ICNC 2017

Jan 28, 2017

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

Yukio Ogawa
Muroran Institute of Technology,
Hokkaido, Japan

Go Hasegawa and Masayuki Murata
Osaka University,
Osaka, Japan

Contents

- Goal and research objective
- Overview of cloud bursting approach
- Model of a hybrid cloud system
- Evaluation results, conclusion

Background and goal:
Cost Efficiency in hybrid cloud systems

- In private DCs, business-critical application systems are build to handle peak workloads for achieving high performance.

 - Application systems are underutilized most of the time.

- An approach for maximizing utilization to improve cost efficiency is *Cloud bursting.*

 - It deceases fixed capacity in a private DC and adds on-demand resources in a public DC during peak time.

- Our goal is to minimize the total cost of a computing platform while satisfying response time constraints.

DC: Data Center, VM: Virtual Machine

Objective of this study:
Prediction-based approach needs to be validated

- In our target system, future workload is unknown.

  - We need to predict future demand for provisioning optimal computing resources in advance.

- Prediction-based approach have already been discussed in the cases of enterprise applications [1], a video streaming service[2], and production systems[3].

  - Prediction errors can greatly affect the optimal provisioning.
  - We should perform further analysis of the effect of prediction errors on cloud bursting.

[1] T.Guo,U.Sharma,P.Shenoy,T.Wood,andS.Sahu,"Cost-awarecloud bursting for enterprise applications," ACM Trans. Internet Technol., vol. 13, no. 3, pp. 10:1–10:24, May 2014.
[2] 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.
[3] M. Bjorkqvist, L. Chen, and W. Binder, "Cost-driven service provision- ing in hybrid clouds," in Proc. of 2012 5th IEEE SOCA, Dec. 2012, pp. 1–8.

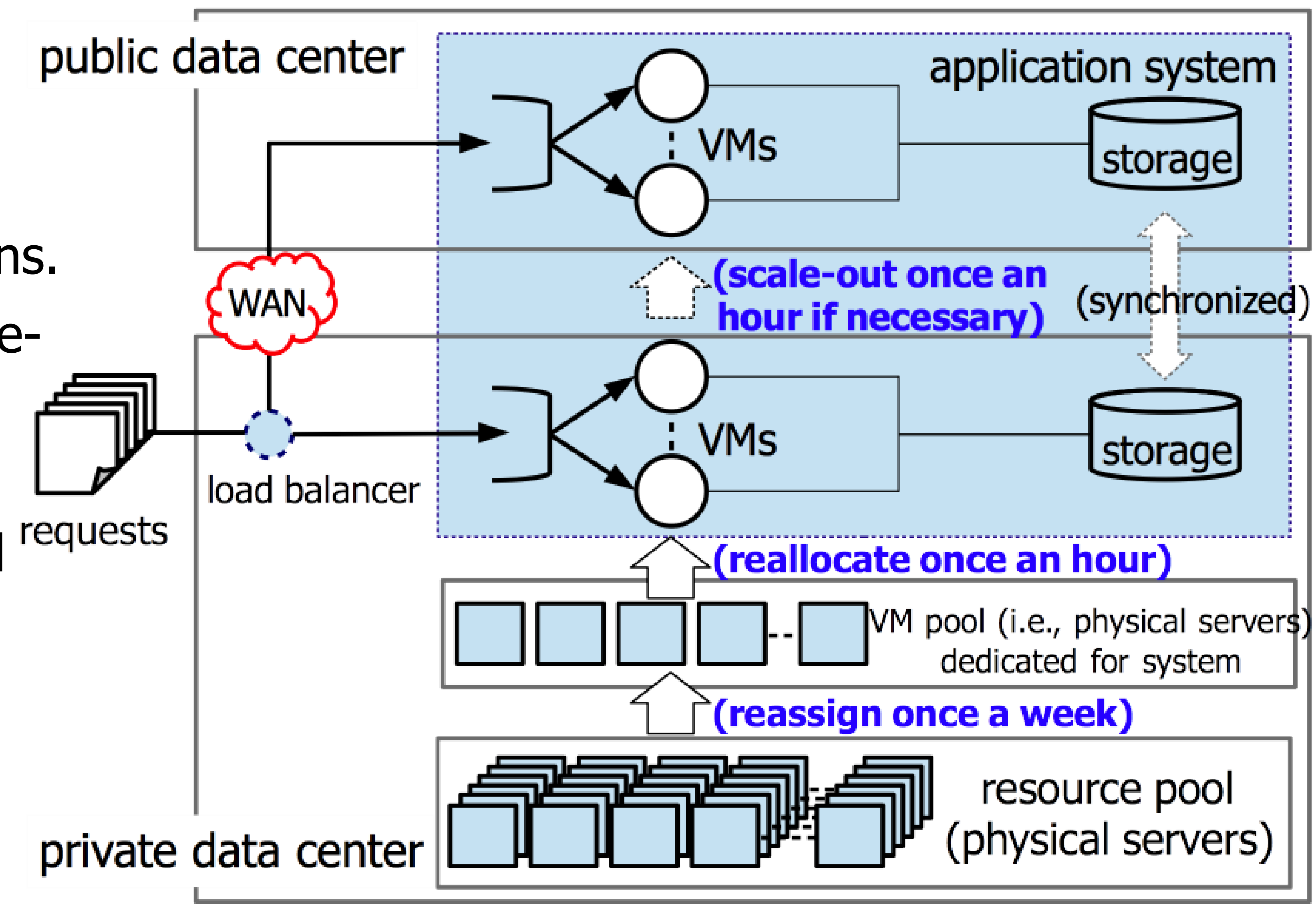
Overview of cloud bursting approach

- A business-critical system is assigned a dedicated cluster of physical servers.
- Physical servers have a longer reallocation interval than VMs.

  - We propose a two-step provisioning.

1. Assign physical servers (i.e., a pool of VMs) in a private DC on the basis of one-week predictions.
2. Activate VMs on the basis of one-hour predictions.

   - in private DC
   - additionally allocate on-demand VMs in public DC if necessary.

