Dissertation Summary

Quantile Function-based Models for Resource Utilization and Power Consumption of Applications

submitted in partial satisfaction for the degree of Doktoringenieur (Dr.-Ing.)

at

Technische Universität Dresden
Faculty of Computer Science

by

Christoph Möbius
born July 26th, 1978 in Wolfen

submitted: December 18th, 2015

Advising Professor:
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

1 Motivation

While there has been research ongoing for many years how power consumption of single machines, clusters, and entire data centers can be reduced, idle power consumption of servers still accounts for up to 60% of the maximum power consumption [5]. A much-noticed article of 2007 showed that average CPU utilization for more than 5,000 servers over a period of six months was between 10% and 50% [2]. A large fraction of the servers utilized CPU even below 10%. This situation of *underutilization* in data centers has not changed much since then: In [12], Delimitrou & Kozyrakis state that average CPU utilization at Twitter is below 20% - while up to 80% of CPU capacity is reserved but usually idle. Reiss et al. find that CPU utilization at a Google cluster is between 30% and 75%, while always more than 100% of the cluster's CPU capacity is allocated [23]. And according to H. Liu the CPU utilization at Amazon EC2 is between 3% and 17% [18].

Obviously, it would be the best strategy to switch a computer on only for the very moment it needs to perform as task – and ideally utilizes all its resources to full capacity – and immediately switch it off afterwards.

Indeed, this is the idea behind *server consolidation*. Its goal is to merge applications (or *services*) of servers that are underutilized on the least amount of servers such that every application still can perform its task, but the employed servers are highly utilized and remaining servers are either switched off or put in a low-power state. Consolidation, however, leads to another problem, called *resource interference*. Since the amount of resources on a server (i.e. CPU, memory, storage and network devices) are limited and access to these resources is shared among all applications it is virtually inevitable that the applications cannot perform their respective tasks with the same speed as if they had sole access to all resources. A large body of work of partial solutions to the problem exist, e.g. pinning applications to particular CPU cores to avoid interference at the cache level [11], systems for interference management [22], or data centers schedulers that attempt to minimize possible interference in the first place [12]. However, during a day the workload (i.e the frequency of requests and the associated size of individual requests in terms of their *difficulty*) usually changes. Typically, this carries forward to changes in the resource demands of an application that process these requests. At the same time a different application can experience changes in its resource utilization due to the changes of the first application. While the change of the resource utilization of the first application is caused by a change of its workload, the change for the second application is due to interference. In order to distinguish the first cause from the other one may (a) periodically perform reclassification of applications [12] with proactively inducing interference (and thus, intentionally reduce an application's performance). Or special algorithms are executed if changes of resource utilization are detected in order to make an educated, but not necessarily correct guess [24, 22]. In both cases, interference is only detected and then mitigated, but not prevented. Until detection and mitigation the affected applications suffer from performance degradation which directly translates to a reduced energy efficiency. It could be thus benefical to incorporate information that relates the workload of an application to its resource demand.

We assume that having such knowledge at hand before actually consolidating workloads can support consolidation decisions in a manner that those consolidation is prevented which would lead to considerable resource interference. Since resource utilization and thus, resource interference, depends on the workload of the application it is therefore necessary to estimate the very relationship between workload of an application and its resource utilization. To enable later statements regarding the impact of the resource interference on energy efficiency it is additionally necessary to examine the relationship between resource utilization and power consumption.

To establish such models, regression techniques are very widely employed [21]. Models of these kinds necessitate that the number of data points in the measurement data for both, the *predictor* parameter(s) and the *predicted* parameter are the same. Or, in other words, that a bijective mapping between the samples of the exogeneous and the endogeneous variables exists. For different reasons this is sometimes not the case. If the samples are time series, resampling techniques are maybe applicable. But if timeing information in the data is unreliable or not available resampling cannot be applied. Quantile-quantile (QQ) curves offer a means to establish the bijective mapping in these cases and this motivates the topic of this thesis.

