Capacity planning with phased workloads

E. Borowsky, R. Golding, P. Jacobson, A. Merchant*,
L. Schreier, M. Spasojevic, and J. Wilkes
Hewlett-Packard Laboratories

Abstract

At the heart of any configuration or capacity planning algorithm for storage systems, there lies a "what if" question: given a device and a set of workloads accessing data on the device, will the quality of service requirement for each workload be satisfied? This is, in general, a hard question to answer because of the complexity of workloads in real life. In this paper, we consider QoS bounds on the 95th percentile of response time and demonstrate an approximate method to verify that the QoS requirement is satisfied for a complex and fairly general set of workloads, including workloads with phasing (on/off behavior) and correlations with other workloads.

1 Introduction

The Forum project [1] addresses the problem of capacity planning and automatic configuration of storage devices to meet QOS requirements of applications. In its simplest form configuration may be thought of as a data layout problem: how to allocate (lay out) storage to a given set of applications on a given set of devices so that specified QOS requirements of the applications (such as response time, minimum throughput or availability) are satisfied. The capacity planning problem is the other side of the coin: it asks what devices are required so that the QoS requirements of a given application set can be met. These are issues typically handled manually by system administrators, who use a combination of experience, rules of thumb and trial and error methods to produce acceptable configurations. This becomes harder and harder as the sizes of storage installations increase, the application requirements become more complex and new devices appear on the market. We automate this process by treating the problem as a constrained optimization: a good solution optimizes objectives (such as cost or performance) while meeting application requirements. There are many optimization methods which may be applied to the problem; however, at the least, they all require an answer to the following modeling question: given a set of devices, a set of applications and a storage layout (an assignment of storage on devices to applications), how does it perform? Does it meet the requirements of the applications? This is the "what if" question that we address, in part, in this paper.

Note that this deceptively simple question is quite hard to answer, requiring good models of the I/O workloads created by the applications, an understanding of how they interact and interfere with other workloads, and how the device responds to the combined workloads. One solution is to use simulation models, since these can handle complex interactions of workloads and devices, but, since this is relatively slow, it is not a practical solution for use in the inner loop of optimization techniques, which may ask thousands of such "what if" questions *en route* to a good solution. Standard analytical techniques can answer such questions if the workloads are sufficiently simple, but in real life, workloads can have complex ON/OFF phasing behavior as well as correlation between the timing of the ON/OFF phases of different workloads. In solution, we have developed approximate analytical techniques which are quite powerful in dealing with complex workloads and devices.

We begin with a specification of the problem and a workload model in Section 2. A model is developed using a new metric "Short term utilization" in Section 3, some validation results are presented in Section 4, and conclusions are in Section 5.

2 Problem and workload specification

We choose to narrow the problem somewhat by considering only the applications using one device, and focusing only on one metric: response time.

*Corresponding author. Address: 1501 Page Mill Rd., MS 1U-13, Palo Alto, CA 94304. E-mail: arif@hpl.hp.com

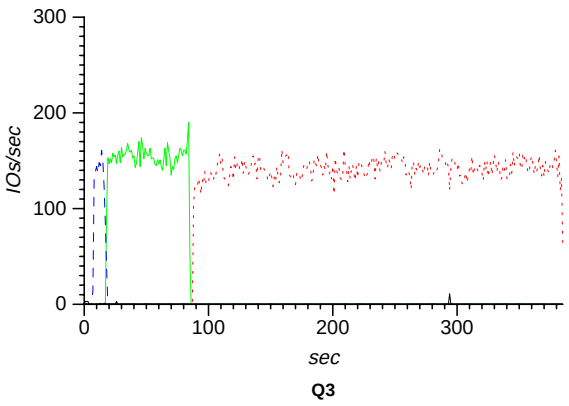

Figure 1: TPC-D Query 3 I/O trace. I/Os to different tables occur in phases which overlap little.

Problem specification

We are given a device D and a set of applications A_i $(i = 1, \ldots, n)$, accessing data stored on D. Each application A_i requires that 95% of its storage requests must see a response time smaller than T_i. Will the requirement of each application be satisfied?

