# Hot set identification for social network applications

Claudia Canali, Michele Colajanni, Riccardo Lancellotti
*Department of Information Engineering*
*University of Modena and Reggio Emilia*
*Email: {claudia.canali, michele.colajanni, riccardo.lancellotti}@unimore.it*

***Abstract*—Several operations of Web-based applications are optimized with respect to the set of resources that will receive the majority of requests in the near future, namely the hot set. Unfortunately, the existing algorithms for the hot set identification do not work well for the emerging social network applications, that are characterized by quite novel features with respect to the traditional Web: highly interactive user accesses, upload and download operations, short lifespan of the resources, social interactions among the members of the online communities.**

**We propose and evaluate innovative combinations of predictive models and social-aware solutions for the identification of the hot set. Experimental results demonstrate that some of the considered algorithms improve the accuracy of the hot set identification up to 30% if compared to existing models, and they guarantee stable and robust results even in the context of social network applications characterized by high variability.**

***Keywords*-Social networks; Predictive algorithms; Performance evaluation**

## I. Introduction

Social networks represent a new class of Web-based applications that include interaction, upload of content, knowledge and resource sharing among communities of online users. The popularity of these applications spans from traditional community-oriented applications, such as blogs [1], to services originally designed as simple data repositories, such as Flickr and YouTube [2], [3].

To guarantee high performance and scalability, even social network applications rely as usually on *content replication*, *caching*, *resource pre-adaptation*, and *CDN delivery* [4]–[7]. These data management tasks are expensive and cannot be applied on the entire data set. Hence, their common goal is to primarily operate on a specific data subset: the so called *hot set*, that corresponds to the set of resources that are likely to receive most requests in the near future [8], [9]. The dimension of the data set, its periodic evaluation, and other parameters depend on the application, workload and underlying architecture. However, we can evidence that existing algorithms for hot set identification that take into account just basic statistics on past resource accesses [9], [10] achieve good results in the context of traditional Web-based applications where the resource popularity changes slowly and according to known patterns. On the other hand, they do not achieve acceptable results when applied to recent social network applications. The main motivation is that the working set of this novel applications consists also of resources supplied by the users, with thousands of new uploads every hour. As a consequence, the lifespan of resources tends to decrease, the resource popularity changes rapidly and it is affected by social relations among the users [1], [11], [12]. The novel user access patterns and the inherent variability of social network workload motivates our research on innovative algorithms explicitly designed to cope with the workload peculiarities of social network applications.

To provide an accurate and robust hot set identification in social network applications, we propose algorithms that exploit predictive and social-aware techniques. Prediction techniques based on time series have been already used in the estimation of Internet traffic [13], server load [14], and hot spots [15], but they have never been applied to resource popularity evaluation. Furthermore, the observation that the access patterns are strongly affected by social connections [12], [16] suggests that the user rank in the social network may be exploited to predict the popularity of the uploaded resources.

In this paper we propose three new classes of runtime algorithms that provide periodical identification of the hot set, namely *Predictive*, *Social-aware* and *Predictive-Social*, providing and example for each class. The Predictive algorithm addresses the fast changes of resource popularity by using the past access patterns to predict the most popular resources in the next future. In this paper, we apply short-term and runtime predictive solutions to the novel context of social network applications, where the workload variability may hinder the effectiveness of the hot set identification algorithms that do not consider the dynamic evolution of workload characteristics. The Social-aware algorithm exploits the knowledge about the user social network. Finally, the Predictive-Social algorithm identifies the hot set by combining information from predictive models and user social network. We demonstrate that the proposed Predictive-Social algorithm achieves a twofold benefit: it improves of up to 30% the accuracy of the hot set identification with respect to existing algorithms with results that are close to the best theoretical algorithm; even more important in a context where the workload is highly variable, the proposed algorithm provides robust performance for all the considered workload scenarios and parameters.

The remainder of the paper is organized as follows. Section II describes the main challenges in identifying the resource hot set for social network applications. Section III presents the proposed algorithms for hot set identification. Section IV describes the experimental results. Section V concludes the paper with some final remarks.

