International Journal of Internet, Broadcasting and Communication Vol.14 No.2 17-22 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.2.17

IJIBC 22-2-3

Effect of Caching and Prefetching Techniques to Accelerate Channel Search Latency in IPTVs

Hyokyung Bahn

Professor, Department of Computer Engineering, Ewha University, Korea
bahn@ewha.ac.kr

Abstract

Due to the recent advances in high-speed communication technologies as well as the easy production of high-quality video contents, IPTV is becoming increasingly popular. Meanwhile, as the number of IPTV channels increases, channel search time to find the desired channel keeps increasing. In this paper, we discuss how to improve the channel search latency in IPTV, and introduce caching and prefetching techniques that are widely used in memory management systems. Specifically, we adopt memory replacement, prefetching, and caching techniques in IPTV channel search interfaces and show the effectiveness of these techniques as the number of channels are varied.

Keywords: *IPTV, Channel Search, Replacement, Caching, Prefetching*

1. Introduction

With the rapid advances in high-speed communication technologies and the easy production and distribution of high-quality video contents, the demand for IPTV is continuously increasing [1, 2, 3]. Unlike terrestrial and satellite broadcasting, IPTV is not limited by frequency bandwidth for each channel, so it can provide hundreds of channels through IP networks [4]. Although the increased number of channels has the advantage of providing various contents to users, there is a problem in that the search time required to find the desired content is greatly increased. For this reason, IPTV users spend a lot of time in searching for channels or selectively watch some limited number of channels [4]. If the set-top box receives additional contents from channels other than the one currently being watched, the channel switching time can be shortened. However, due to limited network resources, the amount of contents that a set-top box can receive per unit time is limited, so it is not easy to reduce the delay experienced during fast channel switching. There is an analysis that the time required to switch channels exceeds one minute in the worst case, implying that it is important to shorten the delay in channel switching in order to improve service satisfaction of IPTV users [5]. Thus, there have been many studies to reduce the latency of channel switching [2, 4, 6].

In this paper, we analyze how caching and prefetching can be utilized as a way of reducing the channel search delay of IPTVs. Caching and prefetching have been widely used in memory management of computer systems, and this paper shows that it can be adopted in IPTV channel management systems [9, 10]. We also

Manuscript Received: February. 2, 2022 / Revised: February. 6, 2022 / Accepted: February. 9, 2022
Corresponding Author: bahn@ewha.ac.kr
Tel: +82-2-3277-2368, Fax: +82-2-3277-2306
Professor, Department of Computer Engineering, Ewha University, Korea

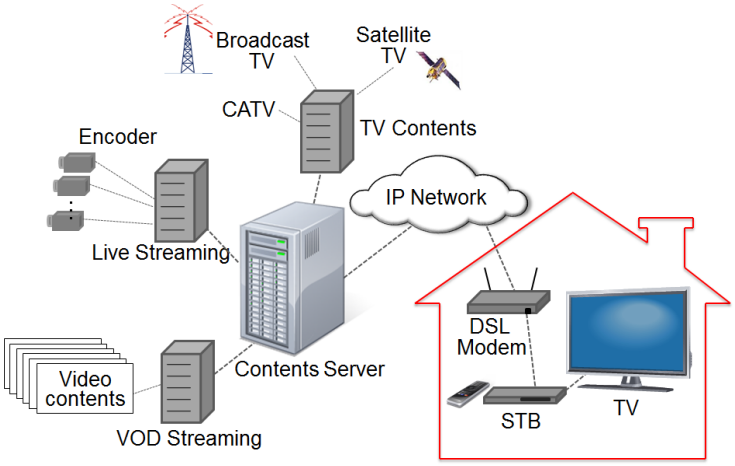

Figure 1. A typical IPTV system architecture with set-top box and modem connected to IP networks.

analyze how effective these techniques are when adopted in a wide spectrum of IPTV situations.

