Quantifying Machine Availability in Networked and Desktop Grid Systems*

UCSB Computer Science Technical Report Number CS2003-37

John Brevik
Department of Mathematics and Computer Science
Wheaton College
Norton, MA 02766

Daniel Nurmi
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106

Rich Wolski
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106

Abstract

In this paper, we examine the problem of predicting machine availability in desktop and enterprise computing environments. Predicting the duration that a machine will run until it restarts (availability duration) is critically useful to application scheduling and resource characterization in federated systems. We describe one parametric model fitting technique and two non-parametric prediction techniques, comparing their accuracy in predicting the quantiles of empirically observed machine availability distributions.

We describe each method analytically and evaluate its precision using a synthetic trace of machine availability constructed from a known distribution. To detail their practical efficacy, we apply them to machine availability traces from three separate desktop and enterprise computing environments, and evaluate each method in terms of the accuracy with which it predicts availability in a trace driven simulation.

Our results indicate that availability duration can be predicted with quantifiable confidence bounds and that these bounds can be used as conservative bounds on lifetime predictions. Moreover, a non-parametric method based on a binomial approach generates the most accurate estimates.

1 Introduction

The rapid proliferation of Computational Grid computing [14, 4] combined with the commercial (if illegal) success of peer-to-peer file sharing systems has sparked an interest in the use of "volatile" desktop machines as an aggregated compute and storage platform. Enterprise computing systems such as those developed by Entropia [12], United Devices [25], and Avaki [2] provide various technologies designed to harvest compute power from personal computers in a commercial setting. Grid computing systems such as Condor [24], Globus [13], GrADSoft [3], and NetSolve [7] make it possible to combine user-controlled resources (workstations, personal computers, etc.) with large-scale clusters and machines to form an integrated computing environment. Finally, community-driven efforts such as SETI@home [23] and the Bovine RC5 effort [6] have demonstrated a significant use of spare compute cycles, promising significant new results.

Part of the challenge to using "desktop" or user-controlled resources comes from their their relative volatility as compared to their shared and managed counterparts. The "owner" of a desktop machine typically exercises ultimate control over the processes that run on it, its connectivity to the network, and its reboot cycle. While system administrators may go to great lengths to ensure that shared server resources (computational or storage) are highly available, they rarely can exercise the same degree of control over resources that are assigned to individual users.

In this paper, we describe a methodology for predicting machine availability from monitoring data in distributed computing environments. Specifically, we focus on the ability to estimate a specified *quantile* for the distribution of

*This work has been supported by grants from the National Science Foundation numbered EIA-9975020 (The GrADS project), CCR-0331645 (The VGrADS project), and NGS-0305390 as well as the DOE SciDAC program.

availability, and a *confidence level* associated with each estimate. Thus we can phrase the problem as follows:

> *From a set of availability measurements taken from a resource (or set of resources that are assumed to be homogeneous), and given a desired percentile p and confidence level c, what is the largest availability duration t for which we can say with confidence c that p percent of the availability time measurements are greater than or equal to?*

The answer to this a question for a given data set, percentile of interest (and take $q = 1 - p$), and desired confidence level, is a lower bound estimate of the q_{th} quantile from the data set. While not a prediction of the exact availability duration, using an estimate of a quantile provides a lower bound on how long a machine (or collections of machines) is likely to be available, and the confidence measure provides a quantitative (but probabilistic) "guarantee" of the estimate's accuracy. For example, a scheduler may wish to establish the availability duration that is smaller than all but 1% of the possible durations, and to be certain of that number with 95% confidence. It is our belief that application schedulers such as those described in [8, 18, 21, 5, 1] can use these estimates to make automatic decisions about where and for how long to run distributed application components.

Our goal is to develop such a quantile estimation methodology that supports "live" predictions of availability so that schedulers (be they human or automatic scheduling programs) can make decisions dynamically. As such, we compare one (parametric) automatic model fitting technique based on Maximum Likelihood Estimation (MLE) and two non-parametric techniques in terms of how well they bound the observed quantile of a given value in a trace-driven simulation.

We apply each method to machine availability data gathered from three different distributed computing environments. At the University of California, Santa Barbara, we have instrumented the student-accessible machines in the Computer Science Department to record their available and unavailable periods. We have also developed a method for recording processor occupancy duration under Condor [24] – a cycle-harvesting system developed and deployed at the University of Wisconsin. Finally, to ensure that our measurement implementation does not introduce an unforeseen bias, we analyze availability data from a 1995 survey of Internet hosts conducted by Long, Muir and, Golding at the University of California, Santa Cruz [15][1].

As motivation for the usefulness of the three techniques when applied to empirical data, we begin with a short verification experiment which performs lower bound quantile estimate using each method on a synthetic, fixed Weibull distribution. For each empirical data set, we then perform an experiment where individual machine availability traces are split into a training set, (which we use to estimate the quantile lower bound) and an experimental set which is used to verify the accuracy of the estimate. Because the training set precedes the experimental set in each machine trace, our results detail how well each estimation method *predicts* machine availability for that machine during the time period covered by the experimental set.

The remainder of this paper is organized as follows. The next section, Section 2, describes three techniques for producing lower confidence bounds for quantiles. Section 3 describes the data sets we use to investigate these methods. In Section 4 we compare the three estimation methods empirically. Finally, we conclude our investigation in Section 5.

