Characterization of a Large-Scale Blog Traffic

Myeongjae Jeon
KAIST

Jeaho Hwang
KAIST

Jaewan Jang
KAIST

Euiseong Seo[†]
Pennsylvania State University

Joonwon Lee[‡]
Sungkyunkwan University

CS/TR-2009-300

February 16, 2009

K A I S T
Department of Computer Science

{mjjeon,jhhwang,jwjang}@calab.kaist.ac.kr, [†]euiseong@cse.psu.edu, [‡]joonwon@skku.edu

Characterization of a Large-Scale Blog Traffic

Myeongjae Jeon, Jaeho Hwang, Jaewan Jang, Euiseong Seo, and Joonwon Lee

Abstract

This report presents a detailed characterization study of massive blog traffic. To our knowledge, ours is the first comprehensive study that covers (i) file-level, (ii) article-level and (iii) user behavioral characteristics. We analyzed Web access logs of Tistory, a large-scale blog hosting service operating in South Korea. The access logs contained on average 8 million requests and 400 GB of data transfer per day. By examining this traffic, we observed several unique characteristics of the blog traffic directly affected by multimedia content. For example, we found that multimedia content accounted for most of Web requests, and music content yielded an exceptionally large traffic volume. It also showed that the transfer size and the file size of non-multimedia content followed a heavy-tailed distribution. Furthermore, visitors prefer to read articles posted along with multimedia content such as image and music files. To provide data for generating synthetic workloads, many of the observed characteristics are approximated by probability distributions.

1 Introduction

The advent of Web 2.0 has changed the way Internet users participate in the Web environment. Before Web 2.0, users generally consumed content supplied by a limited number of content producers. In the Web 2.0 era, however, the separation between consumers and producers is diminishing, because users are encouraged to produce content using easy tools or services necessary to publish content. It has become a major trend for users to create content for other users through various Web 2.0 services on the Internet, including blogs, wikis, and video sharing sites.

Among these Web 2.0 services, the blog is a representative way to exchange ideas, primarily by publishing and reading articles, and thus has become one of the major communication methods for Internet users. As of February 2008, there were 113 million blogs on the Internet and more than 1.6 million posts and 175 thousand new participants per day [1]. Most blogs are now collectively serviced by a few large companies, such as MySpace, Google Blog, Facebook, equipped with blog hosting platforms including authoring tools, templates, configuration tools, and Web server systems. Despite this feverish popularity of blogs, there has been little research activity focusing on the characterization of large-scale blog traffic. Our main goal is to fill this gap and to provide insights into how blog content is supplied and distributed on the Internet.

An in-depth understanding of Web service traffic has proved essential in designing efficient Web service architectures in the past [2][3]. Moreover, recent work on Web 2.0 video sharing sites has led researchers to propose the use of observed characteristics to build more efficient content distribution systems [4] and has implications for network and service providers [5]. Likewise, an empirical traffic does provide first-hand statistics for capacity planning and management of blog hosting servers. This also provides data for generating synthetic workloads.

Blog services exhibit the patterns of information production and consumption distinct from those of traditional Web services, such as Internet news portals or corporation websites. First, the clients generally see the same content repeatedly in the traditional Web servers because they usually serve news or magazine articles produced by the limited number of writers. Thus, the

serviced file set is relatively small and has little dependency on the hit count even if a Web server records a million hits a day. However, the large-scale blog hosting incorporates a large number of diverse writers who actively post articles periodically. As a result, the number of files to be served increases proportional to the number of writers. Second, up-to-date blogs are equipped with various content other than text: photoblogs, vlogs, and musicblogs are utilized as platforms for posting high-quality images, videos, and music, respectively. Based on our observation, these various content attract visitors to articles posted along with such content.

To identify the characteristics of blog traffic in detail, this report presents the anatomy of a real-world massive blog traffic using Web access logs from Tistory, one of the most popular blog services in South Korea [6]. The blogs hosted by the service generate 8 million requests and 400 GB of network traffic per day. Tistory service is organized in typical configurations: clustered Web servers, authoring tools and various templates. Therefore, the log data we utilized enables us to identify common aspects of blog services.

We follow three-step approaches for the analysis of blog traffic. First, we present a high-level statistical analysis of the traffic to highlight user behavioral characteristics. Second, we examine the characteristics of static content, which is a major source of blog traffic, focusing on user activities on blogs, file size and transfer size distributions, file popularity and referencing characteristics. Lastly, we provide characteristics of articles that bridge the interaction between content producers and consumers; in blog services, many bloggers actively post articles and share them with numerous clients via feeds and periodic visits. To obtain article workloads from raw log files, we devise a methodology that groups files in the article based on those relationships.

The results from this analysis reveal unique characteristics of blog traffic. For instance, media files account for most of Web requests and music files have disproportional large traffic volume. It is also shown that the size of non-multimedia files follow heavy-tailed distribution. Furthermore, visitors prefer to read articles posted along with multimedia content such as image and music files.

The main contributions of this work are summarized as follows:

1. We provide an extensive analytic study of large-scale blog traffic, of which the results complement well-known Web server characteristics.

2. By providing probability distributions that approximate the observed characteristics, we aid in understanding the underlying mechanisms that bring about such characteristics. Furthermore, the estimated characteristics would be used for the generation of synthetic workloads.

3. Understanding blog traffic aid in network management, capacity planning, and the design of new systems.

The rest of this report is organized as follows: Section 2 describes additional background of blogs and the traffic used for this study. We presents statistical analysis of our traffic in Section 3 and detailed characteristics of static files in Section 4. We describe our methodology for extracting article access logs from the raw Web access logs and analyze these article access logs in Section 5. After we provide related work in Section 6, we conclude this report in Section 7.

2 Background

2.1 Overview of Blogs

Blogs are one of the most widespread social network communities, where authors describe their thoughts and news on various subjects, such as travel, fashion, education, music, and politics.

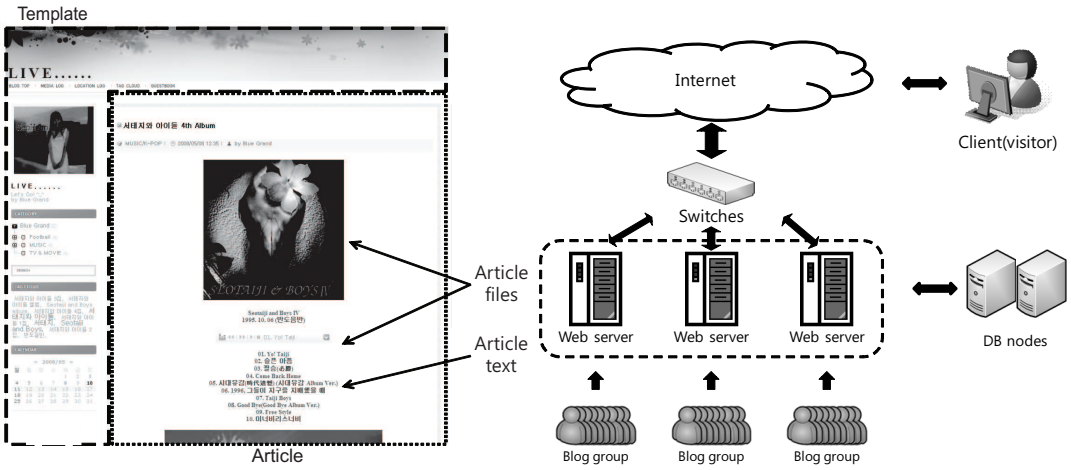


Figure 1: (a) An example of a blog page that consists of a template and an article, (b) Blog hosting server architecture. Web servers cooperate with DB nodes to process incoming requests that are multiplexed by switches. A Web server manages a particular group of blogs.

