HAL
open science

The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems

Derrick Kondo, Bahman Javadi, Alexandru Iosup, Dick Epema

To cite this version:

Derrick Kondo, Bahman Javadi, Alexandru Iosup, Dick Epema. The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems. [Research Report] 2009. inria-00433523

HAL Id: inria-00433523

https://inria.hal.science/inria-00433523v1

Submitted on 19 Nov 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems

Derrick Kondo[1], Bahman Javadi[1], Alexandru Iosup[2], Dick Epema[2]
[1]INRIA, France, [2]TU Delft, The Netherlands

Abstract

With the increasing functionality and complexity of distributed systems, resource failures are inevitable. While numerous models and algorithms for dealing with failures exist, the lack of public trace data sets and tools have prevented meaningful comparisons. To facilitate the design, validation, and comparison of fault-tolerant models and algorithms, we have created the Failure Trace Archive (FTA) as an online public repository of availability traces taken from diverse parallel and distributed systems. Our main contributions in this study are the following. First, we describe the design of the archive, in particular the rationale of the standard FTA format, and the design of a toolbox that facilitates automated analysis of trace data sets. Second, applying the toolbox, we present a uniform comparative analysis with statistics and models of failures in nine distributed systems. Third, we show how different interpretations of these data sets can result in different conclusions. This emphasizes the critical need for the public availability of trace data and methods for their analysis.

I. Introduction

With the increasing functionality, complexity, and scale of distributed systems, resource failures are inevitable. For scientific applications, failures can result in frequent performance degradation or in the worst case, premature termination of execution, or data corruption and loss. For commercial applications, failures can cause the violation of service-level agreements, and cause a devastating loss of customers and revenue [10].

A plethora of models and algorithm exist for analyzing, predicting, and resolving failures [6], [21], [15], [12], [20], [2], [1]. At best, these models and algorithms are evaluated using failure traces of a single or limited number of systems. The trace data sets or methods or models based on them are rarely publicly available. Moreover, studies based on failure traces often use traces of different systems. The result is the fragmentation of failure models and fault-tolerant algorithms, as their comparison or cross-validation on different types of systems is difficult if not impossible.

To remedy this situation, we have created the Failure Trace Archive (**FTA**), which comprises public availability traces of parallel and distributed systems, and public tools for their analysis. The community archive approach has been recognized as useful for sharing data in a common format, and has been employed by several communities in the computing domain. The parallel computing community has built the Parallel Workloads Archive [8], the grid computing community has created the Grid Workloads Archive [13], for instance. Efforts such as the Repository of Availability Traces [9], the Computer Failure Data Repository [22], and the Desktop Grid Failure Traces [15] have led to making failure-related data public, but did not establish the premise of a community archive for distributed computing systems. In particular, they did not build a common format for storing failure-related data, and failed to obtain and publish a sufficient number of data sets. In contrast to these early efforts, our main contribution is threefold:

- We design a public failure trace archive, creating a standard format for failure traces, and a toolbox that facilitates comparative trace analysis (Section III).
- Using the toolbox, we present uniform statistical analyses and failure models for nine diverse distributed systems (Section IV);
- We show that differences in the interpretation of failures can change significantly the models and statistics derived from traces (Section V).

II. Background

Throughout this work, we follow the basic concepts and definitions associated with system dependability as summarized by Avizienis et al. [3]. The basic threats to reliability are failures, errors, and faults occurring in the system. A *failure* is an event in which the system

fails to operate according to its specifications. A failure is observed as a deviation from the correct state of the system. We term the continuous period of a service outage due to a failure as an *unavailability interval*. A continuous period of availability is called an *availability interval*.

An *error* is part of the system state that may lead to a failure. Some errors may not be visible from the outside of the system, that is, they may not reach the external state of the system and thus cause failures; such errors are said to be dormant. Errors that do cause failures are said to be *active*. The root cause of an error is a *fault*.

III. Overview of the Failure Trace Archive

The Failure Trace Archive (FTA) can be used in many ways. First, the FTA allows the comparison and cross-validation of a fault-tolerant model or algorithm across identical trace data sets. Second, it allows the evaluation of the generality of a model or algorithm across different types of resources (in terms of reliability or user base, for example). Third, it allows for the evaluation of the generality of a failure trace, i.e., to determine whether measurements are biased to a particular platform or middleware. Fourth, it allows for the determination of which trace data set is most interesting or applicable for a given algorithm or model. Fifth, it allows for the analysis of the evolution of availability in different systems across long timescales. Sixth, it allows for the integration of failure models with other types of models (such as workloads). Seventh, it facilitates the incorporation of traces with a common format into fault simulators or emulators for model or algorithm evaluation.

A. Archive Format

In our experience, the majority of time in measurement and modeling studies is spent in parsing and interpreting the measurements. To accelerate this processing and analysis for others, we have parsed and interpreted 9 diverse distributed systems in a standard format. Here we describe the rationale of the format.

The majority of our collection of traces record times of failures for *resources*, and contain an alternating time series of availability and unavailability intervals. As such, our format is resource-centric (versus job-centric or user-centric) with respect to failures of individual nodes or components of nodes, such as memory, CPU, or hard disks. We believe the format is also applicable to failures of services deployed on top of resources. However, our format does not explicitly describe higher-level failures, such as job failures, though potentially the FTA format could be extended for this type of failure or perhaps combined with the Grid Workload Archive format. Measuring and understanding the relationship between lower-level failures (for example, of nodes or components) to higher-level failures (for example, jobs) is an area for future research.

