A Parsimonious Multifractal Model for WWW Traffic*

Abdullah Balamash and Marwan Krunz

Department of Electrical & Computer Engineering
University of Arizona
Tucson, AZ 85721
{balamash,krunz}@ece.arizona.edu

Abstract. In this paper, we capture the main characteristics of WWW traffic in a stochastic model, which can be used to generate synthetic WWW traces and assess WWW cache designs. To capture temporal and spatial localities, we use a modified version of Riedi et al.'s multifractal model [18], where we reduce the complexity of the original model from $\mathcal{O}(N)$ to $\mathcal{O}(1)$; N being the length of the synthetic trace. Our model has the attractiveness of being parsimonious and that it avoids the need to apply a transformation to a self-similar model (as often done in previously proposed models [2]), thus retaining the temporal locality of the fitted traffic. Furthermore, because of the scale-dependent nature of multifractal processes, the proposed model is more flexible than monofractal models in describing irregularities in the traffic. Trace-driven simulations are used to demonstrate the goodness of the proposed model.

keywords — WWW modeling, web caching, multifractals, stack distance, self-similarity.

1 Introduction

The ability to assess the performance of WWW caching policies hinges on the availability of a representative workload that can be used in trace-driven simulations [5, 13]. Measured ("real") traces can be used for this purpose. However, due to the difficulty associated with capturing real traces, only a handful of such traces are available in the public domain (see [1]). This makes it hard to provide simulation results with reasonable statistical credibility. A more feasible alternative is to rely on synthetic traces that are derived from a stochastic model. The need for such a model is the main motivation behind our work.

In this paper, we use a modified version of the multifractal model by Riedi [18] to simultaneously capture the temporal and spatial localities in WWW traffic. Riedi's model has the attractiveness of being able to simultaneously capture the (lognormal) marginal distribution and the correlation structure of a time series.

* This work was supported in part by the National Science Foundation under grants CCR 9979310 and ANI 0095625.

Its main disadvantage is its complexity, which grows linearly with the size of the generated trace. We modify this model, reducing its complexity to $\mathcal{O}(1)$. The resulting (modified) model is parsimonious, in that it is characterized by four to five parameters, that represent the mean, variance, and correlation structure of the "scaled stack distance" string (see below). The popularity profile of the traffic is incorporated in the model during the trace generation phase (assuming that the popularity profiles for all documents are given beforehand). Our model is mainly intended for offline generation of the traffic demand *seen by a WWW server*. Accordingly, the popularity profiles can be easily computed from the server logs.

Two datasets were used in our study. The first one was captured at the Computer Science Department of the University of Calgary, while the second set was produced by ClarkNet, a commercial Internet Provider in Baltimore, Washington DC [1, 3]. Details of these traces can be found in [1, 3]. Note that the two traces have contrasting loads (Calgary's load is light while ClarkNet's load is very heavy). The data provide several pieces of information, including the name of host that generated the URL request, the day and time the request was recorded, the name of the requested file, the HTTP reply code (explained below), and the number of transferred bytes in response to the request. Four types of HTTP reply codes were recorded: *successful*, *not modified*, *found*, and *unsuccessful*. In our analysis, we only included the requests with *successful* code, since they are the ones that result in actual data transfer from the server. We also excluded dynamic files (e.g., cgi and pl files).

WWW traffic modeling has been the focus of several previous studies; examples of which are given in [15, 2, 4, 14, 8]. In these studies, the temporal locality of the traffic was represented by the marginal distribution of the stack distance string. This distribution was found to follow a lognormal-like shape. The stack distance string, which is an equivalent representation of a reference string, is obtained by transforming the reference string using the LRU stack. In [2] the authors showed that spatial locality can be captured (at least, in part) through the autocorrelation structure (ACF) of the stack distance string. They argued that the stack distance string exhibits long-range dependence (LRD) behavior. Thus, to simultaneously model the marginal distribution (temporal locality) and the correlation structure (spatial locality) of the stack distance string, they relied on the work in [12], which proved the invariance of the Hurst parameter to transformations of the marginal distribution of an LRD process. More specifically, the authors in [12] proved that under some mild assumptions, a point-by-point transformation $Y = F_y^{-1}(F_x(X))$ of a *Gaussian* self-similar process X with Hurst parameter H results in a self-similar process Y with the same Hurst parameter, where F_x and F_y are the CDFs for X and Y, respectively. It should be noted, however, that the proof of this result is valid asymptotically and only for Gaussian processes (e.g., fractional ARIMA). More importantly, while this result assures the invariance of H, it does not necessarily preserve the shape of the ACF. As an example, consider the transforming of the Gaussian distribution of a F-ARIMA model into a lognormal distribution, which adequately models

the marginal distribution of the stack distance string. The resulting ACFs are shown in Figure 1, along with the ACF of the "real" traffic. The figure illustrates the two main drawbacks of the transformation. First, the transformation distorts the overall shape of the ACF of the F-ARIMA process. Second, the original F-ARIMA model itself is not accurate in representing the real ACF at finite lags.

