Measurement, Modeling, and Analysis of a Large-scale Blog Sever Workload

Myeongjae Jeon*, Jeaho Hwang†, Youngjae Kim‡, Jae-Wan Jang†, Joonwon Lee§ and Euiseong Seo¶
*Rice University, †KAIST, ‡Oak Ridge National Laboratory, §Sungkyunkwan University, ¶UNIST
*mjjeon@rice.edu, †{jhhwang, jwjang}@calab.kaist.ac.kr, ‡yk7@ornl.gov, §joonwon@skku.edu, ¶euiseong@unist.ac.kr

***Abstract*—Despite the growing popularity of Online Social Networks (OSNs), the workload characteristics of OSN servers, such as those hosting blog services, are not well understood. Understanding workload characteristics is important for optimizing and improving the performance of current systems and software based on observed trends. Thus, in this paper, we characterize the system workload of the largest blog hosting servers in South Korea, Tistory[1]. In addition to understanding the system workload of the blog hosting server, we have developed synthesized workloads and obtained the following major findings: (i) the transfer size of non-multimedia files and blog articles can be modeled by a truncated Pareto distribution and a log-normal distribution respectively, and (ii) users' accesses to blog articles do not show temporal locality, but they are strongly biased toward those posted along with images or audio.**

***Keywords*-Online Social Networks (OSNs); Workload Characterization; Measurement; Modeling;**

I. Introduction

The emergence of Online Social Networks (OSNs) is completely changing the way the Internet end-users participate in the Web environment. Before the advent of the OSNs, the Internet end-users were a simple consumer of Web content, which was made of static images, photos, videos, and text files. However, a new class of Web applications – blog and wiki services running on the social networking platforms, have made the Internet end-users a Web content provider as well as a consumer of Web content [1].

As the content sharing using OSNs has become extremely popular, recent years have witnessed the performance issues of server systems that host OSN services. Specifically, previous studies have examined the negative impact of OSN workloads on server performance in terms of CPU usage [2], cache miss behavior [3], and scalability in multicore architectures [4]. Thus, there was a need for optimizing and improving existing server systems hosting the OSN services.

Understanding the workload characteristics of the hosting servers is critical to improving the current systems and designing new systems for the OSN services. However, there has been a lack of research on the workload characterization of the OSN servers. Thus, we collected the Web access logs from Tistory's [5] blog hosting servers and characterized them. Tistory is one of the most popular large-scale blog servers in South Korea. The log data was collected over a 12-day period and is composed of more than 96 million HTTP requests and 4.7 TB of network bandwidth.

We have made several important and noteworthy observations in our work. First, we find that unlike conventional Web workloads [6] [7], the file and transfer size distributions are not heavy-tailed due to multimedia files generated/shared by end-users. However, for non-multimedia files such as HTML and image files, the distributions are heavy-tailed. Second, we find that the user references on blog articles do not show temporal locality, which means that the articles read recently are not likely to be read in the near future. Third, we find that the users prefer to read articles posted along with images and audio. In addition to these findings, we also approximated the characteristics of blog content generation/consumption using several statistical models.

The rest of this paper is organized as follows: Section II presents the basics of blog services and workload collections along with their running system environment. We present the analysis of file-level characteristics in Section III and the analysis of article access logs in Section IV respectively. We discuss related work in Section V. Finally, we summarize our work in Section VI.

II. Background

Up-to-date blogs allow users to post various contents along with text (e.g. photographic images, videos, and audio). The main content in a blog is a series of *articles* (called *posts* or *entries*) written by the blog author. These articles are typically arranged in reverse chronological order with the most recent article appearing at the top of a blog. Due to this inherent organization, the information on new content is easily accessible to potential visitors. There are several ways in which a blog user can reach new blog content:

- *Direct Visit* - a user polls the blogs of interest by typing the URL in the browser to check if new content has been posted.
- *Web Feed* - new content of a blog is published via web feeds and subscribed to by users who registered with a *feed reader*.
- *Blog Search Engine* - blog search engines identify blogs, index content, and allow users to search for blogs, their authors, and the relationships among them.
- *Trackback* - a user can follow content links spanning several blogs networked through trackbacks.