## II. Hot set identification

The quality of the hot set identification depends on the ability to predict the future accesses to each resource by means of information available at the server side.

We consider applications requiring a periodic hot set identification with period $\Delta t$. At time $t$ an algorithm for hot set identification computes, among all resources of the working set $R(t)$, the subset $HS(t)$ containing the resources that are expected to receive the highest number of accesses in the future interval $[t, t + \Delta t]$. A typical algorithm works in three steps. For each resource $r \in R(t)$, it estimates the popularity $p_r(t)$ and sorts the resources according to their popularity. The hot set $HS(t)$ contains the first $Z$ resources of the popularity list, where $Z$ is the dimension of the hot set in terms of number of resources. The hot set of resources is important for the performance of traditional and innovative Web-based applications because multiple management operations (e.g., replication, caching, pre-adaptation, pushing) are applied to it.

The necessity of a periodic evaluation derives from the observation that during the period $[t, t + \Delta t]$ the working set and the resource popularity is likely to change due to user content upload and variations in the access patterns. Consequently, after an interval $\Delta t$, the algorithm is executed again to identify a new hot set $HS(t + \Delta t)$.

Hot set identification exploits the well-known presence of power-law distributions [17], [18] in most workload characteristics of Web-based applications. However, unlike traditional Web-based applications, the workload characteristics of social networks in terms of upload/download patterns [3], [16] limit or nullify the effectiveness of the existing solutions for hot set identification. Let us consider the main challenges that novel algorithms should address.

**Rapid variations of the access patterns**. The short lifespan of resources and the high rate of content uploaded determine sudden changes in the resource popularity. Traffic surges for a resource may be experienced few minutes after the upload [3].

**Workload dynamics related to the social interactions** among the members of online communities. Most users browse social networking sites following social links (e.g., contacts in user pages) [16]. The consequence is a strong correlation between user connections in the social network and popularity of the uploaded resources, that is not considered by existing solutions for hot set identification.

**Variability in the workload characteristics**. The variety of social network applications is extremely vast and heterogeneous, and workload models range from blogs to multimedia content exchange. It is important that the solutions for hot set identification are robust and provide stable performance with respect to several workload parameters.

## III. Algorithms for hot set identification

Algorithms for the hot set identification may consider a large amount of heterogeneous information, such as access history, tags, resource title, ratings and other resource metadata, social links and user interests. In this paper, that for the first time considers the peculiarities of the social network workloads for hot set identification, we focus on algorithms that rely on the knowledge of resource access patterns and user social links. Our choice to consider a subset of any possibly available information derives from two considerations: increasing the amount of information does not always improve algorithm performance and robustness; furthermore, merging multiple heterogeneous data is a not trivial task and may increase the computational complexity to the extent that the algorithms may result inapplicable to a runtime context.

We consider four classes of algorithms for the hot set identification, namely *Existing*, *Predictive*, *Social-aware*, and *Predictive-Social*. The class of existing algorithms derives from the literature on Web resource replication and caching, and from current practices in social network applications [9], [10]. The Predictive, Social-aware and Predictive-Social algorithms represent the original contributions of this paper.

### A. Existing algorithms

In the context of traditional Web-based applications, the future popularity of the resources is mainly determined on the basis of the past access patterns, in terms of absolute number, frequency, or freshness of the accesses [6], [9]. As a basis for comparison, we consider an algorithm that estimates the resource popularity by taking into account just the past access patterns.

The considered algorithm assumes that the popularity of a resource $r$ at time $t$ corresponds to the number of received requests during the last time interval $[t - \Delta t, t]$ as in [6]. Hence, the resource popularity $p_r(t)$ can be expressed as:

$$p_r(t) = \frac{d^r(t)}{\Delta t} \tag{1}$$

where $d^r(t)$ is the number of accesses received by the resource $r$ in the time interval $[t - \Delta t, t]$. The popularity estimation based on a short time period allows this algorithm to be aggressively reactive to workload changes.