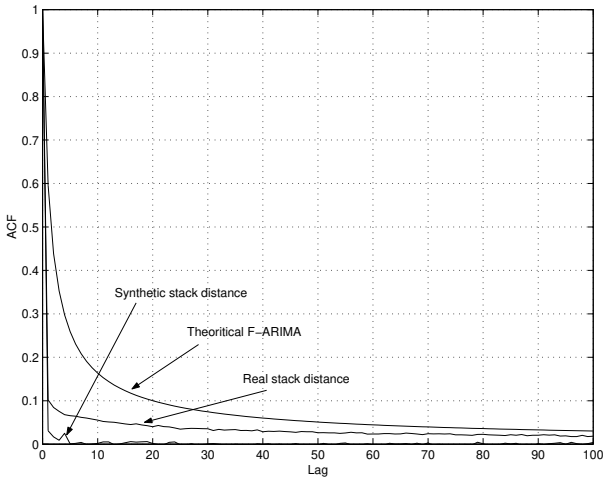

Fig. 1. Impact of transforming the distribution of a F-ARIMA model on the ACF.

To avoid the problems stated above, we resort to multifractal modeling to simultaneously capture the correlation structure and the marginal distribution of the stack distance string. Multifractality is a generalization of self-similarity (monofractality), whereby the Hurst parameter (the scaling exponent) is not fixed, but varies with scale. This variability makes multifractal processes more flexible than monofractal processes in describing "irregularities" in the traffic (e.g., contrasting short-term and long-term behaviors). The reader is referred to [17, 11, 18, 9, 10] and the references therein for comprehensive discussions of multifractal processes. In [18] the authors used a wavelet-based construction of a multifractal process to show that the correlation behavior of a strongly correlated time series can be approximately captured by appropriately setting the second moments of the wavelet coefficients at each scale of the multifractal generation process. This result provides the basis for modeling the ACF of the stack distance string. Combined with the fact that the above multifractal model exhibits an approximately lognormal marginal distribution, they can be used to model both the temporal and spatial localities in WWW traffic.

Relying on the observation that temporal locality is induced by both temporal correlation and long-term popularity [16], the authors in [6] introduced a new measure for temporal locality called the *scaled stack distance*. This measure rep-

resents the deviation of the stack distances from their expected values, *assuming that requests to a given document are uniformaly distributed over the duration of the trace*. The scaled stack distance captures the impact of short-term correlation, but does not capture the spatial locality. For our WWW traffic model, we use a similar measure with the same name, but that measures the deviation of the stack distances from their *empirical* expected values. We model the expected stack distance as a function of the popularity profile. Equally popular documents have the same expected stack distance. This scaled stack distance string was found to have a lognormal-like distribution and the same correlation structure as the original stack distance string.

We use extensive simulations to evaluate the performance of our WWW traffic model and contrast it with the self-similar model in [2] and the model in [6], using the original (real) traces as a point of reference. Our performance measures include sample statistics of the synthetic traces (e.g., mean, variance, correlations, percentiles) as well as the cache and byte hit ratios for a trace-driven LRU (least recently used) cache. The results indicate marked improvement in performance when using the proposed multifractal-based WWW model.

The rest of the paper is organized as follows. Section 2 gives a brief overview of Riedi et al.'s multifractal model and the modification we make to it to render it parsimonious. The proposed WWW traffic generation approach is given in Section 3, followed by simulation studies in Section 4. We conclude the paper in Section 5.

