On the Potential of Recommendation Technologies for Efficient Content Delivery Networks

Mohamed Ali Kaafar
INRIA, France
NICTA, Australia
mohamed-ali.kaafar@nicta.com.au

Shlomo Berkovsky
NICTA
Australia
shlomo.berkovsky@nicta.com.au

Benoit Donnet
Université de Liège
Belgium
benoit.donnet@ulg.ac.be

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

During the last decade, we have witnessed a substantial change in content delivery networks (CDNs) and user access paradigms. If previously, users consumed content from a central server through their personal computers, nowadays they can reach a wide variety of repositories from virtually everywhere using mobile devices. This results in a considerable time-, location-, and event-based volatility of content popularity. In such a context, it is imperative for CDNs to put in place adaptive content management strategies, thus, improving the quality of services provided to users and decreasing the costs. In this paper, we introduce predictive content distribution strategies inspired by methods developed in the Recommender Systems area. Specifically, we outline different content placement strategies based on the observed user consumption patterns, and advocate their applicability in the state of the art CDNs.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Distributed Networks

General Terms

Algorithms, Performance

Keywords

Content placement, personalization, CDN, recommendation technologies, in-network learning

1. INTRODUCTION

The tremendous growth of Internet content has stimulated the development of adaptive information access and filtering techniques, such as Recommender Systems [1]. Recommendation algorithms have been widely studied in the communities of information retrieval, machine learning, and data mining [2]. Due to their commercial potential, recommender systems are extensively deployed by the online industry, and the level of their maturity raises the question of exploiting the recommendation technologies beyond the traditional user interaction and information access contexts.

On the flip side, the proliferation of content delivery networks (CDNs) [3] has created a thrust among the research community and industry to investigate issues related to adaptivity in content delivery and distribution. In this paper, we examine the potential of recommendation technologies being embedded into the core of the infrastructure based CDNs, content servers. Relying either on content-based or collaborative filtering recommendation techniques, we discuss scenarios where the adaptivity has the potential to improve the quality of service and reduce the costs of content distribution. In essence, we firstly consider the use case of content placement for *cold items* introduced to a CDN, and advocate how content-based recommendations can underpin the decisions pertaining to the placement of the newly injected content. Secondly, we discuss the content placement strategies of already existing *warmed items* and outline how collaborative recommendations can predict content consumption and influence the content placement decisions.

The rationale of the proposed techniques relies on a simple yet altruistic intuition: aggregated user behavior related to the consumption of content, if learned correctly, can lead to optimized content placement strategies, and, in turn, to improved service and reduced costs. Recent works suggest that geolocation affects user-generated content consumption and its popularity [4, 5]. Another study suggests that user behavior in consuming video content reflects peculiar patterns depending on several parameters, one of them being the users' geolocation that represents a key differentiator between groups of users [6]. Here, we focus on infrastructure-based CDNs consisting of dedicated servers distributing content to users and do not assume any particular distribution of the servers in a CDN. We postulate that adaptive content placement strategies that take into account the diversity of users and their content consumption patterns, can potentially yield cost-effective and high-quality large-scale CDNs.

The remainder of this paper is organized as follows: Sec. 2 presents the necessary background for this paper; Sec. 3 discusses our assumptions and notations; Sec. 4 introduces recommendation techniques for CDNs; finally, Sec. 5 summarizes the paper and outlines future research directions.

2. BACKGROUND

In this section, we briefly introduce CDNs with focus on infrastructure-based networks (Sec. 2.1). Then, we overview the state of the art recommendation techniques (Sec. 2.2).

2.1 Content Distribution Networks

Infrastructure-based CDNs, consisting in servers-provisioned delivery networks, replicate content over several

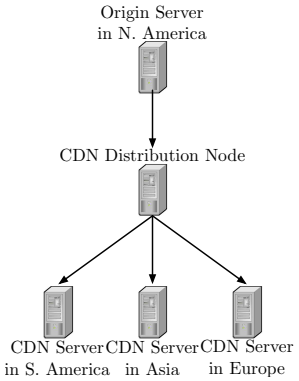

Figure 1: Typical CDN architecture

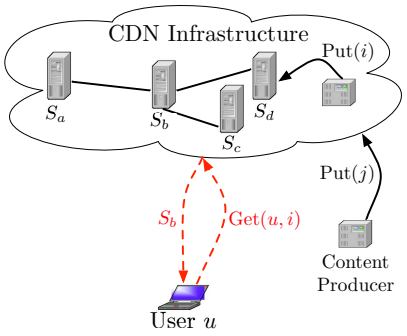

Figure 2: Example of client u requesting content i. Note that the content producer may be (or not) part of the CDN infrastructure

servers placed in various locations. Some of the benefits of using such CDNs are the enhanced quality of service (QoS) perceived by users and the cost reduction of content delivery. This is accomplished by placing servers hosting replicated content copies near the users' location (see Fig. 1).

