Cloud Bursting Approach Based on Predicting Requests for Business-Critical Web Systems

Yukio Ogawa
Center for Multimedia Aided Education
Muroran Institute of Technology
Muroran, Hokkaido 050-8585 Japan
Email: y-ogawa@mmm.muroran-it.ac.jp

Go Hasegawa
Cybermedia Center
Osaka University
Toyonaka, Osaka 560-0043 Japan
Email: hasegawa@cmc.osaka-u.ac.jp

Masayuki Murata
Graduate school of Information Science
and Technology, Osaka University
Suita, Osaka 565-0871, Japan
Email: murata@ist.osaka-u.ac.jp

***Abstract*—*Cloud bursting* temporarily expands the capacity of cloud-based service hosted in a private data center by renting public data center capacity when the demand for capacity spikes. This paper presents a cloud bursting approach based on long- and short-term predictions of requests to a business-critical web system to determine the optimal resources of the system deployed over private and public data centers. 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 formulated a problem that includes the total cost and response time constraints and conducted numerical simulations. The results indicate that our approach is tolerant of prediction errors. 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 a half the current cost while 95% of response time is kept below 0.15 s.**

***Index Terms*—cloud bursting, hybrid cloud, request prediction**

I. Introduction

Business-critical application systems in private data centers are generally built to handle peak workloads, resulting in them being underutilized most of the time. An effective approach for maximizing the resource utilization to improve the cost efficiency of such existing systems is *cloud bursting* [1]. In this approach, an application system uses fixed resources in a private data center for the majority of its computing and *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.

Research for automating cloud bursting is roughly divided into two categories on the basis of whether workload demand is known ahead of time. In the first category, the number of tasks is known in advance, and there is a trade-off between the completion time of the tasks and the amount of required resources. Researchers have proposed solutions to adaptively schedule resources to meet deadlines; this category includes high-performance computing for scientific applications [2], [3]. In the second category, future workload is unknown. Accordingly, it is necessary to estimate future demand and to optimally 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 [4], a video streaming service [5], and production systems [6]. 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. This issue, however, has not been sufficiently discussed in previous studies.

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 [7]. We thus reallocate not only VMs but also physical servers in a private data center. This is made feasible by using a software-defined networking (SDN) framework [8], although physical servers have a longer reallocation interval than VMs in practical deployment. We therefore propose 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 prediction for physical and virtual servers, respectively. 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 error of the long-term physical server provisioning does not affect the total cost much and (2) the short-term VM allocation can enable the application system to satisfy response time constraints. Toward this end, we describe a cost model of an application platform in a hybrid cloud environment and evaluate our approach by using trace data of actual websites.

The rest of this paper is organized as follows. In Sect. II, we introduce an operational procedure. In Sect. III, we define a cost model. In Sect. IV, we describe a method for evaluations. Then, in Sect. V, we evaluate our approach. Finally, in Sect. VI, we give conclusions.

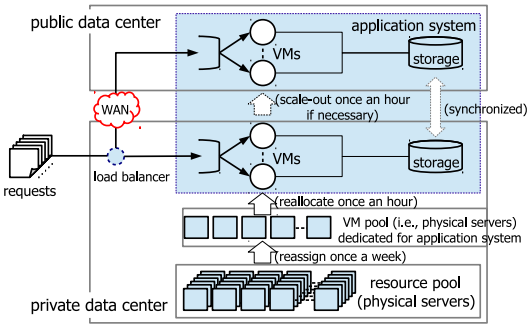

Fig. 1. Overview of cloud bursting approach

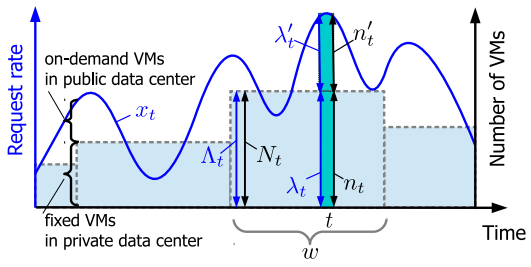

Fig. 2. Main parameters used for cloud bursting approach

II. Overview of Cloud Bursting Approach

In accordance with the cost model of Weinman [9], 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. 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.

