Decentralized view prediction for global content placement

Stéphane Delbruel, Davide Frey, François Taïani

Decentralized view prediction for global content placement

Stéphane Delbruel
University of Rennes 1
IRISA - Inria Rennes, France
stephane.delbruel@irisa.fr

Davide Frey
Inria
Rennes, France
davide.frey@inria.fr

François Taïani
University of Rennes 1 - ESIR
IRISA Rennes, France
francois.taiani@irisa.fr

ABSTRACT

A large portion of today's Internet traffic originates from streaming and video services. Storing, indexing, and serving these videos is a daily engineering challenge that requires increasing amounts of efforts and infrastructures. One promising direction to improve video services consists in predicting at upload time where and when a new video might be viewed, thereby optimizing placement and caching decisions. Implementing such a prediction service in a scalable manner poses significant technical challenges. In this paper, we address these challenges in the context of a decentralized storage system consisting of set-top boxes or end nodes. Specifically, we propose a novel data placement algorithm that exploits information about the tags associated with existing content, such as videos, and uses it to infer the number of views that newly uploaded content will have in each country.

Categories and Subject Descriptors

H.3.2 [**Information Systems**]: Information Storage and Retrieval—*Information Storage*

General Terms

Measurement, Algorithm, Design

Keywords

User-generated content, YouTube, tag, prediction

1. INTRODUCTION

On-line content production and consumption is exploding both in terms of volume and involved users. In June 2015, for instance, Youtube received 300 hours of video every minute, as opposed to 100 hours every minute in 2014 [11]. In order to maintain a high level of service, the corresponding data must be reliably stored and distributed to users through a globally distributed infrastructure.

Nowadays, most of this data is placed in private data centers and served to requesting users using Content Distribution Networks (CDNs) or by placing caching servers into the infrastructure of ISPs [5]. This approach tends to favor big players, and both academia and practitioners have sought to explore alternative designs. In fully decentralized solutions [1, 2, 6, 9, 10], for example, each user (through her computer or set-top box) provides storage capacity and computing resources to the overall system, eschewing the need for any centralized element, either hosted in the cloud or on replicated servers.

Decentralized solutions rely primarily on *local* information (e.g. users profiles, bandwidth, latency, tags). While this helps with scalability, too strong a focus on locality constrains the range of decisions that can be taken by individual nodes, and limits ability of such services to adapt to richer and more advanced phenomena occurring at a global scale.

Recognizing this problem, we study how to make *global predictions* in large-scale decentralized systems, with an application to the placement of video storage in a decentralized UGC video service. Predictions provide a crucial ability to self-adaptive systems: they use information about the past to predict the future, and can help systems anticipate systemic changes, and adapt to new workloads. Global predictions can unfortunately be difficult to achieve in a decentralized system in a scalable manner: for instance, predicting where a newly uploaded video might be most viewed would ideally require processing global viewing patterns of similar videos, a global and hence particularly costly computation.

2. DECENTRALIZED PREDICTION

We propose a *decentralized* architecture that optimizes the storage of videos by predicting their viewing patterns. In our previous work [4], we suggested proactively placing videos in the countries where they were likely to be most viewed. To this end, we proposed an *off-line* tag-based prediction of video views in each countries. Here we explain how to decentralize this approach.

2.1 Baseline prediction with full information

The original insight behind our off-line prediction was to use the number of views currently associated with a tag t in a country c (noted $\mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[c]$) to compute the average number of views that videos associated with t had received in country c (noted $\mathbf{views\_p\_vid}^{\mathcal{V}_{\mathrm{cur}}}(t)[c]$).

$$\mathbf{views\_p\_vid}^{\mathcal{V}_{\mathrm{cur}}}(t)[c] = \frac{\mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[c]}{|\{v \in \mathcal{V}_{\mathrm{cur}} : t \in tags(v)\}|} \quad (1)$$

Let $\mathcal{V}_{\mathrm{cur}}$ be the set of videos currently in the system, and $tags(v)$ the set of tags associated with a video v. Then, Eq. (1) counts the number of per-video views of a tag t in a country c, and normalizes this value by dividing it by the total number of videos containing t among those that are currently in the system. A real system should complement this by taking into account the number of views in a finite time window (discounting very old views). But, for simplicity, we ignore this aspect in the following. Viewing profiles based on sliding windows may easily be integrated by following the example of [3].

