Statistical Analysis of the Workload of a Video Hosting Server

Christoph Möbius and Waltenegus Dargie

Technical University of Dresden, 01062 Dresden, Germany
{christoph.moebius,waltenegus.dargie}@tu-dresden.de

Abstract. The amount of data hosted by Internet servers and data centers is increasing at a remarkable pace requiring more capable and more efficient servers. However, physical efficiency does not necessarily correlate with computational efficiency. In fact, independent studies reveal that Internet servers are mostly over provisioned even though additional servers are deployed each year. Understanding the characteristics of the workload of servers is an essential step to efficiently manage them. For example, from the workload statistics, it is possible to predict idle or underutilized states and to consolidate workload, so that the idle or underutilized servers can be switched off. In this paper, we systematically analyze the characteristics of video servers – since they are responsible for producing the largest Internet traffic – and provide an insight into the relationship between the statistics pertaining to workload, the size of videos, and service time. We shall show that from the distribution of the video sizes on host servers, it is possible to estimate the distribution of the workload size produced by clients and the distribution of the time required to process individual requests.

Keywords: Service time, workload characterization, workload generation, workload size, workload statistics, video server, video size

1 Introduction

The amount of data hosted, processed, and communicated by Internet-based servers and data centers is increasing at a remarkable pace. According to a recent report by Cisco Global Cloud Index[1], the global data center IP traffic will be 554 exabyte per month by 2016. In comparison, this has been 146 exabyte per month in 2011. Likewise, the global cloud IP traffic will reach at 355 exabyte per month by 2016 (from 57 exabyte per month in 2011). The corresponding magnitude of workload per installed cloud server will increase by more than twofold by 2016 compared to the workload per installed server in 2011.

The research community as well as the IT industry approaches this phenomenon in a number of ways. Two of them, and perhaps the most ubiquitous

[1] Cisco Global Cloud Index: Forecast and Methodology, 2011–2016, Cisco Inc. (www.cisco.com)

ones, involve (1) the replacement of existing servers with more capable servers and (2) the deployment of additional servers. The estimated worldwide server deployment in 2010 was 40 million units [9], but additional servers have been steadily deployed since then. The latest statement from the International Data Corporation (IDC)[2] reveals that 1.9 million server units have been shipped in the first quarter of 2013 alone. Unfortunately, these approaches alone do not ensure a sustainable computing due to the fact that a rapid growth in the number and capacity of installed servers results in an equally rapid growth in power consumption [1,14,17].

The third approach presently adopted by the industry combines server virtualization with cloud computing, so that Internet services encapsulated in virtualized "machines" can share hardware resources, but each virtual machine has its own secure and dedicated execution space. Moreover, the virtual machines can be freely migrated from one physical machine (server) to another at runtime. This feature has two advantages: Firstly, virtual machines are not bound to any specific server; their owners can change host servers whenever they wish to. Secondly, infrastructure owners can freely decide where and for how long individual virtual machines should execute, so that they can efficiently utilize hardware resources – this aspect is known in the literature as service or workload consolidation [4] as well as server consolidation [2].

Whether in a virtualized environment or otherwise, understanding the characteristics of the workload of servers is useful for efficiently managing hardware resources accomplish) [8]. Firstly, the workload of underutilized or overloaded servers can be timely migrated to servers which can be loaded optimally. Secondly, from the statistics of resource utilization, it is possible to determine whether and for how long idle servers can be switched off to save power [23,24]. Thirdly, services that consume complementary resources can be scheduled on the same machine whereas services known for competing for similar resources can be scheduled to execute on separate servers [26].

The workload of an Internet server is primarily generated by users issuing requests. Hence, it consists of two independent quantities which cannot be known in advance except in a probabilistic sense. The first quantity refers to the arrival pattern of the requests (request arrival rate) while the second refers to the size of each request or the computational complexity each request induces on the server. Most existing probabilistic models for managing the resources and predicting the performance of Internet services rely on these two quantities.

In this paper, we shall experimentally demonstrate that for online video hosting services, such as Metacafe and YouTube, the statistics of the workload as well as the time needed to serve individual requests can be sufficiently determined from the statistics of the videos they host. The justification for our assertion is that for a large number of videos, there is a strong correlation between the preference of the users who generate videos and the users who view these videos. This knowledge is useful because service providers can estimate (1) the amount

[2] http://www.idc.com/getdoc.jsp?containerId=prUS24136113 (last visited August 20, 2013).

of resources they should make available to accommodate user requests and (2) the quality of service they can achieve for a given resource configuration. In other words, service providers need only examine how the statistics of the videos they host change overtime to balance the supply of resources (for example, the leasing of network bandwidth or storage) with the demand for resources and to make a desirable trade-off between performance and resource consumption (including power). Since the required statistics is always available to them on the server's side, they can make decisions without the influence of external entities.

