Computer Networks 121 (2017) 137–151

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Low complexity content replication through clustering in Content-Delivery Networks

Lazaros Gkatzikis [a], Vasilis Sourlas [b,*], Carlo Fischione [c], Iordanis Koutsopoulos [d]

[a] Huawei Technologies, Paris, France
[b] Dept. of Electronic and Electrical Engineering, University College London (UCL), UK
[c] Dept. of Electrical Engineering, KTH, Stockholm, Sweden
[d] Dept. of Informatics, AUEB, Athens, Greece

ARTICLE INFO

Article history:
Received 10 September 2016
Revised 14 March 2017
Accepted 12 April 2017
Available online 13 April 2017

Keywords:
Content replication
Content clustering
Content-Delivery Networks
Coordinated caching

ABSTRACT

Contemporary Content Delivery Networks (CDN) handle a vast number of content items. At such a scale, the replication schemes require a significant amount of time to calculate and realize cache updates, and hence they are impractical in highly-dynamic environments. This paper introduces cluster-based replication, whereby content items are organized in clusters according to a set of features, given by the cache/network management entity. Each cluster is treated as a single item with certain attributes, *e.g.*, size, popularity, *etc.* and it is then altogether replicated in network caches so as to minimize overall network traffic. Clustering items reduces replication complexity; hence it enables faster and more frequent caches updates, and it facilitates more accurate tracking of content popularity. However, clustering introduces some performance loss because replication of clusters is more coarse-grained compared to replication of individual items. This tradeoff can be addressed through proper selection of the number and composition of clusters. Due to the fact that the exact optimal number of clusters cannot be derived analytically, an efficient approximation method is proposed. Extensive numerical evaluations of time-varying content popularity scenarios allow to argue that the proposed approach reduces core network traffic, while being robust to errors in popularity estimation.

1. Introduction

Content Delivery Networks (CDNs) currently account for 36% of the Internet traffic [1], and they are expected to carry more than half of such a traffic by 2018. In order to meet the growing demand for content, CDN providers deploy cache servers worldwide that host replicas of content. Each content request is redirected to the closest replica rather than being served by the back-end/origin server. Thus, through replication, content requests are served locally and this improves both user Quality of Experience (QoE) (*i.e.*, access latency) and minimizes core network traffic.

Current content delivery services operated by large CDN providers like Akamai [2], Limelight [3] and Netflix [4] can exert enormous strain on ISP networks [5]. This is mainly due to that CDN providers control both the placement of content in surrogate caches/servers spanning different geographic locations, as well as the decision on where to serve client requests from [6]. These decisions are taken without knowledge of the precise network topology and traffic load, and they can result in network performance degradation, thus affecting the experience of end users. To address this issue, the studies in [7,8] propose the notion of an Internet Service Provider (ISP) CDN. An ISP deploys caches over network nodes and manages the resulting limited-capacity distributed CDN service within its network. In contrast to CDN providers, ISPs have global knowledge about the utilization of their network, which causes the problem of optimal content replication in a network of caches.[1]

Existing replication schemes rely on the assumption that the popularity of content items is static or changes slowly. Thus, given an estimate of content popularity, caching can be performed at item-level granularity. For instance, in [9] authors proposed the greedy replication algorithm, which has a computational complexity of NV^2C computations, where N is the number of content items, V the number of caches/nodes and under the assumption that all

* Corresponding author.
E-mail addresses: lazaros.gkatzikis@huawei.com (L. Gkatzikis), v.sourlas@ucl.ac.uk (V. Sourlas), carlofi@kth.se (C. Fischione), jordan@aueb.gr (I. Koutsopoulos).

[1] Note that the proposed replication schemes are not limited for ISP deployed caches but can be used in any network of caches.

http://dx.doi.org/10.1016/j.comnet.2017.04.043
1389-1286/

Table 1
Replica assignment computation time using the five most powerful computers and 10^5 parallel virtual machines ($V = 100, N = 10^9, C = 0.1N$).

