

Measurement, Modeling, and Analysis of Internet Video Sharing Site Workload: A Case Study

Xiaozhu Kang
Columbia University
New York, NY 10027
xk2001@columbia.edu

Haifeng Chen
NEC Laboratories America
Princeton, NJ 08540
haifeng@nec-labs.com

Hui Zhang
NEC Laboratories America
Princeton, NJ 08540
huizhang@nec-labs.com

Xiaoqiao Meng
IBM T.J. Watson Research Center
Hawthorne, NY 10532
xmeng@us.ibm.com

Guofei Jiang
NEC Laboratories America
Princeton, NJ 08540
gfj@nec-labs.com

Kenji Yoshihira
NEC Laboratories America
Princeton, NJ 08540
kenji@nec-labs.com

Abstract

In this paper we measured and analyzed the workload on Yahoo! Video, the 2nd largest U.S. video sharing site, to understand its nature and the impact on online video data center design. We discovered interesting statistical properties on both static and temporal dimensions of the workload; they include file duration and popularity distributions, arrival rate dynamics and predictability, and workload stationarity and burstiness. Complemented with queueing-theoretic techniques, we extended our understanding on the measurement data with a virtual data center design assuming the same workload as measured, which reveals results regarding the impact of workload arrival distribution, Service Level Agreements (SLAs) and workload scheduling schemes on the design and operations of such large-scale video distribution systems.

1. Introduction

Internet Video sharing web sites such as YouTube [2] have attracted millions of users in a dazzling speed during the past few years. It is reported that in the month July 2007, Americans view more than 9 billion video streams online, and 3 out of 4 U.S. Internet users streamed video online; the industry growth of the online video market keeps fast, with the number of video viewed jumping 8.6% from May to July 2007, in just 2 months [4, 3].

Massive workload accompanies those web sites along with their business success. For example, in July 2007, YouTube, the largest Internet Video sharing web site, had daily video views of 79 millions; Yahoo! Video [1], ranked

2nd in U.S. online video properties, delivered 390 millions of video views in the same month [4]. How those companies design and operate their data centers to deliver the service are unknown to the outside, perhaps because the related techniques are deemed as a key part of their core competency over competitors [18]. What's known publicly is that even YouTube is notorious for its unstable service, despite spending several million dollars monthly on bandwidth; as it is said, *You know it's Friday when YouTube is slow* [6].

In order to understand the nature of such unprecedented massive workload and the impact on online video data center design, we analyze Yahoo! Video web site in this paper. The main contribution of our work is an extensive trace-driven analysis of Yahoo! Video workload dynamics. The highlights of our work can be summarized as the follows:

- Compared with other Internet video sharing site workload measurement [9, 12, 10], we are the first to analyze the workload at the time window of minute level (i.e., 30 minutes rather than 1 day or longer).
- We gave a comprehensive workload characterization for the second largest online video site in US.
- We observed that the measured massive workload can be clearly separated into two contrasting components: a predictable *well-behaving* workload, and a bursty, non-predictable *trespassing* workload. We pointed out the nature of each component, and explained its implication on workload management.
- We gave quantitative analysis of the impact of workload arrival distributions (exponential and heavy-tailed), Service Level Agreements (SLAs) (average and tailed distribution performance guarantee), and

workload scheduling schemes (random dispatching and Least Workload Left) on the resource management efficiency of an online video data center.

The rest of the paper is organized as follows. In Section 2 we present the related work on Internet video workload measurement. Section 3 describes Yahoo! Video web site and the data collection method we used for the workload data collection. The analysis of the Yahoo! Video measurement data is presented in Section 4, categorized into static and temporal properties. Section 5 presents a set of queueing-model based analysis results on a virtual VoD data center design, assuming the same measured workload. We conclude this paper in Section 6 with a discussion of future work.

2. Related Work

Measurement study on Internet video sharing services has been focusing on YouTube, the leader in this area. For example, [9] collected data on the video files of two categories, and studied the corresponding statistical properties; they included the popularity life-cycle of videos, the relationship between requests and video age, and the level of content aliasing and illegal content in the YouTube system. The workload on the “most viewed” category of YouTube site is measured in [12] based on the traffic from a campus network, and they examined usage patterns, file properties, popularity, refereeing characteristics as well as transfer behaviors of YouTube. [10] crawled the YouTube web site and collected data information on about 2.7 million YouTube video files; the study showed noticeably different statistics on YouTube videos compared to traditional streaming videos, such as length, access pattern, active life span, etc.. Our work complements these daily based or even rougher measurement works, and distinguish from the previous study by providing richer workload dynamics information in the measurement data that are collected at a frequency of 30 minutes.

