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

Daniel Nurmi, John Brevik, and 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.

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

1 Introduction

As performance-oriented distributed computing (often heralded under the moniker "Computational Grid" computing [13]) becomes more prevalent, the need to characterize accurately resource reliability emerges as a critical problem. Today's successful Grid applications uniformly rely on run-time scheduling [1, 6, 9, 10, 27, 29] 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 in order to employ nationally or globally distributed resource pools (e.g. in the way SETI@Home [30] does) or enterprise-wide desktop resources (as many commercial endeavors do [3, 12, 36]), performance-oriented distributed applications must be able either to avoid or to tolerate resource failures.

Designing the next generation of Grid applications requires an accurate model of resource failure behavior. There has been a great deal of work [14, 18, 21, 20, 25] on the problem of modeling resource failure (or, equivalently, resource availability) statistically. More recently, peer-to-peer systems have used statistical distributions as the basis of their availability assumptions [32, 39]. As Plank and Elwasif point out in their landmark paper [28], however, most of these approaches assume that the underlying statistical behavior can be described by some form of exponential distribution or hyperexponential distribution [21]. In addition, they go on to note that, despite the popularity of these models, they often fail to reflect empirical observation of machine availability. In other contexts, such as process lifetime estimation [16] and network performance [22, 26, 37], researchers often advocate the use of "heavy-tailed" distributions, especially the Pareto. Other work has been done showing that a Weibull distribution is an appropriate model for various resource availability data [38, 17], but this work lacks a detailed analysis of model fitting and verification.

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 [5] system, NetSolve [9], and APST [6], in order to enable effective resource allocation and scheduling. Commercial-enterprise computing systems such as Entropia [12], United Devices [36], and Avaki [3] will also be able to take advantage of automatically determined models as they tune themselves to the characteristics of a particular site. We believe this work will be particularly important to the development of credible and effective Grid and Autonomic Computing [19] simulations. Because Grid architectures are driven by the dynamic resource sharing of competing users, repeatable "en vivo" experiments are difficult or impossible. Several effective emulation [31] and simulation [7, 8, 33] systems have been developed for Grid environments. These systems will benefit immediately from the more accurate models our method produces.

We propose an approach to modeling machine availability based on fitting statistical distributions to observed data, which is outlined in the following manner. In Section 2 we define the statistical distributions used throughout this work and describe our method for estimating the necessary parameters from a given set of availability measurements. We also outline the three data sets used in this study in Section 2. 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 in Section 3, in which we compare the generated models for all three data sets both visually and through the use of two Goodness of Fit (GOF) tests to complement our visual analysis. In Section 4 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 [4] and also to model computer resource availability distributions [38, 17]. Hyperexponentials have been used to model machine availability previously [25] especially when observed data requires a model which can approximate a wide variety of shapes. In order to fit most of the statistical distributions used in this paper to observed data, we implemented Matlab [24] scripts which found the MLE (Maximum Likelihood Estimation) parameters. The problem of finding MLE parameters for the hyperexponential, however, tends to be numerically intractable for large data sets, so instead we use EMpht software [2]. Following are the equations for the models we compare in this work, along with a description of how we estimate the model parameters given a sample data set.

2.1 Statistical Distributions

Throughout this paper, we will use small f for density functions and capital F for distribution functions, subscripted to differentiate among the various types of distribution. These functions, f_W and F_W respectively, for a Weibull distribution are given by

$$f_W(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^\alpha} \tag{1}$$

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

The parameter α is called the *shape* parameter, and β is called the *scale* parameter [1]. Note that the Weibull distribution reduces to an exponential distribution when $\alpha = 1$.

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

$$f_H(x) = \sum_{i=1}^{k}[p_i \cdot f_{E_i}(x)] \tag{3}$$

where

$$f_{E_i}(x) = \lambda_i e^{-\lambda_i x} \tag{4}$$

defines the density function for an exponential having parameter λ_i. In the definition of $f_H(x)$, all $\lambda_i \neq \lambda_j$ for $i \neq j$, and $\sum_{i=1}^{k} p_i = 1$. The distribution function is defined as

$$F_H(x) = 1 - \sum_{i=1}^{k} p_i \cdot e^{-\lambda_i x} \tag{5}$$

for the same definition of $f_{e_i}(x)$.

