Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis

Ting-Ting Y. Lin, Member IEEE
University of California, San Diego
Daniel P. Siewiorek, Fellow IEEE
Carnegie Mellon University, Pittsburgh

Key Words — **Error log, Hard failures, Intermittent and transient faults, Weibull distribution, Failure prediction**

Reader Aids —
Purpose: Presents a new failure prediction technique
Special math needed for explanations: Probability theory
Special math needed to use results: None
Results useful to: Error log analysis and failure prediction

Abstract — **Most error log analysis studies perform a statistical fit to the data assuming a single underlying error process. This paper presents the results of an analysis that demonstrates the log is composed of at least two error processes: transient and intermittent. The mixing of data from multiple processes requires many more events to verify a hypothesis using traditional statistical analysis. Based on the shape of the interarrival time function of the intermittent errors observed from actual error logs, a failure prediction heuristic, the Dispersion Frame Technique (DFT), is developed. The DFT was implemented in a distributed on-line monitoring and predictive diagnostic system for the campus-wide Andrew file system at Carnegie Mellon University. Data collected from 13 file servers over a 22 month period were analyzed using both the DFT and conventional statistical methods. It is shown that the DFT can extract intermittent errors from the error log and uses only one fifth of the error log entry points required by statistical methods for failure prediction. The DFT achieved a 93.7% success rate in failure prediction of both electromechanical and electronic devices.**

1. INTRODUCTION

Trend analysis utilizes the general system behavior as captured in the system error files for fault diagnosis and failure prediction. The basic hypothesis is that there exists a period of instability prior to a hard failure. Work at CMU [8] has demonstrated the feasibility of this hypothesis from observing clusters of disk errors on DEC computers, where the interarrival times of error events were shown to decrease prior to initiation of the repair action. Tsao's research focused on the development of the *tuple* concept as an information organizing and data reduction technique, since multiple reporting of errors was often the first obstacle in data analysis. Another approach to organizing information was developed at the University of Illinois to characterize the relationship among errors recorded in a system error log [2]. A probabilistic model was used to automatically detect symptoms of frequently occurring persistent errors in two large Control Data Cyber systems. It was shown that 85% of the identified error symptoms, typically composed of over 30 error events[1], corresponded to permanent system faults.

The concept of observing system trends for failure prediction was further investigated at Stanford [6]. A methodology that involved three types of analysis was outlined: an average error distribution for each error type, an error distribution for all error types, and a failure/CPU utilization relationship. Both individual and total error distributions demonstrated increasing error generation rates prior to a system crash. A failure prediction algorithm based on the detection of large error clusters, that is, a threshold number of errors, was proposed. Preliminary results using an average of 113 errors for each prediction over six months of data were analyzed showing a 60% chance of success. Although this method has neither been thoroughly tested nor implemented, at least the results clearly indicated that failure prediction based on an increase in error rate, a threshold error number, a CPU utilization threshold, or a combination of these factors may be feasible.

Most prior work has modeled error logs as if they were produced by a single error source. Statistical methods are employed to estimate parameters and identify the error source. Experience with error logs suggest that they intermix entries from a number of error sources. If the error log is analyzed for events such as system crashes (which have a multitude of causes), more data is required to reach a statistically meaningful conclusion than if the events were sorted into contributory causes and analyzed separately. Furthermore, conclusions drawn from analysis of multiple error sources that have been commingled are more difficult to generalize and to apply to new systems.

The paper is divided into five sections. Section 2 describes the experimental data collection system. The data was used in two studies: to determine the statistical characteristics of intermittent and transient errors in section 3 and to validate a predictive trend analysis heuristics for intermittent errors in section 4.

Section 3 develops a methodology for separating an error log into independent sources. The separation methodology is based upon the fact that intermittent errors reoccur, often at an increasing rate. In section 3.2, traditional statistical analysis methods are applied to each intermittent error source showing that at least 25 errors spanning up to 18 months are required to identify the trend. In section 3.3, errors not associated with an intermittent error source are assigned to a single transient error source to which traditional statistical analysis are also applied, yielding Weibull functions whose shape parameters are

[1] In one example shown in the paper, a symptom is extracted from three events with a total of 36 error records.

0018-9529/90/1000-0419$01.00©1990 IEEE

less than 1.0. Next section 3.4 applies traditional statistical analysis to the total error log (ie, all intermittent and transient error sources) and compares the parameters extracted to those derived from analyzing an artificial error log formed from combining two pure error sources. Similarities between the actual and artificial error logs supports the concept that actual error logs are composed of entries from multiple error sources. Thus variations between error logs and within an error log over time are a function of the relative contributions from each pure error source. A good working hypothesis is that a single transient and a single intermittent error source exist in the error log at any given time.

Based on the experience gained in factoring error logs into individual error sources as well as from interviews with maintenance personal, a new heuristic, the Dispersion Frame Technique (DFT) is introduced in section 4. The DFT extracts error log entries caused by individual intermittent faults, and then applies one of its five failure prediction rules according to the interarrival patterns of the errors. The five rules are shown to capture behavior corresponding to that detected by traditional statistical analysis techniques. Moreover, the DFT used only one fifth of the error log entries required by statistical methods, and achieved a 93.7% success rate in failure prediction for both electromechanical and electronic devices when applied to 22 months of error log data from the Carnegie Mellon University (CMU) campus-wide Andrew file system.

Finally, section 5 concludes the paper. In particular, it summarizes previous work on trend analysis including both statistical methods using the Weibull distribution and the DFT as shown in table 1. The table lists the approach, the average number of events required to identify a trend, and the percentage of success in applying the approach. Although the data sets adopted from individual sources are different, it nevertheless shows that the DFT uses the least number of data points given similar error event recording techniques.

TABLE 1
Summary on Trend Analysis Approaches

	Appraoch	Average # of events	Percentage of success
[Iyer 86]	Joint Probability	36	85%
[Nassar 85]	Thresholds on Error Numbers and CPU Utilization	113	60%
Wiebull	Statistical Weibull Fit	25	—
Lin	Dispersion Frame Technique	5	93.7%

2. THE MEASUREMENT ENVIRONMENT

Andrew is the software system consisting of four major components that supports the Carnegie Mellon University large-scale distributed computing environment. The environment consists of personal workstations with raster graphics, high bandwidth networks, and a time-sharing file system called VICE. Currently there are 13 file servers in the VICE file system. The file server hardware is composed of a SUN 2/170 (or SUN 3/280) workstation with a Motorola 68010 (or 68020) microprocessor, a Xylogics 450 disk controller and Fujitsu Eagle disk drives each accommodating up to 800 Megabytes. A file server has at most two disk controllers and each controller supports up to two disk drives. According to published statistics, the Mean Time Between Failure (MTBF) of the disk drive is 20 000 hours. Therefore when the file system is run at full capacity with 52 disk drives spread over 13 file servers, one could expect a hard disk failure in the file system every 400 hours[2]. The load on the file servers was empirically determined to be constant with very little variation over time. Thus one of the variables (workload) previously found to affect error rate, could be removed as a factor resulting in a simplified analysis.

In order to support error collection and analysis, the device driver and the error handling software in the SUN UNIX kernel was modified. This software instrumentation enables logging of error messages prior to system crash and collecting the relevant information in each kernel error report in a uniform format for analysis. Moreover, an on-line monitoring and predictive diagnostic system composed of two major pieces of software, the Agent and the Diagserver, was developed for the VICE File System. The *Agent*, residing in the node under observation, is in charge of kernel message acquisition. The *Diagserver*, residing in a centralized site, performs data analysis for failure prediction and diagnosis. This paper concentrates on the issues and techniques in failure data analysis and prediction. A detailed list of the kernel changes, and the architecture of the on-line monitoring and predictive diagnostic system can be found in [4].

3. STATISTICAL ANALYSIS OF ERROR LOGS

Sources of information for data analysis include the automatic error log of 13 VICE file servers, collected by the on-line predictive diagnostic system, and an operator's log. Data collected from February 1986, the first date of file server operation, until January 1988, was used for studying the characteristics of permanent, transient, and intermittent faults. The increasing number of file servers placed in service over the twenty-two month period resulted in a total of 20 workstation-years of data. Errors are considered to be manifestations of faults. Thus the error log contains errors whose causes are faults. It is shown that the typical error log contains events that are caused by a mixture of transient and intermittent faults. The operator's log contains permanent failure information as well as repair actions attempted to remedy the problem. The terms intermittent and transient have been used interchangeably

[2]However the observed file system disk mean time between failure is 1671 hours and the system mean time to crash due to all hard failures is 504 hours [Lin 88].

in the literature [3]. However, the distinction between the two fault types is repair [1], where intermittent faults are repairable by replacement, while transient faults are not since the hardware is physically undamaged. Before applying the traditional statistical methods to analyze the data, some mathematical background is provided in the next subsection.

