Effect of Proactive Serving on User Delay Reduction in Service Systems

Shaoquan Zhang
The Chinese University of Hong Kong

Longbo Huang
Tsinghua University

Minghua Chen
The Chinese University of Hong Kong

Xin Liu
Microsoft Research Asia

ABSTRACT

In online service systems, delay experienced by a user from the service request to the service completion is one of the most critical performance metrics. To improve user delay experience, in this paper, we investigate a novel aspect of system design: *proactive serving*, where the system can predict future user request arrivals and allocate its capacity to serve these upcoming requests proactively. In particular, we investigate the average user delay under proactive serving from a queuing theory perspective. We show that proactive serving reduces the average user delay *exponentially* (as a function of the prediction window size) under M/M/1 queueing models. Our simulation results show that, for G/G/1 queueing models, the average user delay also decreases significantly under proactive serving.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems

Keywords

Proactive serving, user delay, service system, queueing

1. INTRODUCTION

Consider a service system as shown in Fig. 1. In the system, the single backend server provides service to incoming user requests that arrive at the system according to a Poisson process $\{A(t)\}_t$ with rate λ. When a user request arrives and the server is idle, the request will be served. Otherwise, the request waits in the queue $Q(t)$ for service. The queueing discipline is FCFS. After being served, the request leaves the system. We assume that service times of requests are i.i.d. exponential random variables with mean $\frac{1}{\mu}$. Such system can be modeled as a standard M/M/1 queue. We define user delay as the time from when the user request arrives at the system till it leaves the system.

Now suppose that the service system has proactive serving capability. As shown in Fig. 2, we assume that the system can perfectly predict user request arrivals ω time ahead.

SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.
ACM 978-1-4503-2789-3/14/06.
http://dx.doi.org/10.1145/2591971.2592024.

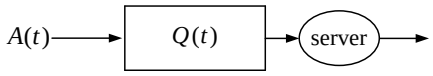

Figure 1: A Single queue service system.

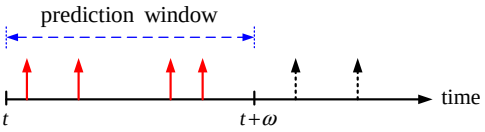

Figure 2: Prediction model.

That is, at time t, the system knows exactly in $(t, t+\omega)$ the request arrival epochs (red solid arrows) and the corresponding users who generate the requests. Meanwhile, we do not assume the knowledge of the workload of each user request.

Based on the arrival prediction, the system can allocate its service capacity to serve future requests proactively. Specifically, the servers can provide service to the users who are predicted to generate requests. The user requests that get pre-served will not enter the system. Such a proactive serving model captures service systems that can perform cache pre-loading or command pre-fetching. As a practical example of cache pre-loading, the web browser in Amazon's Kindle Fire can predict web page requests and pre-load desired web pages to users' tablets beforehand. When the user clicks on the predicted content, it gets the content immediately.

We depict the service system which can proactively serve future requests based on perfect prediction in Fig. 3. Let $Q_0(t)$ represent the queue of the requests that have arrived at the system and are waiting for service at time t, and $W_\omega(t)$ be the prediction window of size ω.

Each user request first goes through the prediction window $W_\omega(t)$ and then enters the queue $Q_0(t)$. The servers can serve the requests in both $Q_0(t)$ and $W_\omega(t)$. We remark that each request entering $W_\omega(t)$ will transit to $Q_0(t)$ after exactly ω amount of time, if it has not been pre-served before that. The requests will not queue up in $W_\omega(t)$. Thus $W_\omega(t)$ should be viewed as a pipe. User delay corresponds to the time that the request spends in $Q_0(t)$ and with the server, and it does not include the time spent in $W_\omega(t)$.

Previous work either focus on request scheduling algorithm design based on future prediction but without proactive serving capacity [1], [2], [3] or investigate how proactive serving reduces the probability of server outage [4]. Different from previous work, in this paper, we are interested in the following fundamental question. How much user delay reduction can we obtain by proactive serving?

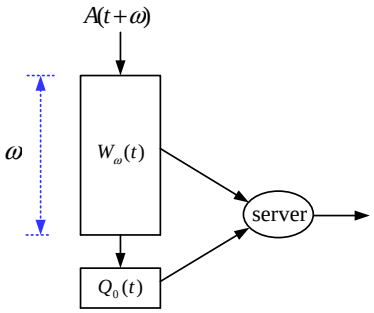

Figure 3: Service system with perfect prediction.

2. AVERAGE USER DELAY

In this section, we characterize the average user delay of the system shown in Fig. 3. Let D^ω denote the user delay when the system can predict ω time ahead.