The content on a blog consists of **articles**, also sometimes called **posts**, written by the blog author. Since articles are typically listed chronologically on a separate page and are published via feeds and polled by the users, newly posted articles are prominently displayed to potentially interested readers.

A blog provides two efficient ways for bloggers to actively interact with each other. One allows visitors to leave **comments** about a specific article, and the other enables bloggers to refer and to link their articles to other bloggers' articles. The latter is called **trackbacks**, and is an efficient method to find bloggers who share similar interests. Readers of an article thus can freely follow links to another blogger's articles through links of the trackbacks. Note that the two methods play a leading role in accelerating interactions between article producers and consumers.

2.2 Blog Pages

A blog page, which is simply shown as a Web page, has a logically uniform structure in the case of most blog sites, as represented in Fig 1(a). Generally, the content in a blog page can be classified into two groups: **template** and **article**. First, a blog template is used to form the design of a blog page using a combination of images, CSS, and JavaScript. Second, a blog article, the unit of a blog post, consists of **article text** and **article file**; the article text is simply text written by a blogger, whereas the article file refers to files posted when a blogger publishes an article. Article files can additionally be sorted into **article image**, **article music**, and **article video**. While most blog hosting companies offer pre-designed templates, bloggers can also freely customize and distribute the templates. Obviously, all blogs that use the same blog template always show identical backgrounds on the blog page. Therefore, if a visitor reads blog pages in a single blog, only blog articles excluding the blog template is changed because the pages share the same template.

Blog servers provide blog content to client browsers that are produced in two manners, termed **dynamic** and **static**. While dynamic content is generated on the fly by server-side scripting before being sent to the client, static content is stored in a disk and fetched using file system operations. The blog servers create all HTML files while they are viewed, except for 'Not

Table 1: Summary of collected log files

Item	Value
Period	12 days
Total requests	96,168,202
Average requests per day	8,014,017
Total visitors	948,290
Average visitors per day	108,473
Total blogs visited	11,629
Average blogs visited per day	6,475

Found' error messages. The HTML files are fully built after the server-side scripting embeds article text as well as other text such as comments, replies, and some statistical information; all text is stored in server databases. Besides the article text, therefore, all other files in the blog page belong to the category of static content.

A blog page includes only one article in most cases. However, if a blogger changes the blog's parameters using configuration tools, access of a particular page may retrieve several articles conglomerated in that page. This poses a challenge to article analysis, as each article in the page should be separated. The details of the methodology to address this challenge are explained in Section 5.

2.3 Data Collection of Blog Workload

The architecture of blog service system consists of a set of switches, Web servers, and database nodes, as shown in Fig 1(b). Each Web server manages a group of blogs and responds to all Web requests headed for the corresponding blog group. An incoming request is first forwarded to one of the Web servers by a switch. The server then processes the request, which requires static or dynamic content. Web servers additionally raise queries to database nodes if there is a request for a HTML page, which needs various textual content stored in the databases.

The raw material for the workload under our analysis is Web server log files. The log data used in this work were collected from one of the largest blog services in South Korea for 12 consecutive days. A summary of the data is presented in Table 1. In total, 96,168,202 valid transactions are logged, wherein we monitored 948,290 visitors and 11,629 blogs. Since the transactions for those requests are logged in each server, the collected log files maintain all system-wide events occurring among all blogs, including client references to templates and articles in blog pages. We do not install reverse proxies in front of Web servers. Therefore, the workload analyzed here is not biased and represents the accessing behavior that most blog services experience.

A logging entry has the following information:

- *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
- *Bytes*: content-length of the file transferred

Table 2: HTTP methods poplarity

Method	Total requests	Prevalence
GET	94,801,206	98.58%
POST	1,314,064	1.37%
Other	52,932	0.05%

- *ProcessingTime*: time taken to process a client's request

Two fields are extended from Extended Common Log File Format(ECLF) for our analysis. *VirtualHost* is extended in order to identify the blog where the file request is headed, as it is necessary to match an incoming file request with the blog owning the file of the request. To measure and analyze the time that the blog servers consume to process each request, we extend *ProcessingTime*, which describes the interval between the request arrival and its completion. In particular, we use the *ProcessingTime* field to determine what sort of requests cause bottlenecks in the blog servers in terms of time to process the requests.

Unfortunately, the period is relatively short to show the long-term tendencies of workload changes. However, we believe the large quantity of log entries and the stability of the workload observed in this period provide the analysis with strong statistical certainty.

3 User Behavioral Characteristics

This section presents a high-level statistical analysis of the data collected for this study. We analyzed the overall features of the blog workload for comparison with the characteristics of workloads from traditional Web services and Youtube, the world's largest Web 2.0 video sharing site. These features include the activity of blog users, file popularity for proxy caching, file type properties, and the effects of static and dynamic content on blog servers.

3.1 Users' Activities on Blogs

We begin with an analysis of the users' activities on blogs by delving into HTTP methods acquired from the *Request* field in log entries. Table 2 presents a breakdown of HTTP methods seen in our blog workload. We discover that GET requests occupy a major portion of the total, achieving a 98.58% prevalence rate. This indicates that the requests from blog users are principally for fetching content exposed on blog articles because a GET message is used to retrieve files from the Web server. Additionally, the POST method accounts for 1.37% of the employed HTTP methods in our workload, which amounts to about 1.3 million requests. In blogs, POST requests are available for posting comments, making trackbacks, and uploading content. The most popular use of the POST is author requests for uploading content such as article images, article music, and article videos, which comprise a total of 348,530 attempts. The behaviors of posting comments and enabling trackbacks follow uploading content with 317,211 and 187,067 trials, respectively.

Although the fraction of POST methods appears to be insignificant, it clearly reveals the eagerness of users to participate in publishing articles and online interactions. For instance, in the case of Youtube traffic gathered from a university, only 0.12% of requests are POST [5], an order of magnitude less than the observed POST requests from the blogs. Furthermore, the 1998 World Cup Web site, a traditional Web site, showed a significantly smaller portion of POSTs at 0.06%, as determined by measurement[1].

[1]The log data is available at "http://ita.ee.lbl.gov/html/contrib/WorldCup.html".

Table 3: StatusCode popularity

StatusCode	Successful	Not Modified	Client error	File redirection	Other	Total
Prevalence	53.8%	41.7%	2.4%	0.6%	1.5%	100%

The number of article reads and writes provides additional insight into blog users' activities. These statistics provide very precise information on the activity of readers and writers of the article. Identifying these two different operations for articles is accomplished by parsing HTTP request lines and relating the keywords extracted from the parsing to the operations. We finally obtained 1,088,210 and 24,855 events for read and write of articles, respectively. With normalization to the 'write' value, this means that 44 articles are, on average, referenced when an article is posted. Our blog service shows less intensive reading for the unit posted (i.e. the article) when compared with videos from Youtube: Youtube serves daily 100 million video views with 65,000 new video uploads per day [7].

Two plausible scenarios could account for the more read-intensive property of Youtube in comparison with blogs. First, Youtube is one of the most well-known Web 2.0 sites in the world, and thus attracts a great number of Internet users. Second, although Web 2.0 sites provide handy tools for publishing content, more technical skill is necessary to make interesting videos than merely writing text and posting it with files such as images and music. At a campus-level edge network, only 133 attempts to upload videos were observed in comparison with 625,539 video requests over a three-month period [5].

3.2 Effect of Proxy Caching

We next look into the response codes in the Web access logs in order to shed light on the types of blog content that gain advantage from the Web proxies. Laying aside the codes that take negligible fractions, responses to clients' requests are generally categorized into two types: *Not Modified*, where the requested file, which is already cached by proxies or Web browsers, is up-to-date; and *Successful*, where the requested file is directly transferred from Web servers. The remainder, except for the above two types, comprise abnormal cases such as errors and file redirections whose requests are not eligible to take advantage of proxy caching.