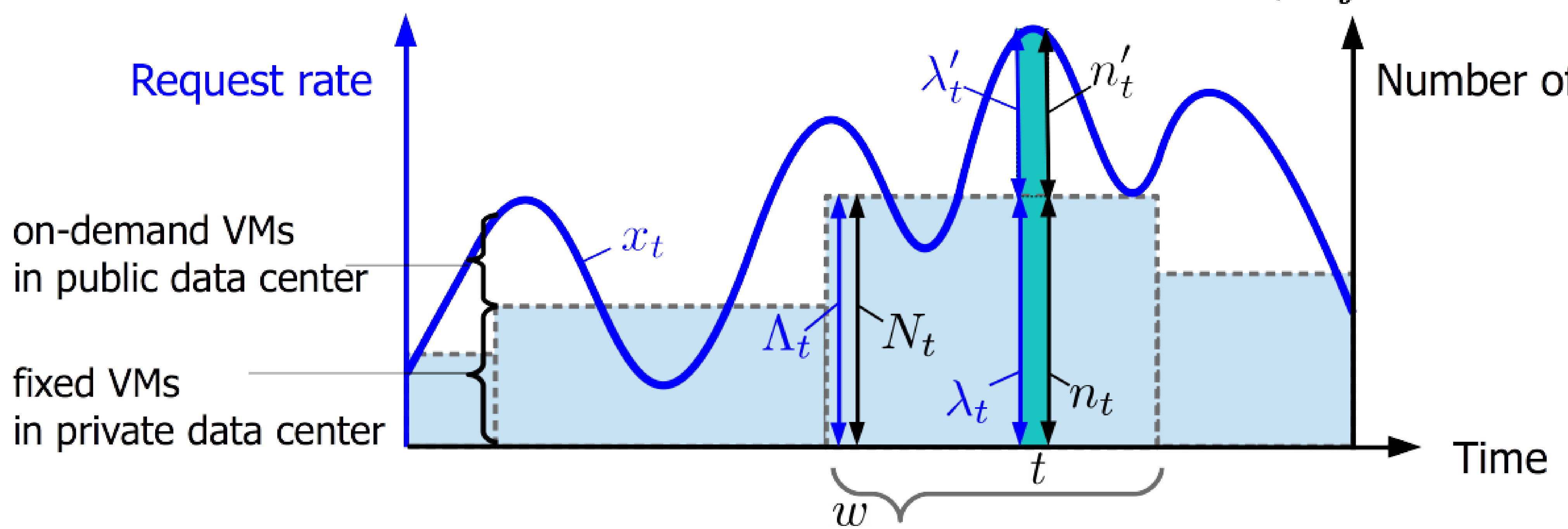


DC: Data Center, VM: Virtual Machine

Model of a hybrid cloud system:

Objective: minimizing total cost

- The size of a VM pool in private DC (N_t) is controlled at every *w*-time slots.
- The numbers of VMs in the private and public DCs (n_t, n'_t) are determined at every time slots by using the request rate for each DC (λ_t, λ'_t), respectively.



- Our objective is to minimize the total cost of an application hosting platform.

objective: minimize $C = \sum_{t=1}^{T} \left(\underbrace{aF(N_t, n_t)}_{\text{private VM cost}} + \underbrace{a'U(n'_t, \lambda'_t)}_{\text{public VM cost}} + \underbrace{O(N_t, n'_t)}_{\text{management cost}} \right),$

Model of a hybrid cloud system:
Detail of cost model

- Cost related to private VMs:

$$F(N_t, n_t) = \underbrace{c_{\mathrm{ps}} \left\lceil \frac{N_t}{n_{\mathrm{vm}}} \right\rceil}_{\text{renting physical servers}} + \underbrace{c_{\mathrm{ec}} p_{\mathrm{ps}} \left(\underbrace{(1-e) \left\lceil \frac{n_t}{n_{\mathrm{vm}}} \right\rceil}_{\text{(base)}} + \underbrace{e \frac{n_t}{n_{\mathrm{vm}}}}_{\text{(proportional to the number of active VMs)}} \right)}_{\text{powering physical servers}}$$

- Cost related to public VMs:

$$U(n'_t, \lambda'_t) = \underbrace{c_{\mathrm{vm}} n'_t}_{\text{using public VMs}} + \underbrace{c_{\mathrm{tr}} d \lambda'_t}_{\text{transferring request/response data to public VMs}}$$

- Cost for operation and management:

$$O(N_t, n'_t) = c_{\mathrm{st}} \left(\frac{1}{n_{\mathrm{st}}} \underbrace{(N_t + n'_t)}_{\text{capacity of fixed private VM pool and on-demand public VMs}} \right)^{\alpha}$$

Model of a hybrid cloud system:
Constraint: keeping response time

- Trade-off between application latency and resource amount.

