# Modeling Machine Availability in Enterprise and Wide-area Distributed Computing Environments*

*UCSB Computer Science Technical Report Number CS2003-28*

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

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

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

## Abstract

*In this paper, we consider the problem of modeling machine availability in enterprise-area and wide-area distributed computing settings. Using availability data gathered from three different environments, we detail the suitability of four potential statistical distributions for each data set: exponential, Pareto, Weibull, and hyperexponential. In each case, we use software we have developed to determine the necessary parameters automatically from each data collection.*

*To gauge suitability, we present both graphical and statistical evaluations of the accuracy with each distribution fits each data set. For all three data sets, we find that a hyperexponential model fits slightly more accurately than a Weibull, but that both are substantially better choices than either an exponential or Pareto. We also test the independence of individual machine measurements and the stationarity of the underlying statistical process model for each data set.*

*These results indicate that either a hyperexponential or Weibull model effectively represents machine availability in enterprise and Internet computing environments.*

*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.

## 1 Introduction

As performance-oriented distributed computing (often heralded under the moniker "Computational Grid" computing [25, 8, 49]) becomes more prevalent, the need to characterize accurately resource reliability emerges as a critical problem. Today's successful Grid applications uniformly rely on runtime scheduling [9, 44, 2, 4, 48, 1, 17, 52, 13, 12, 40] to identify and acquire the fastest, least-loaded resources at the time an application is executed. While these applications and systems have been able to achieve new performance heights, they all rely on the assumption that resources, once acquired, will not fail during application execution. In many resource environments such an assumption is valid, but to employ nationally or globally distributed resource pools (e.g. in the way SETI@Home [50] does) or enterprise-wide desktop resources (as many commercial endeavors do [8, 23, 57, 5]) performance-oriented distributed applications must be able either to avoid or tolerate resource failures.

Designing the next-generation of Grid applications requires an accurate model of resource failure behavior. A great deal of previous work [39, 32, 26, 30, 33] has studied the problem of modeling resource failure (or equivalently resource availability) using statistical techniques. As Plank and Elwasif point out in their landmark paper [46], however, most of these approaches assume that the underlying statistical behavior can be described by some form of exponential distribution or hyperexponential distribution [33]. In addition, they go on to note that despite their popularity, many of these modeling techniques do not accurately reflect empirical observation of machine availability.

Our goal with this work is to develop an automatic method for modeling the availability of enterprise-wide and globally distributed resources. Automatic model determination has several important engineering applications. We plan to incorporate such models into Grid programming systems, such as the Grid Application Development Software [7] system, NetSolve [12], NINF [40], and APST [14] to enable effective resource allocation and scheduling. Commercial enterprise-computing systems such as Entropia [23], United Devices [57], and Avaki [5] will also be able to take advantage of automatically determined models as they tune themselves to the characteristics of a particular site. Moreover, much of the potential benefit offered by Autonomic computing [31] depends, critically, on the ability to model resource characteristics automatically.

We propose a new approach to modeling machine availability based on either the Weibull family of distributions or hyperexponentials depending on the intended use of the model. We describe how to estimate the necessary parameters from a given set of availability measurements, and the implementation of our system for doing so automatically. To gauge the effectiveness of our modeling methodology, we detail and analyze the degree to which an automatically generated model fits three diverse sets of empirical observations:

- reboot intervals taken from the student workstations located in the Computer Science Department of the University of California, Santa Barbara,
- processor occupancy duration measured from the Condor [55, 18] deployment at the University of Wisconsin, and
- availability data gathered by Long, Muir, and Golding in a survey they conducted of Internet hosts, described in [35] and further analyzed in [46, 47].

We compare the distributions generated by our method to both standard exponential and Pareto distributions fit to the data using the same parameter estimation techniques.

Exponential distributions have been studied extensively in fault tolerant computing settings [58, 35, 46, 47, 36]. More recently, peer-to-peer systems have used exponential distributions to as the basis of their availability assumptions [53, 62, 63]. In other contexts such as process lifetime estimation [29] and network performance [43, 42, 59, 20, 34] researchers often advocate the use of "heavy-tailed" distributions, especially the Pareto. We also compare the use of both a Weibull and a hyperexponential to that of a Pareto for modeling our data. In both cases, using a variety of goodness-of-fit metrics, the distributions generated by our method are a significantly better fit for each data set.

The data sets we study are gathered in three different distributed computing contexts, at different times, using different methods. The diversity of the conditions under which each trace was gathered indicates the generality of our results in that regardless of setting or method, either a Weibull or hyperexponential distribution appears to model availability most effectively. More specifically, using p-values from a variety of goodness-of-fit tests as a metric, an appropriately chosen hyperexponential fits each data set best. However, the Weibull model for each set, while slightly less well-fit, offers several attractive properties that make it a better choice in some modeling contexts. Our methodology generates both a "best-fit" Weibull and hyperexponential for each data set, allowing the user to choose between the two.

In addition to the impact this work may have on peer-to-peer system design, checkpoint/migration interval determination, and process scheduling, we believe it is particularly important to the development of credible and effective Grid and Autonomic Computing [31] simulations. Because Grid dynamics are driven by the dynamic resource sharing of competing users, repeatable "en vivo" experiments are difficult or impossible. Several effective emulation [51, 16] and simulation [11, 54, 10] systems have been developed for Grid environments. These systems will benefit immediately from the more accurate models our method produces.

The rest of this paper is organized as follows. Section 2 describes the both the Weibull and hyperexponential distributions and our method of fitting them to a set of machine availability measurements. We discuss how we address both the problem of parameter estimation and how we treat censored data. In Section 3 discusses the various data sets we use in this study. In Section 4 we provide evidence for how well various distributions fit each data set, and review the advantages and disadvantages of each in Section 5. Finally, in Section 6 we discuss the conclusions we draw from this work and point to future research directions it enables.

## 2 Fitting a Distribution to Availability Data

In this study, the two distribution families that consistently fit the data we have gathered most accurately are the Weibull and the hyperexponential. The *Weibull distribution* is often used to model the lifetimes of objects, including physical system components [45, 15, 38, 6]. Hyperexponentials have been used to model machine availability previously [39], but their parameters are more numerically difficult to estimate through rigorous statistical techniques. In particular, the number of phases (c.f. Section 2.3) to use is a free parameter that our method determines by fitting successively larger models. Our algorithm terminates when an additional phase fails to improve the goodness-of-fit. While in practice, the convergence of p-value for a given goodness-of-fit test indicates that no additional phases are needed, in principle this technique must be considered a heuristic. As a heuristic, however, we find that the quality of the fits gen-

erated to be high using a relatively small number of phases.

### 2.1 Weibull Distributions

The density and distribution functions $f_w$ and $F_w$ respectively for a Weibull distribution are given by

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

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

The parameter $\alpha$ is called the *shape* parameter, and $\beta$ is called the *scale* parameter. [1] When $\alpha = 1$, the Weibull is equivalent to an exponential distribution.

The conditional distribution function for a Weibull is given by

$$F_{X|X>t}(x) = 1 - e^{[(t/\beta)^\alpha - (x/\beta)^\alpha]}, \quad (3)$$

which clearly depends on $t$ and not just the difference $x - t$ when $\alpha \neq 1$. When $0 < \alpha < 1$, the probability that a component will survive another time unit *increases* as $t$ increases. For $\alpha > 1$, this probability *decreases*, and when $\alpha = 1$ the distribution is memoryless. Thus a Weibull distribution is capable of modeling different aging effects, depending on its shape parameter.

