An Approach for Bursty and Self-similar Workload Generation

Xingjian Lu, Jianwei Yin, Hanwei Chen, and Xinkui Zhao

College of Computer Science and Technology,
Zhejiang University. Hangzhou, China
{zjulxj,zjuyjw,chw,zhaoxinkui}@zju.edu.cn

Abstract. As two of the most important characteristics of Web systems' workloads, burstiness and self-similarity are gaining more and more attentions. And synthetically generating bursty and self-similar workloads is a key technique for Web system performance analysis. In this paper, a configurable synthetic approach for bursty and self-similar workload generation has been proposed based on a superposition of 2-state Markovian arrival processes (MAP2). This method can generate workload with both specified intension of burstiness and self-similarity. The detailed evaluation show the accuracy and robustness of our method.

Keywords: Workload Generation, Markovian, Burstiness, Self-similarity.

1 Introduction

In recent years, more and more Web-based systems have been moved to cloud computing platforms such as Amazon EC2 and Google App Engine, which can promise of on-demand resource provisioning based on virtualization techniques. And some characteristics such as burstiness of workloads can have critical impact on resource provisioning strategies and performance of cloud platforms. For example, flash-crowd service requests can cause resource allocation problems and seriously degrade system performance [1]; Simultaneously launching jobs for different cloud applications, which are no longer single-program-single-execution applications, during a short time period can immediately aggravate resource competitions and load unbalancing among computing sites [16]. So synthetically generating bursty workloads is an important technique for Web system performance analysis, especially in the context of cloud computing.

Burstiness, which means highly variable request arrival rate or service time, has been observed in Ethernet LAN, Web applications, storage systems [15] and grid systems [10]. Many mathematical methods, including peakedness, peak-to-mean ratio, coefficient of variation, and indices of dispersion for count (IDC) have been proposed to characterize the intension of burstiness. The Markovian Arrival Process (MAP) [5,13], which is a generalization of Markov Modulated Poisson Process (MMPP), is usually leveraged to model bursty request arrivals. And some workload generators such as SWAT [8] and Geist [7] can also support the bursty workload generation.

X. Lin et al. (Eds.): WISE 2013, Part II, LNCS 8181, pp. 347–360, 2013.

However, these methods only focus on the burstiness at some specific time-scale, while self-similarity, which has been also observed in a variety of working communication networks and computing systems, presents a process displaying similar-looking workload burstiness over all or a wide range of time-scales. The intension of self-similarity is often characterized by the Hurst parameter. And some new models, such as chaotic maps [11], fractional brownian motion (FBM) [17] and fractional autoregressive integrated moving average (FARIMA) model [9], have been proposed to describe self-similar behavior in a relatively simple manner. Also, a number of self-similarity models have been developed based on traditional traffic models. Similarly, these methods merely focus on modeling and fitting self-similarity, none of them can synthetically generate workloads that exhibit both specified burstiness and self-similarity degree.

In order to deal with these deficiencies, a markovian approach for bursty and self-similar workload generation has been proposed in this paper based on a superposition of 2-state Markovian arrival processes (MAP2). Our approach can leverage some simple traffic parameters, which can be straightforwardly derived from real system logs or provided by performance analysts, to compose a MAP with both required intension of bursiness and self-similarity. The detailed analysis and experiment results show the accuracy and robustness of our method.

The remainder of this paper is organized as follows. Section 2 introduces the motivation. Section 3 describes the Markovian approach for bursty and self-similar workload generation. Section 4 evaluates our workload generation method by conducting a detailed accuracy and robustness analysis. Finally, section 5 concludes and describes the future work.

2 Motivation

For workload analysis, the IDC has been widely used to characterize the burstiness of arrival. This is a standard burstiness index first used in networking, and then applied to model workload burstiness in Multi-Tier applications [12]. The IDC at time t is the variance of the number of requests arrived in an interval of length t divided by the mean number of requests arrived in this interval:

$$I_t = \frac{Var(N_t)}{E(N_t)} \tag{1}$$

where N_t represents the number of arrivals in the continuous interval of $(0, t)$.

Traditional workload generators such as Surge [4] and Httperf [14], can not support burstiness generation. Then, SWAT [8], Geist [7] and the method proposed in [13] were developed to provide mechanisms for burstiness injection. Although these methods can support injecting burstiness into workloads, the resulting models, based on burstiness characterizations using IDC, are adequate only over a limited range of time-scales. No one can synthetically generate workload with specified long range bursty behavior across large time-scales, i.e. the self-similarity.