There are also many workload measurement studies on traditional online video systems. For example, [7] presented the workload analysis on two media servers located at two large universities in US. [21] used two long-term traces of streaming media services hosted within HP to develop a streaming media workload generator. [13] studied the media workload collected from a large number of commercial Web sites hosted by a major ISP and that from a large group of home users via a cable company. [25] presented file reference characteristics and user behavior analysis of a production VoD system in China called Powerinfo system. More recently, [16] analyzed a 9-month trace of MSN Video, Microsoft’s VoD service, and studied the user behavior and video popularity distribution. While sharing

some similarity, the workload characterization on Internet video sharing sites is quite different from that of a traditional online video system, as reported in the previous studies and also in this paper.

Caching and Peer-assisted streaming are suggested to improve the distribution system performance for Internet video services [16, 12, 9]. While we believe that a centralized data center will be the key component in a video distribution infrastructure in practice, our study does consider the implication of the measured workload characteristics on the effectiveness of those two technologies.

3. Yahoo! Video Workload Measurement

3.1. Yahoo! Video

Yahoo! Video [1] is ranked the 2nd US online video website just after YouTube in terms of total number of video views, following YouTube; during July 2007, it delivered totally 390 millions of video streams to 35.325 millions of unique US viewers, and contributed 4.3% to the US online video traffic [4].

On Yahoo! Video site, all the videos are classified into 16 categories. Each video is assigned a unique ID (integer number), and has the following information posted on its webpage: *title*, *number of views (so far)*, *video duration*, *average rating*, *number of ratings*, *added (uploaded) time*, *source (video producer)*, *content description*, *tags (keywords)*, and *comments*.

3.2. Data Collection

We crawled all 16 categories on the Yahoo! Video site for 46 days (from July 17 to August 31 2007), and the data was collected every 30 minutes. This measurement rate was chosen as a tradeoff between analysis requirement and resource constraint. Due to the massive scale of Yahoo! Video site, we limited the data collection to the first 10 pages of each category. Since each page contains 10 video objects, each time the measurement collects dynamic workload information for 1600 video files in total. Throughout the whole collection period, we recorded 9,986 unique videos and a total of 32,064,496 video views. This can be translated into a daily video request rate of 697064. While the measured requests were only a small subset of the total workload on Yahoo! Video, we found that the represented workload bore very similar characteristics (such as video duration and popularity distributions) like the measured workload on YouTube with much larger data set [10, 9], which will be reported in Section 4.

4. Workload Statistics

4.1. Static Properties

4.1.1 Video Duration

We recorded 9,986 unique videos in total, and the video durations range from 2 to 7518 seconds. Among them, 76.3% is less than 5 minutes, 91.82% is less than 10 minutes, and 97.66% is less than 25 minutes. The mean video duration is 283.46 seconds, and the median duration is 159 seconds. This is similar to the statistics on YouTube videos reported in [10], although YouTube videos seem to have even shorter durations. It might be due to the different video upload policies on the two sites: YouTube imposes the limit of 10 minutes on regular users, while Yahoo! Video allows the video size up to 100MB (more than 40 minutes at 300Kb/s bit rate) to regular users.

On those video sharing sites, the video bit rate is usually around 300Kb/s with FLV format; therefore short video durations in less than 5 minutes mean small videos size in less than 10 MB. If file popularity were also skewed (e.g., following Zipf), then a video server with a few GB memory could easily stream most requests without accessing its disks, therefore maximizing its capacity in terms of number of concurrent connections supportable. Next, we investigate the file popularity in the measured workload.

4.1.2 File Popularity

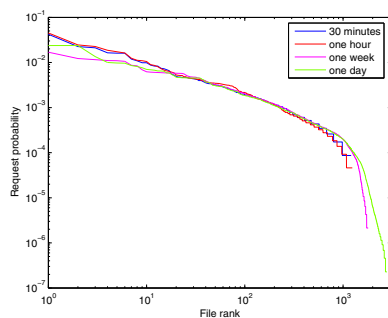


Figure 1. File popularity distribution in different time scales.