A hyperexponential, on the other hand, is only capable of modeling increasing expected lifetime. One can show this by demonstrating that the *hazard function*, which is essentially the rate of failure, is a decreasing function of time for any hyperexponential; this is a straightforward but tedious calculus exercise. Intuitively, throughout the lifetime of a hyperexponentially distributed object, its having survived as long as it has makes it increasingly conditionally probable that its lifetime is governed by the longer phases of the hyperexponential, and so its expected future lifetime will increase.

### 2.2 Weibull Parameter Estimation

Our 2-parameter Weibull, as mentioned above, has parameters for shape and scale. Given a set of sample data $\{x_1 ... x_n\}$, there are many common techniques for estimating the two parameters based on some set of sample data, including visual inspection (e.g. using a two-dimensional graph) and analytic methods. A commonly accepted approach to the general problem of parameter estimation is based on the principle of *maximum likelihood*. The maximum likelihood estimator (MLE) is calculated for any data set, based on the assumptions that each of the sample data points $x_i$ is drawn from a random variable $X_i$ an that the $X_i$ are independent and identically distributed (i.i.d.). The method defines the *likelihood function* $L$, depending on the parameters of the distribution, as the product of the density function evaluated at the sample points. Thus in our case, $L$ will be a function of $\alpha$ and $\beta$ given by

$$L(\alpha, \beta) = \prod_i f(x_i) = \prod_i \alpha\beta^{-\alpha}{x_i}^{\alpha-1}e^{-(x/\beta)^\alpha}.$$

Intuitively, maximizing $L$ is equivalent to maximizing the joint probability that each random variable will take on the sample value. Large values of the density function correspond to data that is "more likely" to occur, so larger values of $L$ correspond to values of the parameters for which the data was "more likely" to have been produced. Thus, the MLE for the parameters is simply the choice of parameters (if it exists) which maximizes $L$. Equivalently,we can maximize the *log-likelihood function* $\log L$, which is simpler to compute because it converts the above product into a sum. In practice, Weibull MLE values always exist. Our approach to computing them numerically is to set the partial derivatives of $\log L$ equal to 0 and using standard non-linear equation solvers to find the critical point corresponding to the maximum of $\log L$.

The other common analytic approach to parameter estimation is the method of *moments*. In the case of a 2-parameter Weibull, the moments-based estimator will be the set of parameters for which the mean and variance of the distribution is equal to the mean and variance, respectively, of the given sample. Moments-based estimators have historically been popular because of their relative ease of calculation, but MLEs enjoy more properties which are considered desirable for estimators. (Specifically MLEs are, under very general conditions, *asymptotically efficient*, which means roughly that the variance of an MLE approaches the theoretical minimum, while moments-based estimators are not asymptotically efficient in general.)

### 2.3 Hyperexponential Distributions

Hyperexponentials are distributions formed as the weighted sum of exponentials, each having a different parameter. The density function is given by

$$f_H(x) = \sum_{i=1}^{k} p_i \cdot f_{E_i}(x)], \_x \geq 0 \quad (4)$$

where

$$f_{E_i}(x) = \lambda_i e^{-\lambda_i x} \quad (5)$$

defines the density function for an exponential having parameter $\lambda_i$. In the definition of $f_H(x)$, all $\lambda_i \neq \lambda_j$ for $i \neq j$,

[1] The general Weibull density function has a third parameter for *location*, which we can eliminate from the density simply by subtracting the minimum lifetime from all measurements. In this paper, we will work with the two-parameter formulation.

and $\sum_{i=1}^{k} p_i = 1$. The distribution function is defined as

$$F_H(x) = 1 - \sum_{i=1}^{k} p_i \cdot e^{-\lambda_i x} \quad (6)$$

for the same definition of $f_{e_i}(x)$. Thus, to fit a hyperexponential to a given data set, the value of $k$, each $\lambda_i$ and each $p_i$ must be estimated. For a specified value of $k$ (which indicates how many phases will be included in the hyperexponential), an MLE technique can be used to determine the remaining $2k - 1$ parameters. However, the optimization problem that arises for even small values of $k$ is often too complex for commonly available computers to solve, especially for larger data sets. [2] As a result, we used the EMpht software package [22] in place of an MLE procedure for all estimated hyperexponentials in this paper. EMpht implements the estimation maximization (EM) algorithm described in [3]. While this technique often yields a good solution (as is evidenced by our results) is is not guaranteed to converge to an optimal solution.

The number of exponential phases (denoted by $k$) that make up a hyperexponential, however, is a free parameter that must be specified rather than estimated. Our approach is is to use EMpht to estimate parameters for successively larger values of $k$ and then to calculate goodness-of-fit metrics (as described in Section 4.2) for each. The algorithm terminates when an additional phase produces no improvement in the metrics.

We have implemented a software system that takes a set of measurements as an ordinary text file and computes both the MLE Weibull and the EM-based hyperexponential automatically. Perhaps unsurprisingly, the quality of the numerical methods that we use is critical to the success of the method. In particular, the MLE computations involve hundreds or thousands of terms (the data sets can be quite large) requiring robust and efficient techniques. At present, the implementation uses a combination of the Octave [41] numerical package, Mathematica [37] (for solver quality), and the afore mentioned EMpht. The resulting system, however, takes data (as described in Section 3) and automatically determines the necessary parameters.

### 2.4 Exponential and Pareto Distributions

The probability density functions (denoted using lower-case $f$ with a subscript) and distribution functions (upper-case $F$ with a subscript) for the exponential and Pareto distributions are as follows:

$$f_e(x) = \lambda e^{-\lambda x} \quad (7)$$

$$F_e(x) = 1 - e^{-\lambda x} \quad (8)$$

$$f_p(x) = \frac{\alpha \beta^{\alpha}}{x^{\alpha+1}} \quad (9)$$

$$F_p(x) = 1 - \left(\frac{\beta}{x}\right)^{\alpha} \quad (10)$$

Note that these techniques say nothing about how well a data sample "fits" a distribution. Rather, it determines what the most likely parameterization must be if the sample is assumed to come from a specified family of distributions. Thus, for a given data sample, we can find the Weibull, exponential and Pareto distributions that are "most likely" to have generated that sample by finding the MLE parameter estimates for each distribution (using the root-finding method described earlier). For the hyperexponential, however, the distribution that is chosen is appropriate (that is likely), but cannot strictly be said to be the one that maximizes likelihood.

## 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 publically 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 [55] 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 [35] and [46] 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 Weibull-based models 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,

[2] While we were able to make MLE estimates for Weibull and Pareto distributions for all data sets using a Pentium IV running Linux, the same numerical algorithms failed for all hyperexponential estimations.

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) [60, 49, 61] 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 April and May of 2003, which corresponds to the bulk of the spring quarter. Thus the resultant data set captures a "production" use period for the CSIL machines and does not span a quarter break during which a correlated reboot (for quarterly maintenance) is likely.

### 3.2 The Condor Data Set

Condor [55, 18] 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 unexecuted 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 [19] 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 the six-week 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 [35] 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 [46] and separately Plank and Thomason [47] 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

The goal of our study is to determine the value of using Weibull and hyperexponential distributions to model resource availability. Our methodology is to compare the MLE determined Weibull and EMpht determined hyperexponential to the MLE exponential and Pareto for each of the data sets discussed in the previous section. For reference, we have included the MLE and EMpht determined model parameters that were used for all fitted distributions discussed and shown in this work (Table 1). As we noted in the introduction, both exponential and the Pareto models have been used extensively to model resource and process lifetime. Thus the value we perceive is the degree to which a Weibull and hyperexponential model more accurately fits each data set.

