

On the Reliability of Service Level Estimators in the Data Centre

Jon Henrik Bjørnstad and Mark Burgess
Oslo University College, Norway

Abstract—The ability to predict achievable service levels is an important part of business strategy in drawing up Service Level Agreements (SLA). Service delivery is a dialogue between supply and demand. Recently it has been argued that adaptive service levels might be predicted using classical queueing theory combined with a data centre model. Using generic tools, we investigate the accuracy with which service levels can be predicted for a single server Web service, based on expected traffic patterns, and hence we identify approximate limits on the reliability for service levels.

I. INTRODUCTION

Service provision is a delicate interplay between supply and demand. The ability to deliver reliable service levels depends on both the resources that are available in the data centre and the pattern of demand driven by the users. In this paper, we try to understand how the incoming demand for a service affects one's ability to formulate rational Service Level Objectives and hence deliver the appropriate supply response to the customers with appropriate Service Level Agreements (SLA).

Over the past decade, a lot of attention has been given to the inter-arrival time distributions in traffic, after it was noticed that traditional assumptions about queue arrivals were not true for the time-scales over which traffic was being observed. This observation has generated a lot of mathematical and theoretical work but this work is not clearly applicable to the matter of service provision in the data centre. One would like to know exactly what engineers in the data centre should do to cope with demand.

Historically one has used models like queues[1] and inventories[2] as estimators for capacity planning. These models are steady state models that do not easily account for changing load, such as one might expect in a real environment. Nonetheless, they are the only real models one has that can be used to describe random phenomena in a service framework. So how useful are they? And if they are not useful, can they be adapted to yield better results that are of use to service managers?

We are particularly interested in what practical lessons can be learned from the traffic characterizations which have both been observed and discussed theoretically. In particular, there is a need for a semi-empirical procedure for finding service levels which does not require access to trunk link data from major ISPs, but rather can be performed on site by any data centre. So we consider the following question: suppose our resources are fixed, what supply-margin ("over-provision") of service is required to be within a predictable margin of an SLA target? For example, when can we say that we are 95%

certain of being able to deliver an SLA target level 80% of the time? Or better still: what is the full probability distribution for service within a given time threshold?

II. QUEUE PROCESSING

We begin our discussion of service level estimators with queues. A queue is the interface between an arrival process and a completion or renewal process[3], [1], [4]. Queues are categorised by the type of random process that delivers requests, the type of process for dealing with the arrivals and the number of independent servers that participate in the load sharing. A deterministic distribution has constant inter-arrival times, with no variance, and 'general' means that the model's results apply for any distribution. The term *memoryless* implies that the arrival process is a steady state process, in which the current state of the distribution does not depend on what happened in the past. These distributions are particularly simple to discuss and occur in a variety of real situations, but they are not ubiquitous.

It is common to have several, sometimes many, servers in parallel, sharing the load of incoming traffic. Indeed modern operating systems and CPU architectures are designed with multi-threading in mind, so the simple view taken by basic queueing models is far from being an accurate reflection of task scheduling. However, we shall deal only with single server models in this work. There are two reasons for choosing single server models:

- We need to have some separation of issues in understanding the service provision process. By considering only single server queues we eliminate part of the complexity of the service process (though not entirely, since single server systems remain intrinsically multi-threaded). The issue of multiple servers is considered in refs. [5], [6].
- It has been argued that a single queue model is the right way to model a multi-tiered system, since there is invariably a bottleneck stage in a system which throttles the performance and dominates the behaviour, e.g. the load sharing dispatcher[7], [8].

Even with a formally 'single' queue, there is room for parallelism. The serving discipline is an additional caveat in the server processing options. By default, queues are assumed to process in a First Come First Served (FCFS) manner, in which case each job is completed before the next is begun. If one allowed Processor Sharing (PS), or multi-threading, then there is a possibility to interleave jobs on a single server (see fig. 1).

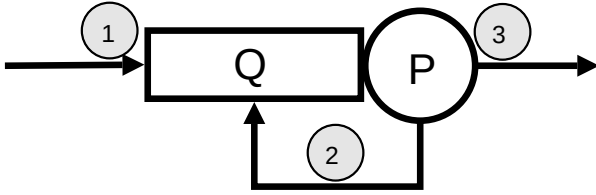


Fig. 1. Model of a simple processor sharing $G/G/1$ queue. Jobs are shifted between a processing and a queueing state until it is completed. Figure based on ref. [9].

The processor sharing approach is a more realistic model of how a real web server works. There are multiple connections being serviced simultaneously as multiple tasks, in spite of there being only one processor on a server in a typical Apache-like web server.

In spite of the need for parallelism in coping with service times, the fundamental limits (minimum response time etc) are properties of the weakest link, i.e. a single server. Thus it is important to study the behaviour of a single server queue in order to gauge the limitations of service quality. We consider the effect of multiple servers elsewhere[5].

III. GOALS

The plan for our paper is as follows. We consider some typical quantities that are used in Service Level Agreements,

- Mean response time (seconds).
- Mean data rate (bits per second).

These are quantities that can be calculated from queueing theory[10] using a variety of models for arrivals and processing[10], [3]. The simplest estimates for the M/M/1 queue are trivial, however it is known that web traffic is unlikely to have an exponential Poisson arrival form in the short term so we must also deal with approximations to traffic patterns that more closely model the features of observed traffic. The Poisson case remains important, however, since it is a fixed point with ideal properties and it is easy to compute.

We can study the expected performance by hand-calculation, using the queueing models, and also by direct traffic generation measured against a real web server (a semi-empirical study). We would like to know:

- Over what time interval (if any) do we expect to see a predictable mean value?
- What processing margins are needed to guarantee a certain mean value?