### B. Predictive algorithms

The novel class of the Predictive algorithms represents an evolution with respect to the existing algorithms for hot set identification because information on the past access patterns is used at time $t$ to predict the future accesses

to each resource in the interval $[t, t + \Delta t]$. There is a plethora of predictive models, but the choice of the most appropriate model depends on the context characteristics. As the algorithm has to work at runtime in highly variable scenarios, we are interested to simple but robust models that are able to adapt their behavior to unstable workload conditions. Our choice goes to a model that represents the past accesses to each resource $r$ as a time series and applies the Exponential Weighted Moving Average (EWMA) function for the evaluation of $p_r(t)$. The EWMA model adopts weighting factors decreasing exponentially for older data points [13] and gives more importance to recent values while not discarding older observations. This algorithm is characterized by a low computational cost that is suitable to runtime contexts.

For each resource $r \in R(t)$, the Prediction algorithm defines the past resource accesses as a time series of $n$ elements $D^r = \{d^r(t), d^r(t - \Delta t), \ldots, d^r(t - (n-1)\Delta t)\}$, where $d^r(t)$ is the number of accesses to the resource $r$ in the interval $[t-\Delta t, t]$, $d^r(t-\Delta t)$ is the number of accesses in the interval $[t-2\Delta t, t-\Delta t]$, and so on until $d^r(t-(n-1)\Delta t)$ that is the $n$-th element of the time series.

The Predictive algorithm evaluates the resource popularity $p_r(t)$ by estimating the frequency of accesses in the future interval $[t, t + \Delta t]$, that is $p_r(t) = \hat{d}^r(t + \Delta t)/\Delta t$, where $\hat{d}^r(t + \Delta t)$ is the expected number of accesses in $[t, t + \Delta t]$ evaluated through the EWMA model [13]:

$$\hat{d}^r(t + \Delta t) = \gamma \hat{d}^r(t) + (1 - \gamma) d^r(t), \tag{2}$$

where $\gamma = \frac{2}{n}$ represents a typical choice for EWMA-based prediction.

### C. Social-aware algorithms

This is a completely new class of algorithms that compute the hot set by exploiting information on the user social network characteristics. Several studies on social network applications suggest that many accesses to the resources are based on user navigation through social links (e.g., links from the page of a community member to another). Hence, there is a strong correlation between the amount of accesses to a resource and the number of user social links. We should also consider that most social network applications allow users to designate other members of the online community as their contacts. Hence, the number of *reverse contacts* of the users can be considered as a measure of their social network size, where a reverse contact for the user A is a user that has designated A as a contact [12]. We define the *connection degree* of each user as the number of his reverse contacts within the online community. The basic idea is that the resources uploaded by users with high connection degree are likely to receive more accesses [12], [16].

For each resource $r$, the proposed Social-aware algorithm considers the connection degree $c_r(t)$ of the user that uploaded the resource as a basis for the estimation of $p_r(t)$. Furthermore, since the popularity of the resources may rapidly decrease over time in the context of social network applications [11], this algorithm takes into account the age $a_r(t)$ of the resource $r$. In this way, the most recently uploaded resources are more likely to be included in the hot set. The algorithm estimates the popularity of a resource as:

$$p_r(t) = \frac{c_r(t)}{c_{max}(t)} \cdot \frac{1}{a_r(t)} \tag{3}$$

where $c_r(t)$ and $c_{max}(t)$ are the connection degree of the user that uploaded the resource $r$ and of the user with the highest number of reverse contacts within the social network, respectively.

### D. Predictive-Social algorithms

This is the most original class of algorithms combining predictive and social metrics. Merging these types of information is a not trivial task, due to their inherent heterogeneity in terms of temporal dependencies and probability distributions. We propose an algorithm that combines predictive and social-aware information.