In each case, we use three different techniques to evaluate model fit: graphical, the Kolmogorov-Smirnov [21] (KS) test, and the Anderson-Darling [21] (AD) test. Graphical evaluation is often the most compelling methodology [56] but it does not provide the security of a quantified result. The other two tests come under the general heading of "goodness-of-fit" tests [3]

### 4.1 Graphical Analysis of The Availability Measurements

To gauge the fit of a specific model distribution to a particular data set, we plot the cumulative distribution function (CDF) for the distribution and the empirical cumulative distribution for the data set. The form of the CDF for the Weibull, hyperexponential, exponential and Pareto are given by equations 2, 6, 8, and 10 respectively (c.f. Section 2). The empirical distribution function (EDF) is the CDF of the actual data; it is calculated by ordering the observed values as $X_1 < X_2 < \cdots < X_n$ and defining

$$F_e(x) = \begin{cases} 0, & x < X_1; \\ j/n, & X_j \le x < X_{(j+1)}; \\ 1, & x \ge X_n. \end{cases} \qquad (11)$$

We start by comparing the empirical observations to the CDF determined by the MLE estimated Weibull. As Figure 1, 2 and 3 show, a Weibull distribution appears to track

[3] The best known goodness-of-fit test is based on the Chi-squared distribution. Both the Kolmogorov-Smirnov and the Anderson-Darling tests are thought to be more appropriate for continuous distributions than the Chi-squared test, which is designed for categorical data. We therefore use these methods in place of the more familiar one.

| Data Set | Weibull | | Hyperexponential | | | | | | Exponential | Pareto | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | $p_1$ | $p_2$ | $p_3$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda$ | $\alpha$ | $\beta$ |
| CSIL | .545 | 275599 | .464 | .197 | .389 | .00000111 | .000195 | .00000832 | 2177800 | .087 | 1 |
| Condor | .49 | 2403 | .592 | .408 | NA | .00296 | .0000750 | NA | .00018 | .149 | 1.005 |
| Long | .61 | 834571 | .282 | .271 | .474 | .000000305 | .0000124 | .00000139 | 78886000 | .079 | 1 |

**Table 1. Table of fitted model parameters**

the observed distribution in each case. The track is never perfect, but the shape and scale of the model appear to be well suited to the trends in the observed data.

Similarly, Figures 4, 5, and 6 show the results of fitting a three phase, two phase, and three phase EMpht generated hyperexponential to the CSIL, Condor, and Long datasets respectively. As was previously mentioned, the choice of number of phases is a value specified by the user when attempting to fit a hyperexponential using the EMpht software. To determine the number of phases to report in the visual analysis, we start with a 2 phase hyper exponential, test the resulting fit with a Kolmogorov Smirnov test (detailed in the Section 4.2), and then repeat with an increased number of phases until the KS test result shows no improvement. As the figures again show, the fitted hyperexponential appear to track the EDF functions extremely well, so much so that the model curve is nearly indistinguishable from the EDFs for the CSIL and Condor data.

When viewed against the MLE exponential and MLE Pareto for each set the superiority of the previous fits is evident. Figures7, 8, and 9 plot the EDF, exponential, and Pareto CDFs for each of the data sets.

One of the most obvious discrepancies lies in the upper tails, which consistently appear too light or too heavy, for exponential and Pareto respectively, when compared to our sample EDFs.

In a modeling context, "tail behavior" can be important, especially if the presence or absence of rare occurrences must be modeled accurately. For example, previous research [28, 29] reveals Unix process lifetimes to be "heavy-tailed" and well-modeled by a Pareto distribution. Thus schedulers and process management systems must be designed for occasionally occurring processes that have very long execution times.

According to Figures7, 8, and 9, however, a Pareto distribution would over-estimate the probability of very long-lived resources by a considerable amount. Indeed, it may be that while Unix process lifetime distributions are heavy tailed, if they are executed in distributed or global computing environments, many of them will be terminated by resource failure since the resource lifetime distributions (both EDFs and their matching Weibull and hyperexponential fits) have considerably less tail weight.

Even beyond the differences in the tails, however, we can clearly see that the general shape of the exponential and Pareto distributions do not seem to fit the sample CDFs well.

Another well accepted method for graphically determining how observations fit a theoretical distribution is by generating Quantile-Quantile (Q-Q) Plots. In a Q-Q Plot, the ordinates from the EDF and the CDF from the theoretical distribution are plotted against one another.

If the observations were exactly drawn from our theorized distribution, the resulting Q-Q plot would be very nearly a line intersecting the origin, having slope 1. As Figures 11, 12, and 13 show, our observation quantiles are very close to being linearly related to the fitted Weibull quantiles for both the CSIL and Long-Muir-Golding data sets, but reveal a less-accurate quantile match for the Condor data set.

Note that we have *censored* the CSIL data to generate the Q-Q Plots by using a cutoff point after which we do not show quantile-quantile relationships. This method is outlined in [21] and has been employed so as to remove obfuscating data near the extreme upper tails of our sample distributions (we show data from quantiles 0.01 to approximately 0.97). Without censoring, we can see that data in the upper tail makes the resulting Q-Q Plot (shown in Figure 10) difficult to interpret as the entire trend of relationships is thrown off by a few unstable data points.

The reason for this censoring stems from the way in which each data set is gathered. In particular, resources that are available at the end of the measurement period generate truncated values. This distorted our data in two important ways. First, the duration of the measurement period artificially created a maximum uptime, when in fact there were several uptimes which continued over this entire period, so that the tail of the EDF was cut off; and second, uptimes which by chance started near the end of the period were assigned extremely short values, thus assigning undue weight to the left end of the distribution.

Figures 14, 15 and 16 show the Q-Q Plots of the dataset quantiles versus the hyperexponential model quantiles. The plots support the original CDF comparison graphs by showing that the relationships, even in the tails, between our empirical observations and fitted models are very near to linear.

To get a feel for the suitability of the Weibull and hyperexponential models versus the exponential or Pareto models, we present Q-Q Plots of our sample data against expo-

