

May 1991

UILU-ENG-91-2236
CRHC-91-25

Center for Reliable and High-Performance Computing

IN-61-CR

71439

p-43

MEASUREMENT-BASED RELIABILITY PREDICTION METHODOLOGY

Linda Shen Lin

(NASA-CR-189913) MEASUREMENT-BASED
RELIABILITY PREDICTION METHODOLOGY M.S.
Thesis (Illinois Univ.) 43 p CACL 09B

N92-19842

Unclas
G3/61 0071439

*Coordinated Science Laboratory
College of Engineering*

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS None													
a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited													
b. DECLASSIFICATION/DOWNGRADING SCHEDULE															
PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-91-2236 CRHC-91- 25		5. MONITORING ORGANIZATION REPORT NUMBER(S)													
a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois	6b. OFFICE SYMBOL <i>(if applicable)</i> N/A	7a. NAME OF MONITORING ORGANIZATION JSEP and NASA													
ADDRESS (City, State, and ZIP Code) 1101 W. Springfield Ave. Urbana, IL 61801		7b. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217 and NASA Langley Research Center, Hampton, VA 23665													
NAME OF FUNDING/SPONSORING ORGANIZATION JSEP and NASA	8b. OFFICE SYMBOL <i>(if applicable)</i>	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-90-J-1270 and NASA NAG-1-613													
ADDRESS (City, State, and ZIP Code) Arlington, VA 22217 and NASA Langley Research Center, Hampton, VA 23665		10. SOURCE OF FUNDING NUMBERS <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <td style="width: 25%;">PROGRAM ELEMENT NO.</td> <td style="width: 25%;">PROJECT NO.</td> <td style="width: 25%;">TASK NO.</td> <td style="width: 25%;">WORK UNIT ACCESSION NO.</td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </table>		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.								
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.												
1. TITLE (Include Security Classification) Measurement-Based Reliability Prediction Methodology															
2. PERSONAL AUTHOR(S) Linda Shen Lin															
a. TYPE OF REPORT Technical	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) July 1991	15. PAGE COUNT 35												
3. SUPPLEMENTARY NOTATION															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">COSATI CODES</th> </tr> <tr> <th style="width: 33%;">FIELD</th> <th style="width: 33%;">GROUP</th> <th style="width: 33%;">SUB-GROUP</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		COSATI CODES			FIELD	GROUP	SUB-GROUP							18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) measurement-based, workload dependency, statistical, regression, reliability prediction, performance evaluation, dependability	
COSATI CODES															
FIELD	GROUP	SUB-GROUP													
4. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>In the past, analytical and measurement-based models have been developed to characterize computer system behavior. An open issue has been how these models can be used, if at all, for system design improvement. This thesis attempts to address this issue. It proposes a combined statistical/analytical approach to use measurements from one environment to model the system failure behavior in a new environment. A comparison of the predicted results with the actual data from the new environment shows a close correspondence.</p>															
DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified													
5. NAME OF RESPONSIBLE INDIVIDUAL		22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL												

MEASUREMENT-BASED RELIABILITY PREDICTION METHODOLOGY

BY

LINDA SHEN LIN

B.S., University of Virginia, 1989

THESIS

**Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1991**

Urbana, Illinois

ABSTRACT

In the past, analytical and measurement-based models have been developed to characterize computer system behavior. An open issue has been how these models can be used, if at all, for system design improvement. This thesis attempts to address this issue. It proposes a combined statistical/analytical approach to use measurements from one environment to model the system failure behavior in a new environment. A comparison of the predicted results with the actual data from the new environment shows a close correspondence.

ACKNOWLEDGEMENTS

I thank my advisor, Professor Ravi Iyer, whose guidance and encouragement made this thesis possible. I thank my fiancé, John Meckley, whose love and support mean so much in everything I do. I thank every friend at the Center for Reliable and High-performance Computing and the Coordinated Science Laboratory whose support and friendship made my years at the University a memorable experience, especially Dr. Rene Llames for his valuable suggestions and help in the drafting of this thesis.

