Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Discrete Mathematics 41 (2013) 221–228

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

Mathematical Analysis of caching policies and cooperation in YouTube-like services

Pablo Romero Franco Robledo Pablo Rodríguez-Bocca Claudia Rostagnol

{promero,frobledo,prbocca,crostag}@fing.edu.uy
Departamento de Investigación Operativa
Universidad de la República
Montevideo, Uruguay

Abstract

Currently, most video on-demand services offered over the Internet do not exploit the idle resources available from end-users. We discuss the benefits of user-assistance in video on-demand systems, where users are both clients and servers, helping with the task of video distribution. The mathematical machinery for the systematic analysis of video on-demand services is not mature yet. In this paper we develop a deterministic fluid model to determine the expected evolution of user-assisted on-demand video streaming services. We theoretically prove that cooperative systems always outperform non-cooperative solutions. A combinatorial optimization problem is proposed, where the goal is to distribute a set of video items into repositories trying to offer the minimum waiting times to end-users. This combinatorial problem is proved to be in the class of NP-Complete computational problem, and is heuristically solved with a GRASP methodology. Predictions inspired in YouTube scenarios suggest the introduction of cooperation is both robust and extremely attractive from an economical viewpoint as well.

Keywords: Video on-demand, Fluid model, Combinatorial Optimization Problem, GRASP.

http://dx.doi.org/10.1016/j.endm.2013.05.096

1 Motivation

Established in 2005, YouTube has become the most successful Internet site providing a new generation of short video sharing service, comprising approximately nearly 10% of all traffic on the Internet [1]. However, the network access is yet working with a client-server architecture, and the operator (Google corporation) must afford more than one million dollars per day just for bandwidth requirements, which is a clear motivation to exploit idle uploading resources from YouTube's users [4]. Experimental works also converge to the fact that user-assistance offloads the server and provides high scalability to video on-demand systems [3,12]. Peer-to-peer networks represent a promising alternative. The design of efficient scheduling policies must face several challenges: resources are highly dynamic, peers are heterogeneous (they have different broadband connections), and there are non-altruistic peers called free-riders (i.e peers who wish to exploit the resources of the whole system without contributing with it).

Scarce mathematical tools have been developed to understand the trade-offs in the design of user assisted on-demand systems. The first analytical model for a mesh-based P2P-VoD system was introduced in 2008 by Jue Lu et. al. [5]. It is similar in spirit to the classical file-sharing model from Qiu and Srikant [7], but proposed some ingredients inherent to VoD systems, with different file size, variable arrival rates and seeders aborting the system as a function of time, which makes it non-linear.

Our goal is to mathematically study the stability and performance of assisted on-demand services versus raw client-server architectures. Sequential systems represent an outstanding service, where users can download several video items serially (not simultaneously). A more ambitious service is the concurrent one, where users keep simultaneous downloads. For the sake of brevity, we will concentrate on the sequential service. It is important to emphasize that sequential systems possess the major practical interest (human beings do not watch two videos simultaneously). The reader can find an in-depth analysis of concurrent systems in [10]. This paper is structured as follows. Section 2 presents assisted and traditional sequential fluid models. Related theoretical results are summarized in Section 3. A caching problem is presented in Section 4, where the issue is to define the video items that should be stored in repositories to offer minimal waiting times to end-users. In Section 5, we take statistical information with a passive YouTube-crawler and compare our caching solution versus a traditional client-server architecture. Finally, concluding remarks and technological concerns are discussed.

2 Fluid models for on-demand video streaming

Consider an open system with K video items with sizes $\{s_1, \ldots, s_K\}$. Peers join the network, download video items sequentially and abort the system when they wish. Denote $x_j(t)$ the number of peers downloading video j in the current time t. They join the network following a poissonian process with rates λ_j, and abort the system with exponential law with rate θ. The number of peers seeding video j at instant t is denoted by $y_j(t)$, and depart the system exponentially with rates γ. We shall assume identical peers, with respective upload and download capacities denoted by μ and c. The exchange effectiveness between peers is a coefficient $\eta : 0 \leq \eta \leq 1$. Super-peers behave like seeders, but they do not leave the system. The number of super-peers seeding video j are denoted by z_j, and have upload capacity ρ. Super-peers as well as seeders upload video streams with fairness, and peers apply an incentive-based policy: their upload capacity is linearly related with the level of altruism. All this information can be summarized in a *Sequential Fluid Model (P2P-SFM)*, specified as follows:

$$\begin{cases} \dfrac{dx_j(t)}{dt} = \lambda_j - \theta x_j(t) - \min\left\{\dfrac{c}{s_j}x_j(t), \dfrac{\mu}{s_j}(\eta x_j(t) + y_j(t)) + \dfrac{\rho}{s_j}z_j\right\} & \text{(1a)} \\ \dfrac{dy_j(t)}{dt} = \min\left\{\dfrac{c}{s_j}x_j(t), \dfrac{\mu}{s_j}(\eta x_j(t) + y_j(t)) + \dfrac{\rho}{s_j}z_j\right\} - \gamma y_j(t) & \text{(1b)} \end{cases}$$

