On Netflix catalog dynamics and caching performance

Walter Bellante
Telecom ParisTech - France
Email: walter.bellante@gmail.com

Rosa Vilardi
DEI - Politecnico di Bari - Italy
Email: r.vilardi@poliba.it

Dario Rossi
Telecom ParisTech - France
Email: dario.rossi@enst.fr

***Abstract*—Multimedia streaming applications have substantially changed the market policy of an increasing number of content providers that offer streaming services to the users. The need for effective video content delivery re-fueled interest for caching: since the Web-like workload of the 90s are not longer fit to describe the new Web of videos, in this work we investigate the suitability of the publicly available Netflix dataset for caching studies. Our analysis shows that, as the dataset continuously evolves (i) a steady state description is not statistically meaningful and (ii) despite the cache hit ratio decreases due to the growth of active movies in the catalog, simple caching replacement approaches are close to the optimum given the growing skew in the popularity distribution over the time. Additionally, we point out that, since the dataset reports logs of movie *ratings*, anomalies arise when ratings are considered to be movie *views*. At the same time, we show anomalies yield conservative caching results, that reinforces the soundness of our study.**

I. Introduction

The growth of multimedia Video-on-Demand (VoD) streaming applications of both user generated/uploaded content (YouTube, DailyMotion, etc.) or movies and TV programs (Netflix, Hulu, etc.) is well known.

Internet is increasingly used as a platform to distribute content from one or more multimedia sources to a very large population of users. Popular content have to be delivered multiple times and user requests are usually asynchronous and uncoordinated. Furthermore, VoD is not only a prerogative of the fixed Internet but also of the mobile environment – which has been made possible by the increasing development of the high-speed 3G and 4G cellular networks on the one hand, and by the ever-growing penetration of mobile smartphones and tablets devices on the other hand. For this reason the IP multicast paradigm is not helpful anymore because the end-to-end TCP/IP principle leads to bandwidth waste.

Caching appears as a natural technique to cope with increasing VoD demand because popular content have to be delivered multiple times to different users. Caches could be placed at the edge of the network to keep the content nearby the users. The main benefits of caching translate into the reduction of backhaul traffic on the one hand, and to better user experience via higher throughput and reduced delays (in terms of both latency and jitter) on the other hand. Undeniable is the ever growing interest in caching architectures such as Content Distribution Networks (CDN) or, more recently, Information Centric Networks (ICN). While CDN operates over-the-top of the Internet and are service-specific, ICN architectures build low-level networks of caches, so that under this new paradigm every router can possibly become a caching entity for all transiting content. Notice that caching can be beneficial also for the mobile infrastructure irrespectively of the underlaying VoD streaming technology: e.g., VoD caching be applied to Content Delivery Network (CDN) services such as YouTube, Hulu, Netflix etc., or even to purely peer-to-peer (P2P) applications such as PPLive, PPStream (mostly popular in the Asian countries). Additional benefits could come from co-localization of transcoding and caching functions (e.g., caching multiple copies at different resolutions, optimized for different screen size of the different handheld devices).

Still, in order to evaluate the potential gain brought by caching, there is need for *realistic workloads* that network architectures will be expected to serve. As we explain in more detail in Sec. II, the currently used workload has severe limitation hampering the realism of the evaluation. Therefore, we deeply analyzed the large-scale, publicly available Netflix dataset [1], that includes 100M ratings on nearly 18K movies made by 480K users over a 6-year period.

In this quest for realistic workloads, we make two contributions. First, we unveil potential problems in the characterization of the Netflix dataset [2] that the networking community should be aware of that relates to both (i) the dynamic nature of the catalog, and (ii) the existence of outliers and other anomalies, that tend to give conservative caching performance. Second, we compare naive caching replacement policies against the optimum gathered by an omniscient oracle, with a special focus on how performance evolves over time. We show that while the *absolute* caching performance decreases over time, due to the increase of the active movies in the catalog, the *relative* distance between optimal and simple policies decreases. This is due to the fact that, despite both the number of active movies and the customer base increasing, the popularity distribution skew increases as well – which makes naive forecast techniques still accurate.

II. Related Work

Caching is not a new research subject. However, properties of multimedia catalogs nowadays largely differ with respect to traditional Web-caching studies. Limits of the current studies generally arise due to simplistic assumptions on both the *static* and the *dynamic* properties of the workload model. Due to the lack of publicly available datasets, work on caching

TABLE I
DATASET COMPARISON. A STAR SIGN ⋆ DENOTES *ratings* INSTEAD OF *views*.