Based on the average number of views per tag, equation (2) then estimates the number of views of a new video as the arithmetic mean (noted $\mathbb{E}$) of the number of per-video views of the uploaded video's tags. Once we have predicted the number of views of a video for each country, we can then store the uploaded video in the country or countries associated with the largest prediction(s).

$$\widehat{\mathbf{views}}(v)[c] = \underset{t \in tags(v)}{\mathbb{E}} \Big(\mathbf{views\_p\_vid}^{\mathcal{V}_{\mathrm{cur}}}(t)[c] \Big) \quad (2)$$

2.2   Towards Decentralized Prediction

The major challenge in decentralizing Eq. (1) and (2) lies in computing the number of videos that contain a given tag across all countries (denominator in Eq. (1)). A precise computation would require a global aggregation process covering all users. But in practice, we do not need to predict absolute numbers of views to place videos, but we only need to rank countries according to where a video might be most viewed in the future. If we further decide to consider that all tags appear in the same number of videos (an obvious approximation), we can simplify Eq. (1) and concentrate on its numerator. This leads us to simply use the average number of views of a video v's tags in a country c as an estimator of the country's fitness for hosting the video v.

$$\mathbf{fitness}(v)[c] = \underset{t \in tags(v)}{\mathbb{E}} \Big(\mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[c] \Big) \quad (3)$$

2.3   From Countries to Hosts

Computing the fitness score in Eq. (3) would be easy if we had a system consisting of one storage server per country that kept track of the videos and tags consumed in this country. However, our goal consists in storing videos on end nodes; and each country may comprise a very large number of them. In this section, we therefore propose a novel approach that allows us to estimate the value of the fitness score in a large-scale decentralized system.

First, we observe that we can rewrite Eq. (3) by summing the average number of per-tag views of each individual user u in country c.

$$\mathbf{fitness}(v)[c] = \underset{t \in tags(v)}{\mathbb{E}} \Big(\sum_{u \in c} \mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[u] \Big)$$
$$= \sum_{u \in c} \underset{t \in tags(v)}{\mathbb{E}} \Big(\mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[u] \Big) \quad (4)$$

Based on Eq. (4), each user simply needs to record her own number of views for each tag, t. The only problem that remains then consists in estimating the value of the summation in a decentralized fashion.

We achieve this by creating a self-organizing overlay [2] in each country. Each user in each overlay maintains a pro-file that contains all the tags associated with all the videos she has viewed, each weighted by a corresponding number of views. Each overlay organizes the users of a particular country in a k-nearest-neighbor graph, with a similarity metrics that reflects the "fitness" of users to one another.

$$\mathbf{sim}(u,v) = \sum_{t \in tags(u) \cap tags(v)} \mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[u] \cdot \mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[v] \quad (5)$$

The similarity considers the profile of a user as a "paradigmatic video" that captures this user's tastes. Thus, we can interpret the above metric as the fitness of one "user-as-a-video" for another user, as we did in (4) for individual videos. The differences with (4) are that a user's tags are weighted by the number of times this user was exposed to that tag, hence the multiplication; and that we do not divide by the number of tags in a video, hence a summation instead of an arithmetic mean.

2.4   Estimating Views using the Overlay

The country-level self-organizing overlay runs as a background task and sets up a routing environment that allows us to estimate the number of views of a video in a given country. In addition, we also employ an additional overlay: a global random-peer-sampling (RPS) protocol [7] that comprises all participating users, regardless of their country.
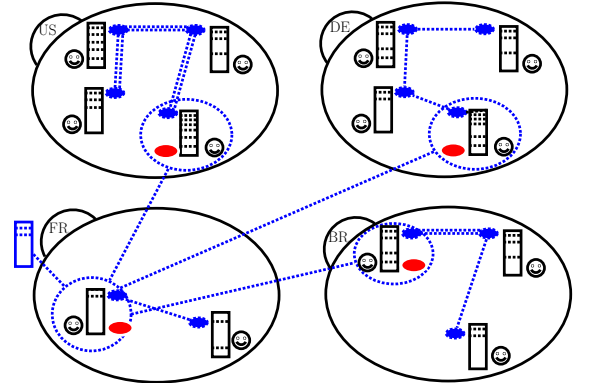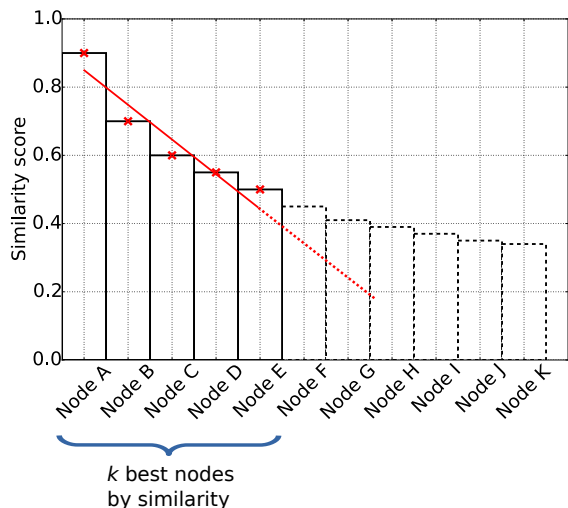

Figure 1: An uploading node with a virtual node in each country

Figure 1 outlines our decentralized prediction. Consider a user $\hat{u}$ in country $\hat{c}$ that uploads a new video $\hat{v}$ with a set of tags $tags(\hat{v})$. User $\hat{u}$ uses the global RPS to identify one node for each country. Depending on the size of the global RPS view, identifying one node per country may require $\hat{u}$ to wait for a few RPS rounds. However, we estimate that a identifying nodes for a sufficiently large subset of countries may be enough. Let W be the set of identified nodes.

