Reduced Burst Score Aggregation in Suppressing the Effect of Delayed-hit Caching

Feri Fahrianto[1,2] and Noriaki Kamiyama[3]

[1]Graduate School of Engineering, Fukuoka University, Japan
[2]Faculty of Science and Technology, Syarif Hidayatullah Jakarta State Islamic University, Indonesia
[3]College of Information Science and Engineering, Ritsumeikan University, Japan

1 Introduction

An online caching system can improve network connectivity by shortening the content distance as close as possible to the requester. Moreover, various online-caching technology, such as Content Delivery Networks (CDN) and Information-Centric Networking (ICN), implement caching systems to improve the Quality of Experience (QoE) of users. A caching strategy is compulsory to be implemented for achieving a high-cache-hit ratio, i.e., cache replacement algorithms such as First In First Out (FIFO), Least Recently Used (LRU) and etc. Unfortunately, these cache replacement algorithms are unable to endure the decline due to delayed-hit caching, as reported in [1, 2, 3]. The cache hit ratio suffered from a significant drop of about 30% and 40% for content download time from the original server 100 and 1000 times from inter-request time, as we reported in [4]. To cope with this problem, we proposed a method that implements a specific cache replacement algorithm to suppress the delayed caching effect in the online caching system, namely Burst Score Aggregation (BSA) cache replacement algorithm. The BSA is required to track all burstiness of each unique content that creates inefficient processing time. Therefore, this paper introduced Reduced Burst Score Aggregation (R-BSA) that can improve the processing time without losing its performance to maintain the cache-hit ratio.

2 Reduced Burst Score Aggregation

The burst score of the particular content can be measured by comparing the occurrences of that content between sampling time and elapsed time. Hoonlor et al. in [4] introduced the burst score of a particular content x for the specific sampling period, which is similar to t_{fetch} in an online-caching server, which can be defined by

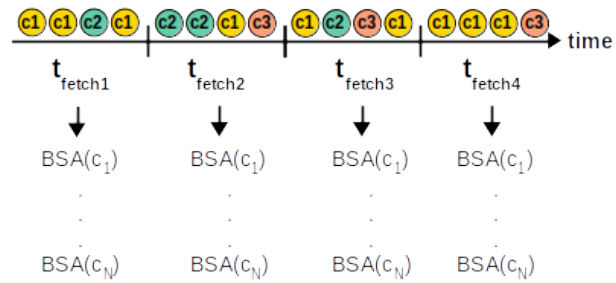
$$Burst(x, t_{fetch}) = (\frac{E_{t_{fetch}}}{E} - \frac{1}{T}), \quad (1)$$

where E_t is the total number of occurrences of event $x \in \{c_1, .., c_N\}$ in sampling time interval of t_{fetch} and E is the total number of occurrences of $x \in \{c_1, .., c_N\}$ in total elapse time T. The set of $\{c_1, .., c_N\}$ express the number of N total unique contents.

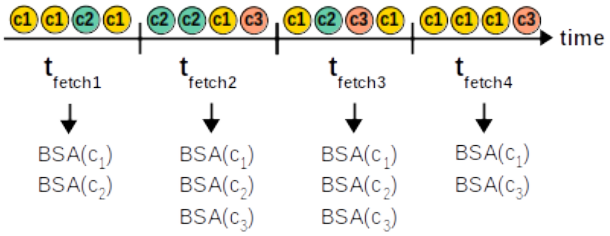
To calculate the BSA, we use a formula that can be defined by

$$BSA(x, t_{fetch}, T) = \sum_{i=1}^{M} Burst(x, t_{fetch}), \quad (2)$$

where M is the total number of t_{fetch} cycles until the elapse time T or in other word $M = T/t_{fetch}$. Therefore, the BSA tracks all historical burst level of all unique contents that requires a lot of calculation time. In order to reduce the calculation time, we modify the requirement of x in the equation 1. The value of x become the set of x within t_{fetch} instead of $x \in \{c_1, .., c_N\}$. It can reduce the number of BSA calculations from N to S where $S << N$ speeds up the overall processing time.



(a) BSA



(b) R-BSA

Figure 1: Comparison of number BSA calculation between original BSA and R-BSA

Figure 1 shows the comparison of calculation processes between BSA and R-BSA. The original BSA cache replacement algorithm demands all unique contents calculation in every content download time, t_{fetch}. This requirement causes an overhead calculation of as many as total unique contents, N, even though not all the content occurred within t_{fetch} as seen in Figure 1(a). On the other hand, the R-BSA only requires calculating the unique content that occurred in the interval t_{fetch}, i.e., in the t_{fetch1}, t_{fetch2}, t_{fetch3}, t_{fetch4}, and the R-BSA count the value of BSA for 2, 3, 3, 2 respectively as seen in Figure 1(b). This mechanism can save overall processing time.