File popularity is defined as the distribution of stream requests on video files during a measurement interval. Fig 1 shows typical file popularity distributions at 4 time scales (30 minutes, 1 hour, 1 day, and 1 week). As we can see, the distributions at different time scales are quite similar. This means user access pattern on content has some *self-similar* property, and implies that request distribution on files may

have strong stationarity. We further validate this hypothesis in Section 4.2.1.

The popularity data follow the widely used Pareto-principle (or 80-20 rule). Our data shows 20% of the top popular videos account for nearly 80.1% of the total views, while the rest 80% of the videos account for the rest workload.

We pick the distribution of one week and perform goodness-of-fit test with several distribution models. It turns out Zipf with an exponential cutoff fits best and well (with Zipf exponent -0.8545). It is interesting that [9] also reported the same fitting result for their YouTube video popularity study. The same paper gave a thorough discussion on the nature of Zipf with a truncated tail and its potential causes and implication on video sharing efficiency, which we do not repeat in this paper.

4.1.3 Correlation between File Popularity and Duration

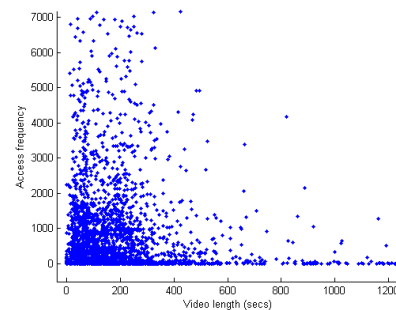


Figure 2. File popularity and duration relationship.

Related to video duration distribution and popularity are questions involving user preference: the relationship between size and popularity. Figure 2 shows the scatter plot for the data collected from July 25th to Aug 31st. In the scatter plot, each dot represents the number of views of the corresponding video size. The figure shows obvious larger density for shorter videos, and the most popular files are mostly quite small. Therefore, Figure 1 and Figure 2 tell us that, in addition to the tendency for video files as being created to be small, video viewers also prefer short videos.

4.2. Temporal Properties

4.2.1 Request Size Distribution Stationarity

Request size distribution is defined as the distribution of stream requests on video durations during a measurement interval. Its stationarity property has a strong correlation

with file access locality. Understanding it is also important in data center performance management as size-based workload scheduling schemes have been shown outstanding performance for provisioning general web services [15, 26].

Now we check the stationarity of request size distribution in terms of service time (video duration). We use *histogram intersection distance* to measure the change between two request size distributions at different time points. The histogram intersection distance is defined as [20]:

$$d_{1,2}^h = 1 - \sum_{m=0}^M \min(p_1[m], p_2[m])$$

where p_1, p_2 are the two distribution histograms, and m the bins. In the study, we generated the histogram of a workload distribution with file size range of [2, 7518] (in seconds) and arbitrarily picked the bin number as $M = 751$; for each time scale, we calculated the pair-wise histogram intersection distance of two adjacent data points along the time.

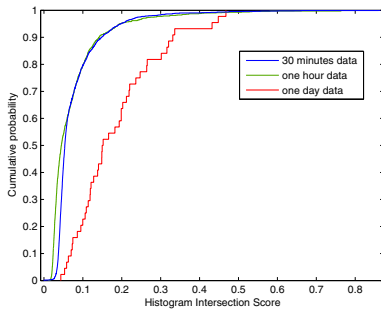


Figure 3. Histogram intersection distance distribution.

Figure 3 shows the CDFs of histogram intersection distance distribution for 3 time scales. We can see that with 30-minute and one-hour scales, the histogram distance is very small for most of the time. For example, 80% of the time it is no more than 0.1; 90% of the time it is no more than 0.15. But from day to day, the difference of request size distributions is obvious. Therefore, if we want to do short-term capacity planning, the current workload distribution is a good indication for the next time period, and dynamic provisioning only needs to focus on request arrival rate dynamics. However, if we want to carry out the capacity planning at daily basis, both arrival rate dynamics and request size distribution dynamics need to be taken into account.

4.2.2 Arrival Rate Predictability

The first metric we look at is the over-provisioning factor, which is defined as the ratio of the maximal arrival rate to the average arrival rate through the whole time. This is one value a data center operator takes into consideration in planning his/her system. At a time scale of 30 minutes, the over-provisioning factor is 13.2; at one hour, it is 12.9. Until at one day or one week the aggregation effect reduces the factor to a number less than 2. Clearly, the cost could be high if an operator over-provisioned the data center based on the over-provisioning factor.