[1] *Tistory* is an Internet website in South Korea. It provides free blog hosting services in South Korea. The website: http://www.tistory.com/

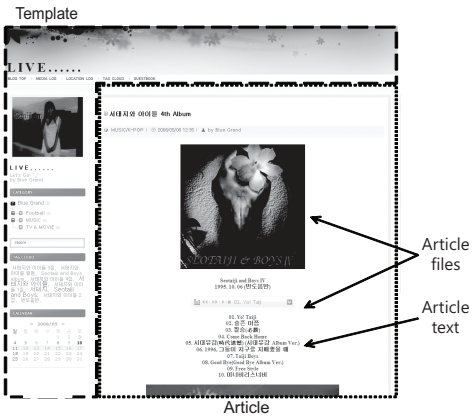


Figure 1. Example of a blog page

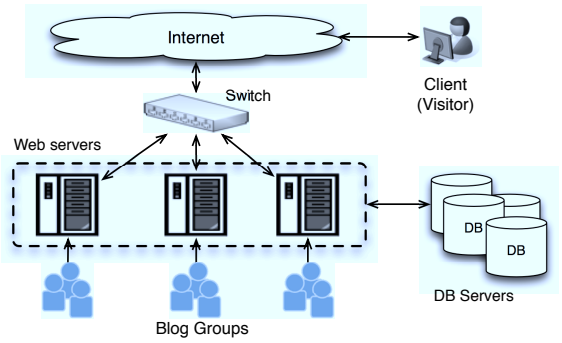


Figure 2. The architecture of a blog service system.

All four methods described above not only facilitate the grouping of the blog community, but also ultimately accelerate the interactions between content producers and consumers.

In general, a *blog page* (a Web page) displays the content with logically uniform structure, as represented in Figure 1. The *template* is used to form the design of a blog page using a combination of images, CSS, and JavaScript. The *article* is the unit of blog posting and consists of *article text* and *article file*; the article text is simply text written by a blogger, whereas the article file refers to files uploaded when a blogger publishes an article. Article files can additionally be sorted into *article image*, *article audio*, and *article video*, according to the file type. All blogs that use the same blog template always show identical backgrounds on the blog page. Thus, if a user visits several blog pages in a certain blog, only the article is changed because these pages share the same template.

Blog servers provide users with blog content that is produced in two manners; one is dynamic and the other is static. While the *dynamic content* is generated on-the-fly by server-side scripting when it is sent to the clients, the *static content* is stored in disk and fetched using file system operations. In Tistory's case, generating blog pages requires server-side scripting to dynamically embed text such as article text, comments, and replies, which are all stored in server database. Besides such text, all other content in the blog page is managed by server file system, thus belonging to static content.

The blog service system of Tistory consists of a set of switches, Web servers, database nodes, as shown in Figure 2. A switch node controls the incoming requests and forwards them to the Web servers depending on their destination. Each Web server manages a group of blogs and responds to all the incoming HTTP requests. The server finally retrieves dynamic or static content in response to the HTTP requests.

For example, an HTTP request for a blog page itself gets an HTML file generated dynamically by having a Web server submit queries of necessary article text, comments, and replies to database nodes. Then, requests for the embedded files of the blog page arrive later, and the Web server issues file system requests to the local storage of the sever to retrieve those files.

We collected the access logs of blog servers over 12 consecutive days. Every log entry is composed of several fields; those fields are explained in Table I. After eliminating unsuccessful requests from the access logs as done in [8], a total of 50,118,065 requests were obtained, wherein we observed 948,290 visitors and 11,629 blogs over that period of time.

In this paper, we restrict our analysis on static content for several reasons. First, the naming of dynamic content changes over time. This inhibits us from characterizing all the files accessed. Second, static content is a major element that influences the performance of the OSN hosting servers. This is because static content greatly increases stall cycles due to data cache misses [3] and significantly impacts the response time to clients and the server bandwidth usage [9].

