A Model for Web Workload Generation Based on Content Classification

CARLOS MARCELO PEDROSO
Federal University of Parana
Department of Electrical Engineering
Centro Politcnico, Curitiba PR
BRAZIL
pedroso@eletrica.ufpr.br

KEIKO VERONICA ONO FONSECA
Department of Electrical Engineering
University of Technology - Parana
Av. Sete de Setembro, Curitiba PR
BRAZIL
keiko@utfpr.edu.br

Abstract: Web server performance is tightly bound to the workload the server has to support. Therefore, understanding the nature of the server workload is particularly important in capacity planning and overload control of Web servers. Web performance analysis can be done, a priori, with a synthetic generation of Web system workload. However, performance analysis results depend on the accuracy of this workload. In this paper, we propose and describe a workload-generation model based on group classification of Web server files according to their contents. This model, henceforth referred to as SURGE-CC (SURGE Content Classification), is an extension of the SURGE (Scalable URL Reference Generator) model. SURGE-CC is simple, very easy to understand and, most important of all, can be readily customized for specific applications. The parameter settings in our model allows the influence of Web server contents on output load to be investigated from both a qualitative and quantitative point of view. The results of a workload-generation tool based on our model implementation show the workload dependence on the nature of the server contents, the model ability to generate self-similar traffic and the accuracy of the synthetic workload. The model was validated by a careful statistical analysis of massive data from several servers, computational simulations and by comparison of results found in literature. We point some future application of the SURGE-CC model and discuss the new investigation branches derived from the novelty of our model approach.

Key–Words: Performance, modeling, world wide web, workload generation.

1 Introduction

Synthetic Web traffic generators have been an essential tool in Internet traffic simulation for the last ten years and are likely to continue to be essential into the near future.

The great importance of the Web has increased the demands on researchers to develop better Web server models to mimic the Web behavior. A good model allows accurate forecasting of performance metrics, which in turn allows improved capacity planning and the development of new techniques for Web systems.

Among the desired characteristics of any model are simplicity and tractability. The simplicity can achieved by limiting the model variables to those that significantly affect the system under analysis, and its tractability is usually associated to the complexity in generating analyzable results from the model. Networking research experiments are typically based on simulation models. The correctness of the simulation results relies, among others, on the correctness of source traffic generation. The source traffic should be as near as the real users/application traffic as possible.

Web server capacity planning requires performance models that accurately capture real server behavior, as inaccurate models would lead to under-provisioning or over-provisioning of server resources. Much effort has been spent on the development of models that properly describe computer network traffic, as in [1], [2], [3] and [4]. The results of these studies indicate that the self-similar model is appropriate for describing the aggregate traffic observed at the outputs of Web

servers. A further important finding of these studies is that classical Markovian models are not suitable for representing the actual traffic characteristics [5].

A traffic model for Web servers may or may not make use of client behavioral or server contents.

In this paper, we propose a model for Web servers based on classification of the contents (files) transmitted by the server and also on user behavior. Our model exploits the time dependencies of requests in an user session, allowing a better workload generation if compared with other models available. When a Web user sends a request, this will cause a sequence of file transfers; our model is able to reproduce the time interval between file requests, as well as the appropriate file sizes. This was not explored by any model before. Besides, the proposed model employs a Markov Chain to represent the file transfer sequence, with good prospects for analytical developments. The novelty of the approach presented here can be summarized as follows: (a) a detailed view of the server contents results in better traffic generation; (b) our approach opens up new perspectives for performance analysis; and (c) new techniques for Web systems can be deployed and tested (for instance, cache server techniques, server content adaptation or clustering).

The model was validated by a careful statistical analysis of data from several servers whose public logs are available for use in research projects. We analyzed log files from the World Cup 1998 Web servers [6], the IRCache proxy servers; IRCache is an NLANR (National Laboratory for Applied Network Research) project [7], and a recent trace collected at the Pontifical Catholic University of Parana (PUCPR). The NLANR cache servers register access from the whole Internet and are geographically distributed throughout the U.S.A. to provide load balancing for all continents. Therefore, the millions of requests to the NLANR cache servers reflect typical Web-server access behavior, as the requests are not restricted to a specific group of clients or applications. All the Web access in the PUCPR network must be managed by a Proxy Server - we collect all Web Clients traffic for two days to study their behavior. At the time of data collection, the PUCPR had 30,000 students and an Internet link of 20Gbps.

The remainder of this paper is organized as follows: section 3 describes the model proposed in this paper, which is based on a semantic classification of the contents (files) transmitted by Web servers; section 4 describes a system characterization based on data collected from real systems; section 5 contains the results of a computational simulation of the SURGE-CC model; and section 6 presents the discussion of main results. Conclusions and possible future studies are presented in section 7.

2 Models for Web Workload Generation

The SURGE (*Scalable URL Reference Generator*) model proposed by Barford and Crovella [8] captures the user behavior and server characteristics to mimic real Web traffic. SURGE is based on an ON-OFF automaton that captures user behavior and employs probability distributions to model the size of transmitted files. When the system is in the ON state, the session is active, transmitting a series of files. The time interval between transmission of these files is called *active off* time. Figure 1 shows the time-dependent variables used in the model. The SURGE main variables are:

- *OFF Time:* Refers to the user reading time.
- *Request sizes:* The set of transferred files. This set differs from the set of file sizes because one file can be transferred multiple times or not at all. Commonly modeled by the Pareto probability distribution;
- *Embedded references:* The number of files transmitted in a user session. This is also modeled by the Pareto distribution;
- *Popularity*: The number of times an individual file is accessed. File popularity in a Web server follows Zipf's law.
- *Active off time:* The time between two successive requests for files in a user session. Commonly modeled by the Pareto distribution;
- *Temporal locality:* The temporal locality indicates the increase in the probability of a file being accessed again if it was recently accessed.

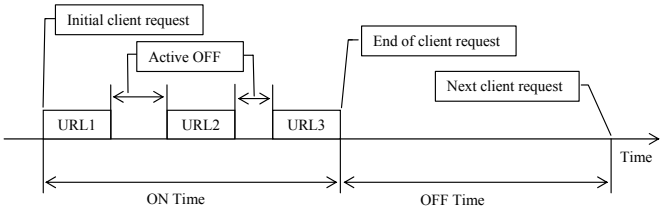

Figure 1: SURGE model

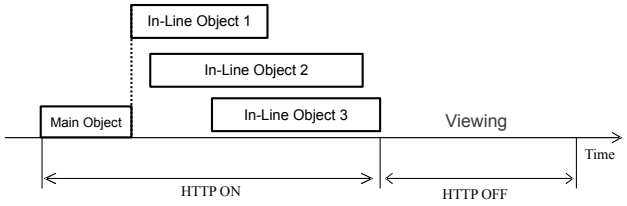

Figure 2: Model proposed by Choi and Limb [9]

Previous researchers have reported that the size of the files transmitted by Web servers normally fits a heavy-tailed distribution, such as the Pareto distribution [1] [8].