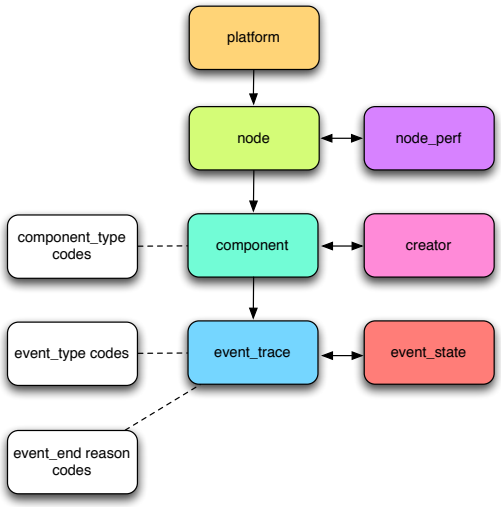


Fig. 1. Overview of the FTA structure.

The trace format is organized hierarchically as follows: Platform → Node → Component → Event Trace. Figure 1 depicts the structure of the FTA, where boxes represent database tables. We summarize the meaning of each table below. Table names are shown in bold.

- A **platform** contains a set of nodes. Examples of a platform include desktop PC's at Microsoft, or nodes in the LANL clusters.
- A **node** contains a set of components, which is a software module or hardware resource of the node. Each node can have several components (e.g. CPU speed, available memory, client availability), each of which has a corresponding trace.
- The **node_perf** describes the node performance, as measured through benchmarks, for example.
- A **component** describes attributes of a software module or hardware resource of a node.
- A **creator** is the person responsible for the trace data set. This table stores details about data copyright, and about projects and published material that use the data.
- An **event_trace** is the trace of an event, with all of corresponding timing information (e.g. start and end times).
- The **event_state** is the state corresponding to an **event_trace**. For example, for CPU availability, the event_state could be the idleness of the CPU. For host availability, it could be the monitoring information associated with the event.

In addition, we have codes that correspond to different types of components (for example, memory, CPU, hard

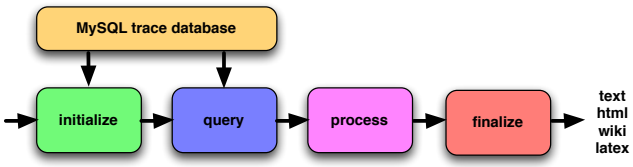


Fig. 2. FTA Toolbox Design.

disk), events (for example, availability or unavailability), and event reason codes (for example, disk crash and CPU overheating).

The best test of a format is its application to real data sets for different types of systems with different types of failures measured in different ways. We applied this format to nine systems ranging from desktops on the Internet to supercomputing clusters. The types of failures included host, CPU, and even service failures. These failures were measured using a variety of methods, such as periodic probing, event notification, load measurement, and even human observation. Given that all of these data sets could be presented in this format with ease, we believe the format is good first step towards a standard. To anticipate future extensions of the format, we have several generic tables with double and string field that can contain additional new information should it arise.

B. FTA Toolbox

We implemented a FTA toolbox to facilitate the comparative analysis of failure traces (see Figure 2). The toolbox is implemented in Matlab, and uses several open-source Matlab packages, such as the Mysql and DataTable packages.

The toolbox takes as input four functions for initializing, querying, processing, and finalizing the data analysis. The initialization and query stages allow one to extract the necessary data from traces located in the MySQL database into Matlab in-memory data structures. By contrast to loading entire data sets into memory from large files, this method allows one to extract into memory only the data that is needed for processing.

Initialization and querying is separated from processing to allow expensive initialization queries to be conducted only once, after which any amount of processing can be done. Also this separation allows the same initialization and queries to be used for many different processing functions. This facilitates code reuse.

The results of initializing and querying are then passed to the processing function. This function is run across each of those results. The processing output is then fed into the finalize function, which can produce tables automatically in latex, HTML, text, and wiki formats using the DataTable module. All graphs and tables in Sections IV and V were produced using the FTA toolbox.

C. Trace Data Sets

The FTA currently has nine formatted data sets, which are listed in Table I, and seven others currently with raw data only. We describe each formatted data set and measurement method briefly. **lanl05** is a data set of 22 HPC systems at Los Alamos National Laboratory. It contains a record for every failure that happened in these systems as well as the root cause [21]. The **g5k06** data set is a trace of a computational grid platform in France (i.e., Grid'5000) which consists of 9 sites, 15 clusters and more than 2,500 processors [12]. The data was collected by periodic inspection and logging of each node's status through the grid middleware called OAR. The **microsoft99** data set contains log files of 51,663 desktops PCs at Microsoft Corporation where their reachability was determined with a ping every hour [6]. The data set of **websites02** was derived from probe-based measurements where a single machine at Carnegie Mellon sent a HTTP file request to web servers periodically every 10 minutes [4]. **pl05** consists of trace data measured between all pairs of PlanetLab nodes using pings every 15 minutes [23]. The **ldns04** data set includes the probe results of 62,201 local DNS servers where the inter-arrival time of the probes followed an exponential distribution with mean of one hour [19]. The **overnet03** data set is a probe-based measurement conducted over the Overnet peer-to-peer file-sharing system [5]. In this data set, the availability of 3,000 hosts was checked every 20 minutes. The **nd07cpu** data set contains traces recorded by Condor from the desktop systems at the University of Notre Dame [20]. The data set is comprised of time-stamped CPU load and idle times of each system, recorded every 16 minutes. Finally, the **skype06** data set is collected by application-level pings of nodes in the Skype superpeer network, every 30 minutes [11].

IV. Analysis of FTA Traces

In the following, we analyze data sets of FTA in two steps. First, we inspect the basic statistics of the traces. Second, we fit distributions for modeling failures in terms of probability distributions of availability and unavailability intervals.