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
1.1 Motivation	1
1.2 Related Research	2
1.3 Overview	3
2. MODELING AND PREDICTION METHODOLOGY	5
2.1 Measured Data	6
2.2 Workload/Failure Model	7
2.3 Parameters of Semi-Markov Models	12
2.4 Development of Prediction Methodology	14
2.5 Prediction of the Resource-Usage/Error/Recovery Model	19
2.6 Results of the Prediction Methodology	21
3. A SECOND EXAMPLE	24
3.1 Data	24
3.2 Modeling	25
3.3 Regression	26
3.4 Prediction	27
3.5 Discussion	27
4. CONCLUSIONS	32
REFERENCES	34

LIST OF TABLES

Table	Page
2.1: IBM3081 Data Intervals	7
2.2: An Example of CPU Bound Clustering	9
2.3: Transition Probabilities from Resource-Usage States to Error States for Figure 2.2	12
2.4: Transition Probabilities	23
2.5: Mean Holding Time	23
3.1: IBM370 Data Intervals	26
3.2: Expected vs. Predicted p_{ij} for $i =$ Resource-Usage States and $j =$ Error States for the Sixth Interval	29
3.3: Expected vs. Predicted τ_{ij} for $i =$ Resource-Usage States and $j =$ Error States for the Sixth Interval	30

LIST OF FIGURES

Figure	Page
2.1: Resource-Usage State Transition Diagram Corresponding to Table 2.2	10
2.2: Resource-Usage/Error/Recovery State Transition Diagram Corresponding to Figure 2.1	11
2.3: Regression Models for Software Error	15
2.4: Predicted Resource-Usage/Error/Recovery Model for the Fourth Interval	22
2.5: Actual Resource-Usage/Error/Recovery Model for the Fourth Interval	22
3.1: Limiting Destination Probability from Normal Resource-Usage States to Error States	28

1. INTRODUCTION

1.1 Motivation

The evaluation of computer system performability is an important research issue. The need for both higher performance and dependability motivates the development of accurate and powerful models to aid in system design, tuning, and reliability evaluation. This thesis is concerned with developing measurement-based performability models and the use of such models for predicting system behavior under new and yet unmeasured conditions.

Previous research [1], [2] has shown that system failure rate is dependent on resource usage and that increased resource usage is accompanied by increased failure. Physically, this dependence can be caused by several factors: Increased usage can result in higher probability of detecting faults as more of the system is exercised. Increased usage can also result in more stress on the hardware, in the form of higher levels of electronic noise,

temperature, and device and mechanical stress. Finally, increased usage can result in increased likelihood of human error which can lead to system failure.

The dependence of reliability on resource usage suggests that error/failure behavior may be predicted. By observing the resource usage behavior and failure occurrences of a system over some period of time, the interaction between workload and reliability can be characterized. Using this characterization, the failure behavior of a new usage environment may be predicted, given knowledge of the workload characteristics of that environment. This thesis characterizes the relationship between usage and failure by building measurement-based state-transition models, and presents a method for using these models to evaluate the reliability of the system under specified usage characteristics.

1.2 Related Research

Past studies have focused heavily on developing analytical models for system failure [3], [4], [5], [6]. The approach has been to assume a distribution for the time to failure of various system components. Whether the modeling technique is combinatorial or Markov, exponential distributions have been typically assumed because of the tractability of the resulting models. However, this assumption is in general not supported by real data.

On the other hand, field measurements have been used in several studies. The consensus is that system failure is resource-usage dependent. Early studies based on real measurements [1], [2], [7], [8] show that the operational environment of a system is an

important factor in predicting its reliability. Increased system failure rates due to increased utilization have been documented and modeled [2], [9]. Results in [10] indicate that CPU-related failures increase exponentially with resource usage after the system utilization reaches a saturation point. A fault prediction method based on failure patterns in error log files has been explored in [11]. Performability models which combine analytical modeling and measurements have been developed [12]. The study in [13] describes a diagnostic methodology for detecting anomalous behaviors of a network environment. A failure prediction method based on intermittent error characteristics has been investigated in [14]. Studies based on real data not only provide accurate quantification of system dependability but also reflect dynamic changes in system behavior. Several analytical models that take into account resource-usage effects have recently been proposed [15], [16].

This thesis constructs performability models based on real data as in [12] but goes further by developing a methodology to predict reliability using the performability models.

1.3 Overview

Based on the assumption that system failure is resource-usage dependent and given that resource-usage information is available, failure characteristics should be predictable.

This thesis investigates this predictability. Based on measurements of the resource-usage/error/failure behavior of a system, we attempt to predict the failure behavior of a new environment for which only the resource-usage information is known.