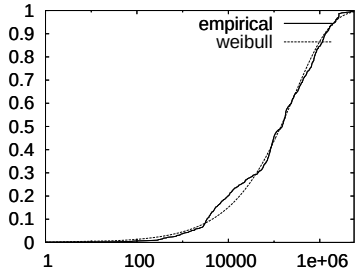


**Figure 1. CSIL data with Weibull fit**

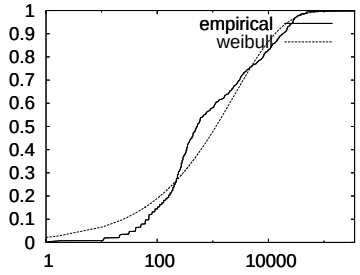


**Figure 2. Condor data with Wiebull fit**

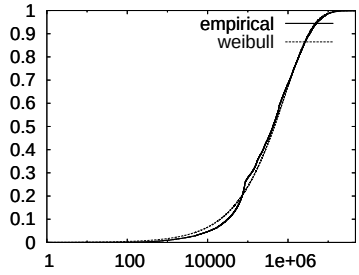


**Figure 3. Long data with Weibull fit**

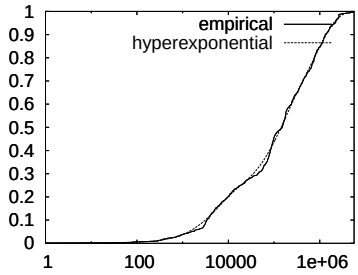


**Figure 4. CSIL data with hyperexponential fit**

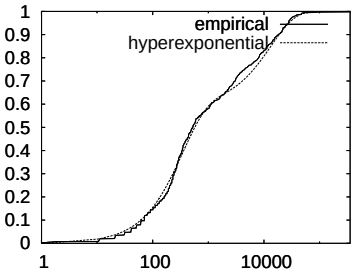


**Figure 5. Condor data with hyperexponential fit**

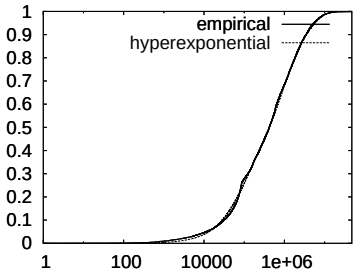


**Figure 6. Long data with hyperexponential fit**

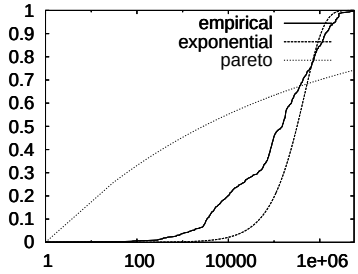


**Figure 7. CSIL data with exponential and Pareto fits**

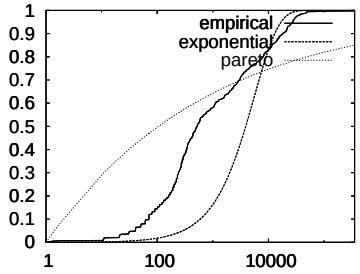


**Figure 8. Condor data with exponential and Pareto fits**

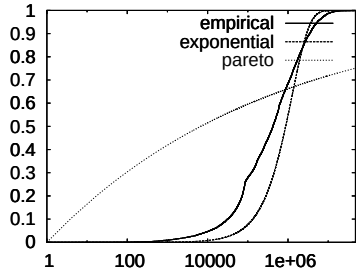


**Figure 9. Long data with exponential and Pareto fits**

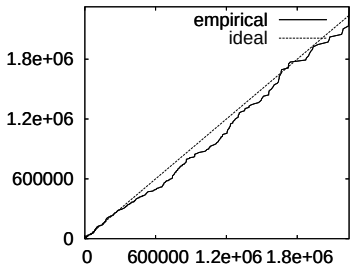


**Figure 11. CSIL Q-Q Plot of sample/Weibull quantile relationships**

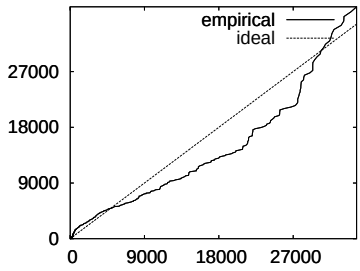


**Figure 12. Condor Q-Q Plot of sample/Weibull quantile relationships**

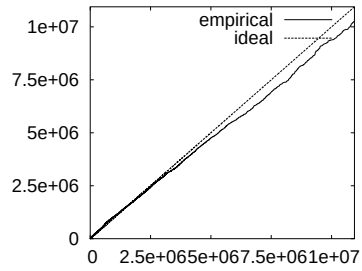


**Figure 13. Long Q-Q Plot of sample/Weibull quantile relationships**

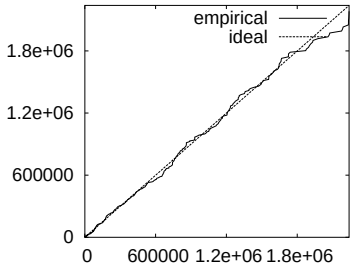


**Figure 14. CSIL Q-Q Plot of sample/hyperexponential quantile relationships**

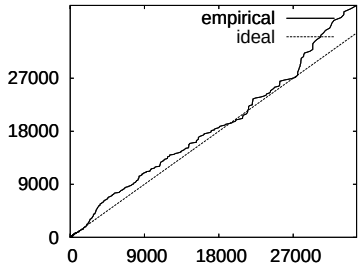


**Figure 15. Condor Q-Q Plot of sample/hyperexponential quantile relationships**

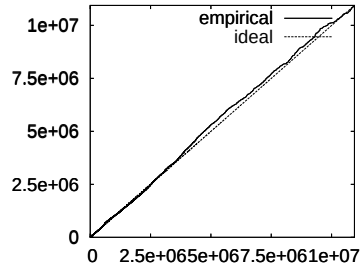


**Figure 16. Long Q-Q Plot of sample/hyperexponential quantile relationships**

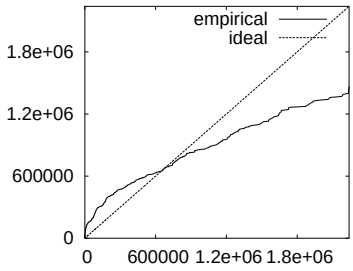


**Figure 17. CSIL Q-Q Plot of sample/exponential quantile relationships**

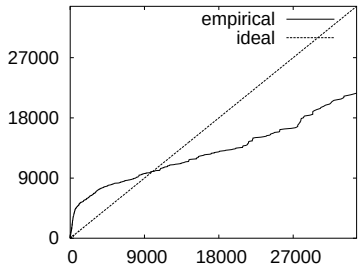


**Figure 18. Condor Q-Q Plot of sample/exponential quantile relationships**

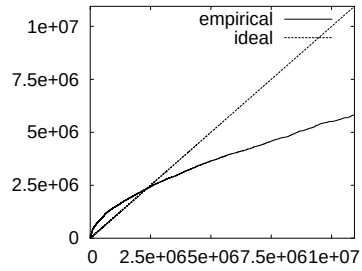


**Figure 19. Long Q-Q Plot of sample/exponential quantile relationships**

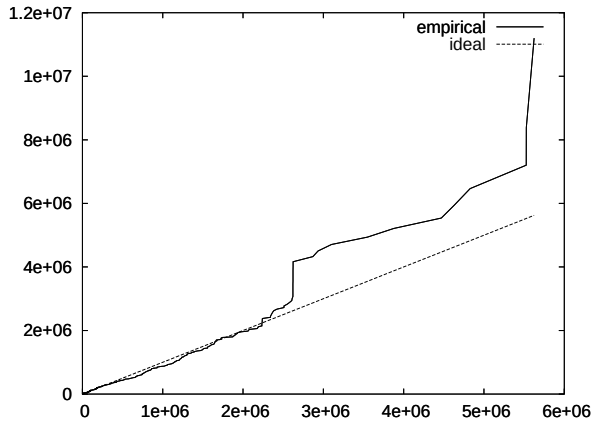


**Figure 10. CSIL Q-Q Plot of uncensored sample/Weibull quantile relationships**

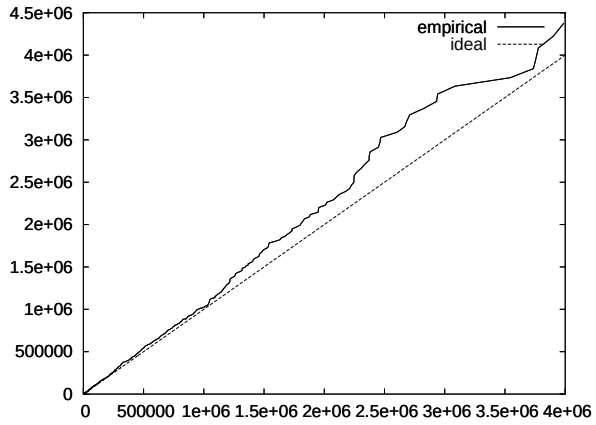


**Figure 20. Q-Q Plot showing ideal case of sample drawn directly from tested Weibull distribution**

nential model in Figures 17, 19 and 19. We do not show the Q-Q Plots for the Pareto fit as the quantile relationships are so far from the ideal linear graph that the plot is difficult to render. It suffices to say, however, a Pareto distribution in not indicated by any Q-Q Plot of our data sets.