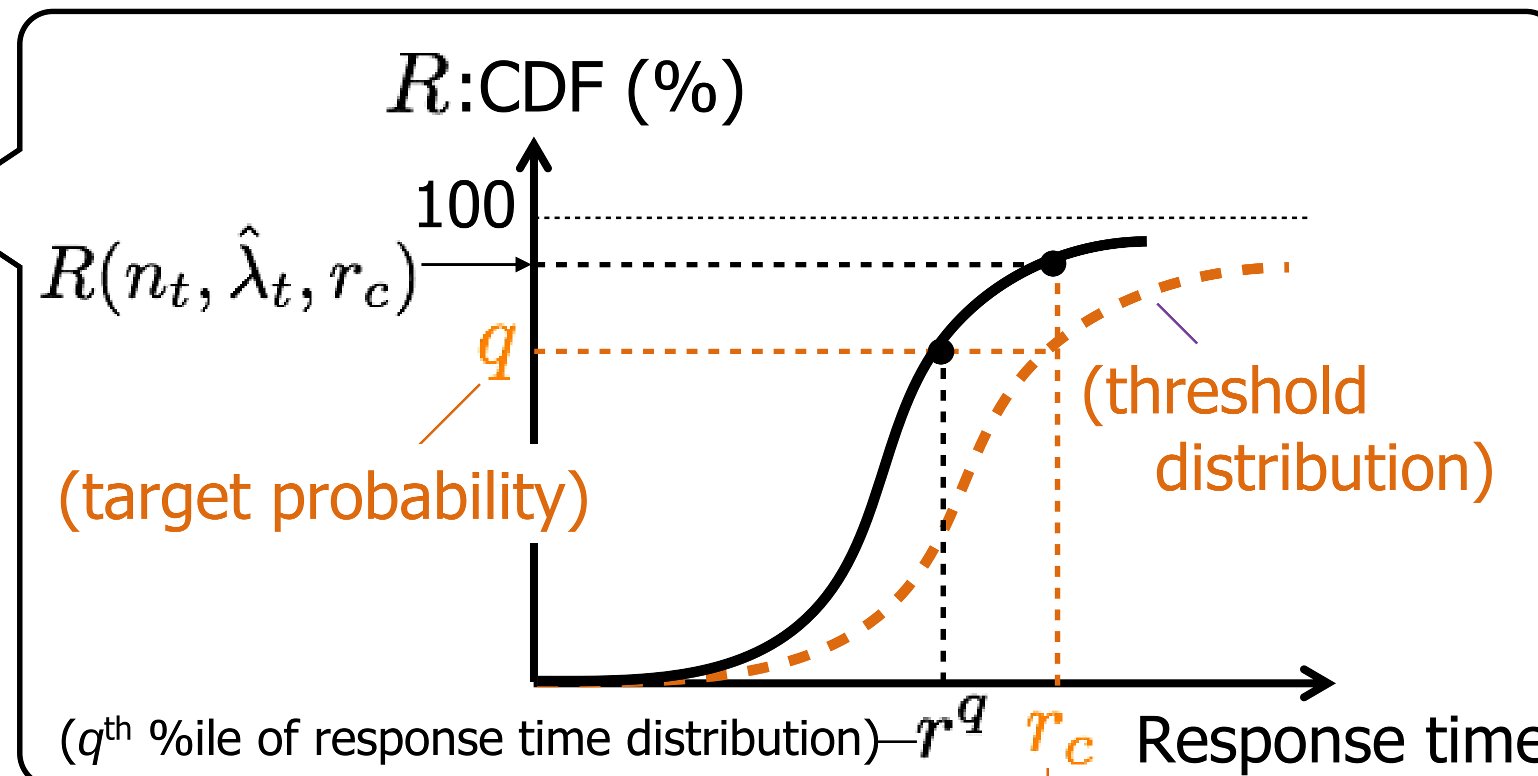
- We pose constraints on response time for both private and public DCs.

subject to:

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

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

(predicted request rate)



- R is defined by following M/M/m queuing model.
- **Constraints are applied by using predicted request rates ($\hat{\lambda}_t, \hat{\lambda}_t^\prime$).**

- **Actual response time ($r^q, r^{q\prime}$) can exceed r_c due to prediction errors.**

CDF: Cumulative Distribution Function

Model of a hybrid cloud system:
Request rate prediction

- Adopting the ARIMA model to predict request rates.

- Backward shift operator B by $Bx_t = x_{t-1}$

- Stationary time series by differencing $y_t = (1-B)^d(1-B^s)^D x_t$

$$y_t = \underbrace{\sum_{i=1}^{p} \phi_i B^i y_t}_{p\text{th-order autoregressive process}} + \underbrace{(1 + \sum_{j=1}^{q} \theta_j B^j)\epsilon_t}_{q\text{th-order moving average process}}$$

- one-time-slot-ahead prediction: $\hat{y}_{t+1}$
- *h*-time-slot-ahead prediction: $\hat{y}_{t+h}$

- Error term: $\epsilon_t \sim N(0, \sigma^2)$

- Confidence interval of one-time-slot-ahead prediction: $y_{t+1} \sim N(\hat{y}_{t+1}, \sigma^2)$

- Confidence interval of *h*-time-slot-ahead prediction:

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

ARIMA: AutoRegressive Integrated Moving Average

Method for Resource Allocation

- At the end of each w-time-slot interval,

- Predict the request rates over next w-time-slot interval ($\{\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+w}\}$) line 3
- Determine the size of a VM pool in private DC (N_{t+1}) over the next w-time-slot interval line 4

- At each time slot,

- Predict the request rate of next time slot ($\hat{x}_{t+1}$) line 7
- Recalculate the numbers of private and public VMs at the next time slot (n_{t+1}, n'_{t+1}) line 8