III. Model of Hybrid Cloud System

In this section, we describe a cost model and response time constraints of an application system in a hybrid cloud environment (called a hybrid cloud system). We call VMs in private and public data centers private VMs and public VMs.

A. 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 I. 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.

TABLE I
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 maximum average processing rate ($n_t \le N_t, \lambda_t \le \Lambda_t$))

TABLE II
Constants for describing total cost of application platform

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

$c_{\rm ps}$	Cost of renting a physical server per time slot (￥22.8 [10])
$n_{\rm vm}$	Number of VMs per physical server (2)
$c_{\rm ec}$	Energy charge rate (￥16/kWh [11])
$p_{\rm ps}$	Energy consumed per physical server (550W [10])
e	Energy-proportional coefficient [12] (0.6 [12])

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

$c_{\rm vm}$	Cost of an on-demand VM per time slot ($0.732/hour [13])
$c_{\rm tr}$	Cost of forwarding requests per unit size ($0.14/GB [13])
d	Average amount of transferred data per request (7800 bytes for a campus website and 4100 bytes for a consumer website)

(c) Constants for defining the cost for operation and management

$c_{\rm st}$	Personnel cost per VM per time slot (￥1250/h)
$n_{\rm st}$	Number of VMs managed by a staff member (100 [14])
α	Constant for specifying economics of scale ($\alpha \le 1$) [15] (0.6 [15])

*Values in brackets are used in evaluations in Sect. V.

Our objective is to minimize the total cost of an application hosting platform, C, defined as the sum of the cost related to the fixed private VMs, F, that related to the on-demand public VMs, U, and that for the operation and management, O, over a time horizon.

Objective: minimize

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

where a is a constant for estimating the total cost including the networks, storage, etc. from the cost related to the servers in the private data center, and a' is that in the public data center. We summarize the constants in Table II to detail Objective (1).

First, the cost related to the private VMs is defined as

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

where, on the right side, the first term is the cost of renting $\left\lceil \frac{N_t}{n_{\rm vm}} \right\rceil$ physical servers. The second term is the cost for powering the physical servers [12], where $\left\lceil \frac{n_t}{n_{\rm vm}} \right\rceil$ physical servers are needed for allocating and turning on n_t VMs.

Second, we define the cost related to the public VMs as

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

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 wide-area network (WAN), i.e., 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 is

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

where the management 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 [15].

B. 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. III-D), 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 Constraints (5). The same relationship is also given to the public data center by Constraints (6). Here, we add the notation $\hat{}$ to the parameter of a predicted value.

Subject to:

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

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

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.

C. Request Rate Prediction

We adopt the ARIMA model [16] 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.

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

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 $y_{t+h} \sim N(\hat{y}_{t+h}, \sigma^2 \sum_{\tau=0}^{h-1} \psi_\tau^2)$.

D. Estimation of Response Time Distribution

We define the cumulative distribution function of response time at time slot t by applying the M/M/m queuing model [17]. 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

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

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

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

$$\rho_t = \frac{\lambda_t}{\mu}. \quad (9)$$

This function is also applied for the public data center.

IV. Method for Resource Allocation

As explained in Sect. II, we use long- and short-term VM provisioning. At the end of each w-time-slot interval, 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 so as to minimize Objective(1), which is counted up by using n_{t+h} and $n'_{t+h} (h = 1, 2, \cdots, w)$ calculated on the basis of the w-time-slot predictions of request rate $\{\hat{x}_{t+h} \mid h = 1, \cdots, w\}$. Moreover, at each time slot, 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} determined above.

V. Evaluation

We evaluate the total cost and response time of a web system and analyze the affect of prediction errors on them. In the evaluations, each time slot is set to one-hour long.

A. Simulation Settings

1) Datasets: We used the arrival traces collected from two actual 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 World Cup website [18] (called a consumer web).

2) Cost model: The parameter settings for constants are noted in brackets in Table II. All physical servers in the private data center (which have 8 CPU cores and a 32-GB memory [10] each) were assumed to be used on a three-year lease. A single server price was set to ¥600,000, resulting in $c_{\rm ps} = ¥600,000/(3 \times 365 \times 24) = ¥22.8$ per physical server per hour. On the other hand, each public VM was assumed to be a m4.2xlarge instance at Amazon EC2 [13]. 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. Note that the above 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. Moreover, the cost for operation and management ($c_{\rm st}$) was set to ¥900,000 per month $/(30 \times 24)$ = ¥1250 per hour per staff member. In addition, $a = 2$ [14] and $a' = 1.25$ [13] in Objective (1). We convert dollars into yen at an exchange rate of ¥120 to $1.