2. Related Work

Li et al. present a channel recommendation framework, which maintains a short list of channels for each user based on deep neural network models to capture the static and dynamic user switching behaviors [2]. Unlike such extensive optimizations, we focus on simple and efficient techniques used in memory management. Joo et al. propose an IPTV channel control algorithm that aims at reducing network delay as well as video decoding delay [3]. To do so, they co-optimize the number of broadcasting channels seeking high network utilization and video decoding delay by considering a new encoding structure with additional I-frames. Our study does not focus on video encoding/decoding issues as we are interested in caching and prefetching of channels. Lee et al. present a new channel arrangement scheme that interleaves channels based on their priority [4]. Specifically, the most preferred channel is fixed in the center, and channels with odd rankings are positioned on its left side, whereas even rankings on its right side. The problem of this scheme is that the priorities of the channels should be known beforehand, which is different from the problem domain of our study that allows for the dynamic change of channels' priorities as the watching behavior changes.

3. Caching and Prefetching in IPTV Channel Search

In memory management systems, various techniques such as replacement, prefetching, and caching are used to manage limited memory spaces efficiently. In this section, we analyze how these techniques can be utilized in IPTV channel search interfaces.

3.1 Using Memory Replacement Techniques in Channel Search

Memory replacement refers to the policy that selects data to be discarded from memory when there is no free memory space. To decide data to maintain or not in memory, replacement policies analyze the usage patterns of each memory data. The LRU (Least Recently Used) and LFU (Least Frequently Used) algorithms are widely used replacement policies [7]. LRU sorts data in memory based on their access time, and evicts data that are least recently used. LRU utilizes the property that recently used data are more likely to be reused in the near future. LFU manages data in memory based on their access count, and when free memory space is

needed, the data with the least access frequency is evicted. This makes use of the property that more popular data used in the past are more likely to be used again.

Now, let us discuss how to apply the properties used in memory replacement to the channel search method. When the principle of LRU is used, the "most recent channel first search" that traverses the most recently viewed channel first can be used when searching for a channel, implying that users are likely to watch the recently watched channel again. Also, if we adopt the principle of LFU, the "most popular channel first search" that traverses the channel with the most viewing count first can be used when searching for a channel, implying that users are likely to watch channels they have viewed many times in the past. Channel search methods aforementioned are advantageous as they can reduce the number of channels passed through until the desired channel is found.

3.2 Using Prefetching Techniques

A prefetching technique refers to a method of fetching not only the currently requested data but also data that are likely to be used in the near future when requested data are fetched from storage to memory [7]. When prefetched data are actually used, storage access time can be saved. However, as the memory capacity is limited, current data in memory may be discarded if prefetching data are increased. Thus, prefetching is effective only for data highly likely to be actually used.

When prefetching is used in channel search, the contents of a channel can be received before users actually requests that channel, thereby reducing the delay time that occurs when switching to the corresponding channel. This implies that prefetching of channels that users are likely to request or pass through in the near future will be effective in reducing the channel search time. We can consider two types of prefetching, "popular channel prefetching" and "adjacent channel prefetching" for IPTV channel management systems.

Popular channel prefetching makes use of the property that channels watched frequently in the past are likely to be watched again in the near future. Thus, in popular channel prefetching, channels are sorted based on the number of views in the past, and those with the highest number of views are preferentially prefetched. Popular channel prefetching is effective when we use the popular channel first search method as it is highly likely to pass through popular channels in this search method. In addition, even though channel search methods other than the popular channel first search is used, it is likely that a popular channel will be the final target channel, so that the channel switching time can be reduced.

Adjacent channel prefetching fetches channels that are adjacent to the currently watched channel in advance during a channel search operation. Such channels are highly likely to be visited in searching the target channel, and thus it is effective in reducing the latency of channel search. That is, adjacent channel prefetching is effective because at least one prefetched channel is essentially visited during the target channel search regardless of the channel search method employed with it.

Prefetching is effective in reducing the channel flipping latency whereas channel search methods can reduce the number of channels passed through. Thus, it will be more effective if the two techniques are used together.