Our blog workload reveals approximately 42% of the total files requested from clients were cached by Web proxies, as shown in Table 3. This result is different from traditional Web workloads, where the cached requests do not exceed 29%, even in the largest workload [2][3][8]. Although the evolution of Web caching techniques could partially account for this observation, the existence of blog templates is the main factor. Once a template is chosen for a blog, every access to the articles of that blog will accompany the files in the template. These files therefore will be repeatedly requested and cached by proxies when a client views certain blogs using the same blog template; this phenomenon is the same as the case where users customize their own templates. Actually, 90% of *Not Modified* requests were for files used as the template, all of which are identified as static files through the present examination. The remaining 10% requests were also for static content included in the blog articles, while none were for dynamic content.

A statistical summary of *Successful* and *Not modified* logs is provided in the second and third column of Table 4, respectively[2]. The summary illustrates that *Successful* requests closely reflect the total bytes delivered from the servers, although quite a few requests are excluded. This is not surprising given that all content of the *Not Modified* is directly serviced by proxies and client browsers. Hereafter, the workload used in this report only refers to *Successful* log entries unless otherwise noted. These entries account for all the content successfully returned

[2]Some values are not available because the sizes of files are not described in `Not Modified` entries.

Table 4: Summary of workload statistics

Item	Total	Successful	Not modified	Static file (Successful)	Dyn
Total requests	96,168,202	50,118,065	40,134,803	36,138,738	
Requests per day	8,014,017	4,176,505	3,344,567	3,011,562	
Total bytes transferred(GB)	4732.6	4729.53	N/A	4452.53	
Bytes transferred per day(GB)	394.13	393.89	N/A	371.04	
Mean transfer size(KB)	N/A	98.95	N/A	128.74	
Standard dev. of transfer size(KB)	N/A	655.49	N/A	739.87	

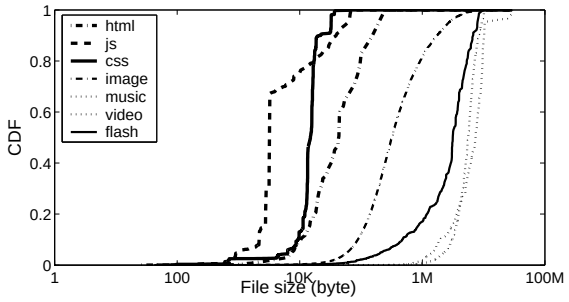


Figure 2: CDF of unique file sizes for each type; 'js' refers to java script file and 'css' refers to cascading style sheets.

to clients by the blog servers.

3.3 File Type Properties

In the reduced data set, there are many types of files of different sizes. The cumulative size distribution (CDF) and a statistical summary of unique files for music, videos, images, and other content types are presented in Fig. 2 and Table 5, respectively. The results show that images significantly dominate in number, while music and videos dominate in terms of size. The explosive use of image files is primarily due to the blog templates, which are mainly composed of one or two of language files (e.g. CSS and JavaScript) and dozens of small-sized image files. Furthermore, photographic images formatted as JPG and BMP are also widely used in the blog space, occupying more than half of all images. We additionally observed that many bloggers upload music content onto their blogs. However, video sharing does not seem to have matured yet.

As a result of the prevalence of massive content, we found that the average transfer size is about 100 KB, several times larger than that (5.7 KB $\sim$ 13 KB) documented in the past [2][3]. Ten years ago, Arlitt foresaw the future growth in the use of media files (i.e. music, high-quality images, and videos), which might eventually have a dramatic impact on Web server workloads [2]. We now see the initial impact of this: various media files that tend to be of greater size than other files affect the mean file size transferred over the network.

To examine in detail why the transfer size becomes larger and what is the effect of media files on this change, we separated the files according to their types. Fig. 3 shows three unique aspects of blogs relative to the known characteristics of Web services identified in previous works [2][9][3][8]:

1. HTML requests are drastically reduced from 10% $\sim$ 30% to 0.003% due to the prevalence of dynamic Web pages, which noticeably replace static HTML files.

Table 5: Statics of unique files for each type

	html	JavaScript	CSS	image	music	video	flash
Number of files	123	1,438	7,989	401,921	15,414	409	1,208
Average file size (KB)	18.42	3.042	7.020	132.8	4,427	4,163	603.2
Standard deviation (KB)	30.17	5.067	7.588	269.6	2,164	3,645	1,301

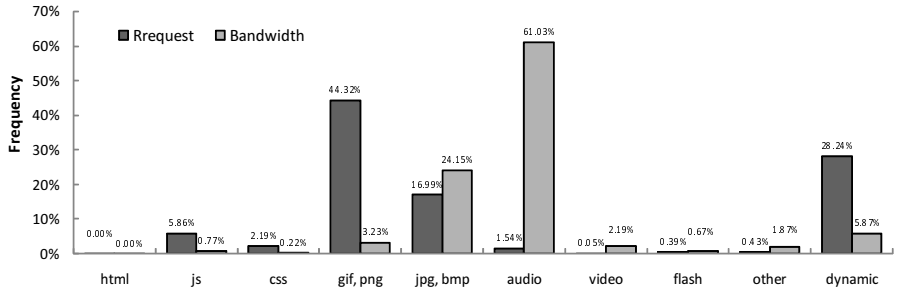


Figure 3: The number of requests and bandwidth used for each file type

2. Music files have large traffic volume in spite of a few requests. It is thought that musicblogs have made music files a main traffic source.

3. The bandwidth consumed by images is much smaller than that of traditional Web workloads, despite that the percentage of the number of requests is comparable.

One might expect the effect of images on the server network to be strong owing to the use of photoblogs. In most cases, photoblogs employ high-quality images, formatted as JPG and BMP rather than GIF and PNG. We therefore divided the images into two groups, as shown in the figure. Interestingly, JPGs and BMPs account for 17% of the requests and 24.2% of the network usage, thus comprising a considerable portion of the total usage. Although blogs are also used for uploading video content, the influence of this is still minor.

From the analysis of file types, we observed that media files play a dominant role in network usage in blog services. To sum up, the files consumed 87.4% of the entire network resources, although the number of requests for these files only occupies 18% of the total.

3.4 Comparison of Static and Dynamic Content

Processing requests for dynamic Web content often involves a substantial amount of time due to the overhead for invoking programs via server-side scripts. Therefore, some Web servers that service content extensively are suffering from high CPU load rather than network pressure [10][11][12]. In blogs, dynamic Web technologies are also widely used to create interactive and animated communities. To identify the probability of CPU being a bottleneck in blog servers, we investigate files in terms of static and dynamic content in this section.

First, however, it should be clarified whether the network used for static files is a bottleneck in our workload. *Successful* requests are divided into two groups, static and dynamic, and are summarized in the fourth and fifth column of Table 4, respectively. It can be readily inferred from Section 3.3 that the network usage is dramatically biased toward static files. The following describes in detail the differences between the two groups:

1. Approximately 94% of the total bandwidth, over 4.4 TB, is used for delivering static files.

2. The number of requests for static files is more than 2.6 times larger than the number of requests for dynamic files.

Table 6: Average processing time and standard deviation

Type	Average Time (us)	Standard Deviation (us)
Static files	624,635	10,738
Dynamic files	316,536	557

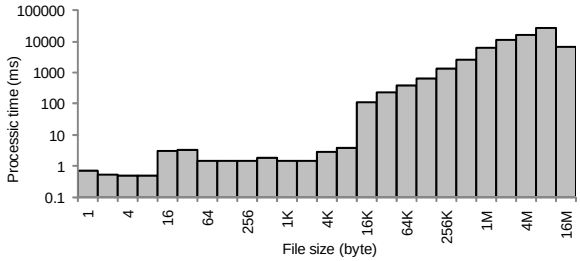


Figure 4: Average processing time of static files for different sizes. The horizontal-axis is a log-scale.

