Article Title: An Observed Study on Improved Caching by Adaptive and Partial Aggressive Prefetching

Authors: S. P. Ponnusamy, E. Karthikeyan

Abstract: This paper gives the observed study to investigate the advantages over adaptive prefetching with proxy caching for large multimedia streaming. The adaptive and partial prefetching method fetch the media chunks dynamically based on the user access pattern in the proxy servers and updates the current access pattern in to media server. The study analyzed the proxy caching study of iRcache and applied adaptive prefetching with partial aggressive technique on caching to reduce the latency. Our study comprises with adaptive predictor engine and partial aggressive prefetcher in the proxy server. The study shows that 35% of prefetching will give the optimum performance on prefetching.

Keywords: Proxy Caching, Prefetching, Web latency, multimedia streaming.

I. INTRODUCTION
The requirement of multimedia based streaming like Video-on-Demand (VoD) by the users increased rapidly in the recent years. This increased the Internet traffic, huge consumption of bandwidth due to large size of media, and maximum latency between client and server due to the above factors while transferring media to client from media servers. The transferring of media content is a big challenge since the Internet provides the best-effort service which does not provide the guaranteed best quality of service. To improve the performance on multimedia content delivery, the content provider places the replication of media using Content Delivery Network (CDN) and many researchers actively work on proxy caching approach. The implementation of CDN is very expensive and difficult to replicate in all the places which are closer to the clients.

The proxy caching for web content such as HTML files contain text and images has been provided the better performance and the same technique cannot be offered for the multimedia content delivery due to the large size. For that reason, the media objects need to be divided into smaller chunks of independent content. The researchers provide many caching schemes such as fragmented proxy caching[6], cooperative proxy caching[7], segment based caching[8] and so on. The cache replacement polices also involved in the above factor to provide frequent requested fragments. Caching schemes don't always guarantee the continuous delivery because the portions to be viewed may not be cached in the proxies on time.

To further improve the performance of continuous delivery, prefetching schemes are used i.e., fetch-ahead the media before the request of the media by the client. Many prefetching schemes have been proposed such as adaptive prefetching [2], aggressive prefetching [11], optimal prefetching [9], distributed prefetching [10] and so on. The prefetching schemes overcome the limitation of cache mechanisms through preprocessing contents before a user request comes. Prefetch schemes expect future requests through log file analysis and prepare the expected requests before receiving it. Compared with web cache schemes, prefetch schemes focus on the spatial locality of objects when current requests are related with previous requests.

However, despite these benefits, three difficulties prevent prefetch schemes from being exploited in web cluster systems[2]. First, it is difficult to find which objects are related with the incoming requests. At the server side, access patterns are dynamic because of the web cache mechanism. Second, it is difficult to find an optimal prefetch rate. Too aggressive prefetch schemes may hurt overall performance due to the shortage of memory. Finally, a prefetch scheme in a system should be considered along with an efficient resource management. The inappropriate resource management drains the resource of one backend server, while other backend servers have the available resource. To overcome these difficulties, we propose an adaptive and partially aggressive web prefetch scheme over proxy cache. The adaptive prefetching scheme defines which media chunks to be prefetched and adopts the dynamic change in media access pattern in server. The partial aggressive prefetching technique used to prefetch the media chunks in an interleaved manner.

II. RELATED WORK
The prefetching mechanisms are used to try to avoid the discontinuity of streaming media delivery by prefetching the portions to be viewed while clients are viewing current positions. Heung Ki Lee et al. [2] proposed the aggressive prefetching model with three components; Double Prediction-by-Partial-Match Scheme (DPS), Adaptive Rate Controller (ARC) and Memory Aware Request Distribution (MARD). The results produced 10% of performance improvement on average in various workloads.
Junli Yuan et al. [11] projected the aggressive prefetching scheme aims to reduce the startup delay for first-time accessed objects by aggressively prefetching them in advance. To ensure the prefetching accuracy, they introduced the assistance of the media servers by having the servers to locate the most popular media objects and provide such information to proxies as hint for prefetching. To handle the large size problem, they adopted the segment prefetching mechanism. Trace-driven simulation results show that this scheme can effectively reduce the ratio of delayed requests by up to 38% while introduce very marginal increase in traffic.

Domenech et al. [3] studied the impact of the web architecture on the limits of latency reduction. They concluded that latency reduction depends on the predictor location: it can be reduced by 36%, 54%, and 67% when the predictor is located at the server side, client, or proxy, respectively. Latency reductions higher than 90% could be obtained if the predictor works collaboratively at different elements of the architecture.

Balamash et al. [4] proposed a mathematical model for a web prefetching architecture. Their results showed that prefetching was profitable even with the presence of a good caching system. Ossa et al. [5] focused on how to implement web prefetching in real environments making it compatible with commercial products and standard protocols.

This paper studies the upper bounds in latency savings by adaptive partial aggressive prefetching, assuming real conditions analyzed by IRCache [12] only with caching and analyzing how these conditions impact on the design of our prefetching architecture.