These goals are actually quite ambitious, and are subject to all kinds of uncertainties in an end-to-end system, so we attempt to isolate a single component of the uncertainty, namely the server process itself. To do this, our experiment consists of a direct connection between client and server, simulating an ISP connection with a fixed, large capacity so that the connection is not a factor in the uncertainty. Moreover, we fetch fixed size data objects in every transaction so that only time-measurements are required. The uncertainty in average data rate $R = D/T$, where D is data size and T is time is

given from error theory as

$$\Delta R = \sqrt{\left(\frac{\partial R}{\partial D}\right)^2 (\Delta D)^2 + \left(\frac{\partial R}{\partial T}\right)^2 (\Delta T)^2}. \quad (1)$$

By our assumptions $\Delta D = 0$, hence this reduces to

$$\frac{\Delta R}{R} = \frac{\Delta T}{T}, \quad (2)$$

i.e. the fractional uncertainty (percentage) in rate is the same as that in the time measurements. This result is easily derived from dimensional analysis and is therefore independent of a Gaussian or symmetric peak assumption.

Although we are not using real traffic, our results are nevertheless relevant for the purpose of probing the limits of the system (statistical estimates). We require an estimate of the apparent steady-state performance over a time-scale of hours and minutes (since this is the time-scale relevant to users). To compare traffic with different statistical properties in both theory and experiment is a non-trivial exercise and some calibrations of characteristic values are required. We shall show below that these calibrations can be made well within the accuracy we required for our conclusions.

We close this section by noting acknowledging some limitations. By now everyone knows that the answer to traffic modelling and service is complicated. We are not looking for a generic theory that captures the detailed statistical properties of traffic. Many have tried this and little has come of it in practice. Rather, we are looking rather for simple order of magnitude approximations, estimators that are usable by a data centre engineer. If such things exist, they are of great value. If they do not exist, there are dire implications for the authors of Service Level Agreements.

IV. TRAFFIC PATTERNS

What one guesses from classical queueing theory is that the occurrence of non-Poissonian traffic patterns would lead to greater difficulties in predicting and handling server load. However, coming up with a simple rule of thumb has not been easy due to the failure of simple queueing models.

In the late 1980's and early 90's it became clear that the classical Poisson arrivals assumptions which worked so well for telephone traffic were inadequate to describe Internet behaviour. Leland et al. [11] showed that Ethernet traffic exhibited self similar characteristics, later followed up in ref. [12]. Paxson et al. [13] found that application layer protocols like TELNET and FTP were modelled quite well by a Poisson model. However, the nature of data transfer proved to be bursty with long-range dependent behaviour.

Web traffic is particularly interesting. Web documents can contain a variety of in-line objects such as images and multimedia. They can consist of frames and client side script. Different versions of HTTP (i.e. version 1.0 and 1.1) coexist and interact. Implementations of the TCP/IP stack might behave slightly different, etc. depending on the operating system. This leads to great variability at the packet level[14]. Refs. [15], [16] have suggested several reasons for the traffic characteristics.

In spite of the evidence that for self-similarity of network traffic, it is known that if one only waits for long enough, one should see Poisson behaviour[17]. This is substantiated by research done in [18] and elsewhere, and is assumed true in measurements approaching the order of 10^{12} points. However this amounts to many years of traffic[19].

V. PREVIOUS QUEUE MODELLING

Queueing theory on network traffic has traditionally been dealt with by Poisson modelling of arrival and service processes. The main reason for this is the analytical simplicity and attractive computational properties of such distributions. However, in literature it has been indicated that such modelling can lead to a overly optimistic queueing performance.

Queueing performance is strongly affected by the inter-arrival time behaviour of network traffic. For routers, bursty traffic fills up queueing buffers routers with packets waiting to be processed. Traffic bursts arriving at web server systems can cause request waiting queues to grow significantly. In general, it was noted in [20] that the presence of self-similarity, with large values for the Hurst exponent, network traffic can have severe performance degrading effects for networks with significant buffering, leading to poor Quality of Service.

Erramilli et al. [21] studied the effects of long range dependent traffic on queueing systems and found that long range dependence had measurable and practical impact on queueing systems. They also found that the nature of network traffic had significance for a number of packet traffic engineering problems, such as buffer sizing, admission control, and rate control. It was also noted that if traffic characteristics are ignored it can lead to overly optimistic performance predictions, and hence lead to an under-provision of service capacity.

Other research has studied queues with application to specific server operation. In [9] the operation of the omnipresent Apache web server software was modelled. Apache can can operate with several processes running at once, several threads running once, or with a combination. Therefore, modelling regular, one processor web system job scheduling as *First Come First Served*(FCFS), an assumption a lot of simple queueing theory relies on, is not correct. Also pointed out in by this work was that neither service or arrival time distribution is necessarily Poissonian distributed. A sophisticated model taking general service times and number of processes/threads running in processor sharing fashion was therefore derived - the $M/G/1K(PS)$ queue. The purpose of this model was to study the blocking probability of the web server, i.e. the rejection probability for requests given a certain HTTP listen queue size. Results from simulation with the model and experimental results with Poissonian arrivals and general service time distribution showed to be in well agreement with each other. In summary, we must consider long-tailed behaviour in estimators.

A. Server performance modelling

To achieve a desired level of quality of service under any traffic conditions, extensive knowledge about how web server hardware and software interacts and affects each other is

required. Knowing the expected nature of traffic and queueing system only tells us how we might expect a system to perform; knowledge about software and hardware interaction enables performance tuning.