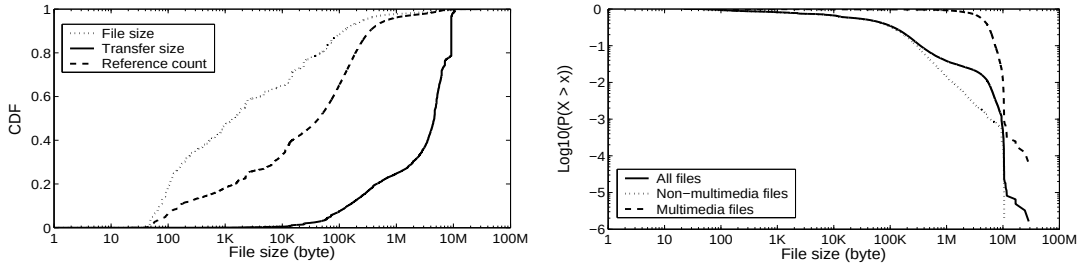
Clearly, most bandwidth is consumed by static files, even though the number of requests for dynamic files is somewhat comparable to that for static files. The reason for this sharp contrast in network usage is apparently related to the domination of media files with respect to bandwidth consumption.

We next analyzed the time spent to process incoming requests in order to assess which file group affects the execution of Web servers. Table 6 shows the average processing time and its standard deviation (SD) for the two types of requests. The processing time of a static file is 0.62 seconds whereas 0.31 seconds is required for a dynamic file, a surprising twofold difference between the execution of a script program and a file fetch. Furthermore, the sum of time required to treat all static file requests constitutes 82% of the total processing time of all requests. This result is contradictory to the trends observed for some other Web servers where lengthy processing of dynamic files has been identified as a target for improvement [13][11].

In order to determine why the processing time is also dominated by the static content in spite of active use of dynamic content, we plot the average processing time of static content with respect to different range of size in Fig. 4. The time is bounded below 10 ms for files smaller than 16 KB, but increases proportionally as the size increases. If it is supposed that that the growing popularity of Web 2.0 will induce up-to-date blogs to accommodate various media content, which are horizontally scaled up in size, then, the processing of large static files could eventually place a serious burden on blog servers.

4 Static Files Analysis

In this section, we focus on a detailed characterization of the static content along with an approximation of characteristics using well-known distribution models. Specifically, the following were investigated: file sizes, transfer sizes, and file popularity. By attaining a deep understanding of the content, we can emphasize how media content affects the use of storage and network, and thus how blog servers could be improved. For this purpose, the requests for static files were extracted as traces to be used in this section.



(a) (b)

Figure 5: Distributions of static files. (a) CDFs for file size, reference count, and transfer size, (b) LLCDs for file size of all files, multimedia files, and non-multimedia files

4.1 File Size and Transfer Size Distributions

Understanding the nature of both stored and transferred files is advantageous, for instance, to manage the resources in storage systems efficiently. Supposing that a media file, which is usually huge, is referenced multiple times, the impact of the file on the network usage would then be much greater than the impact of small-sized files. In this sense, a detailed analysis of the sizes of distinct files that are both stored and transferred was carried out with a focus on the effects of large media files.

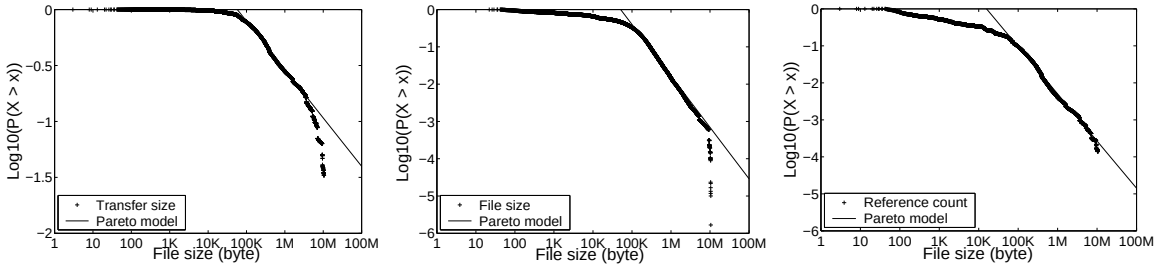
Fig. 5(a) shows the CDF of file sizes, file references, and transfer sizes. At first glance, the transfer size is highly skewed toward the large files, thus indicating that a small number of large files stored on the disk substantially affect the server network usage. We observed that files larger than 400 KB account for 80% of the total bytes transferred, but only 7.45% of the total bytes stored. Among the stored files exceeding 400 KB in size, the most two prevalent types are high quality image files such as jpg and bmp (59%), and music files (33%). The remainder consists of videos, pdf files, compressed files, and document files. Among the transferred files over 400 KB, by contrast, a wide disparity exists between music files (80.7%) and image files (11.1%) due to the fairly larger size of music files relative to image files. This provides evidence that media content posted by bloggers causes a massive volume of traffic on the network.

The high skewness toward such large files in the distribution of transfer sizes is a unique characteristic of Web 2.0 service workloads. For instance, a concentration of files above 1 MB in size was observed at campus-level traffic for YouTube service [5] while previous Web workloads have few or no references for files over 1 MB [2][9][8]. This is an important issue because when a large file is cached, cache systems should sacrifice more requests on the files evicted by accepting the large file. For these systems, therefore, it is important to carefully decide which strategy to utilize between increasing the hit ratio of requests and reducing the server's network overhead. However, the popularity of large files should be considered together because caching unpopular files is wasteful.

In a more rigorous study, we next showed that the plots in Fig. 5(a) are heavy-tailed. Heavy-tailed behavior means that, regardless of the distribution of small values, relatively few large values occupy a significant portion of the resources such as storage or network bandwidth.

We selected the Pareto distribution [14] to observe the heavy-tailed characteristic of our CDFs. To determine whether the CDFs fit the Pareto distribution well, we adopted a method to transform the CDF to a log-log complementary distribution (LLCD)[3]. Afterwards, if a linear slope is shown in the LLCD, the measured distribution might be modeled with a Pareto distri-

[3]LLCD is transformed from CDF by plotting $log_{10}(1 - F(x))$ on vertical-axis and $log_{10}x$ on horizontal-axis, where x is in heavy-tailed range and $F(x)$ is cumulative value of x.



(a) (b) (c)

Figure 6: LLCDs for non-multimedia files: (a) transfer size, (b) file size, (c) reference count

Table 7: Approximation results for transfer size, file size, and reference count

	Transfer size	File size	reference count
tail index(α)	0.43	1.37	1.27
goodness-of-fit(R^2)	0.98	0.99	0.99

bution with high certainty; the degree of heavy tail (i.e. the tail index (α)[4]) and the confidence level (i.e. goodness-of-fit(R^2)[5]) should be accompanied to ensure reliable modeling.

A distinctive characteristic of our workload is that the CDFs in Fig. 5(a) follow heavy-tailed behavior only for non-multimedia files (i.e. files excluding music and videos). At first, we failed to approximate the three original CDFs into a Pareto distribution. Interestingly, the reason for the failure is that the existence of large multimedia files affects the non-multimedia files, which are originally heavy-tailed. For instance, the file size distribution in Fig. 5(b) illustrates that non-multimedia files yield a heavy-tail shape plot with an exact linear slope, whereas multimedia files shift the shape.