In particular, resource usage and error-related activities are first recorded over a period of time. Using the recorded data, a state-transition mode is constructed to represent resource-usage behavior. This model is then extended to include the recorded error/failure behavior. The relationship between the usage-only model and the full usage/error model is characterized empirically by regressing the parameters which describe error/failure rates on the resource usage indices (e.g., CPU utilization). Reliability under a new usage environment is then predicted by using the regression relations to estimate the error/failure rates corresponding to the new usage environment. In effect, the usage model of the new environment is extended to the full resource-usage/error/recovery model through regression-based estimation.

The remainder of the thesis is organized as follows. Chapter 2 discusses the modeling and prediction method in detail. Chapter 3 validates the method by applying it to an independent data set. Chapter 4 summarizes the method, offers some concluding remarks, and explores possibilities for future work.

2. MODELING AND PREDICTION METHODOLOGY

The prediction method is developed on data collected from an IBM 3081 system, but it does not use any information that is unique to the particular IBM system. Therefore, the approach is system-independent, and in principle can be applied to other systems as well. In essence, the method consists of four steps:

1. Collect resource-usage and error/failure/recovery data on a system for different periods of time.
2. Using the measured data, construct the system resource-usage model and its corresponding resource-usage/error/recovery model for each period of time.
3. The models identified above are then used to derive, via nonlinear regression, empirical relationships between resource-usage indices (e.g., CPU utilization) model parameters as well as other necessary auxiliary relationships.

4. The derived regression models are used to extrapolate the system failure behavior of a new environment by predicting the resource-usage/error/recovery model that corresponds to the resource-usage model of the new environment.

2.1 Measured Data

The data used to develop the proposed methodology are collected from an IBM 3081 dual processor and channel system running under the MVS operating system over a period of about three months. The resource usage data are recorded by the IBM MVS/370 system Resource Management Facility. The sampling time is 0.5 seconds. Every hour, the average values are computed for each index and stored to depict resource usage of the particular hour. The resource-usage indices that describe the state of the system include, CPU utilization (the percent of time that the CPU is executing instructions), channel busy (the fraction of time that the channel is busy and the CPU is waiting, which reflects the memory contention), I/O usage (the number of successful Start I/O and Resume I/O instructions issued to the channel), and disk usage (the number of requests serviced on the direct access storage devices).

The error and recovery data are logged by the operating system. At every occurrence of an error, the operating system records the time, description, system status, and recovery attempts associated with the error. Because the manner in which errors are detected and reported in a system, a single fault may manifest itself as more than one error, depending on the activity at the time of the error. To address this problem, errors occur in

Table 2.1: IBM3081 Data Intervals

interval	span	no. of obs.	no. of err.
1	1/7/85 - 1/18/85	101	151
2	1/28/85 - 2/4/85	100	187
3	2/4/85 - 2/22/85	102	204
4	2/22/85 - 3/15/85	100	184

close succession (within five minutes) are merged together [12]. The resulting error data are classified as CPU-related errors, channel errors, software errors, direct access storage device errors, and multiple errors. Multiple errors are identified for instances in which different types of error occur in close succession, but due to a common cause.

The data are divided into four temporal intervals. Models are constructed for each interval. Models from the first three intervals are used in the regression. The last interval is reserved for verification of prediction results. The data set, divided into intervals, is shown in Table 2.1.

2.2 Workload/Failure Model

Having divided the available data, the next step is to use the data to identify state transition models for each interval to quantify system characteristics. This is similar to the analysis performed in [12]. First, the normal resource-usage model is identified. The set of resource-usage data depicts the state of the system at each time interval in n-dimensional space; each dimension describes one aspect of the system status (CPU utilization can be one of the dimensions). The set of data may potentially have an infinite

number of n -dimensional vectors. To make the problem manageable, cluster analysis is used to summarize the data. The n -dimensional vectors are partitioned into a small number of groups or clusters suggested by the data. The vectors in the same cluster tend to be similar in various dimensions; those in different clusters tend to be dissimilar.

A k -means clustering algorithm is used in this study. The k -means algorithm minimizes the sum of the squares of the Euclidean distances between the members of each cluster and the centroid in the dimensions of the clustering variable and, at the same time, maximizes the intercluster centroid distances. In essence, observations that are spatially close in the dimension(s) of the clustering variable(s) are grouped into the same clusters and represented by the centroids of the clusters which are determined by the means of the members of the cluster.