Q-Q Plots are, themselves, sensitive to the shape of the tails in the distribution being plotted. This sensitivity is reflected by the sample size used (i.e. the smaller the sample size, the less accurately the tails may be rendered). To calibrate the sensitivity of Q-Q Plotting in our setting, we draw 1000 random values from the Weibull distribution we fit to the CSIL data (shown in Figure 1). We then generate the Q-Q Plot for this random sample against the actual Weibull from which it is drawn. This plot corresponds to the ideal case in that the observed sample comes from the distribution against which it is plotted. Figure 20 shows the results.

Even in the ideal case, the tails diverge because of the small number of data points that are likely to be drawn from the tail. Comparing this ideal Q-Q plot to the one shown in Figures 10 in which the actual data is used in uncensored form further supports case for a Weibull fit to the to the CSIL and Long, Muir, Golding data sets, but leaves the Condor data set open to suspicion. However, based on the visual comparison of the plots in Figure 12 and Figure 18, the MLE Weibull and EM-based hyperexponential seem clearly better choices than the exponential and Pareto.

In addition to comparing CDFs and empirical quantiles directly and Q-Q Plots, we also show probability-probability (P-P Plots) to illustrate distribution fit the entire range or probabilities. P-P Plots are generated by evaluating the EDF and MLE CDFs *at each sample point* and plotting them against each other.

If the sample points were exactly drawn from the theoretical distribution function, we would see a perfect linear (i.e. zero-intercept, slope 1) relationship between. Note that a P-P Plot differs from a Q-Q Plot in that it depicts the distributions only at the sample points, thereby eliminating any effects the necessary interpolation might introduce in a Q-Q Plot. That is, in an EDF, the quantile values that occur between sample data points must be interpolated. The P-P Plot does not require this interpolation because it only evaluates each function at the sample data points.

As Figures 21, 22, and 23 show, the Weibull/EDF relationships are almost linear throughout the probability range, especially for the CSIL and Long datasets. The hyperexponential/EDF relationships are also shown, in Figures 24, 25, and 26, to be quite close to linear.

## 4.2 Goodness-of-fit Tests

Before we present results for the Kolmogorov-Smirnov (KS) and Anderson-Darling (AD) test results, some discussion of goodness-of-fit (GOF) tests in general may be helpful.

Each of these tests is designed to test the hypothesis that our sample observations, $(x_1, x_2, ..., x_i)$, are drawn from random variables having some specified statistical distribution, by producing a measure of the strength of the evidence *against* this hypothesis provided by the data.

That is, we wish to examine the question whether

$$F_e(x) = F_d(x)$$

where $F_e(x)$ is the EDF of our sample data and $F_d(x)$ is a hypothesized and completely specified distribution model (e.g. MLE Weibull, exponential, or Pareto in this study). Such tests are set up with null hypothesis