Algorithm 1 Resource allocation in hybrid cloud system

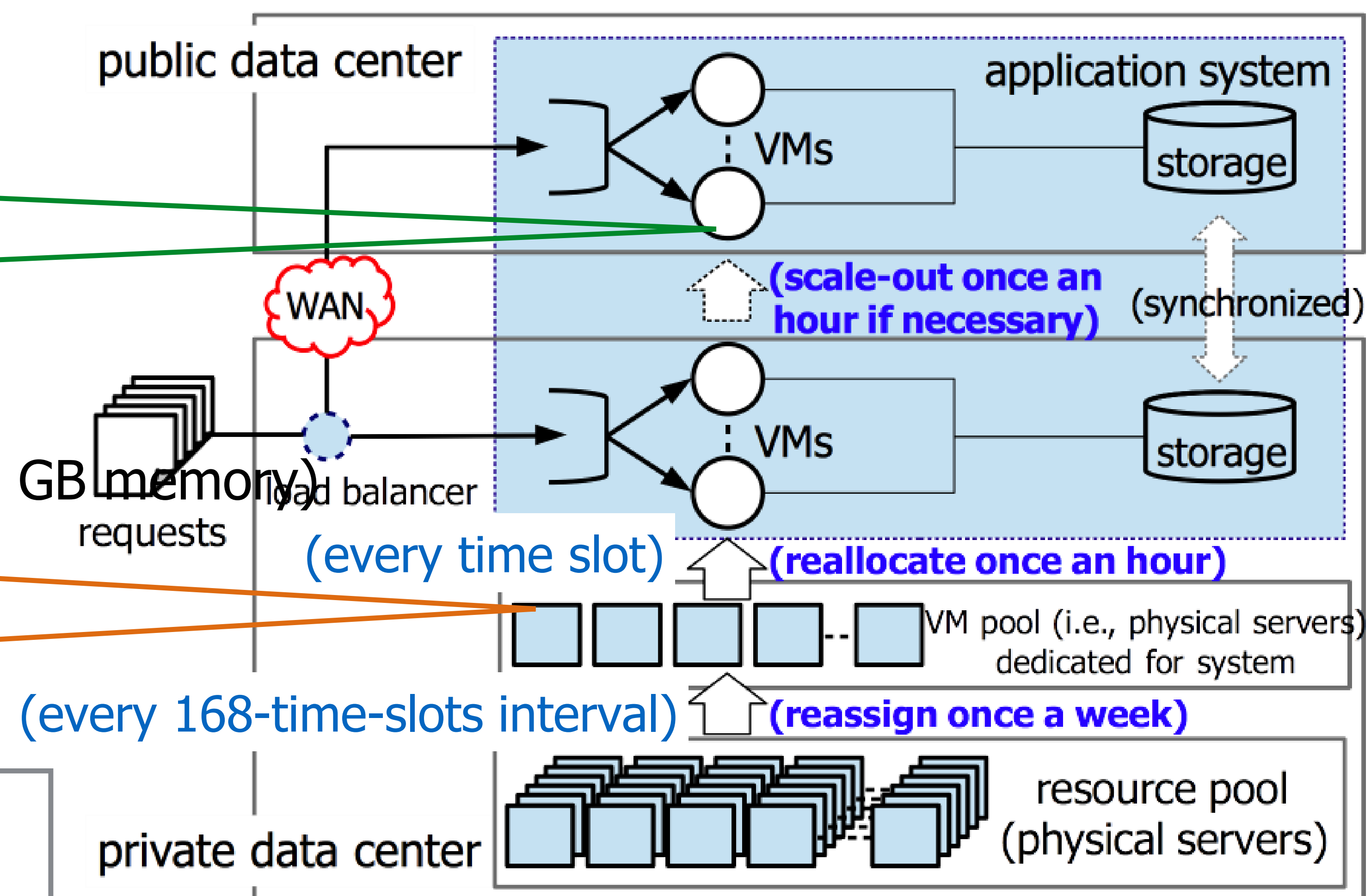
1: **for each** time slot t ($t = 1, \cdots, T$) **do**
2:     **if** $t \bmod w = 0$ **then**
3:         Predict $\{\hat{x}_{t+h} \mid h = 1, 2, \cdots, w\}$ according to Eq.(7).
4:         $N_{t+1} \leftarrow \textsc{VMPoolSize}(\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+w})$.
5:         The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

Evaluation:
Simulation settings

- Datasets- arrival traces collected from two actual web systems:
  - **Campus web**: 5-month access log for a campus website of a university
  - **Consumer web**: 2.5-month access log for the 1998 World Cup website[1]

- Public DC
  - m4.2xlarge instance at Amazon EC2

- Private DC
  - Dell PowerEdge R430(8 CPU cores, 32 GB memory)
  - 3-year lease
  - 2 VMs per server



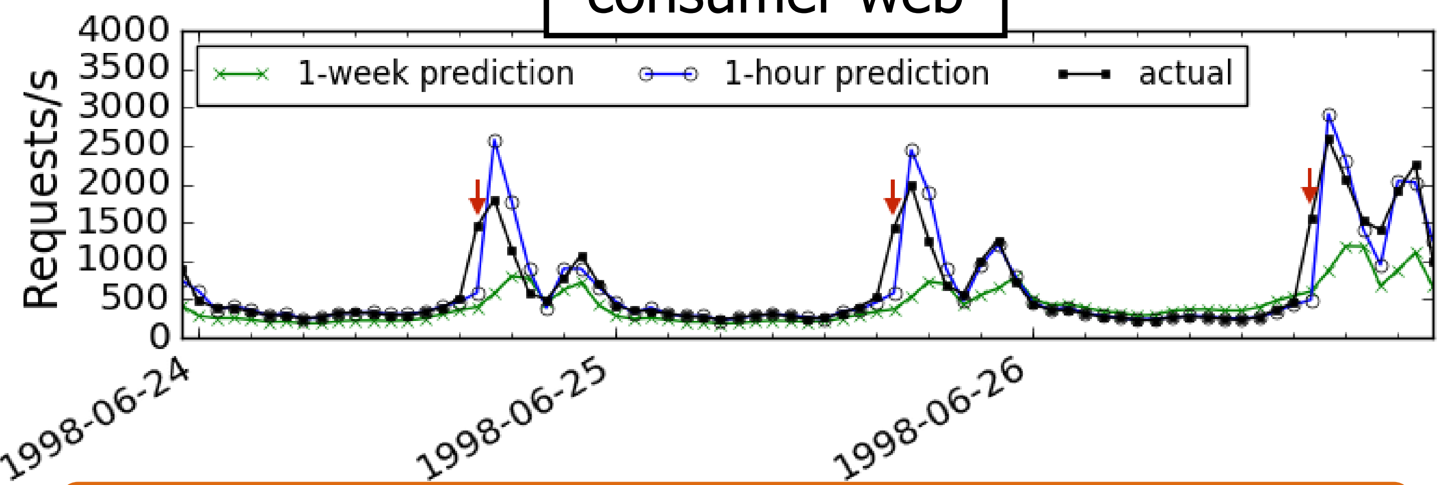
- **Response time constraint**
  - **95 %ile of response time distribution is not more than 0.15 s**