Choi and Limb present in [9] a variant to SURGE model to allow the simultaneous file transfers. This model differs from SURGE in characterizing how the files are transfered in ON Time. Figure 2 represents the file transfers. The first file requested is called *Main Object*, followed by In-line objects. This model is more realistic to mimic the actual traffic because can represent the automatic file transfers requested by Web browsers, necessary assemble the requested web page. Today, this is the most widely used model for Web workload generation [10]. The viewing time is reported to be exponentially distributed with mean of 30 seconds.

Hashemian et al. [11] analyzed the performance of Workload generators for Web applications. Acording to authors, *reported response times could be grossly inaccurate, and that the generated workloads were less realistic than expected, causing server scalability to be incorrectly estimated*. The cause apparently was the Java implementation and the generation of inaccurate time dependencies - one session depend on the responses of earlier requests in the session. Furthermore, Jianliang et al. in [12] postulate that the most of invariants for workload generation identified from 1994 will continue to hold in the future, because they represent fundamental characteristics of how humans organize, store, and access information on the Web.

Tsompanidis et al. [13] use Markov chains to model the user behavior in good or bad network conditions. The model can be used for network planing and research applications.

More recently, Pries et al. presents in [10] the results of an observation of top one million visited Web pages. The authors examine the changes in size and number of objects and comparing the findings with well-known Web traffic models. As a main conclusion the paper indicates that there is a trend towards larger web pages with an increasing number of inline objects and that today's web pages are not loaded from a single web server but created gathering content from multiple servers. Table 1 presents the parameters of main models for Web workload generation.

3 A New Model for Web Server Workload Generation

In this paper, we propose that the files transmitted by Web servers be classified using the file types, so that file sizes can be modeled for individual file classes, and furthermore, a class probability transition matrix is modeled for the whole system.

The user activity in our model is also modeled by an ON-OFF automaton. The period that the user is active producing requests is defined as the ON period, hereafter called user session. The user session is also characterized by an ON-OFF automaton. In order to differentiate these states will called as HTTP-ON and HTTP-OFF. The HTTP-OFF is the user reading time. The HTTP-ON is initiated by an user action, and involves the transfer of objects required to assemble the requested web page. The HTTP-ON and HTTP-OFF states are similar to those proposed by [9], and illustrated in Fig. 2.

However, in our model the HTTP-ON state is addressed in greater detail than other models. The available models do not differentiate inline objects in HTTP-ON state, and use a single probability distribution to model the size of these objects. This may lead to an inaccuracy for the identification of probability distribution to model the

Table 1: Overview of main models for Web Workload Generation, partially taken from Pries et al. [10]

Source	Main object size	Inline object size	Number of inline objects	Reading time
Choi [9]	Lognormal mean=10kB sdev=25kB	Lognormal mean=7.7kB sdev 126kB	Gamma mean=5.5 sdev=11.4	Weibull mean=39.5s sdev=92.6s
Mah [14]	Pareto $\alpha = 0.85 - 0.97$	Pareto $\alpha = 1.12 - 1.39$	- mean=2.8-3.2	- mean=1000-1900s
Barford [8]	Pareto $\alpha = 1$ $k = 1000$	Pareto $\alpha = 1$ $k = 1000$	Pareto $\alpha = 2.43$ $\beta = 1$	Weibull $\alpha = 1.46$ $\beta = 0.38$
Lee [15]	Lognormal mean=11.9kB sdev=38kB	Lognormal mean=12.5kB sdev=116kB	Gamma mean=5.07	Lognormal mean=39.07s sdev=324.9s
Pries [10]	Weibull mean=31.5kB sdev=49.2kB	Lognormal mean=23.9kB sdev= 10.3kB	Exponential mean 31.93	Lognormal mean=39.7s sdev=324.9s

size of these objects, since the file sizes greatly differs in size, eg. the scripts tend to be much smaller than videos, resulting in a inadequate workload generation.

We propose to model the sequence of file types requested in the HTTP-ON state using an automaton. Figure 3 shows a hypothetical state diagram with three file classes: *hypertext*, *image* and *script*. Suppose the initial state to be *hypertext*. In this case, p_{11}, p_{12}, p_{13}, and p_{14} represent the probability of the next requested file class being the *hypertext*, *image* and *script*, respectively. The *end* state indicates the end of HTTP-ON state and starts a user reading time period.

The state transition diagram can be represented by a square matrix P with the transition probabilities as follows:

$$P = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0(n-1)} \\ p_{10} & p_{11} & \cdots & p_{1(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(n-2)0} & p_{(n-2)1} & \cdots & p_{(n-2)(n-1)} \\ 0 & 0 & \ldots & 1 \end{bmatrix} \quad (1)$$

where n represents the number of classes, including the end-of-session class (in the matrix, the class $n-1$ is the end of session class). For a given state k, $\sum_{j=0}^{n-1} P_{kj} = 1$. This state transition diagram represents the sequence of transmitted file

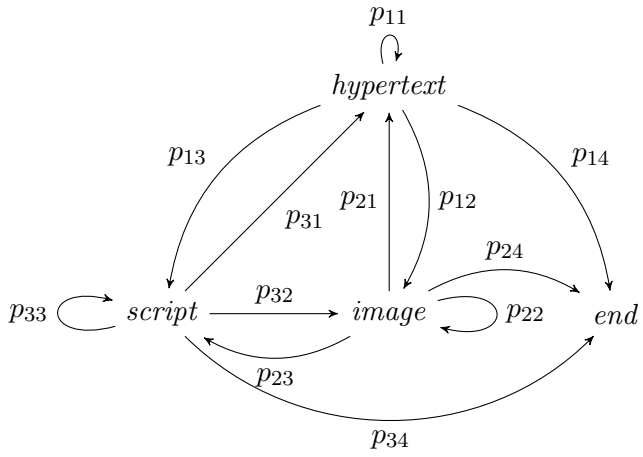

Figure 3: Hypothetical diagram of the file-class sequence during the ON state

classes. The probability distribution of the file sizes for each class can be found by analyzing the server log files.

