PERFORMANCE ESTIMATION AND IMPROVEMENT OF SERVERS USING THE BATCH SERVICE THROUGH A MULTI-CHANNEL ARCHITECTURE

Ying-Wen Bai, Yung-Sen Cheng, Cheng-Hung Tsai and Yin-Sheng Lee
Department of Electronic Engineering, Fu Jen Catholic University, Taipei, Taiwan
E-mail: bai@ee.fju.edu.tw

ABSTRACT
With respect to the popularization of the Internet, as the client-server mode in general is used now, thus the server system needs to utilize some methods to maintain its performance. Recently it has become possible to apply the batch service mechanism to the server system to effectively reduce the emergence of high system utilization. But using the batch service mechanism for processing requests on a server system causes a relatively higher system response time when the setting of the batch size is increased. Hence in this paper we propose a multi-server architecture method utilizing the batch service as an improvement over the performance of the single-server architecture using the batch service. We adopt the queueing model method to analyze the system performance of the single-server and the multi-server when using the batch service. According to the adjustment of the batch size we estimate the beneficial number of service channels and achieve a reduction of the high response time and the high system utilization.

KEY WORDS
Batch service, client-server, multi-server, cluster, queueing model, and system performance

1. Introduction
With the coming of “Web 2.0” and the technology of the Internet developing ever and ever faster, more and more applications and services are available through the Web site. Most users access information for their daily life from the Web server through the Internet. Fig. 1 shows the Internet environment of the client-server and the very important role played by the performance of the server system. The quality of this performance will the length of time influence how much users will access this server system, so the improvement of the system’s performance is an important target in the future [1]. The three primary factors for judging system performance are high availability, linear scalability and high cost-effectiveness [2]. So one of mentioned methods can improve the performance of a server system by using schedule algorithms or queueing theory to distribute jobs appropriately and thus deal with the waiting requests efficiently [1][3].

By utilizing queueing theory the “batch service” operates in such a way that the queue accumulates events until reaching a pre-set number, and then the system processes all events in the queue as a batch [4]. This method can be applied in the architecture of a client-server [5-7] as shown in Fig. 2. The purpose of adopting the batch service mechanism is to reduce both the switching frequency of state transitions of the server system and the high system utilization which is caused by the random arrival of events or packets from the Internet [8].

Except for reducing the system utilization, as the batch service mechanism has to reach the lower boundary of the set batch size to start, thus the system will produce a relatively higher response time [9]. To improve the response time of a single-server using the batch service we adopt the architecture of the multi-server using the batch service. This architecture distributes the arrival requests uniformly and reduces the system mean response time effectively, thus improving the overall performance.
In this paper we use a queueing model to represent a server system to analyze the performance of a single-server using the batch service and to observe the variation of mean response time and mean utilization. We also analyze the performance of the multi-server system to see whether there is a reduction of the mean response time or not. From the derived equation of the mean response time we are able to estimate the beneficial number of service channels.

The organization of this paper is as follows. In Section 2 we study the system performance of a single-server system when using the batch service. In Section 3 we adopt the multi-server architecture to improve the system mean response time when using the batch service. In Section 4 we analyze the beneficial number of service channels when adopting the multi-server using the batch service. In Section 5, we draw the conclusions.

2. Performance Estimation of a Single-server System Using the Batch Service

The performance of a server system depends on its mean response time and mean utilization. First we use the queueing theory to analyze the performance of a single-server using the batch service. The queueing model is as shown in Fig. 3. We assume that $\lambda$ represents the equivalent arrival rate of requests from users and $\mu$ the batch service rate of a server system respectively.

The state transition diagram of a single-server using the batch service is shown in Fig. 4 where $i$ represents the batch size of requests. As the number of requests reaches $i$, the server system starts the batch service, therefore, process all requests which are waiting and accumulated in the queue.

2.1 Mean System Response Time for a Single-server System Using the Batch Service

Following the state transition diagram of Fig. 4 we adopt the Little’s Formula to calculate the mean response time [2] as shown in Eq. (1). We assume that $E_{sb}(t)$ is the mean response time of a single-server system when using the batch service, and $i \times \lambda_{eq}$ represents the amount of the accumulated requests in the queue. The assumption is that $\mu_b$ needs to be greater than $i \times \lambda_{eq}$ to avoid the emergence of blocking.

$$E_{sb}(t) = \frac{1}{\mu_b - (i \times \lambda_{eq})}$$ (1)

We plug the specific parameter values into Eq. (1) and use a mathematics software package to show the variation of the mean response time when using the batch service for a single-server system as shown in Fig. 5. From the figure we see that as the batch size increases, the mean response time of the single-server system gradually also increases. If the mean response time is higher, the performance of the single-server system worsens, and conversely, it improves.