2 Thesis Goals

In an early phase of the CRC 912 of the Deutsche Forschungsgemeinschaft – in which this thesis is embedded – a part of the participating research groups agreed on using a video server and a transcoder as central examples. Within our research group it was decided to provide the applications with realistic workloads so that the results of our experiments would have more than purely academic meaning. As it points out, there is no workload generator for a video server freely available that produces realistic workloads. Thus, the following question needs to be answered at first:

1 What are the statistical properties of video server traffic and how can we generate traffic that resembles these properties.

During the review of related work to answer this question the particularly interesting finding was encountered that the heavy-tailed distribution of *request* sizes of the Internet traffic is mainly due to the heavy-tailed distribution of *file* sizes and that the effect of file popularity seems only to have little effect [8]. It was concluded that a server operator could be able to estimate the (unknown) distribution of request sizes by merely considering the (readily available) distribution of file sizes. This would allow her to estimate the workload to the system without the necessity to learn each file's popularity. But this necessitates a functional expression of the relationship between the two distributions. Such an expression was not readily available and motivated the second research question:

2 How can we apply (linear) regression to samples of different lengths where time stamps are not meaningful, or unavailable, or unreliable?

The found answer promised to provide the means to answer the main research question:

3 How can we stochastically describe the resource utilization of an application as a function of its workload?

3 A Realistic Workload Generator for Video Servers

In [1] Barford & Crovella define seven statistical properties (*features*) of web server traffic that form the basis our implementation. The relevant features for a video server are (1) the distribution of interarrival times (2) the distribution of file sizes, (3) the distribution of popularity of single files, (4) the distribution of request sizes, and (5) the probability that a recently requested file is requested again soon (i.e. *temporal locality).*

Among the body of related work that examines these features for current video platforms like Youtube, Dailmotion, or Metacafé no team of authors examined *all* of these feaures. Also, for some features no recent work seems to exist. Thus, the findings of several teams of authors are combined by (a) focusing on results for Youtube since this platform is investigated by most authors, (b) employing the findings of the majority of publication if relatively recent work exists, (c) anticipate the development of properties where findings are obviously already outdated, and (d) defaulte to properties of usual Internet traffic if no specific information is available at all.

In summary, the distribution of file sizes follows a lognormal distribution with parameters $ln\mathcal{N}(2.1, 2.3)$ (applying methods (b) and (c)), and file popularity is modeled by a lognormal distribution with parameters $ln\mathcal{N}(1.72, 229.51)$ [20]. Features (1) and (4) are not explicitly modeled but are the result of how the popularity of a file develops with time, how many additional views a video gains during the next week as a result of this popularity dynamic, and how these additional views are distributed across weekdays and daytimes. The amount of additional views, v, for the next week for a video are calculated following the apporach in [7]: $v = v_0 \cdot \frac{(1+\mu)^p}{\mu^p}$ where v_0 denotes the overall amount of views the video has gained until now, μ denotes the age of the video in weeks, and p reflects the development of the file's popularity which is modelled by a Weibull distribution with parameters $\mathcal{W}(2, 0.9)$ [5].

To model the change of request rates during a day a sine-wave is employed that peaks at 18:00 'o clock. A sine wave is also used to model the change of request rates with weekdays where the wave reaches its peak between Sunday and Monday. Both are an approximation to Trace 5 in [25].

4 Regression between Samples of different Lengths

Typically, a regression problem requires pairs of dependent and independent variables, (y_i, x_{ij}), with $i = \{1, n\}$, $j = \{1, k\}$, where n denotes the number of observations, and k the number of independent variables. To find pairs of variables in cases where $x \in X$ and $y \in Y$ with $|X| = m$, $|Y| = n$ and $m \neq n$ it is exploited in this thesis that the associated empirical distribution functions, $F_m(x)$ and $G_n(y)$ are non-decreasing and that $F_m(x) := 0$ for $x < inf(X)$ and $F_m(x) := 1$ for $x > sup(X)$.