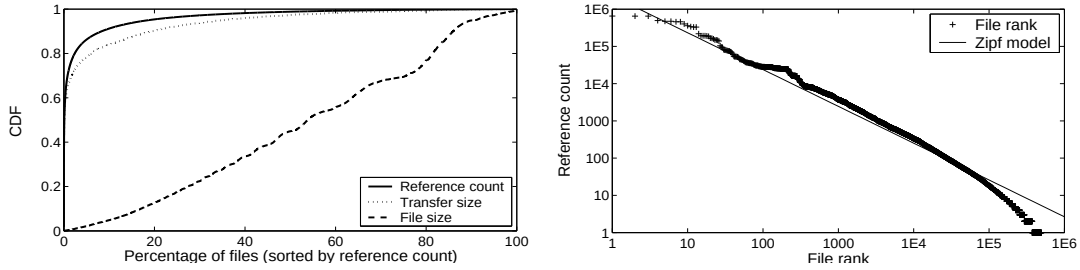
Fig. 6 and Table 7 provide the LLCD and the tail index with the confidence level, respectively, for three CDFs without multimedia files. The CDFs are modeled well with a truncated Pareto distribution [15], where there is a natural upper bound that truncates the tail. The tail length of the LLCDs ranges from an order of magnitude for transfer size to about 3 orders of magnitudes for file size and reference count[6]. Although the three CDFs strongly fit the Pareto model according to the goodness-of-fit test, each tail index presents a different value. As compared with traditional Web workloads that have ($0.93 \leq \alpha \leq 1.33$) for file size and ($0.71 \leq \alpha \leq 1.42$) for transfer size [2][9], the present results show the tail index of file size is comparable whereas the transfer size is more heavy-tailed. This is reasonable given that we have seen the incredible impact of large-sized files on the network usage.

Remember that heavy-tailed properties of both transferred and stored files would cause the system to erroneously fall into resource misallocation due to unexpected large file size. In particular, the bandwidth consumption of large files is much harder to predict and bound in blog servers. The aspect should be considered by system designers of blog services.

[4]If $0 < \alpha < 2$, then the distribution has infinite variance, whereas if $0 < \alpha \leq 1$, then the distribution has infinite mean.

[5]The approximation process becomes more precise as the goodness-of-fit approaches to 1.

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



(a) (b)

Figure 7: Concentration of references for ranked files: (a) CDF of file size, reference count, and transfer size, (b) Zipf analysis (reference count versus rank)

4.2 Popularity Properties

Treating popular files preferentially is another useful strategy for efficient resource management in Web server systems. The presence of strong file concentration in the server workload enables server caching strategies to be more predictable and controllable. Furthermore, load-balancing in clustered Web servers could be more powerful by taking into account the request and network throughput caused by frequently accessed files. In this sense, a detailed analysis is presented with the reference frequency and transferred data of distinct files, with a focus on media content.

Our first analysis aimed at measuring the relationship between popular files and their actual resource usage for the network and the storage. For this purpose, files are sorted by their reference counts and the most popular file assumes the highest ranking. The disk space usage, network usage, and reference count are then accumulated into a CDF graph, as shown in Fig. 7(a).

The resulting graph reveals 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. This rate is similar to past works where popular files have also shown a high degree of concentration [16][2][8].

The most remarkable observation obtained through this study is that a small working set of popular files contributes to an enormous volume of bytes. We observed that 10% of the most highly ranked files take up to 6.1 GB disk space and 3,737 GB traffic amount. The working set size of the files is not severely large, yet these files dominate the traffic volume used for the blog service. Furthermore, a surprisingly high portion of both resources was used by media files, with 6.0 GB of storage and 3,648 GB of bandwidth. Only two-fifths of the working set, however, was observed to be media files. This demonstrates the dominant influence of media files on the traffic during the action of blog services. One might wonder why large as well as popular media files influence the volume of traffic greatly. The answer is that the two groups are correlated, in that 70% total bandwidth is simultaneously used for both.

A typical method to model the popularity distribution is using Zipf's law [17]. According to Zipf's law, the number of file references shows a consistent decline as the file rank decreases. Letting (`R`) represent the rank of a file, then the number of references (`P`) is derived from the following formula:

$$P \backsim R^{-\beta}$$

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 drawn by placing log-scaled `R` on the horizontal-axis and log-scaled `P` on the vertical-axis.

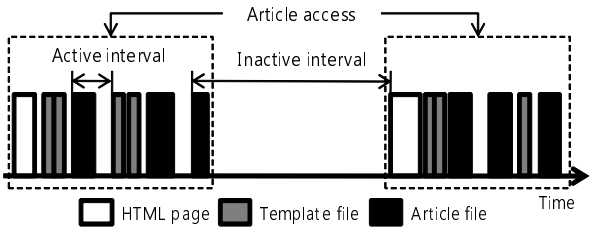


Figure 8: Access pattern when a client reads several articles in a blog

Table 8: Summary of interval samples (three major bloghosts)

	Short interval		Long interval	
	Number of samples	Average time (SD)	Number of samples	Average time (SD)
Blog 1	248,815	0.55s (0.99)	11,452	5,803s (40,266)
Blog 2	688,621	0.15s (0.49)	21,342	5,870s (41,863)
Blog 3	64,754	0.18s (0.51)	1,187	21,875s (85,585)

Fig. 7(b) shows that Zipf's law approximates the popularity of our traces well, with β close to 0.99 and 0.98 of correctness by the R^2 goodness-of-fit test. Similar parameters resulting from analyses based on Zipf's model have been suggested in other Web workloads [2][16].

We lastly analyzed the occurrence of one-time file referencing. The percentage of number of files and the sum of the sizes stored at disk are 38.5% and 38.3%, respectively, whereas their portions of network usage are only 0.7% and 1.2%, respectively. We noted that the results of our traces are not relevant to former researches because all Web workloads have different one-time referencing properties [2][9][8].

5 Articles Analysis

An article is a basic communication unit among bloggers. In blogs, articles are produced by users all the time, and these articles are directed to other users in many useful ways such as Web feeds, comments, trackbacks, and other recommendation websites. Therefore, analyzing our blog workload based on the unit of an article provides more precise insight into the behavior of bloggers. In this section, we present a methodology to organize article access logs from raw Web access logs used in previous sections. Using the article access logs, we describe several characteristics of blogs.

5.1 Methodology Overview

It is essential to extract article access data from the raw access logs because no information is provided about articles. We exploited two phases to obtain the article access data. The first phase is to use request intervals. When a client visit a blog, she or he usually reads one article and, after a while, another article. This sequence shows a unique request pattern, as shown in Fig. 8. For a clear explanation, we define two terms: (1) *active interval*, the time elapsed between two consecutive file requests within an article; and (2) *inactive interval*, the time elapsed between two consecutive file requests for different articles. We measured average active intervals and average inactive intervals from requests for three popular blogs and present the results in Table 8. The difference between the two kinds of intervals is so marked that we

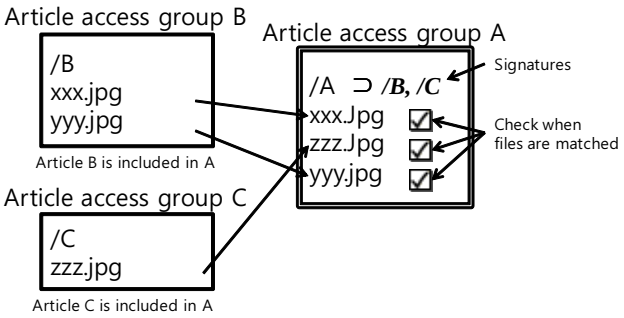


Figure 9: Procedure of file matching when an article access group A contains accesses to unique articles, B and C

Table 9: Summary of article statistics

	Number of articles	Number of article files	Number of media files
Unique articles	87,332	235,586	233,262
Unique article accesses	1,088,210	2,909,894	2,737,666

can easily distinguish one article access from another. Using these two intervals, we grouped the raw access logs into article access groups.

The following is the detailed procedures for extracting the article access groups:

1) Select a collection of file access logs, corresponding to a unique pair of a blog and a client.