Name	Proc. Cap. (*petaflops*)	Repl. time
Tianhe-2	33.86	≈ 7 h
Titan	17.59	≈ 15 h
Sequoia	17.13	≈ 16 h
K Computer	10.51	≈ 26 h
Mira	8.586	≈ 32 h
10^5 cluster cores	$\approx 0.5 \cdot 10^{-3}$ per core	≈ 5 h

nodes have the same storage capacity *C*, where *C* is the maximum number of items to be stored in the cache. However, in reality content popularity changes from time to time and given the vast number of items circulated over the network and that an ISP's network consists of numerous nodes/caches, even such a polynomial complexity algorithm cannot be applied frequently enough or is extremely costly.[2]

Consider for example a realistic content catalog of size $N = 10^9$ items and a domain of $V = 100$ nodes. We depict in Table 1 the amount of time required by the five most powerful non-distributed computers (according to http://www.top500.org) for the computation of a new replication assignment according to the greedy algorithm of [9], when each cache/node can hold 10% of the content catalog. We also depict the performance if the computation was parallelized over 10^5 typical Virtual Machines in a public or private cloud. We observe that even the most powerful computer requires more than 7 hours to compute a new replication assignment, whereas in the parallel execution approach, even if we neglect the communication delay and that the greedy algorithm is not fully parallelizable, more than 5 hours would be required. This implies that the problem complexity is substantial, and *the time required to calculate a new cache assignment matches or even exceeds the time scales dictated by the dynamics of content popularity*. This issue is further amplified by fragmentation of items into equally sized chunks, which is a requirement of many replication mechanisms, such as [11,12]. Thus, novel replication schemes of lower complexity are required.

Here, we propose the alternative of *content aggregation through clustering* to reduce the complexity of replication and thus enabling frequent cache updates according to popularity variations. Clustering is a machine learning technique [13], which groups items into clusters based on a certain similarity metric. In our context, item clustering significantly reduces the input size (dimension) of the replication problem. By selecting the number of clusters one may finely tune replication complexity. A cluster of content items is treated as a single item of certain attributes, and replication decisions are taken for the whole cluster as being one item.

Whereas existing works have demonstrated the potential of clustering to reduce complexity [14,15], *this is the first work that provides an efficient method to calculate the optimal number of clusters*. In detail, our original contributions are as follows.

- We model the impact of computational complexity of the underlying replication scheme on the overall network performance under content popularity dynamics.
- We characterize the tradeoff between the time required to calculate a new replication configuration and the sub-optimality of the corresponding replication decisions.
- We propose an optimization-based approach to compute the optimal number of clusters to form, so that overall network traffic is minimized.
- We propose a replication-aware clustering scheme that takes into account the spatial diversity of content popularity.
- We compare the proposed clustering scheme against a replication-agnostic clustering scheme, as well as various item-level partially coordinated caching schemes, assuming different levels of coordination (from fully coordinated to totally uncoordinated).

The rest of the paper is organized as follows. In Section 2 we survey related work, whereas in Section 3, we present the system architecture and identify the impact of content popularity dynamics on replication decisions. In Section 4, we introduce the replication-aware clustering scheme, as well as an optimization-based approach to compute the optimal number of clusters, and we describe the alternative of partially-coordinated caching mechanisms. We evaluate numerically the performance of those alternatives for realistic network topologies and traffic data, and we demonstrate that clustering enhances the robustness of replication to content popularity variations in Section 5. Finally, Section 6 concludes our study.

2. Related work

2.1. Replication in static environments

The problem of optimal content replication and placement in a network of distributed caches has received significant interest lately [11,16]. It is an NP-hard problem [9,17], and as such, approximation algorithms of polynomial complexity have been proposed. In [18], authors model the cache assignment problem as a distributed selfish replication (DSR) game in the context of distributed replication groups (DRG). Under the DRG abstraction, nodes utilize their caches to replicate information items and make them available to local and remote users with the objective of minimizing the overall network traffic. The pairwise distance of the nodes in [18] (transfer cost between any two nodes) is assumed to be the same and thus no network characteristics are taken into account. In the context of DRG and under the same distance assumption of [18], a 2-approximation cache management algorithm is presented in [19]. In [11] the authors develop a cache management algorithm for maximizing the traffic volume served by the caches and hence for minimizing network bandwidth cost. They focus on a set of distributed caches, either connected directly, or via a parent node, and they formulate the content placement problem as a linear program to benchmark the globally optimal performance. In the popular area of information-centric networks, a set of offline cache planning and replica assignment algorithms are proposed in [20], whereas in [21] a distributed cache management architecture is presented that enables dynamic reassignment of content items to caches in order to minimize overall network traffic.

The above mentioned replication schemes rely on the assumption that the popularity of content items is either static or changes slowly. In reality, content popularity changes significantly over time [22–24] and the design of a content replication scheme that updates caches following closely those changes is a challenging management task, since replication is a time-consuming process.

2.2. Reducing the complexity of replication process

In order to reduce the computational complexity of the replication process, partially coordinated replication was introduced in [25]. Nodes use a fraction of their cache capacity to store content in a coordinated manner (CDN-like replication) and the

[2] Netflix, for example, performs a nightly push of the nationally (US) most popular movies to all its regional caches [10].