2 Multifractal Analysis of WWW Traffic

As indicated earlier, multifractality is a generalization of monofractality (self-similarity), where the fixed (scale independent) H parameter of a self-similar process is now scale dependent. Certain multifractal processes, including the one considered in this paper, inherently exhibit lognormal-like marginals, in line with the shape of the marginal distribution of typical WWW traces. This convenient feature allows us to skip the risky step of transforming the marginal distribution, leaving us with the task of fitting the ACF. In this section, we first briefly describe Riedi et al.'s multifractal model [18]. This model uses a wavelet-based construction to approximately capture the correlation behavior of a given time series by appropriately setting the second moments of the wavelet coefficients at each scale. We then describe how we modify this model to reduce its complexity from $\mathcal{O}(N)$ to $\mathcal{O}(1)$. We then apply the modified model in characterizing the temporal and spatial localities of WWW traffic.

2.1 Riedi et al.'s Multifractal Model

Riedi et al.'s model relies heavily on the discrete wavelet transform. The idea behind the wavelet transform is to express a signal (time function) $X(t)$ by an approximated (smoothed) version and a detail. The approximation process is repeated at various levels (scales) by expressing the approximated signal at a

given level j, say X_j, by a coarser approximation at level $j-1$, say X_{j-1}, and a detail D_{j-1}. At each scale, the approximation is performed through a scaling function $\phi(t)$, while the detail is obtained through a wavelet function $\psi(t)$. More formally, a wavelet expansion of the signal $X(t)$ is given by:

$$X(t) = \sum_k U_{J,k}\phi_{J,k}(t) + \sum_{j=J}^{\infty}\sum_k W_{j,k}\psi_{j,k}(t) \tag{1}$$

where

$$W_{j,k} \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} X(t)\psi_{j,k}(t)dt \tag{2}$$

$$U_{j,k} \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} X(t)\phi_{j,k}(t)dt \tag{3}$$

and $\psi_{j,k}$ and $\phi_{j,k}$, $j,k = 0,1,2,\ldots$, are *shifted* and *translated* versions of the wavelet and scaling functions $\psi(t)$ and $\phi(t)$, respectively, and are given by:

$$\psi_{j,k}(t) \stackrel{\text{def}}{=} 2^{-j/2}\psi(2^{-j}t - k) \tag{4}$$

$$\phi_{j,k}(t) \stackrel{\text{def}}{=} 2^{-j/2}\phi(2^{-j}t - k). \tag{5}$$

In (1), the index J indicates the coarsest scale (the lowest in detail). The coefficients $W_{j,k}$ and $U_{j,k}$ are called the wavelet and scale coefficients at scale j and time $2^j k$. Together, they define the discrete wavelet transform of the signal $X(t)$ (assuming that $\phi(t)$ and $\psi(t)$ are specified).

Several wavelet and scale functions have been used in the literature, giving rise to different wavelet transforms. One popular (and simple) transform is the Haar wavelet transform. This transform, which is specified by the coefficients $W_{j,k}$ and $U_{j,k}$ for all j and k, can be obtained recursively as follows (we adopt the same convention of [18], in which the higher the value of j, the better the approximation of the original signal):

$$U_{j,k} = \frac{U_{j+1,2k} + U_{j+1,2k+1}}{\sqrt{2}} \tag{6}$$

$$W_{j,k} = \frac{U_{j+1,2k} - U_{j+1,2k+1}}{\sqrt{2}} \tag{7}$$

To initialize the recursion, the values of $U_{j,k}$, $k = 0,1,\ldots,2^j - 1$, at the highest value of j are taken as the empirical trace to be modeled.

In order to generate synthetic traces with a given autocorrelation structure, the Haar transform is reversed by rewriting (6) and (7) as:

$$U_{j+1,2k} = \frac{U_{j,k} + W_{j,k}}{\sqrt{2}} \tag{8}$$

$$U_{j+1,2k+1} = \frac{U_{j,k} - W_{j,k}}{\sqrt{2}} \tag{9}$$

Now to generate nonnegative data, which in our case represent the stack distance string, we need to have $|W_{j,k}| \leq U_{j,k}$. To satisfy this constraint, the wavelet coefficients can be defined as:

$$W_{j,k} = A_{j,k} U_{j,k} \tag{10}$$

where $A_{j,k}$ is a random variable (rv) defined on the interval $(-1, 1)$. Using (8), (9), and (10), the following recursion can be obtained for synthesizing the scale coefficients:

$$U_{j+1,2k} = (\frac{1 + A_{j,k}}{\sqrt{2}}) U_{j,k} \tag{11}$$

$$U_{j+1,2k+1} = (\frac{1 - A_{j,k}}{\sqrt{2}}) U_{j,k} \tag{12}$$

The rvs $A_{j,k}$ must also satisfy the following additional constraints [18]:

1. $A_{j,k}, k = 0, 1, ..., 2^j - 1$ are *i.i.d.*
2. For each j, the probability density function of the rvs $A_{j,k}, k = 0, 1, \ldots, 2^j - 1$, is symmetric with zero mean.
3. A_j is independent of A_l for $l > j$ and is also independent of $U_{0,0}$.