2) Gather requests for a HTML file and all following non-HTML files until another HTML is requested or a non-HTML request is submitted over a time threshold. We choose the threshold to be 5 seconds on the basis of inspecting *inactive interval* data.

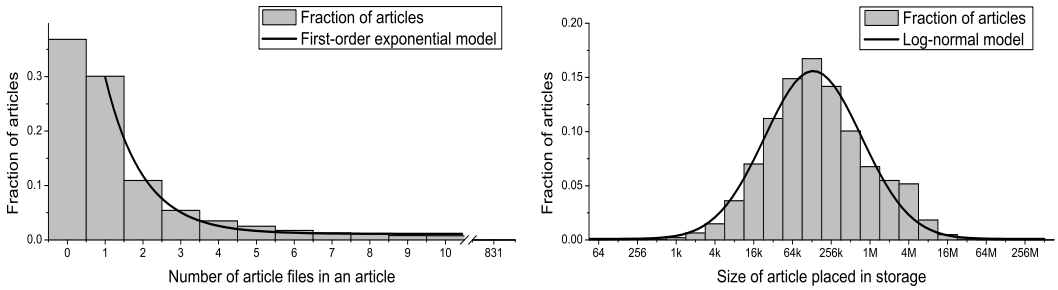
3) Given the requests from step 2), use the requests for a blog article as an article access group. Afterwards, go to step 2) for picking up requests for another blog article.

4) Restart from step 1) until all file access logs are checked.

The article access groups acquired from the first phase include redundant article accesses. Some article access groups include several articles together because some pages of a blog may contain several articles. For instance, the front page of a blog usually has one or more articles to encourage visitors to read recent articles. With the raw access logs for this kind of blog page, 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 Fig. 9. Suppose that we have three article access groups: A, B, and C. Each article access group usually has article files other than article text. 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 groups, B and C. Using this method, we refined article access groups from the first phase and obtained article access logs.

Through these phases, we found a total of 87,332 unique articles stored in the blog servers and 1,088,210 accesses for such articles during the trace period. Table 9 presents an overall summary of the articles, where we can estimate that one article is read about 12.5 times on average. One notable observation is that media files occupy most of the article files in our



(a) (b)

Figure 10: Histograms of unique articles: (a) histogram of unique articles by the number of article files, (b) histogram of unique articles by size

workload. It is thus expected that article characteristics perceived in this report would be helpful to better understand the current Web trend where media files are highly exploited in large-scale blog services.

5.2 Characteristics of Blog Articles

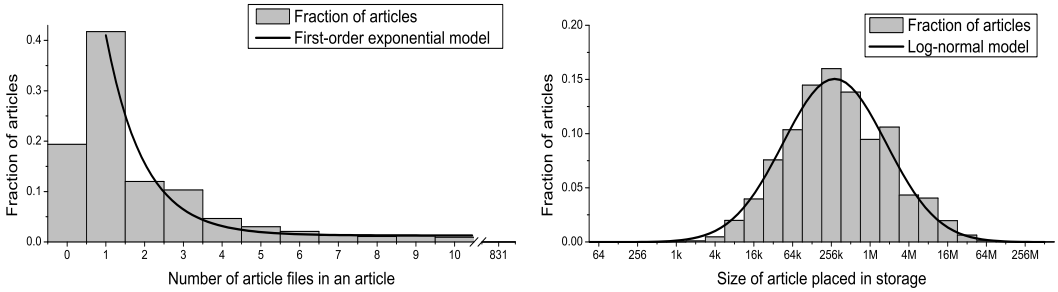
Characterization of blog articles are first discussed in terms of sizes of articles produced by writers and how frequently those articles were read by visitors. We further analyzed the inter-reference times for articles to investigate whether the pattern of subsequent accesses shows a temporal locality. The results of these investigations provide new insights into blog users' behavior and suggest useful implications for capacity planning and load balancing in a clustered server environment.

5.2.1 Unique Article

We eliminated a few unique articles in the subsequent analyses due to being of unknown size. According to our observations, the effect of these articles is very small, accounting for only 3% of the total unique articles, to which accesses account for 0.5% of the total. Consequently, we used 84,776 unique articles of which the size is recognized as a valid trace set hereafter.

Fig. 10(a) shows a histogram of unique articles according to the number of article files. The article contains 2.7 files in an average case, and the median and SD are 1 and 8.9, respectively. The number of articles having less than 4 files occupies 87% of the total, indicating that bloggers prefer to publish their own articles with a small number of files. We can approximate the histogram in Fig. 10(a) with a first order exponential distribution ($0.01 + 0.77e^{-x/1.0}$) except for zero-file articles. The R^2 goodness-of-fit value of 0.99 shows a strong fit to the distribution. This approximation suggests that articles exist in blogs with an exponentially decreasing rate as the number of article files increases.

Fig. 10(b) shows a histogram of unique articles by size, where we can determine the sizes of articles placed in storage. The average size is 1.1 MB with 4.6 MB of SD and the median size is 203 KB. Articles are mostly located within a range of 32 KB to 2 MB, accounting for approximately 74% of the total. We approximated the histogram with a log-normal distribution [18] ($\mu = 12.3, \sigma = 1.80$, natural log), as shown in Fig. 10(b). It shows a strong R^2 goodness-of-fit value of 0.98.



(a) (b)

Figure 11: Histograms of accesses to unique articles: (a) histogram of accessed unique articles by the number of article files, (b) histogram of accessed unique articles by size

5.2.2 Unique Article Access

User accesses to unique articles exhibit a strong tendency that all of the files or none of the files in an article are transmitted from blog servers to clients. To verify this, we analyzed how many articles are supplied by the server with partial delivery of the included files. Among 433,443 transfers for articles having more than two article files, 13,846 in number, approximately 3.2% of the total, are observed to correspond to this case. The remainder is constituted of 317,554 and 102,043 accesses to articles with all of the files and none of the files, respectively.

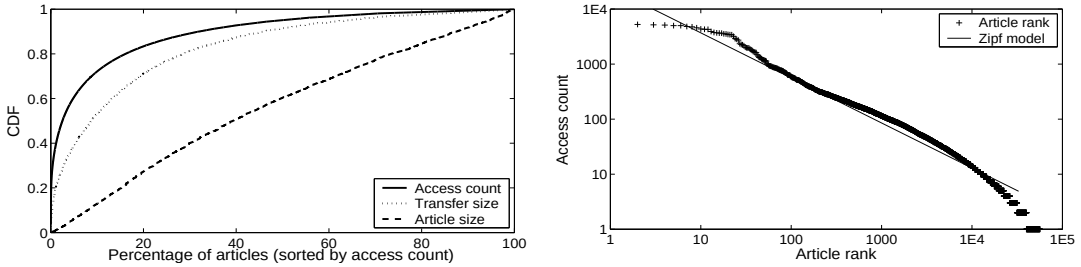
Fig. 11(a) shows a histogram of accessed articles according to the number of files. 2.7 files on average are included in the article, of which the median and SD are 1 and 8.8, respectively. The number of articles that are composed of less than 4 files accounts for 86% of the total. When the percentage of access for zero-file articles is compared with that of unique articles, it is found that blog visitors prefer to read articles posted along with article files such as image and music files. The histogram in Fig. 11(a) was also approximated with a first order exponential distribution ($0.01 + 1.10e^{-x/0.98}$) except for zero-file article accesses. The R^2 goodness-of-fit value of 0.97 shows a strong fit to the distribution.

Fig. 11(b) shows a histogram of articles, which are retrieved from storage, by size. The average size of the articles is 1.07 MB, and the median and SD is 202 KB and 3.46 MB, respectively. Articles are aggregated within a range of 32 KB to 2 MB, occupying approximately 74% of the total. By comparing these values with those of unique articles, we observe that accesses to the articles in storage are fairly evenly distributed. We approximated the histogram with a log-normal distribution ($\mu = 12.3, \sigma = 1.80$, natural log), as shown in Fig. 11(b). It shows a strong R^2 goodness-of-fit of 0.98.