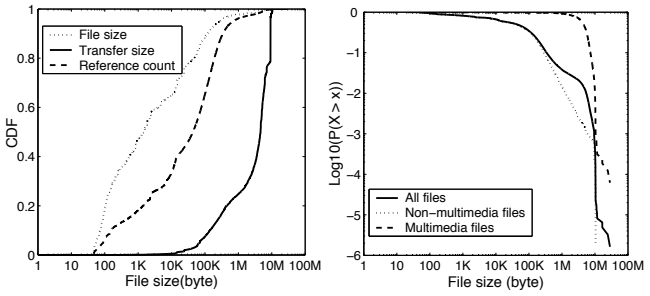
Field	Description
VirtualHost	particular domain assigned to a blog
ClientHost	client who issued the request
Date&Time	time the request was received
Request	actual HTTP request line indicating the requested file and its HTTP method
StatusCode	HTTP response code signifying the result of processing a client request
Size	content-length of the transferred file

Table I
DESCRIPTION OF LOG ENTRY'S FIELDS. EVERY LOG ENTRY FOLLOWS EXTENDED COMMON LOG FORMAT (ECLF).

III. FILE-LEVEL ANALYSIS OF STATIC CONTENT

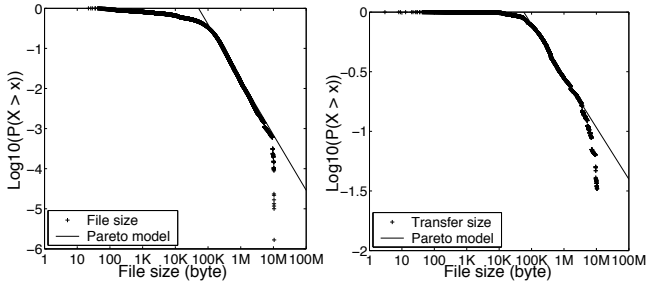
In this section, we show our analysis of static content in workloads[2]. The static content can be categorized into two major file types, multimedia and non-multimedia files.

[2]We use the terms 'static content' and 'static files' interchangeably.



(a) CDF (b) LLCD

Figure 3. (a) Cumulative Distribution Functions (CDFs) of static content with respect to file size, reference count, and transfer size. (b) Log-Log Complementary Distributions (LLCDs) of static content with respect to file size. Note that (b) shows three different lines for multimedia, non-multimedia file types and sum of these.



(a) File Size (b) File Transfer Size

Figure 4. LLCD transformation for non-multimedia files.

	File transfer size	File size
tail index(α)	0.43	1.37
goodness-of-fit(R^2)	0.98	0.99

Table II
APPROXIMATION RESULTS FOR FILE SIZE AND TRANSFER SIZE DISTRIBUTIONS OF NON-MULTIMEDIA FILES.

The multimedia content includes audio and video files, and the non-multimedia content includes static content such as images, JavaScript, CSS, and Flash.

A. File Size and File Transfer Size

Figure 3(a) presents a CDF of static content with respect to file size, reference count, and transfer size. In the figure, the file transfer size denotes aggregate bandwidth usage for a specific file size. It has been simply calculated by $file\ size \times reference\ count$.

From Figure 3(a), we observe that a small number of large files requires high server bandwidth, whereas a large number of small files requires low server bandwidth. For instance, 92% of total static files are smaller than 400KB, whereas their bandwidth demands are less than 20% of total server bandwidth. A major part of the bandwidth consumption is generated by the large-size article files posted by users; in particular, among the unique files larger than 400KB, photographic images (59%) and audio files (33%) are the two most prevalent types.

Moreover, in Figure 3(a) we see that the file size distribution visually follows a heavy-tailed distribution, whereas the file transfer size distribution does not show the visual trend of a heavy-tailed distribution. Thus, we confirm our visual finding from the figure by modeling the distributions with a Pareto distribution [10], which is a well-known model for representing the heavy-tailed distribution. Thus, we transformed the CDF plots of file size and transfer size to the Log-Log Complementary Distribution (LLCD) plots [10] and investigated their R^2 *goodness-of-fit* test values.

From our modeling results, we found that the LLCDs of file size and transfer size could not be modeled in a Pareto distribution. Surprisingly this observation is completely against those observed in conventional Web workloads where the file size and the transfer size show heavy tailed distributions [6], [7], [8]. For further analysis, we attempted to model the LLCD distributions of file size and transfer size for multimedia and non-multimedia files. We found that only non-multimedia files showed heavy-tailed distributions for their file size and transfer size.

