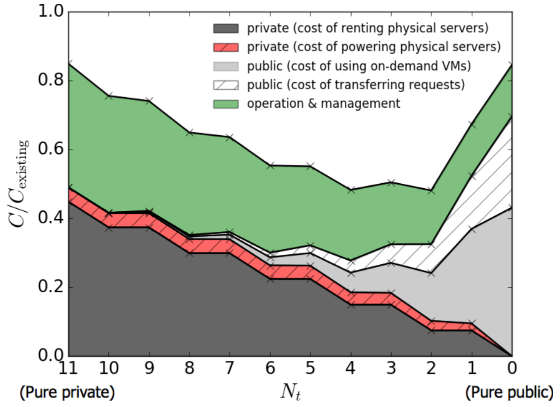


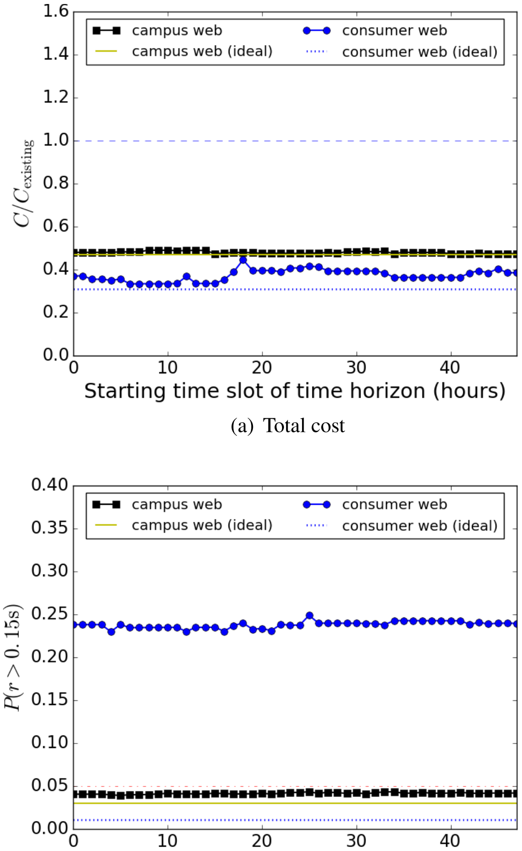
Fig. 5 Example of total cost as function of N_t (consumer web, $n_{\text{vm}} = 2$).

5.2.3 Total Cost and Response Time

Figure 6(a) indicates the evaluation results of the total cost, which is expressed as the ratio of the optimized one (C) to the existing one (C_{existing}). Figure 6(b) further shows those of the response ratio of more than the threshold r_c (0.15 s). Each *ideal* assumes a case in which the future requests are known a priori.

For the campus web, the relative total cost corresponded to its ideal, and the response ratio of more than the threshold r_c (0.15 s) was totally below the (transformed) target probability of 0.05 (= 1 - q (0.95)) because both one-hour and one-week predictions had high accuracy. The response ratios were almost the same for the starting time slots because the ratio depended on the number of active VMs. Each number was determined on the basis of the corresponding one-hour ahead prediction having the same accuracy (see Fig. 4).

For the consumer web, the total cost was slightly larger than its ideal, while the response ratio of more than 0.15 s was much more than the target of 0.05 and reached 0.23. The slight difference in the total cost was mainly caused by errors in the one-week predictions. In this case, these errors shifted to the positive side, resulting in an over-provisioned VM pool in the private data center. Furthermore, this optimized cost was lower than that for the campus web, which meant that the consumer web was less utilized than the campus web when both webs ran in the existing manner. On the other hand, the response time was degraded by errors of the one-hour predictions. Although the consumer web had small MAPEs in the one-hour predictions, it sometimes received



(a) Total cost

(b) Response time

Fig. 6 Evaluation results when $n_{\text{vm}} = 2$.

bursty requests exceeding estimated values of the one-hour predictions (see arrows in Fig. 3); these errors made VMs under-provisioned, resulting in delaying the response time. In addition, the ideal case of the consumer web was much less than the target probability of 0.05 because the processing rate of a single VM for the web was relatively large, which absorbed a certain level of request rate variation.