4 Main File Types and Transition-Probability Matrix

As part of this study, several programs were developed to extract session information from server logs using heuristics similar to those originally developed by [16] and [17], where a session is identified as beginning with the first access by an IP address to the server and includes subsequent requests from this same address. The sequence of requests identified is inferred to reflect the behavior of the user session. If no user activity is detected during a certain amount of time, the session is considered to have finished. The probability of a file class be accessed during the session can be mapped to the transition-probability matrix. In this study, a threshold of 120 seconds was used, after which the client session was considered to have ended.

The results presented here are based on a careful analysis of three sets of data.

The first system analyzed was the World Cup 1998 Web server (WC98), which had already been studied by [6]. Although WC98 data is relatively old, the existence of previously published studies of this system makes comparison of our results with those of other studies possible. Besides, [12] states that since 1996 there was no significantly modification in the invariants to model the user behavior for Web servers. Three different days were studied, giving a total of approximately 20 million page requests. The log files that were analyzed are shown in Table 2.

The second system studied was the IRCache, an NLANR (National Laboratory for Applied Network Research) project [7]. The NLANR proxy servers register client requests from the whole Internet and are geographically distributed over the USA in order to provide load balancing for requests from all continents. Thus, the thousands of daily requests reflect the typical behavior of accesses to Web servers, as the requests are not restricted to a specific group of customers or to specific applications.

The third data set comes from the PUCPR (Pontifical Catholic University of Parana) network, were all Web access are intermediated by a proxy system, implemented by a cluster of servers. We were able to collect all Web traffic generated from users. At the time of measurements, the university has 30,000 thousands of students and staff. We analyzed data along five days, from Monday to Friday. Its is important to notice that the Proxy server of PUCPR network implements a black list to forbidden the access of inappropriate contents.

Thus, the WC98 is a stand alone Web system accessed by Internet clients, IRCache represent a pool of Internet Web servers accessed by many Internet Web clients and PUCPR represent a particular population of Web clients (of students and staff of university) accessing a pool of Internet Web servers.

Table 2: Web traces in study

Sample	Place	Date
WC98	Paris, France	May 31, 1998
WC98	Paris, France	June 15, 1998
WC98	Paris France	July 6, 1998
IRCache	New York, USA	November 28, 2004
IRCache	New York, USA	November 29, 2004
IRCache	Palo Alto, USA	November 29, 2004
PUCPR	Curitiba, Brazil	August, 31, 2012

4.1 Identification of the main file classes

The analysis revealed the main file classes for the Web systems under study. The semantic classes were identified mainly by observing the extension of the transmitted files. In order to identify how each transmitted class contributes to the total output traffic, the volume of traffic actually transmitted by each class of traffic was observed.

In the WC98 server, only a few classes contribute significantly to the output traffic, and this led us to build a simple model. The most important file types were ZIP, html, GIF and JPEG as shown in the Table 3.

We identified the following main classes in the IRCache system: html, OCTET-STREAM, html, PLAIN, GIF, XLM, MPEG, FLASH and PDF. Table 3 shows the contribution of each class to the total output traffic.

One of the explanations of the causes of self-

Table 3: Volume transmitted (as a percentage of the total volume transmitted in bytes) for each of the main file classes in the systems under study

PUCPR 2012		IRCACHE 2004		WC 1998	
GIF	27.27%	JPEG	21.04%	ZIP	30.79%
JPEG	24.33%	OC	18.51%	html	27.57%
html	6.84%	html	12.42%	GIF	23.20%
PNG	4.47%	PLAIN	7.55%	JPEG	10.68%
PDF	3.09%	GIF	6.87%	OTHERS	3.03%
MPEG	2.15%	XML	2.94%	MOV	1.99%
ZIP	1.95%	MPEG	2.25%	HQX	1.24%
AVI	1.42%	FLASH	1.57%	CLASS	0.86%
GZ	1.13%	PDF	1.28%	PL	0.64%

similar nature of Web traffic is size of transmitted files, which commonly follows a heavy tail distribution [1] [2]. A random variable X is heavy tailed distributed if

$$\Pr\{X > x\} \sim x^{-\alpha}, \quad x \to \infty \tag{2}$$

where $0 < \alpha < 2$ is called the shape parameter, c is a positive constant and means that radio of two sides tends to 1 as $x \to \infty$. The value of α is important to many practical situation, with $0 < alpha < 1$, the average and variance of X do not converge, with $1 < alpha < 2$ the average converges but the variance does not. For $\alpha > 2$, the variable X has average and variance. The simple and most important method to estimate α is by plotting the complementary distribution function $\overline{F}(x) = 1 - F(x)$ on log-log axes [18]. For large x, the heavy tailed distribution have the property that

$$log(\overline{F}(x))/d\,log(x) = -\alpha, \tag{3}$$

Thus, we will investigate if the file classes also can be characterized by heavy tail distributions, using $\overline{F}(x)$ plotted on log-log axes. Linear behavior on the plot for upper tail is evidence of heavy-tailed distribution [18] and a sudden decay in the tail indicates a variable with short range dependence.

The Figures 4.1, 4.1 and 4.1 present the $\overline{F}(x)$ on log-log axes for the main classes for WC98, IRCache and PUCPR, respectively. In the WC98 classes, there is no heavy tail detected. This is a particular characteristic of this system - the number of files is relatively limited, therefore limiting the variance of transmitted file sizes. For the IRCACHE, the file sizes represent a large pool of

Table 4: Summary of statistics and goodness-of-fit for the WC98 server file sizes

Class	Model	Parameters	
GIF	Lognormal	$\mu = 1646$	$\sigma = 4100$
html	Lognormal	$\mu = 13550$	$\sigma = 16284$
JPEG	Lognormal	$\mu = 10210$	$\sigma = 8142$
ZIP1	Lognormal	$\mu = 254100$	$\sigma = 159773$
ZIP2	Lognormal	$\mu = 1564000$	$\sigma = 233676$
MOV	Lognormal	$\mu = 1441000$	$\sigma = 285251$
PL	Lognormal	$\mu = 64720$	$\sigma = 212349$
HQX	Lognormal	$\mu = 2236000$	$\sigma = 451173$
CLASS	Lognormal	$\mu = 4627$	$\sigma = 994$
OTHERS	Lognormal	$\mu = 18210$	$\sigma = 68428$

Web servers and, in this case, the size of files for all classes show a heavy tail behavior (with the exception of XML class). The file sizes for classes of PUCPR system are presented by Figure 4.1, that also exhibits heavy tail behavior for all classes.