The wavelet energy at a given scale is defined as the variance of the wavelet coefficients at that scale. It has been shown that the correlation structure of the signal can be approximately captured by controlling the wavelet energy decay across scales. The ratio of the energy at scale $j - 1$ to the one at scale j (j is finer than $j - 1$) was found to be [18]:

$$\eta_j = \frac{E[W_{j-1}^2]}{E[W_j^2]} = 2 \frac{E[A_{j-1}^2]}{E[A_j^2](1 - E[A_{j-1}^2])} \tag{13}$$

Assuming that $E[W_j^2]$ is given for all j, Equation (13) can be used to solve for $E[A_j^2]$, $j = 1, 2, \ldots$. The recursion can be initialized using $E[A_0^2] = \frac{E[W_0^2]}{E[U_0^2]}$, where W_0 and U_0 are the wavelet and scale coefficients at the coarsest scale.

In [18], the authors suggested two different distributions for A_j. One of them is a symmetric beta distribution that has the following pdf:

$$f_{A_j}(x) = \frac{(1 + x)^{\rho_j - 1} (1 - x)^{\rho_j - 1}}{\beta(\rho_j, \rho_j) 2^{2\rho_j - 1}} \tag{14}$$

where ρ_j is the parameter of the rv and $\beta(.,.)$ is the beta function. The variance of this random variable is given by:

$$\text{var}[A_j] = \frac{1}{2\rho_j + 1}. \tag{15}$$

The other distribution is a point-mass distribution defined as:

$$\Pr[A_j = c_j] = \Pr[A_j = -c_j] = r_j$$
$$\Pr[A_j = 0] = 1 - 2r_j$$

In the case of a beta distributed A_j, the parameter ρ_j at each scale can be found by solving (13) and (15), resulting in:

$$\rho_j = \frac{\eta_j}{2}(\rho_{j-1} + 1) - 1/2 \tag{16}$$

This, however, assumes that $E[W_j^2]$ is given for $j = 1, 2, 3, \ldots$. Since η_j, $j = 1, 2, \ldots$, cannot be obtained using a parametric model, it would be computed from the empirical data, which makes the number of fitted parameters in the model in the order of N; N being the trace length.

On the other hand, if A_j has a point-mass distribution, then (13) by itself is not sufficient to compute both parameters of A_j (c_j and r_j). An alternative approach to computing these parameters is to rely on the following expression for the moments of the scaling coefficients at different scales [18]:

$$\frac{E[U_j^q]}{E[U_{j-1}^q]} = 2^{-q/2} E[(1 + A_{j-1})^q], \quad q = 1, 2, \ldots \tag{17}$$

However, to apply (17) one needs to have two moments (i.e., two values for q) for each scale j. Again, unless we can compute these values using a parametric model, we need to rely on the empirical data to do so, which makes the model more complex than if a beta distributed A_j were to be used.

It was shown in [18] that the above model (with either distribution of A_j) generates positive-valued autocorrelated data with an approximately lognormal marginal distribution.

2.2 Reducing the Number of Parameters

As shown in the previous section, whether A_j has a beta distribution or a point-mass distribution, one needs to provide the second moments of the wavelet coefficients or two moments of the scale coefficients at each scale in order to completely determine A_j, $j = 1, 2, \ldots$. This significantly increases the complexity of the model, as the number of parameters to be computed a priori is in the order of the trace length (unless we have a parameterized model to compute these values). Moreover, the point-mass rv is not rich enough and has only three possible values.

To reduce the complexity of the model, we let A_j be a triangular rv in the range $[-c, c]$. This distribution is richer than the point-mass distribution and has only one parameter. It allows us to fit the second moment of the scale coefficients for all scales using (17), provided that we can compute the second moments analytically knowing the mean μ and the variance σ of the modeled data, as will be shown later in this section.

