Accuracy-Based Cache Consistency Management for Numerical Object Replication

Hideya Ochiai
The University of Tokyo
jo2lxq@hongo.wide.ad.jp

Hiroshi Esaki
The University of Tokyo
hiroshi@wide.ad.jp

Abstract

Object replication and cache consistency have been one of major design issues in the recent Internet applications. In this paper, we forcus on accuracy-aware applications (i.e., sensor applications), which use numerical data with allowing some errors. As long as the cache consistency between the origin's and replicated objects is guaranteed to some error levels, the system do not need to refresh its cache, resulting in saving network workloads. We propose a numerical cache consistency model and a cache control method for such applications. The results of our experments on weather temperature data have shown (1) network traffic was dynamically reduced depending on application requested accuracy, and (2) the rate of exceedance to the threshold of allowable error was controllable, which was not in the traditional time-based cache validity control.

1 INTRODUCTION

Recent Internet applications replicate objects among distributed servers, improving user request-to-response performance and reducing network workloads. In wide area sensor networks, sensor readings collected by a server could be replicated to other servers to which most of users periodically access. Static cache lifetime setting, which is widely carried out in the Internet applications(e.g., DNS, HTTP), does not always satisfy the requirements of accuracy-aware applications, where inconsistency of caches is associated with the numerical error between the origin server's and cached value.

Applications, in this paper, allow some inconsistencies (i.e., errors) between the origin's and cached value, modeling those allowable error numerically by *accuracy requirement*. We propose a numerical cache consistency model and a cache validity test method for such accuracy-aware applications. Applications that utilize sensor readings are instances of such application families. Our proposed method reduces the traffic between the origin and cache proxy servers as much as possible with guaranteeing given accuracy requirement.

In this paper, we assume that continuously generated numerical objects at the origin server should be fetched by proxy servers distributed over the Internet, and that IP network latency and limited bandwidth should not be ignored for transferring data. This situation occurs especially when these servers are distributed internationally or over poor datalinks.

In the Internet applications, the cache validity is practically modeled by *lifetime*. They usually assume that objects should be discrete with respect to *time*; i.e., the updated documents must be available within specific minutes or hours. However, some applications (e.g., sensor) can assume that objects change numerically and continuously. The absolute subtracts of objects between the origin's and cached are more appropriate for the model of cache consistency than the lifetime model.

In our proposed system model, a proxy server dynamically calculates cache validity time depending on the changing frequency of the value and the accuracy requirement of user applications. When the objects at the origin server change more frequently, the time for cache validity decreases, while increasing the origin-to-proxy traffic. On the other hand, when they change less frequently, the validity time increases, resulting in the reduction of traffic.

The rest of this paper is organized as follows. In section 2, we describe related works. In section 3, we propose a method for accuracy-based cache validity control. Section 4 describes evaluation. In section 5, we provide discussion and conclusion.

2 RELATED WORK

Yu and Vahdat[3, 4] have proposed a numerical consistency model in write propagation replication ser-

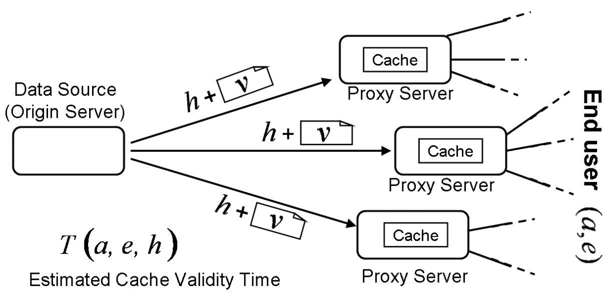

Figure 1. Value replication with h, the information that describes the hidden state of $v(t)$

vices, supposing that user applications should allow some errors. In their model, a origin server waits for propagating updates until the differences between its and replicated values come to the threshold that user requested.

Olston et al[2] have proposed approximate caching and quering methods, which maximize performance gains by dynamically adapting precision setting to user application requirements.

In these works, they focus on the situation where updates should be propagated by *push* manner, and the propagation is initiated by the origin servers. We focus on *pull-based* cache replication, which objects are replicated on demand. The validity of replicated data are tested by the proxies and end users.

3 ACCURACY-BASED CACHE VALIDITY CONTROL

3.1 System Model

Figure 1 illustrates the system we discuss. Proxy servers make replications of the value managed at an origin server, initiated by a request of user applications. Let $v(t)$ be the value produced by the origin server at time t, a be the accuracy requirement (i.e., allowable error) given by the application to be guaranteed, and e be an expected exceedance of values to the accuracy requirement. For example, if $e = 0.1$, the cache inconsistency is expected to exceed the allowable error at the rate of 0.1. The end users submit latest data retrieval queries to their local proxy servers with a and e.

Initiated by the end user, the proxy server downloads the current value $v(t_0)$ from the origin server if it does not have any valid value in its cache. Here, we associated that time as t_0. The proxy stores $v(t_0)$ in the cache after downloaded.

