Mitigation of Flash Crowd in Web Services By Providing Feedback Information to Users

Harumasa TADA^{†a)}, Member, Masayuki MURATA^{††b)}, and Masaki AIDA^{†††c)}, Fellows

SUMMARY The term “flash crowd” describes a situation in which a large number of users access a Web service simultaneously. Flash crowds, in particular, constitute a critical problem in e-commerce applications because of the potential for enormous economic damage as well as difficulty in management. Flash crowds can become more serious depending on users’ behavior. When a flash crowd occurs, the delay in server response may cause users to retransmit their requests, thereby adding to the server load. In the present paper, we propose to use the psychological factors of the users for flash crowd mitigation. We aim to analyze changes in the user behavior by presenting feedback information. To evaluate the proposed method, we performed subject experiments and stress tests. Subject experiments showed that, by providing feedback information, the average number of request retransmissions decreased from 1.33 to 0.09, and the subjects that abandoned the service decreased from 81% to 0%. This confirmed that feedback information is effective in influencing user behavior in terms of abandonment and retransmission of requests. Stress tests showed that the average number of retransmissions decreased by 41%, and the proportion of abandonments decreased by 30%. These results revealed that the presentation of feedback information could mitigate the damage caused by flash crowds in real websites, although the effect is limited. The proposed method can be used in conjunction with conventional methods to handle flash crowds.

key words: web application, user behavior, cloud computing, stress test

1. Introduction

The demand for Web services has been increasing in recent years. Nowadays, e-commerce accounts for a large proportion of economic activity and is essential to our daily lives. Since Web services are the foundation of e-commerce, their failure results in great damage, both socially and economically.

On the other hand, we frequently read news items about Web services crashing. There are many examples of system downtime in ticket booking systems (e.g., [1]) pre-ordering systems (e.g., [2]), and the like. Although such events may not always be reported as news, these failures often occur even in small-scale web services.

These events are brought about by simultaneous access by an unexpected number of users—a situation referred to as a flash crowd. The occurrence of a flash crowd is rapid but not persistent, and the event eventually disappears if the requests are processed in order. However, the rapid increase in the server load may damage the processing power of the server or cause it to go down. Such performance decrease may cause the recovery from the flash crowd to take much more time than it normally should.

Flash crowds, in particular, constitute a critical problem in e-commerce for the following reasons. First, they result in enormous economic damage. The resultant response delay and potential service downtime increase the possibility of user abandonment, resulting in loss of business. Second, it is difficult to prepare for such events. In general, prediction of the scale and timing of occurrence for flash crowds is complex. If redundant server resources were to be prepared in anticipation of a flash crowd occurrence, they would constitute a waste during normal operation periods.

Previous methods for coping with flash crowds have focused on servers. Some of them reject excessive requests to prevent servers from going down (e.g., [3]). Another approach is to increase the processing power of servers to accept flooding requests (e.g., [4]). With these methods, however, it is difficult to solve the trade-off between the increase in users’ abandonment and the waste of server resources due to excessive deployment.

In the present study, we considered mitigating the effect of flash crowd with a different approach. We focused on user behavior because the effect of a flash crowd can become more serious depending on this aspect. If the server load increases due to the flash crowd, the response will be delayed. The delay in response may cause users to retransmit their requests, resulting in a further increase in the server load. In other words, if users change this behavior, it might be possible to prevent flash crowds from becoming a serious issue.

To influence user behavior, we considered using feedback messages following the study of Nah [5]. His experiments on the waiting period for file downloads showed that subjects were more tolerant to a longer waiting period when they received feedback information. In recent years, many studies have been conducted on the effect of feedback information to users on the perception of online waiting time [6]–[10]. These results suggest that feedback information may be able to prompt users to wait instead of repeating retransmissions in flash crowds. However, these studies focus on

Manuscript received February 26, 2020.

Manuscript revised July 10, 2020.

Manuscript publicized September 18, 2020.

[†]The author is with the Faculty of Education, Kyoto University of Education, Kyoto-shi, 612–8522 Japan.

^{††}The author is with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871 Japan.

^{†††}The author is with the Graduate School of Systems Design, Tokyo Metropolitan University, Hino-shi, 191–0065 Japan.

a) E-mail: htada@kyokyo-u.ac.jp

b) E-mail: murata@ist.osaka-u.ac.jp

c) E-mail: aida@tmu.ac.jp

DOI: 10.1587/transinf.2020MPP0003

short, predictable waiting time in normal times. In the case of flash crowd, since the system is overloaded, the waiting time is very long and difficult to predict. Therefore, it is uncertain whether the results of previous studies hold in flash crowds. Although some websites currently provide feedback information to users when experiencing performance problems (e.g., “If you are experiencing issues, we recommend you restart your PC”), such feedback may prompt users to retransmit their requests, resulting in further increase in the server load. It is therefore necessary to carefully choose the type of transmitted feedback information.

The present study contributes to the literature by answering the following two questions.

- How feedback information influences the behavior of Web service users who have been waiting for a long time?
- How changes in user behavior due to feedback information help mitigate actual flash crowds?

For the first question, we performed experiments in which subjects were exposed to a delay in response from a Web service. Different feedback information was presented to the subjects during the waiting time, and their behaviors were recorded. The observations suggested that user behavior could be influenced by feedback information. To answer the second question, we conducted stress tests on a real server. The test results revealed that the presentation of feedback information decreased the server load and user abandonment.

The early results of the present study have been reported in our conference paper [11]. The differences between this paper and the previous paper are as follows. We conducted additional subject experiments to increase the amount of data, which improved the confidence of the results of the experiment. Then, we discussed the result based on the tests conducted in a more appropriate way. For stress tests, we improved the test scenario to make stress tests more realistic. In addition, we added discussions on ethical considerations of subject experiments and social aspects of the proposed method.

The remainder of this paper is organized as follows. Section 2 provides a literature review. In Sect. 3, we report the subject experiments of the present study. In Sect. 4, we present the results of the stress tests. Section 5 concludes the study.

2. Related Works

Many studies on flash crowds on the Web have focused on the difference between a DoS attack and a flash crowd (e.g., [12]–[14]). These studies are complementary to our work. In the present paper, we focus on flash crowds caused by legitimate access.

There are studies which have aimed to predict the occurrence of flash crowds [15], [16]. If prediction of the occurrence of flash crowds is possible, advance preparation, such as enhancing server resources, can be done. However,

because of the popularization of social networking services, such as Twitter, diffusion of information has become much faster in recent years. Therefore, increase of access and flash crowds are becoming more rapid, and its prediction seems to be getting more difficult.

There are also some studies that investigated the characteristics of flash crowds based on actual traces [17]–[20]. The focus of the present study was not on the properties of the actual flash crowd, but on the psychological features of the Web users who generate them.

There are a number of studies on how to mitigate the effects of flash crowds.

The most conservative approach is to reject excessive access by admission control [3], [21], [22]. Chen et al. [22] reported that admission control increases the stability of the P2P live streaming systems against flash crowds. Although this approach is effective for avoiding server down time due to flash crowds, there is a possibility of lowering the user’s satisfaction and causing the user to leave.