The file sizes for each class were characterized for the WC98 server. Figure 7 shows the empirical cumulative distribution compared with the Lognormal distribution. The continuous line represents the theoretical distribution, and the dashed line the sample distribution. Note the good fit between the data and the Lognormal distribution. A summary of the basic statistics for the WC98 logs that were analyzed is given in Table 4. The characterization observed in this study is consistent with that of a similar, earlier study by Arlitt and Jin [6].

The size of files for the classes of IRCache and PUCPR systems present heavy tail behavior. The estimation for α was done by visual inspection of complementary distribution function on log-log axes and applying the Equation 3. Table 5 shows the results for each class. The heavy tail behavior can be observed in all important classes, and the typical value for α parameter is 1.0. This indicates that the aggregation of files of same type for many servers results in a hight variability for file sizes.

4.2 Identification of the Transition-Probability Matrix

Table 6 shows the transition-probability matrix for the WC98 system. The start and end of the sessions were inferred using the heuristics already described at the beginning section 4. The col-

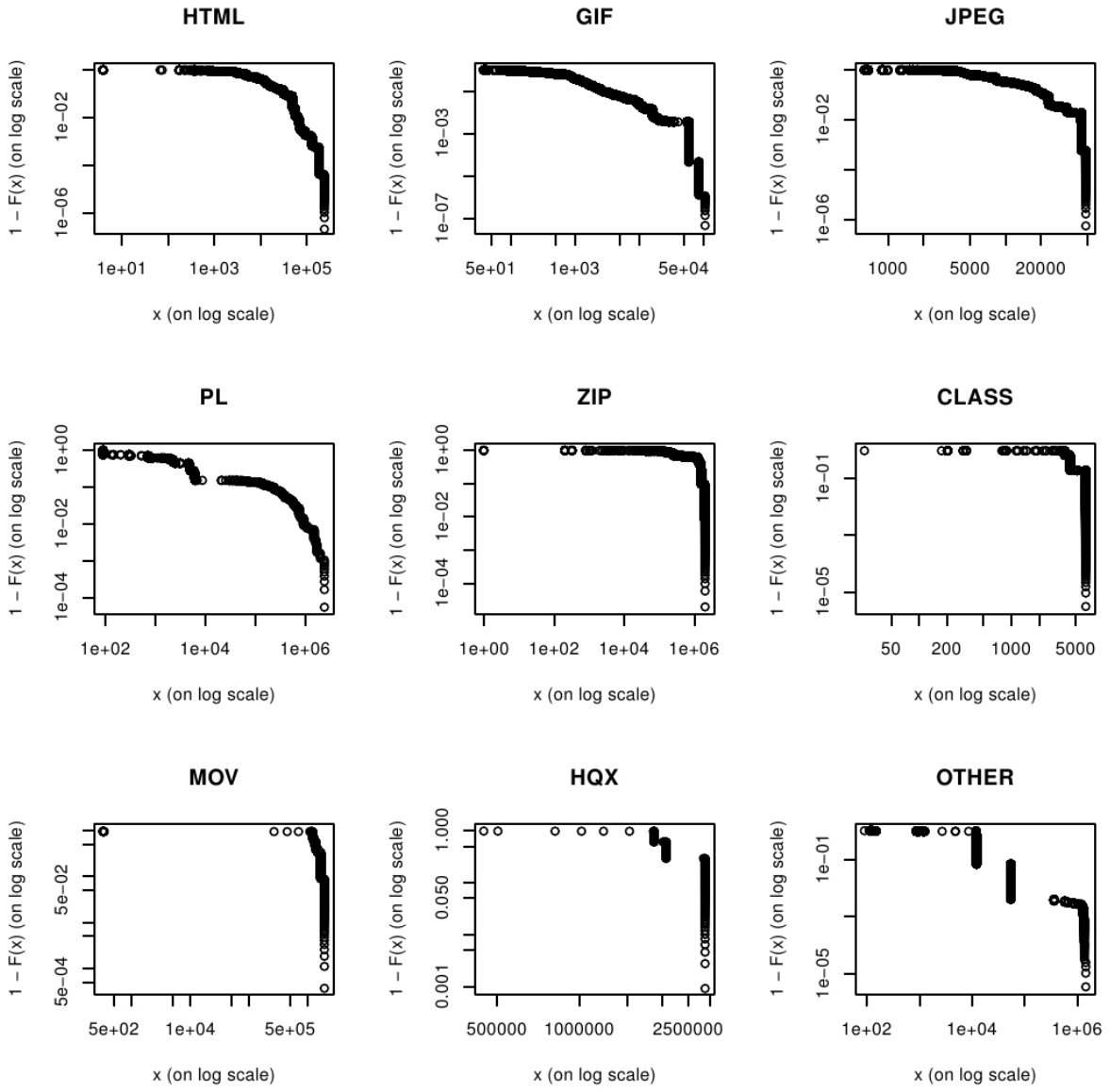

Figure 4: Complementary distribution function on log-log axes for main file classes of WC98

Table 6: Transition-Probability Matrix for the WC98 Web Server

	html	JPEG	GIF	ZIP1	ZIP2	MOV	HQX	CLASS	PL	OTHERS	END
html	0.3105	0.0881	0.5244	0.0003	0.0002	<0.0001	<0.0001	0.0045	0.0002	0.0219	0.0495
JPEG	0.1279	0.3145	0.5075	0.0014	0.0002	0.0003	<0.0001	0.0053	0.0002	0.0089	0.0333
GIF	0.0657	0.0346	0.8634	0.0001	0.0002	<0.0001	<0.0001	0.0110	0.0002	0.0064	0.0180
ZIP1	0.1539	0.0258	0.1658	0.2601	0.0159	<0.0001	0.0001	0.0008	0.0025	0.0117	0.3630
ZIP2	0.0908	0.0122	0.1142	0.0091	0.1323	<0.0001	0.0025	0.0008	0.0039	0.0039	0.6298
MOV	0.1234	0.1428	0.1076	0.0014	0.0014	0.3094	<0.0001	<0.0001	<0.0001	0.0093	0.3043
HQX	0.0815	0.0163	0.1068	0.0036	0.0724	<0.0001	0.1757	<0.0001	0.0054	0.0126	0.5253
CLASS	0.0661	0.0202	0.8622	<0.0001	<0.0001	<0.0001	<0.0001	0.0217	<0.0001	0.0042	0.0251
PL	0.0794	0.0451	0.3973	0.0008	0.0023	<0.0001	<0.0001	0.0002	0.2850	0.0067	0.1828
OTHERS	0.1830	0.0509	0.6424	0.0005	0.0003	<0.0001	<0.0001	0.0082	0.0002	0.0380	0.0761