Slothouber [22] derived a simple model for considering web server systems founded on the notion of serial queues. Several components interact during a web conversation and a goes through different stages, with different queues at each stage. The response time is therefore an aggregate of the times spent at different levels within the web server.

Mei et al. [23] followed this line of reasoning in a subsequent paper. One of the focal points in their work was to investigate the effects of for response time and server blocking probability due to congestion in networks. High end web servers usually sit on a network with excess bandwidth to be able to shuffle load at peak rates. However, customers inherently has poorly performing networks with with possible asynchronous transfer mode, such as ADSL lines. Therefore, returning ACKs in from client to server can be severely delayed. In turn this causes the request to linger for a longer time in the system than would be the case if the connecting network was of high quality. Aggregation of such requests could eventually cause the TCP and HTTP listen queue to fill up, making the server refuse new connection requests even if it is subject to fairly light load.

These models are simple and somewhat unrealistic, but possess the kind of simplicity we are looking for for our comparisons.

VI. THEORY

A queue with general inter-arrival and service time distribution is denoted as $G/G/1$. We shall use this as a basis for comparing theory and experiment. No easily usable exact result exists for these queues due to the difficulty in determining the transition probabilities between queueing states [24]. However, several approximate upper bounds have been proposed for queue length and average waiting time, in the FCFS discipline. They apply in *heavy traffic approximations* and are valid for $\rho \mapsto 1$ [24]. One of these is discussed in ref. [25] and has the following form.

$$E[W] \approx \frac{\rho E[S](C_S^2 + C_A^2)}{2(1 - \rho)} \quad (3)$$

As the total system time is a combination of both queueing, or waiting, time and service time, the upper bound for response time becomes:

$$E[T] \approx \frac{\rho E[S](C_S^2 + C_A^2)}{2(1 - \rho)} + E[S] \quad (4)$$

where S is the service time, A is the arrival time and $C_x \equiv \sigma_x/E(x)$ is the scaled deviation. For exponentially distributed inter-arrival time it is fairly easy to see that we get the response time formula based on the well-known Pollaczek-Khinchin equation[1]. For such inter-arrival times, $C_a^2 = 1$ as $\sigma_a^2 = (E[A])^2$. This follows from standard deviation and mean of exponential distributions. With an exponentially distributed service time, this formula is equal to the well-known result for $M/M/1$ queues. Thus, for all queue constellations with

exponentially distributed inter-arrival times, formula 4 is said to be exact.

Another way of doing heavy traffic approximation is presented in [26] and is derived from work done in [10]. It is similar to the one presented here and yields approximately the same results.

Owing to the difficulties in dealing with long-tailed traffic patterns, we do not expect to find a perfect model for describing service provision. We are looking mainly for universal features of the available queueing models, which can lead us to semi-empirical and *simple* rules for scaling of services. Our aim is to exploit the assumed relationship between predictability and the ability to make reasonable claims:

$$\text{QoS predictability} \Rightarrow \text{SLA reliability.}$$

VII. EXPERIMENTAL TESTING

We consider the following problem: suppose data centre resources are fixed (i.e. the service characteristics and rate S, μ are determined from CPU and disk performance rates and constant), can we predict what rate of service completions we can safely promise to a client, given that the character of the incoming traffic could vary over the time of the contract? The procedure must be implementable by any potential application service provider and must cope with worst-case events.

To accomplish this, we base results on *generated* traffic, but use it to drive a real hardware back-end. We claim that this is sufficient to test the interesting issues that are of interest to a data centre manager. Results are measured for an Apache web server running on a GNU/Linux PC computer. The client generator is connected directly to the server by null-modem cable to avoid uncertainties due to network switching.

Traffic is initiated using a traffic generating program which simulates inter-arrival distributions, generating a stream of page-lookups which are passed to a PHP-enabled web server. Each page involves CPU-bound iterations (spin delay) to ensure that the server is significantly loaded ; this was necessary to offer the server a difficult load as might be generated by a large number of users. Such a load is not easily generated in a laboratory setting, as the server CPU hardware was comparable to that of the traffic generator. To make sure that this compensation was an adequate facsimile of true load, we verified that the relationship between iterations and load was linear (see fig. 2).

The slope of the curve represents the time requirement per iteration, and the constant represents the minimum delay. These parameters are used to tune an expectation value for the service time.

We have compared the behaviour of our server to FCFS queueing models for traffic types ranging from exponential Poisson inter-arrivals to moderate Pareto traffic[27]. Sample results are shown in fig. 3. The models weaknesses are already apparent.

A noticeable trend in the data was the low response time for experimental results for larger service distribution variance. The highest response times of the hand simulations are obtained with Pareto distributed service time with $\alpha = 2.01$. Corresponding experimental values are strangely amongst the

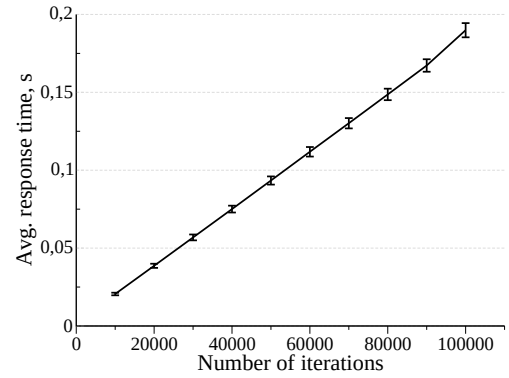


Fig. 2. Calibration of the service time. We see a linear response, hence this will not distort the simulation.