For a discrete time series $X = \{X_i : i = 1, 2, \ldots\}$, we define $X^{(m)} = \{X_i^{(m)} : i = 1, 2, \ldots\}$ to be the aggregated time series of X at level m:

$$X_n^{(m)} = \sum_{i=nm-m+1}^{nm} X_i, n = 1, 2, 3, \ldots, N/m \tag{18}$$

where $m = 1, 2, 4, 8, ...N$; N is the length of X. Note that if the aggregation level m corresponds to scale j, then the aggregation level $2m$ corresponds to scale $j - 1$. From the definition of the Haar wavelet transform, the following holds:

$$\frac{E[(X^{(m)})^q]}{E[(X^{(2m)})^q]} = 2^{-q/2} \frac{E[U_j^q]}{E[U_{j-1}^q]}, \quad \text{for } q = 1, 2, \ldots \tag{19}$$

From (19) and (17) we get:

$$\frac{E[(X^{(m)})^q]}{E[(X^{(2m)})^q]} = 2^{-q} E[(1 + A^{(2m)})^q] \tag{20}$$

where $A^{(2m)} = A_{j-1}$. Let $c^{(2m)}$ be the parameter of the rv A_{j-1} at aggregation level $2m$. From (20) and the definition of the triangular random variable, we obtain the following expression for $c^{(2m)}$:

$$c^{(2m)} = \sqrt{6(4\frac{E[(X^{(m)})^2]}{E[(X^{(2m)})^2]} - 1)} \tag{21}$$

To reduce the number of parameters in the multifractal model, we analytically obtain the second moments of the scaling coefficients, as shown next. The variance at a given level of aggregation, $\text{var}[X^{(m)}] = V^{(m)}$, can be computed analytically as a function of the autocorrelation function of the signal [7]:

$$V^{(m)} = mv + 2v \sum_{k=1}^{m} (m - k)\rho_k \tag{22}$$

The mean, $E[(X^{(m)})] = \mu^{(m)}$, is given by:

$$\mu^{(m)} = m\mu \tag{23}$$

where μ and v are the mean and the variance of the original signal, respectively. The second moment of $X^{(m)}$ is then given by:

$$E[(X^{(m)})^2] = mv + 2v \sum_{k=1}^{m} (m - k)\rho_k + m^2\mu^2 \tag{24}$$

From Equations (21) and (24), the parameter of the rv A_j can be computed for all scales $j = 1, 2, \ldots$, given μ, v, and the correlation structure of the time series being modeled. For WWW traffic stack distance strings, we found that

the form $\rho_k = e^{-\beta \sqrt[n]{g(k)}}$, $k = 0, 1, \ldots$, fits the correlation structure very well, where g is a function of the lag k. For the ClarkNet trace, $g(k) = k$ produced a good fit to the empirical ACF, while for the Calgary trace, $g(k) = \log(k+1)$ was found appropriate.

In summary, to use the multifractal model for modeling the scaled stack distance string, we only need four parameters:

- Mean of the stack distance string (μ).
- Variance of the stack distance string (v).
- Autocorrelation structure (parameterized by β, n, and g).

Using these parameters, along with (24) and (21), one can compute the parameter $c^{(m)}$ at each aggregation level (scale).

The synthesis process starts from the highest level of aggregation. At this level we can start with l data points that are normally distributed with mean $m_h \mu$ (the mean at aggregation level m_h) and variance of $\mathrm{var}[X^{(m_h)}]$, where m_h is the highest aggregation level, which is the length of the trace that needs to be generated. After that, the process can be carried out using Equations (11) and (12).

3 Modeling WWW Traffic

In this section, we describe our approach for modeling the stream of file objects generated by a WWW server. Let U be the number of unique files (or objects) at the server and let fr_i be the fraction of times that the ith file, $i = 1, 2, ..., U$, appears in the reference string (fr_i is the popularity profile of file i). The modeling approach proceeds in three steps. First, we extract the stack distance string from the URL reference string. Then, we apply some form of scaling to capture both sources of temporal locality (temporal correlation and long-term popularity). The modified multifractal model described in the previous section is then applied to model the scaled stack distance string after computing its mean and variance and after fitting its correlation structure. Finally, we incorporate the popularity profile of the traffic during the process of generating synthetic reference strings. These main steps are described next.