The Predictive-Social algorithm uses a linear function to merge the popularity estimations obtained from the Predictive and the Social-aware algorithms. Figure 1 shows the main steps of the algorithm: for each resource $r$, it evaluates the social metric $p_{r,social}(t)$, deriving from the Social-aware algorithm, and the predictive metric $p_{r,predictive}(t)$, based on the Predictive algorithm. The values are normalized so that $p_{r,social}(t)$ and $p_{r,predictive}(t)$ are in the range $[0, 1]$.

The popularity of a resource $r$ is obtained through the following linear combination:

$$p_r(t) = \delta(t) p_{r,predictive}(t) + (1 - \delta(t)) p_{r,social}(t) \tag{4}$$

The weight $\delta(t)$ is evaluated as a non-linear combination of the values of the sets $P_{predictive}(t) = \{p_{r,predictive}(t), \forall r \in R(t)\}$ and $P_{social}(t) = \{p_{r,social}(t), \forall r \in R(t)\}$, containing the popularity of the resources of the entire working set at time $t$ computed by the Predictive and Social-aware algorithms, respectively.

We have to consider that median and average values are not representative in a context where different heavy-tailed distributions must be combined. Hence, we combine the heterogeneous measures by means of the *two-sided quartile-weighted median* (QWM) metric [19], that is a statistical function robust and independent of any assumption on the distribution of the measures:

$$QWM(P(t)) = \frac{Q_{75}(P(t)) + 2 * Q_{50}(P(t)) + Q_{25}(P(t))}{4} \tag{5}$$

where $Q_i$ denotes the $i$-th quantile of the values in the data set $P(t)$. We compute $\delta(t)$ as:

$$\delta(t) = \frac{QWM(P_{social}(t))}{QWM(P_{predictive}(t)) + QWM(P_{social}(t))} \tag{6}$$

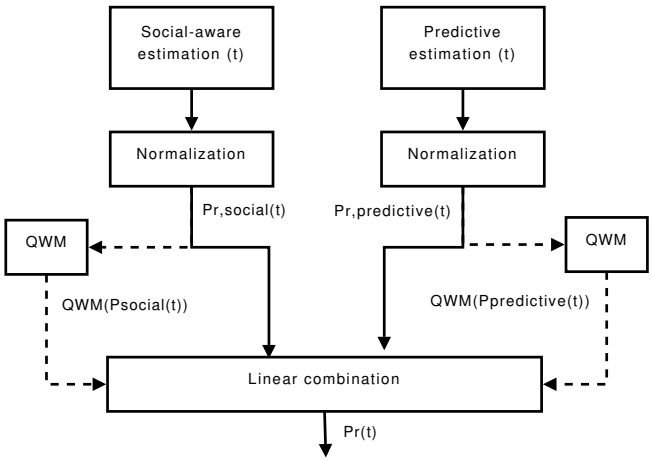


Figure 1. Scheme of the PS-Quartile algorithm

## IV. Experimental results

### A. Experimental testbed

We evaluate the performance of the algorithms for hot set identification through a discrete event simulator based on the Omnet++ framework. We implement a simulated Web system where a server receives requests from a population of concurrent users.

As it was impossible to have access to logs of popular sites of social network applications, we rely on models based on the state-of-the-art workload characterizations. A client interaction with the server may involve the download or the upload of a resource, where the percentage of upload operations may range from 1% up to 20%. This large range includes the case of highly interactive social network applications, such as blogs, where users typically post lots of comments for every blog entry [1]. The amount of upload operations affects the working set size and the resource popularity. Any new upload, indeed, increases the working set size. Furthermore, a high number of uploads tends to cause a turnover within the hot set because recently uploaded resources may replace previously popular contents. In the upload operations, the workload model considers also the dependency of the resource popularity on the user connection degree in the social networks. Accordingly to [16], the correlation factor between resource popularity and user connection degree (namely, *user/resource popularity correlation*) may range from 0.6 to 0.8.