Service	*U*	*C*	*V*	*Time*			*Space*	*Method*	*Topic*
	10^3	10^3	10^6	*Start*	*Days*	*Grain*			
YouTube	– 16	252 303	539 0.6	2007 2008	6 14	day ms	G L	crawling packet traces	Catalog analysis [2] Catalog analysis [3]
Dailymotion	15 –	– 1194	2.0 1795	2010 2008	120 14	ms week	L G	packet traces crawling	Request analisys [4] Workload analysis [5]
Yahoo!	–	99	770	2008	1	-	G	crawling	Workload analysis [5]
Veoh	–	269	588	2008	1	-	G	crawling	Workload design [5]
Metacafe	–	239	3076	2008	1	-	G	crawling	Workload design [5]
MovieLens	6	4	1.0 ⋆	2000	365	s	G	ratings log	Collaborative filtering [6]
PowerInfo	42	8	20	2004	210	s	L	request log	VoD system analyis [7], System design [8]
Hulu	–	2	0.01	2010	3	ms	L	packet traces	Prefetching [9]
Netflix	480	18	100 ⋆	1998	2725	day	G	ratings log	User deanonymization [10], Collaborative filtering [11]

architectures resort to synthetic workloads. Generally, the content popularity is assumed to be highly skewed (e.g., Zipf distributed) despite no consensus having been reached on its specific tuning (e.g., Zipf varies $\alpha \in [0.6, 2.5]$ in [12]–[17]).

Yet, another more important simplification concerns the catalog dynamics. Catalog size and popularity, in fact, are generally considered to be constant over time. While this simplification is partly justified since caching performance are evaluated over timescales that are short relative to popularity evolution, problems nevertheless remain. Consider indeed that the estimate of the Zipf α parameter from real catalogs is carried over long time-scales: for instance, [18] considers the total number of views of YouTube videos since their first upload, and thus includes videos that are no longer popular – nor even active any longer. In turn, this implies that the value of α estimated over long time-scales in [18] is possibly significantly biased for simulation over short time-scales, where the number of *active movies* over the catalog at any time is surely much smaller. To partly counter this problem, in our previous work [19], [20] we exploited a wider range of α values. At the same time, while it is known that catalog sizes and popularities are evolving over time, little is known about how they do evolve and the community is lacking good models of catalog dynamics. We have compiled a taxonomy of the available datasets in Tab. I.

The table reports the VoD *service* upon which one or more studies are based, whose main topics are briefly described by *keywords*. Each dataset enumerates the number of *users U*, *content C*, and *views V* (notice that some datasets may report video *ratings*, rather than views). In addition, the table reports some time-related dataset properties. Finally, we assess other relevant *spatial* properties, such as whether the dataset is representative of *local (L)* vs *global (G)* views. Time and user granularities are clearly tied to the dataset collection *methodology*: they are generally coarse-grained for *crawling*, and fine-grained for *packet traces* or *session level logs*.

Since the tremendous popularity of YouTube, many works [3], [18], [21]–[23] focused on its study. Briefly, Cha et al. [18] crawled and analyzed the YouTube catalog, Cheng et al. [22] have focused on its social networking aspects, Zink et al. [3] assessed the benefits of caching at a campus network and Khemmarat et al. [23] presented a recommendation-aware prefetching scheme. Yu et al. [7], characterized PowerInfo, China's leading streaming service for TV shows and movies, correlating the time evolution of the catalog with the "recommended" and "most popular" video lists, and describing users' zapping behavior. Krishnappa et al. [9], considered Hulu and studied caching techniques (based on requests collected from a campus network) combined with prefetching techniques (based on the list of most popular videos, obtained by crawling the Hulu website). Finally, Mitra et al. [5] compared different streaming services (Dailymotion, Yahoo! video, Veoh, and Metacafe) extending the validity of previous findings [3], [18].