The $v(t_0)$ is transferred with information that describes the hidden state of $v(t_0)$, which is denoted by $h(t_0)$. In our model, $h(t)$ is used for cache time estimation and should be correlated with valid cache time under given a and e. The changing rate of $v(t)$ (i.e., $\frac{dv}{dt}\mid_{t=t_0}$) is the one of candidates for $h(t_0)$.

We define estimated cache validity time as,

$$T(a, e, h)$$

This function tells the threshold for cache validity by time on the a and e requirements. Normally, T is strictly increasing function with regard to a and e; i.e., the change of accuracy requirements from 0.1 to 1.0 prolongs cache validity.

We here address how to test cache validity. Suppose that a proxy has cached a value at time t_0; $v(t_0)$ is cached with $h(t_0)$. When a user requests with a_1 and e_1 at time t_1 to the proxy ($t_1 > t_0$),

when,

$$t_1 - t_0 \leq T(a_1, e_1, h(t_0)) \tag{1}$$

then, the cached $v(t_0)$ is valid. The proxy should return $v(t_0)$ to the requested user.

when,

$$t_1 - t_0 > T(a_1, e_1, h(t_0)) \tag{2}$$

then, the cached $v(t_0)$ is invalid for the user, it must be refleshed.

Since the T varies on the a and e, the actually estimated cache validity time can be minimum T. Less variation of the a and e improves the effectiveness of our proposed method, and systems should be implemented to maximize such effectiveness.

3.2 Formulating the Estimated Cache Validity Time

The implementation of $T(a, e, h)$ depends on the features of $v(t)$. It could be formulated by statistically analyzing $v(t)$. We propose a method to get $T(a, e, h)$ by analyzing distribution (i.e., probability density) of the h and actual cache validity times, which we denote by τ, under several accuracy requirements a.

In this paper, we denote the distribution by $D_a(h, \tau)$. Formally, τ can be calculated by the following formula,

$$\tau(t) = \min_{t_0}\{t_0 \mid |v(t_0) - v(t)| > a\} - t \tag{3}$$

$$(t_0 > t)$$

At finer accuracy requirement (e.g., $a = 0.1$), τ is biased to be small, whereas at grainer accuracy requirement (e.g., $a = 1.0$), τ is widely distributed. As h becomes larger, smaller τ becomes in the case of h being $|\frac{dv}{dt}|$.

After analyzing some D_a, we can get $T(a, e, h)$ by the following transformation rule. $T(a, e, h)$ is the T that satisfies,

$$\int_0^T D_a(\tau \mid h) d\tau = e \tag{4}$$

Here, e is the rate of exceedance to the accuracy requirement, and $D_a(\tau|h)$ is,

$$D_a(\tau \mid h) = \frac{D_a(h, \tau)}{\int_0^{+\infty} D_a(h, \tau) d\tau} \tag{5}$$

$T(a, e, h)$ is the cache time such that the exceedance of the threshold a are expected to e on given a and h. In practice, after getting $T(a, e, h)$ from the statistical analysis, it should be approximated to a formula.

4 EVALUATION

We evaluated how much our proposed method reduces network traffic and satisfies application requirements compared to the time-based cache validity control method.

We used Live E![1] temperature sensor readings for evaluation. We chose 39 sensors that have the same feature in sampling data: i.e., they are organized by the same sensor model. The total count of the temperature values for analysis was about 4.2 million.

In this analysis and simulation, we associated h as $|\frac{dv}{dt}|$, the absolute changing rate of the value. Cache validity time has a correlation with $|\frac{dv}{dt}|$.

4.1 Estimated Cache Validity Time

We analyzed $D_a(h, \tau)$ from $a = 0.1$ to $a = 1.0$. Figure 2 shows the distribution in the case of $a = 1.0$. Using these data, we transformed it to $T(a, e, h)$ using the formula (4).

After the analysis of the form of $T(a, e, h)$, we concluded that the following formula approximately provides $T(a, e, h)$.

$$T(a, e, h) \approx 0.22ae \exp\{3.5e(e-1)\} \exp\{-0.00963h\} [sec]$$

We used this formula for estimating the validity time in the following experiments.

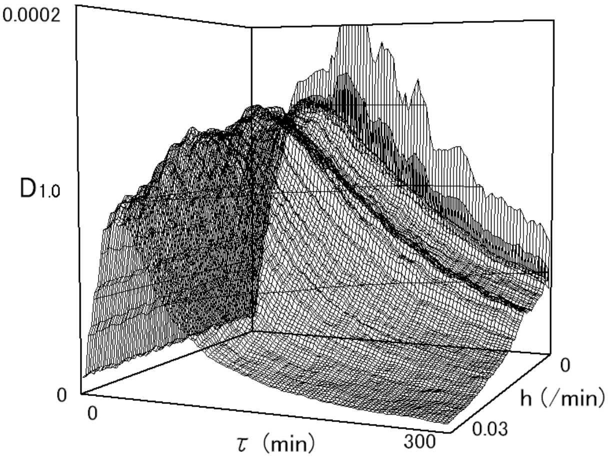

Figure 2. The distribution of the valid cache time τ and changing rate of value h at the accuracy requirement 1.0 on Live E! weather temperature data

4.2 Reduction of Traffic

Figure 3 shows the traffic on accuracy requirements under the accuracy-based cache control. As grainer the accuracy requirement becomes, the less traffic it generated. As more expected exceedance is allowed, the less traffic it generated.