The probability density and distribution functions for the exponential and Pareto distributions, respectively, are as follows:

$$f_E(x) = \lambda e^{-\lambda x} \tag{6}$$

$$F_E(x) = 1 - e^{-\lambda x} \tag{7}$$

[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.

$$f_P(x) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}} \quad (8)$$

$$F_P(x) = 1 - \left(\frac{\beta}{x}\right)^\alpha \quad (9)$$

2.2 Data Sets

In this work, we use three data sets which we believe exhibit availability behavior typical of hosts currently residing on the Internet. The first data set is from UCSB's CSIL computer science student lab. Each measurement records the time from when a workstation is able to run a user process to when it no longer can do so. The second data set is drawn from the Condor [34] pool running at the University of Wisconsin. Each condor availability measurement is the time from when the Condor scheduler starts one of our monitoring processes to when the allocated workstation evicts our monitor process. Finally, we have obtained the dataset from a work by Long, Muir, and Golding [23] in which they remotely measured Internet host availability. We measured 83 machines in the CSIL lab for 8 weeks. The Condor dataset comes from 210 during a 6-week period, and Long, Muir and Golding gathered data from 1170 machines over a 3 month experimental period.

Before attempting to capture the distribution behavior of our data sets, we wanted to explore the data characteristcs of independence and identical distribution. We assume data independence since we intuitively believe, for instance, that one uptime interval on some machine has no effect on the length of the next uptime interval. To inspect the identical distribution characteristics of the data, we performed a Kruskal-Wallis test for identical location. The test strongly *rejected* the null hypothesis that the data is i.d. This is not, however, entirely surprising as the machines we are monitoring have a wide range of usage models which impact their availability. This does imply, however, that although we can use models which fit combined machine availability data, we cannot infer from these models any information about the individual machines that make up the combined data set.

3 Analysis

Data Set	Weibull		Hyperexponential						Exp.	Pareto	
	α	β	p_1	p_2	p_3	λ_1	λ_2	λ_3	λ	α	β
CSIL	.545	275599	.464	.197	.389	$1 * 10^{-6}$	$2 * 10^{-4}$	$8 * 10^{-6}$	$2 * 10^{6}$	.087	1
Condor	.49	2403	.592	.408	NA	$3 * 10^{-3}$	$7 * 10^{-5}$	NA	.00018	.149	1.005
Long	.61	834571	.282	.271	.474	$3 * 10^{-7}$	$1 * 10^{-5}$	$1 * 10^{-6}$	$7 * 10^{7}$	.079	1

Table 1. Table of fitted model parameters

The goal of our study is to determine the value of using Weibull and hyperexponential distributions to model resource availability. Our method 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-determined 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 the Weibull and hyperexponential models more accurately fit each data set.

In each case, we use three different techniques to evaluate model fit: graphical; the Kolmogorov-Smirnov [11] (KS) test; and the Anderson-Darling [11] (AD) test. Graphical evaluation is often the most compelling method [35] 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. [2]