To find pairs of x and y, for some x_i, a y_j is determined, such that

$$y_j = G_n^{-1}(F_m(x_i))$$

where $G_n^{-1}(p)$ denotes the quantile function. Thus, pairs of x and y are determined by means of the QQ curve.

Quantile functions are only defined for continuous distributions. But measurement data is always discrete. Hence, to be able to determine $y_j = G_n^{-1}(p)$ for arbitrary values of p, y_j needs to be estimated. This is a problem that typically arises when QQ curves need to be plotted. In [17], Hyndman and Fan summarize and examine different approaches to quantile estimation that are implemented in statistical libraries. Commonly, the estimate of the quantile of order p is defined as

$$\hat{Q}(p) = (1 - \gamma)X_{(j)} + \gamma X_{(j+1)}$$

where $X_{(j)}$ denotes the j-th order statistic of X,

$$\gamma = p_k n + l - j$$

and p_k can be expressed as

$$p_k = \mathsf{M}(F(X_{(k)}))$$

where $n = |X|$, $j = \lfloor p_k n + l \rfloor$, $\lfloor q \rfloor$ denotes the largest integer smaller than q, $l = \alpha + p(1 - \alpha - \beta)$, α, β are some constants – typically in the range $[0, 1]$ – and $\mathsf{M}(\cdot)$ denotes the median. Since order statistics are known to be β-distributed, the crucial part is the estimation of the median of the incomplete β function ratio. The estimate by Hyndman & Fan in [17] seems not to be entirely correct since the definition the authors give coincides with the estimate of $E(X_{(k)})$ – the *mean* of the k-th order statistic – reported by C. Cunnane in [9]. It would only be justified to employ this approximation if the underlying distribution is normal since in this case mean and median are the same. According to [9], a value of $\alpha = 0.31$ was reported in [3][1] as approximation to $\mathsf{M}(F(X_{(k)}))$ and therefore these values are used in this thesis to estimate quantiles.

It is clear that estimates of distribution functions obtained in this manner will deviate from the actual distribution functions and a means to assess the goodness-of-fit is needed. Conventional goodness-of-fit (GoF) tests (2-sample Kolmogorov–Smirnov, k-sample Anderson–Darling, Carmér–von Mises) are, however, not appropriate since these tests are targeted at a fundamentally different problem: They are designed to test the hypotheses that two samples origin from the same distribution and provide merely a qualitative answer. But a quantitative answer is necessary to decide *how* good a particular model is. And most importantly, named GoF tests assess the difference between distribution functions along the y-axis. However, since in cases of less good fits there will be a deviation along the x-axis it is more meaningful to assess this deviation.

It appears that only K. Doksum (albeit in collaboration with others) has investigated this problem, yet with different intentions [15]. For two empirical distribution functions, $F_m(x)$ and $G_n(y)$ he defines the empirical shift function $\hat{\Delta}_N(x)$ as

$$\hat{\Delta}_N(x) = Y_{(\lfloor mF_n(x) \rfloor + 1)} - x$$

where $Y_{(i)}$ denotes the i-th order statistic of Y and $\lfloor k \rfloor$ denotes the largest integer smaller than k.

In lack of any other means this definition is employed to assess the deviation between the actual empirical distribution function and its estimate.

Also, to be able to compare relative performance of a model a scale-free error definition is needed where the classical mean absolute percentage error (MAPE) is probably the most widely used. In terms of the empirical shift function it is defined as

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\hat{\Delta}_N(x_i)}{x_i} \right| .$$

The MAPE, however, does not exist whenever $x_i = 0$ for some i. The value of the MAPE also inflates if small values are present in the sample of actual values. Alternative definitions or extensions of MAPE in [19] or [6] introduce merely other issues and do not present a reliable solution to the problem. Thus, a median version of the MAPE is defined additionally:

$$mAPE = 100 \cdot \mathsf{M} \left(\left| \frac{\hat{\Delta}_N(x)}{x_i} \right| \right) .$$

The $mAPE$ exists as long a less than 50% of the actual data is different from 0. If this is not the case either, a percentage error cannot be determined.