A. Global Statistics

Statistics of availability and unavailability intervals for all data sets are listed in Tables II and III respectively, where the time unit is in hours. The statistics in the tables are mean, trimmed mean, median, standard deviation (std),

System	Type	# of Nodes	Target Component	Period	Year
lanl05	SMP, HPC Clusters	4,750	host	9 years	1996-2005
g5k06	Grid	1,288	host	1.5 years	2005-2006
microsoft99	Desktop	51,663	host	35 days	1999
websites02	Web servers	131	host	8 months	2001-2002
pl05	P2P	692	host	1.5 year	2004-2005
ldns04	DNS servers	62,201	host	2 weeks	2004
overnet03	P2P	3,000	host	2 weeks	2003
nd07cpu	Desktop Grid	700	CPU, host	6 months	2007
skype06	P2P	2,081	host	1 month	2005

TABLE I. Summary of nine data sets in the Failure Trace Archive.

Trace	Mean	TrMean	Median	Std	CV	IQR	Max	Min	Skewness	Kurtosis	No.
lanl05	1779.99	1208.09	280.28	3462.33	1.95	1593.37	34480.23	0.02	3.09	14.29	19874
g5k06	32.41	18.41	7.09	94.24	2.91	24.07	10157.73	0	15.06	695.83	294318
microsoft99	67.01	40.39	10	138.47	2.07	55	840	1.0	3.4	15.8	526078
websites02	11.85	5.17	0.83	40.10	3.38	5.17	1196.55	0	9.02	135.89	47843
pl05	159.48	71.42	1.71	475.61	2.98	35.60	6051.49	0	4.91	34.26	24928
ldns04	140.93	125.79	28.29	193.39	1.37	213.47	559.27	0	1.24	2.97	223596
overnet03	2.29	1.48	1.33	4.63	2.02	1.00	120.11	0	8.03	113.34	33443
nd07cpu	13.73	5.46	1.07	60.05	4.37	7.11	3783.57	0	25.49	1228.74	134176
skype06	16.27	10.12	5.11	34.57	2.12	11.87	465.95	0	4.81	34.38	29217

TABLE II. Statistics of availability intervals for different data sets. (Values given in hours.)

Trace	Mean	TrMean	Median	Std	CV	IQR	Max	Min	Skewness	Kurtosis	No.
lanl05	5.88	1.67	0.97	78.39	13.32	1.98	5325.70	0	43.96	2289.91	23451
g5k06	7.41	0.94	0.05	60.24	8.13	0.19	6314.95	0	26.26	1237.26	294145
microsoft99	16.49	9.15	2	46.50	2.82	14	840	1.0	8.52	105.12	493687
websites02	1.18	0.49	0.17	22.92	19.46	0.34	3311.51	0	111.03	14311.32	47714
pl05	49.61	12.86	0.5	269.90	5.44	6.36	9329.47	0	15.10	340.33	24236
ldns04	8.61	5.47	2.28	20.68	2.40	7.82	533.22	0	8.62	123.06	161395
overnet03	11.98	4	0.33	36.82	3.07	1.67	167.83	0	3.66	15.11	35449
nd07cpu	4.25	0.47	0.27	62.83	14.77	0.36	3616.70	0.04	33.72	1307.29	134026
skype06	14.31	9.45	6.16	30.23	2.11	14.30	596.03	0.02	6.26	62.72	27136

TABLE III. Statistics of unavailability intervals for different data sets. (Values given in hours.)

coefficient of variance (CV), interquartile range (IQR), maximum duration, minimum duration, skewness (the third moment), kurtosis (the forth moment) and number of intervals.

These tables have three types of descriptive statistics. Statistics of the first type (mean, median, trimmed mean) reflect the central tendency of the distributions. Statistics of the second type (CV, IQR, minimum, maximum) measure the spread of the distribution. Statistics of the third type (kurtosis, skewness) reflect the shape of the distribution.

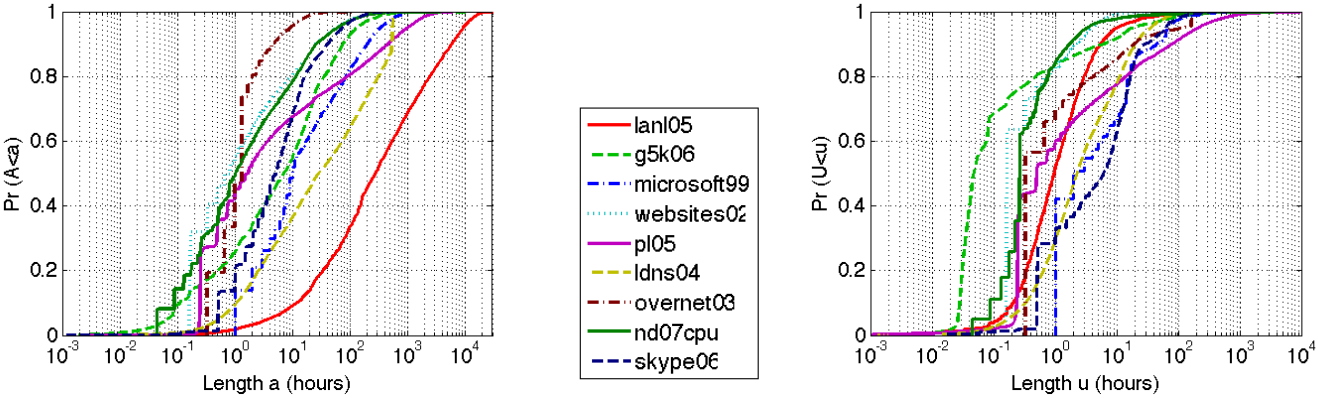
The results reveal that the ratios between the mean and the median for availability and unavailability intervals are quite different for each data set. This indicates that single parameter distributions might not be a good option for the failure model. This could be confirmed by the skewness and kurtosis values that show the availability distributions are right-skewed and long-tailed. Moreover, the unavailability distributions are highly right-skewed and longer tail than the availability distributions.