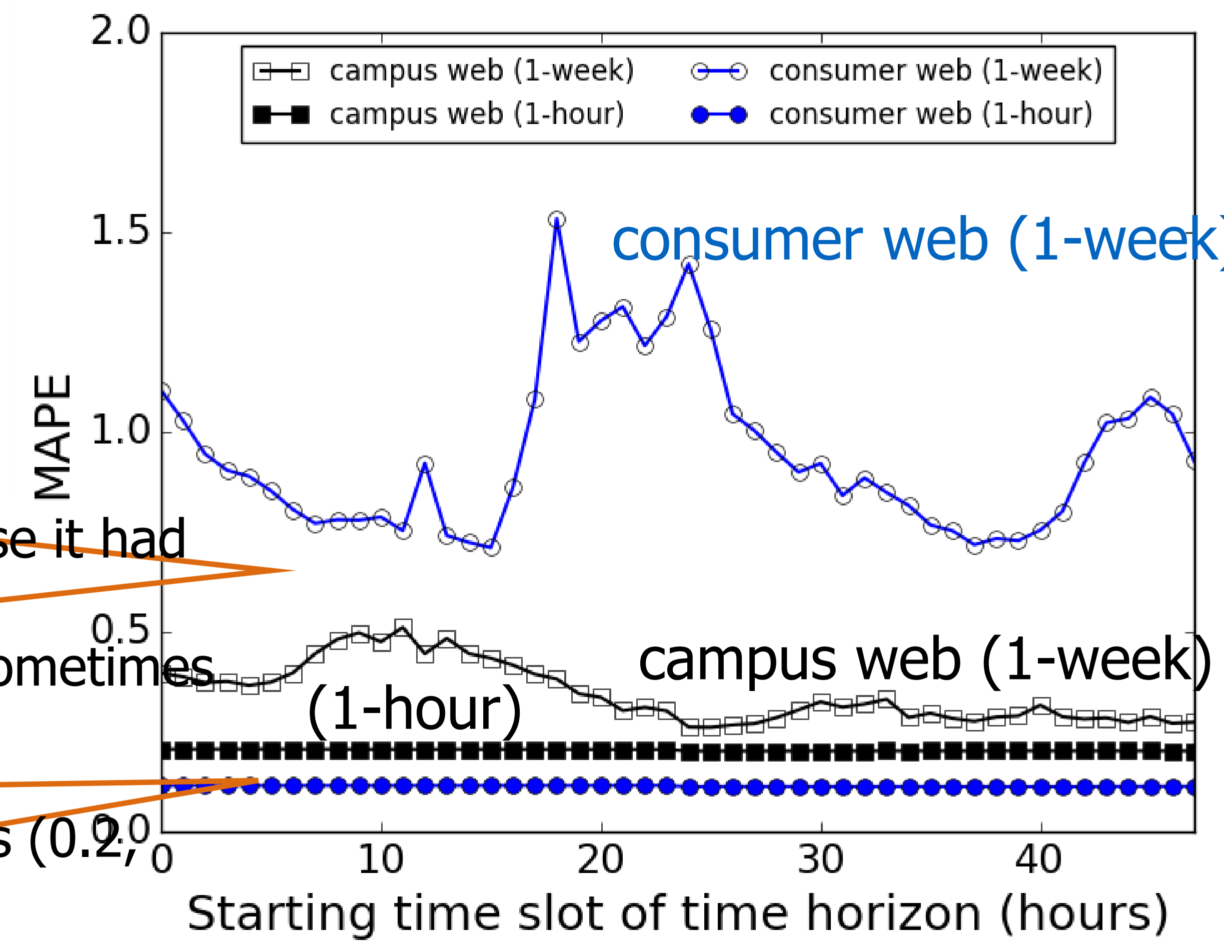
[1] The Internet Traffic Archive, "1998 world cup web site access logs," http://ita.ee.lbl.gov/html/contrib/WorldCup.html

Evaluation results:
Prediction error of request rate

- Identifying ARIMA model parameters ← last 3 week data, logarithmic scale conversion
- Performing the allocation process 48 times by changing staring time slot
- Analyzing the MAPE defined as $\frac{1}{T}\sum_{t=1}^{T}\frac{|x_t - \hat{x}_t|}{x_t}$

consumer web





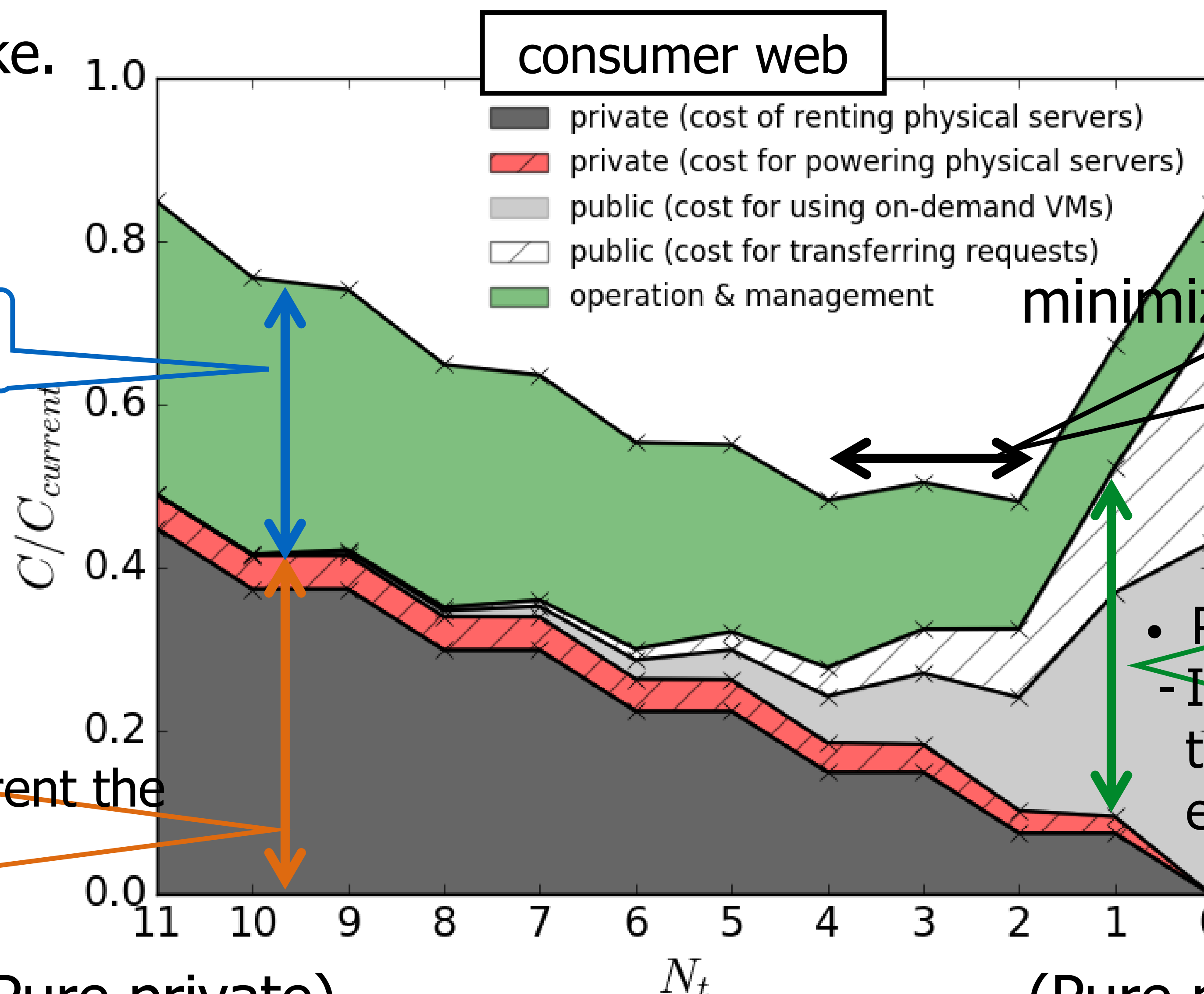
- One-week predictions
  - Campus web showed relatively small (0.34) error because it had regular predictable patterns.
  - Consumer web showed a large (0.94) error because it sometimes received unexpected request spikes.