The largest catalog we are aware of, characterized by both user and time finer granularity, is the publicly available Netflix dataset [1]. We point out that the Neflix dataset has been used in the context of privacy leakage and user de-anonymization [10] or collaborative filtering for recommendation systems [11], but that it has received little attention from the networking community. The only exception is represented by Cha et al. [2], that extends part of their YouTube methodology [18] to study Netflix content popularity and age distributions. However, as our analysis shows, there are many factors –such as non-stationarity and outliers– undermining the relevance of Netflix analysis in [2] that the community should be warned about. Furthermore, while [2], [18] investigate the effectiveness of caching and peer-assisted distribution of YouTube content, they do not extend the analysis to the Netflix catalog, which we are thus first to address in this work.

III. NETFLIX DATASET ANALYSIS

A. Dynamics

In 2006, Netflix held a competition to improve its existing movie recommendation system [11]. A large dataset was made publicly available to the research community [1], containing over 100M ratings on nearly 18K movies from 480K anonymous customers in a 6-year period. As shown in Tab. I, this dataset represent a very good tradeoff between the

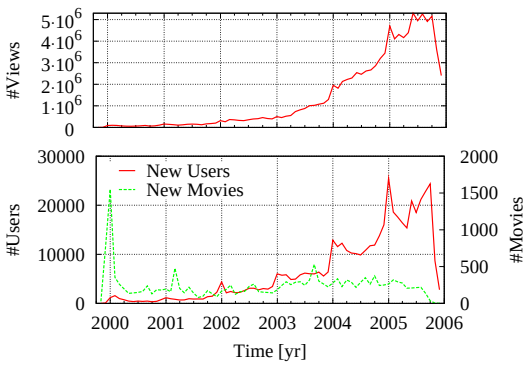

Fig. 1. Netflix user-base and catalog: evolution of the number of monthly views, new users and new movies.

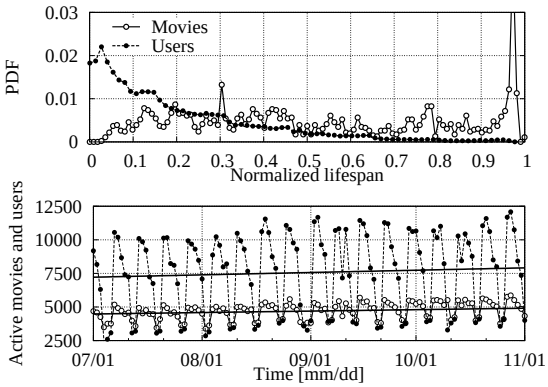

Fig. 2. Users and movies normalized lifespan during the whole dataset (top) and activity during the second half of 2001 (bottom).

catalog, views and user size, the global extent and the temporal duration and accuracy, making it a natural candidate workload for large scale caching studies.

Previously, [2] considered movie *ratings* as video *views*, and investigated the popularity statistics of the Netflix catalog. Moreover, they argued that (i) the number of views is proportional to the number of ratings, and further that (ii) the number of ratings can be considered as a lower bound to the actual number of views, since people usually only ask for the content they want to consume and rarely participate actively in expressing their opinions. To assess whether this assumption holds, and to further point out issues arising when giving a static characterization of a dynamic dataset, we carried out a preliminary dataset investigation.

First, Fig. 1 depicts the evolution of the number of monthly views[1] (top plot) and the number of new users and movies per month (bottom plot) over the entire period. The Netflix user-base boosted around 2003, with only less than 10% of the movies initially entered into the catalog at time 0, and an average of nearly 7 new movies per day. Since user-base, catalog size and service load have all significantly evolved over time, it would be difficult for a steady-state static characterization of the whole dataset to be representative of any transient phase of its evolution.

In order to better understand dataset dynamics, Fig. 2 depicts the active lifespan duration of both users and movies (top plot) and illustrates their activity pattern and trends (bottom plot). Users (movies) are considered to be active if they watch at least one movie (if they are watched by at least one user) in a day. Normalized lifespan is measured as the time difference between the last and the first day of activity over the whole catalog duration. The PDF of the lifespan curves show an highly volatile behavior for users (negative exponential distribution shows a good fit, with mean user lifetime of about 15 months), whereas the movie trend is more persistent (movie activity is roughly uniformly distributed, except for a bulk of movies whose activity span the whole dataset duration, and that are thus consistently popular since their injection at the beginning of the Netflix service).