3.1 *Mathematical Background*

The failure distribution mathematically characterizes the probability of system failures as a function of time. The exponential, gamma, Weibull and lognormal are all well known distributions in failure analysis [5]. The Weibull function is used in this research and is defined as:

$$R(t) = e^{-(\lambda t)^{\alpha}}, \ \alpha > 0 \text{ and } \lambda > 0$$

where α is the shape parameter, and λ is the scale parameter. Note that when α equals 1, the distribution function reduces to the exponential:

$$R(t) = e^{-(\lambda t)}$$

The hazard function is the time-varying failure rate. The Weibull hazard function is defined as:

$$z(t) = \alpha\lambda \ (\lambda t)^{\alpha-1}$$

The shape parameter α directly influences the hazard function as follows:

- If $\alpha < 1$, the hazard function is decreasing with time;
- If $\alpha = 1$, the hazard function is constant with time, ie, the exponential distribution;
- If $\alpha > 1$, the hazard function is increasing with time.

It has been suggested that transient faults are characterized by $\alpha < 1$, permanent faults by $\alpha = 1$, and intermittent faults by $\alpha > 1$ [7]. In our study of permanent faults (ie, those obtained from the operator's log that involve actual repairs), the statistics showed an estimate of α to be 0.92 and the data followed an exponential distribution within 0.05 level of significance [4]. In this paper, we will show that while $\alpha < 1$ is appropriate for transients, that $\alpha > 1$ is an oversimplification for intermittent faults. However, we must first develop a methodology for separating the error log into its constituent error sources.

3.2 *Analysis of Intermittent Faults*

Since intermittent faults are repairable by replacement, they are associated with a physical subsystem, sometimes referred to as a field replaceable unit (FRU). Error log entries are identified by the hardware or software error detection mechanism which reported the error. For each FRU a timeline was constructed, composed of only those error log entries identifying the FRU. Since a FRU may be repaired several times during the course of the study, a heuristic was required to differentiate between two independent intermittent faults. The time based clustering heuristic previously used to collapsing multiple error log events into a single logical event [8] was extended to collapse multiple intermittent errors into a single physical failure. The system operator's log was used to identify the repair activities which culminated in insertion of a new FRU. There were 29 repairs of hardware failures in the 22 months of observation including 7 disks, 7 CPUs, 7 memory boards, and 8 disk controllers.

If the interarrival time between errors on a FRU's timeline were more than a week apart (168 hours), the errors were considered to be unrelated and the growth of the timeline was terminated. Error events left on a timeline were called intermittent while all the rest were called transient. The threshold of 168 hours was chosen based on two observations. First, in the study of transient errors, discussed in section 3.3, the average interarrival time for transients regardless of FRU was found to be 354 hours over all systems, and the smallest mean observed in an individual system was 154 hours. The transient data was analyzed for several thresholds between 154 to 354, but all resulted in the same Weibull parameters. Second, in order to determine a single threshold, 168 hours was chosen to avoid potential cyclic patterns due to daily dependencies on workload.

The timelines of 16 of the 29 repair actions and their corresponding intermittent errors are shown in [4]. The remaining 13 had fewer than five error entries, too few for analysis. Four of these timelines are reproduced in figure 1 with '+' representing device errors and '^' indicating device repair. Periods of increasing error rate, which appear as either clusters of errors or decreasing interarrival times between errors (suggesting a Weibull failure distribution with $\alpha > 1$), are observed. Although the clustering patterns shown in figure 1 spanned about 200 hours, the majority of the 16 failures which recorded error log activity were preceded by error log entries over 1000 hours prior to repair.

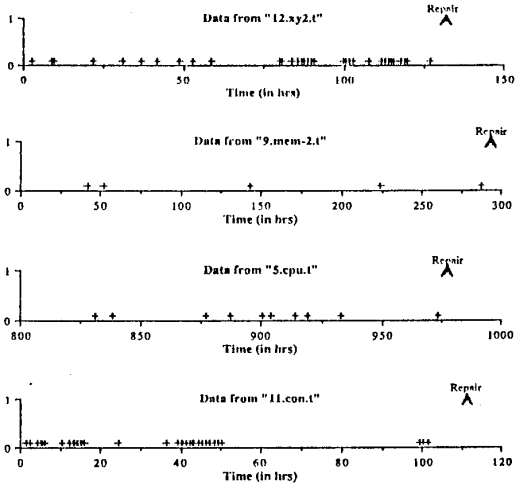

Figure 1. Timelines of Intermittent errors Leading to Corresponding Disk, Memory, CPU, and Disk Controller Repair Actions on Selective File Servers.

Modeling of intermittent faults begins with the analysis of the interarrival times of their manifestations (eg, errors) [7]. This is done by calculating the difference between the time stamp information of each intermittent error and by formulating the hazard function to identify the associated distribution function. A linear regression analysis was performed on the interarrival time data to estimate the maximum likelihood estimates for the Weibull parameters. The technique is based upon the transformation of the Weibull cumulative distribution function into a linear function of $\ln(t)$, where the initial elements α and λ are obtained from the slope and the Y-intercept of the straight line. The linear estimates can be used as initial values for an iterative Newton-Raphson solution method to obtain the maximum likelihood estimates of α_{ML} and λ_{ML}. Subsequently, a chi-square goodness of fit is performed to evaluate the fit of the Weibull to the observed data. A significance level of 0.05 was selected, which means the probability that a chi-square random variable with m degrees of freedom, where m is calculated as the number of categories[3] minus the number of parameters to be estimated minus 1, will exceed χ_c^2 is c (ie, 0.05).

[3] Categories are chosen such that the expected observations in each category is no less than 4.

All 29 repairs and their associated intermittent errors were evaluated. The maximum likelihood estimates of Weibull and exponential parameters and the chi-square goodness-of-fit tests were calculated and listed in table 2. Each of the 29 repair actions is listed under the corresponding file server. The FRU (Field Replaceable Unit) column shows the repaired device, the Errors column lists the number of intermittent error events before repair, and the Mean column shows the average of the interarrival times between intermittent errors. The next three columms list the parameters of the Weibull function fit and the chi-square test result. α(linear) and λ(linear) are the linear regression estimates, and α_{ML} and λ_{ML} are the maximum likelihood estimates. #cat. and Chi-sq show the number of categories and the result of the Chi-square statistic. The last two columns list the λ_{ML} parameter of an exponential fit and its chi-square statistic.

Although the amount of data is insufficient to perform the chi-square goodness-of-fit tests for all 29 failures, implying their estimates of the parameters are inconclusive, there are several interesting findings [4]. First, on average, 21 intermittents were observed per repair activity, and the mean of the interarrival times is 58 hours. This indicates that the first symptom might occur as early as 50 days (1218) hours prior to the attempted

TABLE 2
Failure Distributions for Intermittent Faults Leading to Corresponding Repair Actions

					Weibull						Exponential		
#	File Server	FRU	Errors	Mean	α (Linear)	λ (Linear)	α (MLE)	λ (MLE)	#Cat.	Chi-sq	λ (MLE)	# Cat.	Chi-sq
1	Vice2	xy0-2	48	19	0.6419	0.0455	1.5356	0.0116	2	0.07	0.0501	2	0.01
2		xy0-1	1	–	–	–	–	–	–	–	–	–	–
3		mem	11	102	0.7809	0.0097	0.9210	0.0092	2	1.33	0.0098	2	1.53
4	Vice3	con	13	118	0.8474	0.097	0.7631	0.0098	2	2.25	0.0084	2	2.57
5	Vice4	cpu	7	174	1.2728	0.0084	1.1737	0.0054	1	0.00	0.0057	1	0.00
6	Vice5	cpu(A+B)	14	275	0.2251	0.0342	0.4647	0.0078	2	1.62	0.0036	2	2.46
7		xy2	13	21	0.6671	0.1599	0.6848	0.0381	2	0.19	0.0476	2	0.10
8		con	13	21	0.6671	0.1599	0.6848	0.0381	2	0.19	0.0476	2	0.10
9	Vice6	xy2-1	40	10	0.1677	9.8432	0.5213	0.0303	2	9.23	0.0930	2	6.98
10		xy2-2	18	39	0.1683	3.5546	0.4781	0.0225	1	0.00	0.0255	1	0.00
11	Vice7	cpu	5	106	1.3338	0.0191	1.0784	0.0091	1	0.00	0.0094	1	0.00
12	Vice8	cpu	18	210	1.2794	0.0080	0.9116	0.0047	3	1.57	0.0047	3	2.09
13	Vice8	con	0	–	–	–	–	–	–	–	–	–	–
14	Vice9	con-1	0	–	–	–	–	–	–	–	–	–	–
15		con-2	3	–	–	–	–	–	–	–	–	–	–
16		con-3	0	–	–	–	–	–	–	–	–	–	–
17		mem-1	6	43	1.2870	0.0220	0.6640	0.0298	1	0.00	0.0023	1	0.00
18		mem-2	4	59	6.3050	0.0130	8.0870	0.0119	3	0.00	0.0169	1	0.00
19	Vice10	cpu	0	–	–	–	–	–	–	–	–	–	–
20	Vice11	cpu	0	–	–	–	–	–	–	–	–	–	–
21		con	34	69	0.0877	∞	0.3427	0.0123	1	0.00	0.0143	2	0.21
22		mem+cpu	2	–	–	–	–	–	–	–	–	–	–
23	Vice12	mem(A+B)	23	113	0.3656	0.0152	0.6559	0.0093	3	3.54	0.0088	3	4.33
24	Vice12	xy2-1	341	1	0.3298	5.5697	1.0285	0.0400	3	2887	0.9266	3	2863
25		xy2-2	1	–	–	–	–	–	–	–		–	–
26		con+cpu	0	–	–	–	–	–	–	–	–	–	–
27	Vice13	mem	0	–	–	–	–	–	–	–	–	–	–
28	Vice14	mem	0	–	–	–	–	–	–	–	–	–	–
29		cpu	3	65	0.8525	0.0223	2.2249	0.0089	1	0.00	0.0152	1	0.00