In this study, the effect of CPU utilization on failure is of interest. The two dimensions used for cluster analysis are CPU utilization and channel busy. Hence, those observations in the same cluster have similar CPU utilization and channel busy characterizations. Channel busy is used in addition to CPU utilization because channel busy is inversely related to CPU utilization and has the effect of improving the clustered partitions. Table 2.2 shows the result of the k -means algorithm applied on the first interval. The data are grouped into four clusters; the numbers of observations that belong to each cluster are 9, 5, 25, and 61. The centroids are defined by the means of the clustering variables (CPU utilization and channel busy) over the observations in the cluster. For

Table 2.2: An Example of CPU Bound Clustering

Cluster	No. of observations	CPU utilization	Channel busy
1	9	0.233	0.133
2	5	0.244	0.371
3	25	0.667	0.099
4	61	0.961	0.114

example, Cluster 1 is defined by a CPU utilization of 0.233 and channel busy of 0.133, the mean CPU utilization and channel busy over the 9 observations in that cluster.

After the resource usage data are clustered, each cluster is used to depict a system state, and a state transition model is constructed. Each state is defined by the centroid of the cluster in terms of the clustering variables. A state transition model is defined by interstate transition probability and holding time distribution. The interstate transition probability p_{ij} between any two states is defined to be

$$p_{ij} = \frac{\text{observed number of transitions from state } i \text{ to state } j}{\text{observed number of transitions from state } i}$$

Observe that consecutive observations belonging to the same state are not defined as transitions. In addition, a "nonmeasured" state is defined to represent time intervals for which measurements have not been recorded. Figure 2.1 is an example of a state transition diagram. The arrows originating from states 3 and 4, but not terminating on any of the other states, indicate transitions to the "nonmeasured" state. Similarly, the fact that state 2 is not entered from any of the other states shown, indicates that it is entered from the "nonmeasured" state. The holding time between any two states, i.e., the time the process remains in a state before it makes a transition to another state, is

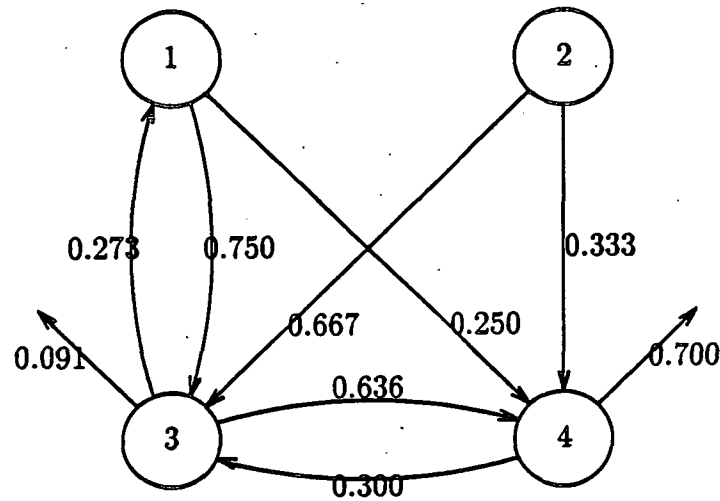


Figure 2.1: Resource-Usage State Transition Diagram Corresponding to Table 2.2

defined by the distribution $h_{ij}(t)$. Thus, the state transition model of resource-usage is generated. The model can give insights to other important parameters of the system.

In a similar fashion, the measured error events are classified into five different categories: CPU error, software error, channel error, DASD error, and multiple error, as suggested by the data. The recovery procedures are also divided into categories based on recovery cost which is measured by the system overhead required to handle an error. At the lowest cost level is the hardware recovery which uses an error correction code or hardware instruction retry. At the second level is the software recovery which involves software-initiated recovery. At the highest level, no recovery is possible; the system has to be brought down for repair.

As a final step in model construction, the normal resource-usage, error, and recovery models are combined into one unified model. Figure 2.2 shows an example of the resource-usage/error/recovery model that corresponds to the resource-usage model in Figure 2.1.