The rest of this paper is structured as follows: In section 2, we analyze related work. In section 3, we describe our experimental setting and how we generate workload. In section 4, we analyze our measurement data and discuss our observation. Finally, in section 5, we provide summary and conclusion.

2 Related Work

The term *workload* is understood in the literature in one of the following two ways: In the first, it refers to the magnitude of client requests processed by an Internet server[3] [30,19,6,10,18]; and in the second, it refers to the magnitude of utilization of hardware resources (such as CPU and memory) [13,21,4,22,28]. The main difference lies in the quantity that is available for modeling and analyzing the characteristic of a service. In this paper, we adopt the first association.

Regardless of the way a workload is understood, obtaining sufficient statistics to accurately model and analyze Internet services is a difficult task because of privacy concerns and business secrecy. In the past, researchers have tried to piece together several parameters that can characterize the workload of web servers, particularly, video hosting applications. Some have made use of publicly available data, such as traces of CPU utilization of real-world web servers, so as to model and reason about similar web servers running on different platforms [15], [28]. Others have employed either web crawling to obtain metadata of files hosted by Internet services or filtered and analyzed Internet packets destined to or arriving from hosting sites at particular gateways. Evidently, all of these approaches can only provide partial views of the real workloads.

Barford et al. [3] identify seven statistical properties that characterize (conventional) HTTP traffic. These are the probability distribution of file sizes at the server side, the file popularity, temporal locality, the request sizes, active OFF times, inactive OFF times, and the number of embedded references. They assert that file sizes have heavy-tailed probability density functions and (client-sided) request sizes can be determined by (server-sided) file sizes.

Tang et al. [25] analyze the traffic of a media streaming server. Their model builds on the idea of Barford et al. but relaxes the assumption that file popularity is statistically stationary. Instead they define a life-span distribution to account for a file popularity that changes over time. Moreover, they determine two types

[3] As long as the context is clear, we use the terms *service* and *server* interchangeably. We use the term *physical machine* when we wish to put the emphasis on the hardware server.

of life-spans: a regular life-span following a log-normal distribution and a news-like life-span following a Pareto distribution. The parameters for both types of life-span distributions are normally distributed. Their approach is the only approach known to us which examines prefix durations (i.e., aborted sessions).

Gill et al. [11] investigate the traffic of YouTube at a campus network. The central finding of their work is understanding the relationship between file types and traffic size: Whereas only 3% of all requests were for video files, 98.6% of the traffic was caused by video files. The majority of requests, i.e. 86%, were for images and text files which account for less than 1% of all traffic. The remaining 11% of requests were for applications and script data which account for 0.5% of all traffic. In addition, the study reveals that video file sizes are not considerably variable and therefore, cannot be modeled as long-tailed random variables. This is most likely due to the 10-minutes duration restriction for videos existing at the time which has been increased to 15 minutes as of July 2010. Today it is possible to upload videos larger than 20 GB as a result of which the probability density function of video size can be expected to be heavy-tailed.

Similarly, Cheng et al. [7], Cha et al. [5] and Mitra et al. [16] analyze the traffic of several video hosting applications. One of the observations common to all is that the popularity of a video does not follow a purely Zipf function. Instead, it exhibits a cutoff at the lower end. In other words, less popular videos still receive more views than assumed by a purely Zipf function. Like Tang et al. these researchers emphasize the need to capture a change in file popularity (life span). The analysis of Cheng et al. reveals that the life span of a video follows a Pareto density function. The most important parameter for the life-span is the growth trend factor, p. A value of $p > 1$ indicates a rise in popularity while $0 < p < 1$ indicates a decline in popularity. According to Cheng et al. in 70% of all the videos they considered, $p < 1$. Based on this observation, Cheng et al. propose a model to predict the amount of additional views a video receives in future, which takes video age, current popularity, and p as its input parameters.

Cha et al. [5] analyze the popularity patterns of videos in YouTube, Daum, and Lovefilm. They find no correlation between video length and video popularity. The popularity follows a power law with an exponential cutoff, an observation confirmed by Mitra et al. [16]. Gummadi et al. [12] attribute this cutoff to a post-filtering process by recommendation systems. According to Cha et al. 99% of all videos of the video hosting applications are shorter than 10 minutes (which, again, is most probably influenced by the then existing 10-minute upload limit).

Finally, the investigation of Cha et al. reveals that the share of workload generated by users' activities (ratings and comments) is almost negligible: For YouTube only 0.22% of all views result in a rating and only 0.16% of all views result in a comment. This observation agrees with an earlier observation [11]. Similar observations are made by Mitra et al. – The workload due to ratings, comments, and uploads is typically several orders of magnitudes less than the workload generated by video views.