5.2.4 Handling of One-Hour Prediction Errors

Since the VMs are allocated and activated/deactivated on the basis of the point estimates for future request rates, estimation errors sometimes made VMs under-provisioned, resulting in degrading the response time as in the case of the consumer web explained in the previous section. To prevent this response delay, we thus use the upper bound of the interval estimate instead of the point estimate. Figure 7 shows the total cost and the response ratio as functions of the upper bounds of the confidence interval for the one-hour predictions in the case of the consumer web. The error bars indicate the maximum and minimum of the 48 trials. Here, we still used the point estimates for the one-week predictions. Note that we provide two additional cases when $n_{\text{vm}} = 4$ and $n_{\text{vm}} = 8$ in Fig. 7, which will be explained in Sect. 5.2.6.

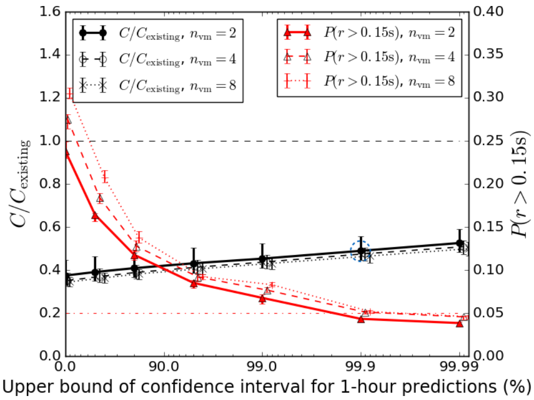


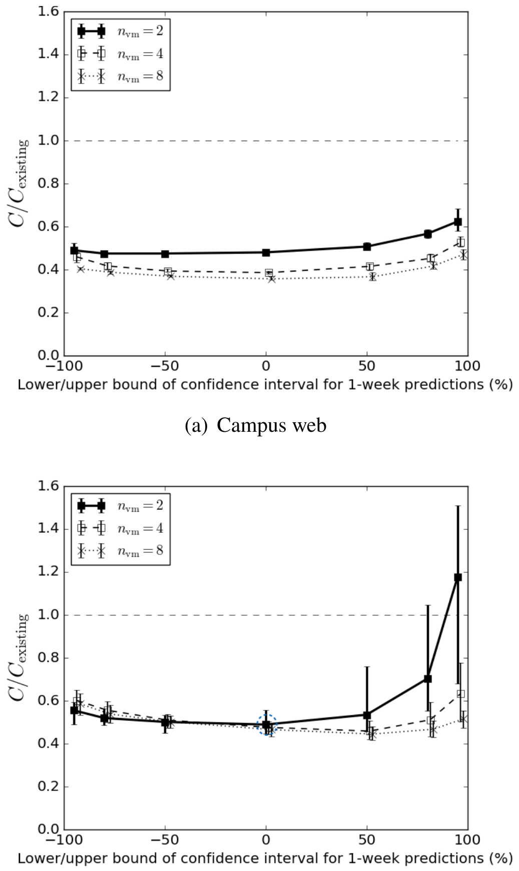
Fig. 7 One-hour error handling for consumer web.

Figure 7 indicates a trade-off between the total cost and the response-time performance. When we provisioned with the upper bound of a 99.9% confidence interval, the response ratio was below the target probability (0.05) and the total cost increased but still remained half that of C_{existing}. The errors of the one-hour predictions were relatively small, which suppressed the increase of the total cost. Figure 7 also shows that the response ratio did not decrease below about 0.04 even when we applied the upper bound of a more than 99.99 % confidence interval; this reveals the limitation of the prediction-based provisioning using the fixed length (i.e., 1 hour) for time slots in our evaluation environment. We also note that the selection of a confidence level determines how strictly the response time constraints are enforced, and this strictness depends on the requirements of a corresponding application system.