Note 1. Repair actions 7&8 occurred within a short period of time. The disk controller was replaced due to a string of disk errors. However, xy2 disk was lost at power up. Therefore data points were accounted for both repairs.

repair. Second, among the 17 Weibull shape parameters (α_{ML}) estimates made, six are greater than 1 (Vice2.xy0-2, Vice4.cpu, Vice7.cpu, Vice9.mem-2, Vice12.xy2-1, and Vice14.cpu), three are close to 1 (Vice2.men, Vice3.con, and Vice8.cpu), and eight are less than 1 [4]. Thus simply looking for α greater than 1 is insufficient to identify the trend of an intermittent fault. Third, the statistical analysis shows that most of the repair action were performed before system statistical trends developed, indicating that users do not tolerate the large number of errors required for a statistical method to predict a failure trend. Therefore a new method as described in section 4 was sought for trend analysis and failure prediction.

3.3 *Distribution of Transient Errors*

Transient errors were extracted from the system event log by subtracting known hard-failure-induced intermittent errors as described in the previous section. Similar to the intermittent errors, the transient errors were a combination from several pure error sources. The data exhibited three types of the most commonly seen transient errors: system software errors (event type SOFT), parity errors (event type MEM), and unscheduled system reboots (eg, watchdog resets). Each type of transient error could have been composed from multiple pure error sources but there was insufficient information associated with the error types to further separate them into unique error sources. Hence only the total population of transient errors was modeled.

Modeling of transient errors begins with the analysis of their interarrival time [7]. The interarrival times are calculated using the time stamp information from the system event log. The hazard function is formulated and analyzed to identify its associated reliability function. In total, 446 transient errors[4] are plotted in figure 2. The x-axis divides the interarrival times into 20-hour bins, while the y-axis shows the number of occurrences in each bin. The obvious skew toward the low end for all the data indicates that the Weibull distribution is a likely candidate for the reliability function.

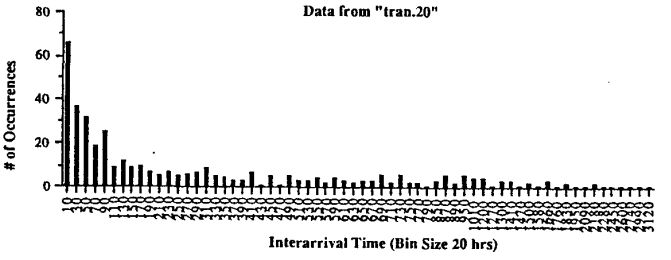

Figure 2. Hazard Functions of the VICE File System Transient Errors

Table 3 shows the shape and scale parameters of the transient errors for the thirteen file servers. Its format is identical to that of table 2 except that the FRU column is replaced by the "total" time in hours each file server was under observation. In order to perform the chi-square test, at least four categories are required for a Weibull and three for an exponential fit. Rows marked with an asterisk represent those file servers whose number of categories were insufficient to perform the chi-square goodness-of-fit tests, implying their estimates of the parameters are inconclusive. Even so, only one out of the four asterisk-marked rows has an α greater than one. The α_{ML} values for all the other servers are less than one. The underlined chi-square values failed the chi-square test. In fitting the data to the Weibull function, VICE2 is the only server that failed

[4]The total number of system crashes due to transient was 269, accounting for 90% of the total number of system crashes.

TABLE 3
Statistics for Transient Errors on VICE File System

				Weibull						Exponential		
File Server	Time	Errors	Mean	α (Linear)	λ (Linear)	α (MLE)	λ (MLE)	#Cat.	Chi-sq	λ (MLE)	# Cat.	Chi-sq
Vice2	16770	41	386	0.5223	0.0035	0.5666	0.0023	5	*6.28*	0.0026	4	*8.59*
Vice3	16770	54	262	0.7991	0.0045	0.8047	0.0041	10	8.10	0.0038	9	12.64
Vice4	16770	27	561	0.4427	0.0022	0.6569	0.0022	5	5.59	0.0018	5	*11.01*
Vice5	15360	31	291	0.8148	0.0031	0.6471	0.0037	5	5.32	0.0034	4	*8.78*
Vice6	15360	87	167	0.3387	0.0096	0.7161	0.0059	14	11.28	0.0059	14	17.23
Vice7	13584	25	407	0.9137	0.0024	0.8886	0.0025	5	3.28	0.0025	4	5.89
Vice8	12936	25	370	1.0931	0.0039	0.7818	0.0030	5	3.34	0.0027	5	5.67
Vice9*	12936	18	515	0.4065	0.0048	0.4013	0.0018	1	0.00	0.0019	1	0.00
Vice10	12936	62	154	0.3276	0.0131	0.6407	0.0065	9	5.89	0.0065	9	10.12
Vice11*	12936	11	668	1.3767	0.0022	1.0686	0.0015	2	0.67	0.0015	2	0.44
Vice12*	12672	20	435	0.3883	0.0049	0.5855	0.0028	3	4.27	0.0023	3	5.14
Vice13*	12672	12	733	1.0093	0.0015	0.5817	0.0020	2	2.01	0.0014	2	2.56
Vice14	12672	33	351	0.5998	0.0031	0.4686	0.0034	4	4.35	0.0028	5	6.69

Key: Symbol * indicates insufficient numbers of categories to formulate Chi-sq test.

the chi-square test assuming a 0.05 significance level. VICE2 passed the chi-square test at a significance level of 0.1. We believe that some extraneous data from testing might have been captured due to the experimental nature of VICE2 in is early stage of operation (ie, it was the first file server). Therefore, transient faults follow the Weibull distribution with a decreasing failure rate. In summary, each file server had an average of 34 transient errors with an average interarrival time of 354 hours. The smallest mean of the interarrival times for transient errors was 154 hours in VICE10. Table 3 indicates that a minimum of 25 error points spanning up to 18 months are required to gain a useful estimate of the parameters of the function. In addition, the assumption of near constant load on the file servers was tested by sampling the load every 53 hours (ie, the system mean time to crash). The system usage was found to be uniform, thus verifying the assumption that system load could be factored out as a variable in this study.

As we have seen, the error log is a mixture of both transient and intermittent errors. The arrival process of transient errors was shown to have a decreasing failure rate while intermittent errors tend to have a constant or increasing failure rate. A similar observation was found in a study by [McConnel 79] where the α of the Weibull distribution for parity errors (ie, transients) was 0.5 while the α for all crashes (mixture of transient and intermittent errors) was closer to 1.0 with a value of 0.8. In order to study the impact of mixing two error sources in more detail, a simulation study was conducted and compared with the experimental data.