In the following sections, we build on the ideas and concepts discussed in this section to generate realistic workloads for a video hosting server.

3 Workload Generation

Whether running on a privately owned server or on a leased public cloud platform, understanding the workload of a server is vital for planning and managing resources (for example, resource-efficient schedulers employ workload statistics to determine where a given request should be processed). We assert that some of the statistics of a workload can be established from the statistics of the files the server hosts. We shall analytically as well as experimentally illustrate the correctness of this assertion by taking a video hosting server as an example. We assume that the videos hosted by the server are generated and viewed by users who are independent of the service management.

Our server consists of eight quad core Intel Xeon E5-4603 processors, 16 GB memory, 10 Gbps Intel NIC, and an XFS-formatted 6 TB hard disk structure with a theoretical sustained data rate of $\sim$465 MB/s (the data rate reduces to $\sim$300 MB/s during heavy contention). The server hosts 5000 videos of different sizes and an Apache application server streams users' download requests.

The first step towards examining our assertion is to generate a realistic workload and to feed this workload to the video hosting server. We combine together the different models we reviewed in section 2 to generate the workload. As can be recalled, the models are developed by independent researchers who had access to actual Internet workloads (they employ traces or web crawlers). These models refer to: (1) The distribution of file sizes on the host server, (2) the file popularity at the start of the experiment, (3) the popularity growth factor, (4) the age of the files, and (5) the distribution of weekly views. We explain these features in more detail in the subsequent subsections and report how they relate to the workload of our server.

3.1 Video Size

As we already mentioned earlier, determining the distribution of the sizes of files in existing video hosting platforms is difficult for lack of access to the actual servers and because file sizes are not parts of publicly available meta data. In addition, most sites do not allow web crawling. However, early analysis of Internet traffic shows that the density of data size exhibits a heavy-tailed density function [20,29,3]. Studies contending this assertion (for example, [11,7]) often refer to restrictions made by service providers on the size of videos that can be uploaded on their servers. For example, YouTube currently limits the uploaded video duration to 15 minutes for most users, but for users with *good conduct record* this limit is pushed to 12 hours[4]. Likewise, Vimeo currently allows uploads of up to 5 GB for standard users and up to 25 GB per upload for Pro users. Even so, it is reasonable to assume a heavy-tailed density for video file size.

[4] https://support.google.com/youtube/answer/71673?hl=en

While a theoretical heavy-tailed distribution has an infinite variance, there is a practical limit to the maximum file size. Therefore we fix the maximum permissible file size and the median file size; and randomly generate the sizes of the videos hosted by our server. We take a previously published value (median = 8.215 MB) to determine the minimum median value for the video size [11]. Even though results published by Barford et al. [3] suggest a Pareto distribution for the density of the traffic size, the `rpareto` function from GNU R's VGAM package we employ for our analysis produces hardly controllable variates. We therefore decide to replace it by the Weibull density which is implemented by a `rweibull` function in GNU R's stats package. We choose the parameter values $k = 0.3$ (shape) and $\lambda = 30$ (scale) to produce variates comparable to the above mentioned medians and maximum values. We then generate 5000 variates with $M_s = 8.514$, $\mu_s = 255.8$, and $max_s = 24680$, where M_s refers to the median video size; μ_s the mean video size, and max_s the maximum video size.

We then convert these figures to bytes and add an offset term to avoid a 0 byte video size. Gill et al. [11] use minimum payload sizes ranging from 452 to 95760 bytes for four different Youtube traces. We pick a random clip of 1s duration from Youtube and determine the minimum video size accordingly; it is 11500 bytes.

Based on these specifications we fragment a large video file of approximately 25 GB into 5000 randomly generated video clips. The sizes of these clips follow a Weibull distribution.

3.2 Video Popularity

Popularity refers to the number of times a particular video has been viewed in the past. Researchers who study the statistics of video popularity assert that it follows a power law[5] [11], but because most existing video servers employ recommendation systems, which make popular videos more popular (highly probable to be viewed) and less popular videos even less popular (less probable to be viewed), the distribution function experiences an exponential cutoff at the lower end of the density function [25,7,5,16]. For our case, we do not employ a recommendation system and, therefore, the video popularity is assumed to obey a power law. As a basis for establishing the parameter values of the power-law variates, we use actual values from *Dailymotion* as presented in [16] with $\alpha = -1.72$, and the $max_v = 2,895,396$.