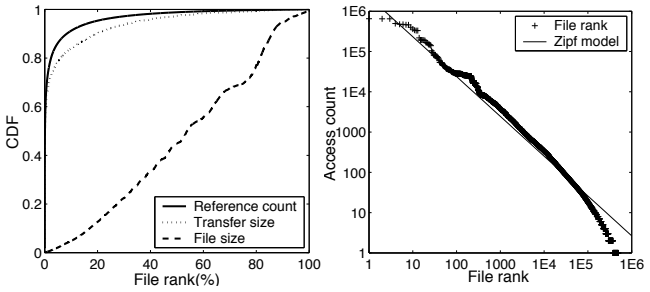
Figure 4 provides the LLCD of those files for both file size and transfer size. From the figure, we see that both can be neatly modeled in a truncated Pareto distribution [11] where there exists a natural upper bound that curtails the tail. The tail index (α) is shown in Table II with the confidence level (R^2). The tail length of the LLCD ranges from an order of magnitude for file transfer size to three orders of magnitude for file size[3]. Although the distributions strongly fit in the Pareto distribution model with high R^2 values, each α presents a fairly different value. Table II shows that α is 0.43 for the file transfer size and 1.37 for the file size, indicating that the distribution of file transfer size is much more heavy-tailed than that of the file size. Obviously, this result is mainly due to the dominant bandwidth usage by photographic images, which is 59% of the total.

The implication of our data is that conventional Web servers may not be perfectly suited for blog service workloads. The file size and transfer size of Web workloads have been known to be heavy-tailed in several studies. However, the large-scale blog workload shows that both the file size and transfer size distributions are heavy-tailed only for non-multimedia content. Assuming all the blog content files are managed by a central server, the blogs may generate significantly different request patterns to the underlying server systems.

B. File Referencing Behavior

To analyze file referencing behaviors, we measured the relationship between popular files and their actual storage

[3]Our blog service has a restriction on the file size which should not exceed 10MB. This might affect the length of the tails.



(a) CDF (b) Zipf Analysis

Figure 5. Concentration of references for ranked files: (a) CDF of file size, reference count, and transfer size, (b) Zipf analysis (access count versus rank).

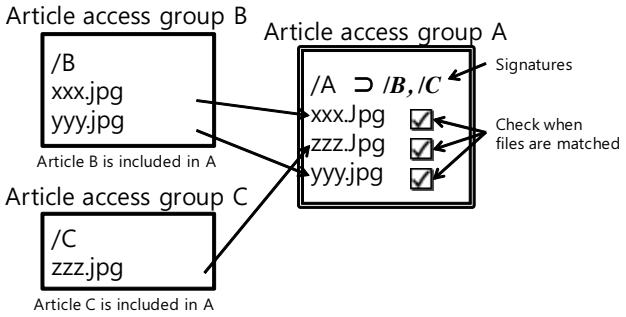


Figure 6. File matching method when an article access group A contains accesses to unique articles, B and C.

and network bandwidth usages. For this purpose, files are sorted by their reference counts and the most popular file is assumed to be the highest in rank. Disk space usage, network usage, and reference count are then accumulated into a CDF graph, as shown in Figure 5(a). We see from the figure that 10% of the most frequently referenced files are responsible for 84% of bytes transferred over the network and 91% of total requests. The files, however, only account for 4.8% of the overall consumption in the storage.

Previously we observed that both large files (>400KB) and popular files (top <10%) influenced a huge volume of traffic. This is because 70% of total bandwidth is used by the files that lie on the intersection of the two groups.

A typical method of approximating the popularity distribution is to exploit Zipf's law [12]. According to Zipf's law, the number of file references shows a consistent decline as the file rank decreases. If (R) represents the rank of a file, then the number of references (P) is: $P \backsim R^{-\beta}$ where β is the slope value. The formula shows that the number of references is inversely proportional to the rank of the file. In other words, if the popularity follows Zipf's law, a linear shape is shown in the plot by placing log-scaled R on the horizontal-axis and log-scaled P on the vertical-axis. Figure 5(b) shows that Zipf's law approximates the popularity of our traces well, with β and R^2 goodness-of-fit value close to 0.99 and 0.98 respectively.

IV. Article-level Analysis of Static Content

This section explains the characteristics of the static content at the article level, with each characteristic being approximated by a theoretical distribution.