2 Inference for Quantiles

In this section, we examine the problem of determining lower bounds, at a fixed level of confidence, for quantiles for a given population whose distribution is unknown. More typically, statistical inference aims to find confidence *intervals*, but for our application, we are only concerned with lower bounds, and placing all of the potential error on the bottom end of the interval allows us to produce somewhat more accurate values. At the same time, it provides a minimum "quality of service" guarantee with a quantifiable estimate of the probability that the guarantee will be violated.

For example, if a scheduler or machine user would like to know the minimum amount of time a machine is likely to run before it reboots, the 0.05 quantile of the *true* availability distribution provides this number with 95% confidence. That is, random variation will cause the machine availability to be less than the 0.05 quantile 5 times out of 100, *if the quantile is exact*. In practice, we do not know the exact distribution, so the quantile must be estimated. There are a variety of techniques for estimating quantiles (three of which we describe in this section) but to be used as a guarantee, we require a confidence value for the estimated quantile itself. That is, since the exact quantile is an uncertain, we model it as a random variable, and calculate a confidence bound for the quantile in order to make a guarantee. By choosing the lower confidence bound, we ensure that the true quantile is larger than the bound value with a specified confidence. Thus, machine availability will be larger than this bound with the specified confidence – it is, in effect, a conservative estimate (high confidence) of a conservative estimate (low quantile) of the availability.

While we are using the quantile to give us a minimum

[1] We gratefully acknowledge Dr. Divikant Agrawal at the University of California, Santa Barbara, Dr. Miron Livny at the University of Wisconsin, and Dr. Darrell Long at the University of California, Santa Cruz for their support of this work.

availability estimate, and a lower confidence bound to give us a minimum quantile estimate, all of the methods outlined below can be adapted easily to producing confidence intervals (or, of course, upper bounds).

2.1 Parametric Approach: Model Fitting

Perhaps the most intuitive method for making quantile inferences uses an optimization-based model-fitting technique to compute a "good" continuous model for a given data set, and then computes confidence intervals for the necessary parameters as a way of bounding the estimate of each quantile. For example, the machine availability data described in Section 3.1 seems to be modeled reasonably well by a distribution from the Weibull family [17]. The density and distribution functions f_w and F_w respectively for a two-parameter Weibull distribution are given by

$$f_w(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^{\alpha}} \quad (1)$$

$$F_w(x) = 1 - e^{-(x/\beta)^{\alpha}} \quad (2)$$

Using a Maximum Likelihood Estimation (MLE) technique to estimate the parameters from the data, the resulting Weibull density function is

$$F_w(x) = 1 - e^{-(x/249105)^{0.536852}}. \quad (3)$$

That is, $\alpha = 0.536852$ and $\beta = 249105$. A comparison of the empirical data set to the MLE-determined Weibull is depicted in Figure 1. Notice, however, there is enough discrepancy between the MLE Weibull model and the actual empirical data, especially near the left end of the distribution, that using model-based quantiles may be inadequate.

Moreover, it is possible to calculate confidence intervals for the parameters α and β. MATLAB [16], which we use for all MLE and parameter confidence interval computations in this paper, uses the Fisher Matrix Method for computing independent confidence bounds on each parameter of a distribution at a specified confidence level σ. In this work, we use the simplifying assumption that the true Weibull model is bounded by the two Weibull models corresponding to the computed lower and upper parameter pairs. Furthermore, since each parameter confidence interval is computed independently, we take the region between the two bounding distributions as the σ^2 confidence region. The resulting graphs are shown in Figure 2.

We generate these graphs using the endpoints of the confidence range for each parameter. In this example, the 94.868% confidence range for the parameter α is $(0.51795 \leq \alpha \leq 0.55574)$ and for β the range is $(109710 \leq \beta \leq 678720)$. We consider the bounded region to be 94.868% confident on α, and 94.868% confident on β, which gives us a $94.868^2 = 90\%$ confidence region. If the data were generated by a Weibull, the parameters of that Weibull would have only 10% chance of being outside this range (5% chance of being below the lower, 5% chance of being above the upper).

However, as Figure 1 demonstrates, even using MLE to determine the parameters yields a Weibull distribution that is only approximate. Thus, in practice, the confidence interval for the fitted Weibull may not accurately predict the number of values that actually fall inside and outside the interval.

In this example, we have presented a confidence range for the Weibull based on confidence intervals calculated for the parameters. Again, for the practical purposes of prediction, we are only interested in ensuring that the machine availability measures are larger than the "leftmost" distribution with a given confidence level. Because the range, in this example, is a 90% confidence interval and is symmetric, we would expect no more than 5% of the measured values to fall below the left boundary. Using this technique, we can calculate such graphs for any given confidence range allowing the left edge to be tuned as needed.

2.2 Non-parametric Methods

It is also possible to determine quantiles and confidence intervals for them without first hypothesizing an underlying distribution (such as a Weibull). The advantages of using these methods is that they do not rely on the specific properties of a particular distribution (e.g. the shape) and that they do not require a potentially large number of parameters to be estimated (each with its own concomitant estimation error).