3.1 Graphical Analysis of The Availability Measurements

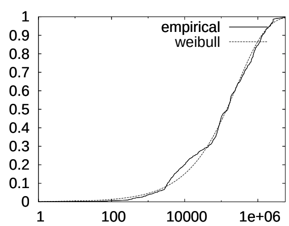

Fig. 1. CSIL data with Weibull fit

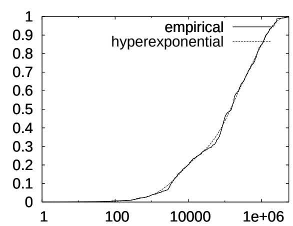

Fig. 2. CSIL data with hyperexponential fit

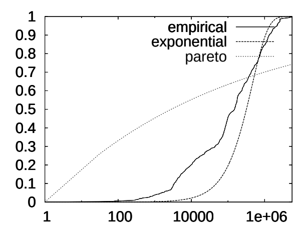

Fig. 3. CSIL data with exponential and Pareto fits

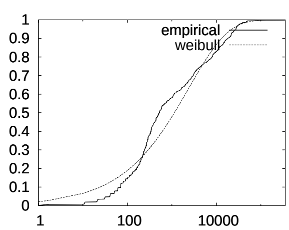

Fig. 4. Condor data with Wiebull fit

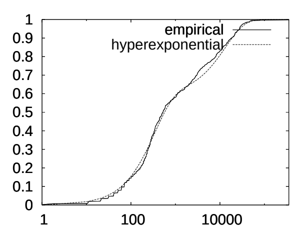

Fig. 5. Condor data with hyperexponential fit

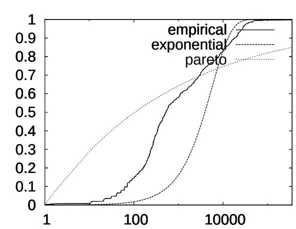

Fig. 6. Condor data with exponential and Pareto fits

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

[2] The best known goodness-of-fit test is the Chi-squared test. Both the Kolmogorov-Smirnov and the Anderson-Darling tests are considered more appropriate for continuous distributions than the Chi-squared test, which is designed for categorical data and would thus require artifical "binning" of data. We therefore use these methods in place of the more familiar one.

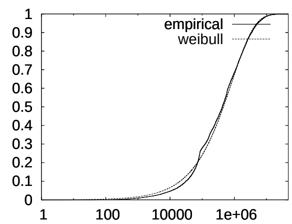

Fig. 7. Long data with Weibull fit

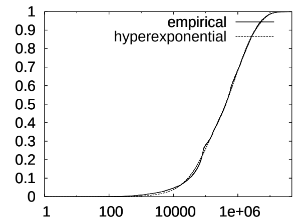

Fig. 8. Long data with hyperexponential fit

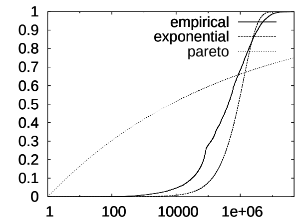

Fig. 9. Long data with exponential and Pareto fits

empirical cumulative distribution for the data set. The form of the CDF for the Weibull, hyperexponential, exponential and Pareto are given by equations 2, 5, 7, and 9 respectively (*cf.* 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

$$F_e(x) = j/n, X_j \leq x < X_{(j+1)} \tag{10}$$

We start by comparing the empirical observations from the CSIL data set (as an EDF) to the CDF determined by the EMpht-estimated hyperexponential, and the MLE-estimated Weibull, exponential, and Pareto distributions (shown in Figures 1, 2, and 3). In all of the figures depicting distributions in this paper, the units associated with the x-axis are seconds of machine availability. We use a log scale for the x-axis to better expose the nature of each fit. Both the hyperexponential and the Weibull fit the data substantially better than either an exponential or Pareto; the hyperexponential is also able to capture the slight inflection around 10,000 seconds. Since automatic selection of the number of phases to use when fitting a hyperexponential is not part of the EMpht software, we have devised our own method. To determine the number of phases, we begin with a 2-phase hyperexponential, test the resulting fit with a Kolmogorov-Smirnov test, and then repeat with an increased number of phases until the KS test result shows no improvement. In this case, for the CSIL data, the algorithm terminated using three phases.

For the Condor data set, the comparison (shown in Figures 4, 5, and 6) is more striking. Again, the hyperexponential (a 2-phase, in this case) appears to fit the shape of the curve most closely, and the Weibull appears a better choice than either exponential or Pareto. Note in particular how again the hyperexponential is able to capture the inflection points of the Condor EDF around 1000 seconds, while the Weibull is unable to do so.

Finally, the fits (3-phase hyperexponential in this case) for the Long, Muir, and Golding data are shown in Figures 7, 8, and 9.

The comparison is similar to that for the CSIL data. The multi-phase hyperexponential fits slightly better than a Weibull, and both are substantially better than an exponential or Pareto.

Of particular interest are the way in which each hypthetical distribution appears to match the tail of an EDF. In many application contexts, "tail behavior" can be important, especially if the presence or absence of rare occurrences must be modeled accurately. For example, previous research [15, 16] 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 infrequently occurring processes that have very long execution times.

According to Figures 3, 6, and 9, however, a Pareto distribution would overestimate 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.

3.2 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.

Table 2 shows the GOF test results which are the average *p-values* from the 1000 iterations of the experiment.