3.4 *Analysis of Event Logs with Intermittent and Transient Events Intermixed*

The same statistical modeling process was performed on the entire error log data with intermittents and transients intermixed. Table 4 shows the parameters of the entire error log for each file server assuming a Weibull distribution with α_e and λ_e. The row marked with an asterisk indicates that the data from VICE 13 is insufficient to perform the chi-square goodness-of-fit test, implying its estimates of the parameters are inconclusive. The underlined chi-square values denote those that failed the chi-square goodness-of-fit test assuming a 0.05 level of significance. VICE6 passed the chi-square test with a 0.25 level of confidence, VICE9 with a 0.1, and VICE12 with 0.95, which is an unrealistic significance level. It can be seen that α_{ML} values for all the servers are less than one. The effect of intermittent faults on the entire error log can be observed in the parameter changes between tables 3 and 4, and is summarized in table 5. In table 5, it can be seen that given $\alpha_t < 1$ for transient faults, α_i of intermittent faults directly influences α_e of the entire error log. The underlined values denote inconclusive results. The amount of the shift $\Delta\alpha$ (ie, $|\alpha_e-\alpha_t|$) is a function of the ratio of the shape parameters α_t/α_i and the relative number of occurrences of errors N_t/N_i, where N denotes the number of error occurrences. Therefore, if transient errors cannot be isolated from the system error log, the resulting statistical analysis of the entire error log to identify the presence of an intermittent fault will require more data points than for intermittent data alone. Furthermore the analysis is less likely to be conclusive. The effect of α_i on α_e is summarized in table 6. In general, if α_i is less than 1, $\alpha_e < \alpha_t$; if α_i is close to 1 (+0.1), $\alpha_e > \alpha_t$, and if α_i is greater than 1, $\alpha_e > \alpha_t$. However these observations do not hold in all cases (eg, VICE7 and 8 in table 5). Since the amount of naturally occurring data is insufficient to explore the effects of the relative contributions due to multiple pure error sources (in particular each α_t/α_i and N_t/N_i in table 5 is unique), simulation was used.

Mathematically, if one assumes that the failure distributions of the transient and intermittent errors are independent, one should be able to extract one from the entire error log since the form of both distributions are known. The reliability function of the error log, $R(t)$, is the product of the reliability functions of the transients $R_t(t)$ and the intermittents $R_i(t)$. Assuming both have Weibull distributions,

TABLE 4
Statistics for VICE Error Log

						Weibull				Exponential		
File Server	Time	Errors	Mean	α (Linear)	λ (Linear)	α (MLE)	λ (MLE)	#Cat.	Chi-sq	λ (MLE)	# Cat.	Chi-sq
Vice2	16770	101	153	0.3125	0.0200	0.6184	0.0042	10	12.74	0.0065	10	*18.78*
Vice3	16770	67	248	0.7612	0.0046	0.7871	0.0043	11	9.21	0.0040	10	13.55
Vice4	16770	34	432	0.4089	0.0025	0.7006	0.0028	6	7.87	0.0023	6	*13.17*
Vice5	15360	58	223	0.4144	0.0078	0.5037	0.0043	8	5.32	0.0045	8	5.25
Vice6	15360	145	100	0.2751	0.0388	0.5899	0.0060	15	*31.43*	0.0099	14	*29.60*
Vice7	13584	30	367	0.8840	0.0027	0.7995	0.0029	6	4.27	0.0027	6	7.78
Vice8	12936	43	307	1.1003	0.0040	0.8225	0.0031	5	3.32	0.0028	5	5.69
Vice9	12936	31	273	0.2707	0.0171	0.4633	0.0046	4	*4.43*	0.0037	3	*4.31*
Vice10	12936	62	154	0.3276	0.0131	0.6407	0.0065	9	5.89	0.0065	9	10.12
Vice11	12936	47	228	0.2734	0.0285	0.4060	0.0034	4	3.83	0.0044	4	4.64
Vice12	12672	385	30	0.1129	274.98	0.4040	0.00236	8	*1714*	0.0327	8	*2225*
Vice13*	12672	12	733	1.0093	0.0015	0.5817	0.0020	2	2.01	0.0014	2	2.56
Vice14	12672	36	313	0.5871	0.0034	0.4992	0.0036	5	5.32	0.0032	6	7.67

Key: Symbol * indicates insufficient numbers of categories to formulate Chi-sq test.

TABLE 5
The Effect of Combining Error Sources on the Total Error Log Shape Factor

Vice#	α_t	α_i	α_e	$\Delta\alpha$	α_t/α_i	N_t/N_i	μ_t/μ_i
2	0.5666	1.5356 0.9210	0.6284	↑	–	0.68	–
3	0.8047	0.7631	0.7871	↓	1.05	4.15	2.4
4	0.6569	1.1737	0.7006	↑	0.56	3.86	4.5
5	0.6471	0.4647 0.6848	0.5037	↓	–	–	–
6	0.7161	0.4781 0.5213	0.5899	↓	–	–	–
7	0.8886	1.0784	0.7995	↓	0.82	5	4.5
8	0.7818	0.9116	0.8225	↑	0.86	1.39	1.7
9	*0.4013*	0.6640 8.0870	0.4633	↑	–	–	–
10	0.6407	–	0.6407	–	–	∞	–
11	*1.0686*	0.3427	0.4060	↓	3.12	0.31	1.1
12	*0.5844*	0.6559 1.0285	0.4040	↓	–	–	–
13	*0.5817*	–	0.5817	–	–	∞	–
14	0.4686	2.2249	0.4992	↑	0.21	11	7

TABLE 6
Change in α_e from α_t as a Function of α_i

α_i	<1	~ 1	>1
α_e	↓	↑	↑

$$R(t) = R_t(t) * R_i(t)$$

$$R(t) = \exp\{-[(\lambda_t t)^{\alpha_t} + (\lambda_i t)^{\alpha_i}]\},$$

$$R_t(t) = \exp\{-(\lambda_t t)^{\alpha_t}\},\ R_i(t) = \exp\{-(\lambda_i t)^{\alpha_i}\}$$

One can also calculate the individual hazard function as follows:

$$Z(t) = Z_t(t) + Z_i(t)$$

$$Z(t) = \alpha_t\lambda_t(\lambda_t t)^{\alpha_t - 1} + \alpha_i\lambda_i(\lambda_i t)^{\alpha_i - 1}$$

It is obvious that neither $R(t)$ nor $Z(t)$ can be easily evaluated since they do not resemble any well known distributions. However, they can be simulated using superposition of renewal point processes [Cox 62], where the interarrival times between errors were assumed to be independent random variables. Each point process corresponds to a pure error source. There are two characteristics of this model. First, in the limit, as the number of point processes approaches infinity, the overall sequence approaches in exponential independent of the form of the individual point processes. Second, the analysis of the intervals between successive points generated from a collection of independent processes will give very little or no information about the form of the individual processes. In fitting our error log data to this model, it is expected that the overall log, produced from a number of independent error processes, should exhibit an exponential distribution. Moreover, it would be difficult to deduce the distributions of the individual faults from the overall distribution. We simulated the mixing of only two error functions: intermittent and transient. The two error processes are independent Weibull processes. Process 1 has parameters α_1 and λ_1, and process 2 has α_2 and λ_2. The resultant mixed process is fitted to a Weibull function characterized by α_e and λ_e. The number of events from each process is N_1 and N_2 respectively. The ratio N_1/N_2 can be expresses as the ratio of the means, μ_1/μ_2:

$$\mu_1/\mu_2 = \frac{\Gamma(1+\alpha_1^{-1})/\lambda_1}{\Gamma(1+\alpha_2^{-1})/\lambda_2}$$

Since $\Gamma(1+\alpha^{-1}) = (\alpha^{-1})!$, the ratio reduces to

$$\mu_1/\mu_2 = \frac{\lambda_2\alpha_1^{-1}!}{\lambda_1\alpha_2^{-1}!}$$

Our simulation program generated several hundred mixed processes, each with a total of 500 data points. Each resultant mixed process was fitted to a Weibull function characterized by α_2 and λ_2. Table 7 shows a subset of the results from the simulation. Two interesting observations can be made. First, the superposition of two processes shows that the α_e of the overall sequence approaches one (an exponential distribution). Second, given a fixed ratio α_1/α_2 and α_e and λ_e increase with increasing λ_2. This implies that if one process starts generating errors at a higher frequency, the resultant combined error log will have a larger α_e and λ_e.

These simulation results reflect the real data in table 5. Take VICE4 for example, μ_t/μ_i is calculated as:

$$\mu_t/\mu_i = \frac{\Gamma(1+\alpha_t^{-1})\lambda_i}{\Gamma(1+\alpha_i^{-1})\lambda_t} = \frac{\Gamma(1+0.657^{-1})0.0054}{\Gamma(1+1.17^{-1})0.0022} \sim 4.5$$

which roughly corresponds to the ratio of N_t/N_i ($=27/7=3.86$). The small difference between the simulated and observed results is probably due to more than the two pure error sources being present in the real error log. However, the assumption that a single transient and a single intermittent source exists in the error log at any given time is an adequate first order assumption.