5.2.5 Impact of One-Week Prediction Errors on Total Cost

Errors of one-week predictions change the size of the VM pool in the private data center, which can impact the total cost. On the x-axis of Fig. 8, positive and negative values mean the upper and lower bounds of the confidence interval for the one-week predictions. Zero on the x-axis means that the point estimates are applied. For example, 50% on the x-axis means that the private VM pool is over-provisioned when the upper bound of a 50% confidence interval is used. In contrast, -50% means that the pool is under-provisioned when the lower bound of a 50% confidence interval is used. We evaluated up to a 95% confidence interval for the one-week predictions. Here, to make the response ratio less than the target probability, for the one-hour predictions, the point estimate was applied to the campus web and the upper bound of a 99.9% confidence interval was applied to the consumer web. Note that 2 additional cases when $n_{\text{vm}} = 4$ and $n_{\text{vm}} = 8$ in Fig. 8 will be also discussed in Sect. 5.2.6.

Figure 8 reveals that the underestimate of the size of the private VM pool had little effect on the total cost. This was also true for the overestimate, until we used the upper bound of a 50% confidence interval when $n_{\text{vm}} = 2$ for the consumer web (Fig. 8(b)). Although the one-week prediction values,



(a) Campus web

(b) Consumer web (dotted circle corresponds that in Fig. 7)

Fig. 8 Impact of one-week prediction error.

of course, included larger errors especially for the consumer web, the total cost was tolerant of the prediction errors for the following reasons. When we predicted the request rates, we converted them into a logarithmic scale. Owing to this, lower bounds of the confidence interval had smaller fluctuations than upper bounds. Furthermore, the total cost stayed at an equilibrium while N_t was in the range of up to two from the value making the cost optimal (see around $N_t = 3$ in Fig. 5). These advantages come from the VM pool in the private data center being provisioned for the average rate of requests, not for the maximum rate.

5.2.6 Effect of the Processing Power of a Single VM

From Sect. 5.2.2 to Section 5.2.5, while mentioning the evaluation results when all physical servers in the private data center had up to 2 VMs (i.e., $n_{\mathrm{vm}} = 2$) each and all VMs in the public data center performed similarly, we discussed how our provisioning approach is tolerant of prediction errors. When a single VM, however, has a large processing power, it can absorb a somewhat high level of request fluctuation; this may lead to the error tolerance. We therefore additionally examined cases when the VMs have less processing capacity.

Figure 7 shows the effect of a single VM processing power on one-hour prediction errors, the resulting total cost and response delay. Note that horizontal positions of the plots mentioned in Sect. 5.2.6 have been adjusted to keep the markers visible. When all physical servers had up to 4 VMs ($n_{\mathrm{vm}} = 4$) and even 8 VMs ($n_{\mathrm{vm}} = 8$) each in the private data center and VMs of corresponding performances (such as m4.xlarge ($0.366 per hour) and m4.large ($0.183 per hour) instances at Amazon EC2 [27]) were provided in the public data center, the total cost slightly became smaller and the response ratio above 0.15 s appeared larger. This was because a single VM of smaller processing capacity had less surplus capacity; this cut the total cost and enabled the VM itself to afford lower request fluctuations caused by one-hour prediction errors as well.

Figure 8 furthermore indicates the effect of a single VM capacity on one-week prediction errors and the resulting total cost. In the case of the campus web shown in Fig. 8(a), the total cost decreased as the processing capacity of a single VM shrank from $n_{\mathrm{vm}} = 2$ to $n_{\mathrm{vm}} = 8$, for the same reason as in the case of the one-hour error handling above. In this case, the total cost was little affected by the VM pool over- and under-provisioned when the upper and lower bounds of the confidence interval were used, respectively, because the campus web showed relatively small errors for one-week predictions. Moreover, in the case of the consumer web given in Fig. 8(b), when we used the lower bound of a 95% confidence interval, the total cost was slightly increased as the processing capacity became small. This was because the smaller VMs in the under-provisioned private VM pool had little redundant capacity and needed more on-demand public VMs of relatively high cost. On the other hand, when we used the upper bound of a 95% confidence interval, the total cost was significantly increased as the processing capacity enlarged from $n_{\mathrm{vm}} = 8$ to $n_{\mathrm{vm}} = 2$ for the following reasons. The request rates was converted into a logarithmic scale for being predicted and the consumer web had larger errors for one-week predictions shown in Fig. 4. These made the private VM pool of the consumer web much over-provisioned when the upper bound of a 95% confidence interval was used. The over-provisioned private VM pool had the largest redundant capacity when the processing capacity of a single VM is the biggest (i.e., $n_{\mathrm{vm}} = 2$); this greatly increased the total cost.