When $\omega = 0$, *i.e.*, without proactive serving, the system reduces to the classical M/M/1 queue. It's well known that the average user delay is given by

$$E\left[D^0\right] = \frac{1}{\mu - \lambda}. \tag{1}$$

The probability density function of D^0 is also known as

$$f_{D^0}(t) = (\mu - \lambda)e^{-(\mu-\lambda)t}, t \geq 0. \tag{2}$$

To characterize the user delay, we first observe an interesting result that $Q^{sum}(t) \triangleq Q_0(t) + W_\omega(t)$ evolves the same as an M/M/1 system with a properly initialized queue. Based on this observation, the distribution of user delay under proactive serving, *i.e.*, D^ω, turns out to be a "shifted" version of that of user delay without proactive serving as shown in (2). Once we know the distribution of D^ω, we can compute the average user delay $E\left[D^\omega\right]$ as shown in the following theorem. Detailed proof can be found in [5].

Theorem 1. *Assume $\mu > \lambda$. The average user delay is given by*

$$E[D^\omega] = \frac{1}{\mu - \lambda} e^{-(\mu-\lambda)\omega}. \tag{3}$$

Theorem 1 reveals that the average user delay decays exponentially in the prediction window size ω. This result suggests that proactive serving is very powerful in reducing user delay.

From Theorem 1, in the heavy-load regime where the arrival rate λ is close to the service rate μ, (3) can be approximated by $\frac{1}{\mu-\lambda} - \omega$ when ω is small, which is linear in ω. As contrast, when the system is in the light-load regime, (3) decreases exponentially in ω. This indicates that proactive serving is less effective in reducing delay in the heavy-load regime than in the light-load regime. The reason is as follows. In the light-load regime, the number of requests that enter $Q_0(t)$ for service is small and thus most of the serve capacity can be spared to serve future requests. In contrast, in the heavy-load regime, the server is busy with serving requests in $Q_0(t)$ most of time. As a result, the chance that a request gets served proactively by the server is small when going through a small prediction window. Therefore, proactive serving has limited effect on the delay reduction. When the system is in the heavy-load regime, to achieve small user delay, the system needs to guarantee a sufficient large prediction window so that the probability that a request gets served proactively is high.

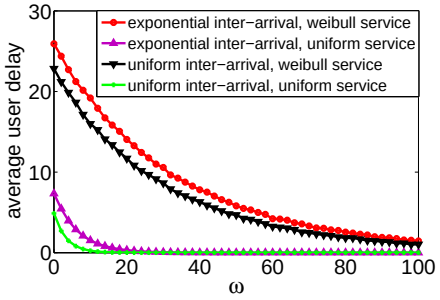

Figure 4: The average request delay vs how far we predict the future under perfect prediction under different inter-arrival time distributions and service time distributions.

3. SIMULATION RESULTS

We carry out simulations to show the impact of proactive serving on reducing the average user delay under different arrival processes and service time distributions. We consider four different pairs of request inter-arrival time distribution and service time distribution: (Exponential, Uniform), (Uniform, Uniform), (Exponential, Weibull), (Uniform, Weibull). The mean arrival rate is $\lambda = 0.4$ and the mean service rate is $\mu = 0.5$. For the Uniform distribution, boundaries are 0 and double mean. For the Weibull distribution, the scale parameter is equal to 1. There is a single server in the system under four settings. The server adopts FCFS as its service policy.

The results are shown in Fig. 4. As seen, proactive serving reduces the average user delay significantly under all settings.

4. ACKNOWLEDGMENTS

This work described in this paper was partially supported by the National Basic Research Program of China Grant Project No. 2011CBA00300, 2011CBA00301, 2012CB315904, 2013CB336700, the National Natural Science Foundation of China Grant Project No. 61033001, 61361136003, 041302007, the China youth 1000-talent grant Project No. 610502003, the MSRA-Tsinghua joint lab grant Project No. 041902004, and several grants from the University Grants Committee of the Hong Kong Special Administrative Region, China (Area of Excellence Grant Project No. AoE/E-02/08, Theme-based Research Scheme Project No. T23-407/13-N, and General Research Fund Project No. 411011).

5. REFERENCES

[1] B. Coleman. Quality vs. performance in lookahead scheduling. In *JCIS*, 2006.

[2] M. Mandelbaum and D. Shabtay. Scheduling unit length jobs on parallel machines with lookahead information. *Journal of Scheduling*, 2011.

[3] J. Spencer, M. Sudan, and K. Xu. Queueing with future information. *arXiv:1211.0618*, 2012.

[4] J. Tadrous, A. Eryilmaz, and H. El Gamal. Proactive resource allocation: harnessing the diversity and multicast gains. *IEEE Trans. Information Theory*, 2013.

[5] S. Zhang, L. Huang, M. Chen, and X. Liu. Effect of proactive serving on user delay reduction in service systems. Technical report. `https://sites.google.com/site/proactivetechreport/paper.pdf`.