### III. SYSTEM MODEL

Web prefetching is split in two main components: a predictor engine, and a prefetching engine. The former predicts the next user accesses by using previous user information. In our architecture, the adaptive predictor and prefetcher are located at the proxy server. The predictor collects the media access pattern from the media server and adopts that current change in the access pattern into exiting pattern which is available in proxy server. The prefetcher works based on the partial aggressive manner to prefetch the media chunks in interim fashion. This architecture is shown in fig.1.

From the fig.1., the proxy server will take care of predicting and prefetching the media chunks. Normally, the proxy server cache the media which is already streamed to one of its clients and served latter if another client is requested the same media. The proxy allocated some memory for the caching. In our architecture, the proxy also uses the cache memory for storing the prefetched media chunks. The adaptive predictor frequently gets updates from media server and updates its prediction for future.
IV. ANALYSIS

The experiments are performed using traces logged by IRCache server bo2.is.ircache.net from October, 1st 2009 to December, 31st 2009. For our experiment, we consider the following factors:

- Number of objects requested
- TCP hit Rate
- Total object served in MBs
- Object Served from cache in MBs

We define a prediction algorithm as an adaptive predictor with three main properties: i) it never provides wrong hints, ii) it can keep only one updated hint in its memory for the media iii) it updates the media server whenever it sense a new hint.

According to property (i), the predictor never gives wrong hint about the media prediction and get success always. The algorithm also predict how much the user will view the content of the media. The property (ii) works that the predictor maintains table for each media in its memory which has high priority in the access pattern. The predictor engine also gets updates from media server frequently to update its hit entry. This preserves the memory in proxy server by avoiding all hints update as separate entries. The property (iii) supports the server in backward updation of hint when a proxy server sensed any new request from the user. This will help the server to improve the hint pattern.

The predictor engine predicts the which media should be prefetched and the prefetching engine only prefetches the required media chunks for play of media without buffering . So the predictor engine must be carefully designed with the percentage of prefetching. To achieve this we prepared the four level of prefetching percentage. The following figures (a), (b), (c) and (d) are used to predict our prefetching consideration and compare the object served totally and object served from cache.

The figure (c) gives the glue that the object needs to be served more from proxy to provide continuous play of the media. It shows very little amount of media chunks are served from cache and most part of the media chunks are fetched directly from the server during the user watching media. This cause irritation to the user due to buffering and playing.

The figure (e) shows the various level of prefetching percentage on the uncached object of media server. The prefetcher chooses the media chunks based on the adaptive predictor engine results given to the prefetcher engine.
The 40% of prefetched media chunks are used lesser than the unused media chunks. So it wastes the memory of the proxy server. This used and unused media chunks also helps the adaptive predictor engine to update the user access pattern into media server.

Due to this unnecessary prefetched media chunks, the predictor engine and prefetching engine overloaded to maintain the dynamic change in prediction and prefetching. This extra prefetching media chunks also affects the replacement strategy, because, these media chunks are recently prefetched and these chunks cannot be replaced from the cache memory from the proxy server.

The figure (f) and (g) shows the used and unused media chunks by the user while played the media from the prefetched amount media chunks. The graphs say that the usage of 25%, 30% and 35% amount of prefetched media chunks gives the good results on used media chunks over unused media chunks. This is happened due to the user dynamic movement on the media chunks while watching.

The figure (h) shows the average used and unused media chunks of three months media reference by the user. Due to the more percentage of media chunks prefetched, in some places, the average percentage of unused media chunks are upper hand than the average percentage of used media chunks.

V. CONCLUSION

Our study is performed on traces provided by IRCache server bo2.is ircache.net from the October, 1st 2009 to December, 31st 2009. In our architecture, the proxy server contains the adaptive predictor engine and prefetching engine (partial aggressive prefetcher) to provide the optimum continuity on play in the media player and yield minimum latency with backward updatability. The adaptive predictor engine dynamically updates the hint entries in its table and prefetcher fetches the media chunks in continuous and interleaved manner.

Our studies analyzed on various percentage level of percentage and the 35% of prefetching on the uncached media chunks yields the optimum performance. There are three reasons for the improved performance on our prefetching architecture; (i). it provides the optimum prediction in the proxy server, (ii). it prefetch the media chunks partially in continuous and interleaved manner, (iii). it provides backward updatability to the server. We are also plan to study the prefetching scheme on dynamic seek support in both forward and backward direction with distributed proxy server.

ACKNOWLEDGEMENT

The authors would like to thank IRCache community for providing support for do the study and also extend the grateful to the Adhiparasakthi Engineering College, Melmaruvathur, Tamilnadu and Bharathiar University, Coimbatore, Tamilnadu for providing sufficient infrastructure to bring out this study.

REFERENCES


S. P. Ponnusamy
Research Scholar
Bharathiar University
Coimbatore, Tamilnadu, India
sppomns2k1@rediffmail.com

Dr. E. Karthikeyan
Assistant Professor
Govt. Arts College,
The Niligiris, Tamilnadu, India
e_karthi@yahoo.com