When accessing content, users download a replica from the closest server, supposedly experiencing a better QoS. Content in infrastructure-based CDNs is not anymore delivered end-to-end, but rather to the servers and then to the users, further reducing content delivery costs. In this paper we focus on Infrastructure-based CDNs (as opposed to P2P-based content delivery networks or an hybrid architecture of thereof), which is hereafter refered as CDNs.

2.2 Recommendation Technologies

The volume of accessible online content has grown rapidly and has far exceeded human processing capability. This leads to *information overload* situations, where user struggle to choose an information item or a service due to insufficient knowledge and time to make an informed decision [7]. This brings to the fore the need for adaptive systems that advise users while taking into account their needs and interests [8], and deliver personalized services in a way most appropriate for the users [9]. Recommender systems are a class of personalized systems that recommend to their users the items they may wish to examine or consume [1]. Recommender systems research produced a variety of methods deployed in numerous applications and Websites. We focus on two established recommendation methods: content-based filtering [10] and collaborative filtering [11].

The Content-based (CB) recommendation approach represents both the items and the users through their associated features. For example, consider a news recommender addressing the topics, people, and geographical origin of the news items as their features. Having observed the items previously liked or consumed by a user, the CB recommender can select not yet examined items, whose features are similar to those of the past items. Many of the commonly used machine learning techniques can be exploited by CB recommenders. We would like to stress that CB recommenders are typically limited to recommending items with features similar to those of previously consumed items and rarely provide serendipitous recommendations [10].

Collaborative filtering (CF) recommenders, on the contrary, rely on the idea that users who agreed in the past will also agree in the future, and use the opinions of like-minded users to generate recommendations. The main stages of the CF recommendation process are: (1) compute user-to-user similarity degrees; (2) select a neighborhood of most similar users; and (3) aggregate the opinions of the selected neighbors. The main advantage of CF over CB recommenders lays in their independence of the representation of users and items. CF systems can generate recommendations for items regardless of their features, such that a single systems can recommend, for example, video, audio, and textual items. However, CF recommenders suffer from the cold-start problem [12], where the volume of available data may not suffice for the generation of high-quality recommendations [11].

3. NOTATION AND ASSUMPTIONS

Let us denote by S the set of servers of a CDN, by I the catalogue of available items, and by U the set of CDN users. We assume that I is synchronized across S and that every item $i \in I$ can be modeled with respect to a set of content features $\{f_j\} \subseteq F$ derived from the content meta-data. For example, video content can be represented by the genre(s) to which it belongs, the name(s) of the creators, duration, technical characteristics, and so forth. We assume that this information is available and accompanies the items. We refer the reader to [13] for a discussion on meta-data extraction and content analysis techniques.

The content placement and access processes can be depicted by two primitives, *Put* and *Get*

$$Put : i \mapsto \{s\} \subseteq S$$

$$Get : (u, i) \mapsto s \in S$$

where for a particular content item $i \in I$ and user $u \in U$, *Put* returns a set of CDN servers on which replicas of i should be placed, and *Get* returns a server storing a replica of i, which u should access. This is illustrated in Fig. 2

We assume a fairly uniform distribution of servers, either from the geographic perspective (e.g., one server per location) or based on placement strategies reflecting the observed CDN topology [14]. Likewise, we assume a fairly uniform load of servers, as managed by the CDN [15]. We denote by $cost(u, s)$ the cost of delivering content from server s to user u, in terms of delay, bandwidth, packet loss, server load, and others factors. Although for a given pair (u, s), content delivery costs may fluctuate due to various network

parameters (load, flashcrowds, etc), we rely on the CDN management policy to balance these and maintain a reasonably stable $cost(u, s)$. Hence, we define the "cheapest" server of u as $serv(u) = \arg\min_{s \in S} cost(u, s)$.

The objective of the replica placement strategy would be to minimize the overall cost of content access for all the possible users and items. Thus, we formulate the replica placement cost minimization objective function as

$$\arg\min_{put(i,s)} \sum_{u} \sum_{i} cost(u, Get(u, i)) \quad (1)$$

Equation 1 highlights the main component that affects the objective function – the delivery costs. This component encompasses the impact of both the replica placement and content access policies. Indeed, the optimization of the replication policy in CDNs has a pivotal role in maximizing the QoS and minimizing the costs [16, 17]. However, this inherently presumes that content consumption parameters are known a priori and stable [14]. Unfortunately, some of these parameters may not be readily available or fluctuate over time. For instance, the number of users consuming i from s, i.e., $u \subseteq U$ interested in i for whom $serv(u) = s$, is not known and depends on the interest of these users in i and on the likelihood of these users to consume i.