- One-hour predictions indicated small errors in both webs (0.2, 0.1)

MAPE: Mean Absolute Percentage Error

Evaluation results:
Sizing of VM pool in private data center

- Total cost ($C/C_{current}$) in a week as a function of private VM pool size (N_t)
  - $C_{current}$: cost when private VMs can handle the maximum request rate and always awake.



(Pure private) (Pure public)

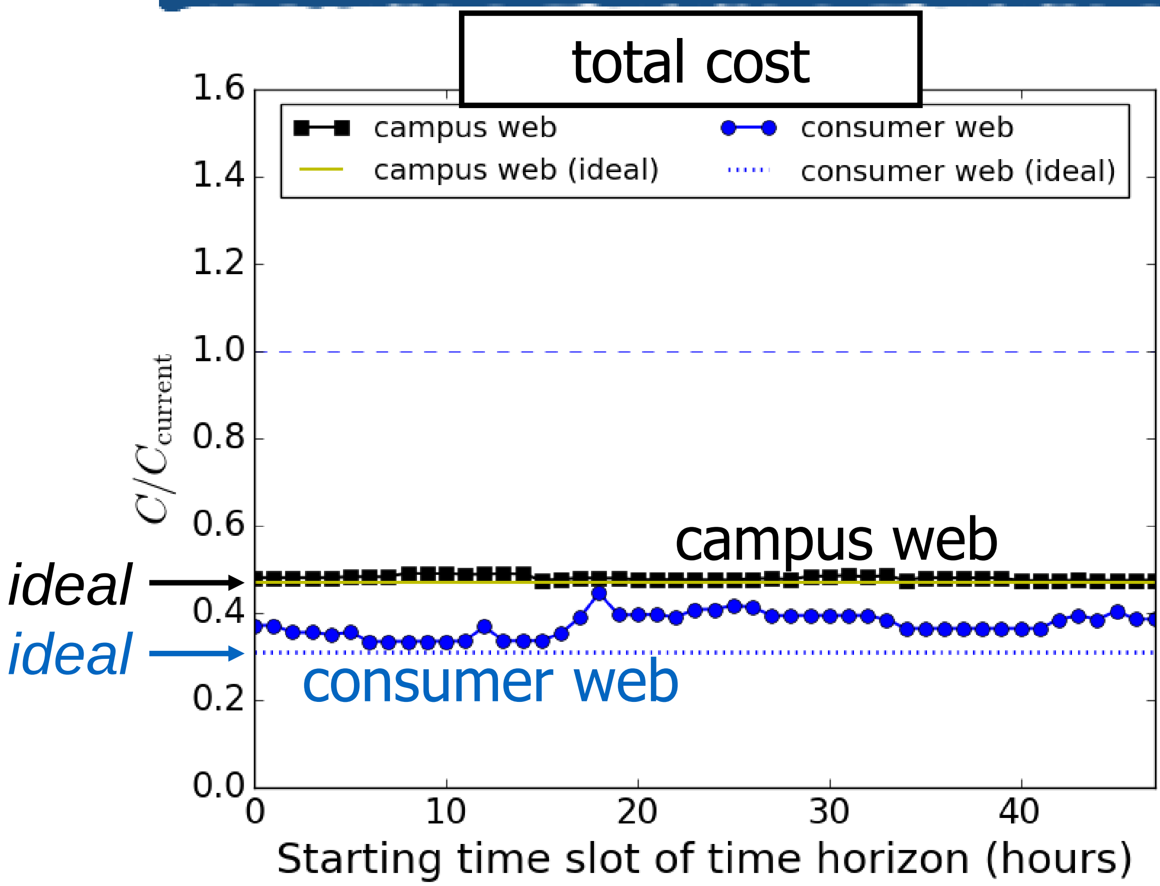
minimized when $N_t = 2 \sim 4$