A common approach in existing research is to distribute access using dynamically increasing replicas of content [4], [23]. This approach has been extensively studied, especially in P2P systems (e.g., [24]). Load balancing using caches or replicas is effective for services such as content distribution and video streaming. However, there are Web services that involve access to databases, such as e-commerce and ticket reservation. For such kinds of services, using caches or replicas requires an extra cost to maintain consistency of replicas.

In recent years, cloud computing has attracted much attention as a way to deal with flash crowds [20], [25]–[30]. In cloud computing, it is possible to flexibly increase the number of servers according to the increase in load. Cloud computing is promising as a relatively inexpensive preparation for flash crowds. However, starting backup servers of clouds requires a few minutes. Overprovisioning to mask this startup time requires an extra cost. In recent years, cloud providers provide burstable instances, which are capable of bursting up to the peak rate of normal instances and are much cheaper than normal instances. Baarzi et al. [31] proposed to using burstable instances for overprovisioning to prepare flash crowds. Their method is effective in increasing the maximum capacity of the system at a low cost and raising the threshold for occurring flash crowd. On the other hand, the purpose of the present study is to prevent the flash crowd from becoming serious when the number of accesses exceeds the capacity prepared in advance.

We focus on the influence of long waiting times caused by flash crowds on the user’s behavior. Some researchers have performed experiments to examine the effect of waiting periods on Web service users. Dellaert et al. [32] considered the website of an internet magazine and found that the subjects judged the website as being better when the waiting period was displayed by a countdown. Galbraith et al. [33] examined the relationship between the download time on a website and user satisfaction. They reported that users were more frustrated by a longer page download

time. Brutlag [34] reported that Web users performed fewer searches when there was a longer delay. Nah [5] examined whether user tolerance of a long download time could be enhanced by the presentation of feedback information and found that the display of a progress bar produced positive results. Based on the results of these previous studies, we considered the use of feedback information for managing flash crowds in the present study.

In recent years, many studies have been conducted on the effect of feedback information to users on the perception of online waiting time. Lee et al. [6] reported that users viewing the progress bar feel less uncertainty, and thus their perceived wait time (PWT) becomes short. Chen et al. [7] showed that there are cultural differences in how the design of the waiting screen affects the perception of waiting time by the user. Chen et al. [8] investigated the effect of the design of a wait indicator of mobile applications on a user's estimate of wait time. Hohenstein et al. [9] reported that by showing the dialogue side animation on the waiting screen, it felt that the waiting time was shorter than that of the progress bar. Soderstrom et al. [10] reported that the perception of the waiting time of the user changes depending on the moving speed of the animation displayed on the waiting screen. The variations of feedback information dealt with in these studies may be applicable to flash crowd mitigation. However, it would be required to re-examine their effectiveness in the flash crowd.

3. Subject Experiments

To ascertain whether the behavior of Web service users could be influenced by feedback information, we conducted experiments using subjects.

3.1 Hypothesis Development

We focus on the number of request retransmissions per user that relates the seriousness of flash crowds. Our goal is to reduce the number of retransmissions using feedback information. In the present study, we consider messages and temporal cues as feedback information. Therefore, the experiments address the following questions.

1. How do messages affect the number of retransmissions users make?
2. How do temporal cues affect the number of retransmissions users make?
3. How do the synergy of using messages and temporal cues affect the number of retransmissions users make?

We hypothesize the effects of messages and temporal cues on the number of retransmissions users make. We consider that a message about the current system state would make users feel less uncertainty and request retransmissions would be reduced. Thus, we hypothesize that:

H1: Users make less retransmissions when viewing a waiting page with a message as compared to one without message.

There are two types of messages that indicate the system status. That is, a message notifying that processing is being performed and a message notifying that access is concentrated. We consider that the user who received the former message would think that his/her task is already in the queue, while the user who received the latter message would feel more uncertainty. Thus, we hypothesize that:

H2: Users make less retransmissions when viewing a message notifying that processing is being performed as compared to a message notifying that access is concentrated.

In flash crowd, the overloaded system may display a system error message that should not be shown to users. We consider viewing a system error message maximize user uncertainty and strongly prompt retransmissions. Thus, we hypothesize that:

H3: Users make more retransmissions when viewing a system error message as compared to any other message.

It was reported that displaying temporal cues such as progress bars or countdown timers can shorten the users' perceived wait time (PWT) [6], [35]. Progress bars are widely used as a progress indicator. In flash crowd, however, the server is overloaded and users' tasks are likely not progressing. Therefore, displaying a progress bar is not realistic. Instead, we introduce a *waiting bar*. A waiting bar was an animation that consisted of a growing bar similar to a progress bar. However, unlike a progress bar, it did not actually reflect the progress of the process. Its growth speed decreased with time and it never reach the end during the preset waiting period. Thus, we hypothesize that:

H4: Users make less retransmissions when viewing a waiting bar as compared to no temporal cue.

We consider that the uncertainty of the user would be alleviated when viewing the expected waiting time. Though the expected waiting time may not really make sense in flash crowd, it may have the effect of changing user behavior. Thus, we hypothesize that:

H5: Users make less retransmissions when viewing an expected waiting time as compared to no temporal cue.

In flash crowd, it is possible that the server cannot respond in the expected waiting time. If there was no response after the expected waiting time, the user's uncertainty would be rather increased, which may cause many retransmissions. Thus, we hypothesize that:

H6: After a certain time has elapsed without response, users make more retransmissions when viewing a too short expected waiting time as compared to a long one.

We consider that a countdown timer that changes over time would be more effective than a fixed expected waiting time. Thus, we hypothesize that:

H7: Users make less retransmissions when viewing a waiting page with a countdown timer as compared to an expected waiting time.

Table 1 Pages of experimental Web site

name	role
home page	introduce the experiment
inquiry page	used to search for an empty room
waiting page	displays feedback information
customer information page	for input of customer information
finish page	informs the completion of the experiment

3.2 Experimental Design

3.2.1 Experimental Website

We developed an experimental website that simulated a hotel reservation service. The experimental website consisted of the pages listed in Table 1.

3.2.2 Participants

The subjects comprised 365 students of the Kyoto University of Education. Additional experiments were conducted in this study; thus, the number of subjects was higher compared with that in our previous paper [11]. They were not given any reward, financial or otherwise. They were recruited by making an announcement to students in some classes that one of the authors was in charge of. We placed a link to the experimental website on the web page for submitting the class report. However, students were informed that participation in the experiment was voluntary and it would never be reflected in class results. To ensure voluntary participation, the experiments were conducted without collecting any personal information. Therefore, the attributes of subjects, e.g. age, and gender are unknown. As the subjects were university students, most of them were estimated to be 18 to 21 years old.

3.2.3 Experimental Procedure

Subjects accessed the experimental website from their place of convenience such as their home or the computer room of the university. The experimenters thus did not locally observe the subjects during the experiment.

Subjects participated the experiment only once. Each subject was randomly assigned one of the feedback information display patterns. The number of subjects in each pattern was not evenly distributed.

The participants were required to first read the introduction of the experiment on the website homepage, where they were informed of the following.