The bottom plot of Fig. 2 focuses on a shorter time window comprising few months in 2001. Both *active movies* and *active users* statistics exhibit a similar behavior, with a slow seasonal growing trend (represented by a linear regression) modulated by a faster weekly periodic behavior. This weekly periodic behavior is in line with previous findings [7], although the peak number of views appears in the *middle* of the week. To understand why this happens, consider that in the period covered by the dataset, an on-line portal allowed users to rate and choose movies that Neflix would then send through the postal service: clearly, users were taking into account the postal delivery delay in order to get the DVD shipped in time for the weekend. Hence, this is a further factor that needs to be taken into account before using the Netflix dataset to assess caching performance: an optimistic viewpoint would be to assume a simple time-offset in user behavior from postal DVD shipping to Internet transmission; a pessimistic viewpoint could instead lean toward more significant and non-trivial changes in the user behavior.

B. Outliers

We investigate distributions (i) of the number of movie ratings per user and (ii) of the number of user ratings received per movie: while the average user rates about 200 movies (given the average lifespan, less than 1 movie every 2 weeks) and the 90th most active users rank 540 movies, there exist *outlier users ranking almost the whole catalog* (the fact that one such outlier ranked over 17000 movies was already known [24]). Another type of anomaly we found concerns even users with relatively low average rating activity, but quite high rating activity on a single day – for instance 9% of ratings are given to more than 10 movies in a single day.

These outliers and anomalies have an impact on the popularity distribution, as the “views” and “ratings” differ in these cases. However, try to sanitize the dataset is not an easy task: a simple filtering approach, constraining for instance a maximum number of daily movies, raises two issues. First, filtering relies on an *arbitrary*, but reasonable, threshold. Second, it would be impossible to discern which subset of

[1]We use the terms rating and view interchangeably, unless otherwise stated.

ratings corresponds to regular views, and which to filter out (e.g., a subset would be selected at random or in adversarial fashion to get performance averages and bounds).

We argue any sanitization approach to be exposed to well-founded criticism, and any heuristic modification of the dataset would go against the realism of the data. As such, we prefer to reason on the type of bias that the above outliers introduce in a caching system. Assuming "ratings" as "views" then the following hold: since outliers rate a large fraction of the catalog, there will be many false positive views raising cache misses, unless the whole catalog is cached. Hence, outliers yield to a lower bound of cache performance. This is a positive observation, since bias in the Neflix dataset would still provide *conservative* results of caching performance.

A more skeptical viewpoint considers that "ratings" may differ from "views" even for low-activity users: in principle, even if a user rated a video a given day, this would not necessarily imply that the user also requested the video. In this case however, there would be no guarantees about the realism nor the conservativeness of the results based on this dataset (since in the case of popular movies, false positive views no longer translate in cache miss events). However, given the Netflix success, as measured by the steady growth of its user base, catalog, and revenue, we can more then likely rule out this skeptical viewpoint as unlikely.

IV. Caching Netflix movies

The daily granularity of the Netflix catalog is not amenable to test reactive caching replacement approaches (i.e., when caching decisions have to be taken on-line and synchronously with respect to each new request arrival), rather the catalog is perfectly suitable to study performance of proactive caching replacement (aka prefetching[2]), that happens off-line and asynchronously with respect to new requests.

We preliminary investigate the average cache hit over the whole dataset by considering a cache of fixed size $C < V$ (given V the size of the entire catalog):specifically, cache content are periodically updated, and the most popular videos of the next period are prefetched according to the forecast of an oracle. We report here the case of an oracle operating a weekly proactive prefetching[3]. Briefly, by replacing off-line (i) the top 100 videos (corresponding to a cache to catalog ratio $C/V = 0.6\%$), the average cache hit equals 27.5%; (ii) the top 1000 movies, cache hit rate reaches 77.7%; (iii) the top 20% movies (implying $C = 3450$ movies), 98% of the requests can be satisfied by a single cache. Unlike commonly find in the literature, we point that these results do not follow the Pareto rule, and are more optimistic with respect to [7].

We now design and evaluate simple policies, comparing their results with the optimum achieved by the oracle. Our previous analysis of user habits is instrumental in giving guidelines in policy design. As user requests follow a weekly pattern (recall bottom of Fig. 2) this suggest that a *weekly*

[2]By abuse of language we use *caching* and *prefetching* interchangeably.

[3]A larger parameter set is reported in [25].