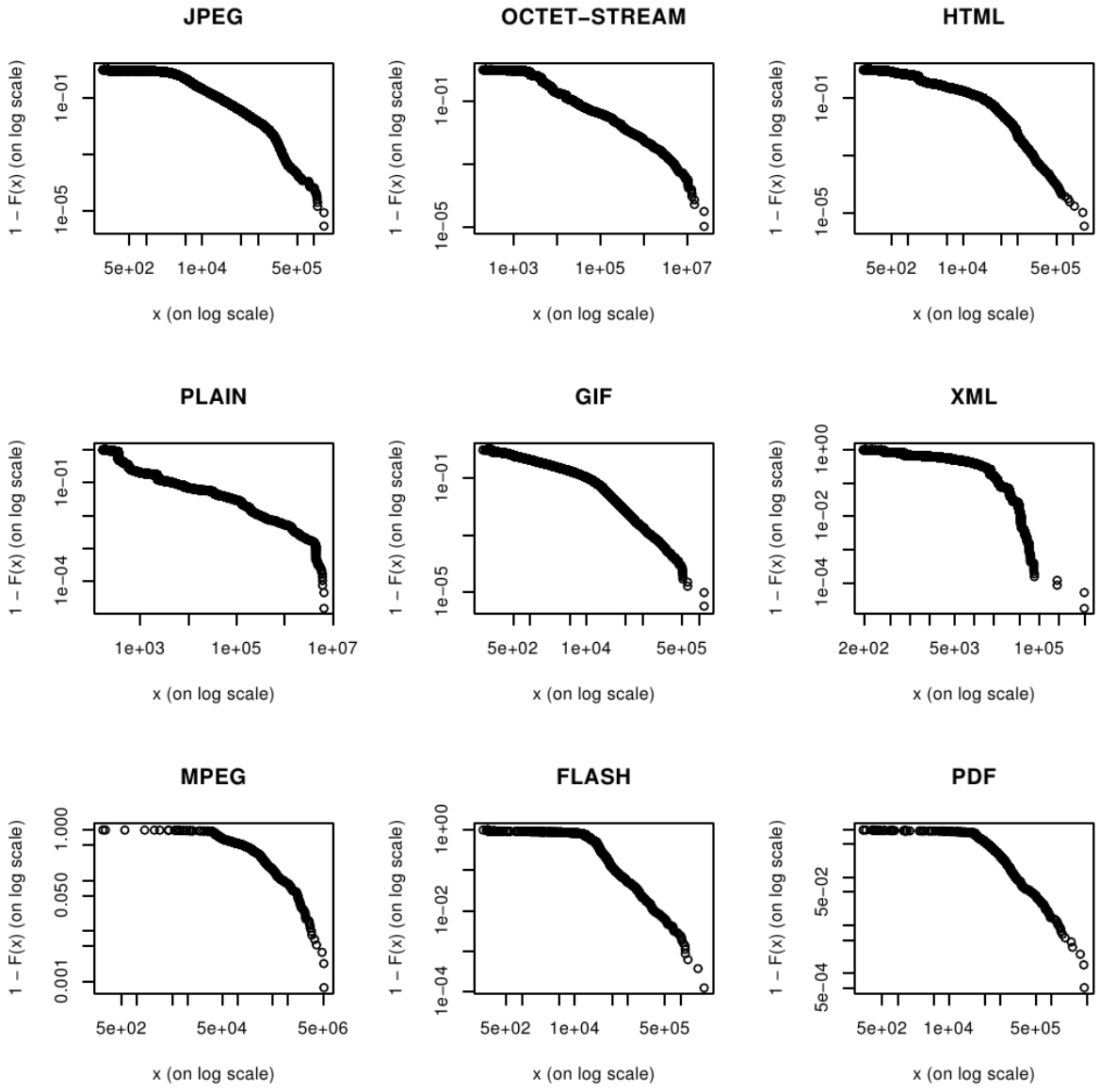

Figure 5: Complementary distribution function on log-log axes for main file classes of IRCACHE