- The purpose of the experiment will be presented after the experiment.
- The subject must not participate in the experiment more than once.
- The subject must not tell others about the experiment.
- The experiment is supposed to be completed in 10 minutes, and the subject can abandon after 10 minutes.

The homepage had a link to start the experiment. Participants clicked the link only if they agreed to be a subject. Subjects then moved on to the inquiry page, where they selected the date for which they wanted to reserve a room, and pressed the "Inquiry" button. They thereafter moved to the waiting page, where they were made to wait for a while. The waiting page displayed feedback information that differed among the subjects. It also contained a "Cancel" button that could be used to return to the inquiry page. After the elapse of the preset waiting period, the waiting page was automatically redirected to the customer information page, where they inputted the required information. If a subject went to another page during the waiting period, he/she went to the customer information page as soon as he/she went back to the waiting page after the waiting period. They finally moved to the finish page, where they were informed the purpose of the experiment.

The operation performed by a subject during the waiting period, such as request retransmission or abandonment, was recorded by the access log of the server. The preset waiting period was 3 min, which is longer than the generally tolerable waiting period for Web services.

3.2.4 Data Collection

We analyzed the access log of the Web server to determine the number of requests that each subject sent and the number of subjects who abandoned the experiment. The subjects were distinguished by their sender IP addresses in the log. Since there is a possibility of access from a shared PC in the computer room, we considered that accesses from the same private IP address that was more than 20 minutes apart were made by different subjects. If the waiting page was repeatedly requested by the same subject, we considered it as request retransmission[†]. If the finish page was requested by a subject, we considered he/she to have completed the experiment. Otherwise, we considered he/she to have abandoned the experiment.

3.2.5 Ethical Considerations

The HTTP (Hypertext Transfer Protocol) access log was the only information collected; no information that can identify individual subjects was collected. Subjects were not constrained to participate in the experiment. They were not exposed to unpleasant images, sounds, or texts through the experiments. The purpose of the experiment was not presented to them because doing so may have affected the result, and this aspect was explained to them in advance.

3.2.6 Feedback Information

The feedback information displayed on the waiting page

[†]If a particular page of the website were to be cached in a user browser, the log would not record the repeated requests of the page. To prevent such, we wrote a code that prohibited the caching of the website pages.

Table 2 Number of subjects

	a-1	a-2	a-3	a-4	a-5	a-6	b-1	b-2	b-3	b-4	b-5	b-6	c-1	d-1
Total	35	22	24	19	33	34	24	29	26	24	25	28	21	21
Completed	24	22	21	17	23	30	7	26	10	12	15	21	3	4
Abandoned	11	0	3	2	10	4	17	3	16	12	10	7	18	17
Proportion of abandonments	31%	0%	13%	11%	30%	12%	71%	10%	62%	50%	40%	25%	86%	81%

Table 3 Distribution of number of retransmissions per subjects

	a-1	a-2	a-3	a-4	a-5	a-6	b-1	b-2	b-3	b-4	b-5	b-6	c-1	d-1
0	21	20	16	17	21	29	6	23	4	7	1	10	3	6
1	9	2	7	1	9	3	4	5	8	8	10	9	5	5
2	4	0	1	1	2	1	9	1	5	9	6	4	4	7
3	1	0	0	0	0	0	3	0	3	0	5	3	4	3
4	0	0	0	0	0	0	0	0	3	0	1	1	3	0
5	0	0	0	0	0	1	1	0	1	0	0	1	1	0
6	0	0	0	0	0	0	0	0	1	0	1	0	1	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
>=10	0	0	0	0	1	0	1	0	1	0	0	0	0	0
Average	0.57	0.09	0.38	0.16	0.79	0.29	1.88	0.24	2.42	1.08	2.20	1.25	2.29	1.33

Table 4 Summary of ANOVA for the effect of message

Source	SS	df	MS	F	P	η^2
Message	129.0521	3	43.0174	17.8384	0.0000**	0.1291

** Significantly different at $\alpha = 0.01$ level (**P < 0.01).

consisted of a message and a temporal cue. The message presented to a particular subject could be any of the following (it was displayed in Japanese because all the subjects were Japanese):

- (a) “Now processing. Please wait for a while.”
- (b) “Accesses are presently concentrated. Please wait for a while and try again.”
- (c) A system error message: “Database connection error.”
- (d) No message. A blank page was displayed.

When (a) or (b) above was displayed, it was accompanied by one of the following temporal cues:

- (1) No temporal cue. Message only.
- (2) A waiting bar. (see Sect. 3.1)
- (3) Expected waiting time (1 min).
- (4) Expected waiting time (3 min).
- (5) Countdown timer (1 min).
- (6) Countdown timer (3 min).

When (c) or (d) above was displayed, No temporal cue was displayed.

In the cases of (3) and (4) above, the waiting time was displayed but did not count down. In the cases of (5) and (6) above, the waiting time was displayed by the countdown timer. Although the displayed time in (3) and (5) was 1 min, the subject was made to wait for 3 min.

Overall, 14 patterns of feedback information were presented to the subjects, namely, a-1, a-2, a-3, a-4, a-5, a-6, b-1, b-2, b-3, b-4, b-5, b-6, c-1, and d-1.

3.3 Results

3.3.1 Number of Retransmissions

Table 2 shows the number of subjects for each pattern. Table 3 shows the distribution of number of retransmissions per subjects. We analyzed the number of retransmissions using ANOVA. The main effect of messages and temporal cues together with their interaction effects on the number of retransmissions. The significant main effect was further analyzed with the Shaffer’s multiple comparison to examine the differences among the factor levels.

Table 4 illustrates the results generated from the ANOVA related to the comparison of the four types of messages. The results indicated that the main effect of messages on the number of retransmissions was significant. Table 5 illustrates the results of the multi comparison. The results showed that there were significant differences between (a) and (d), which supports H1. Moreover, there were significant differences between (a) and (b), which supports H2. However, no significant differences existed between (b) and (c), and between (c) and (d). Thus, H3 was not supported.

Table 6 illustrates the results generated from the two-way ANOVA related to the comparison of the two messages (a and b), and the comparison of six temporal cues (1, 2, 3, 4, 5, and 6). The results indicated that the main effect of temporal cues was significant. Moreover, the interaction effect between messages and temporal cues was significant. As shown in Table 7, the effect of temporal cues was significant when the message was b. Therefore, we conducted

Table 5 Summary of multiple comparison

Pair	Difference	T	df	P	Adjusted P
a – b	-1.0616	6.1396	361	0.0000	0.0000*
a – c	-1.8666	5.1914	361	0.0000	0.0000*
a – d	-0.9142	2.5426	361	0.0114	0.0343*
b – c	-0.8049	2.2300	361	0.0264	0.0791
d – c	-0.9524	1.9873	361	0.0476	0.0953
b – d	0.1474	0.4085	361	0.6832	0.6832

* Significantly different at $\alpha = 0.05$ level (*P < 0.05).**Table 6** Summary of two-way ANOVA for the interaction effect between message and temporal cue (except c-1 and d-1)

Source	SS	df	MS	F	P	η^2
Message	100.3838	1	100.3838	45.0834	0.0000**	0.1133
Temporal cue	68.3707	5	13.6741	6.1412	0.0000**	0.0771
Message $\times$ Temporal cue	24.9832	5	4.9966	2.2440	0.0499*	0.0282