[1] The original resource could not be obtained.

5 Quantile Function-based Models

Modeling the resource utilization of applications is a recurring problem and a large body of work exists in this field. Among the considered body of work the results of Curino et al. [10] and of Desnoyers et al. [13] come close to what is needed for this thesis. But mere statistics (i.e. mean and modes) in [10] may hide relevant relationships between other parts of the distributions. And Desnoyers et al. admit a proportional dependency between the prediction error and the coefficient of variation, c_v, of the service time. Since for $c_v = 2$ the authors report a prediction error of 10%, and since for the workloads produced by our generator $c_v > 7$, we inferred that this error would be even larger in our case.

Modeling the power consumption is as well an old but nonetheless very active research field. If one abstracts from the target domain (e.g. single resources of a server, the server as a whole, or entire data centers) three kinds of models can be distinguished with respect to the parameter sources: (1) models that exclusively incorporate data provided by the operating system, (2) models that exclusively incorporate data provided by the hardware resources, and (3) hybrid models. While models of the first kind have the advantage to be easily portable to other systems (assuming that most server systems run a Unix-like operating system), models of the second kind promise to improve accuracy, yet at the cost of reduced portability. However, this promise is not always kept since even very simple models of the first kind that incorporate merely the CPU utilization [16] can outperform rich models of the second kind [14, 4]. This motivates to focus on models of the first kind.

5.1 Applicability of the Approach

The opportunity to obtain data points for regression problems (a) regardless of the size of the individual samples and (b) without the necessity that time stamps are present in the data seems very promising. The approach was first employed to evaluate the finding by Crovella & Bestavros [8] that the effect of file popularity seems only to have little effect on the request size distribution. Ten different workloads for the video server were generated and the quantiles of the request size distribution were estimated by the simple regression model

$$lg(\hat{Q}_S(q)) = \beta lg(Q_V(q)) + \alpha$$

where lg denotes the decadic logarithm, $\hat{Q}_S(q)$ the estimate of the request sizes quantile function, and $Q_V(q)$ the empirical quantile function of the video file sizes. With respect to the MAPE the regression coefficients $\alpha = 0.471$ and $\beta = 0.985$ minimized the error across the different workloads. The average MAPE across the different workloads is 1.583% which seems to support the finding. It was therefore concluded that the approach is in principle applicable and target to express the relationship between workload and single resources of the video server. During this process, several problems that gravely limit the general applicability of the approach are encountered.

5.2 Limitations of the Approach

The most severe problem is an apparent multicollinearity between the quantiles of request sizes and interarrival times; a relationship that was not initially anticipated. This restricts the approach to the estimation of resource utilization as function of request sizes and renders the third goal of this thesis unachievable.

Linear relationships between variables could not be expected for any of the resource utilization models and the power consumption models. This does not itself present a problem since to (re)establish a linear relationship, one typically transforms the data appropriately, or describes the nonlinear behavior of the variables such that the regression algorithm performs the transformation internally. However, the nonlinear relationships in some of the QQ curves are very complex and cannot be described by a single transformation. Figure 1 provides a typical example:

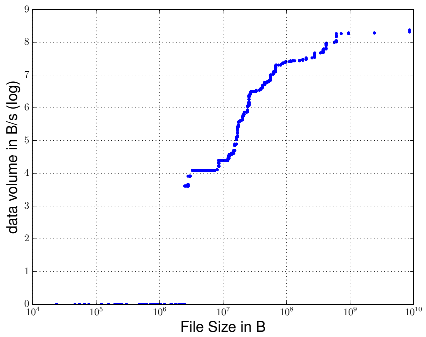

Figure 1: Example of a QQ curve of storage utilization vs. request sizes for the video server.