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

From the table, it is clear that both the exponential and Pareto perform poorly on these 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 performs substantially better than all of the other models for all of the data sets; again this is not surprising since the hyperexponential model can increase its number of phases as needed. For the Weibull, we fail to reject the null hypothesis at $\alpha = 0.05$ significance level on average for subsamples of size 100 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. Although GOF tests cannot provide a positive result, note that if we were taking a random sample directly from a continous distribution, the GOF test would on average result in a p-value of 0.5. This being the case, we consider an empirical data set p-value result close to 0.5 to be essentially indistinguishable, with respect to the GOF test being used, from data actually drawn from a statistical distribution. p-values of this magnitude were clearly obtained for all three data sets and both tests when performed against a hyperexponential null hypothesis, and to a lesser degree for the CSIL and Long data sets from the KS test against the Weibull null hypothesis.

4 Conclusions

From the results presented in this paper, we feel that there is a compelling case for the superiority of Weibull or hyperexponential distributions in the modeling of resource availability data.

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) make a compelling case for the use of either a Weibull or hyperexponential distribution to approximate the behavior of resources in our three environments. The choice of which to use depends on the application for which the model is needed, and how complex a model can be handled under application constraints. The hyperexponential, although it generally shows a better fit for the data, is significantly more complex than the Weibull models due to its larger number of estimated parameters, the fact that the phase parameter is free and must be decided iteratively, and its resistance to the MLE methods used for the other distributions presented. The Weibull distribution, with its two MLE-computable parameters and relative mathematical simplicity, seems a better choice if speed and complexity are of interest. Regardless, the methods we use in this paper can be used to automatically decide which model is best at any given moment based on GOF analysis. Both the Weibull and hyperexponential are significantly better at capturing the distribution of availability time than the exponential or Pareto, and both can be computed automatically from availability measurement data.

From these results, we hope to generate individual resource models and to improve the quality of simulation and modeling for volatile distributed systems.

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 Inter-*

national Parallel and Distributed Processing Symposium (IPDPS 2000), 2000.
2. S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics*, 23:419–441, 1996.
3. The Avaki Home Page. `http://www.avaki.com`, January 2001.
4. C. E. Beldica, H. H. Hilton, and R. L. Hinrichsen. Viscoelastic beam damping and piezoelectric control of deformations, probabalistic failures and survival times.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. W. Chrabakh and R. Wolski. GrADSAT: A Parallel SAT Solver for the Grid. In *Proceedings of IEEE SC03*, November 2003.
11. R. B. D'Agostino and M. A. Stephens. *Goodness-Of-Fit Techniques*. Marcel Dekker Inc., 1986.
12. The Entropia Home Page. `http://www.entropia.com`.
13. I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., 1998.
14. A. L. Goel. Software reliability models: Assumptions, limitations, and applicability. In *IEEE Trans. Software Engineering, vol SE-11, pp 1411-1423*, Dec 1985.
15. 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.
16. M. Harcol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, 1997.
17. T. Heath, P. M. Martin, and T. D. Nguyen. The shape of failure.
18. 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.
19. J. Kephart and D. Chess. The vision of autonomic computing. *IEEE Computer*, January 2003.
20. J.-C. Laprie. Dependability evaluation of software systems in operation. In *IEEE Trans. Software Engineering, vol SE-10, pp 701-714*, Nov 1984.
21. 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.

22. 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.
23. 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.
24. Matlab by Mathworks. `http://www.matlab.com`.
25. 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.
26. V. Paxon and S. Floyd. Why we don't know how to simulate the internet. In *Proceedings of the Winder Communication Conference also* `http://citeseer.nj.nec.com/paxon97why.html`, December 1997.
27. 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.
28. 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.
29. 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.
30. SETI@home. `http://setiathome.ssl.berkeley.edu`, March 2001.
31. 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.
32. 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.
33. 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.
34. T. Tannenbaum and M. Litzkow. The condor distributed processing system. *Dr. Dobbs Journal*, February 1995.
35. E. Tufte. *The Visual Display of Quantitative Information, 2nd Ed.* Graphics Press, May 2001.
36. The United Devices Home Page. `http://www.ud.com/home.htm`, January 1999.
37. 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.
38. J. Xu, Z. Kalbarczyk, and R. K. Iyer. Networked windows nt system field failure data analysis.
39. 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*.