* Significantly different at $\alpha = 0.05$ level (*P < 0.05).** Significantly different at $\alpha = 0.01$ level (**P < 0.01).**Table 7** Simple effects for the interaction between message and temporal cue

Source	SS	df	MS	F	P	η^2
Message at 1	24.1934	1	24.1934	10.8655	0.0011**	0.0273
Message at 2	0.2832	1	0.2832	0.1272	0.7216	0.0003
Message at 3	52.3488	1	52.3488	23.5104	0.0000**	0.0591
Message at 4	9.0822	1	9.0822	4.0789	0.0443*	0.0102
Message at 5	28.3642	1	28.3642	12.7386	0.0004**	0.0320
Message at 6	14.0299	1	14.0299	6.3010	0.0126*	0.0158
Temporal cue at a	9.5438	5	1.9088	0.8572	0.5102	0.0108
Temporal cue at b	89.5775	5	17.9155	8.0460	0.0000**	0.1011

* Significantly different at $\alpha = 0.05$ level (*P < 0.05).** Significantly different at $\alpha = 0.01$ level (**P < 0.01).**Table 8** Summary of multiple comparison (b-1, b-2, b-3, b-4, b-5, and b-6)

Pair	Difference	T	df	P	Adjusted P
b-2 – b-3	-2.1817	5.4135	311	0.0000	0.0000*
b-2 – b-5	-1.9586	4.8095	311	0.0000	0.0000*
b-1 – b-2	1.6336	3.9673	311	0.0001	0.0009*
b-3 – b-4	1.3397	3.1718	311	0.0017	0.0167*
b-3 – b-6	1.1731	2.8865	311	0.0042	0.0417*
b-4 – b-5	-1.1167	2.6187	311	0.0093	0.0926
b-2 – b-6	-1.0086	2.5512	311	0.0112	0.0926
b-5 – b-6	0.9500	2.3137	311	0.0213	0.1493
b-2 – b-4	-0.8420	2.0447	311	0.0417	0.2921
b-1 – b-4	0.7917	1.8378	311	0.0670	0.4022
b-1 – b-6	0.6250	1.5057	311	0.1332	0.5326
b-1 – b-3	-0.5481	1.2976	311	0.1954	0.7816
b-1 – b-5	-0.3250	0.7621	311	0.4466	1.0000
b-3 – b-5	0.2231	0.5337	311	0.5939	1.0000
b-4 – b-6	-0.1667	0.4015	311	0.6883	1.0000

* Significantly different at $\alpha = 0.05$ level (*P < 0.05).

the multi comparison for b- x ($x = 1, 2, 3, 4, 5, 6$). Table 5 illustrates the results. The result showed that there were significant differences between b-2 and b-1, which supports H4. However, no significant differences existed between b-3 and b-1, between b-4 and b-1, between b-5 and b-1, and between b-6 and b-1. Thus, H5 was not supported. Also, both H6 and H7 were not supported because no significant differences existed between b-5 and b-6, between b-3 and b-5, and between b-4 and b-6.

3.3.2 Proportion of Abandonments

Table 2 shows the respective number of subjects that completed and abandoned the experiment for each pattern. The proportions of abandonments for most patterns were lower than that for d-1, which was 81%. This result means that the feedback information was effective for preventing user abandonment. On the other hand, the difference depending on the pattern was large, which means that the difference in

the feedback information had a great influence on the proportion of abandonments. In particular, the proportion of abandonments for c-1 was 86%, which was higher than that for d-1, suggesting that inappropriate feedback information may increase user abandonment.

3.4 Discussion

3.4.1 Effect of Feedback Information

The experiment result showed that a-2 was the best performed pattern. As shown in Table 3, the average number of retransmissions for a-2 was 0.09, which was significantly smaller than 1.33 for d-1. Multiple comparisons showed that message (a) and temporal cue (2) had the effect of reducing the number of retransmissions. This result means that the feedback information can change user behavior.

The significant difference between message (a) and message (b) shown in Table 5 indicated that the difference in message affected the number of retransmissions. However, message (b) includes the phrase "try again", which may have affected more than the phrase "accesses are presently concentrated". Experiments with more subdivided patterns are expected to confirm the effect of these messages. The average number of retransmissions for message (d) was the largest. However, the negative effect of a system error message could not be confirmed by multiple comparisons.

According to temporal cues, the effect of a waiting bar was confirmed. However, the effect of an expected waiting time or a countdown timer could not be confirmed.

In addition, the 2-way ANOVA showed that the interaction effect of message and temporal cue was significant. However, Table 7 showed that the interaction effect was simple, that is, the effects of message (a) and temporal cue (2) masked the effects of other factors. This result means that these message (a) and temporal cue (2) are significantly more effective than other factors.

We did not focus to user abandonment before the experiment, but there was a large difference depending on the pattern. As shown in Table 2, there were no abandoned subjects in a-2, that is, the proportion of abandonments was 0%, which was dramatically less than that for d-1 (81%). Despite the small number of subjects, we can say that this extreme difference indicates that the feedback information influences the abandonment of users. The final goal of studies dealing with flash crowds is not to reduce the server load, but to prevent users from abandonment due to response delay caused by the server overload. Therefore, we consider that the effectiveness of the feedback information for user abandonment is very important.

3.4.2 Discussion on Social Aspects

Subject experiments showed that user behavior can be changed by providing feedback information. However, we cannot immediately conclude that it is desirable to use feedback information; some social or ethical problems apply.

One problem relates to the fact that the feedback information that effectively influences user behavior is not true information. As shown by the experiments, the message "Now processing" and the waiting bar were effective at changing subject behaviors. However, it is deceptive to display the message "Now processing" if the user's process is not actually being performed. As for the waiting bar, its length changes irrespective of the progress of the actual process. Although it is not explicitly stated, many users typically think that the motion of the waiting bar reflects the actual progress using the accepted analogy of progress bars. Therefore, it can be said that the waiting bar is implicitly cheating users. From an ethical viewpoint, this type of feedback information is obviously problematic. The service provider should provide the user with true information and the user should decide how to act upon that knowledge.

Another problem relates to the users not being informed about the fact that the information provided to them is fake. It is desirable for the provider of the Web service to disclose to users that the feedback information is fake and to let them decide whether to use the service or not. However, a dilemma exists in that this disclosure would completely nullify the effect of feedback information.

On the other hand, it can be regarded as acceptable to present fake information if it eventually leads to the user's profit. Even if the users know the true information, it is not always possible for them to choose the behavior that will benefit themselves. In fact, in the results reported in Sect. 3.3, many subjects who were provided "true" information (that accesses are concentrated) abandoned the service. Since flash crowds are a kind of social dilemma, individual selfish behavior of a user will damage the whole profit, eventually even damaging himself/herself. However, in general, users do not know how to trigger flash crowds or how they should act to prevent them. In this case, the feedback information that leads a user to behave in a socially desirable direction may be acceptable even if it is fake.