3.1 Extracting the Empirical Scaled Stack String

In our model, we use the concept of stack distance to model the temporal and the spatial localities in WWW traffic. The authors in [4] extract the stack distances from the original trace assuming an arbitrary initial ordering of the stack. Whenever an object is requested, its depth (stack distance) in the stack is recorded and the object is pushed to the top of the stack. In our model we avoid making any assumptions on the initial ordering of the stack, which we have found to affect the marginal distribution and the correlation structure of the stack distance string. We start with an empty stack and process the empirical reference string

in the reverse direction, starting from the last reference. If a file is referenced for the first time (in the reverse direction), it is pushed to the top of the stack but no stack distance is recorded. Otherwise, if the file has already been referenced before (hence, it is already in the stack), then it is pushed from its previous location in the stack to the top of the stack and its depth is recorded as a stack distance. Finally, the resulting trace of stack distances is reversed to get the correct stack distance string. The following example illustrates the idea. Consider the reference string [a d c b c d d a b], where each letter indicates the name of a file. If we process this string starting from the end, the first reference is to file b. Since this is the first time file b is being referenced, we push it to the top of the stack without recording any distance. The same procedure is performed for the next two references (for files a and d). The fourth reference (from the end) is for file d. Since this file has been referenced before, it gets pushed to the top of the stack and its stack depth is recorded (in this case, the stack depth for file d is one). The procedure continues until all references are processed (see Figure 2). The end result of this process is the stack distance stream [4 3 2 4 1].

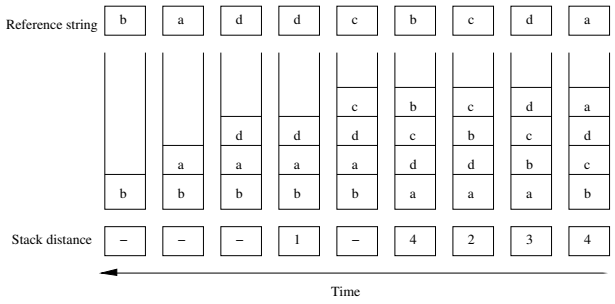

Fig. 2. Example showing our approach for extracting the stack distances from a real trace.

Temporal locality is attributed to both short-term correlations and long-term popularity [16]. Documents that have long-term popularity profiles tend to have small stack distances. Some documents are not popular but have short-term correlation profiles, which make these documents appear often within a short interval of time. As a result, these documents have small stack distances (i.e., they exhibit strong *short-term popularity*). In general, unpopular documents tend to have longer stack distances. The authors in [6] tried to model these trends by modeling the deviation of a stack distance from its expected value; assuming that the documents are uniformally distributed over the whole trace. Instead, we model the deviation of a stack distance from its *empirical expected value* (the scaled stack distance), as we found that the approach in [6] affects the correlation structure. We model the expected stack distance as a function of the popularity profile. Equally popular documents have the same expected stack distance. Figure 3 shows the relationship between the number of requests a file

gets (its popularity profile) and the empirical expected stack distance. In both traces, it is observed that the expected stack distance drops exponentially with respect to the popularity profile.

The scaled stack distance string is obtained by normalizing each stack distance by its expected value. This string was found to have an approximately lognormal marginal distribution and a slowly decaying correlation structure that is almost identical to the correlation structure of the stack distance string.

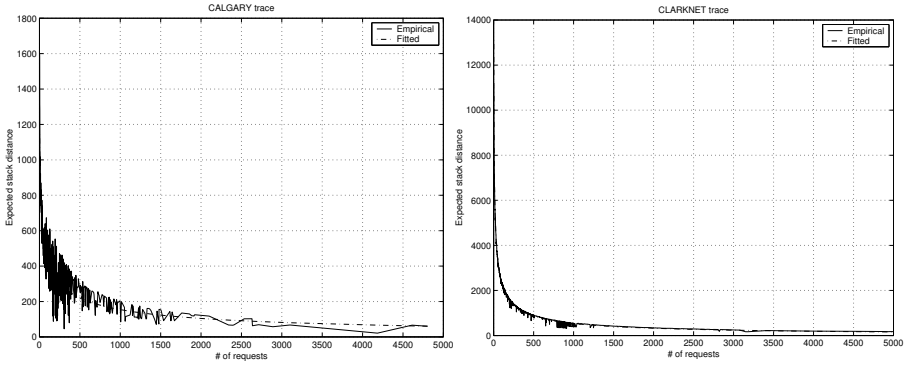

Fig. 3. Number of requests versus the expected stack distance for the two traces.

3.2 Modeling the Scaled Stack Distance String

To model the scaled stack distance string, we need to determine μ, v, β, and n. Once the values of these parameters are determined, the multifractal model described in Section 2 is used to capture the marginal distribution (temporal locality) and the correlation structure (spatial locality) of the scaled stack distance string.