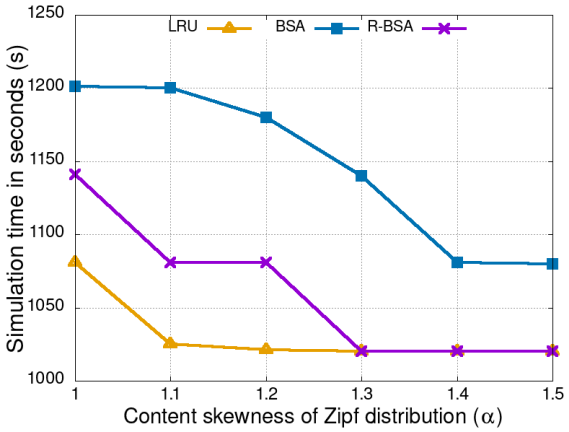


Figure 2: Comparison of simulation time between LRU, BSA, and R-BSA

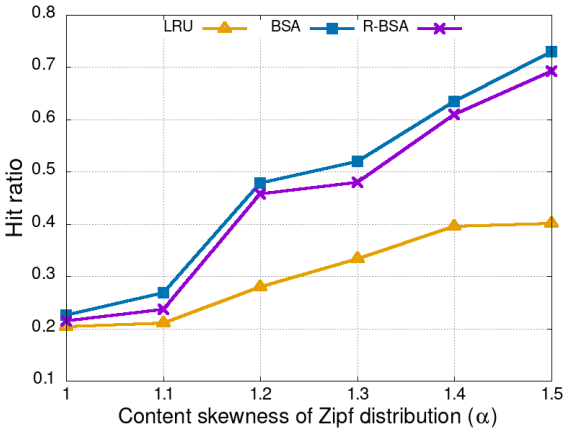


Figure 3: Comparison of hit ratio between LRU, BSA, and R-BSA

3 Numerical evaluation

An experiment using a simulator was conducted to measure the hit-ratio in the cache with different skewness parameter (α). A multi-threaded application was constructed using Python 3.8 to run a simulation about 100.000 request that was equipped with a timer to capture the overall processing time. Table 1 shows the major hardware and software parameters used in the simulation.

Table 1: Setting values of main parameters

Paramater	Value
Number of content items	1000
Cache size	10
Content skewness (α)	1 - 1.5
Content download time (t_{fetch})	1 s
Inter request arrival time ($t_{arrival}$)	0.01 s
CPU	AMD Ryzen 9 3900X
OS	Ubuntu 20.04 LTS
Programming language	Python 3.8

Figure 2 shows that the BSA cache replacement algorithm was about 15% and 20% longer on average in overall processing time compared to R-BSA and LRU, respectively. When the skewness parameter, α, increased, the overall processing time was also constantly decreasing in all cache replacement algorithms. Furthermore, the overall processing time in LRU and R-BSA was equal, starting at α=1.3. This was because R-BSA only counts a set of unique content that occurred in the interval of t_{fetch} due to the higher of α causing less content variety in the interval of t_{fetch}. The gap between BSA and R-BSA located in the higher α confirmed the reduction in the processing time, especially because the content with zero occurrences was omitted in the BSA calculation.

Figure 3 shows the hit ratio of LRU, BSA, and R-BSA cache replacement algorithm. In general, all three cache replacement algorithms experienced an increase in hit ratio when the skewness parameter of Zipf distribution, α, was constantly increasing. The gap of hit ratio between BSA and R-BSA was about 5% on average, where R-BSA under-performed against BSA. The hit ratio of R-BSA was lower than BSA because the R-BSA only counted the non-zero occurrence of contents within the interval of t_{fetch} that caused non-subtraction of BSA value for zero occurrences of contents. As a result, there was a lagging process for updating the ranking of content based on the BSA database whenever the burstiness of particular content was changed. This created an invalid eviction process whenever a cache miss occurred in the cache.

4 Conclusion

The R-BSA cache replacement algorithm can reduce the overall processing time while maintaining its performance, especially the hit ratio. The decline of the hit ratio in R-BSA compared to BSA is because of the removal of zero occurrences of content requests within an interval of content download time. The omission created a lagging ranking process in the BSA database for the eviction process in the cache. In the future, we will investigate a method to reduce the ranking processing time in the cache of the R-BSA cache replacement algorithm.

Acknowledgement This work was supported by JSPS KAKENHI Grant Number 18K11283 and 21H03437, and the Ministry of Religious Affairs of the Republic of Indonesia cooperated with Indonesia Endowment Fund for Education (LPDP).

References

[1] Andrea Detti, Lorenzo Bracciale, Pierpaolo Loreti, Nicola Blefari Melazzi, "Modeling LRU Cache with Invalidation," Comput. Netw. 134, C (April 2018), Pages 55–65, 2018.

[2] N. Atre, J. Sherry, W. Wang, and D. S. Berger,"Caching with Delayed Hits," In Proceedings of SIGCOMM 2020, ACM, New York, NY, USA, 2020.

[3] S. Selvakumar, Swarup Kumar Sahoo, Vijay Venkatasubramani, "Delay Sensitive Least Frequently Used Algorithm for Replacement in Web Caches," Computer Communications, Volume 27, Issue 3, Pages 322-326, 2004.

[4] F. Fahrianto and N. Kamiyama, "Impact of Delayed Caching on Hit-ratio of ICN Router," IEICE 2022 General Conference, BS-3-5, Online, Mar. 2022

[5] A. Hoonlor et al., "An Evolution of Computer Science Research," Communications of the ACM, 56(10), pp. 74-83, Oct. 2013.

[6] F. Fahrianto and N. Kamiyama, "Suppressing Effect of Delayed Caching in ICN Router by Burst Score Aggregation", IEICE 2022 Society Conference, B-14-1, Online, Sep. 2022