Additionally, the unavailability intervals have more variability than availability intervals due to higher values of coefficient of variance. Also, analysis of the trimmed mean (the mean value after discarding 10% of extreme values) confirmed that unavailability intervals have greater variability. So, we may need distributions with higher degrees of freedom, e.g., phase-type distributions, to model the unavailability for these data sets.

B. Failure Models

We refer to the distribution of availability and unavailability intervals as the failure model. The cumulative distribution functions (CDFs) of availability and unavailability intervals are plotted in Figure 3(a) and Figure 3(c), respectively. The difference of data sets in terms of distribution is shown in these figures.

We also conducted parameter fitting for various distributions, namely the Exponential, Weibull, Pareto, Lognormal and Gamma distributions. The fitting was done using maximum likelihood estimation (MLE). We adopted



(a) Availability (b) Legend (c) Unavailability

Fig. 3. CDFs of Availability and Unavailability Intervals.

Trace	Exponential	Weibull	Pareto	Log-Normal	Gamma
lanl05	0.005 0.025	0.416 0.571	0.002 0.010	0.475 0.611	0.345 0.488
g5k06	0.012 0.038	0.472 0.597	0.003 0.018	0.394 0.564	0.409 0.507
microsoft99	0.005 0.084	0.294 0.546	0.000 0.049	0.371 0.611	0.198 0.418
websites02	0.000 0.006	0.079 0.354	0.000 0.027	0.188 0.401	0.055 0.182
pl05	0.000 0.000	0.080 0.245	0.002 0.016	0.168 0.321	0.043 0.131
ldns04	0.009 0.042	0.316 0.510	0.002 0.010	0.357 0.527	0.287 0.472
overnet03	0.045 0.460	0.068 0.532	0.000 0.013	0.160 0.660	0.052 0.481
nd07cpu	0.001 0.011	0.348 0.526	0.002 0.063	0.408 0.596	0.167 0.284
skype06	0.048 0.105	0.373 0.493	0.000 0.002	0.452 0.581	0.257 0.375

TABLE IV. P-values resulting from KS and AD tests for availability. A gray box denotes p-value above significance level of 0.05.

Trace	Exponential	Weibull	Pareto	Log-Normal	Gamma
lanl05	0.000 0.004	0.196 0.346	0.000 0.001	0.481 0.607	0.042 0.095
g5k06	0.000 0.000	0.008 0.073	0.000 0.000	0.037 0.144	0.003 0.022
microsoft99	0.004 0.180	0.048 0.529	0.000 0.376	0.076 0.611	0.052 0.368
websites02	0.000 0.023	0.001 0.150	0.000 0.002	0.005 0.209	0.003 0.090
pl05	0.000 0.000	0.035 0.178	0.000 0.004	0.081 0.274	0.019 0.079
ldns04	0.035 0.112	0.404 0.538	0.000 0.001	0.464 0.607	0.277 0.411
overnet03	0.000 0.040	0.003 0.305	0.000 0.204	0.011 0.389	0.005 0.118
nd07cpu	0.000 0.004	0.028 0.219	0.000 0.031	0.126 0.559	0.003 0.032
skype06	0.071 0.191	0.288 0.478	0.002 0.015	0.182 0.449	0.267 0.408

TABLE V. P-values resulting from KS and AD tests for unavailability. A gray box denotes p-value above significance level of 0.05.

Trace	Exp(μ)	Wbl(k, λ)	LogN(μ, σ)	Gam(k, λ)	Exp(μ)	Wbl(k, λ)	LogN(μ, σ)	Gam(k, λ)
lanl05	1779.99	0.48 816.60	5.56 2.39	0.35 5102.71	5.92	0.58 2.18	0.05 1.42	0.38 15.44
g5k06	32.41	0.48 14.37	1.51 2.42	0.34 94.35	7.41	0.35 0.47	-2.00 2.20	0.19 39.92
microsoft99	67.01	0.55 35.30	2.62 1.84	0.41 162.19	16.49	0.60 9.34	1.42 1.54	0.46 35.52
websites02	11.85	0.46 3.68	0.23 2.02	0.31 38.67	1.18	0.65 0.61	-1.12 1.13	0.50 2.37
pl05	159.49	0.33 19.35	1.44 2.86	0.20 788.03	49.61	0.36 5.59	0.40 2.45	0.21 237.65
ldns04	141.06	0.51 79.30	3.25 2.33	0.39 362.43	8.61	0.63 5.62	0.91 1.64	0.51 16.87
overnet03	2.29	0.85 2.04	0.19 0.98	0.91 2.53	12.00	0.44 2.98	0.08 1.80	0.29 41.64
nd07cpu	13.73	0.45 4.16	0.30 2.20	0.30 46.16	4.25	0.51 0.74	-1.02 1.27	0.28 15.07
skype06	16.27	0.64 10.86	1.60 1.57	0.53 30.79	14.31	0.63 9.48	1.40 1.73	0.50 28.53

TABLE VI. Parameters of distributions for availability (left) an unavailability (right). mean: μ, std: σ, shape: k, scale: λ.

two goodness of fit (GOF) tests, namely the Kolmogorov-Smirnov (KS) and Anderson-Darling (AD) tests, to evaluate the fitted distributions. The results of both tests are reported in terms of the p-values in Table IV and Table V for availability and unavailability distributions, respectively. The p-value shown is the average of 1000 p-values, each of which was computed by selecting 30 samples randomly from a data set. This is a standard method for computing p-values as described in [18], [14] when the number of samples is high.

These results reveal that for availability/unavailability distribution we do not have a heavy-tail distribution as the p-values for Pareto are very low. The only exceptions are the distribution of unavailability of `overnet03` and `microsoft99`, which are close to being a heavy-tail distribution. It is worth noting that the AD test is more sensitive to the tail than the KS test. This explains the difference between p-values of the two GOF tests, especially when we have a heavy-tailed data set.