The *P2P-SFM* is just a balance of entrance and departure of peers in a fair system, including the presence of special components like seeders and super-peers, and cooperation in a BitTorrent fashion (the reader can find more general models in [10]). As soon as peers completely download the video stream, they are promoted to seeders, explaining the additive term on the right hand of (1b). The minimum function means that the bottleneck is either in the cumulative download or upload. This model represents an extension of [11], and was vastly studied in [9,10].

A traditional CDN (with a client-server paradigm) can be viewed as a particular case of this analytical approach. Specifically, users do not cooperate ($\mu = 0$) and super-peers are now static servers. Replacing $\mu = 0$ in (1a), the *Content Delivery Network-Sequential Fluid Model (CDN-SFM)* is defined by:

$$\frac{dx_j(t)}{dt} = \lambda_j - \theta x_j(t) - \min\left\{\frac{c}{s_j}x_j(t), \rho_j z_j\right\} \tag{2}$$

3 Stability and Performance Analysis

The *P2P-SFM* is a linear switched system of ordinary differential equations. By algebraic means, the rest-point can be found forcing $\frac{dx_j(t)}{dt} = \frac{dy_j(t)}{dt} = 0$:

$$\overline{x_j}_{SFM}^{P2P} = \max\left\{\frac{\lambda_j s_j}{\theta s_j + c}, \frac{\lambda_j(\lambda_j - \rho_j z_j - \frac{\mu\lambda_j}{\gamma s_j})}{\lambda_j(\theta + \frac{\eta\mu}{s_j} - \frac{\mu\theta}{\gamma s_j})}\right\}, \qquad \overline{y_j}_{SFM}^{P2P} = \frac{\lambda_j - \theta\overline{x_j}_{SFM}^{P2P}}{\gamma}, \tag{3}$$

Another reason to study sequential systems is the following result:

Theorem 3.1 *The P2P-SFM is always globally stable (proof in [10]).*

The proof is long and tricky, in a large way inspired by the proof from Dongyu Qiu and Wei Qian Sang [6], where the authors study a very similar system, but with a single file and no super-peer assistance. Theorem 3.1 is perhaps surprising, because we can tune all the parameters of the system and it will always converge to the rest-point. Denote T_{SFM}^{P2P} and T_{SFM}^{CDN} the expected waiting times under regime for the respective systems *P2P-SFM* and *CDN-SFM*. A valuable corollary from Theorem 3.1 and Little's law is the following:

Theorem 3.2 $T_{SFM}^{P2P} \leq T_{SFM}^{CDN}$

Proof. The rest-point for the *CDN-SFM* is just $\overline{x_j}_{SFM}^{CDN} = \overline{x_j}_{SFM}^{P2P}|_{\mu=0}$. Given that $\overline{x_j}_{SFM}^{P2P}$ is monotonically decreasing with μ, we get that $\overline{x_j}_{SFM}^{P2P} \leq \overline{x_j}_{SFM}^{CDN}$, and the equality holds only when both rest-points do not depend on μ. Denote $\lambda = \sum_j \lambda_j$ the sum rate. By Little's law the following chain of inequalities holds: $T_{SFM}^{P2P} = \sum_j \frac{\overline{x_j}_{SFM}^{P2P}}{\lambda} \leq \sum_j \frac{\overline{x_j}_{SFM}^{P2P}}{\lambda} = T_{SFM}^{CDN}$. □

So far, we know peer-assisted sequential systems are always stable and outperform raw client-server architectures. In the following section we will further explore the peer-assisted benefits, with scope in a major cause of concern in on-demand peer-to-peer systems: the video caching policies, where special nodes should store video items with constrained resources, in order to minimize the expected waiting time T_{SFM}^{P2P}.