For our experiments we consider a user population that grows from 10000 to 20000 individuals during the experiment duration. Users issue requests to the server for an incoming traffic rate of 100 requests per second on average. The results are collected over runs of 12 hours of simulated time. Each experiment is repeated 10 times and the results are averaged over the runs. We choose a hot set identification interval $\Delta t = 20$ minutes, that is commonly adopted in these contexts, but we should observe that the choice of the best period $\Delta t$ for the hot set identification may represent an interesting research topic by itself. The naive solution of addressing the high variability of the hot set through very short periods of hot set re-evaluation is not feasible, because it would cause an excessive overhead in terms of computational power, storage space and network bandwidth.

To evaluate the effectiveness of the hot set identification, we consider as a term of comparison the so called ideal hot set $HS^*(t)$. It is obtained by a theoretical algorithm that at time $t$ knows the future access patterns in the interval $[t, t+\Delta t]$ for every resource in the working set but it does not take into account resources uploaded after $t$. We consider as the main performance metric the algorithm *accuracy*, that is the ratio $|HS(t) \cap HS^*(t)|/Z$, averaged over the experiment duration. The number $|HS(t) \cap HS^*(t)|$ measures how many resources have been correctly identified by an algorithm, while $Z$ is the number of resources in the hot set and it is used for normalization purposes.

The *robustness* of the results is another critical parameter for the evaluation of the algorithms. Due to the high variability of the workload, solutions that may guarantee stable performance over a wide set of scenarios are preferable with respect to algorithms that achieve the best peak performance for a specific scenario and poor performance elsewhere. To this purpose, we perform a sensitivity analysis with respect to several workload and algorithm parameters: the *hot fraction*, that is the size of the hot set as a fraction of the entire working set (that is $Z/|R(t)|$), the *upload percentage*, that is the percentage of upload operations over the total number of client requests, and the *user/resource popularity correlation*, that is the correlation between resource popularity and user connection degree. Table I summarizes the range and default values in the experimental setup.

Table I
Experimental setup parameters

| Parameter | Range | Default |
|---|---|---|
| Hot fraction [%] | 5% – 30% | 20% |
| Upload percentage [%] | 1% – 20% | 5% |
| User/resource popularity correlation | 0.6 – 0.8 | 0.7 |

### B. Evaluation of algorithms for hot set identification

We initially evaluate the accuracy of the algorithms for the hot set identification for different values of the hot fraction. The primary goal is to evaluate if considering predictive and social-aware metrics may actually improve the performance of the hot set identification in the context of social network applications.

Figure 2 compares the accuracy of the algorithms as a function of the hot fraction. We observe that the Existing algorithm achieves very poor performance with respect to all the other algorithms. On the other hand, the Predictive and

Social-aware algorithms achieve good and similar performance, always above 80% for every value of the hot fraction. The good performance of the Predictive algorithm is due to its forecasting model, that allows the algorithm to be more reactive to changes in the resource popularity with respect to existing solutions for the hot set identification. The result about the Social-aware algorithm confirms the intuition that exploiting information about the social user interactions may guarantee a good accuracy in the estimation of the future resource popularity. However, the best performing algorithm is the Predictive-Social, that consistently outperforms the other solutions. This result is important because it shows that the combination of predictive and social metrics represents a promising solution to improve the effectiveness of the hot set identification.

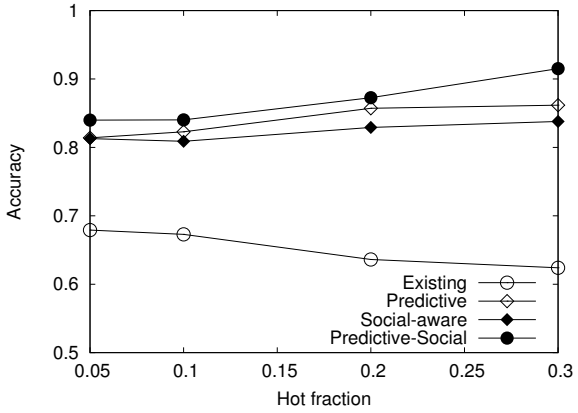


Figure 2. Performance evaluation

We now evaluate how the workload variability can affect the performance of the algorithms for the hot set identification. We consider two important parameters: the upload percentage and the user/resource popularity correlation.