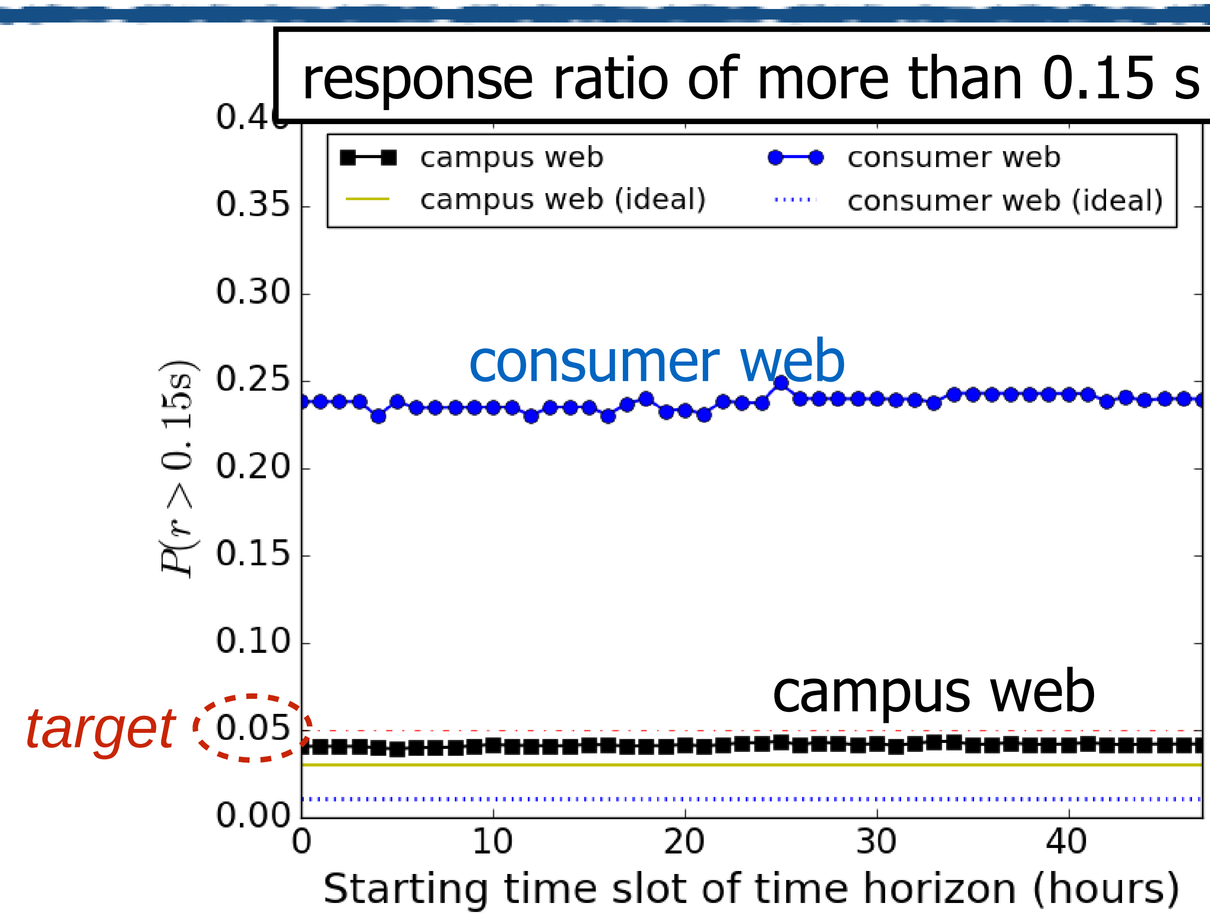
- Management cost

- Private DC cost
  - Most cost went to rent the physical servers.

- Public DC cost
  - Increased rapidly because the number of VMs greatly expanded.

Evaluation results:
Total cost and response time





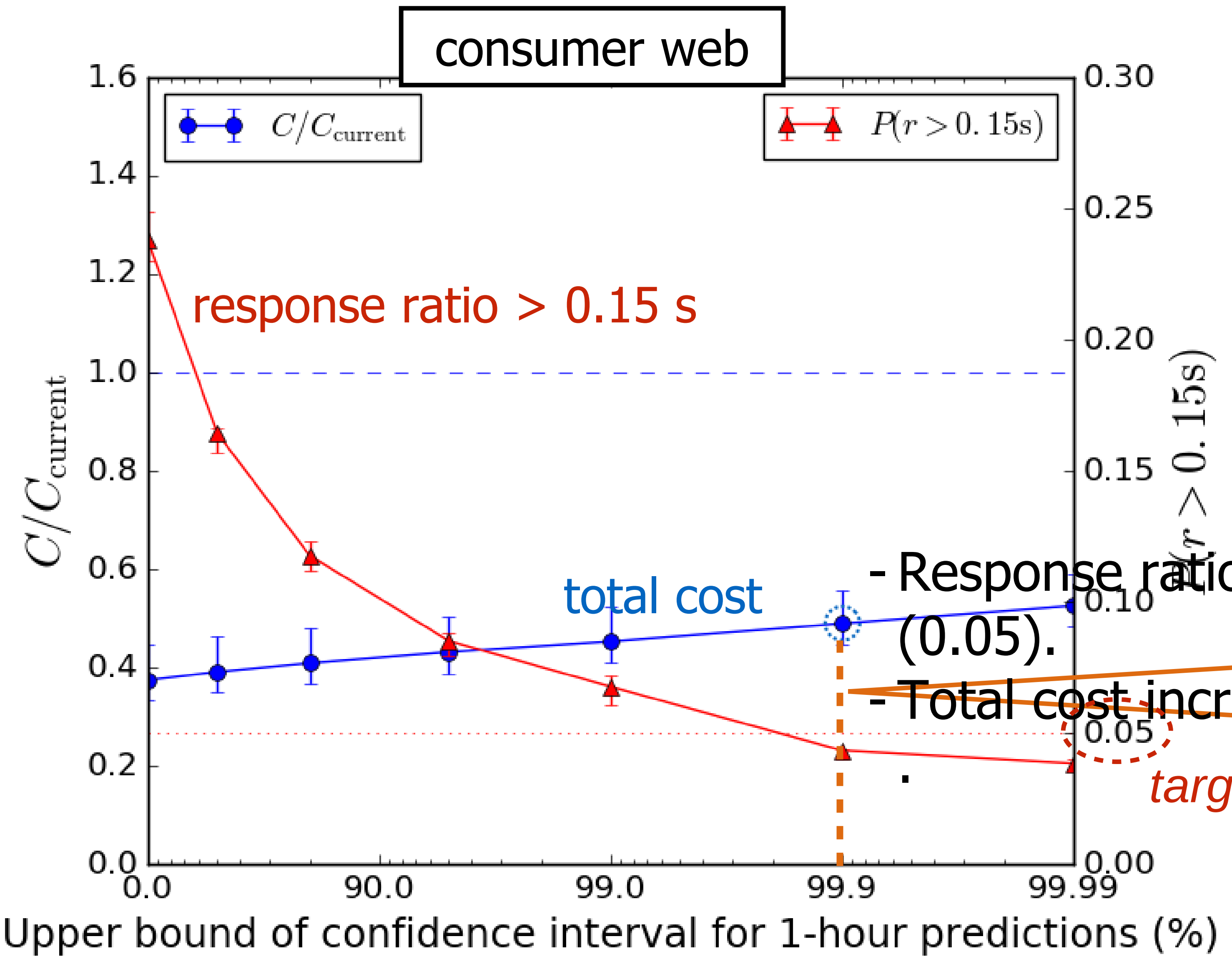
- Campus web
  - Total cost corresponded to its ideal.
  - Response ratio was below the (transformed) target of 0.05.
    ←Both one-hour and one-week predictions had high accuracy.