Workload specification

In order to answer this question we need an accurate characterization of the workload imposed by each application. Our workload model is derived from an analysis of traces of I/O activity from several applications, including the TPC-C [2] and TPC-D [3] benchmarks. Traces of the TPC-D benchmark, in particular, indicate that the request arrival process would not be well characterized by a Poisson or even a general renewal process approximation. I/O activity to a table in the TPC-D benchmark occurs in well-separated phases (Fig. 1). Further, the I/O activity to different tables is not independent; some tables are never accessed at the same time, whereas others are always accessed together. Ignoring these correlations in I/O activity leads to very poor models. For example, when there are several workloads which have high request rates during their ON phases, but no two workloads are ever on together, it may be possible to put all the workloads on the same device. If the model ignores this correlated behavior, then it will require the workloads to be put on separate devices to avoid the possibility of the workloads interfering with each other; this is a much higher cost solution.

Accordingly, we choose a *phased, correlated* workload model. I/Os produced by an application are separated into *streams*; each stream accesses one *store*. A store represents a logical chunk of storage; for example, a database table. Each stream is modeled through a modulated ON/OFF Poisson process. During the ON state, I/O requests from stream P_i arrive according to a Poisson process with rate λ_i; the ON state persists for an exponential length of time with mean on_i. During the OFF state, there are no requests from stream P_i; the OFF state persists for a length of time with mean off_i. We model correlation in the phasing behavior of different workloads by looking at their states at the instant of transition, when a stream P_i goes from OFF to ON state. The correlation between streams P_i and P_j is represented by the probability

$$p_{ij} = \text{Prob [Stream } P_j \text{ is ON when stream } P_i \text{ comes ON].}$$

There are basically four possible correlations:

- P_i and P_j are independent, in which case $p_{ij} = \mathsf{on}_j/(\mathsf{on}_j + \mathsf{off}_j)$, which is the steady state probability of finding P_j in the ON state.
- P_j is always ON when P_i comes on, in which case $p_{ij} = 1$. We define $p_{ii} = 1$ for all i.
- P_j is never ON when P_i comes on, in which case $p_{ij} = 0$.
- Some other correlation, in which case p_{ij} has a measured value.

The values of the parameters λ_i, on_i, off_i and p_{ij} for each i and j in $\{1, 2, \ldots, n\}$ can be derived from I/O traces. (The definition of these and other parameters used in this paper are summarized in Table 1 for easy reference.) Other useful information derived from the I/O traces includes the distribution of request sizes, the fraction of requests that are reads and the distribution of sequential run lengths (i.e., sequences of I/Os which access logically consecutive storage locations). These may be used to derive the distribution of service time for requests of stream P_i at a given device D. The computation of the service time may be simple or complex, depending on the device used (see [4]); in this paper, we assume that the mean and variance of service time at the device can be computed and is available for each stream, and that the service times for requests from a stream are independent and identically distributed.

3 Model

Traditional queueing models deal well with Poisson workload processes, and with some variants of these.

Table 1: Summary of notation.

Parameter	definition
λ_i	request rate of stream P_i in ON state
on_i	mean duration of ON state for stream P_i
off_i	mean duration of OFF state for stream P_i
p_{ij}	probability that stream P_j is ON when P_i comes ON
T_i	bound for 95th percentile of response time for stream P_i
t_i	a time instant when stream P_i comes ON
$W_i(t)$	work (from all streams) arriving at device D in period $(t_i, t_i + t)$
$W_{ij}(t)$	work from stream P_j arriving at device D in period $(t_i, t_i + t)$
$W'_{ij}(t)$	work from stream P_j arriving at device D in period $(t_i, t_i + t)$ given that P_j is ON
$S_{ij}(m)$	service time of mth request from stream P_j after time t_i
a_i	mean service time for request from P_i
b_i	variance of service time for request from P_i
$N_{ij}(t)$	number of requests from stream P_j in period $(t_i, t_i + t)$

However, there are few results available that can accommodate workloads as complex as those proposed here. The usual means of handling such cases is to use very simple metrics, for example, utilization — the long-term average fraction of time that the device is busy. In most cases, even when the workloads and devices involved are complex, it is quite simple to compute the utilization. It is well understood that, for stability, the utilization must be less than 1; also, queue lengths and response times tend to decrease with utilization. This leads to such rules of thumb as "keep the utilization below 0.5" or some other threshold in order to achieve acceptable response times. The actual threshold used in such rules of thumb is based on intuition and experience, and is usually very conservative, since it is based on very little information.