Both log-normal and exponential distributions illustrate the behavior of blog users who were publishing and consuming articles. In particular, the log-normal distribution is common when the average is low and the variance is large. Therefore, the articles viewed from the two user actions (i.e. writing and reading) are biased toward small size.

5.2.3 Accessing Behavior

User references tend to be more distributed over blog articles than in static files that have already been analyzed. To verify this, we conducted a rank-based analysis as done in Section 4.2. Articles are likewise sorted by their access counts, with the most popular assuming the highest rank. Subsequently, the storage usage, network bandwidth usage, and access count are put into CDF graphs, as shown in Fig 12(a). The graphs show that 10% of most frequently visited articles were responsible for 73% of total counts and 53% of bytes used to transfer total



(a) (b)

Figure 12: Concentration of references for ranked articles: (a) CDF of article size, access count, and transfer size, (b) Zipf analysis (access count versus rank)

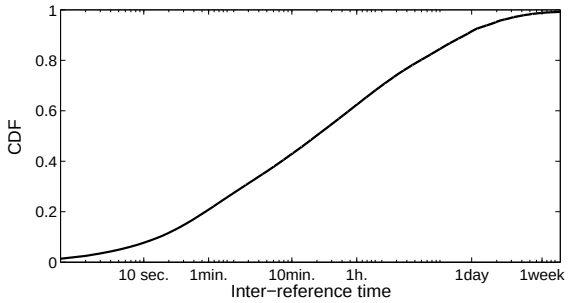
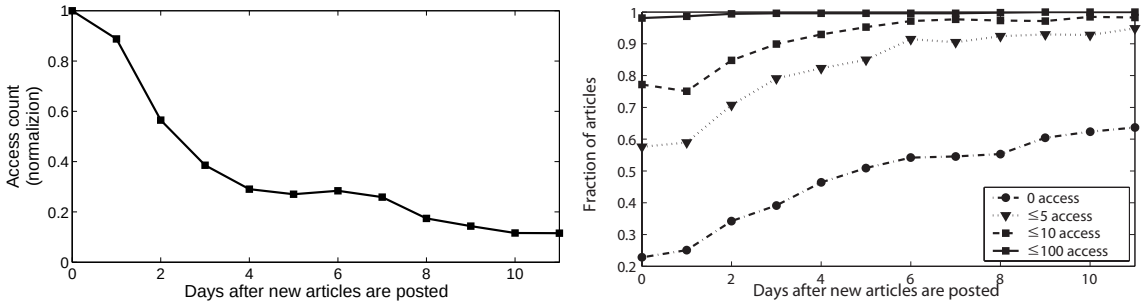


Figure 13: CDF of inter-reference time that means the time for an article to be re-referenced

articles. This rate is much smaller than the rate observed in our analysis of file popularity, which shows 90% of requests and 84 % of bytes were concentrated on the top 10% of most 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 value is 0.82 with a 0.98 of R^2 goodness-of-fit value, as plotted in Fig. 12(b).

In general, popular content in Web 2.0 services, such as articles of blogs or video clips of video sharing sites, appear to have lower access rates than content favored before the Web 2.0 age. A recent study on Youtube traffic illustrated that incoming requests were more widely scattered among videos; the top 10% of videos only accounted for 40% of requests [4]. For most traditional Web workloads, however, the top 10% of most requested files account for more than 80% of the total. We speculate that content published in user communities accompanies the following two trends: massive production scale and various methods to support user interaction.

The access patterns of articles exhibit weak temporal locality in our blog service. Temporal locality refers to the notion that successive references to the same article arise within a short time. Fig 13 shows a typical plot of the cumulative distribution versus the interval (i.e. time taken for an article to be re-referenced). In this plot, the articles that were re-visited within 30 minutes and 1 hour account for 31% and 62% of the total, respectively. A noticeable trend in the graph is a monotonous increase as interval is shifted toward higher values, and thus no hot-spot was detected for the re-referencing pattern. One might expect that temporal locality would be shown in article accesses due to the 10% of top popular articles taking up 73% of total access counts. We project that this decayed locality may be heavily affected by the aforementioned two trends of communities, where article references are distributed among various articles.



(a) (b)

Figure 14: Popularity evolution of newly posted articles during 12 days: (a) change of access counts for new articles, (b) change of the fraction of articles accessed for certain amount, '$\leq$ 100' refers to the amount of articles read less or equal than 100 times on a particular day.

5.3 Evolution of Blog Articles

We continue our discussion on article popularity focusing on how the popularity of newly posted articles evolves over time and how fast or slow it changes. For this study, we used articles published on the first day of the trace period. In total, 526 fresh articles, which comprise 2,565 files and 459 MB size, are identified in our article access logs. Note that the majority of the article files were for media content comprising 2,428 files and 445 MB.

5.3.1 Evolution of New Articles

We first analyzed how access counts for new articles change over a range of 12 days. The results are shown in Fig. 14(a) with data normalized to the amount collected on the first day. During the first three days, the access counts for the articles decrease sharply to 35.8% and fall gradually thereafter. This evolution suggests ephemeral popularity of newly born articles.

Although the frequency of overall article reads falls as time passes, newly posted articles have different popularity in nature. To investigate this detail, we examine several viewpoints by considering a range of access counts from 0 to 100. Fig. 14(b) shows four plots of different access counts, 0, $\leq$ 5, $\leq$ 10, $\leq$ 100, for new articles being viewed during a 12 day period. One noticeable trend in the plots is a consistent rise of portion that each plot takes over time. On the first day, 42% and 23% of articles are viewed over 5 and 10 times, respectively, but the percentages decrease to 5% and 2% after 11 days. Furthermore, only 36% of the articles posted by bloggers are visited on the last day of the trace period. In short, a majority of new articles do not easily become a center of strong interest in blog communities.

However, some of these new articles seem to gather visitors steadily, although many are not requested relatively soon after. To better understand the property of popular articles, in the following we discuss the evolution of these articles focusing on the change of reference rates and inter-reference time. As a sample set, we selected 19 articles that were viewed at least 100 times during the first two days.

5.3.2 Evolution of Famous Articles

For the given 19 articles, we analyzed how maximum, minimum, and average access counts change over time. The plots in Fig. 15 show that on the first day the articles were read 460 and 103 times in the maximum and average cases, respectively. However, the average access count decreased as the days passed, and were constant at a level under 50 after 2 days. This

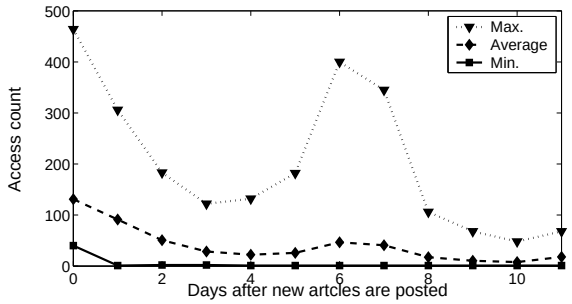


Figure 15: Change of maximum, minimum, and average access counts of famous articles during 12 days

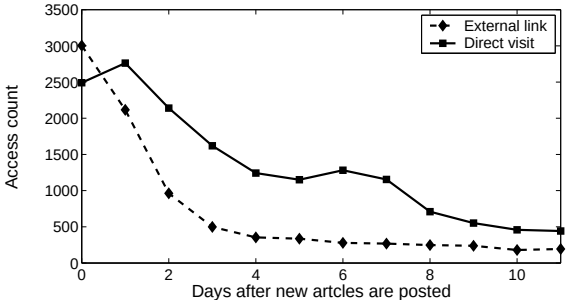


Figure 16: Comparison of access counts classified by external link and direct visit

demonstrates that even if an article did receive a high number of reads during its first two days, it is unlikely that the article will attract many reads in the near future.