3.3 Modeling Popularity and Generating Synthetic Reference Strings

To generate a synthetic WWW reference string, we first need to generate a synthetic scaled stack distance string, as shown in the previous section. The process of generating a synthetic WWW reference string starts by arranging the unique documents of the WWW server in an LRU stack. This is done by sampling from a probability distribution that is weighted by the popularity profiles of the various documents (i.e., the more popular a document is, the more likely it will be placed closer to the top of the stack). To generate a reference string of length N, we first compute the number of references a document can get according to its popularity profile. Then the top document at the LRU stack is considered as the next referenced document in the synthetic reference string. If the required

number of references for this document is reached, then this document is flushed out of the stack. Otherwise, it is pushed down the stack according to the next value in the scaled stack distance string. This is done after scaling back the scaled stack distance by multiplying it by the corresponding expected stack distance for the object in hand (objects with the same popularity profile have the same expected stack distance). This process continues until the popularity profiles of all objects are satisfied (no documents are left in the LRU stack).

4 Experimental Results

In this section, we evaluate the performance of the proposed multifractal model and contrasting it with two other models. The first model is a self-similar (monofractal) model [2, 4], which characterizes the temporal and spatial localities in WWW traffic. This model involves transforming the Gaussian marginal distribution of a fractional ARIMA process into a more appropriate distribution (e.g., lognormal). We simply refer to this model as the LRD model. The second model was proposed by Cherkasova et al. [6], which was discussed in the introduction. The three investigated models were mainly designed for offline operation, with the primary purpose of generating synthetic traces for use in cache design studies. Accordingly, we compare these models in terms of the file and byte miss ratios seen at an LRU cache that is drived by synthetic traces from these models. The comparison is made with reference to the cache performance seen under the real traffic (the two studied traces). The results are shown in Figures 4, 5, 6, and 7.

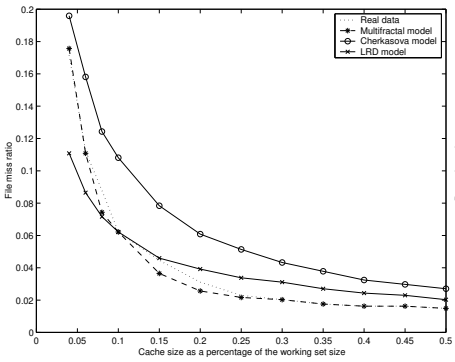

Fig. 4. File miss ratio versus cache size (CALGARY trace).

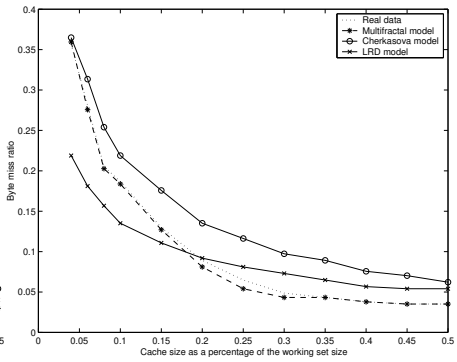

Fig. 5. Byte miss ratio versus cache size (CALGARY trace).

It is clear that of the three models, the proposed multifractal model produces the most accurate performance, especially for small cache sizes. The performance improvement is greater in the case of the CALGARY data. Consider, for example, the CALGARY data with a normalized cache size of 0.3. The percentage

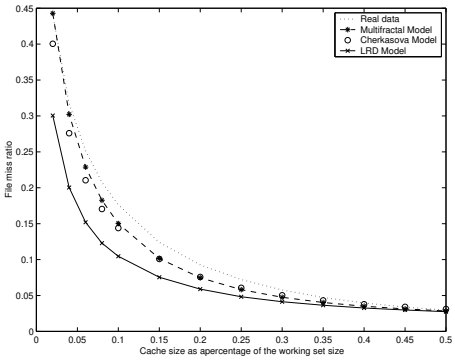

Fig. 6. File miss ratio versus cache size (CLARKNET trace).

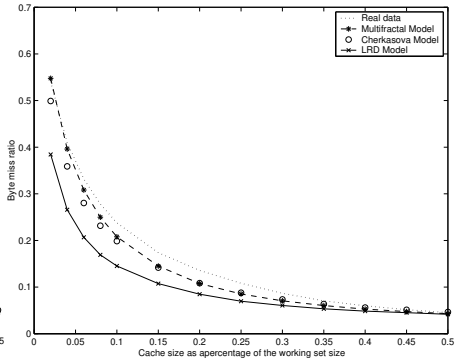

Fig. 7. Byte miss ratio versus cache size (CLARKNET trace).

inaccuracies in the file miss rate for the multifractal model, the LRD model, and Cherkasova et al.'s model are given by 0.5%, 53%, and 111%, respectively. In the case of the byte miss rate, the corresponding values are 4.9%, 65%, and 109%. The overall improvement in the accuracy of the file and byte miss rates due to the use of the multifractal model is significant.