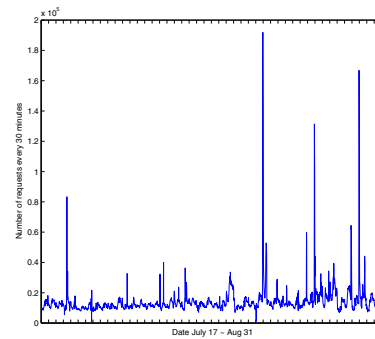


Figure 4. Arrival rate evolution at 30-minutes time scale.

Fig 4 shows the evolution of the request arrival rate at the time scale of 30 minutes. It shows significant dynamics with irregular spikes, which are separate burstiness lasting for short time. The arrival rate evolution at the time window of one hour and one day still contain similar dynamics with irregular spikes. But if we removed those spikes from Fig 4, the rest of the data points seem to be quite regular and predictable, which are shown as belows.

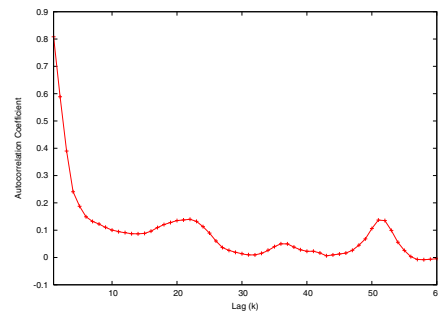


Figure 5. Workload autocorrelation coefficient.

We calculate the autocorrelation coefficient of the arrival rates at the time scale of 30 minutes; as shown in Figure 5,

the workload is highly correlated in short term (lag $k \leq 2$).

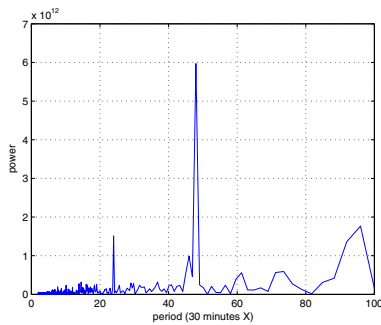


Figure 6. Periodicity of workload.

We also use Fourier analysis to discover the possible periodicity in the workload dynamics after removing the spikes. As shown in Figure 6, the maximum value on the figure indicates that the period is one day. With the strong periodicity components, known statistical prediction approaches like that in [8] can be applied to make accurate rate prediction in dynamic resource provisioning.

4.2.3 Burstiness

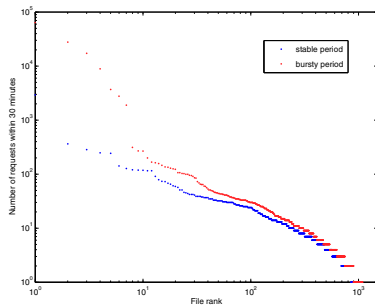


Figure 7. Popularity comparison of 30 minutes stable interval and bursty interval

While we can not predict these unexpected spikes in the workload, it is necessary to learn the nature of the burstiness and find out an efficient way to handle it once a bursty event happens. The comparison of the request (popularity) distribution during one spike interval and that in the normal interval right before it is shown in Figure 7. Clearly, the workload can be seen as two parts: a base workload similar to the workload in the previous normal period, and an extra workload that is caused by several very popular files.

4.3. Discussion

The characteristics of video duration and popularity distributions suggest that commodity PCs can work as high-performance video servers for those video sharing sites. Assuming disk I/O as the performance bottleneck, traditional streaming servers have special hardware configuration with complicated disk access scheduling schemes. But a media server is CPU bounded if it can serve streams from its memory [11], and the latter is supported by our video duration and file popularity statistical data. The trend of multi-core CPU architecture for commodity PCs make them even more powerful and fit for this new video service.

The temporal properties of the measured workload reveal two contrasting components: a “well-behaving” workload component that has strong variability, strong autocorrelation, and strong periodic component; and a “trespassing” workload component that is bursty and unpredictable. For a data center operator with the measured workload, the good news is that for all the time there is the “well-behaving” workload component which is ideal for dynamic provisioning; in rare time the “trespassing” workload comes, there is still positive news that it has extremely high content locality, and the operator can make the best of caching/CDNs, or simply provision extra servers with the best-scenario capacity estimation on memory-based streaming (usually 2.5 to 3 times higher than the capacity estimation on disk-based streaming [11]).