4 Caching Problem

Let P be the number of super-peers (servers) in the system, u_n be the unit column vector of n elements (all its entries are 1), $s = (s_1, \ldots, s_K)^t$ the video sizes, and $S = (S_1, \ldots, S_P)^t$ the super-peers' storage capacity. The decision variable is a boolean matrix E of size $P \times K$, whose entries are $E(p,j) = 1$ if and only if we store video item j in super-peer p. We study the worst cooperative scenario, where seeders are selfish ($\gamma = \infty$). Then, we define the *Caching Problem* in matrix form as follows:

$$\min_E \sum_{j=1}^{K} \max \left\{ \frac{\lambda_j s_j}{\theta s_j + c}, \frac{\lambda_j s_j - \rho z_j}{\theta s_j + \eta \mu} \right\}$$

$$s.t. \begin{cases} E \times s \leq S & \text{(4a)} \\ E^t \times u_P = z & \text{(4b)} \\ z \geq 2 u_K & \text{(4c)} \\ E(p,j) \in \{0,1\}, \forall p \in [P], j \in [K]. & \text{(4d)} \end{cases}$$

The objective is to minimize the mean waiting times of all users in the system. It is worth to notice that the Caching Problem is useful for both $SFM-P2P$ and $SFM-CDN$ systems (where $\mu = 0$ in the latter). Constraint (4a) states that super-peers' storage capacity cannot be exceeded. Constraint (4b) relates the number of replicas z_j for video item j with the matrix E. Constraint (4c) imposes that each video item must be available in the network at least twice, whereas Constraint (4d) states E is a boolean matrix. A solution is feasible whenever all the previous constraints hold.

Theorem 4.1 *$CACHING-FEASIBILITY$ is NP-Complete.*

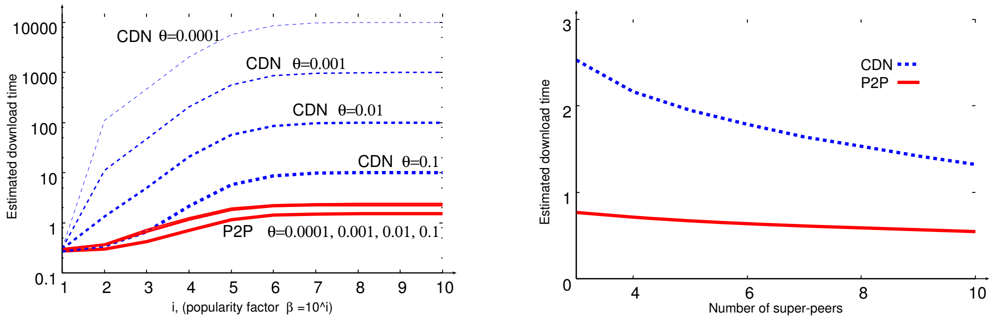
Proof. We can decide in polytime whether the solution is feasible or not, so $CACHING - FEASIBILITY$ is in NP. We will show that if we can determine feasibility of an arbitrary instance of the Caching Problem in polynomial time, then we would solve $PARTITION$ in polynomial time as well. Recall $PARTITION$ is an NP-Complete decision problem, where the issue is to find a subset of integers that sum the half of the global sum [2]. Consider an arbitrary set of positive integers $A = \{a_1, \ldots, a_n\}$ and let us call $a_{sum} = \sum_{i=1}^{n} a_i$. We construct the following Caching Problem instance, with $P = 3$, $S_1 = S$, $S_2 = S_3 = a_{sum}/2$ and $s_j = a_j$ for all $j \in [n]$. This transformation is polynomial. Since $S_1 + S_2 + S_3 = 2\sum_{i=1}^{n} s_i$, Constraint (4c) forces the three super-peers to store video items at their full capacity. Therefore, a feasible solution complies that $S_2 = \sum_B a_i = \sum_{B^C} a_i = S_3 = a_{sum}/2$, for a certain $B \subset A$. As a consequence, if $CACHING - FEASIBILITY$ can be solved for every instance in polynomial time, then every instance of $PARTITION$ can be solved in polynomial time as well. This concludes that $CACHING - FEASIBILITY$ is an NP-Complete decision problem. □

We solved the Caching Problem following the GRASP (Greedy Randomized Adaptive Search Procedure) metaheuristic [8]. The algorithm works as follows. We first sort the content respect to a cost-to-benefit ratio $w_i = s_i \overline{x_i}^{P2P}_{SFM}$, and iteratively pick the two super-peers with highest remaining resources to store these videos, respecting Constraint 4a, until Constraint 4c holds. Therefore, a feasible solution is found (provided that superpeers are resourceful), and a classical local search is applied to delete, add or swap videos into super-peers, until a local minimum waiting time is finally returned. The reader is referred to [9,10] for details. In the remainder, the GRASP algorithm is applied in both systems ($P2P - SFM$ and $CDN - SFM$), with real-life instances taken from YouTube.

5 Results and Conclusions

5.1 Application to a real scenario: YouTube

We further explore the performance and robustness properties in real-life scenarios for both P2P and CDN architectures, under the naive GRASP caching algorithm. For that purpose, we captured statistical information of nearly 60.000 video files using a passive YouTube crawler, in order to contrast P2P network with raw CDN technology. We use selfish seeders ($\gamma = \infty$)



(a) Increasing Popularity. (b) Increasing cache-servers.