Moreover, the complexity associated with determining the bounding curves increases exponentially with the number of parameters in the model. In the case of a two-parameter Weibull, for example, four curves are generated by the confidence intervals on the parameters corresponding to all possible combinations of low and high values for the parameters α and β respectively. For an k phase hyperexponential distribution (which may have a more conforming fit than a Weibull as we note in [17]) the number of parameters that must be estimated is $2k - 1$ yielding 2^{2k-1} different curves. In the Weibull case, it is obvious from the definition of the density function that the high estimates for both α and β will yield the lowest-valued quantiles and similarly that the lowest values for α and β yield the largest quantiles. For other distributions such as a hyperexponential the choice is less clear making an automatic implementation (that will be part of a scheduler) difficult. For these reasons, we examine two methods which are based on the data set itself and not an assumption about the distribution from which it may have been drawn.

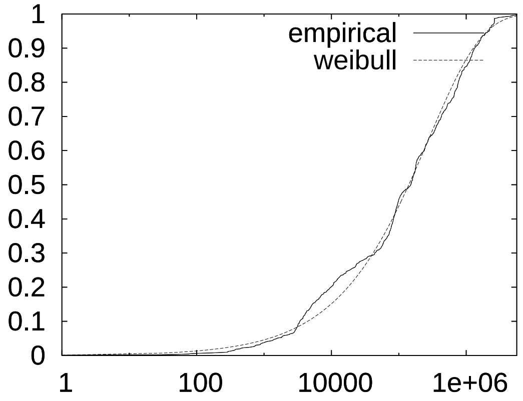

Figure 1. Empirical and MLE fitted Weibull CDFs of machine uptime data gathered from CSIL lab at UCSB ($\alpha = 0.536852$ and $\beta = 249105$).

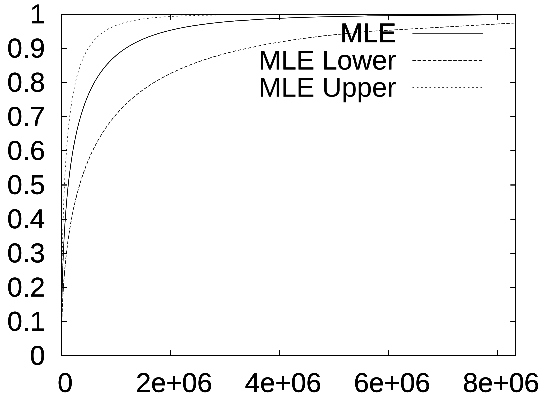

Figure 2. 90% Confidence Interval Graphs for Weibull shown in Figure 1 ($\alpha = 0.536852$ and $\beta = 249105$)

2.3 Resample Method

The first method, which we term the *Resample Method* is based on the standard result that, for given q and sample size n, the *asymptotic* distribution of the sample q^{th} quantile is normal with mean X_q, the true population q^{th} quantile and variance:

$$\frac{1}{[f(X_q)^2]}\frac{q(1-q)}{n} \tag{4}$$

where $f(X_q)$ is the population *density* function, evaluated at X_q (*Cf.* [11], pp. 367 ff.). In principle, then, to establish a 95% lower confidence bound for a specific quantile, one could use the sample quantile, calculate the standard error from the variance given in this equation, and subtract 1.645 (or a suitable critical *t*-value for the normal distribution) times the standard error from the sample mean.

To use this result non-parametrically we can only try to make a ”safe” estimate $f(X_q)$ using histograms from our data and our estimate for X_q even with reasonably large sample sizes. Since our estimated value of $f(X_q)$ is small, and this expression appears in the denominator, our formula for the variance is extremely sensitive to small variations in our estimate, rendering this histogram-based impractical in practice.

This difficulty is circumvented by using resampling methods and creating the distribution of sample quantiles artificially. The distribution of these quantiles should be, again, asymptotically normal. To the extent that this distribution approaches the asymptote, then, one constructs a level-C lower bound for X_q as the $(1-C)^{th}$ quantile of the resampled quantiles.

2.4 The Binomial Method

The second method, which we term the *Binomial Method* is based on the following simple observation: Let X be a random variable, let q be a real number between 0 and 1, and let X_q be the q^{th} quantile of the distribution of X. Then a single observation x from X will be less than X_q with probability q.

Given an independent sample $(x_1, x_2, ...x_n)$ from X, we can thus make inferences about X_q directly, without making any assumptions about the actual distribution of X. The method is as follows. The probability that none of the x_i are less than X_q is equal to $(1-q)^n$. Similarly, the probability that exactly one x_i less than X_q is $n \cdot (1-q)^{n-1} \cdot q$, and the probability that exactly j of the x_i are less than X_q is $\binom{n}{k} \cdot (1-q)^{n-j} \cdot q^j$. Therefore, the probability that k or fewer of the x_i are less than X_q is equal to

$$\sum_{j=0}^{k} \binom{n}{j} \cdot (1-q)^{n-j} \cdot q^j \tag{5}$$

Observe that this calculation is valid (not just asymptotically correct) under the sole assumption that the x_i are i.i.d. and depends only on n, k, and q.