A. Methodology Overview

We collected *article access logs* in two phases. The first phase takes the blog page organization into account, as discussed in [9], to get *article access groups*. A blog page consists of the main blog page and its related embedded files. Thus, the user's request for a single main blog page causes the transfer of multiple embedded files from the blog servers. The following explains the detailed procedure for extracting the article access groups:

Step (1). Select a collection of file access logs corresponding to a unique pair of *VirtualHost* and *ClientHost* fields in log entries.

Step (2). Gather requests for a main blog page and all its embedded files until another main blog page is requested or a file request is submitted over a time threshold. We set the time threshold to be 5 seconds; we based this threshold on our inspection of typical request intervals.

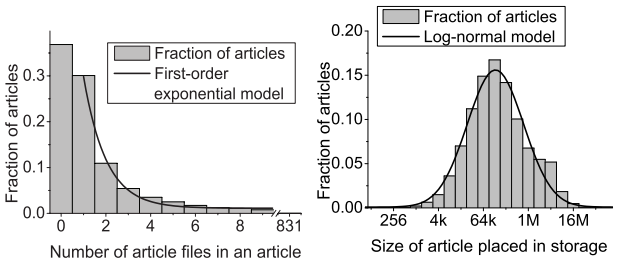
Step (3). Given the requests from Step (2), use the requests for the article files as a request for an article access group. Then, go to Step (2) for picking up requests for another article access group.

Step (4). Return to Step (1) and repeat until all file access logs are checked.

Some article access groups acquired from the first phase include the reference for several articles. For instance, the front page of the blog sometimes contains more than one article to encourage visitors to read several recent articles. Using the information in the raw access logs, we cannot recognize how many and which articles are on a page. Thus, we use the second phase to rectify such article access groups.

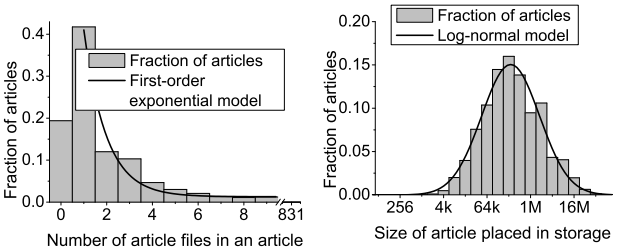
The second phase utilizes a *file matching* method, as shown in Figure 6. Suppose that we have three article access groups, A, B, and C, each having article files identified by their name. By comparing the article file names in A and B, we can determine that A includes B. In the same manner, we can decide that C is also included in A. Finally, A is divided into two article access logs, B and C. Using this method, we refined article access groups from the first phase and obtained article access logs.

After eliminating a few articles (less than 3%) with unknown size, we finally found a total of 87,332 unique articles with 235,586 article files stored in the blog servers and 1,088,210 accesses, which accompany 2,909,894 article file transfers, for such articles during the trace period.



(a) (b)

Figure 7. Histograms of unique articles: (a) the number of article files, (b) article size.



(a) (b)

Figure 8. Histograms of accesses to unique articles: (a) the number of article files, (b) article size.

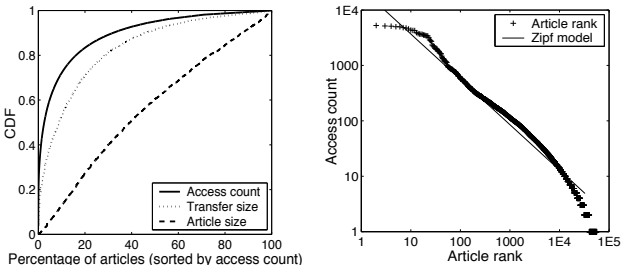
B. Characteristics of Blog Articles

In addition to characterizing blog articles in terms of their sizes, we further analyze the inter-reference time of article accesses to investigate whether subsequent accesses exhibit temporal locality.

Unique Article: Figure 7(a) shows a histogram of unique articles with respect to the number of article files. The article contains 2.74 article files on average and the median and the standard deviation are 1 and 8.89 respectively. The number of articles having less than 4 article files occupies 87% of the total, indicating that bloggers prefer to publish their own articles with a small number of article files. We can approximate the histogram in Figure 7(a) with a first order exponential distribution ($0.01+0.77e^{-x/1.0}$) except for zero-file articles (i.e. articles with only text). The goodness-of-fit (R^2) value of 0.99 shows a strong fit to the distribution. This approximation explains that articles exist in blogs with an exponentially decreasing rate as the number of article files increases.