Let the discrete-time stochastic process $X = \{X_i, i = 0, 1, ...\}$ is used to describe the number of arrivals in the i-th interval (length is Δ). And the aggregated process of X is defined as follows:

$$X^{(m)} = \{X_i^{(m)}\} = \{\frac{X_1 + ... + X_m}{m}, ..., \frac{X_{mk+1} + ... + X_{(m+1)k}}{m}, ...\}$$

Then X is called exactly second-order self-similar with the Hurst parameter $H = 1 - \beta/2$ if

$$Var(X^{(m)}) = \sigma^2 m^{-\beta}. \tag{2}$$

where σ^2 is the variance of X, m is the aggregate level. There are some other equivalent definitions of self-similarity, we mainly consider the one relates to IDC in this paper. That is if X satisfies the following formula, X is self-similar.

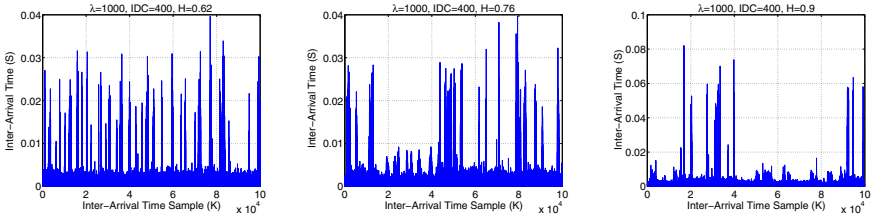
$$I_m = I_{(t=m\Delta)} = \frac{Var(N_{(m\Delta)})}{E(N_{(m\Delta)})} = I_1 m^{2H-1}. \tag{3}$$

where I_1 denotes the IDC of the arrival process at the unit interval Δ.

Some new models such as chaotic maps, FBM and FAIMA have been developed to describe and model the self-similar behavior, and the corresponding approaches are also developed to generate self-similar traffic or workload based on these models. Because the queueing theoretical techniques are hardly to be used for these new models, a number of self-similarity models have been developed based on traditional traffic models too. For instance, MMPP as a superposition of 2-state Markov processes, is used to emulate self-similarity over a certain range of time scales in [2,3,18]. In [6], markovian arrival process as a superposition of a phase type renewal process and an interrupted Poisson Process (IPP), is proposed to approximate real traffic behavior. However, all these methods proposed to generate self-similar workload are only dedicated to the long range bursty behavior across large time-scales, they can't be used to generate self-similar workloads with specified burstiness on certain time-scales.

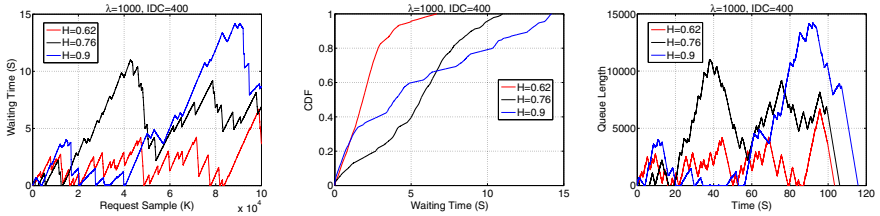
Motivated by the fact that current workload generation methods only focus on either burstiness or self-similarity, we claim a complete and practical workload generator should support workload generation with specified intension of burstiness and self-similarity. Then two questions may come into being:

First, why should we consider both the burstiness and self-similarity during workload generation. Here we use a real case to describe the reasons in the following. Three workloads with identical burstiness profiles ($IDC = 400$) and different intensions of self-similarity ($H = 0.62, 0.76, 0.9$ separately) are described in Fig. 1. For each plot of this figure, there are 10^5 inter-arrival time samples, whose mean value is 0.001 seconds. If single burstiness is enough to describe the bursty characteristic of the workload, the performance of these three workloads should be identical or nearly identical when they are imposed to the same system with identical environment. In order to verify this claim, we show the queueing performance of these workloads with $\cdot/D/1$ queueing network simulation in Fig. 2. The constant service time for each request is set to 0.001 seconds.