5. Workload and capacity management: a virtual design

Over-provisioning is the most typical practice for workload and capacity management of Internet applications. It simplifies the management but with the obvious shortcoming on resource demand, which includes not only hardware but also power/energy consumption and building infrastructure. Given the observations on large over-provisioning factor and good workload predictability in Section 4, we would like to further investigate a few other factors that affect the efficiency of dynamical provisioning on such services. Those factors include arrival rate distribution, workload scheduling schemes, and Service Level Agreements (SLAs).

In the following, we use the workload statistics of the 30-minutes measurement to simulate the requests on a virtual Yahoo! Video data center, and carry out dynamic provisioning at the same frequency. Queueing-theoretic techniques are used to estimate the resource (server) demand with different instance combinations of the three factors mentioned above, and we give a quantitative analysis of those factors with a set of numerical examples.

5.1. Methodology

Non-homogeneous Arrival Model: Because our measured data does not contain detailed information on individual requests, interarrival time distributions can not be inferred directly. In the analysis, non-homogeneous arrival process with homogeneous intervals [19] is used to model the individual request arrivals. In the model, the time is divided into 30-minutes intervals; each interval has its own expected arrival rate according to the measurement data; within each interval, the arrivals come following a constant-rate stochastic proceeds with certain interarrival time distribution. We test two typical traffic interarrival time distributions, exponential and heavy-tailed (i.e., Pareto $P[X > x] = (x_0/x)^\alpha$ with $x \geq x_0, \alpha > 1$).

System model: We model a single video server as a group of virtual servers with First Come First Served (FCFS) queues. The virtual server number corresponds to the physical server's capacity, which is defined as the maximum number of concurrent streams delivered by the server without losing a quality of stream. In the analysis, the number 300 is chosen for the capacity of a video server based on the empirical results in [11]. In this way, we model the video service center as a queueing system with multiple FCFS servers.

Workload Scheduling Schemes: We assume video files are stored in a Storage Area Network (SAN) and a video server can serve any video through this SAN. We choose two well-known scheduling schemes to study: random dispatching, which doesn't make use of any information of the servers and just sends each incoming job to one of s server uniformly with probability $1/s$; Least workload Left (LWL) scheme, which tries to achieve load balancing among servers by making use of the per-server workload information and assigns the job to the server with the least workload left at the arrival instant.

The reason we pick these two schemes is that they represent two extreme cases in the management overhead: random dispatching is stateless therefore robust, while LWL needs up-to-date information on all servers and therefore sensitive to system changes. Clearly the former is preferred to the latter in a data center design if its performance does not lag too much behind that of the latter. The measurement data give us a realistic example to check on the tradeoff.

Service Level Agreements: Service Level Agreements (SLAs) for a service define the Quality of Service (QoS) guarantees for a specified service based on the cost model and the anticipated level of requests from the service customers. We consider two QoS metrics for SLAs: the stream quality for an accepted connection, and the waiting time of a video request in the queue before accepted for streaming. Assume enough network bandwidth, then QoS on stream quality within the data center side can be guaranteed

through admission control based on server capacity. For the waiting time W , we consider two types: *maximal average waiting time* (W_{avg}), defined as $\mathbf{E}[W] \leq x, x > 0$; *bound on the tail of the waiting time distribution* (W_{tail}), define as $P[W > x] < y$ with $x > 0, y < 1$. For example, SLA could be that 90% of the requests experience no more than 5 seconds delays, i.e., $x = 5$ and $y = 90\%$.

5.2. Pareto vs Exponential arrivals

First, we want to understand the effect of job interarrival distribution on the performance measure of the system, thus also the resource demand. According to Kingman's upper bound on the delay for $GI/GI/1$ queue, the arrival rate λ a server can support is [24]:

$$\lambda \geq \frac{1}{m_1 + \frac{\sigma_a^2 + \sigma_b^2}{2(d-m_1)}}, \quad (1)$$

where d is the mean waiting time and σ_a^2, σ_b^2 are the variance of inter-arrival and service time respectively. Thus as the variance of inter-arrival time σ_a^2 increases, more servers will be required to achieve the SLA. It is thus intuitive that we can save servers by reducing the variance of job interarrival times.