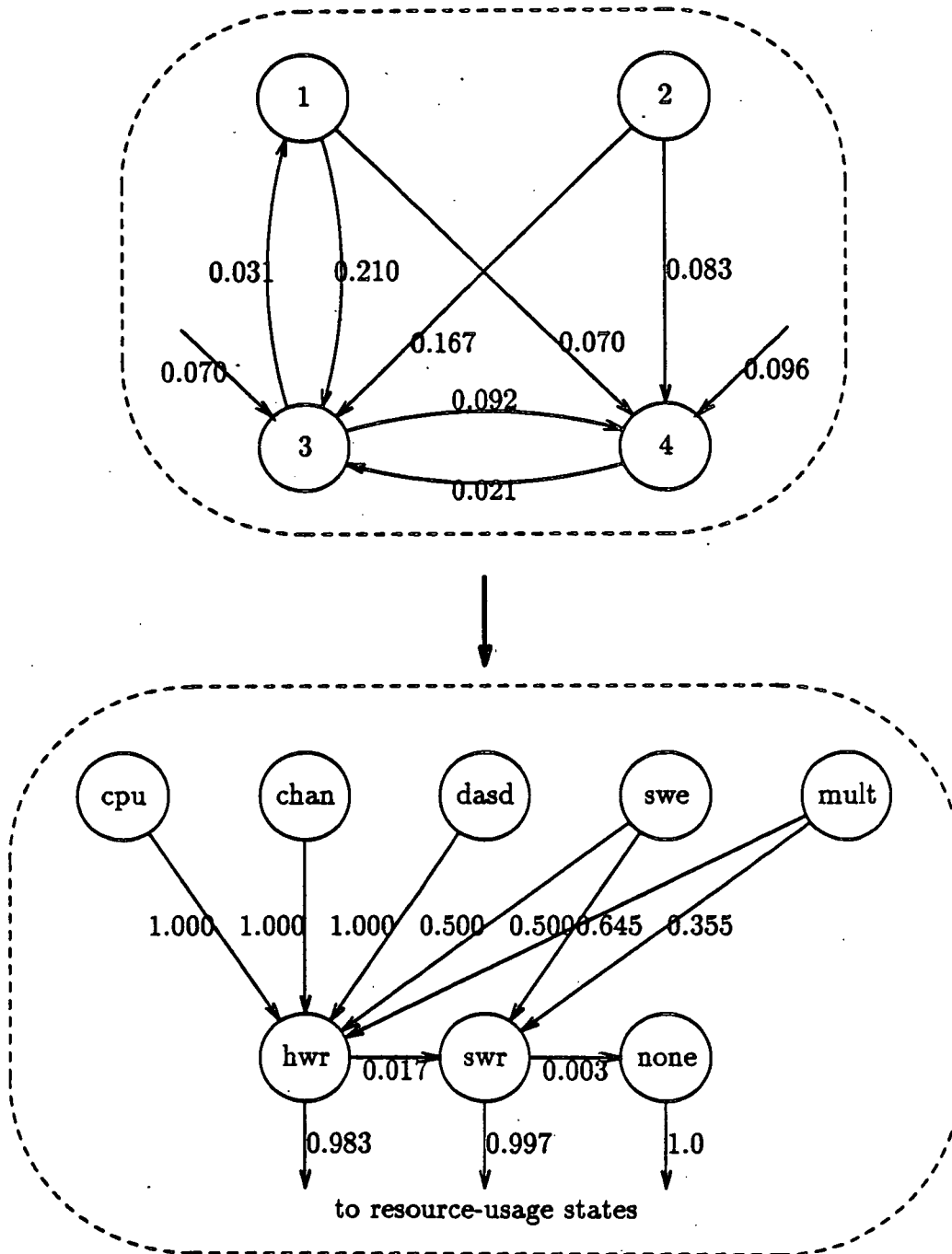


Figure 2.2: Resource-Usage/Error/Recovery State Transition Diagram Corresponding to Figure 2.1

Table 2.3: Transition Probabilities from Resource-Usage States to Error States for Figure 2.2

State	CPU utilization	CPU	Channel	DASD	Software	Multiple
1	0.233	0	0	0.140	0.290	0.210
2	0.245	0	0	0.500	0.167	0.083
3	0.667	0	0.046	0.538	0.123	0.046
4	0.961	0	0.011	0.543	0.202	0.128

To preserve the clarity of the figure, the transition probabilities from normal workload states to error states are not shown, but instead are listed in Table 2.3.

It is very important to note that a resource-usage model contains only normal resource-usage states and a resource-usage/error/recovery model contains error and/or recovery states in addition to the normal resource-usage states.