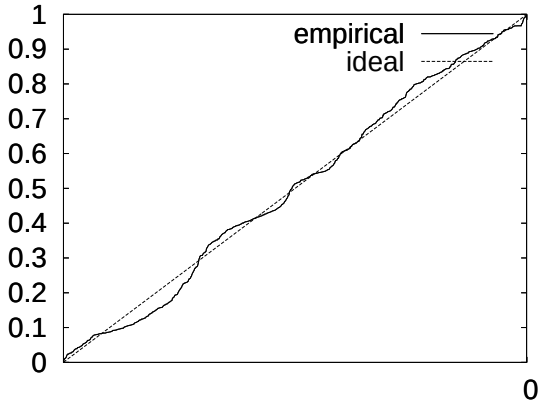


**Figure 21. CSIL P-P Plot of sample/Weibull percentile relationships**

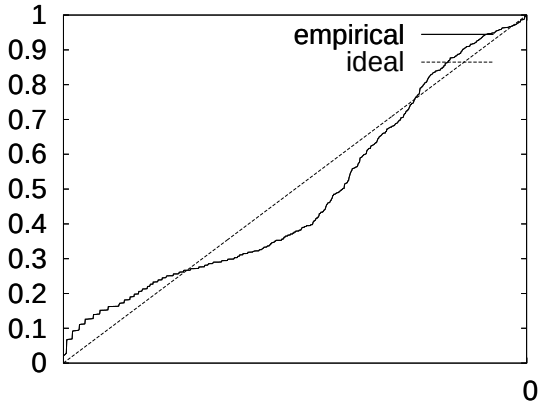


**Figure 22. Condor P-P Plot of sample/Weibull percentile relationships**

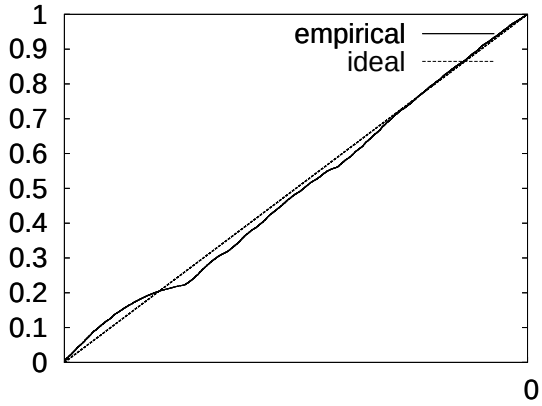


**Figure 23. Long P-P Plot of sample/Weibull percentile relationships**

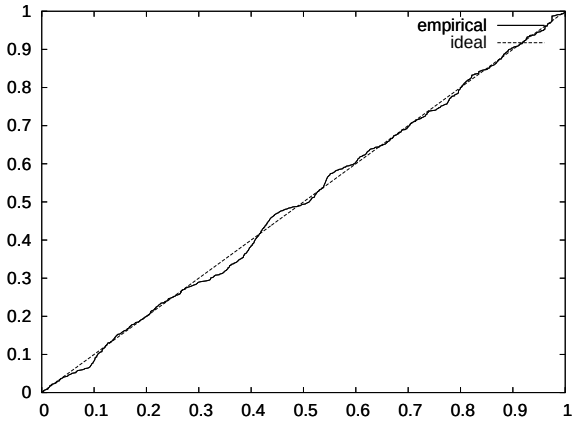


**Figure 24. CSIL P-P Plot of sample/hyperexponential percentile relationships**

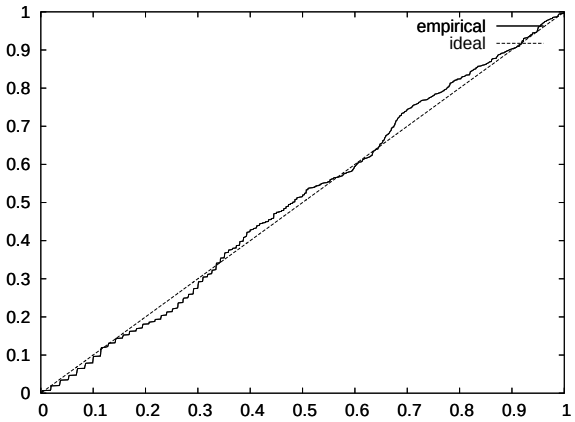


**Figure 25. Condor P-P Plot of sample/hyperexponential percentile relationships**

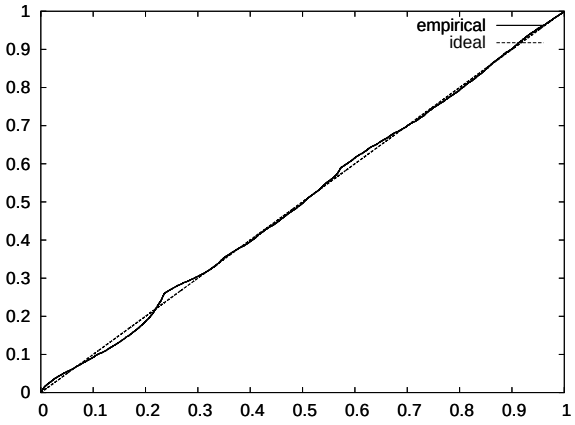


**Figure 26. Long P-P Plot of sample/hyperexponential percentile relationships**

$H_0$ : *the data come from the given distribution*

and alternative hypothesis

$H_a$ : *the data do not come from the given distribution*

Goodness of fit tests have been designed to test against this null hypothesis and essentially allow us to ask whether, at a chosen significance level $\alpha$ (often but rather arbitrarily set at $0.05$ or $0.01$), we *reject* $H_0$ at level $\alpha$. Each test generates its own test statistic, which we can use to compute a *p-value*, which essentially measures how common or scarce such a test statistic would be under the assumption that the null hypothesis is true. After running a GOF test against some observed data, we could end up with $p = 0.07$. This would mean that the probability of getting the same or larger test statistic from the test using a random sample drawn directly from the tested distribution is $0.07$. Thus, we have failed to reject $H_0$ at the significance level $\alpha = 0.05$. If $p$ were less than our chosen significance cutoff point, then we would reject $H_0$ in favor of $H_a$.

Although we realize the failure to reject $H_0$ is in no way the same as accepting $H_0$, we use GOF tests in this study to lend further evidence that some distributions model our data better than some other distributions. We have found that GOF tests will, almost certainly, reject $H_0$ for large data sets, since the power of the tests to reveal discrepancies between the observed data and the test distribution increases with increasing sample size. Thus we find GOF tests somewhat difficult to use for testing large pools of real-world data, which by nature cannot be expected to follow precisely any theoretical distribution function.

For this reason, we employ a novel application of GOF testing to the data sets in this study, keeping in mind that we are presenting these results as further comparative evidence of how well the various tested distributions fit the observed data. Rather than conducting each test using all available measurements (which indicates rejection in all cases), we GOF tests on randomly gathered subsamples to determine if rejection is warranted for small numbers of data points. By doing so, we exploit the ability of each GOF test to determine, from a smattering of measurements, whether rejection is indicated.

### 4.3 Goodness-of-fit Analysis

For this analysis we use both KS and AD goodness-of-fit tests with randomly chosen subsamples from our data sets each having size 100. We then repeat the tests, with different random subsamples, 1000 times to get a range of test results and then we use the average test statistic value

to compute the p-value. Rejection at size 100 indicates that with as few as 100 data points it is evident that the tested distribution is inappropriate. The addition of more data points to the test will only confirm this inappropriateness further.

Test results are shown in Table 2 which are the average p-values from the 1000 iterations of the test.

From the table, it is clear that both the exponential and Pareto fail the goodness-of-fit tests for all three data sets. This is not entirely surprising, since the visual fit was clearly inferior for all three data sets. The hyperexponential is performing significantly better than all of the models for all of the data sets, which was expected to eventually happen as the number of phases was arbitrarily increased. For the Weibull, we fail to reject the null hypothesis at $\alpha = 0.05$ significance level using the KS test for all three data sets. We fail to reject the null hypothesis at $\alpha = 0.05$ significance using the AD test for the CSIL and Long data sets, but reject for the Condor data set, supporting the graphical evidence that the Condor data set is less-well modeled by a Weibull than the CSIL or Long-Muir-Golding data. Despite the rejection, however, the graphical comparison indicates that the Weibull is substantially better than either the exponential or Pareto at modeling the observed data, but somewhat worse than a hyperexponential.

### 4.4 Data Characteristics

When we start using a Weibull distribution to model the data sets presented in this work, we would like to fully understand and be able to characterize many aspects of the sample data we're working with. We need to know how our data can be characterized in terms of stationarity and identical distribution. To kick-start this understanding, we have performed some common statistical tests that begin to help us understand these underlying features that may influence further work with the data. For many of these tests, we are exploiting the fact that we have access to our data sets as they were originally recorded; grouped by machine. Each of our datasets can be expressed as a combined list of availability measurements, or as many sets of availability uptimes grouped according to the machine from which were taken.

We have, based on intuition about the underlying causes for machine reboots and Condor process evacuations, assumed for the length of this paper that our data values are independent. We believe that, for instance, one uptime interval on some machine has no effect on the length of the next uptime interval. We also assume that the data do not follow any particular *trend*; in other words, the uptimes should not be getting generally longer or shorter as time progresses. The combined properties of independence and lack of trend constitute what we will call "stationarity."

To support our assumption that our data are stationary, we have performed on each machine a *runs test*, which is designed to detect whether the sequential data within a time-ordered set appeared "randomly" or whether there was a trend or autocorrelation within the data. The idea of this test is that a trend or positive autocorrelation within a single machine's ordered set of data will tend to create longer and fewer "runs" of consecutive data points above, and also below, the median than would be expected if the data were truly random, and a negative autocorrelation would cause there to be a larger number of such runs due to the tendency of the data to oscillate. Runs test data for the CSIL and Condor data sets were remarkably close to the null distribution. While the Long data set contained significantly more machines (31) to be rejected at the .05 level than would be expected under the null hypothesis (12.75), the number of machines that failed the test because of too few runs was almost identical to the number that failed because of too many (16 and 15 respectively), so these factors may tend to mitigate and contribute to an overall data set that is also fairly stationary.

We would also like to know whether the distributions of uptimes for the various machines are all identical. There are no good non-parametric tests available to test for identical distribution against the most general alternative hypothesis, but we are able to perform the Kruskal-Wallis test for identical location. This test is designed to detect whether a data point's rank in a combined set is affected by its group membership. The testing process showed that we consistently and strongly *rejected* the null hypothesis that our data are identically distributed. This is not entirely surprising, since the data-gathering methods for all three data sets made no attempt to record machines based on some assumed similarity with regards to availability. The importance of this result lies in the possible uses of the fitted models to individual machines. Although we can determine good models for combined sets of machines we cannot, in good conscience, apply this model to for any individual machine. That is, these models capture the availability distributions that are available to a user who wishes to introduce a single process for execution on a randomly chosen machine. The collection of machines with respect to this process is well-modeled, but the availability of an individual machine is not. As we record more data per machine, and are able to perform model fitting methods on the individual machine, we will be more able to address machine-specific availability problems directly.

## 5 Discussion

From the results presented in this paper, we make several observations. First, none of the data are well-modeled by either an exponential or a Pareto distribution. The visual evidence indicates that the fits are poor, and the GOF anal-

| Data Set | Weibull | | Exponential | | Pareto | | Hyperexponential | |
|---|---|---|---|---|---|---|---|---|
| | AD | KS | AD | KS | AD | KS | AD | KS |
| CSIL | 0.071 | 0.36 | 0 | 0.0002 | 0 | 0.0005 | 0.59280 | 0.47 |
| Condor | 0 | 0.07 | 0 | 0 | 0 | 0 | 0.68291 | 0.42 |
| Long | 0.132 | 0.41 | 0 | 0.001 | 0 | 0.0005 | 0.77247 | 0.48 |

**Table 2. Table of p-value results from GOF tests**

ysis seems to confirm that even at a coarse subsampling, the observed data is likely not to have come from either the MLE exponential or Pareto we fit. Thus, the tails of the availability distributions are neither as light as would be described by an exponential, nor as heavy as described by a Pareto. While perhaps not surprising, this conclusion may have wide-ranging effects.