The exponential function seems to be far from the underlying distributions. However, it could be a good fit for the availability distributions of `microsoft99`, `overnet03` and `skype06` and the unavailability distributions of `microsoft99`, `ldns04`, and `skype06` as well. So, the `skype06` data set with the exponential failure model is a good candidate to evaluate Markov models for prediction of host availability/unavailability.

However, we observed that the resolution of the measurement method could have caused the exponential distribution to be a good fit. For example, the `overnet03`, `skype06`, and `microsoft99` systems were measured using probes with periods of 20 minutes, 30 minutes, and 1 hour, respectively. As such, there are no (un)availability intervals less than this length, and in the CDF's shown in Figure 3, there are spikes at those period lengths.

Nevertheless, for all data sets, the Gamma distribution is a good fit for the failure model as the p-values are relatively high. This distribution is a very flexible distribution function and can be adopted for analytical Markov model as well [7].

Additionally, the results of the GOF tests show that the best fit for all data sets are either the Weibull or Log-Normal distributions. As expected from our statistical analysis, the failure model tends to be a long-tailed distribution. However, a few data sets such as `g5k06` and `pl05` do not have a perfect fit for the unavailability distribution.

One possible answer would be the relation of the model with the system architecture. For instance, as the `g5k06` platform has 15 clusters in 9 geographically distributed sites, each cluster could have its own separate model as proposed in [12]. Moreover, as mentioned before, we need distributions with more degrees of freedom such as the hyper-exponential to model the unavailability distributions.

Parameter	Levels: Low	Levels: High	Unit
V: Volatility	$V < 50$	$V > 100$	hour
A: Availability	$A < 60$	$A > 90$	%
M: Measurement	$M < 6$	$M > 12$	month
S: Scale	$S < 1$	$S > 2$	10^3 nodes

TABLE VII. Parameters in the qualitative comparisons. The Medium level is between Low and High levels.

Trace	V	A	M	S	Failure model
lanl05	L	H	H	H	Long-tailed/Long-tailed
g5k06	M	M	H	M	Long-tailed/Long-tailed
microsoft99	M	M	L	H	Short-tailed/Heavy-tailed
websites02	H	H	M	L	Long-tailed/Long-tailed
pl05	L	M	H	L	Long-tailed/Long-tailed
ldns04	L	H	L	H	Long-tailed/Short-tailed
overnet03	H	L	L	H	Short-tailed/Heavy-tailed
nd07cpu	H	M	M	L	Heavy-tailed/Long-tailed
skype06	H	L	L	H	Short-tailed/Short-tailed

TABLE VIII. Qualitative comparison of nine data sets in the FTA. H:High, M:Medium, L:Low. For V, A, M, and S, see Table VII.

Table VI reports the parameter values of different distributions for the availability and unavailability intervals of all data sets under study. For the availability distributions, we analyze the *hazard rate*, i.e., the probability of the next failure with respect to time from the last failure. For the data sets that Weibull or Gamma distributions are good fit, the hazard rate is decreasing. Recall that for such distributions if the shape parameter is less than one, i.e., $k < 1$, then we have a decreasing hazard rate. That means if the systems do not have any failure for long time (longer availability duration) the probability of a failure occurring in the near future decreases. In other words, a decreasing hazard rate could be interpreted as more stability of resources over time [17]. The only hazard rate that is alarming is with `overnet03` where the shape parameter is close to one.

Finally, to have an overall view of the data set characteristics, we present a qualitative comparison of them in Table VIII. The first two parameters are at the node-level, and the other two parameters are at the system-level. The volatility (V) is dependent on the failure rate of each node in the system. The availability (A) is the percentage of time that a node is working properly. The measurement (M) and the scale (S) is the duration of measurement and scale of the system, respectively. The failure model is the tail behavior of availability and unavailability distributions. For each item we assign three different levels as described in Table VII.

For the failure model, we observed that all best fits are long-tailed distributions. However, for the qualitative

table, we applied another classification based on the p-values of the KS and AD tests with a significance level of 0.05. If a data set has acceptable p-values for Pareto or Exponential distribution, the failure model would be heavy-tailed or short-tailed, respectively. Otherwise, the failure model could be classified as long-tailed (for more details about tail behavior, see [7]). Table VIII could provide a good overview of the data sets' characteristics for researchers that want to use real traces for their research.

V. Differences of Interpretation

To emphasize the critical need for public data and analysis methods, we give three examples of where differences of trace interpretation result in differences of the derived models. In particular, we show that differences of interpretation can change dramatically the distribution of failures in terms of passing statistical goodness-of-fit tests and the fitted distributions.

We choose three systems, namely `lanl05`, `g5k06` and `nd07cpu`, where the time of failures can be interpreted differently. On close examination of the lanl05 trace, we found that there are overlapping unavailability intervals. This overlapping of intervals was especially evident in System 16. This system is a cluster of 16 NUMA-based nodes, each of which has 128 processors and 4 NICs.

In some cases, one failure interval completely subsumed another. In other cases, the start time of a failure interval A was greater than the start time of another interval B but less than the stop time of interval B. Moreover, the stop time of A was greater than the stop time B. We believe these intervals might be the result of human error, as the data was manually recorded.

The authors that first described this data set in [21] did not detail the cause of these intervals nor how or why these intervals were interpreted in a certain way. Comparing our statistics of the failures with those in [21], we "reversed engineered" the interpretation, and found that the authors used the *union* of failures intervals having ambiguity. For comparison, we interpret the failure intervals in System 16 differently and optimistically using their *intersection*, calling the resulting post-processed data set **lanl0516B**.