It shows the quantiles of the logarithmically transformed storage utilization in terms of written and read Bytes/s, versus the quantiles of the sizes of requested files. The constant line at the bottom left represents the portion with respect to the *observation period* that the storage utilization is zero. In the example this portion is 0.41. The request sizes, however, are not sampled with respect to time, but only if a request for a file reaches the server. Since the 0.41-quantile of the request sizes is $3.672 \cdot 10^6$ Bytes in this example it would mean that for all request sizes less or equal 3.672 MB the storage is not utilized. This distorts true relationships since every file is read at least once the first time it is requested. Generally, the QQ curve does not necessarily reflect the relationship between the true resource utilization and the (multiple) requests of a file with a certain size. The paramount reason is the different sampling domains for the request sizes (occurence) and the storage utilization (time). Admittedly, the initial idea to not to rely on timing information of the data is the very reason for the distortion. Additionally, the course of the QQ curve is less smooth the less distinct data points the samples of either of the two variables comprise. In these cases the QQ curve shows less, but larger steps, i.e. breakpoints, that cannot be related to properties of the resource or the request size distribution because of the distortion.

5.3 Models for Resource Utilization

In this situation a *tanh* function represents a compromise to approximately describe the course of the QQ curves: It captures at the same time parts of the curve with zero resource utilization, and parts where the resource appears saturated. How well the *tanh* function encompasses the central part of the QQ curve again depends on the smoothness of the curve (i.e. how many distinct data points the samples of both parameters comprise) and which quantiles are mapped against each other – which is, to repeat, subject to distortion.

E.g. in case of the video server the central part of the QQ curve for the network utilization follows an approximately logarithmic course. This is due to an exponential course of the empirical distribution function (EDF) of the network utilization in the upper quantiles and an approximately uniform (i.e. linear) course of the EDF of logarithmically transformed request sizes in the quantiles of the same orders. For this particular case the *tanh* function encompasses most of the points on the QQ curve. For the storage utilization the *tanh* encompasses less points in cases with low request rates. With increasing request rate more data needs to be fetched from the storage and the EDF of the storage utilization shows a noticeably exponential course and more points on the QQ curve are encompassed by the *tanh* function.

But the estimation errors for all resource utilization models for the video server and the transcoder depend not only on how well the *tanh* transform describes the course of the QQ curves. Due to the sensitivity of the $MAPE$ and the $mAPE$ to small values in the measurement data comparatively large errors can be obtained despite the course of the QQ curve is well described by the *tanh* function. E.g. for the video server the average $mAPE$ for the network utilization model is 4.593% across test cases. This is due to the *tanh* function describes most of the QQ curve well *and* the actual data contains only values greater than 10. Contrary, for the CPU utilization model of the video server the actual data contains nearly only values less than 1 and the average $mAPE$ across the test cases is 37.977%. Table 1 summarizes the obtained percentage errors for the video server and the transcoder.

	$mAPE$
Network	4.593
Storage	23.245
CPU	37.977

(a) Video Server

Target Container	FLV	MKV
	$mAPE$	$mAPE$
Network	100.881	100.072
Storage	n.e.	n.e.
CPU	49.667	56.729

(b) Transcoder

Table 1: Percentage errors for the resource utilization models

5.4 Models for Power Consumption

For the power consumption models both variables of the QQ curves, resource utilization and power consumption, are sampled over time, but with different frequencies: While the resource utilization, i.e. CPU utilization, is sampled with 1 Hz, the power consumption is samled with a frequency between 7 and 10 Hz. Therefore, the relationships in these QQ curves are also distorted. Hence, while a logarithmic transform seems appropriate to describe the course of the QQ curve this does not allow any statements about the true relationship between the resouce utilization and the power consumption. For the video server the average $MAPE$ for the power consumption models across test cases is 3.141% and for the transcoder it is 2.656% in case of the FLV target container and 3.361 in case that the target container is MKV.