We take the measurement data on August 1st between 00:00am and 00:30am to extract the workload parameters $\lambda = 1285.103, m_1 = 163.7021, m_2 = 54905.8$. If we set the SLA W_{avg} as $\mathbf{E}[W] \leq 1.5m_1$, then we need 1190 physical servers for Poisson arrivals. If we assume the interarrival distribution is Pareto $\Pr[X > x] = (0.00045/x)^{2.3713}$ (we have picked the parameters such that the expectation of interarrival is the same as Poisson case, and also the variance is finite), then we will need 1440 physical servers, which is 21% more resource demand.

We suggest to introduce a workload shaping component before the job dispatcher like that proposed in [5]; not only because it can help us save servers by transforming a heavy-tailed interarrival (or any) distribution to a light-tailed exponential one, but also it enables us to carry out more accurate resource provisioning with the queueing results for $M/G/s$ system. As pointed out in [22], the best approximation result known theoretically on the waiting time distribution of $GI/GI/s$ model (corresponding to the heavy-tailed arrival scenario) is less reliable because they have not yet been studied sufficiently and evidently depend more critically on the distribution beyond the first two moments of the interarrival distributions.

5.3. From Average to Tailed Performance Demand

Next, we want to understand the impact of SLA requirements on the resource demand. Taking the same measure-

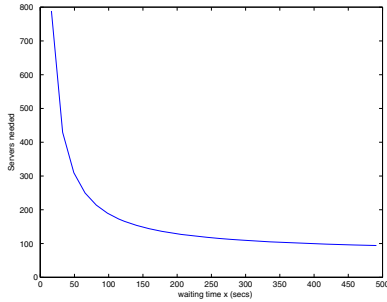


Figure 8. resource demand: as a function of x in W_{avg} .

ment data on August 1st from 00:00am to 00:30am and considering random dispatching scheme, we vary the parameter x in the SLA W_{avg} from $0.1m_1$ to $3m_2$, the resource demand as a function of x is shown in Figure 8. When the SLA requirement is not high ($x > m_1$), the resource demand remains almost flat; after the requirement reaches some cut-point m_1 , the demand increases with a dramatic speed with stricter requirement; a data center operator should take this function into consideration when deciding the cost model in the SLA.

For the SLA W_{tail} , we fix the parameter $x = m_1$, and then vary the other parameter y . We had a similar observation like that for W_{avg} , and note that for this type of SLA, the resource demand is even more sensitive to the SLA parameters.

5.4. Is random dispatching good enough?

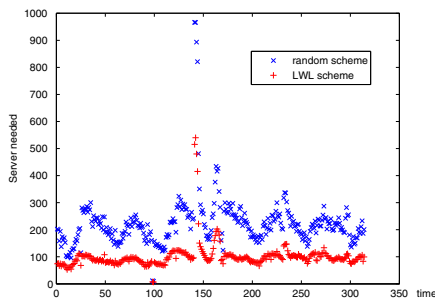


Figure 9. Resource Demand within one week: W_{avg} case

Taking the one-week measurement data from Aug 13th to Aug 20th, we numerically calculated the server demands of random and LWL dispatching schemes with Poisson ar-

rivals (based on results in [23]) and set the SLA requirement as W_{avg} : $\mathbf{E}[W] \leq 1.5m_1$. Figure 9 shows the results; on average 54.9% of servers could be saved with LWL scheme as compared to random dispatching scheme.

Using the same data and set the SLA requirement as W_{tail} : $\Pr[W > m_1] < 0.3$, we repeat the simulation; on average 69.9% of servers could be saved with LWL scheme as compared to random dispatching scheme.

5.5. Discussion

As we can see, by making use of the server load information and implementing dispatching schemes beyond naive random scheme could reduce the resource demand of such a large data center dramatically, especially under strict SLAs. Depending on the job size distribution, the best dispatching scheme may be different as discussed in [14].

One factor we did not count into when doing capacity planning analysis is incomplete video sessions observed in streaming workloads [25, 21]: a significant amount of clients do not finish playing an entire video. Our measurement data missed such information without individual view session information. We note our analysis gave an upbound on the resource demand assuming no incomplete video sessions; our analytical tool can take into such information and may yield results with even lower resource demand.