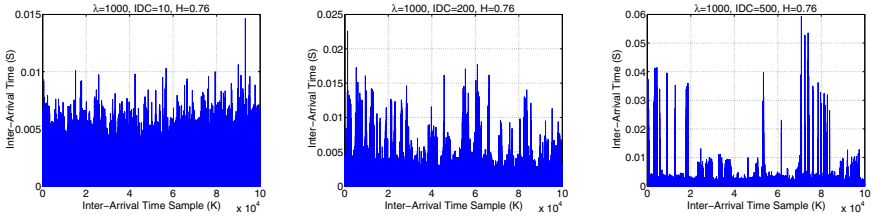
(a) (b) (c)

Fig. 1. Three workloads with identical burstiness ($IDC = 400$) but different self-similarity ($H = 0.62, 0.76, 0.9$) separately



(a) (b) (c)

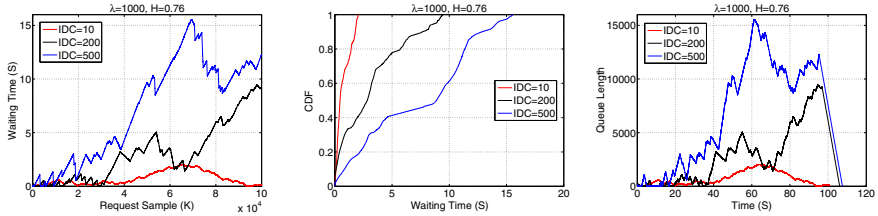
Fig. 2. Performance of the workloads depicted in Fig. 1 with $\cdot/D/1$ queue



(a) (b) (c)

Fig. 3. Three workloads with identical self-similarity $H = 0.76$ but different burstiness ($IDC = 10, 200, 400$) separately

As shown in Fig. 2(a), the waiting time of each request when $H = 0.62$ is much smaller than $H = 0.76$ and $H = 0.9$. Though the waiting time when $H = 0.76$ is larger than $H = 0.9$ for some requests, the average and maximum value when $H = 0.9$ is larger than $H = 0.76$. This observation is validated by Fig. 2(b) and Fig. 2(c). From Fig. 2(b), we can see the waiting time for 80% of the requests is 2.72, 7.26 and 10.05 separately for the corresponding workload ($H = 0.62, 0.76, 0.9$). The queue length curve plotted in Fig. 2(c) also show evident performance differentiation when H is assigned to different values, even the IDC is identical. So we can see it's apparently inaccurate to describe the bursty characteristic of the workload merely by the burstiness parameter IDC.



(a) (b) (c)

Fig. 4. Performance of the workloads depicted in Fig. 3 with $\cdot/D/1$ queue

Similarly, Fig. 3 shows the inter-arrival time samples for another three workloads with identical self-similarity ($H = 0.76$) but different burstiness ($IDC = 10, 200, 500$ separately), while Fig. 4 shows the corresponding performance for these workloads with the same $\cdot/D/1$ queueing network. As shown in Fig. 4, even with identical self-similarity intension, the performance is still significantly different for different value of IDC. And the higher the IDC the worse the performance. So single self-similarity is also not adequate to model the bursty behavior of practical workload. That means combining the burstiness and self-similarity is more completed and practical to generate workloads.

And the second question is how can we combine the burstiness and self-similarity. From previous description we know MMPP2 is often used to model workload burstiness, and some superpositions of Markov processes can be used to generate self-similar workload. So the natural way we may consider is to develop a method to generate the bursty and self-similar workload based on Markovian models, and with which the queueing theoretic techniques developed in the past can be used to guide the performance evaluation.

In a word, motivated by current workload generation methods usually focus on single burstiness or self-similarity, we aim to seek for a completed and practical approach for bursty and self-similar workload generation. Considering the computational tractability and the convenience for performance evaluation based on queuing theoretic techniques, the proposed approach for workload generation is based on Markovian models.

3 Markovian Modeling for Bursty and Self-similar Workload Generation

The proposed Markovian approach for bursty and self-similar workload generation is based on the model by Anderson et al. [2,3], where workload is modeled by the superposition of several 2-state MMPPs. The benefits of using a Markov model is that it is possible to re-use the well-known queuing theoretical techniques developed before and a whole array of tools for calculating performance measures is already available.

3.1 Superposition of Two State Markovian Sources

In this subsection, some main characteristics of MMPP will be summarized first. In the case of m-state MMPP, the underlying Markov process can switch among m Poisson processes, each of which has a unique request arrival rate $\lambda_i, (1 \leq i \leq m)$. That is, the arrival rate is λ_i when the Markov chain is in state i. In the 2-state case, two square matrices Q and Λ are used to define a MMPP2 from a client's point of view.

$$\mathbf{Q} = \begin{bmatrix} -r_1 & r_1 \\ r_2 & -r_2 \end{bmatrix}. \qquad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

For the case of MMPP2, the mean value of N_t is given by

$$E(N_t) = \frac{r_2\lambda_1 + r_1\lambda_2}{r_1 + r_2}t. \tag{4}$$

And the variance of N_t is can be calculated as follows:

$$Var(N_t) = \frac{r_2\lambda_1 + r_1\lambda_2}{r_1 + r_2}t + 2A_1t - \frac{2A_1}{r_1 + r_2}(1 - e^{-(r_1+r_2)t})$$

where $A_1 = \frac{r_1 r_2 (\lambda_1 - \lambda_2)^2}{(r_1+r_2)^3}$.