Additionally to the video server and the transcoder six benchmarks from the SPEC CPU 2006 suite are considered. In comparison to the video server and the transcoder, the QQ curves for the power consumption versus CPU utilization exhibit more emphazised steps in case of the benchmarks. This is due to the *on-off* usage pattern of the benchmarks that approximately either utilize a CPU core to its full extent or not at all. Nevertheless, the logarithm generally describes the overall course of the QQ curves and the percentage errors of the models for all benchmarks are comparable to those of the models for the video server and the transcoder. The respective percentage errors are summarized in Table 2 below:

	mAPE
`cactusADM`	7.316
`milc`	2.091
`povray`	2.738

(a) Floating point benchmarks

	mAPE
`gcc`	4.209
`libquantum`	5.951
`xalancbmk`	1.813

(b) Integer benchmarks

Table 2: Percentage errors for the power consumption models for selected benchmarks.

6 Summary

To recapitulate, this thesis comprises two parts. In the first part, a workload generator was developed that provides a video server with requests that shall replicate statistical properties of actual traffic to current video platforms. It must be noted (a) that no literature is available that covers all relevant properties for one single video platform and (b) that some of the relevant literature is already outdated. Therefore, the generator combines properties that are the result of the investigation of multiple authors if recent literature is available. Where that is not the case, an educated guess concerning the reasonable development of properties is made. In the last resort properties of common internet traffic are employed.

In the second part, models for the resource utilization as function of the workload of a video server and of a transcoder application shall be developed. To generate the workload, the generator from the first part is employed.

To tackle the problem that measurement data is taken from different sources – all with dissimilar sampling frequencies – an approach based on quantile-quantile (QQ) curves is proposed that enables regression between samples of different lengths using the relationship between quantiles. While the approach, in some cases, allows to express functional relationships between variables where otherwise no explicit relationship could be formulated, it is in general not an appropriate means to establish models for the resource utilization as function of the workload. Two principal reasons can be identified. The first is an apparent, unexpected linear dependency between the quantiles of the request size and the request rate (i.e. interarrival time) components of the workload. Therefore, the models only respect the request size component and time-related dynamics of the request size and its effect on resource utilization are not incorporated. The second reason is that a p-quantile of the request size does not necessarily coincide with the quantile of the same order of the utilization of a particular resource: While values of the request size are reported only if requests arrive the system, the latter are sampled continuously and thus, also if no requests are processed. This distorts the true relationship between request sizes and resource utilization, complicates the formulation of general models, and makes human intervention inevitable. The different sampling frequencies for resource utilization and power consumption cause also distortion in the QQ curve between these two variables. Therefore, the approach is also not suitable to derive models for the power consumption as function of resource utilization.

References

[1] Paul Barford and Mark E. Crovella. Generating representative web workloads for network and server performance evaluation. *ACM SIGMETRICS Performance Evaluation ...*, 26(1):151–160, June 1998.

[2] Luiz André Barroso and Urs Hölzle. The Case for Energy-Proportional Computing. *Computer*, 40(12):33–37, December 2007.

[3] Leo R. Beard. Statistical Methods in Hydrology. 1962.

[4] Ramon Bertran, Yolanda Becerra, David Carrera, Vicenç Beltran, Marc Gonzàlez, Xavier Martorell, Nacho Navarro, Jordi Torres, and Eduard Ayguadé. Energy accounting for shared virtualized environments under DVFS using PMC-based power models. *Future Generation Computer Systems*, 28(2):457–468, February 2012.

[5] G Chen, W He, J Liu, S Nath, and L Rigas. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, number 1 in NSDI'08, pages 337—-350, San Francisco, California, May 2008. USENIX Association.