Given a desired confidence level C and quantile of interest X_q, we can use Equation 5 above to obtain a level-C lower bound for X_q. Let $x_{(i)}, i = 1, \ldots, n$, represent the *order statistics*; that is, $(x_{(1)}, x_{(2)}, ..., x_{(n)})$ permutes the sample so that it is in increasing order. To say that we are confident with level C that $x_{(k)} < X_q$ is equivalent to saying that the *a priori* probability that $x_{(k)} \geq X_q$ is less than or equal to $1 - C$; by Equation 5, this gives the equation

$$\sum_{j=0}^{k} \binom{n}{j} \cdot (1-q)^{n-j} \cdot q^j \le 1 - C \qquad (6)$$

taking the largest k for which this equation holds gives x_k as a level-C lower bound for X_q.

It should be noted that the usual normal approximation to the binomial does not produce accurate results for relatively small sample sizes and extreme quantiles such as $X_{.05}$ with which we will typically concern ourselves.

We have investigated the possibility of enhancing the binomial method by applying linear interpolation between the two order statistics whose associated sums surround $(1-C)$ for the confidence level C of interest. As we will see in Section 4, this has proven to be very successful, since for small quantiles, both the our model and empirical CDFs tend to be quite linear. Further, for data sets which are smaller than the minimum size (59 for $X_{.05}$) which would allow us to use the binomial method conventionally, we have also investigated the possibility of using linear interpolation with absolute population minima, the results of which are shown in Section 4.

3 Experimental Data

The data we use in this study measures resource availability in three different settings. At the University of California, Santa Barbara (UCSB) we collected measurements of the time between machine reboots of the publicly accessible workstations in the Computer Science Instructional Laboratory (CSIL). In a second experiment, we measured the process occupancy time observed by a single user of the Condor [24] pool at the University of Wisconsin during a two-month period. Finally, we gratefully acknowledge Dr. Darrell Long from the University of California, Santa Cruz, and Dr. James Plank at the University of Tennessee for supplying us with the original test data used to derive the results in [15] and [19] respectively. Each of these data sets measures machine availability in a different way reflecting the different definitions of "availability" that Grid users may choose. Our goal in using a plurality of measurement methods is to determine how sensitive our quantile prediction methods are to the way in which availability is measured.

3.1 The UCSB CSIL Data Set

At UCSB, the computer science students are given unrestricted access to workstations located in several rooms on campus. Together, these systems make up the Computer Science Instructional Laboratory (CSIL). Physical access to the CSIL is provided to some (but not all) students 24-hours a day when school is in session, and via remote access at all other times to all computer science students. There are no administrator scheduled reboots when school is in session, however software failures, security breeches, and hardware failures result in unplanned restarts by the administrative staff.

What is perhaps most relevant to our study, however, is that the power switch for each workstation is not physically protected. Thus a student with access to a machine's console who does not wish to share that machine with remote users or background processes can "clean off" the machine by power cycling it. Remote users will often choose a new machine when they are unceremoniously logged out without warning, and few background processes are written to automatically restart. Indeed, it is reported anecdotally by many students that the "normal" user response to observed machine slowness is to try a power cycle immediately as a potential remedy.

As anarchistic as it may seem, we believe that this mode of usage and administration accurately reflects failure patterns in enterprise and global desktop computing settings. Users are willing to accept background computing load if it does not introduce unacceptable slowness, but will reclaim the resources they control (through catastrophic means, if need be) if the externally generated load is "too great." Obviously each user has a different tolerance level for external load that may not be known a priori, and individual user patience is likely to be be time and situation dependent. Moreover, it is the combination of user reclamations, administrative restarts, and hardware failures that make up the overall availability distribution that we observe externally.

To measure availability in the CSIL, we designed an "uptime sensor" for the Network Weather Service (NWS) [26, 22, 27] that reads the time since the last machine reboot from the `/proc` file system. All CSIL workstations currently run Linux which records the time since reboot in the `/proc}`directory. The NWS is designed to gather and maintain dynamic performance measurements from Grid resources while introducing as little load as possible. We deployed the NWS uptime-sensor on 83 of the CSIL workstations and recorded the duration between reboots during Feb and Oct of 2003, which corresponds to 2-3 quarters of the school year, one of which being the summer quarter. Thus the resultant data set captures a "production" use period for the CSIL machines as well as a period of relatively low resource usage.

3.2 The Condor Data Set

Condor [24, 9] is a cycle-harvesting system designed to support high-throughput computing. Under the Condor model, the owner of each machine allows Condor to launch an externally submitted job (i.e. one not generated by the owner) when the machine becomes "idle." Each owner is expected to specify when his or her machine can be considered idle with respect to load average, memory occupancy, keyboard activity, etc. When Condor detects that a machine has become idle, it takes an idle job from a queue it maintains, and assigns it to the idle machine for execution. If the machine's owner begins using the machine again, Condor detects the local activity and evacuates the external job. The result is that resource owners maintain exclusive access to their own resources, and Condor uses them only when they would otherwise be idle.

When a process is evicted from a machine because the machine's owner is reclaiming it (e.g. begins typing at the console keyboard), Condor offers two options. Either the evicted Condor process is checkpointed and saved for a later restart, or it is killed. Condor implements checkpointing through a series of libraries that intercept system calls to ensure that a job can be properly restarted. Using these libraries, however, places certain restrictions on the system calls that the job can issue. "Vanilla" jobs, however, are unrestricted but will be terminated (and not checkpointed) during a resource reclamation. Condor's extensive documentation [10] details these features to a greater extent.