Intuitively, queues build up and queueing delays occur in stable systems (where the utilization is less than 1) because of short term variations in the rate that work arrives. When requests arrive faster than the device can service them, then queues form. In the long term, the device "catches up" and the queue disappears, but some requests suffer queue delays in the interim.

One way to limit queue delay is, therefore, to require that the work that arrives at a device in a given "short" period T must not exceed what the device can perform in that period of time. The time T is a tunable parameter; the smaller the value of T is, the more stringent this requirement becomes. For feasibility, T must be larger than the device service time of the request. We show later in this paper that if this constraint is satisfied, the total amount of work in the queue does not exceed what the device can perform in time T. For First-Come-First-Served scheduling, this guarantees that the response time of any request does not exceed T. Since most scheduling methods that we use are likely to be at least as good as FCFS, this constraint allows us to bound response time, at least in a heuristic sense, even when the response time cannot be directly computed. An additional benefit is that this is done using a utilization measure, and utilization measures are easy to compute in most cases.

3.1 Short-term utilization

The following theorem makes concrete the intuition that we can limit queue lengths by limiting the short-term variations in the rate that work arrives for the device.

Theorem 1 *If a work-conserving device starts with no requests pending, and all the requests arriving at the device in any period of length T can be served by the device in time T or less, then all the requests pending at the device at any time t can be served in time T or less.*

Proof: Let

$\mathsf{work\_pending}(t)$ $\equiv$ the sum of the (remaining) service times for requests in queue at time t

$\mathsf{work\_arrived}(t_1, t_2)$ $\equiv$ the sum of service times of requests that arrive in the interval (t_1, t_2)

$\mathsf{busy\_time}(t_1, t_2)$ $\equiv$ the amount of time in (t_1, t_2) that the device is busy.

Since the device starts with no requests pending, $\mathsf{work\_pending}(0) = 0$. Also, $\mathsf{work\_arrived}(t, t+T) \leq T$ for all $t > 0$. We break up time into intervals $(0,T)$, $(T,2T)$, $(2T,3T)$, ..., $(nT,(n+1)T)$,..., and prove that $\mathsf{work\_pending}(t) \leq T$ in each such interval. The proof is by induction on n.

For the first interval $(0,T)$: since the total service time for all the requests arriving in $(0,T)$ cannot exceed T, the proposition holds trivially.

Now, we show that if $\mathsf{work\_pending}(t') \leq T$ for t' in $((n-1)T, nT)$, then this must also hold for the next interval. For t in $(nT,(n+1)T)$,

$$\mathsf{work\_pending}(t) = \mathsf{work\_pending}(t-T) + \mathsf{work\_arrived}(t-T,t) - \mathsf{busy\_time}(t-T,t).$$

Clearly, $\mathsf{busy\_time}(t-T,t) \geq \mathsf{work\_pending}(t-T)$, since $\mathsf{work\_pending}(t-T) \leq T$, and time T has elapsed in $(t-T,t)$. Therefore,

$$\mathsf{work\_pending}(t) \leq \mathsf{work\_arrived}(t-T,t) \leq T.$$

By induction, $\mathsf{work\_pending}(t) \leq T$ for $t \in ((n-1)T, nT)$, $n = 0, 1, \ldots$. ∎

In effect, Theorem 1 says that if it can be guaranteed that the requests from the combined workload at device D arriving in every interval of length T seconds do not require more than T seconds to serve, then the response time for all requests is bounded by T seconds. For practical use, this requirement must be relaxed slightly. In most cases, it is hard to *guarantee* that the service time of requests arriving in any interval of length T will not exceed T, because neither the workload process nor the service times of the requests at device D are known exactly. In general, only a probabilistic description is available. In order to accommodate this, we relax the theorem into a heuristic rule:

> If the total service time required by the requests generated by the combined workload at device D in every interval $(t, t+T)$ containing a request from stream P_i is less than T seconds with a probability p, then the response time for requests from stream P_i is T seconds or less with probability p.