3.3 View Gain

The video popularity serves as a basis for estimating the additional number of views a video gains in future. Cheng et al. [7] derive a quantitative expression for the view gain after x additional weeks as follows:

[5] the probability density function of a random variable x obeying a power law is expressed as: $f(x) = x^{\alpha}$, where α.

$$v(x) = v_0 \cdot \frac{(x+a)^p}{a^p} \tag{1}$$

where v_0 refers to the present popularity of the video; a refers to the age of the video in weeks at the beginning of the observation period and p refers to popularity growth factor. Cheng et al. provide the plot of the CDF of p but left out its mathematical expression. We perform a graphical analysis and estimate the CDF with a Weibull distribution with $W(2, 0.9)$. With this knowledge we calculate the additional view gains for each video our server hosts. We will use the terms *view gain* and *popularity gain* interchangeably.

3.4 Video Age

To determine a video's age, we use a parameter called video upload trend, α. The upload trend of a video hosting server refers to the number of videos it hosts each week: $n = w^{\alpha}$, where n refers to the number of videos currently hosted by the server and w refers to the number of weeks the server has been active. The upload trend of YouTube in 2008 has been estimated to be 2.61 [7]. Since our video server hosts 5000 videos at the time of our experiment, for $\alpha = 2.61$, the oldest video should be 16 weeks old whereas the newest video is 1 week old. Hence, "16 weeks ago" one video was uploaded and the first file size variate is associated with an age of 1. By applying the upload trend like this we calculate $\lceil 2^{2.61} \rceil = 7$ uploads for the second week (i.e., 15 weeks ago) and the next 7 request size variates are associated with an age of 2. After every request size variates is associated with an age, we calculate the video gain for each video using Equation 1. This procedure is repeated until the age of all videos are determined.

3.5 Request Distribution

The view gain expresses the number of additional views a video receives on a weekly basis. This term has to be broken down into days and the time of a day to estimate the workload size per unit time. For our experiment, we consider it sufficient to generate requests for a time span of one day. Therefore, we evenly distribute the view gain (i.e., the number of requests) to the seven days of the week, but a further even distribution of the daily requests to the 24 hourly slots is not plausible, because repeated observations indicate that the daily load of a multimedia server exhibits rather a wave-like distribution [11,31]. However, results in [25,27] show that stationary can be assumed if the day is split into time slots one hour or less. We thus employ a y-shifted cosine function to determine the portion of workload for each time slot.

Mathematically, this portion is determined as:

$$A_s = \int_l^u cos(x) + 1.1 \; dx \tag{2}$$

where $u = 2\pi\frac{s}{24}$, and $l = 2\pi\frac{s-1}{24}$ with $s = \{1, \ldots, 24\}$. Then, the amount of requests a video v receives in the time slot s is expressed as: $r_s^v = \frac{A_s}{A} \times g_d$, where $A = \int_0^{2\pi} cos(x) + 1.1\ dx$ and g_d is the view gain for day d.

3.6 Test Cases

It turned out that generating requests for all the files the video server hosts overwhelms the physical machine due to contention at the disk drive. This problem can be addressed in two different ways: 1) By reducing the number of available videos on the server, since the user request rate depends on this quantity (the larger the number of videos the server hosts, the larger the number of users it attracts); and (2) by scaling down the initial popularity of the video files since the view gain per week (and thus, per time slot) directly depends on this quantity. Similarly, the view gain for each video can be scaled down. For our experiment we adopt the first approach.

With our server configuration, a workload generated for 300 videos slightly overloads the sever (in terms of system load average[6]). When the number of videos is reduced to 200 videos, a low to medium load is generated on the server. When the number of videos is reduced to 100 videos, then the server has sufficient resources to accommodate user requests (low system load average). We considered these three scenarios to generate different request distributions and to analyze the relationship between the statistics of the workload generated by users and the statistics of the videos hosted by the server.

4 Observation and Analysis

In this section, we analyze the relationship between the statistics of (1) the service time for individual requests and the size of individual requests (workload size) (2) the workload size and the video size. In all our investigation, we shall focus on the probability distribution function (CDF) because this function sufficiently expresses a random variable.

As mentioned in section 1, the workload of a video server is a convolution of two random variables, namely, the request arrival rate and the size of individual requests. Another way of representing the convolution operation is to model these two random variables as components of an $M/G/1$ queuing systems where request size directly determines service time. Thus, we expect to see a strong correlation between the service time and the request size.

Two parameters mainly influence the distribution of request sizes: 1) the amount of available videos on the server and 2) the popularity (view) gain of each video in each time slot. We consider both parameters and generate six different types of workloads: We vary the number of available videos to 100, 200, and 300 and we consider four different types of workloads which are produced

[6] The load average refers to the average length of the CPU run queue. If this length is greater than the number of logical cores, we consider the server to be overloaded.

under the assumption that the hosted videos have initial video popularity (v_0 in Equation 1) distributions obeying power law (for all the video sets), normal, uniform, and gamma distributions (for the 200 video set). Fig. 1 shows the complementary cumulative distribution functions of the different view gains we consider to produce the workloads.