2.2 Mean System Utilization for a Single-server System Using the Batch Service

Then we also use queueing theory to calculate the mean utilization [2] of a single-server system when using the batch service as shown in Eq. (2). We assume that $\rho_{sb}$ is the mean utilization for a single-server system, and $\frac{\lambda_{eq}}{\mu_b}$ multiplied by $\frac{1}{i}$ to obtain the mean utilization for a single-server system when setting the batch size $i$. 
\[
\rho_{sh} = \frac{1}{l} \times \frac{\lambda_{eq}}{\mu_b} 
\]

(2)

We plug the specific parameter values into Eq. (2) and use a mathematics software package to show the variation of the mean utilization when using the batch service for a single-server system as shown in Fig. 6. From the figure we can learn that if the batch size is bigger, the mean utilization of single-server system is improves. If the mean utilization is lower, the performance of the single-server system is better, and conversely, it worsens.

The batch service is applied in the single-server system for processing requests so as to reduce the system utilization, but it increases the system response time relatively when it increases the batch size. Because the system response time influences the system performance, in the next section we will investigate how to reduce the system response time.

3. Performance Improvement through the Multi-channel Architecture

Using the batch service mechanism for distributing requests for a single-server system reduces the emergence of higher system utilization, but it also causes a relative increase in the response time when we increase the batch size. We therefore propose the architecture of the multi-server system to reduce the system response time of the single-server system when using the batch service.

The cluster is a common method used to solve the problem of heavy Web requests [10-11]. The cluster is composed of two or more servers including both a dispatcher and several servers as shown in Fig. 7. These provide a higher availability for requests. Users send requests to the cluster through the Internet using a laptop, a PC or a smart phone. All requests are sent to the dispatcher first, which redirects them to one of the servers in the cluster. After the server has received the requests from the dispatcher, the server provides the dispatch function to the users directly and completes the final step of the whole service.

We want to reduce the load of a single-server using the batch service to improve the system response time. So we adopt the architecture of the cluster to improve the performance of the single-server system using the batch service mechanism as shown in Fig. 8. First the requests from users are distributed to the multi-server system via the dispatcher. Then the multi-server system processes these requests by using the batch service. Because the requests are distributed equally to the multi-server system for processing, the load of each server is reduced to improve the overall system performance.

3.1 Performance Evaluation of the Multi-channel Architecture by the Mean System Response Time

To observe the improved performance of adopting the multi-channel architecture we use the queueing model to represent the cluster including the dispatcher and the multi-server using the batch service as shown in Fig. 9. First we divide the system into two parts for examination; one is the dispatcher, and the other is the multi-server. \( \lambda_{eq} \) and \( \mu_b \) represent the equivalent arrival rate of requests from users and the batch service rate of each server system respectively. And we assume that \( \mu_d \) is the service rate of the dispatcher, and that \( n \) represents the number of servers or the number of service channels. \( \frac{\lambda_{eq}}{n} \)
indicates that all requests are dispatched uniformly to each server via the dispatcher.

We calculate the mean response time of the overall system to find the performance of the cluster using the batch service. The mean response time of the overall system is divided into two parts; one is the mean response time of the dispatcher, and the other is the mean response time of the multi-server. Because the multi-server processes requests in parallel, the mean response time of the overall system is the summation of the mean response times of the dispatcher and a server with the batch size \( i \) as shown in Fig. 10 and Eq. (3).

\[
E_{mb}(t) = E_d(t) + E_{ms}(t) = \frac{1}{\lambda_d - \lambda_{eq}} + \frac{1}{\mu_b - \left(\frac{\lambda_{eq}}{i} \times \frac{1}{n}\right)} \tag{3}
\]

We plug the specific parameter values into Eq. (3) and use a mathematics software package to show the variation of the mean response time as shown in Fig. 12. In this figure we see that if the batch size exceeds 12 and the number of service channels exceeds at least two, the mean response time of the cluster system is lower than the mean response time of the single-server system. Thus if the set batch size is bigger, the system can increase the number of service channels to improve the performance when adopting the batch service to be a mechanism of processing requests.
3.2 Mean System Utilization of the Multi-channel Architecture

We will discuss the mean utilization of each multi-server using the batch service when adopting the multi-channel architecture. We assume that $\rho_{mb}$ is the mean utilization of each server with the batch size $i$ using the batch service as shown in Eq. (4).

$$\rho_{mb} = \frac{1}{i} \times \frac{n}{\mu_b}$$  \hspace{1cm} (4)

Then we plug the specific parameter values into Eq. (4) and use a mathematics software package to show the variation of the mean utilization when using the batch service for each server of the multi-channel architecture as shown in Fig. 13. From the figure we see that if the number of service channels $n$ is increased, the mean utilization of each server system improves. If the mean utilization is lower, the performance of the overall system is better, and if conversely, it worsens.

![Fig. 13 The mean utilization when using the batch service for each server of the multi-channel architecture](image)

To observe the performance variation when adopting the multi-channel architecture using the batch service, we will look at the relationship between the mean response time and the mean utilization. From Eqs. (3) and (4) we take the specific parameter values and use a mathematics software package to show the variation and relationship between both as shown in Fig. 14. That figure shows that if the number of service channels increases, the performance using the batch service improves. Thus when using the batch service we can adjust the number of servers based on the system performance. In the next section we will study how to determine the number of service channels which is most advantageous.