In the traditional time-based cache control, the traffic do not vary even though users have different accuracy requirements. The ability of changeing traffic depending on accuracy requirements indicates that the proposed method has reduced the wasteful data exchange workloads in the time-based cache control.

4.3 Exceedance of Error Boundaries

Figure 4 shows the observed exceedance on several expected exceedance specification given by e and accuracy requirements under the accuracy-based control. They almost remained to the expected exceedance in various accuracy requirements.

In the time-based validity control, the exceedance cannot be configured by users. The validity is determined statically by its administrator. As figure 5 shows, the observed exceedance was high for some users (i.e., system provided useless data for those users) and was extremely low for others (i.e., system required high network workloads).

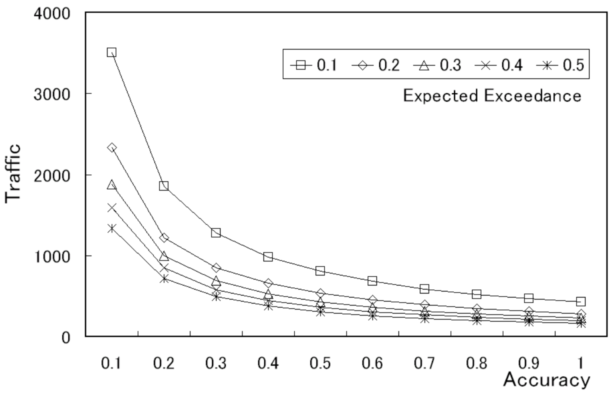

Figure 3. Traffic generated by the origin server

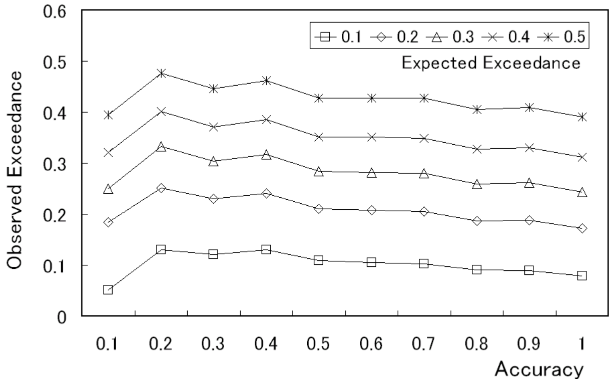

Figure 4. Observed exceedance on accuracy-based cache control

5 DISCUSSION and CONCLUSION

We have proposed a numerical cache consistency model and a validity control method for accuracy-aware applications. In this idea, the absolute subtracts of numerical values between the origin and cache servers are more appropriate for the metric of cache consistency than the lifetime metric. Our method has enabled dynamic cache time control depending on the changing rate and accuracy requirements of applications. The results of our experiments have shown; (1) network traffic was dynamically and appropriately controlled depending on user requested accuracy; (2) exceedance rate to the threshold of error was controllable in accuracy-based cache control, whereas the control was difficult in time-based method.

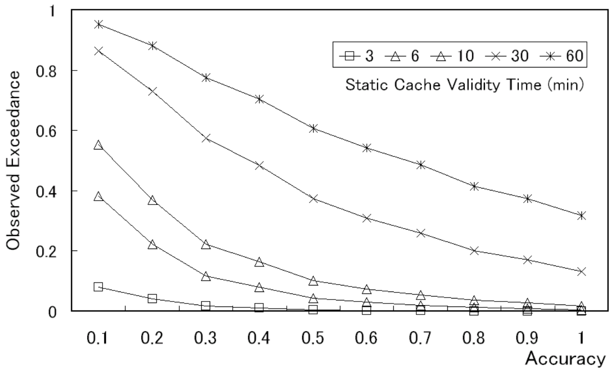

Figure 5. Observed exceedance on time-based cache control

References

[1] M. Nakayama, S. Matsuura, H. Esaki, and H. Sunahara. Live E! project: Sensing the earth. *LNCS*, 4311:61–74, 2006.

[2] C. Olston, B. T. Loo, and J. Widom. Adaptive precision setting for cached approximate values. In *ACM SIGMOD*, pages 355–366, 2001.

[3] H. Yu and A. Vahdat. Efficient numerical error bounding for replicated network services. In *The VLDB Journal*, pages 123–133, 2000.

[4] H. Yu and A. Vahdat. Design and evaluation of a conit-based continuous consistency model for replicated services. *ACM Transactions on Computer Systems*, 20(3):239–282, aug 2002.