3.3 Utilizing Caching Techniques

Caching is a technique that stores a subset of hot data to fast media, thereby reducing the access latency if

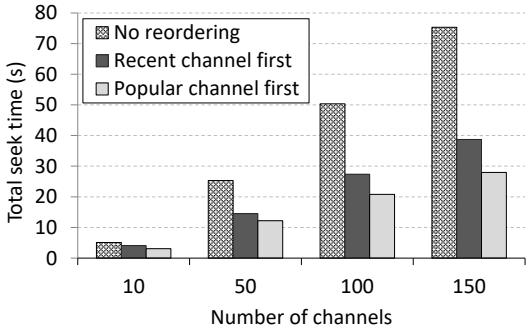

Figure 2. Total seek time for channel search methods as the number of channels are varied.

the cached data are actually used [7]. In IPTV channel search, we can make use of the idea of caching by storing a certain channel contents in the close location from the user's set-top box in order to reduce the channel search latency when switching to the corresponding channel. The prefetching technique introduced in the previous section can also be explained as the principle of caching, and we can consider the caching of popular channels in the low-resolution preview mode instead of the high-resolution mode. Also, classifying channels into hot and cold channels, and caching hot channels in a low resolution mode can accelerate the channel switch latency. We identify hot and cold channels based on the user's history of channel selection behaviors in terms of watching frequency and recency similar to caching in memory management [11]. By doing so, it not only distinguishes channels watched frequently and those that are not so, i.e. hot/cold channels, but also distinguishes those channels that are hot but are getting colder, and those that are cold but are getting hotter. Visiting hot channels first can further reduce the channel search latency.

4. Simulation Experiments

In this section, we validate the effectiveness of the caching and prefetching techniques through trace-driven simulations. The popularity of each channel was modeled based on the Zipf distribution, which is a representative model that reflects the biased distribution [8]. Such distributions are widely used to model not only the popularity of TV channels, but also the number of visits to web pages and the number of books borrowed from the library. The viewing probability of the *i*-th popular channel in the Zipf distribution is modeled as follows.

$$P_i = \frac{(1/i)^{\theta}}{\sum_{k=1}^{n}(1/k)^{\theta}} \tag{1}$$

where n is the total number of channels, and θ ($0 \le \theta \le 1$) is a parameter that determines the popularity bias of channels. When θ *is* 0, all channels are equally popular. As θ increases, the skewness of channel's popularity grows, and the popularity bias is the largest when θ becomes 1. In our experiment, we investigate and analyze the result when θ is 1, which represents the original Zipf distribution. The length of the trace, i.e. the total number of channels (including revisits) over time, is set to 10,000, and the experiments are conducted as a function of the number of channels.

Figure 2 shows the total search latency of the "recent channel first" and "popular channel first" in comparison with the numerical channel search without reordering as the number of channels is varied. As shown in the figure, the two search methods outperforms the numerical channel search without reordering. When comparing the two methods, "popular channel first" performs better than "recent channel first," and

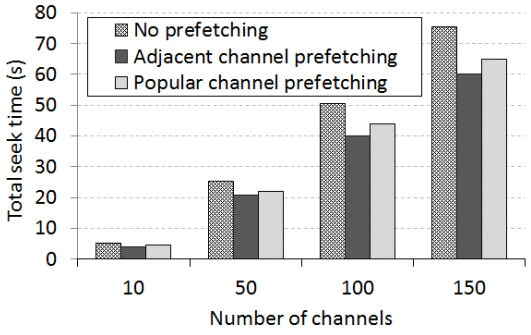

Figure 3. Total seek time for prefetching methods as the number of channels are varied.

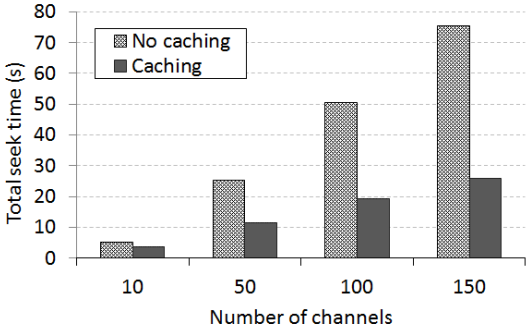

Figure 4. Total seek time for caching and no caching as the number of channels are varied.

their performance improvement against the numerical channel search is 53.3% and 39.2%, respectively. Also, the performance improvement becomes large as the number of channels increases.