Note well that this is **not** a theorem; it is merely an approximation based on Theorem 1. Using this approximation, we may convert the requirement that 95% of the requests see a response time of less than T seconds into the requirement that the work arriving at device D in intervals of length T takes less than T seconds to perform with probability $p = 0.95$; we call this the "short-term utilization constraint". In the following sections, we show how this rule may be used, and some validation of the fact that it usually works well.

3.2 Verifying the Short term Utilization constraint

The response time seen by requests from a stream P_k depends on the total rate of work arriving at the device. The only times at which the rate of work arriving at the device increases is when some stream changes from the OFF state to the ON state. Therefore, to verify that the response time seen by requests from P_k is less than T_k, it is sufficient to verify that the response time for P_k is less than T_k immediately after P_i comes ON, for each possible P_i. In other words, if we wish to verify that an assignment of streams P_1, P_2, ..., P_n to device D satisfies the short-term utilization constraint, it is sufficient to verify that it the total work arriving at device D in the interval $(t_i, t_i + T_k)$ (where t_i is the time that P_i switches ON) is less than the work the device can do in in time T_k, for each pair (i,k), where $i \in \{1,2,\ldots,n\}$ and $k \in \{1,2,\ldots,n\}$ — a total of n^2 tests. As we shall see, this can be reduced to n tests.

Let

$$\begin{aligned} t_i &\equiv \text{a time instant when stream } P_i \text{ comes ON} \\ W_i(t) &\equiv \text{work (total service time of requests) arriving in } (t_i, t_i+t) \\ W_{ij}(t) &= \text{work (total service time of requests) arriving from stream } P_j \text{ in time } (t_i, t_i+t) \end{aligned}$$

We can then represent the short-term utilization constraint as

$$Pr[W_i(T_k) < T_k] \geq p \quad \text{for each } i \text{ and } k \in \{1,2,\ldots,n\} \qquad (1)$$

Verifying the inequality (1) requires the computation of the tail of the distribution of W_i. The distribution of W_i is, in general, difficult to compute; however, we approximate W_i as a normal random variable, which allows us to compute the tail of the distribution from the mean and variance of $W_i(T_k)$. We show next how to compute this mean and variance.

In order to compute the moments of $W_i(T_k)$, we assume that no stream switches state in the interval (t_i, T_k); this is reasonable, since T_k is of the order of

the response_time(k), which is likely to be of a much smaller order than the time between change of state for the streams.

The total work is the sum of work brought in by the individual streams:

$$W_i(t) = \sum_{j=1}^{n} W_{ij}(t).$$

Let

$$\begin{aligned} W'_{ij}(t) &\equiv \text{work (total service time of requests)} \\ &\quad \text{arriving from stream } j \text{ in} \\ &\quad (t_i, t_i + t) \text{ given that } P_j \text{ is ON} \\ p_{ij} &= \text{probability that stream } j \text{ is ON at time } t_i \end{aligned}$$

Then,

$$\begin{aligned} E[W_i(t)] &= \sum_{j=1}^{n} E[W_{ij}(t)] \\ &= \sum_{j=1}^{n} p_{ij} E[W'_{ij}(t)] \\ Var[W_i(t)] &\approx \sum_{j=1}^{n} Var[W_{ij}(t)] \\ &= \sum_{j=1}^{n} p_{ij} Var[W'_{ij}(t)] + \sum_{j=1}^{n} p_{ij}(1-p_{ij}) E[W'_{ij}(t)]^2 \end{aligned}$$