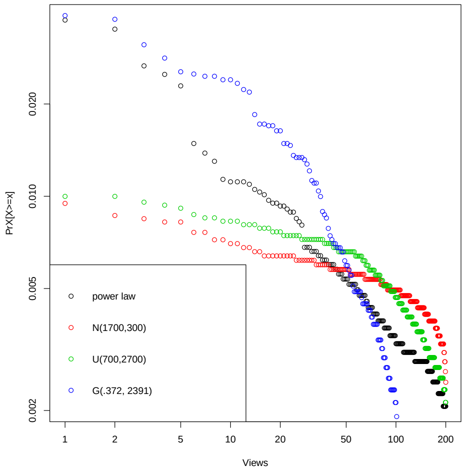


Fig. 1: The frequency-rank (CCDF) plot of the initial video popularity.

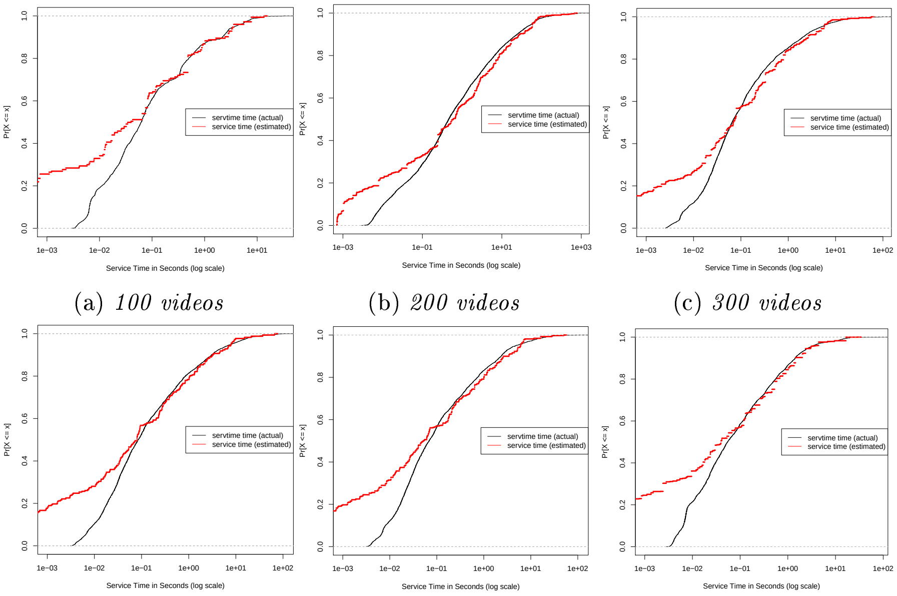
Varying the amount of available videos changes the parameters of the video size distribution while varying the popularity distribution changes the popularity gain and thus, the distribution of the request size. We perform the experiments in the time slots between 16:00 and 19:00 o'clock. However, in this paper, we shall limit ourselves to the analysis of the experiment results of the time slot between 18:00 and 19:00 o'clock (this is the time slot with the highest request rate). The experiment data for the other time slots do not lead to different results.

4.1 Service Time vs Request Size

One of the most critical parameters to evaluate the performance of a server is the service time as seen by clients. We define this time as the time span beginning from sending a request up to the time the requested video is downloaded completely. Technically, it is the time span beginning from starting to establish an HTTP session (wait time) until the HTTP session termination (download time).

The wait time (the time needed to establish a session) does not much depend on the request size; instead, it depends on the request arrival rate. Even so, the wait time is very small compared to the download time and can be neglected – the mean wait time that can be experienced under a heavy load is below

0.08 second whereas the mean service time under the same condition is 13.18 seconds. Therefore, the service time can be approximated by the download time, which depends on the size of the video being downloaded. Hence, we propose to estimate the service time in terms of the request size (video size) using a linear model.



(a) *100 videos* (b) *200 videos* (c) *300 videos*

(d) *200 videos (Normal)* (e) *200 videos (Uniform)* (f) *200 videos (Gamma)*

Fig. 2: Comparison of the cumulative distribution functions of the experimentally obtained and the estimated service times when the available videos for downloading are (top: from left to right) 100, 200, and 300. The workloads are generated with a power law distribution initial video popularity. Bottom: the available videos on the server are 200, but the initial video popularity follows (from left to right) normal, uniform, and gamma distributions.

Fig. 2 displays the relationship between the CDFs of the actual service time we measure and the service time we estimate from the size of incoming requests. While the graphs show deviations in the lower end of the distributions, the middle parts and the upper ends match comparatively well. To evaluate the goodness of fits, we calculate the R^2 values. The best estimation is achieved when the initial video popularity has a gamma distribution ($R^2 = 0.8132$) whereas the worst estimation is achieved when the initial video popularity follows a uniform distribution ($R^2 = 0.648$), which is reasonable, since a video popularity rarely follows a uniform distribution in reality. In general, the results suggest that the assumption that a linear relationship exists between the service time and the request size is a plausible assumption.