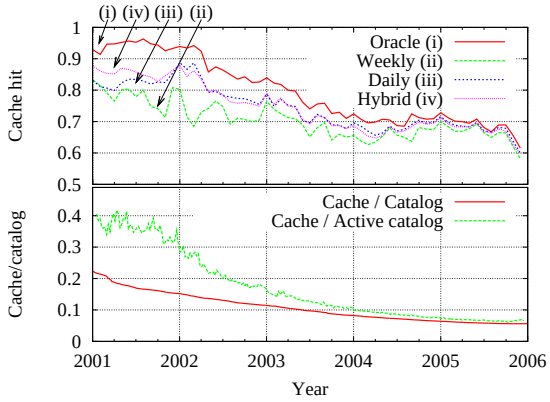

Fig. 3. Prefetching performance comparison

forecast period should be a reasonable choice. Since movie popularity is not volatile (recall top of Fig. 2), this furthermore suggests last week's popularity to be a *good predictor* of next week popularity. Finally, since lifespan at the top of Fig. 2 is normalized over a very long duration (i.e., almost 6 years), it could be useful to operate a smaller portion of prefetching on a higher frequency (e.g., daily basis) to cope with more short-lived viral content.

In summary, we consider the following cache replacement policies: an (i) *Oracle*, storing the C most popular movies each week, a naive (ii) *Weekly* storing the C most popular movies of the i-th week i to serve the $(i+1)$-th one, a (iii) *Daily* policy, storing the C most popular movies of the i-th day to serve the $(i+1)$-th one, and an (iv) *Hybrid* policy, keeping the ϵC most popular daily movies for the next day, and the $(1-\epsilon)C$ most popular weekly videos for the next week. For the sake of the example, we fix $\epsilon = 0.1$ so that most of the videos are decided according to weekly decisions – which minimizes the amount of data to be transferred over the network backhaul, as the bulk of cache replacements happens on a weekly basis. Notice that the Hybrid policy degenerates in the weekly policy when $\epsilon = 0$, and that while the Oracle selects the most popular videos of the *next* week, the Weekly policy selects the most popular videos of the *previous* week.

We now consider how cache performance evolves over time for a fixed cache size. Without loss of generality, in Fig. 3, we simulate the above strategies when $C = 1000$ (i.e., the cache has room for 5.6% of the total catalog size). A larger design space, including parameter sensitivity analysis (e.g., prefetching period, ϵ etc.), and further policies (e.g., weighted on movie ratings) is considered in [25] [4].

Cache hit (top of Fig. 3) shows two trends. First, the *absolute hit rate performance* decreases over time irrespectively of the replacement policy, since the cache to catalog ratio decreases as well (recall that cache size is fixed while the catalog is growing as per Fig. 1). Second, *relative hit rate distance* among policies decreases as well: in 2005, a naive forecast performs as well as the oracle despite the active catalog size growing (this phenomenon is clearly visible for

[4]These investigations are here omitted for space constraints.

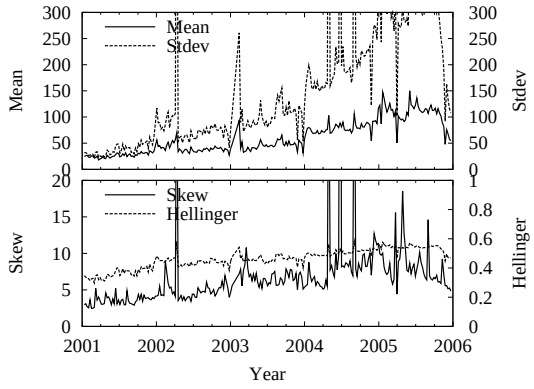

Fig. 4. Evolution of popularity distribution.

all the policies analyzed).

To understand this second trend, we analyze the temporal evolution of the movie popularity distribution. In particular we consider 4 scalar metrics of the number of views per movie: namely (i) the mean, (ii) the standard deviation (stdev), (iii) the skewness (skew), and (iv) the Hellinger distance with respect to the uniform distribution. Notice that (i)-(iii) correspond to the 1st, 2nd and 3rd moment of the view distribution, while (iv) is an alternative skew representation that weights *all* the support and uses the uniform distribution as the reference having the least possible skew. To avoid cluttering the pictures, Fig. 4 reports the statistics over the set of active movies only.