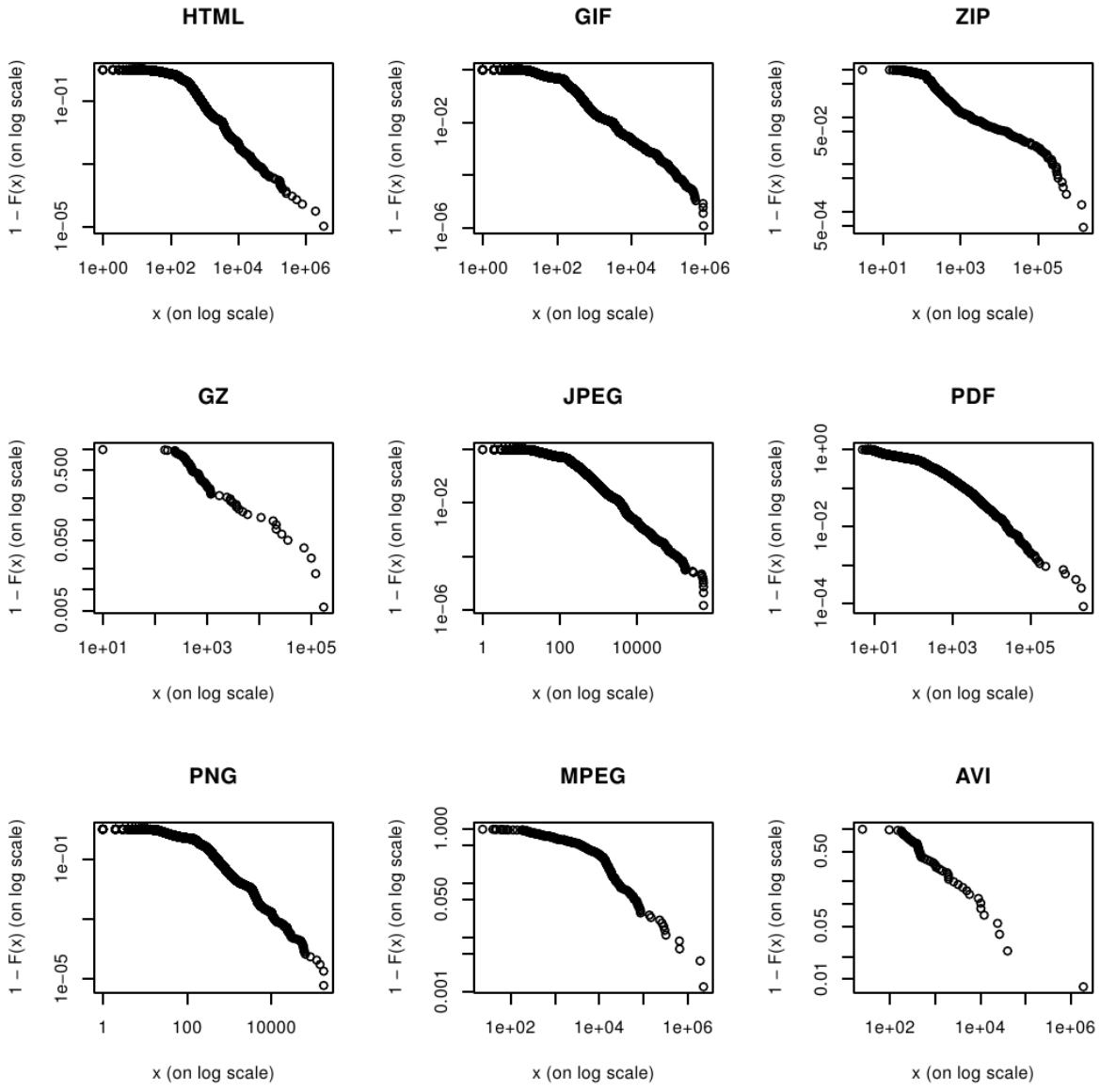

Figure 6: Complementary distribution function on log-log axes for main file classes of PUCPR

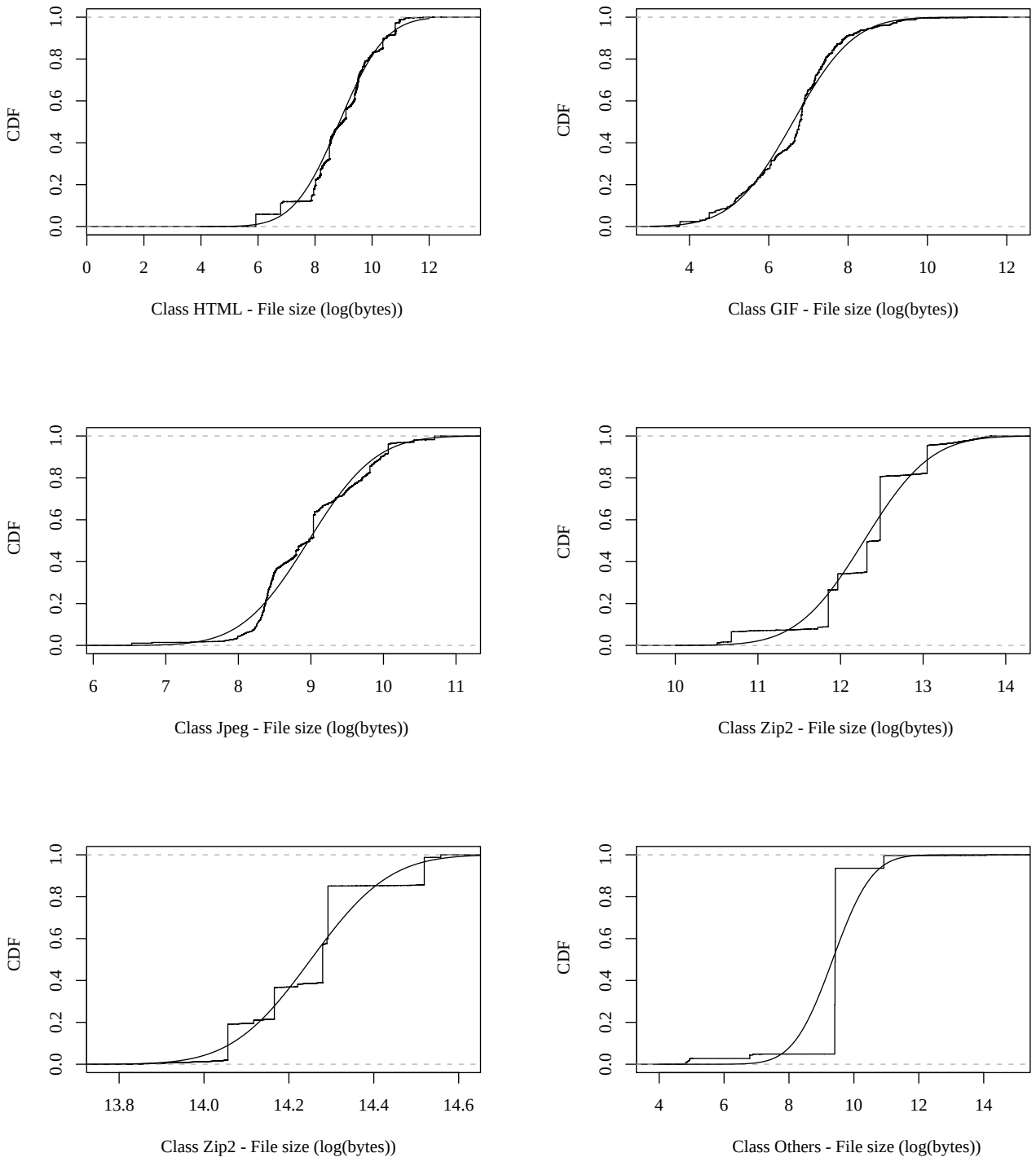

Figure 7: Cumulative Distribution Function (CDF) of the log-transformed file sizes for the main WC98 server file classes vs. fitted Normal Distribution

Table 5: Estimation for α parameter

PUCPR 2011		IRCACHE 2004	
GIF	0.87	JPEG	1.19
JPEG	0.87	OC	0.57
html	1.11	html	1.27
PNG	1.03	PLAIN	0.42
PDF	1.00	GIF	0.89
MPEG	1.19	XML	3.27
ZIP	0.46	MPEG	1.37
AVI	0.95	FLASH	1.11
GZ	0.65	PDF	1.11

lected data are an estimator for Equation 1.