From the graphs it can be observed that the deviations between the CDFs of the actual and the estimated service times start to increase at values below 0.1 which comprise the lower 0.5 quantile of the values. There can be two reasons for this: 1) the lower-bound of the service time is fixed by the minimum wait time and the data rate of the network interface card. For our experiments the minimum wait time is 0.00257 second. 2) Due to the heavy-tail nature of the video size distributions, the request size is heavy-tailed as well (we shall discuss this fact shortly). In this case, the linear model produces better results for larger values of the request size than for smaller values. Since the request size comprises a range between six (for 100 videos) and seven (for 300 videos) orders of magnitudes, but the service time only between five (for 100) and six (for 300 videos) orders of magnitudes, it may not come as a surprise to observe a larger deviation at the lower end of the CDFs.

4.2 Request Size vs. Video Size

Likewise, we examine the existence of a relationship between the statistics of the request size of a workload (client-side property) and the statistics of the video size on the server (server-side property). Similar to the previous test cases we vary the amount of available videos on the server and the distribution of the initial video popularity to generate different workloads.

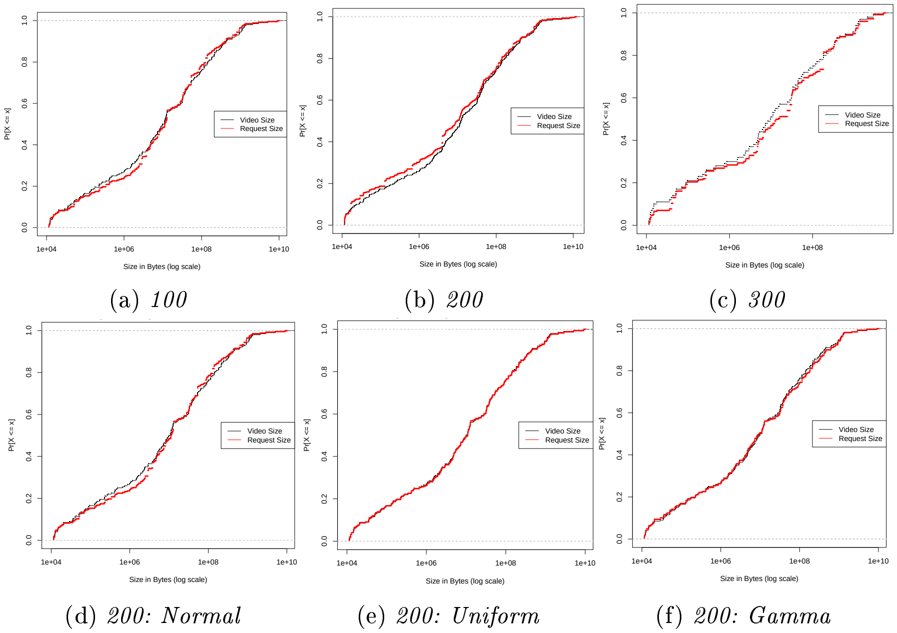
As can be recalled, the video size for all the test cases obeys a Weibull distribution, but each test case results in a different scaling factor. On the other hand, the workloads generated for each test case are dissimilar with each other because of the different popularity distributions we selected. Regardless of these variation, the graphs in Fig. 3 confirm that the distributions of the size of videos on the servers exhibit strong similarity with the distributions of the request size produced by users (it should be noted that both the distributions of the video size and the request size are measured in byte).

Interestingly, for all the test cases, the CDFs of both the request and the video sizes can be estimated by Weibull distributions. Tab. 1 and Tab. 2 summarize the shapes and scales of the two random variables for all the test cases we considered.

The above results clearly show that regardless of the distribution of view gain, the statistics of the request size can be sufficiently determined by the statistics of the size of the videos hosted by the server.

5 Conclusion

In this paper we analyzed the characteristics of a video server. We generated probabilistic workload and examined the relationship between server-side statistics and workload properties. In particular, we studied whether the probability distribution function of the request size - a property which is not influenced by the server configuration - is related to the probability distribution function of



(a) *100* (b) *200* (c) *300*

(d) *200: Normal* (e) *200: Uniform* (f) *200: Gamma*

Fig. 3: Comparison between the cumulative distribution functions of the video size on the server and the request size generated by users when the available videos for downloading are (top: from left to right) 100, 200, and 300. The workloads are generated by a power-law distributed initial video popularity. Bottom: the available videos on the server are 200, but the initial video popularity follows (from left to right) normal, uniform, and gamma distributions.

the size of videos hosted by the server. We found that the distribution function of the request size resembles the distribution function of the file size despite the fact that the distribution of the workload size is technically a convolution of the distribution function of the video size and the distribution function of the video popularity. Furthermore, we examined the relationship between the statistics of the service time and the workload size. We found that a linear relationship exists between the workload size and the service time.