[6] Zhuo Chen and Y Yang. Assessing forecast accuracy measures. pages 1–26, 2004.

[7] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Statistics and Social Network of YouTube Videos. *2008 16th Interntional Workshop on Quality of Service*, pages 229–238, June 2008.

[8] M.E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[9] C. Cunnane. Unbiased plotting positions — A review. *Journal of Hydrology*, 37(3-4):205–222, May 1978.

[10] Carlo Curino, Evan P C Jones, Samuel Madden, and Hari Balakrishnan. Workload-Aware Database Monitoring and Consolidation. In *Proceeding of the SIGMOD'11 conference*, pages 313–324, Athens, Greece, 2011.

[11] Reetuparna Das, Rachata Ausavarungnirun, Onur Mutlu, Akhilesh Kumar, and Mani Azimi. Application-to-core mapping policies to reduce memory system interference in multi-core systems. In *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, pages 107–118. IEEE, February 2013.

[12] Christina Delimitrou and Christos Kozyrakis. Quasar: Resource-Efficient and QoS-Aware Cluster Management. In *Proceedings of the Nineteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Salt Lake City, UT, USA, 2014.

[13] Peter Desnoyers, Timothy Wood, Prashant Shenoy, Rahul Singh, Sangameshwar Patil, and Harrick Vin. Modellus: Automated modeling of complex internet data center applications. *ACM Transactions on the Web*, 6(2):1–29, 2012.

[14] Gaurav Dhiman, Kresimir Mihic, and Tajana Rosing. A system for online power prediction in virtualized environments using Gaussian mixture models. In *Proceedings of the 47th Design Automation Conference, DAC '10*, number 3, page 807, New York, New York, USA, 2010. ACM Press.

[15] Kjell A. Doksum. Some graphical methods in statistics: A review and some extensions. *Statistica Neerlandica*, 31(2):53–68, June 1977.

[16] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. Power provisioning for a warehouse-sized computer. *ACM SIGARCH Computer Architecture News*, 35(2):13, June 2007.

[17] Rob J. Hyndman and Yanan Fan. Sample Quantiles in Statistical Packages. *The American Statistician*, 50(4):361–365, November 1996.

[18] Huan Liu. Host server CPU utilization in Amazon EC2 cloud, 2012.

[19] Spyros Makridakis and Miche'le Hibon. The M3-Competition : results , conclusions and implications. 16:451–476, 2000.

[20] Siddharth Mitra, Mayank Agrawal, Amit Yadav, Niklas Carlsson, Derek Eager, and Anirban Mahanti. Characterizing Web-Based Video Sharing Workloads. *ACM Transactions on the Web*, 5(2):1–27, May 2011.

[21] Christoph Möbius, Waltenegus Dargie, and Alexander Schill. Power Consumption Estimation Models for Processors, Virtual Machines, and Servers. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–1, 2013.

[22] Dejan Novaković, Nedeljko Vasić, Stanko Novaković, Dejan Kostić, and Ricardo Bianchini. DeepDive: Transparently Identifying and Managing Performance Interference in Virtualized Environments. Technical report, École Polytechnique Fédérale de Lausanne, 2013.

[23] Charles Reiss, Alexey Tumanov, Gregory R Ganger, Randy H Katz, and Michael A Kozuch. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *Proceedings of the Third ACM Symposium on Cloud Computing - SoCC '12*, pages 1–13, New York, New York, USA, 2012. ACM Press.

[24] Nedeljko Vasić, Dejan Novaković, Svetozar Miučin, Dejan Kostić, and Ricardo Bianchini. DejaVu: accelerating resource allocation in virtualized environments. In *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS '12*, page 423, New York, New York, USA, 2012. ACM Press.

[25] Michael Zink, Kyoungwon Suh, Yu Gu, and Jim Kurose. Characteristics of YouTube network traffic at a campus network – Measurements, models, and implications. *Computer Networks*, 53(4):501–514, March 2009.