6. Conclusions

In this paper, we present the measurement study of a large Internet video sharing site - Yahoo! Video. With a clear goal to facilitate the data center design, this paper gives a comprehensive workload characterization and proposes a set of guidelines for workload and capacity management in a large-scale video distribution system

The success of YouTube brings online video into a new era, and its success pushes another wave of building large online video sites on the Internet [17]. We believe our results in this paper will give insights for the design and operations management of those new Internet service hosting data centers.

References

- [1] Yahoo! video. <http://video.yahoo.com>.
- [2] Youtube. <http://www.youtube.com>.
- [3] ComScore Video Metrix report: 3 out of 4 U.S. Internet Users Streamed Video Online in May. <http://www.comscore.com/press/release.asp?press=1529>, March 2007.

- [4] ComScore Video Metrix report: U.S. Viewers Watched an Average of 3 Hours of Online Video in July. <http://www.comscore.com/press/release.asp?press=1678>, July 2007.
- [5] D. Abendroth and U. Killat. Intelligent shaping: Well shaped throughout the entire network? In *Proceedings of INFOCOM 2002*, 2002.
- [6] M. Abundo. Youtube tracks outages on status blog. <http://www.insideonlinevideo.com/?p=166>, June 2007.
- [7] J. M. Almeida, J. Krueger, D. L. Eager, and M. K. Vernon. Analysis of educational media server workloads. In *NOSSDAV '01: Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*, pages 21–30, New York, NY, USA, 2001. ACM.
- [8] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing sla violations. In *IM '07: 10th IFIP/IEEE International Symposium on Integrated Network Management*, pages 119–128, Munich, Germany, 2007. IEEE.
- [9] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ACM proceedings of Internet Measurement Conference*, San Diego, CA, USA, October 2007.
- [10] X. Cheng, C. Dale, and J. Liu. Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study. *ArXiv e-prints*, 707, July 2007.
- [11] L. Cherkasova and L. Staley. Measuring the capacity of a streaming media server in a utility data center environment. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 299–302, New York, NY, USA, 2002. ACM.
- [12] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: A view from the edge. In *ACM proceedings of Internet Measurement Conference*, San Diego, CA, USA, October 2007.
- [13] L. Guo, S. Chen, Z. Xiao, and X. Zhang. Analysis of multimedia workloads with implications for Internet streaming. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 519–528, New York, NY, USA, 2005. ACM.
- [14] M. Harchol-Balter, M. E. Crovella, and C. D. Murta. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing*, 59(2):204–228, 1999.
- [15] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal. Size-based scheduling to improve web performance. *ACM Trans. Comput. Syst.*, 21(2):207–233, 2003.
- [16] C. Huang, J. Li, and K. W. Ross. Can internet video-on-demand be profitable? In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 133–144, New York, NY, USA, 2007. ACM.
- [17] D. Kawamoto. NBC, News Corp. push new Web rival to YouTube. ZDNet News, March 2007.
- [18] T. O'Reilly. What is web 2.0: Design patterns and business models for the next generation of software. <http://www.oreillynet.com/lpt/a/6228>, September 2005.
- [19] V. Paxson and S. Floyd. Wide-area traffic: the failure of poisson modeling. *SIGCOMM Comput. Commun. Rev.*, 24(4):257–268, 1994.
- [20] M. J. Swain and D. H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.
- [21] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat. Modeling and generating realistic streaming server workloads. In *Computer Networks: the International Journal of Copmuter and Telecommunications Networking Archive*, pages 336–356, January 2007.
- [22] W. Whitt. Approximations for the GI/G/m queue. *Production and Operations Management*, 2(2):114–161, Nov 1993.
- [23] W. Whitt. Partitioning customers into service groups. *Management Science*, 45(11):1579–1592, Nov 1999.
- [24] R. W. Wolff. *Stochastic Modeling and Theory of Queues*. Prentice Hall, 1989.
- [25] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. In *Proceedings of the 2006 EuroSys conference*, Leuven, Belgium, April 2006.
- [26] Q. Zhang and W. Sun. Workload-aware load balancing for clustered web servers. *IEEE Trans. Parallel Distrib. Syst.*, 16(3):219–233, 2005. Member-Alma Riska and Member-Evgenia Smirni and Senior Member-Gianfranco Ciardo.