As explained above, the total cost and the response ratio was affected by the processing capacity of a single VM. However, the total cost could still be half the existing cost and 95% of response time was less than 0.15 s, even if the predicted request ratio contained somewhat large errors.

6. Conclusion

This paper presented a cloud bursting approach in which we assign a dedicated VM pool for a business-critical web system in a private data center on the basis of one-week predictions and determine which VMs in private and public data centers should be active on the basis of one-hour predictions. We evaluated how prediction errors affect the approach through numerical simulations based on trace data of actual web systems. To avoid the response delay caused by the one-hour prediction errors, we needed to apply the upper bound of a wider confidence interval, resulting increasing the total cost of the system. However, when the upper bound of a 99.9% confidence interval was used for the one-hour predictions to keep 95% of response time less than 0.15 s, the total cost was still half the existing cost, i.e, the cost when VMs are provided to handle the maximum request rate and all VMs always stay active in the private data center alone. Furthermore, the total cost was nearly unchanged when the VM pool in the private data center was under- or over-provisioned for the one-week predictions. These characteristics of our approach, furthermore, are only slightly dependent on the processing capacity of a single VM.

The length of time slots used for predicting request rates and handling VMs was fixed to one hour, which may limit the prediction accuracy of request rates and the resulting distribution of response time as well. Moreover, dynamic adjustment of the confidence level is beyond the scope of this paper. A topic of future study is therefore improving the provisioning accuracy by shortening the time slots and adjusting dynamically the confidence level, with using more trace data for further analysis of this study.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," Technical Report, UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb. 2009.

[2] J. Barr, "Cloudbursting - hybrid application hosting," https://aws.amazon.com/jp/blogs/aws/cloudbursting/, Aug. 2008. accessed Dec. 9, 2016.

[3] A.R. Hummaida, N.W. Paton, and R. Sakellariou, "Adaptation in cloud resource configuration: A survey," J. Cloud Comput., vol.5, no.1, pp.57:1–57:16, Dec. 2016.

[4] T. Lorido-Botran, J. Miguel-Alonso, and J. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," J. Grid Comput., vol.12, no.4, pp.559–592, Dec. 2014.

[5] H. Alipour, Y. Liu, and A. Hamou-Lhadj, "Analyzing auto-scaling issues in cloud environments," Proc. 24th Annual International Conference on Computer Science and Software Engineering, pp.75–89, Nov. 2014.

[6] R.N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," IEEE Trans. Cloud Comput., vol.3, no.4, pp.449–458, Oct 2015.

[7] Y. Niu, B. Luo, F. Liu, J. Liu, and B. Li, "When hybrid cloud meets flash crowd: Towards cost-effective service provisioning," Proc. 2015 IEEE Conference on Computer Communications, pp.1044–1052, April 2015.

[8] H.Y. Chu and Y. Simmhan, "Cost-efficient and resilient job lifecycle management on hybrid clouds," Proc. IEEE 28th International Parallel and Distributed Processing Symposium, pp.327–336, May 2014.

[9] M. HoseinyFarahabady, H. Samani, L. Leslie, Y.C. Lee, and A. Zomaya, "Handling uncertainty: Pareto-efficient bot scheduling on hybrid clouds," Proc. 42nd International Conference on Parallel Processing, pp.419–428, Oct. 2013.

[10] S. Imai, T. Chestna, and C.A. Varela, "Accurate resource prediction for hybrid IaaS clouds using workload-tailored elastic compute units," Proc. IEEE/ACM 6th International Conference on Utility and Cloud Computing, pp.171–178, Dec. 2013.

[11] T. Guo, U. Sharma, P. Shenoy, T. Wood, and S. Sahu, "Cost-aware cloud bursting for enterprise applications," ACM Trans. Internet Technol., vol.13, no.3, pp.10:1–10:24, May 2014.

[12] H. Zhang, G. Jiang, K. Yoshihira, and H. Chen, "Proactive workload management in hybrid cloud computing," IEEE Trans. Netw. Serv. Manage., vol.11, no.1, pp.90–100, March 2014.