We now compute the mean and variance of $W'_{ij}(t)$. Suppose that stream j is ON at time t_i, and there are $N_{ij}(t)$ requests with service times $S_{ij}(1)$, $S_{ij}(2)$, ..., $S_{ij}(N_{ij}(t))$ from stream j to device D in the interval $(t_i, t_i + t)$. Since the stream P_j ($j \neq i$) is assumed to be Poisson with mean rate λ_j when it is on, $N_{ij}(t)$ is Poisson distributed with mean $\lambda_j t$.

$$\begin{aligned} E[N_{ij}(t)] &= \lambda_j t \\ Var[N_{ij}(t)] &= \lambda_j t \end{aligned}$$

Let the mean and variance of $S_{ij}(\cdot)$ be a_j and b_j respectively. Then the mean and variance of $W'_{ij}(t)$ can be computed using standard formulae for random sums [6]:

$$\begin{aligned} W'_{ij}(t) &= S_{ij}(1) + S_{ij}(2) + \cdots + S_{ij}(N_{ij}(t)) \\ E[W'_{ij}(t)] &= E[N_{ij}(t)] E[S_{ij}(1)] \\ &= \lambda_j t a_j \\ Var[W'_{ij}(t)] &= E[N_{ij}(t)] Var[S_{ij}(1)] \\ &\quad + Var[N_{ij}(t)] E[S_{ij}(1)]^2 \\ &= \lambda_j t b_j + \lambda_j t a_j^2 \end{aligned}$$

$$\begin{aligned} E[W_i(t)] &= \sum_{j=1}^{n} p_{ij} E[W'_{ij}(t)] \\ &= \sum_{j=1}^{n} p_{ij} \lambda_j t a_j \\ Var[W_i(t)] &= \sum_{j=1}^{n} p_{ij} Var[W'_{ij}(t)] \\ &\quad + \sum_{j=1}^{n} p_{ij}(1-p_{ij}) E[W'_{ij}(t)]^2 \\ &= \sum_{j=1}^{n} p_{ij} \lambda_j (a_j^2 + b_j) t + \sum_{j=1}^{n} p_{ij}(1-p_{ij}) \lambda_j^2 a_j^2 t^2 \end{aligned}$$

We now approximate $W_i(t)$ by a normal random variable with the computed mean and variance. The short-term utilization constraint

$$Pr[W_i(T_k) < T_k] > p$$

translates into

$$\Phi\left(\frac{T_k - E[W_i(T_k)]}{\sqrt{Var[W_i(T_k)]}} \right) > p$$

where $\Phi(\cdot)$ is the cumulative distribution function of the unit normal random variable. Since Φ is a monotone increasing function, we can re-write this as

$$\frac{E[W_i(T_k)] + \Phi^{-1}(p)\sqrt{Var[W_i(T_k)]}}{T_k} < 1 \qquad (2)$$

where Φ^{-1} is the inverse function of Φ, that is, $\Phi^{-1}(\Phi(u)) = u$. We define the quantity on the left hand side of this inequality as the short-term utilization function of device D.

$$STU(p, P_i, T_k) \equiv \frac{E[W_i(T_k)] + \Phi^{-1}(p)\sqrt{Var[W_i(T_k)]}}{T_k}.$$

Then, we can verify that the 95th percentile of response time for P_k by checking that:

$$STU(0.95, P_i, T_k) < 1 \qquad \text{for } i = 1, \ldots, n.$$

Thus, verifying that the response time requirement for all n streams amounts to checking n^2 such inequalities. However, it is easy to show that $STU(p, P_i, t)$ decreases monotonically as t increases. Therefore,

$$\max_{k=1}^{n}(STU(p, P_i, T_k)) = STU(p, P_i, \min_{k=1}^{n}(T_k)).$$

Thus, we can verify the response time for all the streams by checking the n inequalities

$$STU(0.95, P_i, T_{min}) < 1 \qquad \text{for } i = 1, \ldots, n,$$

where $T_{min} = \min_{k=1}^{n}(T_k)$.