The above statistical analysis shows that it usually takes over 25 errors spanning up to 18 months before accurate estimates of the parameters can be made. Moreover, most of the repair actions were performed before system statistical trends developed, indicating that users do not tolerate that large a number of errors. Thus a new method should be sought for fault prediction. The next section introduces the Dispersion Frame Technique. The DFT was developed to perform trend analysis for failure prediction based on the observation that there exists a period of increasing error rate of intermittent errors before most hardware failures. The DFT can identify a problem with as few as 3 errors spanning as little as an hour.

TABLE 7
Superposition of Two Weibulll Processes

#	α_1	λ_1	α_2	λ_2	μ_1/μ_2	α_e	λ_e
1	0.5	0.05	0.5	0.001	0.02	0.6465	0.0047
2	0.5	0.05	0.5	0.005	0.1	0.7378	0.0060
3	0.5	0.05	0.5	0.01	0.2	0.7396	0.0063
4	0.5	0.05	0.5	0.05	1	0.7432	0.0066
5	0.75	0.05	0.5	0.01	0.15	0.8585	0.0068
6	0.75	0.05	0.5	0.02	0.3	0.8715	0.0069
7	0.75	0.05	0.5	0.04	0.6	0.9105	0.0074
8	0.75	0.05	0.5	0.08	1.2	1.085	0.0119
9	0.75	0.05	1	0.001	0.03	0.8413	0.0070
10	0.75	0.05	1	0.005	0.15	0.8569	0.0072
11	0.75	0.05	1	0.01	0.3	0.9672	0.0076
12	0.75	0.05	1	0.05	1.5	1.1467	0.0103
13	0.75	0.05	1.5	0.001	0.034	0.8897	0.0069
14	0.75	0.05	1.5	0.005	0.134	0.9406	0.0071
15	0.75	0.05	1.5	0.01	0.34	0.9867	0.0081
16	0.75	0.05	1.5	0.05	1.34	1.1882	0.0106
17	0.75	0.05	1.5	0.1	3.4	1.3819	0.0168
18	1	0.05	0.5	0.01	0.1	0.9769	0.0075
19	1	0.05	0.5	0.02	0.2	0.9896	0.0080
20	1	0.05	0.5	0.04	0.4	0.9956	0.0088
21	1	0.05	0.5	0.08	0.8	1.0110	0.0092
22	1	0.05	0.5	0.1	1	1.0652	0.0096
23	1	0.05	1	0.001	0.05	1.0194	0.0086
24	1	0.05	1	0.005	0.1	1.0554	0.0080
25	1	0.05	1	0.01	0.2	1.0826	0.0087
26	1	0.05	1	0.05	1	1.1234	0.0116
27	1	0.05	1.5	0.001	0.02	1.0594	0.0088
28	1	0.05	1.5	0.005	0.1	1.0951	0.0090
29	1	0.05	1.5	0.01	0.2	1.1896	0.0092
30	1	0.05	1.5	0.05	1	1.2948	0.0124

4. THE DISPERSION FRAME TECHNIQUE

The Dispersion Frame Technique (DFT) determines the relationship between error occurrences by examining their closeness in time (duration) and space (affected area). The technique utilizes Dispersion Frames (DF) and Error Dispersion Indices (EDI). A Dispersion Frame is the interarrival time between successive error events of the same error type. The Error Dispersion Index is defined as the number of error occurrences in half of a DF. A highly related group of errors exhibits a high EDI. The DFT consists of a set of heuristics developed from the experiences gained in separating error logs into their constituent error sources and the experiences of hardware technicians.

Based on the findings from the statistical analysis in sectin 3, a DF of 168 hours was used to activate the heuristics. The DFT is illustrated in figure 3 and proceeded as follows:

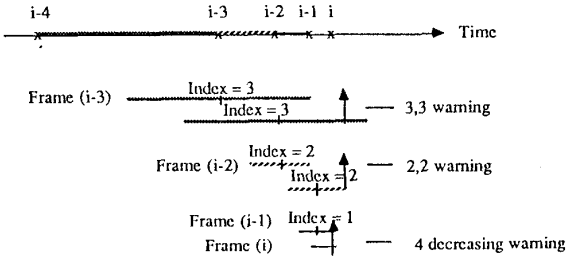

Figure 3. Dispersion Technique

1. For each device, a time line of the five recent error occurrences for that device is drawn. The DFT is activated when a frame size less than 168 hours is encountered. Figure 3 shows the error events i-4, i-3, i-2, i-1, and i.

2. Centered around each error occurrence on the time line are the previous DFs. Frame (i-3) is the interarrival time between events i-4 and i-3, and is centered around events i-3 and i-2; frame (i-2) is the DF between events i-3 and i-2, and is centered around events i-2 and i-1, etc.

3. The number of errors from the *center* to the right end of each frame is measured and designated as the EDI. Figure 3 shows that the EDI is 3 for the first application of frame (i-3), and 2 for the first application of frame (i-2). The DF frames are successively centered on error events later in time, frame (i-3) is shown centered on errors i-3 and i-2 in figure 3.

4. A failure warning, denoted by upward arrows, is issued under the following conditions:

a. *3.3 rule*: when two consecutive indices from successive application of the same frame exhibit an EDI of at least 3 (eg, frame (i-3) centered on errors i-3 and i-2 in Figure 3),

b. *2.2 rule*: when two consecutive indices from two successive frames exhibit an EDI of at least 2 (eg, frame (i-3) centered on error i-3 and frame (i-2) centered on error i-2),

c. *2 in 1 rule*: when a dispersion frame is less than one hour,

d. *4 in 1 rule*: when four error events occur within a 24-hour frame,

e. *4 decreasing rule*: when there are four monotonically decreasing frames and at least one frame is half the size of its previous frame (eg, frame (i-3), frame (i-2), which is less than half of frame (i-3), frame (i-1), and frame (i) in Figure 3).

5. Several iterations among steps 2, 3 and 4 are usually performed before a warning can be issued.

These five rules have been shown to mathematically cover a range of values for α, the Weibull shape parameter observed during the data analysis in section 3. The range of α associated with each rule is derived in the following paragraphs. To

simplify the analysis, consider five errors with interarrival times represented by the following frames: 2w=frame(i-3), x=frame(i-2), y=frame(i-1), z=frame(i). Further, for illustration purpose, 2w is assigned 40 units.

3.3 rule

The 3,3 rule sets up the relationships that two consecutive EDIs from the same frame are greater or equal to 3. This represents the scenario of sharply decreasing interarrival time frames based on a single previous frame. The combinatorial equations to be solved are $w >= x + y$ and $w >= y+z$. Possible values for $2w$, x, y, and z are depicted by the family of DF curves in figure 4, where the envelope of the graph is denoted by squares. The envelope for the error rate covers a range from increasing to decreasing, resulting in an α value that can be locally greater than, equal to, or less than one. This sequence of DFs of error events is termed as a "band" and covers a wide range of α values.

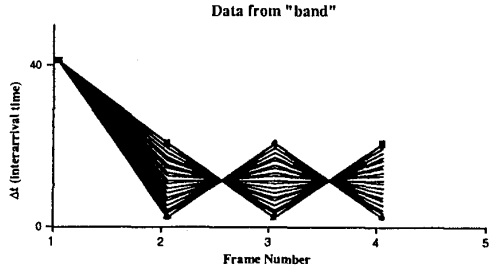

Figure 4. Local Behavior for the 3,3 Rule - a "*Band*"

2.2 rule

The 2,2 rule exhibits a scenario of the uniform decreasing time frames based on two prior consecutive frames. That is, when two consecutive EDIs from two successive frames are greater or equal to 2, the relationships between the frames yield the following combinatorial equations: $w > x$, $w > y$, $x > 2y$, $x > 2z$, $x + y > w$, and $y + z > x/2$. Possible values of $2w$, x, y, and z are depicted in figure 5, showing a mixture of increasing and constant error rate. This range of DFs of error events is termed as a 'cone' shape with α values greater than or equal to 1.

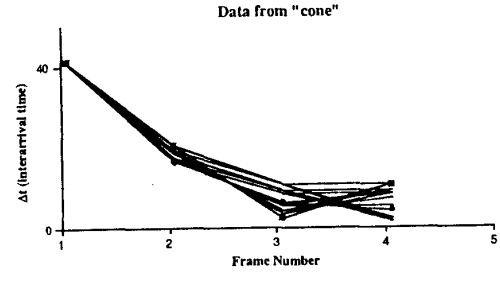

Figure 5. Local Behavior for the 2,2 Rule --- a "Cone"

4 decreasing rule

This rule governs four monotonically decreasing frames and at least one frame is half the size of its previous frame. The represents the scenario of the steady decreasing interarrival time frames. Therefore the relationships of the frames are $2w > x > y > z$, and at least one frame is half the size of the its previous frame. Figure 6 depicts a rectangular envelope representing the cases each frame is half of its previous frame. This illustrates that the error rate is decreasing, thus the range of DFs covers α values strictly greater than one.