4. Estimation of the Beneficial Number of Service Channels

To obtain the number of service channels which can improve the mean response time for a single-server system using the batch service, we assume that $E_{sb}(t)$ is the mean response time of a single-server system using the batch service, $E_{mb}(t)$ is the mean response time of a cluster system using the batch service and “$n$” represents the number of service channels. To improve the performance of a single-server system using the batch service we will calculate “$n$”, which is needed for $E_{mb}(t)$ to be less than $E_{sb}(t)$. The values of $i$ and of $n$ must be natural numbers. A suitable value for “$n$” can be calculated as shown in Eq. (5).

$$\frac{1}{\mu_d - \lambda_{eq}} + \frac{1}{\mu_b - i \times \lambda_{eq}} < \frac{1}{\mu_b - i \times \lambda_{eq}}$$

$$\Rightarrow n > \frac{i \lambda_{eq} (\lambda_{eq} - i \lambda_{eq} + \mu_b - \mu_d)}{-i \lambda_{eq} \mu_b + \mu_b^2 + i \lambda_{eq} (\lambda_{eq} - \mu_d)}$$

$$\therefore n = \left[ \frac{i \lambda_{eq} (\lambda_{eq} - i \lambda_{eq} + \mu_b - \mu_d)}{-i \lambda_{eq} \mu_b + \mu_b^2 + i \lambda_{eq} (\lambda_{eq} - \mu_d)} \right]$$

We plug the specific parameter value into Eq. (5) to obtain the relationship between the batch size and the number of service channels as shown in Fig. 15. In the figure, if the batch size equals 11, the more suitable value for “$n$” is 4, if the batch size equals 12, the more suitable
value for “n” is 3, but if the batch size equals 13, the
two more suitable value for “n” is 2, which respectively
improve the performance of a single-server system using
the batch service. By the derived equation we also decide
the best number of service channels necessary to maintain
the system performance using the batch service.

![Graph showing the relationship between batch size and number of service channels]

The graph shows the relationship between the batch size and the
number of service channels.

Thus if the batch size will be set for a large number, we
can consider adopting the multi-channel architecture to
improve the performance of a single-server when using
the batch service mechanism to process requests.

5. Conclusion

Adopting the batch service mechanism for processing
requests for a server system effectively prevents the
emergence of higher system utilization. When there is a
large number of requests in a single-server using the batch
service, and the setting of the batch size is enlarged, may
have a longer response time for a single-server system.

In this paper we propose a multi-server architecture using
the batch service as an improvement over the performance
of the batch service applied to a single-server system. By
the derived equation we estimate the number of service
channels needed to achieve a better performance than a
single-server system using the batch service. By adjusting
the batch size we achieve a reduction of the high response
time and the high system utilization. In the future one can
apply the mechanism of the multi-server using batch
service in the Web-based multimedia system connected to
networks to verify the reduction of both system utilization
and response time.

References

[1] Daniel A. Menascé and Virgilio A. F. Almeida,
Capacity Planning for Web Services - Metrics, Models, and

[2] H. Krawczyk and A. Urbaniak, Allocation Strategies of
User Requests in Web Server Clusters, Proceedings of the
International Conference on Parallel Computing in
Electrical Engineering (PARELEC ´02), Warsaw, Poland,

- Queueing Theory and Performance Evaluation, Third

[4] Jung-Ren Hu and Ying-Wen Bai, Power Efficiency and
Latency for a Semi-batch Power Management Model in a
Palmtop Multimedia Terminal, Proceedings of the IASTED
International Conference on Modelling and Simulation (MS

[5] Vrinda Tokekar, Sanjiv Tokekar, and A.K.Ramani,
Study of Reneging Behavior in Batched Multimedia Service,
Proceedings of the 4th ACS/IEEE International Conference
on Computer Systems and Applications (AICCSA 2006),
Dubai/Sharjah, UAE, March 8-11, 2006, 806-813.

[6] Gennaro Boggia, Pietro Camarda, Luigi Mazzeo, and
Marina Mongiello, Performance of Batching Schemes for
Multimedia-on-Demand Services, IEEE Transactions on

Scheme for Multicast Video-On-Demand Systems, IEEE
Transactions on Broadcasting, Vol. 47, No. 1, March 2001,
66-70.

[8] Donald Gross and Carl M. Harris, Fundamentals of
1998.

with Batch Service, IEEE Transactions on Communications,

[10] Ying-Wen Bai and Yi-Chao Wu, Web Delay Analysis
and Reduction by Use of Load Balancing of a Dispatcher-
based Web Server Clusters, Proceedings of the 21st IASTED
International Conference on Applied Informatics, Innsbruck,
Austria, Feb. 10-13, 2003, 541-546.

for the Load Balance of a Web-Cluster System, Proceedings
of the 15th IASTED International Conference on Parallel
and Distributed Computing and Systems (PDCS’ 2003),
Marina Del Rey, CA, USA, Nov. 3-5, 2003, 250-255.