5 Conclusions

In this work, we demonstrated the potential of multifractal processes as a viable approach for WWW traffic modeling. We started with the multifractal model of Riedi et al., which is capable of generating approximately lognormal variates with any desired autocorrelation structure. However, to apply this model in traffic fitting and trace generation, one needs to match as many parameters of the model as the length of the trace to be generated. To make the model parsimonious, we modified it by using a different distribution for the multiplier A_j (which relates the wavelet and scale coefficients) and by analytically expressing the parameter of A_j, $j = 1, 2, \ldots$, in terms of the mean, variance, and ACF of the modeled data. As a result, the modified multifractal model is specified by five parameters only. We fitted this model to the scaled stack distance strings of two WWW traffic traces. The proposed model captures the spatial and temporal localities of the real traffic as well as the popularity profile. Trace-drive simulations of the LRU cache policy indicates that our model gives much more accurate cache miss rates than two previously proposed WWW traffic models. Our future research will focus on designing new cache replacement and prefetching policies that exploit the characteristics of the traffic and that rely on model predictions in making file replacement and prefetching decisions.

References

1. Internet traffic archive at http://ita.ee.lbl.gov/.

2. V. Almeida, A. Bestavros, M. Crovella, and A. Oliverira. Characterizing reference locality in the WWW. In *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems (PDIS)*, pages 92–103, 1996.
3. M. Arlitt and C. Williamson. Web server workload characterization: The search for invariants. In *Proceedings of the ACM SIGMETRICS Conference*, pages 126–137, 1996.
4. P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of the ACM SIGMETRICS Conference*, pages 151–160, 1998.
5. P. Cao and S. Irani. Cost-aware WWW proxy caching algorithms. In *Proceedings of the 1997 USENIX Symposium on Internet Technology and System*, pages 193–206, 1997.
6. L. Cherkasova and G. Ciardo. Characterizating temporal locality and its impact on web server performance. In *Proceedings of the Ninth International Conference on Computer Communication and Networks (ICCCN)*, pages 434–441, 2000.
7. D. Cox. Long-range dependence: A review. *Statistics: An Appraisal*, pages 55–74, 1984. The Iowa State University, Ames, Iowa.
8. C. Cunha, A. Bestavros, and M. Crovella. Characteristics of WWW client-based traces. *IEEE Transactions on Networking*, 1(3):134–233, Jan 1999.
9. A. Feldman, A. Gilbert, W. Willinger, and T. Kurtz. The changing nature of network traffic: Scaling phenomena. *Communication Review*, April 1998.
10. A. Gilbert, W. Willinger, and A. Feldmann. Scaling analysis of conservative cascades, with applications to network traffic. *Special Issues of IEEE Transactions on Information Theory on Multiscale Statistical Signal analysis and its Applications*, 1999.
11. A. Gillbert and W. Willinger. Data networks as cascades: Investigating the multifractal of internet WAN traffic. *IEEE Transactions on Information Theory*, pages 971–991, 1999.
12. C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kays. Modeling and simulation of self-similar variable bit rate compressed video: A unified approach. In *Proceedings of the ACM SIGCOM Conference*, pages 114–125, 1995.
13. S. Jin and A. Bestavros. Popularity-aware greedy-dual size web proxy caching algorithms. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, Taiwan, May 2000.
14. S. Jin and A. Bestavros. Sources and characteristics of web temporal locality. In *Proceedings of IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, San Fransisco, CA, August 2000.
15. S. Jin and A. Bestavros. Temporal locality in web request streams. In *Proceedings of the ACM SIGMETRICS Conference*, pages 110–111, 2000.
16. S. Jin and A. Bestavros. Greedy-dual* web caching algorithm. *International Journal on Computer Communications*, 24(2):174–183, Febreuary 2001.
17. R. Riedi. Introduction to multifractals. http://www.dsp.rice.edu/publications/.
18. R. Riedi, M. Crouse, V. Ribeiro, and R. Baraniuk. A multifractal wavelet model with application to network traffic. *IEEE Transactions on Information Theory*, 45(3):992–1018, April 1999.