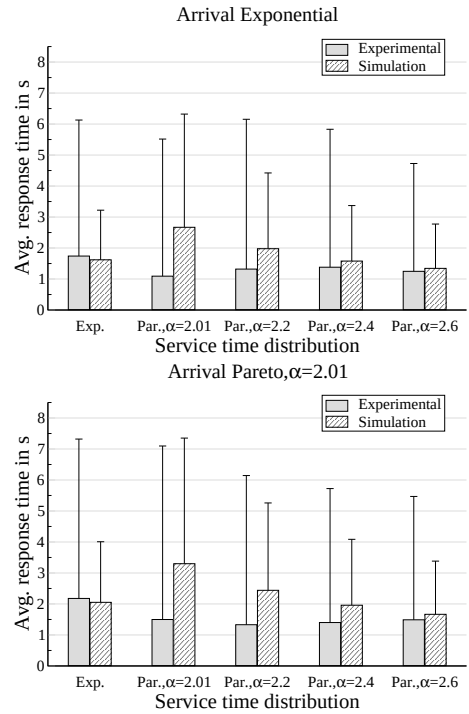


Fig. 3. Mean response time and standard deviation for experimental and hand simulation results. (a) Exponential inter-arrival, (b) Pareto inter-arrival w/ $\alpha = 2.01$, with a variety of server distributions. Note how the standard deviation is significantly larger than the mean, as is typical of tailed distributions.

lowest response times. Additionally, the experimental values has fairly large standard deviations, which is indicative of distribution heavy tails. These phenomena are related to the fact that the serving discipline in a computer server is not FCFS.

Unlike the FCFS discipline assumed by formulas and present in the hand simulation system, computers in general process jobs using processor sharing and priorities. For a system with exponential processing time it is noted in ref. [10] that response time variance is potentially much greater in a Processor Sharing system using round robin, than in a FCFS processing system. The reason is quite intuitive. A job in a system with *exponentially* distributed service times and inter-arrival times ($M/M/1$ queue) whose service requirement is

of average size will, on average, have the same response time independent on whether or not it is a processor sharing or a FCFS system. However, jobs with less than average service time will be better treated in a processor sharing system than with FCFS serving discipline. Vice versa, jobs with greater than average requirement will receive poorer treatment. The *breakpoint* defines the value service time that on average gets the same treatment in an FCFS queue as a processor sharing queue.

A probability distribution for experimental and simulated $M/M/1$ queue are shown in fig. 4. The figure illustrates the differences in response times for FCFS systems and PS (processor sharing) systems. Processor sharing possibly causes the span of response time values to be much larger than in a FCFS system, which in turn manifests itself through higher response time distribution variance (truncated in the figure).

Probability distribution, A: Exp., S: Exp.

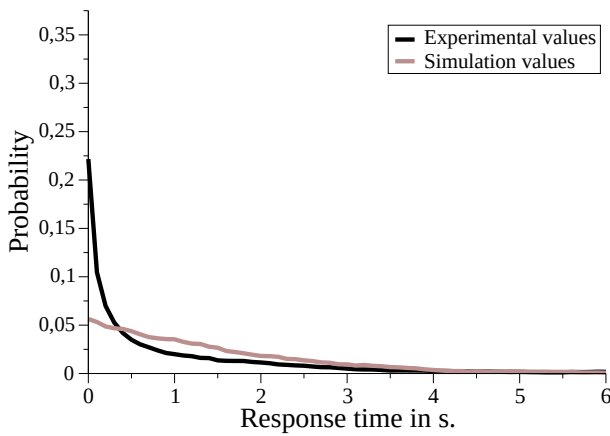


Fig. 4. Probability distribution for experimental and simulated $M/M/1$ queue. The figure is meant to illustrate the differences in response time tail for FCFS systems (simulation) and processor sharing systems (experiment). Processor sharing possibly causes the span of response time values to be much larger than in a FCFS system. The curve has a long tail which has been truncated in the figure.

It should be possible to track differences in degree of response time distribution heavy tailedness as can be seen in the figure. It can therefore be argued that the response time of the web server system does not have a well defined mean and therefore lacks a typical value.

A. Self-similar traffic

For long range dependent distributions no simple formula for calculating the expected response time exists. Our objective was therefore only to investigate performance differences through simulations and experiments.

As before, the measured inter-arrival times were extracted in order to check whether the statistical properties of the distributions were conserved. More specifically, the inter-arrival distributions were checked for self-similarity using the SELFIS tool. The results showed that self similarity was correctly reproduced in our experiment.

Breakpoint count, $M/G/1$

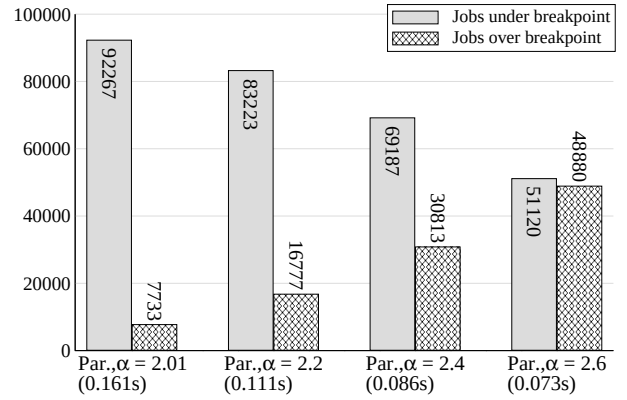


Fig. 5. Figure summarizing count of service requirements falling under and over breakpoint values derived using formula from [10]. The breakpoint values are listed in parenthesis and given in seconds.