In this study, we take advantage of the vanilla (i.e terminate-on-eviction) execution environment to build a Condor occupancy sensor for the NWS. A set of sensors (10 in this study) are submitted to Condor for execution. When Condor assigns a sensor to a processor, the sensor wakes periodically and reports the number of seconds that have elapsed since it began executing. When that sensor is terminated (due to an eviction) the last recorded elapsed time value measures the occupancy the sensor enjoyed on the processor it was using. The NWS associates measurements with Internet address and port number so if a sensor is subsequently restarted on a particular machine (because Condor determined the machine to be idle) the new measurements will be associated with the machine running the sensor.

It is difficult to determine how many machines are available within the Wisconsin Condor pool. The number fluctuates as new machines are added, users decommission old machines, etc. In our study, however, Condor used 210 different Linux workstations to run the 10 NWS sensors over a 5 month measurement period.

Notice also that in this study we consider only the availability of each machine to a Condor user (the NWS, in our case) once the machine is assigned to the NWS. We do not consider the time between assignments during which a particular machine is either busy because its owner is using it, or because Condor as scheduled other useful work. In the CSIL data set, these durations are between 120 and 600 seconds which is the Linux reboot time, depending on the machine in question. For Condor, however, the distribution of resource unavailability is not as constant. Any complete simulation of the Condor pool as a computational engine would require both the distribution of availability and the distribution of unavailability. In this work, we treat only the availability distribution, but we plan a full analysis of Condor's dynamics in the near future.

3.3 The Long-Muir-Golding Data Set

In [15] the authors identify 1170 hosts connected to the Internet in 1995 that would cooperatively respond to a vacuous query of the `rpc.statd` – a system process commonly used on systems running the Network File System (NFS). The hosts were chosen to act as a "cross-section" of the Internet connected hosts at the time, and a probing mechanism based on periodic but randomized RPC calls to `rpc.statd`. A successful response to an RPC constitutes a "heartbeat" for the machine in question, and failure to respond indicates machine failure. Long, Muir, and Golding use this data to make a convincing argument that availability is not accurately modeled by a Poisson process. More recently Plank and Elwasif [19] and separately Plank and Thomason [20] have analyzed it extensively in terms of the suitability of Poisson and exponential models in the context of process checkpoint scheduling. In all three studies, the authors reach the same conclusion which is that the models under study do not accurately reflect the behavior captured by the measurements.

3.4 Discussion

We have chosen to study these three data sets because they measure observable machine availability in different ways, under different conditions, at different times. For the CSIL data set, students engaged in collaborative and competitive activities using the resources at hand strongly influence the measured availability durations. Under Condor, availability measurements capture the idle-busy distribution of resource owners who (in theory) are unaware that Condor is using the resources during idle periods. From the perspective of a Grid or peer-to-peer scheduler, however, these two data sets record the same quantities: the amount of time an application process was able to use a resource before it (the process) was exogenously terminated. We include the Long-Muir-Golding (LMG) data set in our study to ensure that our results are not biased by the measurement techniques we have used. The CSIL and Condor data sets measure availability using two different sensors we have developed for the NWS monitoring infrastructure. As a result, we wished to use data gathered by a separate group using different measurement techniques to remove the possibility that the NWS is biasing the results in an unforeseen way.

Note that the age of the LMG data also indicates the time sensitivity (i.e. non-stationarity) of the effects we observe. Clearly the Internet and its usage patterns have evolved substantially since they gathered the data. Observing similar distributions in all three data sets indicates that the effects we are measuring are persistent and potentially fundamental.

4 Analysis

To investigate the effectiveness of the techniques described in the previous section, we compare the predictive performance of a two-parameter Weibull model which we term *the Weibull method* (c.f. Section 2.1) to the non-parametric *Resample method* and the *Binomial method*. Our first test attempts to verify the predictive power of each method using a synthetic distribution of availability measurements generated from a known distribution. This investigation compares each method's ability to recover a lower bound on given quantile (which is known exactly for the distribution) using different sample sizes. It also illustrates the smallest sample size for which these methods can generate accurate results. We then apply the methods to machines culled from the data sets described in Section 3 and detail their relative accuracy.

4.1 Verifying Predictive Power

In this section, we verify the efficacy of the three methods to estimate a specific quantile lower bound given a sample from a known distribution. Based on the results indicating that machine/process lifetime data typically can be modeled using a heavy tailed distribution such as that shown in Figure 1, we can characterize synthetic machine availability using such a distribution. As a population distribution we choose use a two-parameter Weibull distribution with a shape parameter $a = 0.540976$ and a scale parameter $b = 283068$ as we believe it is typical of an accurate heavy-tailed model for availability measurements. Thus, the distribution of the population is known exactly, and samples drawn from it are consistent with observed machine availability [17].

From this population distribution, we can calculate quantiles numerically. For example, the 0.05 quantile for the Weibull we use in this study is 1167.9. Hypothetically, any machine with this availability distribution will "survive" longer than 1167.9 time units "95% of the time" since the probability of a value being larger than the 0.05 quantile is 95%.

We can determine the accuracy (as a function of sample size) associated with a given quantile bound estimation method by repeatedly sampling the synthetic population distribution with a fixed sample size, using the sample to estimate a quantile lower bound, and then comparing the estimate to the actual known quantile. If we repeat this procedure for a large number of random samples, we can record the percentage of the estimates that are above or below the true estimate as a measure of confidence in the estimate.