We also found different possible interpretations of the `g5k06` trace. In the raw trace, the states of nodes are given as `available`, `unavailable`, `suspected`, or `dead`. `Suspected` is a state (given mostly automatically) when a node does not behave well according to OAR, the Grid'5000 node manager. The "bad" behavior is detected through many tools, such as the node monitor *finaud*, the jobs monitor *sarko*, and the internal OAR state manager *NodeChangeState*. Pessimistic trace processing would interpret the `suspected` state as a failure, and assume unavailability. An optimistic trace processing would interpret the `suspected` state as a fault but not a failure, and assume availability. The former interpretation is used in the `g5k06` trace described in previous sections and in [12]. We denote the latter interpretation as **g5k06B**.

The nd07cpu trace is the third data set for which there is room for interpretation. The trace is comprised of host idle times and CPU loads as well. Defined in [20], the original definition of CPU availability is when the host is idle without any user for more than 15 minutes, and CPU load (which could be independent of the user) is less than 50%. We relaxed this condition to lengthen the CPU availability time by including the time when a user is present (which in turns would cause zero idle time) and CPU load is less than 10%. That is a reasonable definition of CPU availability as the guest job could still be run on the host without interfering significantly with local jobs. The data set with this latter interpretation is referred to as **nd07cpuB**.

In the following, we present the analysis of different failure interpretations for the aforementioned data sets. First, we compare the empirical distributions graphically. Second, we fit several distributions to each of the data sets, and compare the fitted distributions for each pair of data sets. We compare the fitted distributions statistically with p-values, and then graphically with qq-plots.

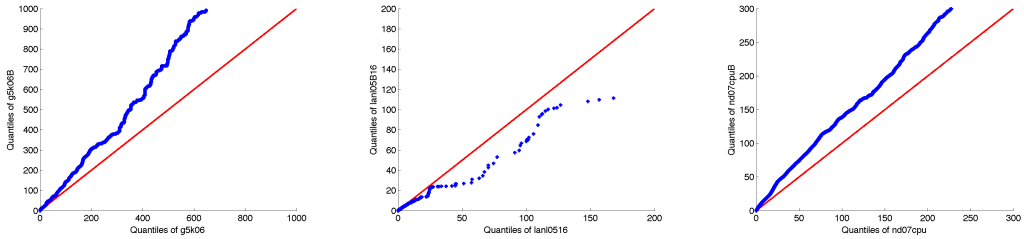
A. Differences of Empirical Distributions

Figure 4 shows the quantiles of the empirical distributions for each pair of data sets. (We only show qq-plots for `g5k06`'s availability, `lanl0516`'s unavailability, `and` `nd07cpu`'s availability to save space.) If the two data sets have the same distribution, their qq-plot will match the line $y = x$, which is plotted in solid red as a reference.

For `g5k06`, in Figure 4(a), we see that `g5k06B` has longer availability intervals (and also shorter unavailability intervals). This is clearly due to the optimistic interpretation of the `suspected` state. The deviation is greatest at the quantile at 1000 hours of `g5k06B`, which corresponds to the the quantile of 600 hours of `g5k06`. Also, the mean availability in `g5k06B` was increased by a factor of 1.50 because the difference of interpretation. The mean unavailability in `g5k06B` was decreased by a factor of 1.13 due to the decrease in the number of failures.

For `lanl0516`, we see little differences of interpretation for availability. This is due to the fact that System 16 is highly available over a long time frame. So changes in unavailability periods do not affect availability periods visibly. However, there are clear differences in the distribution of unavailability as shown in Figure 4(b). The unavailability in `lanl0516B` is much shorter than that in `lanl0516`.

For `nd07cpu`, we find that `nd07cpuB` has longer



(a) g5k06, availability (b) lanl0516, unavailability (c) nd07cpu, availability

Fig. 4. Quantile-Quantile plots of empirical data for ambiguous data sets.

availability and unavailability intervals than `nd07cpu`. In particular, the mean length of availability and unavailability increased by a factor of 1.47 and 1.35 respectively. While the total amount of unavailability decreased in `nd07cpuB`, the small unavailability intervals became availability intervals, after the optimistic processing of the traces. So both the mean lengths of availability and unavailability were increased.

B. Differences of Fitted Distributions

Here we show how the differences of interpretation affect the statistical goodness-of-fit tests of fitted distribution, and their fitted parameters.

Typically, a significance value of 0.05 or 0.10 is used as a threshold for p-values to determine whether to reject the NULL hypothesis that the fitted distribution represents the empirical. We found several cases where the p-values result in conflicting conclusions, i.e., p-values that indicate both rejection and failure of rejection.

For instance, the AD-test for the Weibull distribution fitted to `g5k06`'s unavailability intervals resulted in a p-value of 0.07, whereas the p-value corresponding to `g5k06B` was 0.035. Similarly, for the AD test for the Log-Normal distribution, the p-value is 0.148 for `g5k06` versus 0.057 for `g5k06B`.

Thus, if a threshold of 0.05 is used, we find that the Weibull distribution would not be rejected for `g5k06`'s unavailability distribution, but would be rejected `g5k06B`'s unavailability distribution. Similarly, if a threshold of 0.10 is used, the Log-Normal distribution would not be rejected for `g5k06`, but would be rejected for `g6k06B`'s unavailability distribution.

We find similar cases for `lanl0516` and `lanl0516B`, and `nd07cpu` and `nd07cpuB`. For `lanl0516`, the Gamma distribution is rejected for `lanl0516`'s unavailability intervals but not rejected for `lanl0516B` according to the p-values resulting from the KS test (0.046 versus 0.056). For `nd07cpu`, the Log-Normal distribution is rejected for `nd07cpuB`'s unavailability intervals, but not rejected for `nd07cpu` according to the p-value for the KS test (0.14 versus 0.01).

In addition to quantitative contradictions, we show contradictions graphically as well in Figure 5. There, we plot the quantiles for the fitted Gamma distributions of each pair of data sets. We choose the Gamma distribution as it is analytically easy to use and has a relatively high p-value. In particular, we do so for `g5k06`'s availability distributions, `lanl0516`'s unavailability distributions, and `nd07cpu` availability distributions.