VIII. SERVICE GUARANTEES

With the variances seen in the experimental simulation, it is clear that it is nearly impossible to make a meaningful promise about the average service level to a customer. The mean value is insignificant given the breadth of the distributions: what we need to look at is the complete distribution of values.

A. Variability

The experimental results listed previously are dominated by a large data set spread. Compared to results from simulation, the variance can be many magnitudes larger for experiments. The likely reason for the difference in behavior is the scheduling discipline.

For $M/G/1$ systems the breakpoint between better and worse processing is for jobs with service time requirement x with value $x = \overline{x^2}/2\overline{x}$ [10]. For the queue constellations with exponentially distributed arrival time we can thus find the breakpoint and investigate how many of the service requirements fall under and over this. The result of this procedure is depicted by figure 5. The figure clearly shows and predominance of jobs falling under the breakpoint. It should be noted that the dataset sizes might be too sparse for the breakpoint formula to be valid. However, if this approach is appropriate in this case, the vast amount of jobs under the breakpoint could explain the overall decrease of response time going from simulated to experimental $M/G/1$ queues. It is believed that the same phenomenon partly can explain the differences seen in the rest of the queueing $G/G/1$ constellations. We were not able to find comprehensible breakpoint formulas for these cases.

Thus, processor sharing possibly causes the span of response time values to be much larger than in a FCFS system, which in turn manifests itself through higher response time distribution variance. It should therefore be possible to track differences in degree of response time distribution heavy tailedness.

From the barcharts 3 one can see a clear trend. Service time characteristics seems to have less impact on the response time than the inter-arrival time characteristics. As inter-arrival

time distribution becomes more and more variable for exponential service time, the experiments and simulations show that response time increases. However, the average response time for simulated and experimental $G/M/1$ queues are fairly coherent. There are slightly higher values for the experiments as could be expected since this is not a protected environment as the simulations are. The queueing formula 4 makes no distinction between arrival and service process characteristics which is obviously not accurate for web server systems according to our results.

In a system dominated by such uncertainty, dealing with mean response time for capacity planning and evaluation can easily lead to wrong conclusions about performance. The large variance and the long tail in experimental values are indicative that FCFS modelling of web servers, as the queueing models rely on, does not capture crucial characteristics of the response time. On basis of the probability distribution plots, we therefore argue that classical FCFS modelling is not theoretically motivated for actual web server systems. Consequently, the classical queueing models are not applicable, in the manner in which they were intended. We note an alternative approach using a modified server controller to pre-stack the arrival distribution in ref. [8].

The principal of parsimony says that, under such circumstances, the simplest model that accounts for the wide variance is adequate. One could therefore use an ‘ill motivated model’ as an effective predictor, as is often the case.

The worst case scenario is when we have a high variance inter-arrival time distribution and a regular exponential service requirement. We suggest a procedure that assumes highly variable inter-arrival times and exponential service times for successful capacity planning. However, self-similarity complicates matters, as bursty traffic also affects response time. The self-similar experiments show that inter-arrival time distributions with the exact same mean and standard deviation, but different degree of self-similarity, results in different response time behaviour. Simply considering the first two moments in doing capacity planning when the system is subject to traffic with a high degree of self-similarity might therefore lead to insufficient capacity.

B. Service Level Distributions

We have attempted to give reasonable explanations to the differences in observed values for response time from experiments, simulations and theoretical formulas. Yet to be addressed is the significance of the observed phenomena for service providers.

Due to the fact that both arrival and service processes are stochastic processes without a well-defined mean, combined with the effect of processor sharing has on response time tails, there is an intrinsic lack of typical values. Instead, it is more fruitful to consider the likelihood of being within a certain agreement. For instance, one could consider 80% or 90% compliance of a Service Level Objective. Depending on the nature of traffic the differences between these two levels can be quite significant.

This becomes quite apparent if we plot the Cumulative Distribution Functions (CDF) for the experimental and simulated

response time distributions (see figures 6). Because of the large deviations of values, going from 80% to 90% is not trivial from a service providers point of view. The CDF plots show that going from 80% to 90% in Service Level confidence means, more or less, a double of the service level target.

Thus the proper way to quote service levels is through distributions. The mean response times and rates are useless, even on the time scale of hours. A cumulative probability plot of service times is straightforwardly derived from the experimental results and gives the most realistic appraisal of the system performance. The thick cut lines in the figure show the 80% and 90% levels, and the corresponding response times associated with them. From this one can make a Service Agreement based on probably compliance with a service time, i.e. not merely expectation value but histogram..

For instance, one could consider 80% or 90% compliance of a maximum service level target. Depending on the nature of traffic, the differences between these two levels can be quite significant. Since customers are interested in targets, this could be the best strategy available to the provider.

IX. OVER-CAPACITY PLANNING

Single server studies are the key to understanding capacity planning. While it is true that one can apply load balancing to increase parallelism in multiple task service[5], parallelism cannot help the intrinsic performance of a single task. Each individual transaction takes place on a single server. Thus our study concerns the limits of what can be promised to a single customer for a single request. (It would be prudent in Service Level Agreements to distinguish between promises made about response times for single tasks, and promise made about task volume per unit time.)

We note that the worst scenario, in our results, is when we have a high variance inter-arrival time distribution and a regular exponential service processing. This only really makes sense in the FCFS scenario, and so it is a limiting case and is a possible way of capacity planning. If data centre unit-capacity is based on this limited case, there will always be a calculated over-provision with respect to a server capacity in practice.