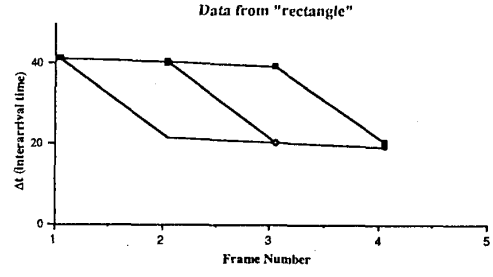

Figure 6. Local Behavior for the 4 Decreasing Rule

2-in-1 and 4-in-1 rules

These two rules represent the conventional thresholding technique widely used in industry. The 2-in-an-hour rule is termed a *dimple*, and has the shape of a sharp transition in the slope of the DFs. The 4-in-a-day rule is termed a "*valley*", and is in the form of a sharp slope followed by a relatively flat portion.

The five rules were then applied to the intermittent errors leading to each of the 29 permanent failures in the VICE file system to verify their coverage. The rules that first predicted each failure and the local α values are given in table 8 through 11. In most cases there were fewer than the 25 events required to make significant estimates of the α values as determined by the chi-square goodness-of-fit test. However, the α value produced by the statistical methods is recorded as an indicator of the trend in the value of α. Except in the case of VICE3's disk controller repair, all predictions correctly identify the trend in the error events, implying that the DFT is a robust and simple approximation compared to statistical analysis. Moreover, the rule that was fixed for VICE3 (ie, the 4 decreasing) did indicate the existence of the problem which had failed to show up in the statistical analysis. The following sections examine in more detail the intermittent error behavior preceding the hard failures in electromechanical devices (the hard disk) as well as electronic devices such as memory boards, CPU, and disk controllers.

4.1 *Analysis of Electromechanical Devices-the Hard Disk Example*

The Fujitsu Eagle M2351 is a compact, moving-head Winchester disk with a storage capacity of up to 474 megabytes.

The non-removable media has 6 platters, 842 cylinders, 20+1(servo) heads/drive. Two typical disk error messages are shown in Figure 7.

```
ErrLog:DISK:9/18044/563692507/829000:errmsg:xylg:syc:cmd
  6:reset failed (drive not ready) blk 0
ErrLog:DISK:10/18044/563692507/8690000:xyopen:xyl:xyc0:cmd
  6:unit not online
```

Figure 7. Example Disk Error Messages

The first message starts with event type DISK, followed by the sequence number 9, the process ID 18044, the UNIX time 563692507 seconds and 829000 μsec, detected in kernel routine *errmsg*, drive xy1, partition 'g', then followed by the UNIX system message stating that it was detected by the xyc disk controller while executing command 6 (reset) and was returned with an error "reset failed (drive not ready)" at block 0. The second disk message was issued 40000 μsec later by the *xyopen* routine after exercising xyc0, the first disk controller. It had a sequence number 10 and an error "unit not online" returned while executing command 6 (reset) on drive xy1.

The data from the 13 file servers over the 22 months showed seven disk repairs, resulting in a MTTF of 1671 hours over all disks. Table 8 shows all the file servers repaired, number of intermittent errors accounted for by the failure, the failed device (FRU), the estimated Weibull α_i from statistical methods, the first rules fired, the elapsed time between the first rule firing and the disk repair, and the corresponding α predicted by the DFT rules, the α_{dft}. The * next to the file server name means the repair occurred prior to the instrumentation of the UNIX kernel for on-line monitoring and predictive analysis [Section 2]. In one case, there was only one related error message reported twenty-one days prior to the catastrophic failure. In the other case, in which the CPU, both disk controllers, and the Ethernet board were replaced, a parity error occurred just prior to the loss of the disk and thus was not accounted for. These two cases indicate that conventional operating systems are incapable of providing enough information for failure prediction, demonstrating the need of better instrumentation and error logging than that provided by UNIX.

TABLE 8
Prediction Time for Seven Disk Failures

File Server	* Points	FRU	α_i	DFT Rule	Prediction Time	α_{dft}
Vice2	48	xy0	1.536	2,2 & 4 dec.	702	> =
Vice2*	1	xy0	—	—	—	—
Vice5	13	xy2	0.685	3,3	83	> = <
Vice6	40	xy2	0.521	2 in an hour	141	—
Vice6	18	xy3	0.478	2 in an hour	10	—
Vice12	341	xy2	1.029	3,3 & 4 dec.	60	> = <
Vice12*	—	xy2	—	—	—	—

Key: Symbol > means the α value is greater than 1, = means equal to 1, and < means less than 1.
Symbol * indicates repairs occured prior to the installation of the on-line monitoring and prediction system.

Among the five successful predictions, several triggered more than one rule. The earliest prediction time was 702 hours before the repair action took place and the shortest was 10 hours. Furthermore, the α_{dft}, suggested by the DFT methods correspond closely to those obtained from the statistical analysis, α_i, where i stands for intermittent. Over the 22 months, there were only two false alarms, suggesting these DFT rules actually differentiate intermittent error sources from the rest of the error log entries.

4.2 *Analysis of Electronic Devices-CPU, Disk Controller, and Memory Boards Examples*

The CPU

The computational power of the SUN2 file server lies in the Motorola 68010 processor with a 10 MHz clock (SUN3 file servers use the 25-MHz MC68020 processor). The 68010 CPU supports virtual memory operation (up to 16 megabytes of virtual address space) and is interfaced to an internal bus that connects to a high-speed RAM (optional for SUN2), and the Multibus. The Multibus provides links to the disk controller, the tape controller, the Ethernet, the floating point processor (optional for SUN2) and I/O devices.

The multiplicity of interconnections makes the separation of CPU errors from other errors difficult. Among the seven CPU repair actions over the twenty-two months, three occurred prior to the instrumentation of the operating system and are marked by * in table 9. One replacement fixed a hung server, one fixed a watchdog reset[5], and one was associated with the first message in the error log after system installation. The other four were caused by a collection of parity and software errors. Of the four parity-error-related repairs, one was a disk controller repair, one was preceded by a memory board replacement, followed by seven transient parity errors during a 3-day period, one was in conjunction with a memory board replacement, and the last had only three parity errors.

Table 9 lists the first rule fired and the prediction time for each repair action. Three out of the four repairs (VICE4, VICE5, VICE 7, and VICE14), in which parity errors were logged, were successful predictions using DFT rules. Although VICE14 had less than 5 error log entries, the premature prediction result was nevertheless listed to show the effectiveness of the 2-in-an-hour technique. Four observations were noted. First, frequent watchdog resets were only repaired by CPU replacements. Second, it is shown that the repairs with * had inadequate information to form a prediction. Thus, improved kernel instrumentation [4] enables tracking of error events, including watchdog resets, potentially leading to an identification for failure prediction. Third, multiple parity errors or parity errors occurring in more than one memory board in a short period of time (less than one minute) indicate possible CPU-related faults. One possible explanation of multiple and/or

[5]Watchdog timers, which can be implemented in sofware or hardware, detect software failures. If the timer is not reset before it expires, the corresponding process has probably failed.

TABLE 9
Prediction Time for Seven CPU Repairs

File Server	# Points	α_i	2 in an hr.	Prediction Time	α_{dft}	Comments
Vice4	7	1.174	2,2	940	> =	Repaired w/ Controller
Vice5	14(7)	0.465	3,3	73	> = <	A memory repair followed by 7 transient errors
Vice7	5	1.078	×	—	—	Repaired w/ Memeory
Vice8*	18	0.912	×	—	—	Watchdog Reset
Vice10*	0	—	—	—	—	First Message in Log
Vice11*	0	—	—	—	—	System Hung
Vice14	3	2.225	2 in an hr.	1	—	—

Key: Symbol > means the α value is greater than 1, = means equal to 1, and < means less than 1.
Symbol * indicates repairs occured prior to the installation of the on-line monitoring and prediction system.

distributed parity error indications is the high speed operation between the CPU and the bus. Therefore, the maintenance personnel also attributes the address and bus error messages caught by the *panic* routine in the kernel as manifestations of CPU errors as depicted in figure 8. These messages have an error type SOFT, followed by the same event header information described for the disk error messages, and the UNIX system message.

```
ErrLog:SOFT:26/61/551333847/2090000/:panic: Bus error
ErrLog:SOFT:15/22249/55117908/769000/:panic:Address error
```

Figure 8. Example CPU Error Messages

And fourth, the α_{dft} values of VICE4 and VICE5 correspond to those α_i's obtained from statistical analysis, thereby supporting the validity of the effectiveness of the DFT.