For example, we can draw 1000 samples of size 100 from the population distribution. For each sample, we calculate the 0.05 quantile lower bound with 95% confidence using a particular method (Weibull, Resample, or Binomial). We then calculate the percentage of the 1000 estimates generated by the method that are below the true 0.05 quantile for the synthetic population distribution.

In the synthetic case, we also examine how well a parametric model captures the true quantile (instead of the lower bound). Since we have chosen a two-parameter Weibull for the population distribution and we are using a two-parameter Weibull as the parametric model, this method (termed the *Quantile method*) represents the most ideal parametric setting. Our expectation is that 50% of the Quantile Method estimates will be below the true quantile. The motivation for this expectation is that sample quantiles should be asymptotically normal [11] with center at the true quantile, and even if the lower bound is skewed because of a small sample size, the center may be better represented. However, as Table 1 and Figure 3 reflect, even for sample sizes as large as 100, the sample .05 quantile shows substantial bias (and in fact its distribution is still noticeably skewed as well). For the quantile lower bound methods, we expect the percentage of values to be equivalent to one minus the confidence level. For example, if we are estimating the lower bound of the 0.05 quantile with 95% confidence, we expect 95% of the estimates to be below the true quantile (if the confidence bound is accurate).

Table 1 shows the results generated by calculating the 0.05 quantile estimate using three different methods and a variety of sample sizes. In each case, we apply the relevant method to a sample drawn from the synthetic population to obtain an estimate for the 0.05 quantile, and then we record whether the estimate is above or below the true 0.05 quantile 1167.9. The first column shows the sample size, and the percentage of 1000 samples that are above the true 0.05 quantile are shown in the remaining columns.

From this table it is clear, not surprisingly, that sample size dramatically affects the accuracy associated with an estimation method. It is, in fact, striking that the Binomial Method captures the desired percentage of quantiles for a sample as small as 20. The Resample Method shows reasonable success using 100 samples, but drops dramatically when sample sizes are below 50. The Weibull Method, our only parametric lower bound estimation technique, shows a very low success rate, even with sample sizes as large as 100. For instance, the Weibull Method, using samples of size 100, resulted in a mere 53.3% of the computed lower bound values $<=$ the true distribution quantile even when the population distribution itself is Weibull. In contrast, Resample Method starts showing significant failure with sam-

Subsample Size	Weibull Method	Binomial Method	Resample Method	Quantile Method
100	53.3	95.0	91.2	54.3
50	51.1	96.7	93.2	56.5
20	48.1	94.0	62.6	59.2
10	47.0	86.0	50.3	49.0

Table 1. Percent estimated 0.05 lower bounds (Weibull, Binomial, and Resample Methods) and 0.05 quantile (Quantile Method), for four different subsample sizes, that are less than or equal to the true 0.05 quantile from a set Weibull distribution $a = 0.540976$, $b = 283068$.

Subsample Size	Weibull Method Variance	Binomial Method Variance	Resample Method Variance
100	255110	175510	263100
50	499760	204790	484990
20	2730100	340710	17514000
10	13234000	1194300	51805000

Table 2. Variances of quantile lower bound estimates for two methods for four difference subsample sizes.

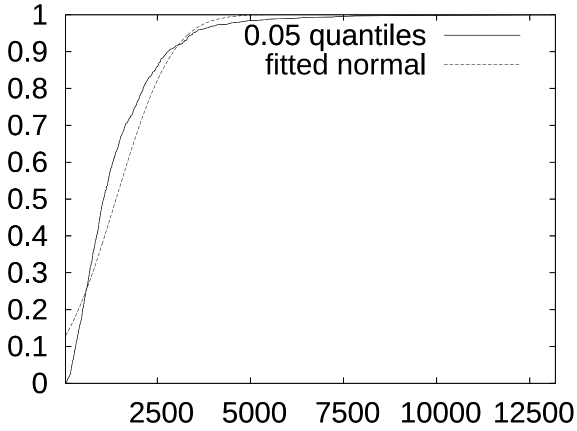

Figure 3. CDF of 0.05 quantiles from 1000 subsamples (100 samples each) from a set Weibull distribution with a fitted normal CDF. The mean of the quantiles was 1391.8, the true population quantile was 1167.9, indicating considerable bias.

ple sizes below 50, while the Binomial Method results in high success rates until the sample size is extremely small, somewhere between 10 and 20 samples.

This sample size estimate is important because it indicates the minimum number of measurements a particular machine must have before an estimate, or a confidence bound on an estimate, can be trusted. In the remainder of this study, we will assume that predictions should be made using as little data as possible. Moreover, because a Weibull model does fit true machine trace data well, we believe a sample size of 20 will give us reliable results for non-synthetic trace data.

Table 2 shows a comparison of the variance of the estimators from the three quantile bound estimation methods; in addition to the percentage of reliability that an estimator provides, it is desirable that the variance of this estimator be relatively small, so that it is not overly susceptible to sampling variation. Notice that at every sample size in the table, the variance of the Binomial Method estimator is considerably smaller than that of the Resample Method and Weibull Method estimators. Thus, not only does the Binomial Method produce more reliable confidence bounds over the full range of sample sizes, but the numbers obtained are “tighter” in the sense of being less variable from one random sample to the next.