Next, $\hat{u}$ builds a profile data structure for video $\hat{v}$. This profile, $P(\hat{v})$ contains all the tags associated with $\hat{v}$, each with a value of 1. Node $\hat{u}$ then asks each node in W to run a virtual node in the country-level overlay using a copy of $P(\hat{v})$. When computing the similarity between a user and $P(\hat{v})$, Eq. (5) reduces to Eq. (6).

$$\mathbf{sim}(u, P(\hat{v})) = \sum_{t \in tags(u) \cap tags(\hat{v})} \mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[u]$$
$$= \Big[\underset{t \in tags(\hat{v})}{\mathbb{E}} \Big(\mathbf{views}^{\mathcal{V}_{\mathrm{cur}}}(t)[u] \Big) \Big] \cdot |tags(\hat{v})| \quad (6)$$

Figure 2: Trend graph for an uploaded video. The dashed node scores are those that are ignored because not within the k best. The linear-regression slope from the k bests is shown in red.

Each virtual overlay node participates in a country-level overlay until the corresponding view converges. When this happens, the view of each virtual node will contain a set of nodes with the highest similarity values, that is with the highest values of Eq. (6). If we plot all these similarity values in decreasing order, we obtain a plot like the one in Figure 2, which we refer to as a *trend graph*.

The trend graph lists all the entries in a virtual node's view in decreasing order of similarity. Since similarity values correspond to the sum of the number of views of the video's tags (Eq. (5)), the integral of the trend graph over the entire set of nodes in country $\hat{c}$ corresponds to the fitness of video $\hat{v}$ in $\hat{c}$ multiplied by $|tags(\hat{v})|$ (Eq. (4)).

Clearly, scalability reasons make it impossible for clustering views to contain the entire set of a country's nodes. We therefore compute an approximation of the integral of the trend graph. For example, we can consider only its "visible" portion, or approximate Eq. (4) by using the area under its linear regression. We leave the evaluation of these and other heuristics, including higher-order approximations, to future work.

After computing the approximation for all the considered countries, user $\hat{u}$ selects the highest ranked ones and stores the videos on the top scoring host(s) in each of them. We plan to experiment with different options: one versus multiple countries, and one versus multiple top-scoring hosts.

3. CONCLUSION AND DISCUSSION

In this paper, we have outlined a protocol and an architecture to estimate where a newly uploaded video might be most viewed, and place this video accordingly, in a fully decentralized manner. Because it avoids the need for a central support infrastructure, our approach hints at the possibility of smart analytics in pure peer-to-peer systems. This can be useful both to promote alternatives to the cloud-centered model of current UGC video services, but also to improve the hybrid P2P/cloud architectures already used by some leading companies (Spotify [8], Akamai [12]) by offloading

complex adaptive tasks to the P2P part of a hybrid system. We are currently continuing this preliminary work by evaluating several approximation techniques on realistic traces.

4. REFERENCES

[1] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, Dec. 2004.

[2] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy. The gossple anonymous social network. In *Middleware*, 2010.

[3] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec. WhatsUp Decentralized Instant News Recommender. In *IPDPS*, 2013.

[4] S. Delbruel, D. Frey, and F. Taïani. Exploring the Geography of Tags in Youtube Views. Research Report RT-0461, IRISA, Inria Rennes, Apr. 2015.

[5] Google. Google peering and content delivery . https://peering.google.com/about/ggc.html. (accessed 2/5/2015).

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

[7] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. Gossip-based peer sampling. *ACM TOCS*, 25, 2007.

[8] G. Kreitz and F. Niemelä. Spotify – large scale, low latency, P2P music-on-demand streaming. In *P2P*, 2010.

[9] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, et al. Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices*, 35(11):190–201, 2000.

[10] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez. The little engine(s) that could: scaling online social networks. In *SIGCOMM*, 2010.

[11] L. Youtube. Statistics, viewership . http://www.youtube.com/yt/press/statistics.html. (accessed 2/5/2015).

[12] M. Zhao, P. Aditya, A. Chen, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, B. Wishon, and M. Ponec. Peer-assisted content distribution in Akamai NetSession. In *IMC*. ACM, 2013.