This likelihood can be estimated through the history of previous item requests initiated by these users and we will discuss this in detail in the following section. Meanwhile, we denote by P_s the profile of a server s, which reflects the aggregate content consumption of users $u \subseteq U$ for whom $serv(u) = s$. More formally, we represent the server profile $P_s = \{(f_j, sc_j)\}$ through a set of features $\{f_j\} \subseteq F$ of the consumed items and their corresponding scores. For example, consider a scoring method that computes the score sc_j of a feature f_j as the portion of items including f_j among all the requests served by s. With every consumption of an item i initiated by a user u whose cheapest server is s, we update the score sc_j of all the features $\{f_j\}$ of the consumed item. Hence, updating the server profile P_s for every content item request implies recomputing the scores of i's features, which is negligible in comparison to other operations performed by a CDN while handling user requests.

It should be highlighted that here we do not consider the *network* cold-start state. That is, we assume to possess a reliable a priori knowledge regarding the past consumption of items by every user and server, i.e., the history of requests initiated by every user $u \in U$ and the cheapest server $serv(u)$ of every user.

4. AN IN-NETWORK RECOMMENDER SYSTEM APPROACH

In this section, we present two practical use cases for the application of recommendation technologies in CDNs. Both advocate a modification of existing replica placement policies by predicting the likelihood of certain items to be consumed through certain servers. The first use case focuses on the placement of new *cold* content items, whereas the second on the placement of *warmed* items for which previous consumption history is already available. In both cases, the predictions are generated using established algorithms adopted from the recommender systems research.

4.1 Cold Items: Suggest Replica Placement

We first discuss the placement of new cold items in the CDN. That is, we address the problem faced by CDNs and analysts when a newly released content item i is added to the network: what are the best locations, where replicas of the new content should be stored? In essence, the problem can be reduced to a recommendation of a subset of servers $\{s_m\} \subseteq S$ amongst the set of CDN servers, on which replicas of i will be placed.

For a cold item i, no consumption history on any CDN servers is available. Hence, we apply the content-based recommendation method to predict the likelihood of i to be consumed through a given server s_l. We rely on the set of features of $\{f_j\}$ of i and consider it to be a reliable representation of the content of i. Then, we estimate the likelihood of consumption of i through s_m by computing a similarity score between the set of features of i and the server profile P_{s_m}. Specifically, we compute

$$score_{CB}(i, s_m) = sim(i, P_{s_m}) \quad (2)$$

In Equation 2, $score_{CB}$ refers to the content-based score of the cold item i on a candidate CDN server s_m, and $sim(.,.)$ can be any multi-dimensional similarity metric, e.g., cosine similarity, Pearson's correlation, or distance function.

Upon computing the content-based score $score_{CB}$ of each CDN server, there are two approaches to place the replicas of the new content item i across the servers. The first one would simply imply placing the replicas at an a priori defined number of servers having the highest scores. This, however, may potentially put replicas of an isoteric item on a server, through which it is unlikely to be consumed. The second approach would put a replica of i at all the servers, whose similarity score passes an a priori defined threshold. While this guarantees that only servers with high scores will store a replica of i, this may lead to a situation where too many (or too few) servers store replicas.

However, we note that the content-based placement of replicas of new content items is likely to lead to the segmentation phenomenon. As the servers are scored from the content similarity perspective, and this drives the placement of replicas, the servers will naturally "specialize" in certain content features at the expense of other features. For instance, consider a French CDN server that primarily serves users whose IP addresses are geolocated in France, then the profile of this server will naturally contain a high ratio of French items. Hence, the content-based score of this server for a newly inserted French item will also be high and the server will attract more and more French content.

4.2 Warmed Items: Revisit Replica Placement

The above content-based placement can address the placement of new cold items, for which no prior consumption history is available. However, as the item gets warm more and more item consumption information gets available, CDN management policy may need to revisit the existing replica placement and consider storing a replica of the item at servers not yet storing a replica.

We consider a scenario where every server $s_m \in S$ stores the popularity of an item i on the server, $pop(s_m, i)$ across the users whose cheapest server is s_m. This can be quantified, for example, by the frequency of request for i by these users. Given a set of servers $\{s_m\} \subseteq S$ already storing a

replica of i, we can use the collaborative filtering recommendation technique to decide on the placement of replicas on other servers.

We estimate the likelihood of consumption of i through a not yet storing replica server s_n through the popularity of i on similar servers already storing a replica of i. Specifically, we compute

$$score_{CF}(s_n, i) = \frac{\sum sim(P_{s_m}, P_{s_n}) \times pop(s_m, i)}{\sum sim(P_{s_m}, P_{s_n})} \quad (3)$$

In Equation 3, $score_{CF}$ refers to the collaboratively predicted score of the cold item i on a CDN server s_n, and $sim(.,.)$ can be any metric quantifying the degree of similarity between two CDN servers through their content consumption profiles. Similarly to the content-based replica placement decision, new replicas can be placed either on a fixed number of top-scoring servers or on all the servers scoring above a certain threshold.