Figure 3 shows the total search latency of the "popular channel prefetching" and "adjacent channel prefetching" in comparison with no prefetching. As shown in the figure, adopting prefetching exhibits better results than channel search without prefetching in all cases. When comparing the two prefetching methods, "adjacent channel prefetching" outperforms "popular channel prefetching", and this trend apparently appears as the number of channels increases. This is because adjacent channels are essentially visited while searching the desired channel in any case. The performance improvement of "adjacent channel prefetching" and "popular channel prefetching" is 20.1% and 13.2%, respectively, compared to the channel search without prefetching.

Figure 4 shows the results of channel search with caching and without caching as the number of channels increases. As shown in the figure, caching improves the channel search time consistently, as switching to cached channels saves the channel flipping time. Specifically, caching improves the channel search time by 52.9% on average compared to the channel search without caching.

5. Conclusion

In this paper, we introduced memory management techniques that can be adopted in IPTV channel search acceleration and analyzed their effectiveness. Specifically, we adopted memory replacement, prefetching, and caching techniques in IPTV channel search and evaluated them based on trace-driven simulations. Our empirical results showed that "popular channel first search" and "adjacent channel prefetching" perform better than "recent channel first search" and "popular channel prefetching" respectively. We also showed that caching

can improve the channel search latency significantly in IPTV channel search.

Acknowledgement

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub) and the ICT R&D program of MSIT/IITP (2020-0-00121, development of data improvement and dataset correction technology based on data quality assessment).

References

[1] T. Kim, H. Bahn, "Implementation of the Storage Manager for an IPTV Set-top Box," IEEE Trans. Consumer Electronics, Vol. 54, No. 4, pp.1770-1775, 2008.
DOI: http://doi.org/10.1109/TCE.2008.4711233

[2] G. Li, L. Qiu, C. Yu, H. Cao, Y. Liu, C. Yang, "IPTV Channel Zapping Recommendation With Attention Mechanism," IEEE Trans. Multimedia, vol. 23, pp. 538-549, 2021.
DOI: http://doi.org/10.1109/TMM.2020.2984094

[3] H. Joo, H. song, D. Lee, I. Lee, "An Effective IPTV Channel control Algorithm Considering Channel Zapping Time and Network Utilization," IEEE Trans. Broadcasting, Vol. 54, no. 2, pp.208-216, 2008.
DOI: http://doi.org/10.1109/TBC.2008.915767

[4] E. Lee, J. Whang, U. Oh, K. Koh, H. Bahn, "Popular channel concentration schemes for efficient channel navigation in internet protocol televisions," IEEE Trans. Consumer Electronics, vol. 55, no. 4, pp. 1945-1949, 2009.
DOI: https://doi.org/10.1109/TCE.2009.5373754

[5] Agilent Technologies, "Ensure IPTV Quality of Experience," White Paper, 2005.

[6] H. Lee, S. Lee, H. Kim, H. Bahn, "Personalized recommendation schemes for DTV channel selectors," IEEE Trans. Consumer Electronics, vol. 52, no. 3, pp. 1064-1068, 2006.
DOI: http://doi.org/10.1109/TCE.2006.1706508

[7] O. Kwon, H. Bahn, K. Koh, "Popularity and prefix aware interval caching for multimedia streaming servers," Proc. IEEE CIT Conference, pp. 555-560, 2008.
DOI: http://doi.org/10.1109/CIT.2008.4594735

[8] G. K. Zipf, Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology, Addison Wesley Press, 1949.

[9] H. Bahn, H. Lee, S. Noh, S. Min, K. Koh, "Replica-aware caching for web proxies," Computer Communications, vol. 25, no. 3, pp. 183-188, 2002.
DOI: https://doi.org/10.1016/S0140-3664(01)00365-6

[10] R. Karedla, J.S. Love, B.G. Wherry, "Caching strategies to improve disk system performance," IEEE Computer, vol. 27, no. 3, pp. 38-46, 1994.
DOI: https://doi.org/10.1109/2.268884

[11] S. Lee, H. Bahn, S. Noh, "CLOCK-DWF: A write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures," IEEE Trans. Computers, vol. 63, no. 9, pp. 2187-2200, 2014.
DOI: https://doi.org/10.1109/TC.2013.98