The fact that the feedback information is fake raises another concern. That is, if users notice that the feedback information is fake, its effect should be lost. For example, if an impatient user who resent the request ignoring the message "Now processing" received the response immediately, he/she learns that the message is fake, and thereafter, will continue to ignore the message. Such a situation is problematic from another perspective. That is, an impatient user will have an unfair advantage by sending more requests over the patient users, while a patient user becomes the victim of the fake feedback information. In order to prevent such a situation, the server should detect multiple requests issued by the same user and impose a penalty such as cancelling them. Though it consumes extra server resource, this cost is necessary to maintain users' trust in feedback information. It is also possible to implement clients not to send multiple requests.

On the other hand, patient users who obey feedback information should also be cared for. Though the "Now processing" message was very effective, this message implies

that the user is not required to resend his/her request. Hence, when displaying the “Now processing” message, the client should provide a function for the automatic retransmission of the user’s request after the flash crowd has disappeared.

Another kind of problem relates to competing service providers. It can be said that keeping users at a particular website with fake information unduly deprives them of the opportunity to select competing service providers. However, in reality, when competing websites exist, many users can access them simultaneously and select the one with the quickest response. Therefore, we do not believe that presenting information to keep users at a particular Web service is a fatal problem.

3.5 Limitations

We acknowledge that certain limitations make for cautious interpretation of our results and derived implications. First, all subjects are university students. Most of them were estimated to be 18 to 21 years old. Although they represent the target users of web services, one could collect additional data with various users of web services to improve the validity.

Second, the gender of the participants may be biased, which may have affected the result of the experiment. However, we do not think that the bias was so significant because the male to female ratio of the class from which participants were recruited is almost 1:1.

Third, our experiment was not conducted in a laboratory. It is possible that cheating was performed in the experiment. For example, the same person has participated in the experiment multiple times, participants have heard the purpose of the experiment in advance, and so on. However, given the lack of rewards, there seems to be no incentive to cheat in the experiment.

Fourth, variations of feedback information we examined in the experiment are limited. As mentioned in Sect. 2, various types of feedback information were examined in previous studies. Fortunately, we have found the effective feedback information pattern from the limited options. There might be more effective feedback information. We hope that future studies will clarify the effectiveness of various feedback information for flash crowd mitigation.

Fifth, it is uncertain whether user behavior can be changed by feedback information for Web services important to users. In our experiments, subjects easily abandoned the experiment because their motivation for completing the process with the target Web service was not high enough. However, it is difficult to conduct an experiment in which all subjects would require the target Web service.

Finally, the long-term effects of feedback information are also uncertain. In order to investigate the long-term effect, the subject needs to participate the same experiment repeatedly. However, it is impossible in reality because knowing the purpose of the experiment influences the behavior.

It may be possible to ascertain the real behaviors of users and long-term effects of feedback information by ana-

lyzing the access log of an actual Web service.

4. Stress Test

To examine how changes in user behavior due to feedback information help mitigate actual flash crowds, we conducted stress tests on a real server.

4.1 Methods

4.1.1 User Behavior Model

To perform the stress tests, it is necessary to model user behaviors and devise test scenarios based on them. To grasp the behavior of each user in the experiment in Sect. 3 in detail, we examined the time when each subject retransmitted the request and calculated the request retransmission interval. The request retransmission interval refers to the time between the previous transmission and the retransmission of the request. Figure 1 shows the distribution of retransmission intervals of the last retransmissions of each user and remaining retransmissions. The two distributions are clearly different. In the case of last retransmissions, the retransmission intervals were distributed mainly around 30 s, but many beyond 120 s were also noted. Conversely, in the case of the remaining retransmissions, the retransmission intervals were concentrated within 40 s, and those exceeding 40 s were few.

Observing these distributions, we considered that the typical user behavior was as follows. The users retransmitted their requests several times within a relatively short period. Then, some of them subsequently waited for a long time while others abandoned the experiment.

Based on these observations, we developed a model of the behaviors of users. The model is shown in Algorithm 1.

In this model, each user transmits n successive requests to the server during the process. The function *sendreq* is a function that sends a request. It returns *true* if the server responds that the request was successful. It returns *false* if there is a failure response or if there is no response after the timeout period (T_{out}) has elapsed. There is an operation waiting time (wait 1) before a request is transmitted, corresponding to the time during which he or she thinks and

Fig. 1 Distribution of retransmission intervals

Algorithm 1 User behavior model

Input: $rmax, type$

```

1: function  $S_{endreq}(r)$ 
2:   send  $r$  to server
3:   if reply to  $r$  is received within  $T_{out}$  seconds then
4:     if  $r$  succeeds then return true
5:     else return false
6:     end if
7:   else return false
8:   end if
9: end function

10: label mainloop:
11: for  $i \leftarrow 1$  to  $n$  do
12:   (wait 1) ▷ operation waiting time
13:   if  $S_{endreq}(r_i) = \text{true}$  then
14:     continue mainloop
15:   else if  $rmax \geq 1$  then
16:     for  $j \leftarrow 1$  to  $rmax$  do
17:       (wait 2) ▷ retransmission interval
18:       if  $S_{endreq}(r_i) = \text{true}$  then
19:         continue mainloop
20:       end if
21:     end for
22:   end if
23:   if  $type = A\text{-type}$  then ▷ the case of the A-type user
24:     return false ▷ The process failed.
25:   else ▷ the case of the W-type user
26:     (wait 3) ▷ last waiting period
27:     if  $S_{endreq}(r_i) = \text{true}$  then
28:       continue mainloop
29:     else return false ▷ The process failed.
30:     end if
31:   end if
32: end for

```

inputs data. Then $sendreq$ is called to send a request. If it returns *true*, the user goes to the operation waiting time for the next request. Otherwise, $sendreq$ is called again to resend the failed request. However, the number of retransmissions is limited to the predefined value ($rmax$). The retransmissions are spaced by the retransmission interval (wait 2). If all the retransmissions fail, users would behave differently according to the user type ($type$). An A-type user would immediately abandon the process, while a W-type user would see out the last waiting period (wait 3) and then call $sendreq$ to retransmit the request one more time. If it returns *false*, the W-type user would abandon the process. If a user abandons the process, he or she would not transmit subsequent requests. The process is successful only if all n requests succeed.

Since the model has not been validated, there is concern that it may behave differently from the actual user. In order to validate the model, we need to verify that the user behavior generated by the model matches that of actual users. However, it requires more experimental data. Constructing a valid user behavior model is left for future work.

We also recognize that the experimental situation in which users wait for minutes is uncommon. There are many references (e.g., [5]) that report that most users abandon their requests after seconds, not minutes. In common cases, Web users are very impatient because the site is not so im-

portant for them or there are some alternative sites. However, we believe that if the Web service is irreplaceable for the user, he/she can wait a long time. For example, in access concentration on the ticket website of 2020 Tokyo Olympics, it was reported that many people waited more than an hour [36].