The discussion regarding placing new replicas immediately entails the challenge of removing existing replicas from some server. Since the initial placement turned out to be sub-optimal and more replicas are needed in the CDN, some of the existing replicas might have been placed redundantly and can be removed in order to free up space on the server for other content items. This challenge has, however, been elaborately investigated in prior works on caching [18, 19] and we leave it beyond the scope of our work, noting that the state-of-the art caching solutions would be applicable in this case.

Note that the collaborative placement of replicas for warm items is likely to resolve the fragmentation problem of the content-based placement. The collaborative scoring of content items on a server does not imply content feature similarity, but rather consumption similarity. Hence, the score of items to be stored by a server may be affected by popularity of items on various servers, which will increase content diversity of the target server. We should, however, mention the need for bootstrapping, i.e., past consumption information, of the collaborative approach. This makes it applicable only to cases, where a considerable volume of item consumption information on a number of CDN servers is available.

5. CONCLUSION

In this paper, we advocated that content delivery networks (CDNs) can benefit from the state of the art recommendation technologies. We focused on content placement strategies, which, if learned and adapted to user "tastes" within the network, can potentially enhance content placement. We discussed how two established recommendation techniques can influence content placement strategies, and distinguished between newly injected content and content for which prior consumption history is available. We postulated that the application of these techniques can potentially improve the quality of service and reduce content delivery and network management costs for CDN operators.

This paper is a first step toward the design of "personalized" user-centered networks learning from the observed content consumption patterns. This calls for further work including an in-depth study of the practical impact of the proposed techniques. In particular, when they are largely applied "in the wild", we are interested in the perceived user quality of experience, as well as in the induced costs of content storage and delivery. We also only assumed a server-provisioned CDN architecture. A hybrid approach consisting of a semi-provisioned CDN, where peer-to-peer and client-servers models co-exist, can also benefit from learning user consumption patterns. The potential of aplpying personalized recommendation techniques in caching strategies is also of interest and should be investigated.

6. REFERENCES

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*. Springer, 2011.

[2] R. Burke, A. Felferning, and M. H. Goker, "Recommender systems: an overview," *AI Magazine*, vol. 32, no. 3, pp. 13–18, 2011.

[3] M. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," University of Melbourne, Technical Report GRIDS-TR-2007-4, 2007.

[4] K. Huguenin, A.-M. Kermarrec, K. Kloudas, and F. Tïari, "Content and geographical locality in user-generated content sharing systems," in *NOSSDAV*, 2012.

[5] A. Brodersen, S. Scellato, and M. Wattenhofer, "Youtube around the world: Geographic popularity of videos," in *Proc. International Conference of World Wide Web (WWW)*, 2012.

[6] Z. Li, J. Lin, M.-I. Akodjenou, G. Xie, M. A. Kaafar, Y. Jin, and G. Peng, "Watching videos from everywhere: a study of the PPTV mobile VoD system," in *Proc. Internet Measurement Conference (IMC)*, 2012.

[7] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, no. 7, pp. 30–40, 1994.

[8] S. Berkovsky, "Decentralized mediation of user models for a better personalization," in *Proc. International Conference on Adaptive Hypermedia*, 2006.

[9] P. Brusilovsky, A. Kobsa, and W. Nejdl, *The Adaptive Web Methods and Strategies of Web Personalization*. Springer, 2007.

[10] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*, 2011, pp. 73–105.

[11] Y. Koren and R. M. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, 2011, pp. 145–186.

[12] A. I. Schein, A. Popescul, L. H. Unger, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proc. SIGIR Conference on Research and Development in Information Retrieval*, 2002.

[13] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Application of video-content analysis and retrieval," *IEEE Multimedia*, vol. 9, 2002.

[14] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *IEEE INFOCOM*, 2001.

[15] B. Molina, C. E. Palau, and M. Esteve, "Modeling content delivery networks and their performance," *Computer Communications*, vol. 27, no. 15, pp. 1401–1411, 2004.

[16] D. Dowdy, L. Foster, "Comparative models of the file assignment problem," *ACM Computer Surveys*, vol. 14, no. 2, pp. 28–313, 1982.

[17] M. Karlsson, C. Karamanolis, and M. Mahalingam, "A framework for evaluating replica placement algorithms," HP Laboratories, Technical Report HPL-2002-219, 2003.

[18] F. Lo Presti, C. Petrioli, and C. Vicari, "Dynamic replica placement in content delivery networks," in *Proc. International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2005.

[19] Y. Chen, R. Katz, and J. Kubiatowicz, "Dynamic replica placement for scalable content delivery," in *Proc. International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.