In other words, we will be able to cope with the widest possible range of SLA promises, given fixed processing capacity. For FCFS modelling of web servers, SLA compliance in any case is given by assumptions of high variance in both inter-arrival and service time. Moreover, capacity planning based on the latter will result in overprovisioning also for the web server. Indeed, considering the web server to be a FCFS system will enable compliance of SLA's and might therefore be a good strategy for performance evaluation to meet SLAs even though this can possibly result in significant excess capacity. However, this presupposes the absence of inter-arrival self-similarity, which could make it worse again, so the margin might be necessary.

Our best predictive prescription (with noted flaws) is therefore roughly as follows:

- 1) Calculate the server rate μ for the weakest server that

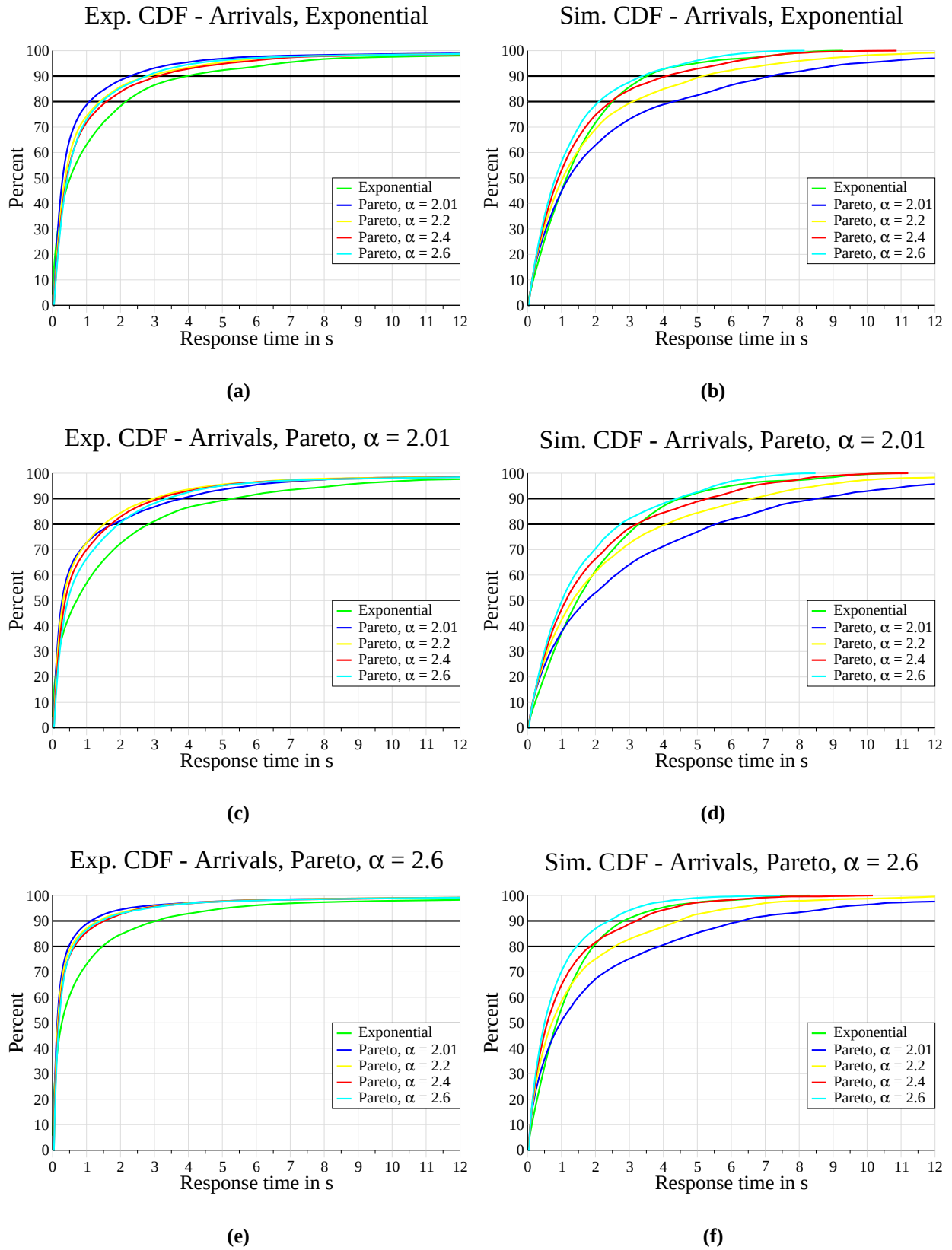


Fig. 6. The cumulative percentage distribution for response processing for sample exponential and Pareto arrivals. The thick cut lines show the 80% and 90% levels, and the corresponding response times associated with them. The figures in each row show the comparison between experiment (on the left) and theory (on the right). The theoretical values are too pessimistic here, which is good in the sense that it would indicate a recommended over-capacity, but the value is surely too much.

might handle a request. e.g. for a CPU bound process

$$\begin{aligned}\mu &= \text{av. job size} \times \frac{\text{av. instructions}}{\text{job size}} \\ &\times \frac{\text{CPU cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycles}} \\ &= \frac{\text{av.instructions} \times \text{RISC/CISC ratio}}{\text{MHz}} \quad (5)\end{aligned}$$

- 2) Estimate traffic type at maximum load.
- 3) Construct the Cumulative Probability Distribution as a function of response times using FCFS.
- 4) Make SLA promises about probability of transaction rate based on where the curves cross basic thresholds.

With this recipe, one overestimates service promise times to achieve a given level of certainty (or equivalently overestimates required capacity for a given time) in a non-linear way — the amount can be anything up to four hundred percent! This explains perhaps why most sites have seemingly greatly over-dimensioned web-servers. Given the problems that can occur if a queue begins to grow, and the low cost of CPU power, this over-capacity might be worth the apparent excess.