The transition-probability matrix remains constant throughout the three days which the WC98 server logs were studied. This suggest that this matrix is a Web system invariant and can be used for performance evaluation as well as load generation. The importance of web invariants is discussed at [19].

From PUCPR proxy log, we first identified the ten most accessed Web servers. Among the most frequently accessed servers, there are some that have not been used, for instance, *swupmf.adobe.com* and *google-analytics.com*. Those server was excluded because it does not represent request from users, but requests made by certain software to automatically check for updates or upload statistics. Table 8 shows the result of analysis.

For each top ten Web server, the HTTP-ON state were analyzed and the transition-probability matrix were extracted. The user activity was extracted using source and destination IP address and port. We used the threshold of 20 seconds on inactivity to conclude about the end of HTTP-ON state. In order to adjust this threshold we tested the threshold of 2, 5, 10, 20 and 120 seconds and compare with a data set previously analyzed. The threshold of 20 seconds presents better results.

Using HTTP-ON data from user sessions for each server, we classified the files using the file extension, because the mime type do not always reflected the extension of the requested file. The similar file extension, as *html* and *htm* or *jpeg* and *jpg* were grouped in same class. Table 9 shows the transition-probability matrix for a major brazilian news portal. The transition probability matrices for the remaining servers are presented in *http://www.eletrica.ufpr.br/pedroso.*

Table 7: Servers with the highest number of hits of PUCPR log

Table 8: tab:toptenPUC

Server	Type of Web Server	Number of file transfers
google.com	search	110839
mail.google.com	mail	53821
google-analytics.com	statistics upload	52676
globo.com.br	news	47047
icisaude.org.br	government portal	39739
terra.com.br	portal	61257
l.yimg.com	statistics upload	31153
yahoo.com	search	30025
br.adserver.yahoo.com		25667
googleads.g.doubleclick.net	24194	
col.stb.s-msn.com	23447	
gfx1.hotmail.com	20781	
ad.yieldmanager.com	20217	
rad.msn.com	20118	
globoesporte.globo.com	sport news	18652
h.msn.com		18409

5 Simulation and validation

The first simulation consists in testing the capacity of proposed model to produce aggregated traffic that presents self similar characteristics. The proposed model was parametrized according Tables 4 and 6 (WC98 server) and submitted to the NS-2 simulator [20]. The network topology is shown in the Figure 8.

Figure 9 shows the observed server output traffic on a 1-second scale. The Hurst param-

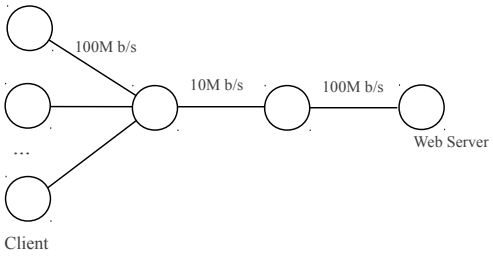

Figure 8: Simulation topology

Table 9: Transition-Probability Matrix for the *www.oglobo.com*, a major Brazilian news portal

Class	html	gif	jpg	javascript	css	png	other	end
start	0.2138	0.2419	0.2730	0.0425	0.2273	0	0.0015	0
html	0.4256	0.0119	0.2934	0.0408	0.1601	0	0.0031	0.0651
gif	0.0020	0.4034	0.0494	0.2572	0.0227	0.0002	0.0101	0.2551
jpg	0.0108	0.1028	0.2749	0.2254	0.2716	0	0.0007	0.1138
javascript	0.0004	0.0901	0.0545	0.8129	0.0011	0.0001	0.0032	0.0376
css	0.0040	0.0553	0.1461	0.0557	0.7095	0	0.0002	0.0293
png	0	0	0	0.1429	0	0.7143	0	0.1429
other	0.0149	0.3433	0.0448	0	0.0224	0	0.1119	0.4627

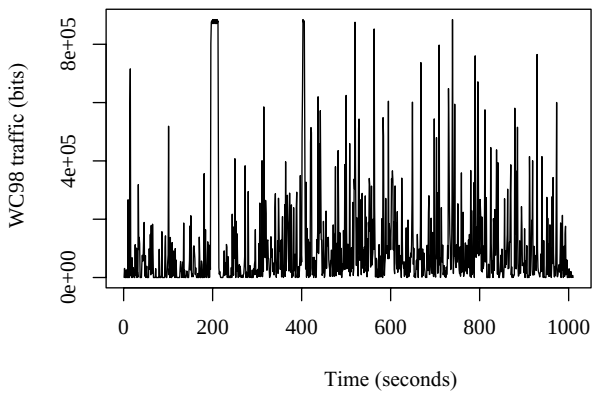

Figure 9: Time series representing the simulated output traffic for the WC98 server on a 1-second scale

eter was estimated using the Wavelet transform method [3] and the tools developed by Roughan et al. [21]. More information about Hurst parameter evaluation can be found in [3]. The Hurst parameter is (95% of confidence level in the interval $[0.843, 0.957]$), illustrating the model's ability to generate traffic with high-variance characteristics, as reported by [1] for Web servers.

In a second test we generate synthetic traces using the proposed model and the model proposed by [15]. The models were parametrized using the data extracted PUCPR logs. The synthetic traces was compared with real traffic using the autocorrelation function. Figure

6 Results and Discussion

The SURGE model is one of the best models available for Web workload generation. However, as the model uses a single probability distribution to characterize the size of the transmitted files, and the use of a heavy-tailed distribution is usually necessary, problems with statistical analysis can arise [22]. [8] suggests splitting the transmitted file size distribution into two distributions: a lognormal distribution for the body and the Pareto distribution for the tail. This leads to another problem, namely, how to choose the correct parameter identification when modeling real systems. In addition, it causes a problem with implementation of the model in network simulators (e.g. NS-2). The NS-2 simulator is not capable of implementing a file-size probability distribution composed of two different distributions and generating the transmitted file set accordingly. Thus, the user is forced to use only one file-size probability distribution, which can lead to incorrect traffic generation.

The model proposed in this paper presents advantages if compared to original SURGE model mainly in the following aspects. First, in SURGE, the file size is modeled by only one probability distribution. However, the data analysis show that is necessary to employ different distributions for the body and the tail, leading to a major difficulty: it is very difficult accurately characterizing complex empirical mixture distributions. Our model solve these problems by separating the transmitted files into classes, transferring the complexity to the file class selection. The file size of each class is modeled by a single probability distribution, thus avoiding mixed probability distributions. In this study, we use the file extension and file size as classification criteria. However, more elaborate criteria could have been used depending on the system characteristics and the purpose of the analysis. We showed that the lognormal distri-

bution usually fits the file size distribution and that the Weibull distribution fits the distribution of the permanence time in each file class.