In [46], Plank and Elwasif examine the "cost" of using an exponential model as the basis for optimal checkpoint interval determination. The authors reproduce Long, Muir and Golding's analysis [35] and find that an exponential model is unlikely to be accurate. However, they go on to determine that using an exponential model produces acceptable if conservative results for parallel systems. In a global computing environment, however, where the number of active processes in a program scales to much greater levels, and checkpoints are sure to traverse over-taxed network links, the need to control checkpoint-induced load becomes critical. Indeed, the exponential model under-estimates lifetime, particularly in the tail, causing checkpoints to be taken with greater frequency than necessary. In wide-area network environment, the cost of checkpointing too frequently is both lost compute time and lost network throughput if the checkpoints are sent to a remote site for storage.

Process scheduling is similarly affected. Downey and Harchol-Balter [29] make a convincing case for using Pareto distributions to model Unix process lifetimes. Based on non-negligible probability that a randomly chosen process will be long lived, they argue that the overhead of process migration can be amortized by the benefit of load balancing. However, if the machines under consideration are part of a federated system, as are the machines in our study, processes may substantially outlive machine availability periods (or periods between resource reclamations, in the case of Condor). Under these conditions, the value of checkpointing and migration is easily amortized since the alternative is either to restart long-running processes that are terminated by a failure, or to abandon them altogether.

More theoretically, much of the recent work in peer-to-peer systems [53, 62, 63] assumes exponential lifetime models. From an equational tractability standpoint, these simplifying assumptions are attractive, and simulation using these models indicates that their use does not pose a significant risk to stability or performance. We contend, however, that the effect we observe should be considered in any future peer-to-peer system formulation. The non-memoryless aspect of the Weibull and hyperexponential distributions implies that past history carries important information about future availability. To our knowledge, no peer-to-peer system accounts for this possibility despite the recent widespread interest in peer-to-peer systems design.

In terms of building credible Grid, desktop, and global computing simulations, these results clearly indicate that both Weibull and hyperexponential models of resource lifetime must be considered. Enterprise-wide systems like those supported by Condor, Entropia [23] and United Devices [57] must certainly consider parameterizations of these models possible operating regimes. Moreover, Grid testbeds such as the Grid Application Development Software [7, 27] testbed often include desktop resources like the ones we study in the CSIL and Condor data sets. Grid Simulation packages such as the MicroGrid [51, 16], SimGrid [11], Bricks [54] and Gridsim [10] can more accurately capture federated resource behavior using these model types.

The question of whether to use a Weibull or hyperexponential model in simulation and/or analysis is a difficult one one to address quantitatively. Using reasonably powerful goodness-of-fit tests, the p-values for a hyperexponential that has been fit with an EM procedure are generally larger than those generated from an MLE Weibull. While not a rigorous comparison, comparing p-values in this way does indicate which of the two models is most probably the more accurate. In addition, a hyperexponential has some attractive analytical properties that make it a useful choice in queuing contexts, when analytical solutions are needed [24]. For simulation, however, Weibull models offer potential advantages. Having only two parameters, "sweeps" of the simulation parameter space are substantially less complex than with a hyperexponential. Recall that a $k$ phase hyperexponential requires $2k - 1$ parameters, and that the value of $k$ must be chosen *a priori*. In our study, $k = 3$ is complex enough to capture the distributional behavior, but higher phase degrees may be necessary. Moreover, the Weibull is a algebraically invertible model making visual techniques such as Q-Q plots less complex to generate.

Regarding the general process of model fitting to empir-

ical performance data, Feldmann and Witt provide evidence that any heavy-tailed distribution can be approximated by a hyperexponential distribution with a sufficient number of phases [24]. However, their preferred method is to use an MLE technique to determine a heavy-tailed model (either Pareto or Weibull in their paper) and then to use an EM technique to approximate the model with a hyperexponential. In our experience, fitting a hyperexponential directly to the data using EM provides a better fit, using fewer phases than they report.

The applicability of both Weibull and hyperexponential models is startling clear in the analysis of the Long, Muir and Golding data set. This study gives some indication of what availability may be like in a web-services based computing environment. We suspect that with the current widespread of proliferation of computer viruses, an attempt to reproduce the Internet survey for the current Internet will not be successful. However, given the similar nature of the Weibull fits across data sets, we conjecture that the effect extends at least to Internet connected hosts and servers. While the reliability of the network fabric has certainly improved since the survey was conducted, we conjecture that machine availability is unlikely to have undergone a similar improvement. For example, in 1995 the predominant server operating system is likely to have been some variant of the Solaris operating system from Sun Microsystems. Since then, many if not the majority of web systems are supported on machines running some variant of Linux. There is little reason to suspect that this shift has produced an increase in availability. However, the increased virus and "spam" activity combined with the greater proliferation of even less stable operating system platforms, may have decreased availability leaving the appropriateness of even exponential models (with lighter tails) as an open question for the web services community.

Finally, we have attempted to characterize our data sets in terms of stationarity and identical distribution by consulting common tests. We failed to reject a null hypothesis for stationarity, but very strongly rejected a null hypothesis for identical distribution. However, during these experiments, we noticed an interesting feature in some of our data sets, namely that when four outlier machines (those with extremely large contributions to the Kruskal-Wallis test statistic) were removed from the Condor data set, the Kruskal Wallis test for i.d. jumped from an essentially 0 p-value to a p-value of 0.2. As can be seen in Figure 27, the removal of outlier machines did not affect the suitability of a new MLE Weibull fit to the remaining data. From this observation, we suspect that our data may have very discrete subsets of machines that, as a group, may benefit from a model separate from the model for the combination of all machines. We intend to explore this idea further in the future in the hopes of finding large enough i.d. groups of machines such that predictions made from group models can be used to accurately predict behavior of any one group member.

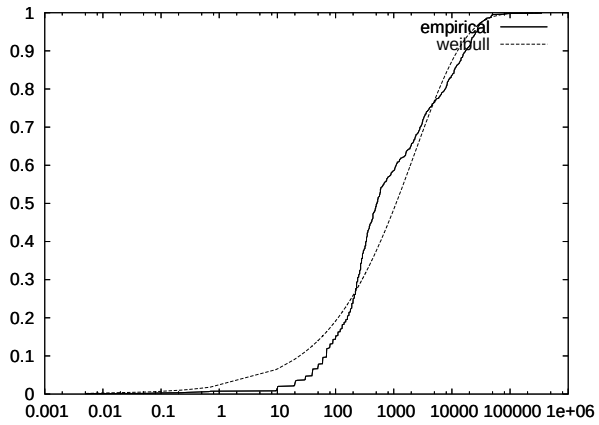


**Figure 27. Condor data with with removed outliers and MLE Weibull fit ($\alpha = 0.493, \beta = 2304.8$)**

Automatically determining this clustering of availability is the subject of our on-going research. However, as collections of resources, the combination of MLE Weibull and EM hyperexponential fitting provides good models automatically from NWS-generated availability data. Given the evidence for stationarity, we can gather data and use both techniques to fit models that are applicable for long periods. By doing so, our method provides a dynamic (if slowly changing) characterization of resource pools that is useful in several distributed computing contexts.

## 6 Conclusions

The need to model resource availability and to characterize groups of resources in terms of their availability is critical to desktop Grid, peer-to-peer, and global computing paradigms. Previous related work has used exponential (memoryless) or Pareto distributions, but our work shows that Weibull and hyperexponential distributions are more accurate choices. Visual evidence and GOF results (when applied repeatedly to subsamples) show with a high degree of statistical significance that the data we have is distributed according to Weibull and hyperexponential distributions. The choice of which to use depends on the application for which the model is needed, and from an engineering perspective, they can be made equivalent. Both, however, are significantly better at capturing the distribution of availability time we analyze in this study and both can be computed automatically from on-line NWS data.