The results here indicate that one should perhaps be cautious about exactly which parameters are promised in a service agreement. Mean data rates and service times are highly volatile. If one will specify a maximum acceptable service time, one must expect to over-dimension server capacity accordingly.

X. SUMMARY AND DISCUSSION

We have used queue modelling and experiment with a variable traffic generator to provide a generic traffic source to test a web server's performance, with an aim to investigate the predictability of SLA compliance. We find that classical First Come First Served queueing formulae do not model traffic and lead to an over-estimated response time, or required capacity[28]. Regardless of incoming traffic type, the web server system is dominated by high variance in response time. Thus to meet Service Level Objectives, a provider with a statically configured system has no alternative but to over dimension the system by a wide margin. We show how to determine that margin using a cumulative probability distribution.

The question asked was whether or not response time was affected by varying the statistical properties apart from the mean value of the arrival and service process. Our results supports this hypothesis. However, varying traffic characteristics did not have the effect predicted by queueing formulas. Increased distribution variance should give a larger average response time and hence greater unpredictability. Observation of this phenomenon in the $G/M/1$ case revealed that increase of service time variance proved to have the opposite effect. Our experimental results consistently showed a decrease in average response time average for moving from exponential to Pareto distributed service times. We note that our datasets of 100000 data points might be too sparse in order for theoretical convergence to occur, but they are realistic nevertheless: if infinitely many data points are needed in order for the formulas

to be valid, they have lost their practical value. The formulas did show more promising for the simulated FCFS queues as they were able to capture trends, but even here the formula overestimated the response time mean value.

As the formulas are heavy traffic approximations and said to be valid as the traffic intensity reaches 1, the traffic intensity level which where chosen herein might not be sufficiently high. What is strange, however, is the discrepancies between formula, simulated and experimental $M/G/1$ queues as the formula in these cases have been theoretically proven to be accurate. Again, the distribution sizes might prevent average value convergence. In order to evaluate the formula for these queues, a larger set of data would be beneficial. However, formulas that have these limits on area of application are not of any practical value for providers of web services.

If reponse rate scales approximately linearly with datasize, then our results will not be significantly affected by variable job size, though such changes could affect the breakpoint ratio.

In this work the effects of arrival process self-similarity has also been studied. These studies were not targeted to do comparison with theoretical values as the response time formulas relies on independent service and arrival processes. In self-similar traffic each arrival possibly has a large correlation with previous events and thus are not independent. The results of our experiments indicates that the degree of self-similarity significantly affects server response time. Especially, as the level of self-similarity increases, the effect seems to aggregate.

It was originally planned to compare the results of these studies with experimental results for $M/M/1$ queues. But since we were not able to generate self-similar distributions with an equal level of dataset variance, that had to be abandoned. The results of the self-similarity experiments indicates that not only are the first two moments of the arrival and service process decisive for response time; the history or clustering of short and long inter-arrivals is also of great importance.

We also asked whether a web server could be modelled as one single queueing system with FCFS scheduling. Comparison of results from simulation of a single FCFS queueing system and experiments does not substantiate this. For exponentially distributed inter-arrival and service time, mean response time from simulation are not that different from that obtained through experiments. However, as the variance of service time increases, the gap between experimental and simulation results grows. Even if these are short term effects, they are the true for the timescales that clients care about.

It is clear from related work[5] that one must avoid the region of server performance saturation by having sufficient over-capacity. If one fails to do this, even a small burst in traffic can essentially lead to an unrecoverable queue growth. But how shall we determine the correct margin for over-provision?

We found that the real server system has increased time variability compared to the FCFS queue prediction, with a longer tail. Response time tail behaviour is a focal point for many SLAs [29]. In considering usual service level targets we found that highly variable inter-arrival times and exponentially distributed service times constitute the worst case scenario.

Using this as basis for capacity planning would enable a service provider to be in compliance with service level targets for all the traffic types studied here.

Moreover, assuming highly variable service and inter-arrival times and FCFS scheduling will create an even worse scenario in terms service level target value. Thus, it would provide a buffer against unforeseen traffic fluctuations and service providers will generally be able to meet SLAs a significant larger portion of the time. This presupposes no traffic self-similarity under such conditions this rule of thumb might not be valid.

XI. CONCLUSIONS

Data centre engineers need simple methods to estimate expected performance in relation to load. Using queueing models to estimate server performance is a highly non-trivial matter. We have shown that simplistic queue modelling is basically unreliable, but can be used with some effort to determine a *probable server capacity*.

Our experiments have many limitations, but this has been partly the point of the study. If it is so difficult to predict server capacity and the effects of traffic variations then no one will bother. The motivation must be to save money in needless data centre capacity[30]. Even the simple models used here would require a computer assisted tool to calculate the expected results, and the results are still quite unreliable.

For technical reasons, we were not able to adjust the distribution variances as we would have liked, which made investigations of highly variable self-similar traffic impossible. It was also noted in [31] that the most appropriate way for generating self-similar traffic was through many independent ON/OFF processes with the on and off periods following a heavy tailed distribution. In addition, it was recommended that 100 sources were employed. Functionality for such ON/OFF processes could be implemented into the client constructed in conjunction with this experiment and make it distributed to achieve the number of recommended sources. Such studies could possibly be conducted using a virtual machine environment. We believe that these issues are still worth investigating in future work.

We are grateful to Sven Ingebrigt Ulland and Gard Undheim for helpful discussion.