Figure 3 shows the algorithm accuracy as a function of the upload percentage. We observe that high amounts of uploads have a negative effect on the Existing and Predictive algorithms. In particular, the Existing algorithm shows the highest sensitivity to the upload percentage, with an accuracy nearly halved as the upload percentage passes from 1% to 20%. This result was expected because the Existing algorithm considers only the most recent data for the estimation of the future popularity. Hence, a high turnover in the hot set increases the noise in the data values that are used for the popularity estimation, with a consequent negative effect on the resulting accuracy. The Predictive algorithm is less sensitive to the upload percentage, even if its accuracy decreases of about 20% as the upload percentage grows. The lower sensitivity of the Predictive algorithm with respect to the Existing algorithm is due to the moving average function, that acts as a filtering technique that reduces the noise effects. As expected, the Social-aware algorithm is almost insensitive to the workload variability because it relies on information about the user connection degree, that is characterized by a less dynamic behavior with respect to the upload frequency. However, the most interesting result is about the Predictive-Social algorithm, that achieves the best and the most robust performance, with an almost negligible variation of the accuracy as the upload percentage ranges from 1% to 20%.

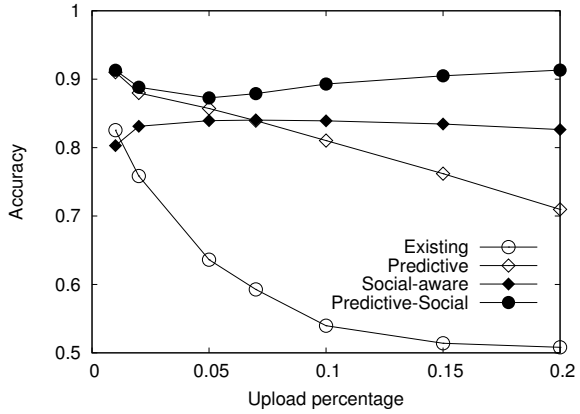


Figure 3. Sensitivity to upload percentage

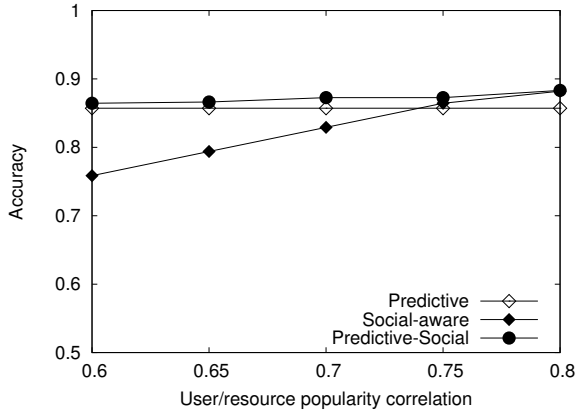


Figure 4. Sensitivity to user/resource popularity correlation

It is interesting to evaluate whether and to which extent the user/resource popularity correlation influences the accuracy of the Social-aware and Predictive-Social algorithms. Figure 4 shows the algorithm accuracy as a function of the user/resource popularity correlation. In this evaluation we do not consider the Existing algorithm due to its poor and unstable performance. We observe that the Social-aware algorithm outperforms the Predictive algorithm for correlation greater than 0.75, but its accuracy is highly sensitive to the variations of the popularity correlation parameter, while, as expected, the Predictive algorithm is insensitive to this social parameter. The Predictive-Social algorithm confirms best

accuracy and robust results for every value of user/resource popularity correlation.

From the previous analyses we can conclude that Predictive and Social-aware algorithms may represent an initially appreciable solution to the problem of identifying the hot set in the context of social network applications, but their results are not robust for any workload scenario. In highly variable and heterogeneous contexts, achieving robust performance is even more important than obtaining high peak performance for small ranges of workload parameters. The combination of multiple techniques allows us to address both the issues of peak performance and robustness. In the Predictive-Social algorithm the overall accuracy is improved for every value of the upload percentage, and the robustness of the results is improved to the extent that this algorithm is almost insensitive to workload variability and social correlation.