Notice that all statistics grow over time: this suggests the fact that despite the catalog growing larger, and the number of views per movies as well (mean), more popular movies become more popular (stdev, skew). Moreover, notice that while mean, stdev and skew are not robust against anomalies (notice the spikes), the Hellinger distance is more robust (as it weights all the support). Hence, it turns out that (i) despite the catalog and user base growth and that almost the full catalog becomes active during a week (as per bottom plot of Fig. 3), the (ii) the bulk of views is concentrated on a smaller number of movies (as per Fig. 4), which makes prediction easier.

V. Conclusions

In this work, we investigate the suitability of the Netflix dataset for caching studies. Our analysis shows that (i) the dataset is continuously evolving, so that a steady state description, e.g., by means of a single probability distribution of its content is not statistically meaningful (ii) since the dataset reports video *ratings* instead of *requests*, a number of anomalies arise if the above two terms are used interchangeably in the context of caching. At the same time, we argue anomalies yields conservative results, for which we examine how proactive caching policy performance evolves over time. Interestingly, we show that (i) despite an absolute reduction in performance, due to the rise of catalog size, (ii) the relative distance between naive and optimal policies decreases also, given the growing skew in the popularity. As part of our future work, we aim at building new useful datasets through large-scale measurement campaigns.

Acknowledgment

This work was carried out at LINCS http://www.lincs.fr, and was funded by the FP7 mPlane project (GA no. 318627).

References

[1] "Netflix," https://signup.netflix.com/global.
[2] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.
[3] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network - measurements, models, and implications," *Comput. Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009.
[4] Y. Carlinet, T. D. Huynh, B. Kauffmann, F. Mathieu, L. Noirie, and S. Tixeuil, "Four Months in DailyMotion: Dissecting User Video Requests," in *IEEE TRAC*, 2012.
[5] S. Mitra, M. Agrawal, A. Yadav, N. Carlsson, D. Eager, and A. Mahanti, "Characterizing web-based video sharing workloads," *ACM Trans. Web*, vol. 5, no. 2, May 2011.
[6] S. Banerjee and K. Ramanathan, "Collaborative filtering on skewed datasets," in *ACM WWW*, 2008.
[7] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *ACM SIGOPS/EuroSys*, New York, NY, USA, 2006.
[8] M. S. Allen and et al., "Deploying video-on-demand services on cable networks," in *IEEE ICDCS*, 2007.
[9] D. K. Krishnappa, S. Khemmarat, L. Gao, and M. Zink, "On the feasibility of prefetching and caching for online tv services: A measurement study on hulu." in *PAM*, 2011.
[10] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE SP*, 2008.
[11] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *ACM SIGKDD*, New York, NY, USA, 2007.
[12] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache : An overlay architecture for information-centric networking," *Computer Networks*, pp. 1–11, 2011.
[13] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate Models for General Cache Networks," *IEEE INFOCOM*, 2010.
[14] G. Carofiglio, M. Gallo, and L. Muscariello, "Bandwidth and Storage Sharing Performance in Information Centric Networking," in *ACM SIGCOMM, ICN Worskhop*, 2011.
[15] K. Y. Leung, E. W. M. Wong, and K. H. Yeung, "Designing efficient and robust caching algorithms for streaming-on-demand services on the internet," *ACM WWW*, 2004.
[16] C. Jayasundara, A. Nirmalathas, E. Wong, and N. Nadarajah, "Popularity-aware caching algorithm for video-on-demand delivery over broadband access networks." in *IEEE GLOBECOM*, 2010.
[17] D. De Vleeschauwer and K. Laevens, "Performance of caching algorithms for iptv on-demand services," *IEEE Trans. on Broadc.*, vol. 55, no. 2, pp. 491–501, 2009.
[18] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *ACM SIGCOMM IMC*, New York, NY, USA, 2007.
[19] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," in *IEEE INFOCOM, NOMEN Workshop*, Mar. 2012.
[20] ——, "Caching performance of content centric networks under multi-path routing (and more)," *Elsevier Computer Communication (to appear)*.
[21] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *ACM SIGCOMM IMC*, New York, NY, USA, 2007.
[22] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *IWQoS*, 2008.
[23] S. Khemmarat, R. Zhou, L. Gao, and M. Zink, "Watching user generated videos with prefetching," in *ACM MMSys*, New York, NY, USA, 2011.
[24] "A single customer that rated 17,000 movies," http://www.netflixprize.com/community/viewtopic.php?id=141.
[25] W. Bellante, "Analysis and enhancement of netflix streaming service basing on caching," Master thesis, Politecnico di Torino, 2012.