3) Response Time Constraints: In Constraints (5) and (6), target probability q was defined as 95%, and the threshold of response time (r_c) was set to 0.15 s [19]. Furthermore, the sums of latency of a WAN and a data center network (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.

4) Request Rate Prediction: Based on the observation of the trace data, we had weekly, i.e., $24 \times 7 = 168$ time slots, periodicity in the datasets. To make the original time series x_t stationary, we applied the transformation of $y_t = (1-B)(1-B^{168}) \log_{10} x_t$. We convert the time series into a logarithmic scale for counteracting the effect of the rapid increase and decrease. 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 $ARIMA(p, 1, q)$ [16].

B. Evaluation Results

1) Prediction Error of Request Rate: We performed the allocation process 48 times by changing the starting time slot of a time horizon. Fig. 3 shows an example of bursty requests and their predictions. Fig. 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-slots, i.e., one-week, predictions, the campus web showed relatively small 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).

2) Sizing of VM Pool in Private Data Center: 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 a current provisioning approach (denoted by $C_{current}$). This $C_{current}$ 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. As shown in Fig. 5, Objective (1) was minimized when N_t was 2 and almost unchanged till $N_t = 4$.

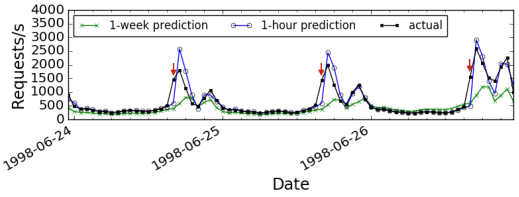

Fig. 3. Example of bursty requests and predictions (for consumer web)

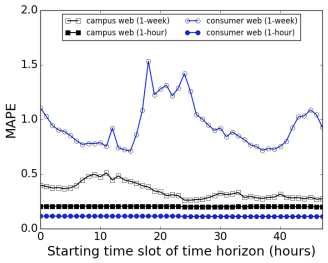

Fig. 4. Prediction errors

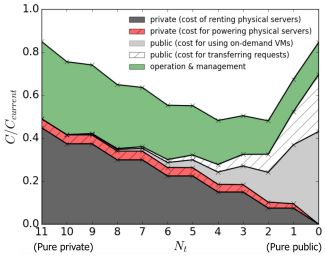

Fig. 5. Total cost as function of N_t

3) Total Cost and Response Time: Fig. 6 indicates the evaluation results of the total cost, which is expressed as the ratio of the optimized one (C) to the current one ($C_{current}$). 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.

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 0.05 and reached 0.23. The slight difference in the total cost was mainly caused by errors of the one-week predictions. In this case, these errors shifted in the positive side, resulting in an over-provisioned VM pool in the private data center. On the other hand, the response time was degraded by errors of the one-hour predictions. The consumer web sometimes received 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. To prevent this delay, we thus use the upper bound of the interval estimate instead of the point estimate. Fig. 7 shows the total cost and the response ratio as functions of the upper bounds of the confidence interval for the one-hour predictions. The error bar indicates the maximum and minimum of the 48 trials. Here, we still used the point estimates for the one-week predictions. Fig. 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 $C_{current}$. The errors of the one-hour predictions were relatively small, which suppressed the