We thus confirmed our assertion that the performance (service time) of a video server can be sufficiently predicted by examining the statistics of the video files it hosts.

References

1. Abts, D., Marty, M.R., Wells, P.M., Klausler, P., Liu, H.: Energy proportional datacenter networks. SIGARCH Comput. Archit. News 38(3), 338—-347 (2010)
2. Apparao, P., Iyer, R., Zhang, X., Newell, D., Adelmeyer, T.: Characterization & analysis of a server consolidation benchmark. In: Proceedings of the Fourth ACM

	Request Size Distribution		Video Size Distribution	
Subset	shape	scale	shape (rel. err)	scale (rel. err)
100 Videos	3.2934e-01	4.5283e+07	3.1854e-01 (.0328)	3.6322e+07 (.1979)
200 Videos	3.5754e-01	3.4089e+07	3.3792e-01 (.0549)	3.4331e+07 (.0071)
300 Videos	3.2390e-01	3.1389e+07	3.3279e-01 (.0274)	3.9909e+07 (.2714)

Table 1: The estimated parameters and the corresponding relative error ($rel.err = \frac{actual - estimated}{actual}$) for the video and the request size distributions. The server makes 100, 200, and 300 videos available for downloading and the workloads are generated with the assumption that the initial video popularity distribution obeys a power law.

	Request Size Distribution		Video Size Distribution	
Subset	shape	scale	shape (rel. err)	scale (rel. err)
Power law	3.576e-01	3.4724e+07	3.1854e-01 (.1092)	3.6322e+07 (.0439)
$N(1700, 300)$	3.543e-01	3.5378e+07	3.1854e-01 (.1009)	3.6322e+07 (.0267)
$U(700, 2700)$	3.350e-01	3.6285e+07	3.1854e-01 (.0491)	3.6322e+07 (.0010)
$\mathit{\Gamma}(.372, 2391) \cdot 10^6$	3.277e-01	3.9370e+07	3.1854e-01 (.0279)	3.6322e+07 (.0774)

Table 2: The estimated parameters and the corresponding relative error ($rel.err = \frac{actual - estimated}{actual}$) for the video and the request size distributions. The server makes 200 videos available for downloading and the workloads are generated with the assumption that the initial video popularity distribution obeys normal, uniform, and gamma distributions.

SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. pp. 21–30. VEE '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1346256.1346260

3. Barford, P., Crovella, M.: Generating representative web workloads for network and server performance evaluation. ACM SIGMETRICS Performance Evaluation . . . 26(1), 151–160 (Jun 1998), http://portal.acm.org/citation.cfm?doid=277858.277897http://dl.acm.org/citation.cfm?id=277897
4. Beloglazov, A., Buyya, R.: Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In: Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science - MGC '10. pp. 1–6. ACM Press, New York, New York, USA (2010), http://portal.acm.org/citation.cfm?doid=1890799.1890803
5. Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.Y., Moon, S.: Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems. IEEE/ACM Transactions on Networking 17(5), 1357–1370 (Oct 2009), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4801529
6. Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., Doyle, R.P.: Managing energy and server resources in hosting centers. In: Proceedings of the eighteenth ACM symposium on Operating systems principles - SOSP '01. p. 103. ACM Press, New York, New York, USA (2001), http://portal.acm.org/citation.cfm?doid=502034.502045
7. Cheng, X., Dale, C., Liu, J.: Statistics and Social Network of YouTube Videos. 2008 16th Interntional Workshop on Quality of Service pp. 229–238 (Jun 2008),