Second, changes in server contents can be easily identified and incorporated to the model, involving only the re-characterization of few file classes - in original SURGE the entire file set must be studied. For example, assuming the sequence of files of a session request not changing but only the files size, the invariance of the transition-probability matrix can be assumed. The file distribution parameters can easily be used to fine tuning server contents adaptation to evaluate the workload pattern generated from it. Third, the invariance of the transition probability matrix lead us to conjecture about better content adaptation solutions based on our model parameters. Finally, an analytical model can be build through semi-Markov chain the from the file class state diagram. This model could be used to investigate Web service performance, as proxy systems.

In the SURGE-CC is only necessary to analyze server characteristics and not the network traffic, which is an advantage compared with the model proposed by [17], [23], [24] and [4]. Our model validation shows the SURGE-CC model is capable to generate output traffic with self similar characteristics.

7 Conclusion and Future Work

This paper presents a new model (SURGE-CC) to generate synthetic traffic of web servers. The SURGE-CC requires the following information extracted from the server log files to characterize Web server traffic:

1. File classes;
2. Average file size and standard deviation of file size for each class;
3. Class transition-probability matrix;
4. Client inter-session arrival times.

The better accuracy of the traffic generation could be achieved by the use of only one probability distribution for each class instead the use of a combination of heavy tailed distributions for the entire set of transmitted files.

Our approach to the model development opens several new applications to web performance evaluation based on the server contents. The contributions include new possibilities in terms of analytical models (for example, use of Semi-Markov chains); more accurate traffic generation (the approach represents an improvement over the SURGE model); and new possibilities for research into techniques to improve Web systems performance (for instance, the development of cache management algorithms, server contents adaptation).

Changes on the server characteristics can be easily identified and related to the changes on file classes, file sizes or even other content parameters with the SURGE-CC model. Its flexibility to parametrization made it simple to represent server contents changes. Presently, we are not aware of any model with such flexibility - hard work of statistical characterization of collected data set is required to represent changes in server contents.

Future work refers to model applications: as an example, compression rates of image files can be predicted to adapt output server traffic to the available bandwidth. Similarly, new network resource capacities could be planned based on new server contents. Another application refers to cluster computing - file classes separated on servers made possible the parameters customization to each cluster element aiming a better system performance.

References:

[1] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1995.

[2] W.E. Leland, M.S. Qaqqu, W. Willinguer, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[3] W. Willinger and K. Park. *Self-similar network traffic and performance evaluation.* John Wiley & Sons, New York, 1st edition, 2000.

[4] L. Muscariello, M. Mellia, M. Meo, and M. Ajmone-Marsan. An MMPP-based hierarchical model of internet traffic. In *IEEE international conference on communications ICC2004*, 2004.

[5] Vern Paxson and Sally Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ ACM Transactions on Networking*, 3(3):226–244, 1995.

[6] Martin Arlitt and Tai Jin. A workload characterization study of the 1998 World Cup Web site. *IEEE Network*, 14:30–37, 2000.

[7] D. Wessels and K. Claffy. Evolution of the NLANR cache hierarchy: Global configuration challenges, 1996. Technical report, NLANR, October 1996. http://www.nlanr.net/Papers/Cache96/.

[8] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *Joint International Conference on Measurement and Modeling of Computer Systems - Performance Evaluation Review (SIGMETRICS '98/PERFORMANCE '98)*, 1998.

[9] Hyoung-Kee Choi and John O. Limb. A behavioral model of web traffic. In *Network Protocols, 1999. (ICNP '99) Proceedings. Seventh International Conference on*, pages 327–334, 1999.

[10] R. Pries, Z. Magyari, and P. Tran-Gia. An http web traffic model based on the top one million visited web pages. In *Next Generation Internet (NGI), 2012 8th EURO-NGI Conference on*, pages 133–139, 2012.

[11] Raoufehsadat Hashemian, Diwakar Krishnamurthy, and Martin Arlitt. Web workload generation challenges: an empirical investigation. *Software: Practice and Experience*, 42(5):629–647, 2012.

[12] Jianliang. Xu, Samuel T. Chanson, and SpringerLink (Online service). *Web Content Delivery*. Web Information Systems Engineering and Internet Technologies Book Series ;. Springer US,, Boston, MA :, 2005.

[13] I. Tsompanidis, A.H. Zahran, and C.J. Sreenan. Mobile network traffic: A user behaviour model. In *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, pages 1–8, May 2014.

[14] B.A. Mah. An empirical model of http network traffic. In *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 2, pages 592–600, 1997.

[15] J.J. Lee and M. Gupta. A new traffic model for current user web browsing behavior. Technical report, Intel Cooperation, 2007. Santa Clara, CA, USA.

[16] Bruce A. Mah. An empirical model of http network traffic. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 592, Washington, DC, USA, 1997. IEEE Computer Society.

[17] F. Hernandez-Campos, K. Jeffay, and F.D. Smith. Tracing the evolution of the web traffic: 1995-2003. In *IEEE/ACM MASCOTS 2003 – The 11th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2003.

[18] Mark Crovella and Murad S. Taqqu. Estimating the heavy tail index from scaling properties. In *Methodology and Computing in Applied Probability*, pages 55–79, 1999.

[19] Daniel A. Menasc and Virgilio A. F. Almeida. *Capacity planning for Web performance*. Prentice Hall, 1998.

[20] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in network simulation. *IEEE Computer*, 33(5):59–67, 2000.

[21] Matthew Roughan, Darryl Veitch, and Patrice Abry. On-line estimation of the parameters of long-range dependence. In *Proceedings Globecom '98*, volume 6, pages 3716–3721, Sydney, 1998.

[22] Wei-Bo Gong, Yong Liu, Vishal Misra, and Donald F. Towsley. Self-similarity and long range dependence on the internet: a second look at the evidence, origins and implications . *Computer Networks*, 48(3):377–399, 2005.

[23] Joel Sommers and Paul Barford. Self-configuring network traffic generation. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 68–81, New York, NY, USA, 2004. ACM Press.

[24] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. Weigle. Stochastic Models for Generating Synthetic HTTP Source Traffic. In *IEEE Infocom*, 2004.