4 Validation results

The short term utilization bound is based on a few large assumptions. The foremost of these assumptions is that the generalization of Theorem 1 to the 95th percentile is accurate. The other large assumption is that $W_i(t)$ can be approximated as a normal random variable. To test whether these assumptions hold, we ran a series of experiments to determine accuracy and tightness of the short term utilization bound for a sample synthetic workload. We then ran a number of sensitivity experiments to determine which parameters highly impact the correctness and tightness of the bound.

We first present some definitions, and describe the baseline test case. Next, we present the sensitivity tests and their results, and finally we close with a summary and notes for improvement of the model.

4.1 Definitions

We say that a model is *accurate* if every configuration that the model predicts to be feasible, is actually feasible. In this case, feasibility means that the short term utilization bound is actually greater than the measured short term utilization of that configuration, or that the bound is pessimistic.

The *tightness* of a model is the ratio between the response time bound predicted by the model and the actual value based on measurements. If the model is perfectly accurate, then the tightness should never be less than one. If the model is a perfectly tight predictor, the tightness should be exactly equal to one. Note that the short-term utilization constraint, as presented, does not actually predict the response time; it merely verifies a bound on the response time. However, we can find the lowest response time bound that a given configuration can meet under the short-term utilization constraint. Tightness is the ratio of this bound to the actual 95th percentile of response time found using simulation.

The *sensitivity* of a model is a measure of how much the tightness varies depending on the values of the parameters to the model. A model is sensitive to its parameters if the variance in tightness is high.

4.2 The baseline experiment

A configuration for the short-term utilization model is fully characterized by the following parameters:

n The number of streams

λ_i The arrival rate for stream P_i

μ_i The service rate for stream P_i

$p_{i,j}$ As above, the correlation between streams P_i and P_j.

T_i The required bound on 95th percentile of response time for stream P_i.

For purposes of explanation, we will assume the last four are identical over a fixed set of streams; however, this assumption is relaxed in the actual experiments.

To evaluate the short term utilization model, we concentrate on the last of these parameters T to compare against measured values. Given n, λ, μ and $p_{i,j}$, we take the smallest value of T for which the formula holds as the predicted 95th percentile of response time. Since finding this value of T is difficult from the formula, we find the value experimentally through a bisection search on possible values of T.

We obtain the actual 95th percentile of the response time by simulating a storage device with a given distribution of service time, using the queuing simulation package that is part of the Pantheon simulator [5]. Given the specification of a set of streams, we generate a synthetic workload matching the parameters of the streams and measure the 95th percentile of the response time for each stream over a batched set of independent runs.

The simulation experiments consist of one set of baseline experiments and several sets of sensitivity experiments in which one parameter is varied from the baseline values. The baseline input to the simulator is a set of 8 streams, each with a Poisson arrival process with rate $\lambda = 1$ request/sec, with a mean ON time of 5 seconds and a mean OFF time of 3 seconds. The device serves requests from each stream in an exponentially distributed length of time with mean 0.15 seconds. We cluster the streams into three correlation groups. Group one consists of streams 0 through 3, group two of streams 4 and 5, and group three of streams 6 and 7. Groups one and two are completely correlated within the group and anti-correlated between groups. Streams in group three are completely independent of any of the other streams. The tightness results for this are shown in Table 2. The conclusion to be drawn is that, at least for the baseline case, the response time bound produced by the model is accurate, but not very tight.

4.3 Sensitivity analysis

We next show the results of a set of experiments in which one parameter is varied from the value set in the baseline case; the remaining parameters are as in the baseline experiment. The purpose of these experiments

Table 2: Baseline results.

stream	predicted bound	measured 95th percentile	tightness	accurate
stream0	0.063	0.047	1.36	yes
stream1	0.063	0.049	1.30	yes
stream2	0.063	0.049	1.30	yes
stream3	0.063	0.050	1.28	yes
stream4	0.049	0.045	1.08	yes
stream5	0.049	0.048	1.02	yes
stream6	0.065	0.049	1.33	yes
stream7	0.065	0.049	1.33	yes

is to test the sensitivity of the tightness and accuracy of the bound produced by the short-term utilization model to changes in each parameter.