Since any MMPP obtained by superposing several MMPP2s can be described by a superposition of several interrupted Poisson processes (IPP) and one Poisson process. We consider the required MMPP is composed of $d(> 1)$ IPPs and one Poisson process. ith IPP can be give by

$$\mathbf{Q}_i = \begin{bmatrix} -r_{1i} & r_{1i} \\ r_{2i} & -r_{2i} \end{bmatrix}, \qquad \mathbf{\Lambda}_i = \begin{bmatrix} r_i & 0 \\ 0 & 0 \end{bmatrix}.$$

The superposition can be described as follows

$$\mathbf{Q} = \mathbf{Q}_1 \bigoplus \mathbf{Q}_2 \bigoplus \ldots \bigoplus \mathbf{Q}_d$$
$$\mathbf{\Lambda} = \mathbf{\Lambda}_1 \bigoplus \mathbf{\Lambda}_2 \bigoplus \ldots \bigoplus \mathbf{\Lambda}_d \bigoplus \lambda_p,$$

where $\bigoplus$ is the Kronecker's sum and λ_p means the arrival rate of the Poisson process. The whole arrival rate of the superposition process λ can be given by

$$\lambda = \lambda_p + \sum_{i=1}^{d} \frac{r_{2i}\lambda_i}{r_{1i} + r_{2i}} \tag{5}$$

In the next subsection, we show how to determine the parameters of the IPPs and the Poisson process.

Table 1. Preliminarily Required Parameters

Parameter	Meaning
λ	Average arrival rate of the whole process.
m_{min}, m_{max}	Minimum and Maximum of the time-scales over which self-similarity is taken into consideration.
$I_{m_{min}}$	The IDC value at the minimum time-scale.
H	Hurst parameter.
d	Number of IPPs.

3.2 Applied Parameterizing Algorithm

In this subsection, a procedure is given to determine the parameters of the IPPs and the Poisson procss to construct a MMPP such that the properties of the workload generated by our approach match predefined values. Table 1 shows the preliminary required parameters for our generation model.

Let $N_{t|i}$ and $N_{t|p}$ be the number of arrivals during the t-th time slot in the i-th IPP and the Poisson process separately, and let $N^m_{t|i}$ and $N^m_{t|p}$ be the corresponding aggregated processes of them. Considering the computational tractability, we assume the r_1 and r_2 satisfy the following relation, for each IPP.

$$f = \frac{r_{2i}}{r_{1i} + r_{2i}}, \qquad (1 \leq i \leq d) \tag{6}$$

Then using (5), we have

$$\lambda = \lambda_p + \sum_{i=1}^{d} f\lambda_i \tag{7}$$

and using (4), we obtain the variance of the i-th IPP as

$$Var(N_{t|i}) = f\lambda_i t + \frac{2(1-f)^2\lambda^2 t}{r_{1i}f} - \frac{2(1-f)^3\lambda_i^2}{fr_{1i}^2}(1 - e^{-\frac{r_{1i}}{1-f}t}) \tag{8}$$

The variance of aggregated arrival process $N^{(m)}_{t|i}$ can be expressed as

$$Var(N^{(m)}_{t|i}) = \frac{Var(N_{(m\Delta)|i})}{(m\Delta)^2} \tag{9}$$

where Δ is previous mentioned sampling resolution. Here we consider Δ one time unit, using (8) and (9), we can get

$$Var(N^{(m)}_{t|i}) = \frac{f\lambda_i}{m} + 2f(1-f)^2\eta_i\lambda_i^2 \tag{10}$$

where

$$\eta_i = \frac{1}{mr_{1i}} - \frac{1-f}{m^2r_{1i}^2}(1 - e^{\frac{1}{f-1}mr_{1i}}) \tag{11}$$

The corresponding variance of the Poisson process is λ_p/m. For independent subprocesses, the variance of the superposition equals the sum of individual variances, so the variance of the whole process is given by

$$Var(X_t^{(m)}) = \frac{\lambda_p}{m} + \Sigma_{i=1}^{d} Var(N_{t|i}^{m}) = \frac{\lambda}{m} + 2f(1-f)^2 \sum_{i=1}^{d} \eta_i \lambda_{1i}^2 \tag{12}$$

where we used (7). Then using (12) and (1), we can get

$$I_m = \frac{Var(N_{(m\Delta)})}{E(N_{(m\Delta)})} = \frac{m^2 Var(X_t^{(m)})}{m\lambda} = 2f + \frac{2mf(1-f)^2}{\lambda} \sum_{i=1}^{d} \eta_i \lambda_{1i}^2 \tag{13}$$