The Disk Controller

The Xylogics 450 disk controller includes two sequencers and a microprocessor which provide interfaces for up to four disk drives. Disk errors occurring at either the disk controller, disk drives, or disk media are reported through the disk controller to the device driver software. In the 22-month observation period, there were eight disk controller replacements. Five replacements occurred prior to the installation of the on-line predictive diagnostic system, marked by * in table 10 and thus had no more than 3 error events in the three months prior to repair (VICE8, 9, and 12). Three had more than 5 disk errors and all were identified before repair actions, including a system hung. Two of the three α values suggested by the DFT correspond to those obtained from the statistical analysis. In the VICE3 case, DFT did indicate the existence of a problem prior to the development of a statistical trend and that the number of data points was insufficient for a useful estimate of α.

Five replacements were fixes to system hangs. In particular, the VICE3 replacement was preceded by several disk error messages followed by three system hangs 23 hours prior to the repair. These three system hangs were due to the same reason (the df[6] operation waiting for the controller to complete), thus coercing into our VICE3 event. According to technicians' experiences, disk controllers are often the cause of system hangs due to the stringent timing demand to coordinate the high speed CPU and slower speed disk drives. Moreover, in the cases of VICE5 and VICE11, both disk controller and drive were replaced, indicating better device driver code could be written in the operating system for fault isolation.

TABLE 10
Prediction Time for Disk Controller Repairs

File Server	# Points	α_i	DFT Rule	Prediction Time	α_{dft}	Comments
Vice3	13	0.763	4 dec.	0	>	System Hung
Vice5	13	0.685	3,3	34	> = <	Lost Disk also
Vice8*	0	—	—	—	—	System Hung
Vice9*	0	—	—	—	—	System Hung
Vice9*	3	—	—	—	—	—
Vice11	34	0.343	3,3	29	> = <	Lost Disk also
Vice12*	0	—	—	—	—	System Hung

Key: Symbol > means the α value is greater than 1, = means equal to 1, and < means less than 1.
Symbol * indicates repairs occured prior to the installation of the on-line monitoring and prediction system.

The Memory Boards

Each file server contains from four to seven 1-megabyte memory boards. The main memory for the SUN2 uses parity error detection code (SUN3 uses Error Correcting Code). Memory board errors usually manifest themselves as parity errors. A typical parity error message has an error type MEM, followed by the event header, and the UNIX message with the error address and data as illustrated in figure 9.

[6] df is a UNIX shell command that prints out the amount of free disk space available on the specifiec file system.

```
ErrLog:MEM:4/1247/561587203/183000:Parity Error: Address
  0x2f906a, Data 0x90, Bus Error Reg 4e91 <VALID,LPARERR>
```

Figure 9. Example Memory Error Message

Since parity errors have long been considered a manifestation of transient faults, it is necessary to separate those parity errors that are truly transient from those that are due to a board defect. This is achieved by using the address field of the parity error message to identify the source of errors. Parity errors occur in the same memory board are identified as "intermittent", and the board is replaced to correct the problem. If parity errors were generated from more than one board, usually all the memory boards involved were swapped or replaced to isolate fault locations. For example, since each board contains one megabyte, the address 0x2f906a belongs to the third megabyte thus the third memory board, and the address of 0x11b2e4 belongs to the second board.

In the 22 months of observation, there were 24 memory board replacements and four memory board swaps. Among the 24 replacements, 12 were involved in the three occasions when all the boards were replaced. The other 12 were single board replacements. Two events, four-memory-board replacements in VICE5 and two-memory-board replacements in VICE7, were performed in conjunction with CPU repairs discussed earlier and will not be repeated here. It was found that not all replacements resulted in successful repairs. One example is VICE12's all-memory-board replacement where the error rate after the repair attempt was three times higher than before. The memory boards were subsequently swapped to VICE7, and the problem was solved after replacing two memory boards. Therefore only seven memory board replacements were identified as effective repairs and summarized in table 11. Repairs in VICE 11, 13, and 14, marked by *, were performed prior to the instrumentation of the UNIX kernel, resulting in an unsuccessful prediction. The four repairs preceded by a minimum of four parity errors were successfully predicted by DFT. Moreover, three out of the four repairs showed that the α_{dft} values predicted by the DFT reflected the trends indicated in the statistical analysis parameters α_i.

4.3 *Evaluation of the Dispersion Frame Technique*

Evaluation of DFT is performed in two parts. The first part concerns the frequency of rule firings for each device. A total of 29 rules were fired during the twenty-two month period of data analysis including seven from the 3,3 rule, three from the 2,2 rule, four from the 4 decreasing rule, five from the 4-in-a-day rule, and ten from the 2-in-an-hour rule. Table 12 shows the frequency of the rules fired during the fault prediction analysis for each device. Two numbers are listed under each rule, the number of total firings and the number of times that particular rule was first to detect the trend. Athough the 4 in 1 rule (four events in one day) did not succeed in issuing the first warning, it nevertheless was activated five times prior to repairs. Furthermore, there were cases where more than one rule was fired simultaneously to detect the system trend. Since each rule was fired more than once and for more than one device, some degree of confidence is generated in the generality and robustness of the individual rules. Although no single rule is adequate to identify all the trends, it is speculated that only a small rule set will be required.

The second evaluation involves the performance of the prediction rules on each device. Table 13 lists the number of total repairs, repairs occurred after the instrumentation of the UNIX kernel, number of successful predictions, repairs occurred prior to the instrumentation, and false alarms for each device. Since the UNIX kernel did not provide enough information to perform trend prediction, the effectiveness of the DFT will only be evaluated on the repair actions occurring after the enhanced instrumentation. The high success prediction rate of 93.7% (15/16 after enhanced instrumentation), using a small set of rules and only five false alarms, shows that the DFT is very effective when coupled with good system instrumentation.

5. CONCLUSIONS

Data collected from the 13 public-domain file servers over a 22 month period were analyzed. In the 20 workstation-years of data, there were 29 permanent failures, 610 intermittent errors, 446 transient errors, and 296 system crashes. 13 of the

TABLE 11
Prediction Time of the Memory Repairs

File Server	# Points	α_i	DFT Rule	Prediction Time	α_{dft}	Comments
Vice2	11	0.921	2,2	88	> =	System Hung
Vice9	6	0.664	3,3	119	< = >	—
Vice9	4	8.087	4 decreasing	1	>	—
Vice11*	2	—	—	—	—	With CPU
Vice12	23	0.656	2 in an hour	30	—	Swapped with Vice7
Vice13*	0	—	—	—	—	—
Vice14*	0	—	—	—	—	Message File Corrupted

Key: Symbol > means the α value is greater than 1, = means equal to 1, and < means less than 1.
Symbol * indicates repairs occured prior to the installation of the on-line monitoring and prediction system.

TABLE 12
Frequency of DFT Rules Fired

	3,3		2,2		4 decreasing		4-in-a-day		2-in-hour	
	Total Firing	First Firing	Total Firing	First Firing	Total Firing	First Firing	Total Firing	First Firing	Total Firing	First Firing
Disk	3	2	1	1	2	2	3	0	7	2
Mem	1	1	1	1	1	1	1	0	1	1
CPU	1	1	1	1	0	0	0	0	1	1
Disk Controller	2	2	0	0	1	1	1	0	1	0

TABLE 13
Component Performance on DFT

	Total	Repairs after Instrumentation	Prediction Succeeded	Repairs before Instrumentation	False Alarms
Disk	7	5	5	2	2
CPU	7	4	3	3	0
Disk Controller	8	3	3	5	1
Mem	7	4	4	3	2

29 failures had three or fewer error log entries, 8 of the 13 had no error log entries, therefore a thorough error logging mechanism is needed for data analysis. The mean time between permanent system failure was calculated to be 6552 hours, mean time between intermittent errors was 58 hours, mean time between transient errors was 354 hours, and mean time between system crash was 689 hours. The ratio between permanent faults and total system crashes was 0.1. The ratio between intermittent errors and permanent faults was 21, indicating that the first symptom could appear over 1200 hours prior to repair. Moreover, the ratio between system crashes due to nonpermanent failures and total errors was 0.255, meaning that on average only one in four errors results in a system crash.