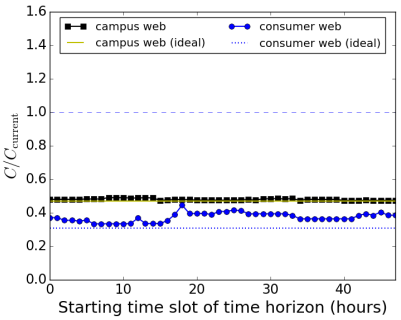

Fig. 6. Evaluations of total cost

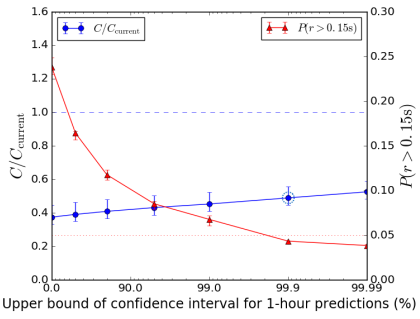

Fig. 7. One-hour error handling for consumer web

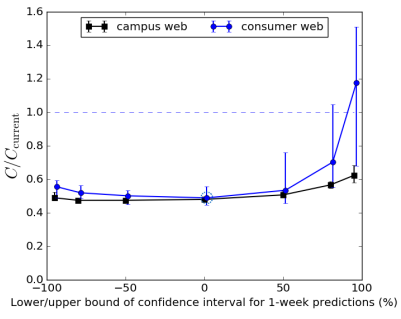

Fig. 8. Impact of one-week prediction error

increase of the total cost.

4) Impact of One-Week Prediction Errors on 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. E.g., 50 and -50 on the x-axis mean that the private VM pool is over- and under-provisioned with using the upper and lower bounds of a 50% confidence interval, respectively. 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. Fig. 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 up to the upper bound of a 50% confidence interval. Although the one-week prediction values, of course, included larger errors, 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.

VI. Conclusion

This paper presented a cloud bursting approach in which we assign a dedicated VM pool for a system in a private data center on the basis of one-week predictions and determine the active VM in private and public data centers on the basis of one-hour predictions. Prediction errors become large, particularly for bursty requests. However, when the upper bound of a 99.9% confidence interval was used for the one-hour predictions to satisfy the response time constraints, the total cost was still half the current cost. 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. Finally, the length of time slots and the performance of a single VM were fixed, which may impose limitations on this study. A future topic is therefore to improve by controlling variable time slots and various VM instances.

References

[1] J. Barr, "Cloudbursting hybrid application hosting," https://aws.amazon.com/jp/blogs/aws/cloudbursting/, Aug. 2008.

[2] H.-Y. Chu and Y. Simmhan, "Cost-efficient and resilient job life-cycle management on hybrid clouds," in *Proc. of 2014 IEEE 28th IPDPS*, May 2014, pp. 327–336.

[3] M. HoseinyFarahabady, H. Samani, L. Leslie, Y. C. Lee, and A. Zomaya, "Handling uncertainty: Pareto-efficient bot scheduling on hybrid clouds," in *Proc. of 2013 42nd ICPP*, Oct. 2013, pp. 419–428.

[4] 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.

[5] 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, Mar. 2014.

[6] M. Bjorkqvist, L. Chen, and W. Binder, "Cost-driven service provisioning in hybrid clouds," in *Proc. of 2012 5th IEEE SOCA*, Dec. 2012, pp. 1–8.

[7] VMware, Inc., "vsphere and vsphere with operations management," http://www.vmware.com/products/vsphere/features/availability.html, 2016, accessed June 12, 2016.

[8] 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, Mar. 2014.

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

[10] Dell Inc., "PowerEdge R430," http://www.dell.com/jp/business/p/poweredge-r430/pd, accessed May 10, 2015.

[11] Tokyo Electric Power Company Holdings, Inc., http://www.tepco.co.jp/e-rates/corporate/charge/charge07-j.html, accessed May 10, 2015.

[12] D. Wong and M. Annavaram, "Knightshift: Scaling the energy proportionality wall through server-level heterogeneity," in *Proc. of 2012 45th IEEE/ACM MICRO*, Dec. 2012, pp. 119–130.

[13] Amazon.com, Inc., "Amazon Elastic Compute Cloud (EC2)," http://aws.amazon.com/ec2/, accessed May 10, 2015.

[14] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Computer Communication Review*, vol. 39, pp. 68–73, Dec. 2008.

[15] F. T. Moore, "Economies of scale: Some statistical evidence," *The Quarterly Journal of Economics*, vol. 73(2), pp. 232–245, May 1959.

[16] R. J. Hyndman and G. Athanasopoulos, "Forecasting: principles and practice," https://www.otexts.org/book/fpp, accessed May 10, 2015.

[17] R. Jain, *The Art Of Computer Systems Performance Analysis*. John Wiley & Sons, Apr. 1991.

[18] The Internet Traffic Archive, "1998 world cup web site access logs," http://ita.ee.lbl.gov/html/contrib/WorldCup.html, accessed Nov. 19, 2014.

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