Since the superposition of d IPPs and a Poisson process is expected to show self-similarity over d different time-scaless, and the sojourn time of each IPP is in accordance with the different time-scales, so there are d different points $m_i (1 \leq i \leq d)$. According to the range of time-scales specified by the input parameters, we have $m_{min} \leq m_i \leq m_{max}$, let

$$m_i = m_{min} a^{i-1} \tag{14}$$

where

$$a = (\frac{m_{max}}{m_{min}})^{\frac{1}{d-1}}, \quad d > 1. \tag{15}$$

In order to reduce the number of parameters which have to be determined, we also assume $m_i r_{1i} = 1$, i.e.

$$r_{1i} = \frac{1}{m_i}, \quad (1 \leq i \leq d). \tag{16}$$

Then using (6), (14)-(16), we can obtain r_{2i} for each IPP. Now the parameters we need to obtain are only f and λ_i, since λ_p can be derived from (7) if λ_i is determined. Based on the above analysis, the applied parameterizing algorithm is in the following:

– **SETP1. Determine λ_i as the function of f.** From (4) and (14), we have

$$I_1 \begin{bmatrix} m_1^{(2H-1)} \\ m_2^{(2H-1)} \\ \vdots \\ m_d^{(2H-1)} \end{bmatrix} = 1 + \mathbf{B} \begin{bmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \vdots \\ \lambda_d^2 \end{bmatrix} \tag{17}$$

where $\mathbf{B}$ is the $d \times d$ matrix whose (i, j) element is

$$\mathbf{B}_{ij} = \frac{2f(1-f)^2}{r_{1i}\lambda} - \frac{2f(1-f)^3}{m_i r_{1i}^2 \lambda}(1 - e^{\frac{m_i r_{1i}}{f-1}}) \tag{18}$$

Solving this, we can determine λ_i as the function of f.

– **STEP2. Find the value of f heuristically.** First, find the range of f heuristically, and set an initial value for f and the largest number of iterations. Then, use **STEP1** to obtain λ_i and further other needed parameters to determine the MMPP. Next, use this model to generate specified number of the inter-arrival time sample. Then we calculate the value of the average arrival rate, IDC and H from the generated sample data, to obtain the combined error. The value of f that minimizes the combined error is selected as the final value of f. Then other parameters can also be determined to generate the required workload.

To conclude, we compare our method with that of [2] and [18] in Table 2. Here, we call the procedure of [2] covariance method, the procedure of [18] variance method, and ours IDC method. The generation procedure of our method and the variance method are exactly constructed while that of [2] contains some approximations. Furthermore, the variance method does not hold when $Var(N_t^{(m)}) \leq \lambda/m$ or $Var(N_t^{(m)}) \geq \lambda/m + \lambda^2$, while our method does not have this constraint. This is significant to workload generator, which not only needs to fit the original trace, but also need to allow the generation of workload with desired characteristics which may cover a large different range.

Table 2. Comparison between IDC, Variance, and Covariance Methods

	IDC	Variance	Covariance
Required Parameters	$\lambda, H, I_1, d,$ Time scale	$\lambda, H, d, \sigma^2,$ Time scale	$\lambda, H, d, r(1),$ Time scale
Type of Component MMPPs	IPP	IPP	SPP
Parameter Fitting	Exact	Exact	Approximation
Constraint	None	$\frac{\lambda}{m} < Var(N_t^{(m)}) < \frac{\lambda}{m} + \lambda^2$	None

4 Evaluation

Accurately and robustly generating required workloads is the most important criteria to evaluate a workload generator. Thus in this section, we mainly evaluate the accuracy and robustness of our bursty and self-similar workload generation approach with the notion of average deviation, which is the relative error between the derived indicator parameter values with the expected ones. And it can be calculated as follows:

$$Avg\_Dev = \frac{1}{n} * \sum_{i=1}^{n} \frac{Dev(X_i) - Expec(X_i)}{Expec(X_i)} \tag{19}$$

where $Dev(X_i)$ and $Expec(X_i)$ denotes the derived and expected value of λ, IDC or H during i-th execution. For each indicator parameter, we execute the generation approach $n = 100$ times to derive the average value of deviations.