[13] M. Bjorkqvist, L. Chen, and W. Binder, "Cost-driven service provisioning in hybrid clouds," Proc. 5th IEEE International Conference on Service-Oriented Computing and Applications, pp.1–8, Dec. 2012.

[14] G. Sun, Z. Lu, J. Wu, X. Wang, and P. Hung, "A novel reactive-predictive hybrid resource provision method in cloud datacenter," pp.33–47, Lecture Notes in Computer Science, vol.9464, pp.33–47, Springer Nature, 2015.

[15] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," Proc. 2011 IEEE 4th International Conference on Cloud Computing, pp.500–507, July 2011.

[16] VMware, Inc., "vSphere and vSphere with Operations Management." http://www.vmware.com/products/vsphere.html. accessed Dec. 9, 2016.

[17] C. Dixon, D. Olshefski, V. Jain, C. DeCusatis, W. Felter, J. Carter, M. Banikazemi, V. Mann, J.M. Tracey, and R. Recio, "Software defined networking to support the software defined environment," IBM J. Res. & Dev., vol.58, no.2/3, pp.3:1–3:14, March 2014.

[18] Y. Ogawa, G. Hasegawa, and M. Murata, "Cloud bursting approach based on predicting requests for business-critical web systems," Proc. International Conference on Computing, Networking and Communications, pp.443–447, Jan. 2017.

[19] J. Weinman, "Hybrid cloud economics," IEEE Cloud Comput., vol.3, no.1, pp.18–22, Jan. 2016.

[20] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," Proc. 16th International Conference on World Wide Web, pp.331–340, May 2007.

[21] A. Greenberg, J. Hamilton, D.A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," SIGCOMM Comput. Communi. Rev., vol.39, pp.68–73, Dec. 2008.

[22] D. Wong and M. Annavaram, "KnightShift: Scaling the energy proportionality wall through server-level heterogeneity," Proc. 45th Annual IEEE/ACM International Symposium on Microarchitecture, pp.119–130, Dec. 2012.

[23] F.T. Moore, "Economies of scale: Some statistical evidence," The Quarterly Journal of Economics, vol.73, no.2, pp.232–245, May 1959.

[24] L.A. Barroso and U. Hölzle, "The case for energy-proportional computing," IEEE Computer, vol.40, no.12, pp.33–37, Dec. 2007.

[25] R.J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice." https://www.otexts.org/book/fpp. accessed Dec. 9, 2016.

[26] R. Jain, The Art Of Computer Systems Performance Analysis, John Wiley & Sons, April 1991.

[27] Amazon.com, Inc., "Amazon Elastic Compute Cloud (EC2)." http://aws.amazon.com/ec2/. accessed Dec. 9, 2016.

[28] The Internet Traffic Archive, "1998 World Cup Web Site Access Logs." http://ita.ee.lbl.gov/html/contrib/WorldCup.html. accessed Dec. 9, 2014.

[29] Dell Inc., "PowerEdge R430." http://www.dell.com/jp/business/p/poweredge-r430/pd. accessed Dec. 9, 2016.

[30] Tokyo Electric Power Company Holdings, Inc. http://www.tepco.co.jp/ep/corporate/plan_h/index-j.html. accessed Dec. 9, 2016.

[31] J.D. McCabe, Network Analysis, Architecture and Design, Second Edition, Morgan Kaufmann Publishers, San Francisco, 2003.

Yukio Ogawa received his M.S. degree in Science from Nagoya University, Japan, in 1994, and Ph.D. degree in Information Science and Technology from Osaka University, Japan, in 2012. He joined the Hitachi Central Research Laboratory in Japan, in 1994. He is currently an associate professor at the Center for Multimedia Aided Education, Muroran Institute of Technology. His research interests include network architectures for cloud systems. He is a member of IEEE, ACM and IEICE.

Go Hasegawa received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 1997 and 2000, respectively. From July 1997 to June 2000, he was a Research Assistant in the Graduate School of Economics, Osaka University. He is now an Associate Professor at the Cybermedia Center, Osaka University. His research is in the area of transport architecture for future high-speed networks and overlay networks. He is a member of IEEE and IEICE.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Cybermedia Center, Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than nine hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM and IEICE.