Figure 7(b) shows a histogram of unique articles with respect to their sizes. Here, the average size is 1.1MB, the standard deviation is 4.6, and the median size is 203KB. Articles are mostly located within a range of 32KB∼2MB, accounting for approximately 74% of the total. We can approximate the histogram with a log-normal distribution [13] ($\mu = 12.3, \sigma = 1.80$, natural log), as shown in Figure 7(b). It shows a strong R^2 value of 0.98.

Access Frequency of Unique Article: Figure 8(a) shows a histogram of articles with regard to their access frequency.



(a) (b)

Figure 9. Concentration of references for ranked articles.

The average access frequency is 2.67, the standard deviation is 6.12, and the median is 1. The number of articles composed of less than 4 article files accounts for 86% of the total. When the percentage of access for zero-file articles is compared with that of unique articles, we observe that blog visitors prefer to read articles posted along with article files such as image and audio files. The histogram of Figure 8(a) can also be approximated with a first order exponential distribution ($0.01 + 1.10e^{-x/0.98}$) except for zero-file article accesses. The R^2 value of 0.97 shows a strong fit to the distribution.

Figure 8(b) shows a histogram of the size of articles. The average size of articles is 1.07MB, the standard deviation is 3.46, and the median is 202KB. Articles are aggregated within a range of 32KB∼2MB, occupying approximately 74% of the total. We can approximate the histogram with a log-normal distribution ($\mu = 12.3, \sigma = 1.80$, natural log), as shown in Figure 8(b). It shows a strong R^2 value of 0.98.

Similar to exponential distribution, the log-normal distribution is common when the average is low and the variance is large. Our analysis shows that articles' access and creation patterns of most users are highly concentrated in small articles or articles with a small number of article files.

Article Referencing Behavior: We conducted the rank-based analysis, as discussed in Section III-B. We first sorted the articles by their access frequency. In Figure 9(a), we show the CDFs of the storage and network bandwidth usages, and access frequency of the article files. The graphs show that 10% of most frequently accessed articles are responsible for 73% of total reference counts and 53% of bytes used to transfer total articles. This rate is much smaller than the value observed in our analysis of file popularity that 90% of requests and 84% of bytes are concentrated on the top 10% of most frequently requested files. Due to the diluted concentration on the articles, modeling the popularity with Zipf's law yields a lower slope value than the case of file popularity. The β and the R^2 are 0.82 and 0.98 respectively, as plotted in Figure 9(b).

Temporal Locality: We found that the access patterns of articles exhibited weak temporal locality. Figure 10 shows the CDF of inter-reference time of articles. We observed that the articles that were revisited within 30 minutes and 1

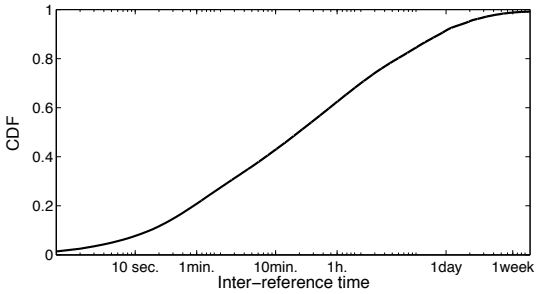


Figure 10. CDF of inter-reference time for articles.

hour account for 31% and 62% of the total, respectively. A noticeable trend in the graph is a monotonous increase as the interval is shifted toward higher values, and thus no hot spot is detected for the re-referencing pattern. One might expect that the temporal locality would be shown in article accesses since the top 10% of popular articles take up 73% of total reference counts. We speculate that this decayed locality may be strongly affected by active content creation, where article references are distributed among various articles.

V. Related Work

Several studies have focused on the patterns or evolution of user generated content in the OSNs. Guo *et al.* [14] studied knowledge-sharing OSNs like blogs. They found that the user posting behavior in these OSNs exhibits strong daily and weekly patterns and that the posted content is not mainly contributed by a small number of users. Cha *et al.* [15] studied information propagation in Flickr and showed that popular photos do not spread widely and quickly throughout the network. Burke *et al.* [16] analyzed server log data from Facebook and found that newcomers who see their friends contributing go on to share more content themselves. Moreover, for newcomers who are initially inclined to contribute, receiving feedback and having a wide audience are also predictors of increased sharing. Leskovec *et al.* [17] studied the temporal aspects and topological patterns of information propagation in blogs, and found that the popularity of blog posts drops with a power law. In addition, they showed that the size distribution of cascades (number of involved posts) follows a perfect Zipf's distribution.