4.2 Empirical Results

From our verification experiment, we have learned that our methods for predicting lower bounds on quantiles are empirically functioning as theorized but also that the method based on binomials gives us accurate results at a far smaller sample size than the method of resampling. While it is clear that these results hold up for a well-behaved model distribution, the next step is to explore the usefulness of our technique with the real world lifetime data outlined in Section 3.

In our experiment, we apply each of the estimation methods to availability traces for individual machines. For each machine, we divide availability times into a *training period* and an *experimental period* which follows it chronologically. The training period data is used to derive a lower-bound estimate (from each of the three estimation methods) that is then tested against the experimental period. Thus, the results of this study demonstrate how well each method uses observed data (occurring during the training period) to predict future values (in the experimental period) for individual machines.

From our verification experiment, we noticed that 20 samples were sufficient for making an accurate lower bound estimate using the Binomial Method, and therefore use machines whose training and experimental periods contains 20 measurements or more. We theorize that both methods, if functioning properly, should result in estimates similar to those we observed in the verification experiment. The need for data parsimony is also demonstrated by this experiment. In each of the three data sets (CSIL, Condor, and Long-Muir-Golding) there were few machines with more than 40 availability measurements over the observation period covered by each data set.

Table 3 shows the result of our experiment for the three lower bound quantile estimation methods and three different data sets. The values reported are percentages of individual machine traces in each set where the method was "successful." For a given machine trace and method, we record a success if the method estimate from the training period is <= 95% to 100% of the measurements from the experimental period.

The results of this experiment are very close to the results we observed from our verification experiment and clearly show that even when using real machine and process lifetime data, with as little as 20 measurements upon which to base an estimate, the Binomial Method finds a value <= 95% of future values for 87.5% to 98.9% of the machine traces. The Resample Method, under the same conditions, only made correct estimates for 53.4% to 62.4%of the machines. The Weibull Method shows the same approximate success rate we would expect on the CSIL and Long data, but leapt to a dramatic 95.92% for the Condor data. The reason for this high success rate, unfortunately, most likely stems from the fact that the Condor data is not very well modeled using a Weibull distribution and made extremely conservative estimates during the experiment. In total, out of 202 combined experiments, the Binomial Method produced a valid lower bound for the $.05$ quantile a total of 96.0% of the time, while the Resample and Weibull Methods succeeded in only 57.5% and 75% of the cases respectively.

We emphasize that the strength of the result using the Binomial Method is not just the high success rate, but the fact that the success rate is so close to the target of 95%. (In fact, the observed percentage provide no evidence at any significance level $[P = .628]$ of a global success rate different from exactly 95%.) The method has thus produced lower bounds that are meaningful at the desired level of confidence, rather than being too conservative and producing a success level too close to 100%. This converse can be seen in the case of the Weibull Method on Condor data, where we obtained a very high success rate, but the estimates themselves were extremely conservative. As such the success rate actually exceeds the 95% target, which indicates that this method is not exhibiting evidence that the confidence level can be meaningfully specified.

4.3 Discussion

Our experiment has allowed us make predictions that address the question of how long we expect a machine or process to live 95% of the time. It is noteworthy that the Binomial Method is making successful predictions 96.0% of the time using only 20 measurements. However, the predictions we investigate are unconditional. As a result, they predict how long a machine will be available at the time it reboots. In practice, a scheduler would want a prediction of *remaining availability* from any arbitrary point in time. Since the underlying population distributions are not likely to be "memoryless" the conditional prediction of remaining lifetime will depend on how long a particular machine has been running at the time the prediction is made. Moreover, the population distributions do seem to be well-modeled by heavy-tailed distributions such as a Weibull or hyperexponential [17], the unconditional estimates we generate in this study are necessarily conservative. That is the longer a process has already "lived," the longer it is likely to live, and therefore the expected availability from a random point in time is *longer* than the expected total availability from its last restart. Note also that while we can control the confidence bound on the unconditional prediction, and that the Binomial Method yields the tightest bounds, we cannot yet quantify how conservative the unconditional prediction is with respect to a conditional one. As part of of our future work, we are investigating similar methods for making instantaneous conditional predictions that are less conservative, but which require the scheduler to interact with the predictive methodology.

Data Set	Number of Machines	% Weibull Success	% Binomial Success	% Resample Success
CSIL	16	56.25	87.5	62.5
Condor	97	95.92	98.9	60.2
Long	83	57.95	94.3	53.4

Table 3. Percent of machine traces from each set for which the given method estimate was $<=$ 95% to 100% of the experimental period measurements.

5 Conclusions

In this work, we have shown three methods for establishing a confidence lower bound on a population quantile from a population sample. To gauge the effectiveness of each method, we performed a verification experiment that used the methods to form confidence bounds using random samples of fixed size from a known population distribution. For the 0.05 quantile, our experiment strongly indicated that the Binomial Method performed far better than the Resample Method and Weibull Method on small subsample sizes. We attribute the fact that the Resample Method fails to perform well with small sample size to the underlying assumption of the method, which is that subsample quantiles from a population are asymptotically normally distributed. The poor performance of the Weibull Method indicate that the parameter estimate technique (MLE) is very sensitive to variations in the given data, and as such produce models from small subsample sizes that do not accurately capture the true population distribution. Our verification experiment showed that with samples as small as 20, the Binomial Method is the only tested method that successfully estimates a lower bound on the 0.05 population quantile.