4.1.2 Measure of Evaluation

For stress tests, the method of evaluating the seriousness of damage caused by flash crowds in each parameter setting should be considered. Obviously, the most serious case concerns the server going down. However, there are cases in which flash crowds become serious even when the server continues to function. If the proportion of A-type users (defined in Sect. 4.1.1) is large, and the number of retransmissions is small, server downtime may be avoided as a result of abandonment by many users. From the viewpoint of a business, user abandonment itself in flash crowds causes serious damage even if the server does not go down. To evaluate the seriousness of the damage quantitatively, we use the proportion of the number of abandonments to the total number of burst users (defined in Sect. 4.1.5). The higher the proportion, the more serious the damage of the flash crowd. However, if the proportion of A-type users is small, and the number of retransmissions is large, the server may go down. When the server goes down, it does not respond to any request and the stress test cannot be continued. This is clearly more serious than the case where the test is completed. Therefore, if the server goes down in a test, the proportion of abandonments of the test is recorded as 1 (100%). When calculating the average value of multiple tests, the average proportion is defined as 1 if the server goes down in at least one test.

4.1.3 Target System

The target system was a small website providing a hotel reservation service. It consisted of a single server running WordPress [37]. We used MTS Simple Booking [38], which is a WordPress plugin implementing the reservation service. A reservation process performed by a user of the website comprised five steps:

1. Calendar search.
2. Input of check-in date.
3. Input of arrival time.
4. Input of customer information.
5. Confirmation of reservation.

Each step involved the transmission of a request message to the website server.

4.1.4 Test Environment

Table 9 shows the specifications of the stress test environment.

The request sequence sent to the server was generated

Table 9 Stress Test Environment

Virtual Server	
Hypervisor	VMware ESXi 5.5
Number of CPUs	1 CPU
RAM capacity	2 GB
Database	MySQL 14
Maximum connections	150
Hosting Server	
CPU	Intel Xeon E5-2620
RAM	DDR4-2133
Storage	Intel SSD 530 Series

by an Apache JMeter [39], using a scenario based on the model described in Sect. 4.1.1.

4.1.5 User Arrival Model

To express the difference in server load between normal time and a flash crowd, we consider two types of users with different arrivals. Throughout the stress test, the users arrive at the system at a constant rate. We call such users ordinary users. In a certain period of time, a large number of users arrive at the system in addition to the ordinary users. We call this period the burst period, and such additional users are called burst users. Figure 2 shows an example of user arrivals in stress tests.

4.1.6 Parameter Setting

For each user, r_{max} is assigned a random value according to the exponential distribution with a fixed mean value. The mean value was given by a parameter ($mean_{r_{max}}$). As shown in Table 3, the shape of distribution of the number of retransmissions seem to be an exponential distribution rather than a normal distribution or a uniform distribution. Moreover, the random number generator according to the exponential distribution is easy to implement.

Each user is either an A-type user or a W-type user. The ratio of the number of A-type users to the total number of users was given by a parameter ($ratio_{Auser}$). If $ratio_{Auser} = 0.4$, there are 40% A-type users and 60% W-type users. T_{out} in List 1 was set to 10 s for our test.

The waiting times of each user were assigned a random value according to a uniform distribution. We investigated the distribution of retransmission intervals (wait 2) in the experiment. However, since we do not have enough data, it was difficult to fit them into probability distributions. Moreover, we do not have any data on the operation waiting time (wait 1) or the last waiting period (wait 3). Therefore, we used a uniform distribution to simplify the implementation of the test scenarios.

The ranges of the operation waiting time (wait 1) were determined by the particular operation (reservation step) being performed by the user. The ranges for reservation steps 1–5 were set to 0–0, 10–20, 5–10, 40–80, and 5–10 s, respectively. The range of the retransmission interval (wait 2) was given as a parameter ($range_{rint}$). The range of the last waiting period (wait 3) was set to 60–180 s.

Fig. 2 Example of user arrivals (denoted by arrows) in stress tests

Table 10 Parameter Setting

Pattern	$mean_{r_{max}}$	$range_{rint}$	$ratio_{Auser}$
a-2	0.09	10–50 s	0.0
d-1	1.33	10–50 s	0.81

In each test, the value of $mean_{r_{max}}$, $ratio_{Auser}$, and $range_{rint}$ was the same for the ordinary and burst users. The burst period begins 60 s after the start of the test and lasts for 60 s. During the burst period, burst users arrive at fixed intervals. The number of burst users was given by a parameter (num_{burst}). Ordinary users arrive according to the Poisson process, with an average interval of 1 s by default.

4.2 Results

The test was repeated 10 times for each parameter setting. Presented below are the respective average results for the different settings.

4.2.1 Effect of Feedback Information

First, we examined the effects of feedback information. We compare the best performed pattern in the subject experiment (a-2) and the pattern which does not display any feedback information (d-1).

Parameters were set as shown in Table 10. The value of $mean_{r_{max}}$ was set based on the average number of retransmissions of each pattern shown in Table 3. The value of $ratio_{Auser}$ was set based on the proportion of abandonments of each pattern shown in Table 2. The range $range_{rint}$ was set to reflect the distribution of the retransmission intervals. Figure 3 shows the distribution of retransmission intervals for a-2 and d-1 in the experiment. In the case of d-1, there are no requests retransmitted within 10 s, and the number of requests retransmitted at intervals of 50 s or more was relatively small. As for a-2, we could not estimate the range because there were only two retransmissions. Based on this, we set $range_{rint}$ to 10–50 s. It should be noted that this setting may cause shorter retransmission intervals than actual ones for both a-2 and d-1. The stress test was then conducted by varying num_{burst} .

Figure 4 shows the proportion of abandonments of the burst users. As can be observed, the proportion of abandonments of a-2 was always lower than that of d-1. The proportion of abandonments for d-1 increased rapidly with num_{burst} , while that for a-2 was suppressed to almost 0 at $num_{burst} \leq 200$. We found that the damage caused by a flash crowd tends to increase rapidly when the number of burst

Fig. 3 Distribution of retransmission interval

Fig. 4 Proportion of abandonments of burst users

users exceeds a certain threshold. Feedback information can raise this threshold, thereby reducing the probability of increased flash crowd damage.

The difference was maximum when $num_{burst} = 300$, which was 30%. For $num_{burst} \leq 200$, the rate of increase in the proportion of abandonments of a-2 was lower than d-1. However, for $num_{burst} \geq 300$, there were no significant differences between the rates of increase for a-2 and d-1. This result shows that the effect of feedback information is limited. That is, when the number of burst users was high, the extent of damage was not reduced with the provision of feedback information. In other words, feedback information has little effect on reducing damage once a serious flash crowd occurs.

These findings mean that the effect of feedback information was similar to that of the conventional approaches (e.g. [31]), which raise the threshold for occurring flash crowd by increasing the maximum capacity.

Figure 5 shows the average number of retransmissions per request for the burst users. The number of retransmissions for a-2 was always lower than that for d-1. The reduction rate was maximum when $num_{burst} = 700$, which was 41%. This result is not surprising given that $mean_{rmax}$ in a-2 is much smaller than that in d-1. However, in terms of the server load, the reduction of requests by more than 40% is not small. This result shows that feedback information can actually reduce the server load in flash crowds.