This work is supported in part by the EC IST-EMANICS Network of Excellence (#26854)

REFERENCES

- [1] R.B. Cooper. *Stochastic Models*, volume 2 of *Handbooks in Operations Research and Management Science*, chapter Queueing Theory. Elsevier, 1990.
- [2] H.L. Lee and S. Nahmias. *Logistics of Production and Inventory*, volume 4 of *Handbooks in Operations Research and Management Science*, chapter Single Product, Single Location Models. Elsevier, 1993.
- [3] G.R. Grimmett and D.R. Stirzaker. *Probability and random processes (3rd edition)*. Oxford scientific publications, Oxford, 2001.
- [4] J. Walrand. *Stochastic Models*, volume 2 of *Handbooks in Operations Research and Management Science*, chapter Queueing Networks. Elsevier, 1990.
- [5] M. Burgess and G. Undheim. Predictable scaling behaviour in the data centre with multiple application servers. In *Proc. 17th IFIP/IEEE Distributed Systems: Operations and Management (DSOM 2006)*, volume submitted. Springer, 2006.
- [6] M. Burgess and S.I. Ulland. Uncertainty in global application services with load sharing. In *Proc. 17th IFIP/IEEE Distributed Systems: Operations and Management (DSOM 2006)*, volume submitted. Springer, 2006.
- [7] W. Xu, X. Zhu, S. Singhal, and Z. Wang. Predictive control for dynamic resource allocation in enterprise data centers. In *Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, pages 115–126. IEEE Press, 2006.
- [8] X. Li, L. Sha, and X. Zhu. Adaptive control of multi-tiered web applications using queueing predictor. In *Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, pages 106–114. IEEE Press, 2006.
- [9] J. Cao, M. Andersson, C. Nyberg, and M. Kihl. Web server performance modeling using an m/g/1/k*ps queue. In *Telecommunications, 2003. ICT 2003. 10th International Conference on*, volume 2, pages 1501–1506, Feb/March 2003.
- [10] Leonard Kleinrock. *Queueing Systems: Computer Applications*, volume 2. John Wiley & Sons, Inc., 1976.
- [11] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.
- [12] K. Raatikainen. Symptoms of self-similarity in measured arrival process of ethernet packets to a file server, 1994.
- [13] Vern Paxson and Sally Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.
- [14] Hyoungh-Kee Choi and John O. Limb. A behavioral model of web traffic. In *ICNP '99: Proceedings of the Seventh Annual International Conference on Network Protocols*, page 327, Washington, DC, USA, 1999. IEEE Computer Society.
- [15] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, 1997.
- [16] Kihong Park, Gitae Kim, and Mark Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *ICNP '96: Proceedings of the 1996 International Conference on Network Protocols (ICNP '96)*, page 171, Washington, DC, USA, 1996. IEEE Computer Society.
- [17] Thomas Karagiannis, Mart Molle, and Michalis Faloutsos. Long-range dependence: Ten years of internet traffic modeling. *IEEE Internet Computing*, 8(5):57–64, 2004.
- [18] Jin Cao, William S. Cleveland, Dong Lin, and Don X. Sun. On the nonstationarity of internet traffic. In *SIGMETRICS '01: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 102–112, New York, NY, USA, 2001. ACM Press.
- [19] K.I. Hopcroft, E. Jakeman, and J.O. Matthews. Discrete scale-free distributions and associated limit theorems. *Journal of Mathematical Physics*, A37:L635–L642, 2004.
- [20] Kihong Park and Walter Willinger, editors. *Self-Similar Network Traffic and Performance Evaluation*. John Wiley & Sons, Inc., New York, NY, USA, 2000.
- [21] Ashok Erramilli, Onuttom Narayan, and Walter Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. Netw.*, 4(2):209–223, 1996.
- [22] L.P. Slothouber. A model of web server performance. In *The 5th International World Wide Web Conference, Paris, France*, 1996.
- [23] R.D. van der Mei, R. Hariharan, and P.K. Reeser. Web server performance modeling. *Telecommunication Systems*, 16(3 - 4):361–378, March 2001.
- [24] Richard C. Larson and Amedeo R. Odoni. *Urban Operations Research*. Prentice-Hall, 1981. Online book: http://web.mit.edu/urban_or_book/www/book/.
- [25] David L. Jagerman, Baris Balçoglu, Tayfur Altioek, and Benjamin Melamed. Mean waiting time approximations in the g/g/1 queue. *Queueing Syst. Theory Appl.*, 46(3-4):481–506, 2004.
- [26] Daniel A. Menasce, Lawrence W. Dowdy, and Virgilio A. F. Almeida. *Performance by Design: Computer Capacity Planning By Example*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.
- [27] W. Willinger, V. Paxson, and M.S. Taqqu. Self-similarity and heavy tails: structural modelling of network traffic. in *A practical guide to heavy tails: statistical techniques and applications*, pages 27–53, 1996.
- [28] D. A. Menasc and V. A .F Almeida. *Capacity Planning for Web Services*. Prentice Hall, Upper Saddle River, New Jersey 07458, first edition edition, 2002.
- [29] Bruno Abrahao, Virgilio Almeida, Jussara Almeida, Alex Zhang, Dirk Beyer, and Fereydoon Safai. Self-adaptive sla-driven capacity manage-

- ment for internet services. In *10th IEEE/IFIP Network Operations and Management Symposium*, pages 557–568, 2006.
- [30] J. Sauvé et al. Sla design from a business perspective. In *IFIP/IEEE 16th international workshop on distributed systems operations and management (DSOM)*, in LNCS 3775.
- [31] G. Horn, A. Kvalbein, J. Blomskøld, and E. Nilsen. An empirical comparison of generators for self similar simulated traffic. *Submitted to Elsevier Performance Evaluation*, 2005.