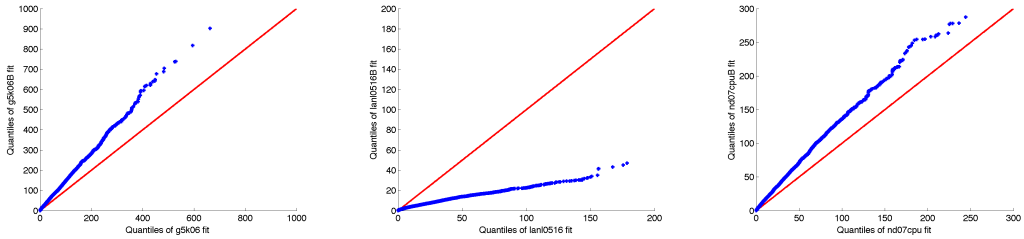
We observe from the qq-plots that the distributions fitted to different interpretations of the same data set are dramatically different. For example, for `lanl0516`, we see that the quantile of 40 hours for `lanl0516B` corresponds to the quantile of 180 hours for `lanl516`.

Furthermore, the impact on the distribution parameters is significant and shown Table IX. Significant differences in parameters are highlighted in grey. For instance, the mean of the exponential distribution for `g5k06B`'s availability is a factor of 1.50 times greater than `g5k06`. Also, different interpretations affect greatly the scale parameter of the Gamma. For instance, the scale parameter of the gamma distribution for `g5k06B`'s availability is factor of 1.39 times greater than `g5k06`.

The fact that both the empirical distributions and fitted distribution mismatch emphasizes the need for public data sets and methods. Otherwise, one cannot determine how the data was exactly interpreted much less why.

VI. Related work

Here we compare our archive with other projects, and also previous works on statistical modeling. While several archives exist, the FTA differs in several respects. First, it defines a standard format that facilitates use and comparison of the traces. Second, the archive contains over 9 data sets provided in this format, and raw traces for over 16 systems. Raw data and the scripts to parse and analyze them are publicly available. The data sets are derived from diverse distributed systems over a long time-frame of over



(a) g5k06, availability (b) lanl0516, unavailability (c) nd07cpu, availability

Fig. 5. Quantile-Quantile plots of fitted distributions for ambiguous data sets.

System	Exp(μ)	Wbl(k, λ)	LogN(μ, σ)	Gam(k, λ)	Exp(μ)	Wbl(k, λ)	LogN(μ, σ)	Gam(k, λ)
g5k06	32.41	0.48 14.37	1.51 2.42	0.34 94.35	7.41	0.35 0.47	-2.00 2.20	0.19 39.92
g5k06B	48.61	0.52 22.66	2.08 2.21	0.37 131.78	6.54	0.35 0.31	-2.36 2.07	0.18 37.00
lanl05	1779.99	0.48 816.60	5.56 2.39	0.35 5102.71	5.92	0.58 2.18	0.05 1.42	0.38 15.44
lanl05B	1774.21	0.48 812.98	5.55 2.39	0.35 5087.60	5.06	0.59 2.12	0.03 1.40	0.41 12.28
nd07cpu	13.73	0.45 4.16	0.30 2.20	0.30 46.16	4.25	0.51 0.74	-1.02 1.27	0.28 15.07
nd07cpuB	20.12	0.48 7.21	0.91 2.07	0.33 61.74	5.75	0.49 0.83	-0.91 1.21	0.26 21.72

TABLE IX. Parameters of distributions for availability (left) and unavailability (right) for ambiguous data sets. mean: μ, std: σ, shape: k, scale: λ. A grey box indicates significant a difference of parameters between data sets.

10 years. Third, it provides a public toolbox for failure trace analysis.

The *Grid Observatory* [16] provides numerous data sets. However, the repository is currently limited to EGEE resources, and only raw data is provided without a common format nor scripts for parsing or analysis. The *Computer Failure Data Repository* [22] provides traces to several supercomputers and clusters. However, no standard format is defined, and the trace data is in raw format only. The *Repository of Availability Traces* [9] contains traces for 5 distributed systems in a common format, and scripts used for parsing the raw data. While a standard format is defined, we believe this format excludes critical information for capturing a range of failure types and systems. For example, the format does not contain causes of failures, the creator, or the component type. The *Desktop Grid Trace Archive* [15] is focused specifically on desktop grids. A generic failure format was not provided, nor were traces of other types of distributed systems.

The *Parallel Workloads Archive* [8] and *Grid Workloads Archive* [13] provide traces of jobs submitted to clusters or Grids. However, this application-level information does not contain information about job, service, or resource failures.

In terms of statistical modeling, our work presents the first uniform statistical analysis and comparison of the nine distributed systems. Most studies, in particular [5], [6], [11], [19], [23], do not focus on modelling issues. A few other studies have also conducted modelling of the distribution of failures. However, the modelling work usually focuses on a particular data set and so the generality of the model was not confirmed. Moreover, the process of failures modeled was significantly different.

For instance, in [21], the authors model the time between failures, and the repair time. They do not model availability interval lengths directly. Yet we believe this is essential for stochastic scheduling algorithms that conduct task assignment based on the probability of task completion.

Also, in [12], the authors model the inter-arrival time between failures. Again, the temporal structure of availability intervals and unavailability intervals is not captured in this model but is needed for stochastic scheduling. Clearly, an inter-arrival time does specify the availability or unavailability length during that time frame.

Similarly, in [4] the entity being modeled is different. In that study, the authors model the *number* of machines available at some time point, considering correlated failures, in the context of a distributed storage system. By contrast, our study focuses on the the continuous *durations* of availability and failures.