4.1 Accuracy Analysis

In this subsection, we evaluate the accuracy of generating workloads with specified intension of IDC and H. During these experiments, we experimentally set the expected average arrival rate $\lambda = 1000$, the number of IPPs $d = 4$, the burstiness $IDC = 10, 50, 100, 200, 400, 500$ (in practice, the maximum value 500 is enough to present the typical large value of IDC when $\lambda = 1000$), the self-similarity $H = 0.55, 0.62, 0.69, 0.76, 0.83, 0.9, 0.97$, and the minimum and maximum time-scale is 1 and 10^4 separately. By changing the value of IDC and H, we can derive the generating accuracy of our approach under different intension of burstiness and self-similarity. For giving an intuitive presentation of the generated workload by our approach and describing the motivation of this paper, we plot one set of the inter-arrival time samples with identical burstiness and different self-similarity in Fig.1, while inter-arrival time samples with identical self-similarity but different burstiness in Fig.3. And the corresponding queueing performance of these samples is depicted in Fig.2 and Fig.4 separately.

In table 3, we describe the average deviation of λ for each composition of IDC and H. From this table, we can see the deviation of λ is low, even when $IDC = 500$ and $H = 0.97$ the value is only 6.63%. And the tendency is evident that the deviation of λ increases with IDC (or H) when the value of H (or IDC) is identical. That is the higher the intension of burstiness or self-similarity the lower the accuracy of our method. The main reason for this behavior is that higher intension of burstiness or self-similarity means more variability of the inter-arrival times, which often brings more difficulty in accurately estimating the mean value, so the resulted average arrival rate may have a larger deviation.

The average deviation of IDC is described in table 4, generally the value of these deviations is larger than the ones of λ, since the calculation of IDC is more complex and inaccurate than the mean value of average arrival rate. It is also obvious that the deviation of IDC increases with the expected value of IDC and H. For instance, the deviation of IDC is only 0.75% when $IDC = 10$ and $H = 0.55$, while the value of deviation reaches 10.06% when $IDC = 500$ and $H = 0.97$. The reason is similar to the one of the deviation of λ.

However, the average deviation of H, as shown in table 5, shows a different tendency compared with the one of λ and IDC. First, the deviation of H doesn't show a complete increasing or decreasing tendency during the entire range of H. It decreases initially and then increases with the value of H. The main reason can be explained as follows: Our approach is developed based on the assumption that the required workload is self-similar, it doesn't work well under the case of no or low intension of self-similarity. Thus the lower the value of H the higher the inaccuracy to generate self-similar inter-arrival time samples, and further the higher the inaccuracy to derive the expected value of H. Furthermore, when the value of H closes to the maximum value 1, the relative error to get the required parameters of the MMPP model is larger than the low or moderate H, so the deviation of H begins to increase again after the minimum value. Second, although the deviation of H show an increasing tendency with the value of IDC, the increasing rate is not identical for different intensions of self-similarity. From

table 5, we can see the increasing rate when H closes to the extreme value (1/2 or 1) is much larger than the one when H is moderate. That means the value of IDC plays less influences on the deviation of H when the self-similarity is moderate. The reason is also due to the extreme values of H make the fitting method more inaccurate during the workload generation process.

From above analysis, we can see our bursty and self-similar workload generation approach can ensure the accuracy within a reasonable range ($< 10\%$) for a wide range of specified intension of IDC and H.

Table 3. Average deviation of average arrival rate λ (%)

IDC	H						
	0.55	0.62	0.69	0.76	0.83	0.90	0.97
10	0.11	0.13	0.18	0.23	0.44	0.51	0.91
50	0.20	0.32	0.53	0.51	0.93	1.01	2.12
100	0.35	0.51	0.62	0.79	0.80	1.61	3.07
200	0.49	0.54	1.10	1.23	1.73	2.34	4.82
400	0.85	0.92	1.56	2.37	3.15	4.26	5.89
500	1.54	2.03	2.98	3.52	4.36	5.49	6.63

Table 4. Average deviation of burstiness intension IDC (%)

IDC	H						
	0.55	0.62	0.69	0.76	0.83	0.90	0.97
10	0.75	0.80	0.94	1.25	1.97	3.51	5.81
50	0.67	0.71	1.06	1.37	2.33	3.63	6.32
100	0.59	0.71	1.11	1.45	2.23	3.58	6.85
200	0.84	0.95	1.47	1.55	2.56	3.79	7.94
400	1.11	1.56	2.16	2.79	3.25	4.56	8.60
500	1.94	2.23	2.94	3.44	4.32	5.07	10.06

4.2 Robustness Analysis

For a robust workload generation approach, it is not only to ensure the accuracy for different input parameters of the generation model, but also required to make sure the number of samples won't influence the generation accuracy. In order to evaluate the impact of the number of samples on the accuracy, we test the deviation of λ, IDC and H with different number of generation samples. During these experiments, we change the number of samples from 10^5 to 10^6 with the interval of 10^5, and for each of which, we generate the specified number of inter-arrival samples 100 times for different compositions of the value of IDC and H. And other parameters are also set $\lambda = 1000$, $d = 4$, the minimum and maximum time-scale 1 and 10^4 separately.