Fig. 1. Download time for CDN and P2P.

and medium effectiveness $\eta = 0,5$, and perform a statistical estimation of video popularities λ_j, stressing the system with an artificial popularity factor $\beta = 10^m$ ($m \in \{1, \ldots, 10\}$) to the vector $(\lambda_1, \ldots, \lambda_K)$. In this way, we can contrast the performance of a CDN vs P2P deployment under different scenarios. We use abortion rates $\theta = 10^{-n}$ with $n \in \{1, 2, 3, 4\}$, a common download rate of $c = 1$ Megabytes per second, $\mu = c/4$ following asymmetrical broadband accesses from end-users, and $P = 4$ super-peers (or servers) with $\rho = 10$ Megabytes per second each, holding $K = 59000$ video items.

Figure 1(a) shows the performance of both P2P and CDN models versus β. It underlines three essential features. First, the expected time for a P2P sequential system is never worse than the one of a raw CDN technology, as predicted by Theorem 3.2. Second, the performance of both systems presents no gap for low-populated scenarios, but peer savings are remarkable under high-populated ones. Third, the expected waiting time is highly sensitive to abortions in a CDN system, whereas the P2P system shows to be more stable, outstanding its high scalability.

Figure 1(b) illustrates the expected waiting time for both P2P and CDN systems (with solid and dashed lines respectively) versus P, when $\beta = 10^3$ and $\theta = 0,1$. It helps to figure-out robustness: how the system's performance can be affected in terms of single point of failures. For a fixed popularity factor we want to find the mean waiting time for different number of super-peers (servers). From this experiment, we conclude that P2P systems are resilient under environmental failures, while a raw CDN can be considerably damaged when a server fails.

5.2 Conclusions

In this paper, a general framework for the analysis of sequential video on-demand assisted services is provided. Under this framework of expected evolution, the sequential system is always globally stable, converging to a known rest-point. We found closed expressions for the expected waiting times in both CDN and P2P approaches, and we theoretically proved that the peer-to-peer philosophy always outperforms traditional CDNs. An experimental validation of the P2P and CDN systems and their performance is presented regarding real-traces passively taken from a YouTube crawler. The results are encouraging, showing that a P2P assisted platform preserves its resilience against adverse environments like flash crowds. Our trends for future work include the stability and capacity analysis in concurrent video on-demand assisted scenarios.

References

[1] Cheng, X., Dale, C., Liu, J.: Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study. Tech. rep., Cornell University (2007)

[2] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Company, New York, NY, USA (1979)

[3] Hefeeda, M., Bhargava, B.K., Yau, D.K.Y.: A hybrid architecture for cost-effective on-demand media streaming. Computer Networks **44**(3), 353–382 (2004)

[4] Huang, C., Li, J., Ross, K.W.: Can internet video-on-demand be profitable? SIGCOMM Computer Communication Review **37**(4), 133–144 (2007)

[5] Lu, Y., Mol, J.D., Kuipers, F., Mieghem, P.V.: Analytical Model for Mesh-Based P2PVoD. In: Proceedings of the 2008 Tenth IEEE International Symposium on Multimedia, ISM '08, pp. 364–371. IEEE Computer Society, Washington, DC, USA (2008). URL http://dx.doi.org/10.1109/ISM.2008.30

[6] Qiu, D., Sang, W.: Global stability of Peer-to-Peer file sharing systems. Comput. Commun. **31**(2), 212–219 (2008)

[7] Qiu, D., Srikant, R.: Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In: Proceedings of SIGCOMM'04, pp. 367–378. ACM, New York, NY, USA (2004). DOI http://doi.acm.org/10.1145/1015467.1015508

[8] Resende, M.G.C., Ribeiro, C.C.: Greedy Randomized Adaptive Search Procedures. In F. Glover and G. Kochenberger, editors, Handbook of Methaheuristics, Kluwer Academic Publishers (2003)

[9] Robledo, F., Rodríguez-Bocca, P., Romero, P., Rostagnol, C.: A new caching policy for cloud assisted peer-to-peer video-on-demand services. In: Proceedings of the 12th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P'12) (2012)

[10] Romero, P.: Mathematical analysis of scheduling policies in Peer-to-Peer video streaming networks. Ph.D. thesis, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay (2012). URL http://www.fing.edu.uy/inco/pedeciba/bibliote/tesis/tesisd-romero.pdf

[11] Tian, Y., Wu, D., Ng, K.W.: Analyzing Multiple File Downloading in BitTorrent. In: Proceedings of ICPP'06, pp. 297–306. IEEE (2006). DOI 10.1109/ICPP.2006.23

[12] Yu, H., Zheng, D., Zhao, B.Y., Zheng, W.: Understanding user behavior in large-scale video-on-demand systems. SIGOPS Oper. Syst. Rev. **40**(4), 333–344 (2006)