The prior work most similar to ours is that of Duarte *et al.* [18]. They analyzed HTTP requests from a Brazilian blog service and covered characteristics of user session requests on blogs, blogs accessed, and aggregated accesses to the blog server. One of their two findings in the server context is relevant to ours; file transfer size can be modeled by a Pareto distribution. As a complementary study to [18], this work underscores the importance of the comprehensive analysis of aggregated server accesses in the server context.

At a much lower semantic level, there have been studies examining the (negative) impact of the workload of OSNs on server performance. Ohara *et al.* [4] showed that requests arising from user's contributory nature of OSNs often retrieve and update persistent data, leading to reduced performance and inhibiting us from exploiting multicore architectures. Nagpurkar *et al.* [3] found that stalls due to data cache misses form the dominant component of stall cycles. In this regard, our analysis lays out a foundation to guide the design of better server architectures to host OSNs.

VI. Conclusion

We have presented an empirical analysis of real-world Web access logs collected from Tistory (one of the largest Korean blog hosting sites) over 12 consecutive days. We have not only investigated user activities of blog servers but also studied the behavior of user-created content and their article distribution. In our analysis, we observed detailed characteristics and modeled them using statistical distributions. We believe that the observations presented in this paper will provide useful guidelines when modeling OSN workloads, performing workload synthesis, and designing new systems for hosting OSNs.

References

[1] "Social networks and blogs now 4th most popular online activity," in *Nielsen Reports*, 2009.
[2] C. Stewart, M. Leventi, and K. Shen, "Empirical examination of a collaborative web application," in *IEEE IISWC*, 2008.
[3] P. Nagpurkar, W. Horn, U. Gopalakrishnan, N. Dubey, J. Jann, and P. Pattnaik, "Workload characterization of selected JEE-based web 2.0 applications," in *IEEE IISWC*, 2008.
[4] M. Ohara, P. Nagpurkar, Y. Ueda, and K. Ishizaki, "The data-centricity of web 2.0 workloads and its impact on server performance," in *IEEE ISPASS*, 2009.
[5] *Tistory site. http://www.tistory.com*.
[6] C. W. A. Williams, M. Arlitt and K. Barker, "Web workload characterization: Ten years later," *Web Content Delivery. Springer*, 2005.
[7] A. Oke and R. B. Bunt, "Hierarchical workload characterization for a busy web server," in *TOOLS*, 2002.
[8] M. F. Arlitt and C. L. Williamson, "Internet web servers: workload characterization and performance implications," *IEEE/ACM Trans. Netw.*, 1997.
[9] M. Jeon, J. Hwang, J. Jang, E. Seo, and J. Lee, "Characterization of a large-scale blog traffic," in *TR/CS-2009-300 KAIST*, 2009.
[10] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, 1997.
[11] I. B. Aban, M. M. Meerschaert, and A. K. Panorska, "Parameter estimation for the truncated pareto distribution," *Journal of the American Statistical Association*, 2006.
[12] G. K. Zipf, "Human behavior and the principle of least-effort," *Addison-Wesley*, 1949.
[13] E. Limpert, W. A. Stahel, and M. Abbt, "Log-normal distributions across the sciences: Keys and clues," *BioScience*, 2001.
[14] L. Guo, E. Tan, S. Chen, X. Zhang, and Y. E. Zhao, "Analyzing patterns of user content generation in online social networks," in *ACM SIGKDD*, 2009.
[15] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *WWW*, 2009.
[16] M. Burke, C. Marlow, and T. Lento, "Feed me: motivating newcomer contribution in social network sites," in *ACM CHI*, 2009.
[17] J. Leskovec, M. Mcglohon, C. Faloutsos, N. Glance, and M. Hurst, "Cascading behavior in large blog graphs," in *SDM*, 2007.
[18] F. Duarte, B. Mattos, A. Bestavros, V. Almeida, and J. Almeida, "Traffic characteristics and communication patterns in blogosphere," in *ICWSM*, 2007.