The gap between the maximum and average plots reflects that only a very small number of fresh articles attracted visitors' interest while the remainder were more ignored. The maximum access counts were steady more than 50 for 12 days, but suddenly reached a peak between 6 and 7 days after initial publishing. Based on these findings, it is determined that the maximum plot is mainly influenced by a particular article, of which the owner posts a series of interesting articles that draws the reader back weekly. Obviously, identifying such owners and favoring them would be of benefit in managing a blog service.

5.3.3 The Effect of Incoming Path

Understanding what directs visitors to a fresh article and how fast or slow their influence on access counts changes provides useful information on the evolution behavior. We classified various ways of reading an article into two representative approaches: *external links*, where clients visit blogs through various online interactions such as RSS, trackbacks, and recommendation sites; and *direct visits*, where clients type the address of blogs on the Web browser to view articles directly.

The use of *external links* faded quickly, whereas the use of *direct visits* declined at a slightly lower pace, as shown in Fig. 16. On the first day, the frequency of *external links* was about 1000 times more than that of *direct visits*, but fell steeply to 499 after three days. This indicates that once an article is posted, this information propagates quickly via *external links*. By contrast, the number of new articles read by *direct visits* decreases from 2,762 on day one to 1,150 on day five. In summary, the way of visiting to fresh articles through *external links*, which are devised to promote user interactions, is more ephemeral than the way of viewing articles by *direct visits*.

6 Related Work

Recently, many researches for social network communities have targeted on blogs to understand user behavior. Nakajima *et al.* characterized bloggers based on their roles, such as an agitator or a summarizer, using blog thread [19]. A blog thread is a set of blog articles comprising a conversation on a specific topic. Baumer *et al.* focus on readers' behavior [20]. They gathered data from employed participants and suggested a qualitative study of blog readers, including common blog reading practices and relationships between identity presentation and perception. Adar *et al.* analyzed various kinds of Web content including blog articles to understand past behaviors and to predict future behaviors [21]. These studies show users' behavior using context or link of blogs. However, we utilized Web access logs of blog servers and analyzed comprehensive characteristics of both content producers and consumers viewed from file-level to article-level.

Instead of blogs, there have been several studies to analyze YouTube service. Gill *et al.* characterized the YouTube mainly focusing on the traffic monitored on a local campus network [5]. They presented an extensive analysis results of the Youtube workload and found that there were many similarities to traditional Web and media streaming workloads. Cha *et al.* crawled Youtube site to examine especially the popularity distribution, popularity evolution, and content duplication of user-generated video contents [4]. Both works were the first attempt to investigate the content of Web 2.0 services, which is useful for understanding rapid increase in the use of user-generated content.

There have also been numerous studies of traditional Web server workloads. Arlitt and Williamson introduced useful invariants to explain the nature of Web workloads [2]. Pitkow presented a good summary of Web traffic characteristics [22] and Crovella *et al.* presented the evidence why a number of file size distributions in the Web exhibit heavy-tailed [23]. In 2000, a workload characterization is conducted using the access logs of 1998 World Cup Web site [24]. To obtain characteristics of a busy Internet Web server, Oke and Bunt analyzed access logs of a busy commercial Web server [9]. They took a concept of Web server access hierarchy, which views the workload from three different levels: multiple clients, individual clients, and sessions of individual clients. Bent *et al.* presented a study of the properties of a large number of Web sites hosted by a major ISP [3]. Most recently, Web server workload invariants were revisited using scientific Web sites with suggesting new three invariants [8].

7 Conclusion

We have presented an extensive analysis on the user activities, user-generated content distributions, and article distributions and evolution using Web access logs collected from a blog service. To the best of our knowledge, this is the first empirical study that demonstrates the characteristics of various content, such as media content, and articles in blogs.

We found that user activity to produce content is greater than traditional Web services and Youtube. The effects of media content on the server network is huge enough to be a main source of bandwidth consumption despite the apparently small working set. This result indicates that caching media content in a reverse-mode proxy server would be beneficial to reduce the burden of processing incoming requests in blog servers.

After an extensive analysis of blog articles, we identified that visitors prefer to read articles posted along with article files such as image and music files. Especially, reading articles exhibit a unique strong behavior that all of the files or none of the files in an article are transferred from blog servers. This behavior is a useful guideline for designing an efficient server file system, which makes article files in an article as the unit of disk access.

References

[1] *Technorati. http://www.technorati.com.*

[2] M. F. Arlitt and C. L. Williamson, "Internet web servers: workload characterization and performance implications," *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 631–645, 1997.

[3] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao, "Characterization of a large web site population with implications for content delivery," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*, 2004, pp. 522–533.

[4] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.

[5] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 15–28.

[6] *Tistory. http://www.tistory.com.*

[7] *Business Intelligent Lowdown. Top 10 Largest Databases in the World.*

[8] A. M. Faber, M. Gupta, and C. H. Viecco, "Revisiting web server workload invariants in the context of scientific web sites," in *SC '06: Proceedings of the 2006 ACM/IEEE Supercomputing Conference*, 2006.

[9] A. Oke and R. B. Bunt, "Hierarchical workload characterization for a busy web server," in *TOOLS '02: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools*, 2002, pp. 309–328.

[10] J. Challenger, "A distributed web server and its performance analysis on multiple platforms," in *ICDCS '96: Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS '96).* IEEE Computer Society, 1996, p. 665.

[11] V. Holmedahl, B. Smith, and T. Yang, "Cooperative caching of dynamic content on a distributed web server," in *HPDC '98: Proceedings of the 7th IEEE International Symposium on High Performance Distributed Computing.* IEEE Computer Society, 1998, p. 243.

[12] A. Dingle, E. MacNair, and T. Nguyen, "An analysis of web server performance," in *GLOBECOM 99: Proceedings of the IEEE Global Telecommunications Conference*, 1999.

[13] A. Iyengar and J. Challenger, "Improving web server performance by caching dynamic data," in *USITS '97: Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems.* USENIX Association, 1997, pp. 5–5.

[14] V. Paxson and S. Floyd, "Wide-area traffic: the failure of poisson modeling," in *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, 1994, pp. 257–268.

[15] I. B. Aban, M. M. Meerschaert, and A. K. Panorska, "Parameter estimation for the truncated pareto distribution," *Journal of the American Statistical Association*, vol. 101, pp. 270–277, 2006.

[16] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *SIGMETRICS '98/PERFORMANCE '98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, 1998, pp. 151–160.

[17] G. K. Zipf, *Human Behavior and the Principle of Least-Effort.* Addison-Wesley, 1949.

[18] E. Limpert, W. A. Stahel, and M. Abbt, *Log-normal Distributions across the Sciences: Keys and Clues.* BioScience, 2001, vol. 24, no. 4.

[19] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and K. Tanaka, "Discovering important bloggers based on analyzing blog threads," *Proceedings of WWW 2005 2nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2005.

[20] E. Baumer, M. Sueyoshi, and B. Tomlinson, "Exploring the role of the reader in the activity of blogging," *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp. 1111–1120, 2008.

[21] E. Adar, D. Weld, B. Bershad, and S. Gribble, "Why we search: visualizing and predicting user behavior," *Proceedings of the 16th international conference on World Wide Web*, pp. 161–170, 2007.

[22] J. E. Pitkow, "Summary of www characterizations," *World Wide Web*, vol. 2, no. 1-2, pp. 3–13, 1999.

[23] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the world wide web," pp. 3–25, 1998.

[24] M. F. Arlitt and T. Jin, "A workload characterization study of the 1998 world cup web site," *IEEE Network*, vol. 14, no. 3, pp. 33–37, 2000.