Since hard failures account for less than 10% of the total system crashes. The other 90% were caused by a combination of intermittent and transient errors. Statistical analysis of logs composed of both intermittent and transient errors requires more data points to identify a trend. Thus the logs need to be factored into individual components. A methodology for factoring out intermittents from transients was proposed and validated by analyzing each of the 29 physical repairs. Furthermore, the majority of intermittent errors leading to those permanent faults showed periods of increasing error rate prior to hard failures. Sixteen of the 29 permanent failures were recorded by the on-line monitoring and predictive diagnostic system [4]. Each of these 16 were preceded by intermittent errors suggesting that problems slowly develop into catastrophic failures and that their characteristics can be used for failure prediction. The initial 13 permanent faults observed by the unmodified UNIX error logging either had their symptoms went unrecorded by the conventional operating system error logging or rapidly proceeded to a catastrophic failure. With the help of statistical analysis to obtain estimates of transient and intermittent characteristics, the DFT was developed. The DFT was able to extract intermittent errors from the transient errors in the system error log and provide a set of rules for fault prediction. These rules only require between three and five events in order to make a decision and cover a variety of error patterns possessing the same failure characteristics corresponding to those obtained from statistical analysis. These five rules predicted 93.7% of the hard failures under the augmented instrumentation of the UNIX operating system, with an average of 160 hours prior to repair activity. If the predicted failures were removed prior to catastrophic failure, the Andrew file system MTTF would be 16 times (ie, the probability of a failure characteristic being recorded and predicted is 15/16 or a factor of 16/1 improvement) better with the availability of the on-line monitoring and predictive diagnostic system. Further work is required to see if the DFT rules can be successful in an on-line failure prediction system. In particular, we should determine whether the five trend analysis rules are adequate or whether they need to be augmented.

ACKNOWLEDGMENT

This research was supported by the Office of Naval Research under contract N00014-85-K-008. In addition we thank the *Guest Editor* and referees for their helpful comments.

REFERENCES

[1] M. A. Breuer, "Testing for intermittent faults in digital circuits", *IEEE Trans. Computers*, vol C-22, 1973 Mar, pp 241–246.

[2] R. K. Iyer, L. T. Young, V. Sridhar, "Recognition of error symptoms in large systems", in *Proc. 1986 Fall Joint Computer Conf.* Dallas, Texas, 1986 November.

[3] S. Kamal, "An approach to the diagnosis of intermittent faults", *IEEE Trans. Computers*, vol C-24, 1975 May, pp 461–467.

[4] T-T. Y. Lin, "Design and evaluation of an on-line predictive diagnostic system", *PhD Thesis*, technical report, Department of Electrical and Computer Engineering, CMUCSD-88-1, Carnegie Mellon University, 1988 April.

[5] T. Nakagawa, S. Osaki, "The discrete Weibull Distribution", *IEEE Trans. Reliability*, vol R-24, 1975 Dec, pp 300–301.

[6] F. A. Nassar, D. M. Andrews, "A methodology for analysis of failure prediction data", *CRC Technical Report No. 85-20*, Stanford University, Polo Alto, California, 1985 September.
[7] D. P. Siewiorek, R. S. Swarz, *The Theory and Practice of Reliable System Design, Digital Press*, 1982.
[8] M. M. Tsao, "Trend analysis and fault prediction", *Technical Report 130*, Carnegie Mellon University, Department of Computer Science, Pittsburgh, Pennsylvania 1983 May.

AUTHORS

Dr. Ting-Ting Y. Lin; Department of Electrical and Computer Engineering; University of California, San Diego; La Jolla, California 92093-0407 USA.

Ting-Ting Y. Lin (S'84, M'88) received the BS (1980) in Control Engineering from Chiao-Tung University, Hsinchu, Taiwan, R.O.C, and her PhD degree in Computer Engineering from Carnegie Mellon University in 1988. She is an Assistant Professor in the Electrical and Computer Engineering Department, University of California, San Diego. Her current research interests include fault-tolerant computing, system performance and reliability, and design for testability.

Dr. Daniel P. Siewiorek; Department of Electrical and Computer Engineering School of Computer Science; Carnegie Mellon University; Pittsburgh, Pennsylvania 15213 USA.

Daniel P. Siewiorek. For biography see *IEEE Trans. Reliability*, vol 39, 1990 Oct, p 408.

Manuscript TR90-302 received 1990 January 20; revised 1990 May 18.

IEEE Log Number 37712 ◄TR►

MANUSCRIPTS RECEIVED *MANUSCRIPTS RECEIVED*

"A Monte Carlo simulation algorithm for finding MTBF", Dr. Chul Kim □ Agency for Defense Development □ POBox 35 □ Taejeon 300-600 □ Republic of KOREA. (TR90-109)

"Fixed-time life tests based on fuzzy life characteristics", Akihiro Kanagawa, Research Associate □ Dept. of Industrial Engineering □ College of Engineering □ University of Osaka Prefecture □ Sakai, Osaka 591 □ JAPAN. (TR90-110)

"A note on the Venn & Ben diagrams", Dr. Hoang Pham □ M/S 2406 □ EG&G Idaho Inc. □ POBox 1625 □ Idaho Falls, Idaho 83415-1625 □ USA. (TR90-111)

"Deterministic reliability modeling of dynamic redundancy", Dr. Klaus D. Heidtmann □ Dept. of Computer Science □ Universitaet Hamburg □ Bodenstedtstrasse 16 □ D-2000, Hamburg 50 □ Fed. Rep. GERMANY. (TR90-112)

"Periodic 'garbage collection' policies", Dr. Toshio Nakagawa □ Dept. of Industrial Engineering □ Aichi Institute of Technology □ 1247 Yachigusa Yagusa-cho □ Toyota 470-03 □ JAPAN. (TR90-113)

"Bayesianism is for turkeys", Ed M. Dougherty Jr. □ SAIC □ 311 Park Place Blvd; Suite 360 □ Clearwater, Florida 34625 □ USA. (TR90-114)

"Risk evaluation: Power system induced bush & grass fires and the catastrophe potential", R. H. Stillman, ME □ Electrical Engineering Dept. □ The University of Queensland □ St. Lucia, Queensland 4067 □ AUSTRALIA. (TR90-115)

"Analysis of a fault-tolerant scheme for processor ensembles", Dr. S. J. Upadhyaya □ 129 Bell Hall □ Dept. of Electrical & Computer Engineering □ State University of New York □ Buffalo, New York 14260 □ USA. (TR90-116)

"Optimal time for software release with respect to learning rate and testing effort", Dr. Florin Popentiu □ Assoc. of Scientists in Romania □ POBox 22-162, sect. 1 □ Bucharest 70148 □ ROMANIA. (TR90-117)

MANUSCRIPT RECEIVED *MANUSCRIPTS RECEIVED*

"A new reliability & availability model for distributed-control systems and telephone systems", Dr. Amr S. Badawi □ Electronics & Communications Dept. □ Cairo University □ Giza □ EGYPT. (TR90-118)

"Characterizations of a mixture of gamma distributions and of negative binomial distributions", A. Adatia □ Dept. of Mathematics & Statistics □ University of Regina □ Regina, Saskatchewan S4S 0A2 □ CANADA. (TR90-119)

"On the quasi-stationary distribution of the residual lifetime", Dr. S. Kalpakam □ Dept. of Mathematics □ Indian Institute of Technology □ Madras—600 036 □ INDIA. (TR90-120)

"Discrete reliability-growth models based on a learning-curve property", Dr. Arthur Fries □ IDA, Operational Evaluation Division □ 1801 North Beauregard Street □ Alexandria, Virginia 22311 □ USA. (TR90-121)

"Sample sizes for estimating the Weibull hazard function from censored samples", Dr. William Q. Meeker Jr. □ Dept. of Statistics □ Snedecor Hall □ Iowa State University □ Ames, Iowa 50011 □ USA. (TR90-122)

"Optimal design of large software systems considering reliability & cost", Noushin Ashrafi □ Dept. of Management Sciences □ University of Massachusetts □ Boston, Massachusetts 02125-3393 □ USA. (TR90-123)

"Comment on: An efficient non-recursive algorithm for computing the reliability of k-out-of-n systems", Dr. Ali M. Rushdi, Professor □ Dept. of Electrical & Computer Engineering □ King Abdul Aziz University □ POBox 9027 □ Jeddah 21413 □ Kingdom of SAUDI ARABIA. (TR90-124)

"A reliable k-out-of-n:G system and its applications", Dr. Zeng-Qi Yao □ Institute of Automation □ Chinese Academy of Sciences □ POBox 2728 □ Beijing—100 080 □ Peop. Rep. CHINA. (TR90-126)

"Correlation-based evaluation of behavior-level fault models of digital designs", Dr. Sumit Ghosh □ LEMS, Box D □ Division of Engineering □ Brown University □ Providence, Rhode Island 02912 □ USA. (TR90-128)