4.2.2 Effect of Web Service Importance

In the parameter settings in Sect. 4.2.1, $mean_{rmax}$ is a rel-

Fig. 5 Average number of retransmissions

Fig. 6 Proportion of abandonments of burst users (Web service important to users)

atively small value. This reflects the fact that many subjects abandoned the service after a few retransmissions in the experiments described in Sect. 3. This may be because the experimental Web service was not important to the subjects. In contrast, if the Web service was very important to them, they would not immediately abandon it and would retransmit the request multiple times. To simulate such a situation, we performed a stress test with $mean_{rmax} = 10$ and $ratio_{User} = 0$. For Web services important to users, users would not stop retransmission if they received feedback information. Therefore, the impact of feedback information, if any, would appear as the change in retransmission interval. We performed tests by changing $range_{rint}$.

The result is shown in Fig. 6. As num_{burst} increases, the proportion of abandonments increases rapidly and reaches 1. This means that since most users did not abandon the service, once the server load starts to increase, users' retransmissions did not stop until the server went down. However, the value of num_{burst} for which the proportion of abandonment reaches 1 depends on $range_{rint}$. This result suggests that if it is possible to change the retransmission interval with feedback information, it may be possible to prevent the server from going down in some cases.

4.3 Threats to Validity and Limitations

The parameters used in the stress test were determined based on the results of subject experiments. That is, the limitations mentioned in Sect. 3.5 threaten the validity of results of the stress tests. Moreover, the values of parameters such as operation waiting time, and last waiting period were deter-

mined arbitrarily because we do not have underlying data. In addition, the probability distribution that the random numbers used in the stress test follow was not selected on solid basis. These facts are also threat to validity of the results.

Limitations include not considering the impact of the network. When discussing the Web service performance, networks problems should be taken into consideration in addition to the server load and processing power. It should be noted that networks problems may prevent the delivery of the planned feedback information to users. The present study, however, focused on a flash crowd due to server problems alone. The effect of networks will be considered in a future work.

5. Conclusions

In the present study, we considered the mitigation of the damage caused by flash crowds by providing feedback information to users. Subject experiments showed that, by providing feedback information, the average number of request retransmissions decreased from 1.33 to 0.09, and subjects abandoning the server decreased from 81% to 0%. This confirmed that feedback information can affect the behaviors of Web service users. Stress tests showed that the average number of retransmissions decreased by 41%, and the proportion of abandonments decreased by 30%. These results confirmed that changes in user behavior due to feedback information reduced the probability of increased damage due to flash crowds.

On the other hand, stress tests revealed that the effect of feedback information was similar to the conventional methods. That is, it can raise the threshold for occurring flash crowd, while it has little effect on reducing damage once a serious flash crowd occurs. Considering that the presentation of feedback information can be implemented at relatively low cost, our method does not replace the conventional methods, but should be used in conjunction with them.

In a future study, we will consider incorporating client-side assistance. When flash crowd occurs, the server cannot afford to conduct extra processing. Therefore, the function of displaying feedback information should be served mainly by the Web client. We consider that it can be implemented using modern Web technologies. Moreover, as mentioned in Sect. 3.4.2, the client should care for impatient and patient users. We plan to implement these aspects and consider more advanced cooperation between the client and server when dealing with flash crowds.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP26280032 and JP17H01737.

References

- [1] T. John, "Star Wars: Rogue One Pre-Sales Are Crashing Online

- Ticket Site," Nov. 2016, <http://fortune.com/2016/11/28/star-wars-story-rogue-one-tickets/>, accessed Feb. 17. 2020.
- [2] D. Ackerman, "iPhone X preorders are a mess – let's fix the system," Oct. 2017, <https://www.cnet.com/news/preordering-is-broken-and-heres-how-to-fix-it-iphone-x/>, accessed Feb. 17. 2020.
- [3] X. Chen and J. Heidemann, "Flash crowd mitigation via adaptive admission control based on application-level observations," *ACM Trans. Internet Technology*, vol.5, no.3, pp.532–569, Aug. 2005.
- [4] I. Ari, B. Hong, E.L. Miller, S.A. Brandt, and D.D. Long, "Managing flash crowds on the internet," *Proc. 11th IEEE/ACM Intl. Symp. on Modeling, Analysis and Simulation of Computer Telecommunications Systems*, pp.246–249, 2003.
- [5] F.F.H. Nah, "A study on tolerable waiting time: How long are web users willing to wait?," *Behaviour & Information Technology*, vol.23, no.3, pp.153–163, 2004.
- [6] Y.G. Lee, A.N. Chen, and T. Hess, "The online waiting experience: Using temporal information and distractors to make online waits feel shorter," *J. Association for Information Systems*, vol.18, no.3, p.1, 2017.
- [7] A.N. Chen, Y. Lee, and Y. Hwang, "Managing online wait: Designing effective waiting screens across cultures," *Information & Management*, vol.55, no.5, pp.558–575, July 2018.
- [8] C.H. Chen and S. Li, "The effect of visual feedback types on the wait indicator interface of a mobile application," *Displays*, vol.61, Article No. 101928, Jan. 2020.
- [9] J. Hohenstein, H. Khan, K. Canfield, S. Tung, and R. Perez Cano, "Shorter wait times: The effects of various loading screens on perceived performance," *Proc. 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp.3084–3090, May 2016.
- [10] U. Söderström, M. Bååth, and T. Mejttoft, "The Users' Time Perception: The effect of various animation speeds on loading screens," *Proc. 36th European Conference on Cognitive Ergonomics*, pp.1–4, 2018.
- [11] H. Tada, M. Murata, and M. Aida, "Web Service Flash Crowd Mitigation Using Feedback to Users," *2017 IEEE 31st Int. Conf. Advanced Information Networking and Applications (AINA)*, pp.213–219, IEEE, 2017.
- [12] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS attacks from flash crowds using flow correlation coefficient," *IEEE Trans. Parallel Distrib. Syst.*, vol.23, no.6, pp.1073–1080, June 2012.
- [13] S. Behal, K. Kumar, and M. Sachdeva, "Discriminating Flash Events from DDoS Attacks: A Comprehensive Review," *IJ Network Security*, vol.19, no.5, pp.734–741, 2017.
- [14] R. Tandon, A. Palia, J. Ramani, B. Paulsen, G. Bartlett, and J. Mirkovic, "Defending Web Servers Against Flash Crowd Attacks," *2019 IEEE 27th Int. Conf. Network Protocols (ICNP)*, pp.1–2, IEEE, 2019.
- [15] Y. Baryshnikov, E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. Yimwadsana, "Predictability of web-server traffic congestion," *Proc. 10th Intl. Workshop on Web Content Caching and Distribution*, pp.97–103, IEEE, 2005.
- [16] A. Marnerides, D. Pezaros, and D. Hutchison, "Flash crowd detection within the realms of an Internet service provider (ISP)," *Proc. 9th Annual Postgraduate Symp. on the Convergence of Telecommunications, Networking and Broadcasting*, 2008.
- [17] B. Li, G.Y. Keung, S. Xie, F. Liu, Y. Sun, and H. Yin, "An empirical study of flash crowd dynamics in a p2p-based live video streaming system," *Proc. IEEE GLOBECOM 2008*, pp.1–5, 2008.
- [18] B. Zhang, A. Iosup, J. Pouwelse, and D. Epema, "Identifying, analyzing, and modeling flashcrowds in bittorrent," *Proc. 11th IEEE Intl. Conf. on Peer-to-Peer Computing*, pp.240–249, 2011.
- [19] P. Wendell and M.J. Freedman, "Going viral: Flash crowds in an open CDN," *Proc. ACM SIGCOMM 2011*, pp.549–558, 2011.
- [20] J. Ge, Y. Tian, L. Liu, R. Lan, and X. Zhang, "Understanding e-commerce systems under massive flash crowd: Measurement, anal-