## V. Conclusions

This paper proposes novel algorithms for the hot set identification in the context of social network applications. The algorithms consider predictive and social-aware solutions to estimate the future resource popularity. The combination of these types of information allows this Predictive-Social algorithm to achieve performance close to that of the ideal algorithm, and to show robust results with respect to a wide range of workload scenarios and algorithm parameters.

The Predictive-Social algorithm investigated in this paper represents just a first step towards the exploration of new algorithms for hot set identification in the context of social network applications. Which information is convenient to consider and how it is better to combine the different metrics remain interesting open issues that deserve further investigation.

## References


[1] F. Duarte, B. Mattos, A. Bestavros, V. Almeida, and J. Almeida, "Traffic Characteristics and Communication Patterns in Blogosphere," in *Proc. of ICWSM Conference*, Seattle, WA, Apr. 2007.

[2] J. Li, S.-F. Chang, M. Lesk, R. Lienhart, J. Luo, and A. W. M. Smeulders, "New challenges in multimedia research for the increasingly connected and fast growing digital society," in *Proc. of ACM SIGMM MIR Workshop*, Sep. 2007.

[3] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "YouTube traffic characterization: A view from edge," in *Proc. of IMC Conference*, Oct. 2007.

[4] A. Davis, J. Parikh, and W. E. Weihl, "EdgeComputing: extending enterprise applications to the edge of the Internet," in *Proc. of WWW Conference*, 2004, pp. 180–187.

[5] S. Podlipnig and L. Böszörmenyi, "A survey of Web cache replacement strategies," *ACM Computing Surveys*, vol. 35, no. 4, pp. 374–398, 2003.

[6] M. Karlsson, "Replica placement and request routing," *Web content delivery*, 2005, (Tang, Xu, Chanson eds.), Springer.

[7] C. Canali, M. Colajanni, and R. Lancellotti, "Performance Evolution of Mobile-Web based Services," *IEEE Internet Computing*, Mar./Apr. 2009.

[8] M. Colajanni, R. Lancellotti, and P. Yu, "Distributed architectures for Web content adaptation and delivery," *Web Content Delivery*, 2005, (Tang, Xu, Chanson eds.), Springer.

[9] M. Rabinovich and O. Spatscheck, *Web Caching and Replication*. Addison Wesley, 2002.

[10] S. Sivasubramanian, G. Pierre, M. van Steen, and G. Alonso, "Analysis of Caching and Replication Strategies for Web Applications," *IEEE Internet Computing*, vol. 11, no. 1, pp. 60–66, 2007.

[11] 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 *Proc. of ACM SIGCOMM Conference*, San Diego, CA, Oct 2007.

[12] K. Lerman, "Social Information Processing in News Aggregation," *IEEE Internet Computing*, vol. 11, no. 6, pp. 16–28, 2007.

[13] A. Sang and S.-Q. Li, "A predictability analysis of network traffic," in *Proc. of IEEE INFOCOM Conference*, Mar. 2000.

[14] M. Andreolini, S. Casolari, and M. Colajanni, "Models and framework for supporting runtime decisions in Web-based systems," *ACM Transactions on the Web*, vol. 2, no. 3, pp. 1–43, Aug. 2008.

[15] Y. Baryshnikov, E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. Yimwadsana, "Predictability of Web-server traffic congestion," in *Proc. of WCW Workshop*, Sep. 2005.

[16] K. Lerman and L. Jones, "Social Browsing on Flickr," in *Proc. of ICWSM Conference*, Mar. 2007.

[17] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. of IEEE Infocom 1999*, Mar. 1999.

[18] T. Yamakami, "A Zipf-Like Distribution of Popularity and Hits in the Mobile Web Pages with Short Life Time," in *Proc. of PDCAT Conference*, Taipei, Taiwan, 2006.

[19] N. G. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *Proc. of IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.