Varying arrival rate

In this experiment, mean interarrival time varies from 0.2 to 2.0 seconds: 0.2, 0.3, 0.4, 0.6, 0.8, 1.0 (baseline), 1.3, 1.6, 2.0. The tightmess of the bounds is shown in Fig. 2. The bound based on the short-term utilization constraint is inaccurate when inter-arrival times are very large, or, in other words, when the arrival rate is very low. This can be traced to the fact that every request sees a response time that is at least as large as its own service time; however, the model is based on the distribution of work arriving at the device in short intervals. At low request rates, the probability that there will be *any* requests in the interval is small, and hence, the predicted bound on response time is smaller than a single-request service time. If accuracy of the model at low arrival rates is an issue, the inaccuracy here can be corrected in a number of ways. Perhaps the simplest is to treat the bound produced by the model as a bound on the queue delay rather than the response time; equivalently, one adds one service time to the bound produced by the model to get a bound for the response time. The tradeoff is that this leads to a looser bound at higher arrival rates.

Varying the heterogeneity of inter-arrival time

In this experiment, we changed the streams from being homogeneous to having varying mean interarrival times. To compute the inter-arrival time $r(i)$ for a stream P_i, assuming a baseline value b:

$$r(i) = 2^{\log(b)+(\lfloor n/2 \rfloor - i)c}$$

where c, the expansion factor, controls the spread of the points. This formula spreads the n points

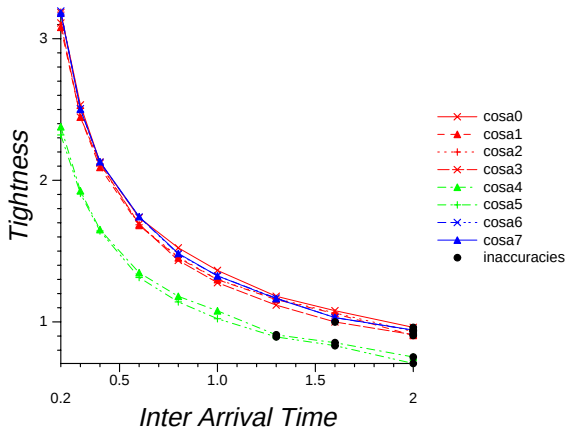

Figure 2: Tightness of response time bound as inter-arrival time varies.
Inaccuracies are points where the lowest response time bound predicted using the model was *higher* than the value measured from simulation (i.e., $tightness < 1$).

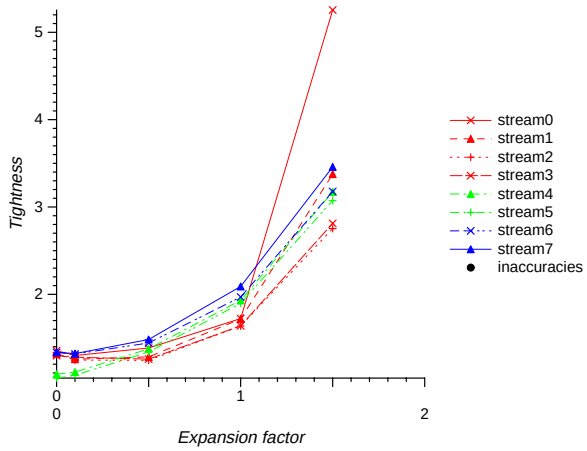

Figure 3: Tightness of response time bound as the heterogeneiety of inter-arrival time varies.