VII. Conclusion

Despite the importance of failures in (large-scale) distributed computing environments, few traces collected from real environments are publicly available. To address this situation, which restricts the applicability of failure

models and the development of failure-aware systems, our contribution in this work is threefold:

- We have created the Failure Trace Archive for facilitating the comparative analysis of failures in distributed and parallel systems. We defined a standard trace format, and showed its suitability by converting nine diverse distributed systems to this format. Given traces of this format, we implemented a toolbox that facilitates the comparison of failure statistics, models, and algorithms. Ultimately, we envision that scientists would use the toolbox as a repository of modeling and predictive methods.
- Using the toolbox, we gave a uniform and global statistical analysis of failure in 9 distributed systems. One key finding was that the Weibull and Gamma are the often the best candidates for availability and unavailability distributions. Moreover, the hazard rate wrt to the Weibull was decreasing in all systems. In some cases, the measurement method (in particular the resolution of probing) seemed to cause bias in the distribution of availability, and we identified these data sets with potential bias.
- Finally, we showed how differences of interpretation of trace data sets can result in dramatically different failure models and statistics. This showed that it is critical to make both trace data and analytical methods publicly available.

VIII. Availability of FTA Data and Scripts

The Failure Trace Archive, including technical documentation on the data format, the toolbox. and the trace data sets are available online at: http://fta.inria.fr.

Acknowledgements

We gratefully acknowledge the generous contributors to the Failure Trace Archive that made their trace data sets publicly available. We thank the INRIA ALEAE project directed by Emmanuel Jeannot for supporting this collaborative work. We thank Eric Heien for creating the initial hierarchical version of the Failure Trace Archive format. We thank Cecile Germain for useful discussions with respect to resource versus job versus user-level failures.

References

[1] A. Andrzejak, P. Domingues, and L. Silva. Predicting Machine Availabilities in Desktop Pools. In *IEEE/IFIP Network Operations and Management Symposium*, pages 225–234, 2006.

[2] Artur Andrzejak, Derrick Kondo, and David P. Anderson. Ensuring collective availability in volatile resource pools via forecasting. In *DSOM*, pages 149–161, 2008.

[3] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Sec. Comput.*, 1(1):11–33, 2004.

[4] Mehmet Bakkaloglu, Jay J. Wylie, Chenxi Wang, and Gregory R. Ganger. On correlated failures in survivable storage systems. Technical Report MU-CS-02-129, Carnegie Mellon University, May 2002.

[5] R. Bhagwan, S. Savage, and G. Voelker. Understanding Availability. In *Proceedings of IPTPS'03*, 2003.

[6] W. Bolosky, J. Douceur, D. Ely, and M. Theimer. Feasibility of a Serverless Distributed file System Deployed on an Existing Set of Desktop PCs. In *Proceedings of SIGMETRICS*, 2000.

[7] Feitelson. D. *Workload Modeling for Computer Systems Performance Evaluation*. 2009.

[8] Dror Feitelson. Parallel Workloads Archive. http://www.cs.huji.ac.il/labs/parallel/workload/.

[9] Brighten Godfrey. Repository of Availability Traces. http://www.eecs.berkeley.edu/~pbg/availability.

[10] Failure Rates at Google. http://perspectives.mvdirona.com/2008/06/11/JeffDeanOnGoogleInfrastructure.aspx.

[11] Saikat Guha, Neil Daswani, , and Ravi Jain. An experimental study of the skype peer-to-peer voip system. In *Proceedings of The 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2006.

[12] Alexandru Iosup, Mathieu Jan, Ozan Sonmez, and Dick H.J. Epema. On the dynamic resource availability in grids. In *8th IEEE/ACM International Conference on Grid Computing*, September 2007.

[13] Alexandru Iosup, Hui Li, Mathieu Jan, Shanny Anoep, Catalin Dumitrescu, Lex Wolters, and Dick H. J. Epema. The grid workloads archive. *Future Generation Comp. Syst.*, 24(7):672–686, 2008.

[14] B. Javadi, D. Kondo, JM. Vincent, and D.P. Anderson. Mining for statistical availability models in large-scale distributed systems: An empirical study of seti@home. In *17th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, September 2009.

[15] Derrick Kondo, Gilles Fedak, Franck Cappello, Andrew A. Chien, and Henri Casanova. Characterizing resource availability in enterprise desktop grids. *Future Generation Comp. Syst.*, 23(7):888–903, 2007.

[16] Charles Loomis. The grid observatory. In *GMAC '09: Proceedings of the 6th international conference industry session on Grids meets autonomic computing*, pages 41–42, New York, NY, USA, 2009. ACM.

[17] Farrukh Nadeem, Radu Prodan, and Thomas Fahringer. Characterizing, Modeling and Predicting Dynamic Resource Availability in a Large Scale Multi-Purpose Grid. In *Proc. of the Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2008)*, Lyon, France, May 2008. IEEE Computer Society.

[18] D. Nurmi, J. Brevik, and R. Wolski. Modeling Machine Availability in Enterprise and Wide-area Distributed Computing Environments . Technical Report CS2003-28, Dept. of Computer Science and Engineering, University of California at Santa Barbara, 2003.

[19] Jeffrey Pang, James Hendricks, Aditya Akella, Bruce Maggs, Roberto De Prisco, and Srinivasan Seshan. Availability,usage, and deployment characteristics of the domain name system. In *Internet Measurement Conference (IMC)*, October 2004.

[20] Brent Rood and Michael J. Lewis. Multi-state grid resource availability characterization. In *8th Grid Computing Conference*, September 2007.

[21] Bianca Schroeder and Garth A. Gibson. A large scale study of failures in high-performance-computing systems. In *International Symposium on Dependable Systems and Networks (DSN)*, June 2006.

[22] Bianca Schroeder and Garth A. Gibson. The computer failure data repository. In *Workshop on Reliability Analysis of System Failure Data (RAF'07)*, 2007.

[23] J. Stribling. Planetlab all paris ping. http://pdos.csail.mit.edu/~strib/pl_app/.