With these results in hand, we performed a similar experiment using real process and machine uptime data collected from three separate sources. In this experiment, we split a machine or process lifetime trace into a training set of size 20, and an experimental set of at least size 20. We then use our three lower bound quantile estimation techniques on the training period and count how many of the measurements from the experimental period were greater than our lower bound estimate. The results of this experiment are similar to our verification experiment, revealing the fact that the neither the Resample Method nor the Weibull Method captures, with only 20 samples, the true 0.05 population quantile, while the Binomial Method correctly estimates a lower bound the expected number of times during the experimental period. We performed this test on 202 different, individual machine and process lifetime traces and found that for 96.0% of the traces the Binomial Method is correctly producing a lower bound, while the Resampling Method only produced a lower bound for 57.4% of the traces. The Weibull Method produced success 56-58% of the time for the machine availability data sets, and produced a somewhat high 95.92% success rate for the Condor data set which unfortunately is due to the inappropriateness of the Weibull model when applied to Condor traces and results in an unusable conservative quantile estimates.

Although we have produced a very effective method for estimating lower bound quantile estimates (Binomial Method), we realize that these estimates, as 95% confidence lower bounds, will be quite conservative. For the purpose of determining the future lifetime of an existing process, our estimates will be particularly conservative, due to the heavy-tailed nature of the distribution of availability measurements. In future work, we will attempt to adapt the methods of the current work in order to address data distributions of this type.

References

[1] B. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Leigh, A. Sim, and A. Shoshani. High-performance remote access to climate simulation data: A challenge problem for data grid technologies. In *Proceedings of IEEE SC'01 Conference on High-performance Computing*, 2001.

[2] The Avaki Home Page. `http://www.avaki.com`, January 2001.

[3] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, L. J. Dennis Gannon, K. Kennedy, C. Kesselman, D. Reed, L. Torczon, , and R. Wolski. The GrADS project: Software support for high-level grid application development. *International Journal of High-performance Computing Applications*, 15(4):327–344, Winter 2001.

[4] F. Berman, G. Fox, and T. Hey. *Grid Computing: Making the Global Infrastructure a Reality*. Wiley and Sons, 2003.

[5] F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, and D. Zagorodnov. Adaptive computing on the grid using apples. *IEEE Transactions on Parallel and Distributed Systems*, 14(4):369–382, April 2003.

[6] The bovine rc5-64 project – `http://distributed.net/rc5/`.

[7] H. Casanova and J. Dongarra. NetSolve: A Network Server for Solving Computational Science Problems. *The International Journal of Supercomputer Applications and High Performance Computing*, 1997.

[8] H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS Parameter Sweep Template: User-Level Middleware for the +Grid. In *Proceedings of IEEE SC'00 Conference on High-performance Computing*, Nov. 2000.

[9] Condor home page – `http://www.cs.wisc.edu/condor/`.

[10] The Condor Reference Manual. `http://www.cs.wisc.edu/condor/manual`.

[11] H. Cramer. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[12] The Entropia Home Page. `http://www.entropia.com`.

[13] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 1997.

[14] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., 1998.

[15] D. Long, A. Muir, and R. Golding. A longitudinal survey of internet host reliability. In *14th Symposium on Reliable Distributed Systems*, pages 2–9, September 1995.

[16] MATLAB Home Page. `http://www.mathworks.com`.

[17] D. Nurmi, J. Brevik, and R. Wolski. Modeling machine availability in enterprise and wide-area distributed computing environments. Technical Report CS2003-28, U.C. Santa Barbara Computer Science Department, October 2003.

[18] A. Petitet, S. Blackford, J. Dongarra, B. Ellis, G. Fagg, K. Roche, and S. Vadhiyar. Numerical libraries and the grid. In *Proceedings of IEEE SC'01 Conference on High-performance Computing*, November 2001.

[19] J. Plank and W. Elwasif. Experimental assessment of workstation failures and their impact on checkpointing systems. In *28th International Symposium on Fault-Tolerant Computing*, pages 48–57, June 1998.

[20] J. Plank and M. Thomason. Processor allocation and checkpoint interval selection in cluster computing systems. *Journal of Parallel and Distributed Computing*, 61(11):1570–1590, November 2001.

[21] M. Ripeanu, A. Iamnitchi, and I. Foster. Cactus application: Performance predictions in a grid environment. In *proceedings of European Conference on Parallel Computing (EuroPar) 2001*, August 2001.

[22] J. Schopf and J. Weglarz. *Resource Management for Grid Computing*. Kluwer Academic Press, 2003.

[23] SETI@home. `http://setiathome.ssl.berkeley.edu`, March 2001.

[24] T. Tannenbaum and M. Litzkow. The condor distributed processing system. *Dr. Dobbs Journal*, February 1995.

[25] The United Devices Home Page. `http://www.ud.com/home.htm`, January 1999.

[26] R. Wolski. Experiences with predicting resource performance on-line in computational grid settings. *ACM SIGMETRICS Performance Evaluation Review*, 30(4):41–49, March 2003.

[27] R. Wolski, N. Spring, and J. Hayes. The network weather service: A distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 15(5-6):757–768, October 1999.