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4539688

8. Dargie, W., Strunk, A., Schill, A.: Energy-aware service execution. 2011 IEEE 36th Conference on Local Computer Networks pp. 1064–1071 (Oct 2011), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6115164
9. Dreslinski, R., Wieckowski, M., Blaauw, D., Sylvester, D., Mudge, T.: Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits. Proceedings of the IEEE 98(2), 253–266 (Feb 2010), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5395763
10. Fan, X., Weber, W.d., Barroso, L.A.: Power provisioning for a warehouse-sized computer. ACM SIGARCH Computer Architecture News 35(2), 13 (Jun 2007), http://portal.acm.org/citation.cfm?doid=1273440.1250665
11. Gill, P., Arlitt, M., Li, Z., Mahanti, A.: Youtube traffic characterization: a view from the edge. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement - IMC '07. pp. 15–28. IMC '07, ACM Press, San Diego, California, USA (2007), http://portal.acm.org/citation.cfm?doid=1298306.1298310
12. Gummadi, K.P., Dunn, R.J., Saroiu, S., Gribble, S.D., Levy, H.M., Zahorjan, J.: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. Proceedings of the nineteenth ACM symposium on Operating systems principles - SOSP '03 p. 314 (2003), http://portal.acm.org/citation.cfm?doid=945445.945475
13. Kansal, A., Zhao, F., Liu, J., Kothari, N., Bhattacharya, A.a.: Virtual machine power metering and provisioning. Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10 p. 39 (2010), http://portal.acm.org/citation.cfm?doid=1807128.1807136
14. Koomey, J.G.: Growth in data center electricity use 2005 to 2010. Tech. rep. (2011), http://www.analyticspress.com/datacenters.html
15. Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., Jiang, G.: Power and Performance Management of Virtualized Computing Environments Via Lookahead Control. 2008 International Conference on Autonomic Computing pp. 3–12 (Jun 2008), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4550822
16. Mitra, S., Agrawal, M., Yadav, A., Carlsson, N., Eager, D., Mahanti, A.: Characterizing Web-Based Video Sharing Workloads. ACM Transactions on the Web 5(2), 1–27 (May 2011), http://portal.acm.org/citation.cfm?doid=1961659.1961662
17. Möbius, C., Dargie, W., Schill, A.: Power Consumption Estimation Models for Processors, Virtual Machines, and Servers. IEEE Transactions on Parallel and Distributed Systems pp. 1–1 (2013), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6570721
18. Nathuji, R., Schwan, K.: Vpm tokens: virtual machine-aware power budgeting in datacenters. In: Proceedings of the 17th international symposium on High performance distributed computing - HPDC '08. p. 119. ACM Press, New York, New York, USA (2008), http://portal.acm.org/citation.cfm?doid=1383422.1383438
19. Padala, P., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., Salem, K.: Adaptive control of virtualized resources in utility computing environments. ACM SIGOPS Operating Systems Review 41(3), 289 (Jun 2007), http://portal.acm.org/citation.cfm?doid=1272998.1273026
20. Paxson, V., Floyd, S.: Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking 3(3), 226–244 (Jun 1995), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=392383

21. Petrucci, V., Carrera, E.V., Loques, O., Leite, J.C., Mossé, D.: Optimized Management of Power and Performance for Virtualized Heterogeneous Server Clusters. 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing pp. 23–32 (May 2011), `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5948593`
22. Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., Zhu, X.: No "Power" Struggles: Coordinated Multi-level Power Management for the Data Center. SIGARCH Comput. Archit. News4 pp. 48–59 (2008)
23. Sotomayor, B., Montero, R.S., Llorente, I., Foster, I.: Virtual infrastructure management in private and hybrid clouds. Internet Computing, IEEE 13(5), 14–22 (2009)
24. Strunk, A., Dargie, W.: Does Live Migration of Virtual Machines cost Energy? In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). pp. 514–521. Barcelona, Spain (2013)
25. Tang, W., Fu, Y., Cherkasova, L., Vahdat, A., Systems, I.: Long-term Streaming Media Server Workload Analysis and Modeling. Technical report, HP Laboratories (2003), `https://www.hpl.hp.com/techreports/2003/HPL-2003-23.pdf`
26. Veeraraghavan, K., Chen, P.M., Flinn, J., Narayanasamy, S.: Detecting and surviving data races using complementary schedules. In: Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. pp. 369–384. SOSP '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2043556.2043590`
27. Veloso, E., Almeida, V., Meira, W., Bestavros, a.: A hierarchical characterization of a live streaming media workload. IEEE/ACM Transactions on Networking 14(1), 133–146 (Feb 2006), `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1597229`
28. Verma, A., Dasgupta, G., Nayak, T.K., De, P., Kothari, R.: Server workload analysis for power minimization using consolidation. In: Proceedings of the 2009 conference on USENIX Annual technical conference. USENIX Association (2009), `http://dl.acm.org/citation.cfm?id=1855835`
29. Willinger, W., Taqqu, M., Sherman, R., Wilson, D.: Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. IEEE/ACM Transactions on Networking 5(1), 71–86 (1997), `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=554723`
30. Wood, T., Tarasuk-Levin, G., Shenoy, P., Desnoyers, P., Cecchet, E., Corner, M.D.: Memory buddies: exploiting page sharing for smart colocation in virtualized data centers. ACM SIGOPS Operating Systems Review 43(3), 27 (Jul 2009), `http://portal.acm.org/citation.cfm?doid=1618525.1618529`
31. Zink, M., Suh, K., Gu, Y., Kurose, J.: Characteristics of YouTube network traffic at a campus network – Measurements, models, and implications. Computer Networks 53(4), 501–514 (Mar 2009), `http://linkinghub.elsevier.com/retrieve/pii/S1389128608003423`