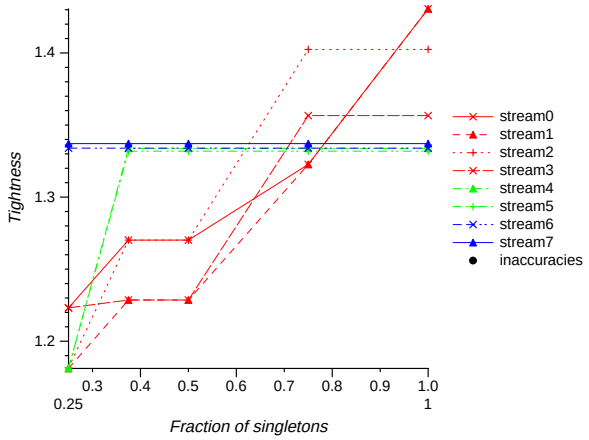

Figure 4: Tightness of response time bound as the fraction of singletons varies.
Singletons are streams independent of all other streams.

evenly in log space, with $c = 0$ representing homogeneity. For this experiment, we evaluated at $c = 0, 0.1, 0.5, 1.0, 1.5, 2.0$. (Note that a larger value of c implies a larger overall load on the system.) The sensitivity of the tightness of the response time bound is shown in Fig. 3. The figure shows that the bound is sensitive to the heterogeneity of interarrival period. However, this is most likely an artifact of increasing the total request rate. There are no tightness values plotted for the case $c = 2.0$ because the queue lengths exploded at this point, leading to very large response times. This was accurately predicted by the model in every case.

Varying correlations

This experiment examines the sensitivity of the short term utilization constraint to the correlations between streams. We vary the fraction of streams that are *singletons*, that is, streams independent of other streams. A large number of singletons means that most streams are independent, whereas a small number means that most streams are interdependent. Fig. 4 shows the tightness of the bound as the fraction of singletons changes. We conclude that the fraction of singletons does not highly impact the tightness of the bound. In general however, the lower the fraction of singletons, the tighter the bound becomes.

4.4 Conclusions from validation experiments

Overall, we find that the bound produced by the short-term utilization method is accurate in the large majority of cases. The cases in which it is not accurate, predicting a response time smaller than actually found by simulation, are primarily cases where the request rates of the streams are very low; clearly these are not usually the cases one is most concerned with. Where this is an issue, the model can be rectified to correct these inaccuracies. The bound is quite conservative in a large fraction of the experiments performed. This is to be expected, since we are dealing with a very general workload model, and the metric we are bounding is the 95th percentile of response time which is generally quite sensitive to variations in the workload. Nevertheless, it must be noted that this method should be used only when correctness of the bound is the primary requirement rather than the tightness of the bound.

5 Summary

In this paper, we have proposed an approximate method to answer a "what if" question that is at the heart of every capacity planning method for storage systems: if a given set of applications (data stores + I/O request streams) is assigned to a given device, will the performance requirements of each application be met? The performance metric we use is the 95th percentile of response time.

We have defined a very general workload model constructed from processes with ON and OFF phases. These processes can be correlated, which allows us to model a wide variety of real workloads. We have defined a metric called the "short-term utilization" and translated the question above into a constraint based on this metric, and shown how to evaluate it. Validation experiments which compare the bound produced by this method to the actual 95th percentile of response time (based on simulation), show that the bound is accurate, but quite conservative in most cases.

This method has been incorporated into a configuration tool for layout of large applications; in the future we shall report on its performance in actual systems.

References

[1] E. Borowsky, R. Golding, A. Merchant, L. Schreier, E. Shriver, M. Spasojevic, and J. Wilkes. Using attribute-managed storage to achieve QoS. 5th Intl. Workshop on Quality of Service, Columbia Univ., New York, June 1997.

[2] Transaction Processing Performance Council. TPC benchmark C, standard specification, revision 1.0. 13 August 1992.

[3] Transaction Processing Performance Council. TPC benchmark D (Decision Support), standard specification, revision 1.2. 9 November 1996.

[4] E. Shriver, A. Merchant, and J. Wilkes. An analytic QoS behavior model for realistic storage devices. To appear in SIGMETRICS '98.

[5] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan. The HP AutoRAID hierarchical storage system. ACM Transactions on Computer Systems 14(1):108-136, February 1996.

[6] S. Karlin and H. M. Taylor. *A First Course in Stochastic Processes.* Academic Press Inc., 1975.