- Consumer web
  - Total cost was slightly larger than its ideal.
    ←One-week prediction errors made the private VM pool over-provisioned.
  - Response ratio was much more than target.
    ←One-hour prediction errors made the VMs under-provisioned.

**ideal* assumes a case in which the future requests are known a priori.

Evaluation results:
Handling of One-Hour Prediction Errors

- VMs are activated on the basis of the point estimate for the request rate.
  - Estimation errors sometimes make VMs under-provisioned.
  - Use the upper bound of the interval estimate instead of the point estimate.



Trade-off between the total cost and the response-time performance

- Response ratio was below the target probability (0.05).
- Total cost increased but still remained half of $C_{current}$.

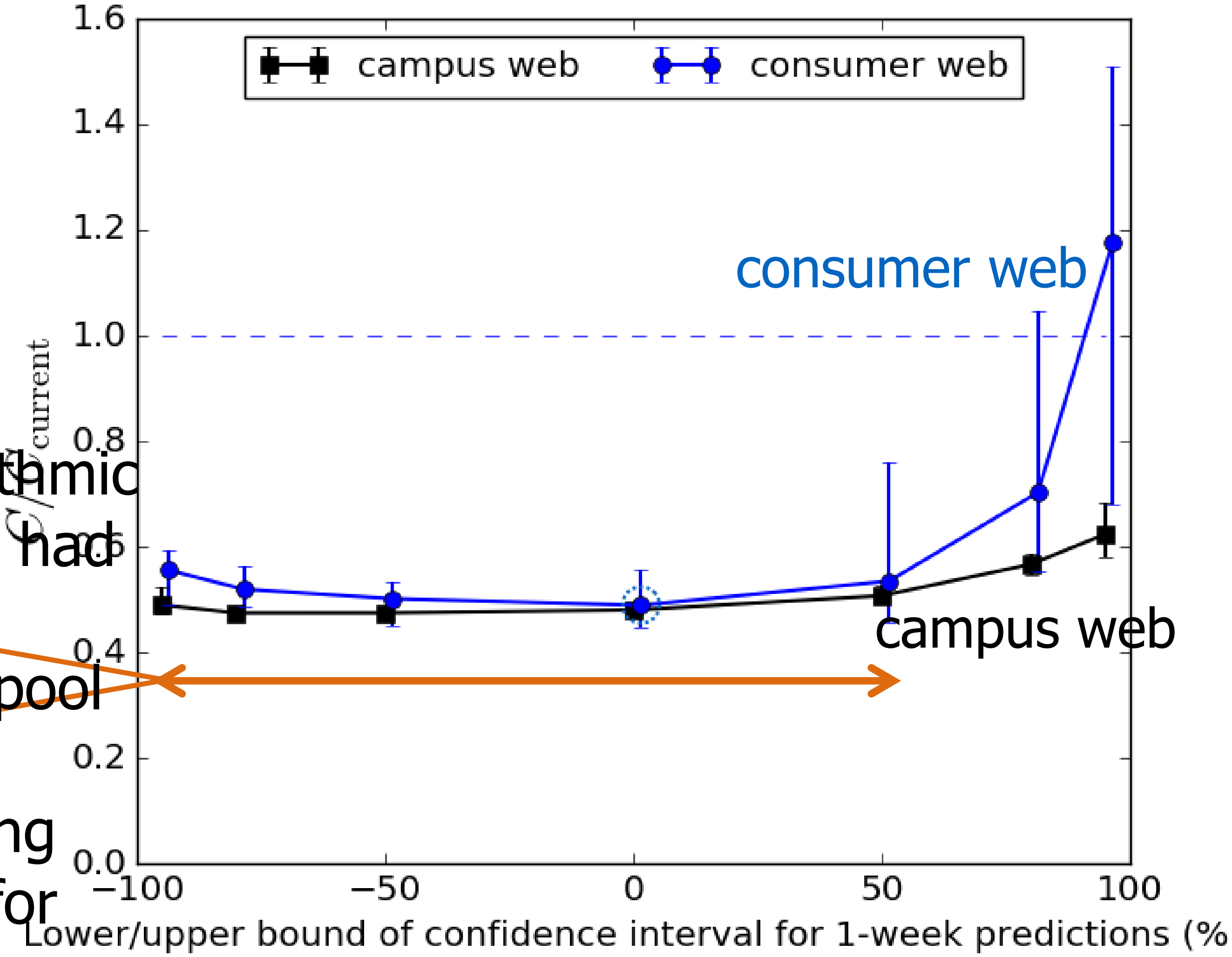
Evaluation results:
Impact of One-Week Prediction Errors on Total Cost

- One-week prediction errors change the size of the VM pool in the private DC.
- The private VM pool is over-/under-provisioned with the upper/lower bund of confidence interval for 1-week predictions.

VM pool size deviation had little effect on total cost.

- Owing to converting request rate into a logarithmic scale, lower bounds of the confidence interval had smaller fluctuations.
- Total cost didn't change largely while the VM pool size (N_t) was near the optimal point.

- These advantages came from the VM pool being provisioned for the average request rate, not for the maximum rate.



Conclusion

- Cloud bursting approach: assigning a dedicated VM pool in a private DC on the basis of one-week predictions and determining the active VMs in private and public DCs on the basis of one-hour predictions.

- One-hour prediction errors caused the response delay.
- One-week prediction errors caused the VM pool in the private DC to be under- or over-provisioned.

- The evaluation results indicate that our approach can become tolerant of prediction errors by handling the confidence interval for predictions.

Thank you for your attention.