We have also examined the stationarity and independence characteristics of the data sets we consider. The

models we fit appear stationary, but independence among the machines in each set is not generally indicated. Thus, the techniques we have developed characterize collections of machines, but not the machines themselves. However, censoring the data (a common technique used to decrease the impact of outliers) reveals that for some data sets, the machines are "almost" independent in that the removal of a few outliers dramatically improves independence-test results. From these results, we hope to generate individual resource models and to improve the quality of simulation and modeling for volatile distributed systems.

## Acknowledgments

We gratefully acknowledge the contributions to this work that various members of the research community have made, and hopefully will continue to make. Dr. James Plank, in the Computer Science Department at the University of Tennessee, has provided us with invaluable insights with regards to the importance of modeling to the fault-tolerance community. Dr. Darrell Long, in the Computer Science Department at the University of Santa Cruz, generously allowed us to use his original study (the data for which Dr. Plank provided) as the basis for our Internet investigation and as a control. Dr. Miron Livny, in the Computer Science Department at the University of Wisconsin, provided almost limitless access to the Condor pool at Wisconsin. Finally, Dr. Allen Downey, at the Olin College of Engineering, provided a whole raft of useful suggestions and analysis.

## References

[1] D. Abramson, J. Giddy, I. Foster, and L. Kotler. High Performance Parametric Modeling with Nimrod/G: Killer Application for Global Grid? In *The 14th International Parallel and Distributed Processing Symposium (IPDPS 2000)*, 2000.

[2] 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. `http://www.globus.org/research/papers/sc01ewa_esg_chervenak_final.pdf`.

[3] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics*, 23:419–441, 1996.

[4] S. Atchley, S. Soltesz, J. Plank, and M. Beck. Video ibpster. *Future Generation Computing Systems*, 19:861–870, 2003.

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

[6] C. E. Beldica, H. H. Hilton, and R. L. Hinrichsen. Viscoelastic beam damping and piezoelectric control of deformations, probabalistic failures and survival times.

[7] 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.

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

[9] 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.

[10] R. Buyya and M. Murshed. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Comcurrency Practice and Experience*, 14(14-15), Nov-Dec 2002.

[11] H. Casanova. Simgrid: A toolkit for the simulation of application scheduling. In *Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, 2001.

[12] 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.

[13] H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In *Proceedings of SuperComputing 2000 (SC'00)*, Nov. 2000.

[14] 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.

[15] R. Chandramouli, N. Vijaykrishnan, and N. Ranganathan. Sprt for weibull distributed integrated circuit failures.

[16] A. Chien. Microgrid: Simulation tools for computational grid research. `http://www-csag.ucsd.edu/projects/grid/microgrid.html`.

[17] W. Chrabakh and R. Wolski. GrADSAT: A Parallel SAT Solver for the Grid. In *Proceedings of IEEE SC03*, November 2003.

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

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

[20] M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5, December 1997.

[21] R. B. D'Agostino and M. A. Stephens. *Goodness-Of-Fit Techniques*. Marcel Dekker Inc., 1986.

[22] Empht home page. Available on the World-Wide-Web. `http://www.maths.lth.se/matstat/staff/asmus/pspapers.html`.

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

[24] A. Feldmann and W. Witt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 21(8):963–976, August 1998.

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

[26] A. L. Goel. Software reliability models: Assumptions, limitations, and applicability. In *IEEE Trans. Software Engineering, vol SE-11, pp 1411-1423*, Dec 1985.

[27] GrADS. `http://hipersoft.cs.rice.edu/grads`.

[28] M. Harchol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. In *Proceedings of the 1996 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, 1996.

[29] M. Harcol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, 1997.

[30] R. K. Iyer and D. J. Rossetti. Effect of system workload on operating system reliabilty: A study on ibm 3081. In *IEEE Trans. Software Engineering, vol SE-11, pp 1438-1448*, Dec 1985.

[31] J. Kephart and D. Chess. The vision of autonomic computing. *IEEE Computer*, January 2003.

[32] J.-C. Laprie. Dependability evaluation of software systems in operation. In *IEEE Trans. Software Engineering, vol SE-10, pp 701-714*, Nov 1984.

[33] I. Lee, D. Tang, R. K. Iyer, and M. C. Hsueh. Measurement-based evaluation of operating system fault tolerance. In *IEEE Trans. on Reliability, Volume 42, Issue 2, pp 238-249*, June 1993.

[34] W. Leland and T. Ott. Load-balancing heuristics and process behavior. In *Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS '86)*, pages 54–69, May 1986.

[35] 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.

[36] D. D. E. Long, J. L. Carroll, and C. J. Park. A study of the reliability of internet sites. In *Proceedings of the 10th IEEE Symposium on Reliable Distributed Systems (SRDS91)*, 1991.

[37] Mathematica by Wolfram Research. `http://www.wolfram.com`.

[38] S. M.H., R. M.L., B. M.C., S. P., , and B. S. Mechanical properties of fabrics woven from yarns produced by different spinning technologies: Yarn failure in fabric.

[39] M. Mutka and M. Livny. Profiling workstations' available capacity for remote execution. In *Proceedings of Performance '87: Computer Performance Modelling, Measurement, and Evaluation, 12th IFIP WG 7.3 International Symposium*, December 1987.

[40] H. Nakada, H. Takagi, S. Matsuoka, U. Nagashima, M. Sato, and S. Sekiguchi. Utilizing the metaserver architecture in the ninf global computing system. In *High-Performance Computing and Networking '98, LNCS 1401*, pages 607–616, 1998.

[41] GNU Octave Home Page. `http://www.octave.org`.

[42] V. Paxon and S. Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3), 1995.

[43] V. Paxon and S. Floyd. Why we don't know how to simulate the internet. In *Proceedings of the Winder Communication Conference also* `citeseer.nj.nec.com/paxon97why.html`, December 1997.

[44] 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.

[45] D. Pettit and A. Turnbull. General aviation aircraft reliability study.

[46] 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.

[47] 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.

[48] 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.

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

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

[51] H. Song, J. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K. Taura, and A. Chien. The MicroGrid: a Scientific Tool for Modeling Computational Grids. In *Proceedings of SuperComputing 2000 (SC'00)*, Nov. 2000.

[52] N. Spring and R. Wolski. Application level scheduling: Gene sequence library comparison. In *Proceedings of ACM International Conference on Supercomputing 1998*, July 1998.

[53] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and K. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *In Proc. SIGCOMM (2001)*, 2001.

[54] A. Takefusa, O. Tatebe, S. Matsuoka, and Y. Morita. Performance analysis of scheduling and replication algorithms on grid datafarm architecture for high-energy physics applications. In *Proceedings 12th IEEE Symp. on High Performance Distributed Computing*, June 2003.

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

[56] E. Tufte. *The Visual Display of Quantitative Information, 2nd Ed.* Graphics Press, May 2001.

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

[58] N. Vaidya. On checkpoint latency. In *Proceedings of Pacific Rim Symposium on Fault-tolerant Systems*, December 1995.

[59] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. In *SIGCOMM'95 Conference on Communication Architectures, Protocols, and Applications*, pages 110–113, 1995.

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

[61] 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.

[62] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz. A resilient global-scale overlay for service deployment. *(to appear) IEEE Journal on Selected Areas in Communications*.

[63] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, U.C. Berkeley Computer Science Department, April 2001.