Table 5. Average deviation of self-similarity intension H (%)

IDC	H						
	0.55	0.62	0.69	0.76	0.83	0.90	0.97
10	2.36	0.95	0.81	0.90	0.87	0.91	0.96
50	2.71	1.05	0.85	0.87	0.84	0.89	1.04
100	2.83	1.38	0.93	0.92	0.86	0.91	1.16
200	3.09	1.17	0.84	0.95	1.08	0.115	1.32
400	3.42	1.32	0.89	0.93	1.32	1.28	1.67
500	3.61	1.43	0.95	0.97	1.41	1.35	1.95

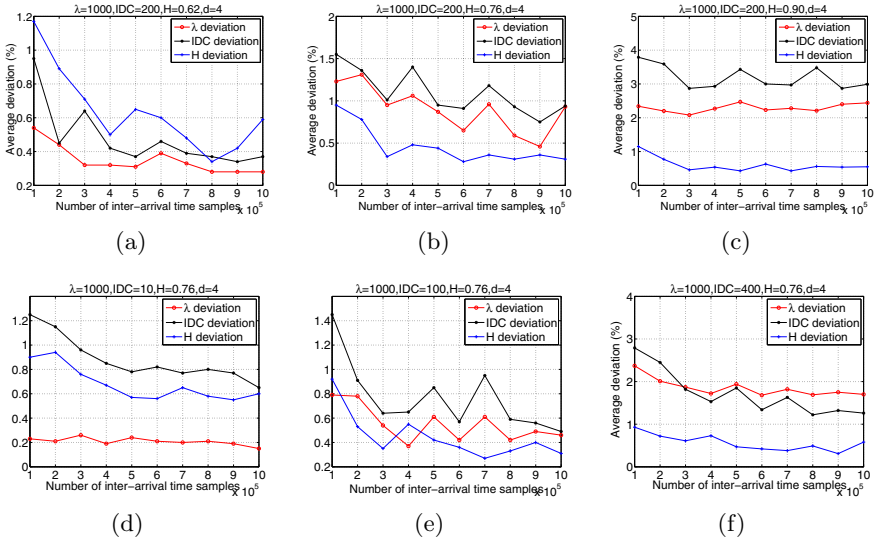


Fig. 5. Accuracy analysis with different number of samples for different compositions of the value of IDC and H

During these experiments, the deviation of λ, IDC and H show little fluctuation with the number of samples. The deviations roughly stay around a constant value when the number of samples exceed 2×10^5 or 3×10^5. We plot the deviations in Fig.5 with six typical compositions of IDC and H. The first three with identical value of $IDC = 200$ and different value of H, while the last three with identical value of $H = 0.76$ but different value of IDC. As shown by these plots, the deviations generally vary a little even though the higher the value of H or IDC, the larger the average value of these deviations. Furthermore, the deviations show significant improvement when the number of samples start to increase initially, while then the improvement begins to ease up until reaching around a constant value. The main reason for this kind of behavior lies in that when the number of samples is very small (e.g. 10^5), there maybe no enough data samples to fitting the expected value of λ, IDC and H. Thus the deviations decrease greatly when the number of samples start to increase initially. However,

once the number of samples is large enough to fit the required parameters, the deviations begin to stay around a constant value, even though the number of samples is still keep increasing.

The results in above experiments show strong robustness of our approach. Even with extremely large number of samples to be generated, our approach can still ensure the accuracy of the required parameters within a reasonable range. This property is meaningful to the practical Web system workload generation, especially in the context of cloud computing, in which the large scale of system architecture and users often require a large number of workload samples to evaluate system performance or do optimal resource provision.

5 Conclusion and Future Work

Synthetical workloads modeling emerging or future applications is extremely important in the design of efficient system architecture. However, current approaches for workload generation only focus on either burstiness or self-similarity. With accurate characterization of the two key properties of the workloads by IDC and Hurst parameter separately, we developed a markovian approach for bursty and self-similar workload generation by fitting a MMPP model as a superposition of several IPPs and one Poisson process. The main contribution of the proposed approach lies in workload generation with specified intension of both burstiness and self-similarity, the simultaneous occurrence of which is the real case for cloud applications in the production. And the experiments and evaluation show the accuracy and robustness of our approach. After focusing on bursty and self-similar workload generation in this paper, our future work on this subject is mainly to evaluate the system performance under such kind of workloads, and find approaches for performance optimalization and resource efficient utilization to reduce the negative impacts of burstiness and self-similarity.