- ysis, and implications,” *IEEE Trans. Services Computing* (Early Access), 2017.
- [21] L. Xie, P. Smith, D. Hutchison, M. Banfield, H. Leopold, A. Jabbar, and J.P. Sterbenz, “From detection to remediation: A self-organized system for addressing flash crowd problems,” *Proc. IEEE ICC* 2008, pp.5809–5814, 2008.
 - [22] Y. Chen, B. Zhang, C. Chen, and D.M. Chiu, “Performance modeling and evaluation of peer-to-peer live streaming systems under flash crowds,” *IEEE/ACM Trans. Netw.*, vol.22, no.4, pp.1106–1120, Aug. 2013.
 - [23] K. Kong and D. Ghosal, “Mitigating server-side congestion in the Internet through pseudoserving,” *IEEE/ACM Trans. Networking*, vol.7, no.4, pp.530–544, Aug. 1999.
 - [24] D. Rubenstein and S. Sahu, “Can unstructured P2P protocols survive flash crowds?,” *IEEE/ACM Trans. Networking*, vol.13, no.3, pp.501–512, June 2005.
 - [25] A. Chandra and P. Shenoy, “Effectiveness of dynamic resource allocation for handling internet flash crowds,” TR03-37, Department of Computer Science, University of Massachusetts, USA, 2003.
 - [26] J. Elson and J. Howell, “Handling Flash Crowds from Your Garage,” *Proc. USENIX Annual Technical Conf.*, pp.171–184, June 2008.
 - [27] A. Ali-Eldin, O. Seleznev, S. Sjöstedt-de Luna, J. Tordsson, and E. Elmroth, “Measuring cloud workload burstiness,” *Proc. 7th IEEE/ACM Intl. Conf. on Utility and Cloud Computing*, pp.566–572, 2014.
 - [28] U. de Paula Junior, L. Drummond, D. de Oliveira, Y. Frota, and V.C. Barbosa, “Handling flash-crowd events to improve the performance of web applications,” *Proc. 30th Annual ACM Symp. on Applied Computing*, pp.769–774, April 2015.
 - [29] Y. Niu, B. Luo, F. Liu, J. Liu, and B. Li, “When hybrid cloud meets flash crowd: Towards cost-effective service provisioning,” *Proc. IEEE INFOCOM* 2015, pp.1044–1052, 2015.
 - [30] Y. Niu, F. Liu, X. Fei, and B. Li, “Handling flash deals with soft guarantee in hybrid cloud,” *Proc. IEEE INFOCOM* 2017, pp.1–9, 2017.
 - [31] A.F. Baarzi, T. Zhu, and B. Urgaonkar, “BurScale: Using Burstable Instances for Cost-Effective Autoscaling in the Public Cloud,” *Proc. ACM Symposium on Cloud Computing*, pp.126–138, Nov. 2019.
 - [32] B.G.C. Dellaert and B.E. Kahn, “How tolerable is delay?: Consumers’ evaluations of internet web sites after waiting,” *J. Interactive Marketing*, vol.13, no.1, pp.41–54, 1999.
 - [33] S. Galbraith, “The cost of poor website performance,” March 2006, <https://www.red-gate.com/simple-talk/dotnet/net-tools/the-cost-of-poor-website-performance/>, accessed Feb. 17. 2020.
 - [34] J. Brutlag, “Speed matters for Google web search,” June 2009, <https://services.google.com/fh/files/blogs/google.delayexp.pdf>, accessed Feb. 17. 2020.
 - [35] M. Ghafurian, D. Reitter, and F.E. Ritter, “Countdown timer speed: A trade-off between delay duration perception and recall,” *ACM Trans. Computer-Human Interaction (TOCHI)*, vol.27, no.2, pp.1–25, March 2020.
 - [36] R. Takahashi, “Visitors to Tokyo Olympics website face long waits as lottery winners announced; overseas sales begin,” June 2019, <https://www.japantimes.co.jp/news/2019/06/20/national/2020-tokyo-olympics-website-backs-ticket-winners-announced-overseas-sales-begin/>, accessed Feb. 17. 2020.
 - [37] WordPress.org, “Blog Tool, Publishing Platform, and CMS — WordPress,” <https://wordpress.org/>, accessed Feb. 17. 2020.
 - [38] MT Systems, “MTS Simple Booking,” <http://mtssb.mt-systems.jp/>, accessed Feb. 17. 2020.
 - [39] The Apache Software Foundation, “Apache JMeter,” <http://jmeter.apache.org/>, accessed Feb. 17. 2020.

of Education, Kyoto University of Education. In April 2010, he became an Associate Professor. Since April 2016, he has been a Professor of the Faculty of Education, Kyoto University of Education. His research interests include computer communication networks and distributed systems. He is a member of the IEEE, IEICE and IPSJ.

1999, he became a Professor of Osaka University, and since April 2004, he has been with the Graduate School of Information Science and Technology, Osaka University. From October, 2017, he is a Director, Office of Management and Planning, Osaka University. Also, he is now a Vice-Research Director of Asia Pacific Institute of Research. He has contributed more than nine hundred papers to international and domestic journals and conferences. His research interests include computer network architecture and brain- and bio-inspired networking. He is a member of the IEEE, ACM, and IEICE.

Metropolitan University since April 2007. His current interests include analysis of social network dynamics and distributed control of computer communication networks. He received the Best Tutorial Paper Award and the Best Paper Award of IEICE Communications Society in 2013 and 2016, respectively. He also received the IEICE 100-Year Memorial Paper Award in 2017. He is a member of the IEEE, ACM and the Operations Research Society of Japan.

Harumasa Tada received the M.E. and D.E degrees in Information and Computer Sciences from Osaka University, Japan, in 1995 and 1998, respectively. From October 1997 to March 2002, he was an Research Associate at the Graduate School of Engineering Science, Osaka University. From April 2002 to March 2006, he was an Research Associate at the Graduate School of Information Sciences and Technology, Osaka University. From April 2006 to March 2010, he was an Lecturer at the Faculty

Masayuki Murata received the M.E. and D.E degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined IBM Japan’s Tokyo Research Laboratory, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with the 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

Masaki Aida received his B.S. degree in Physics and M.S. degree in Atomic Physics from St. Paul’s University, Tokyo, Japan, in 1987 and 1989, respectively, and his Ph.D. in Telecommunications Engineering from the University of Tokyo, Japan, in 1999. In April 1989, he joined NTT Laboratories. From April 2005 to March 2007, he was an Associate Professor at the Faculty of System Design, Tokyo Metropolitan University. He has been a Professor of the Graduate School of Systems Design, Tokyo