Acknowledgments. This work was supported by National Science and Technology Supporting Program of China (No. 2012BAH06F02), National Natural Science Foundation of China under Grant (No. 61272129), Research Foundation for the Doctoral Program by Ministry of Education of China (No. 20110101110066), New-Century Excellent Talents Program by Ministry of Education of China (No. NCET-12-0491), and Zhejiang Science Fund for Distinguished Young Scholars (R13F020004).

References

1. Amini, L., Jain, N., Sehgal, A., Silber, J., Verscheure, O.: Adaptive control of extreme-scale stream processing systems. In: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, p. 71. IEEE Computer Society, Washington, DC (2006)
2. Andersen, A., Nielsen, B.: An application of superpositions of two state markovian source to the modelling of self-similar behaviour. In: Proceedings IEEE INFOCOM 1997, vol. 1, pp. 196–204 (1997)

3. Andersen, A., Nielsen, B.: A markovian approach for modeling packet traffic with long-range dependence. IEEE Journal on Selected Areas in Communications 16(5), 719–732 (1998)
4. Barford, P., Crovella, M.: Generating representative web workloads for network and server performance evaluation. SIGMETRICS Perform. Eval. Rev. 26(1), 151–160 (1998)
5. Bolch, G., Greiner, S., Meer, H.D., Trivedi, K.S.: Queueing Networks and Markov Chains. Wiley-Interscience (2005)
6. Horváth, A., Rózsa, G.I., Telek, M.: A map fitting method to approximate real traffic behaviour. In: 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, p. 32 (2000)
7. Kant, K., Tewari, V., Iyer, R.K.: Geist: A web traffic generation tool. In: Field, T., Harrison, P.G., Bradley, J., Harder, U. (eds.) TOOLS 2002. LNCS, vol. 2324, pp. 227–232. Springer, Heidelberg (2002)
8. Krishnamurthy, D., Rolia, J.A., Majumdar, S.: A synthetic workload generation technique for stress testing session-based systems. IEEE Trans. Softw. Eng. 32(11), 868–882 (2006)
9. Lakehal, M.R., Ferdi, Y., Taleb-Ahmed, A.: Generation of farima (0, α, 0) sequences by recursive filtering: Testing for self-similarity. In: Electronics, Circuits, and Systems, ICECS 2009, pp. 563–566. IEEE (2009)
10. Li, H., Muskulus, M.: Analysis and modeling of job arrivals in a production grid. SIGMETRICS Perform. Eval. Rev. 34(4), 59–70 (2007)
11. Lo, S.C., Cho, H.J.: Chaos and control of discrete dynamic traffic model. Journal of the Franklin Institute 342(7), 839–851 (2005)
12. Mi, N., Casale, G., Cherkasova, L., Smirni, E.: Burstiness in multi-tier applications: Symptoms, causes, and new models. In: Issarny, V., Schantz, R. (eds.) Middleware 2008. LNCS, vol. 5346, pp. 265–286. Springer, Heidelberg (2008)
13. Mi, N., Casale, G., Cherkasova, L., Smirni, E.: Injecting realistic burstiness to a traditional client-server benchmark. In: Proceedings of the 6th International Conference on Autonomic Computing, pp. 149–158. ACM, New York (2009)
14. Mosberger, D., Jin, T.: httperf: a tool for measuring web server performance. SIGMETRICS Perform. Eval. Rev. 26(3), 31–37 (1998)
15. Riska, A., Riedel, E.: Disk drive level workload characterization. In: Proceedings of the Annual Conference on USENIX 2006 Annual Technical Conference, p. 9. SENIX Association, Berkeley (2006)
16. Tai, J., Zhang, J., Li, J., Meleis, W., Mi, N.: Ara: Adaptive resource allocation for cloud computing environments under bursty workloads. In: 30th IEEE International Performance Computing and Communications Conference (IPCCC), pp. 1–8 (2011)
17. Tan, X., Huang, Y., Jin, W.: Modeling and performance analysis of self-similar traffic based on fbm. In: IFIP International Conference on Network and Parallel Computing Workshops, NPC Workshops, pp. 543–548. IEEE (2007)
18. Yoshihara, T., Kasahara, S., Takahashi, Y.: Practical time-scale fitting of self-similar traffic with markov-modulated poisson process. Telecommunication Systems 17, 185–211 (2001)