2.3 Parameters of Semi-Markov Models

The state transition models are semi-Markov and are defined by the transition probability and holding time distribution associated with each transition. In this study, the mean holding time is of interest. When p_{ij} and τ_{ij} are defined for every transition, a semi-Markov model is identified. Formally, p_{ij} is the probability that a semi-Markov process that entered state i on its last transition will enter state j on its next transition, and $\tau_{ij} = \int_0^{\infty} t h_{ij}(t) dt$ is the mean holding time for the transition $i \rightarrow j$, the time the process will spend in state i before making a transition to state j .

Next, there is a series of other interesting parameters that can be defined [17]. Their relevance will become obvious in the next section. The mean waiting time, τ_i , for state

i , defined as

$$\tau_i = \sum_{j=1}^n p_{ij} \tau_{ij} \quad (2.1)$$

is the time that the process will spend in state i before making a transition. In essence, a waiting time is merely a holding time that is unconditional on the destination state.

The limiting destination probability, γ_{ij} , defined as

$$\gamma_{ij} = e_i p_{ij} \tau_{ij} \quad (2.2)$$

is the probability that at a time instant in the steady state, the process is in state i and planning to make its next transition to state j .

The limiting entrance probability, e_i , defined as

$$e_i = \frac{\pi_i}{\tau} \quad (2.3)$$

is the probability of the process entering state i at any time in the steady state, independent of the starting state.

The mean time between transitions, τ , defined as

$$\tau = \sum_{i=1}^n \pi_i \tau_i \quad (2.4)$$

is, in essence, a waiting time that is unconditional on the starting state.

The limiting state probability, π_i , for the imbedded Markov process described by the transition probability from matrix P , is defined as

$$\pi = \pi P \quad (2.5)$$

or

$$\pi_j = \sum_{i=1}^n \pi_i p_{ij} \quad (2.6)$$

Equation (2.6) along with

$$\sum_{j=1}^n \pi_j = 1 \quad (2.7)$$

gives the unique nonnegative solution of π_j .

In this study, only the transitions with $i =$ normal resource-usage state and $j =$ error state are of interest. Unless otherwise specified, for every model parameter, i implies normal resource-usage states and j implies error states.

2.4 Development of Prediction Methodology

Under the premise that system reliability is a decreasing function of system activity, empirical relationships between system resource usage variables and model parameters are sought in order to generalize the effect of resource usage on system error rate.

One of the promising model parameters used in this study is the limiting destination probability γ_{ij} . For each error state, the limiting destination probabilities from normal resource-usage states to the error states are plotted as a function of CPU utilization of the normal resource-usage states, using models from the first three intervals. As an example, Figure 2.3(a) shows the limiting destination probability γ_{ij} to software errors as a function of CPU utilization of all the normal resource-usage states from the first three intervals. Using nonlinear regression, an exponential function is fitted to this data. It is apparent that the limiting destination probability to the software error state is an

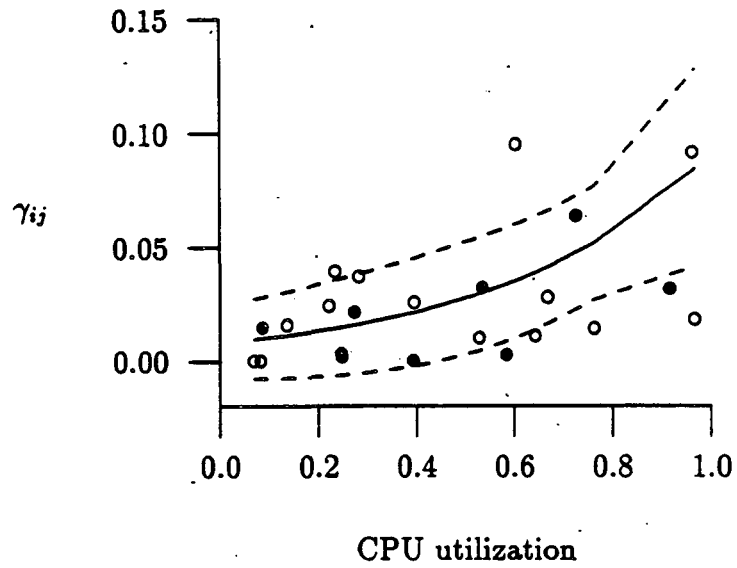
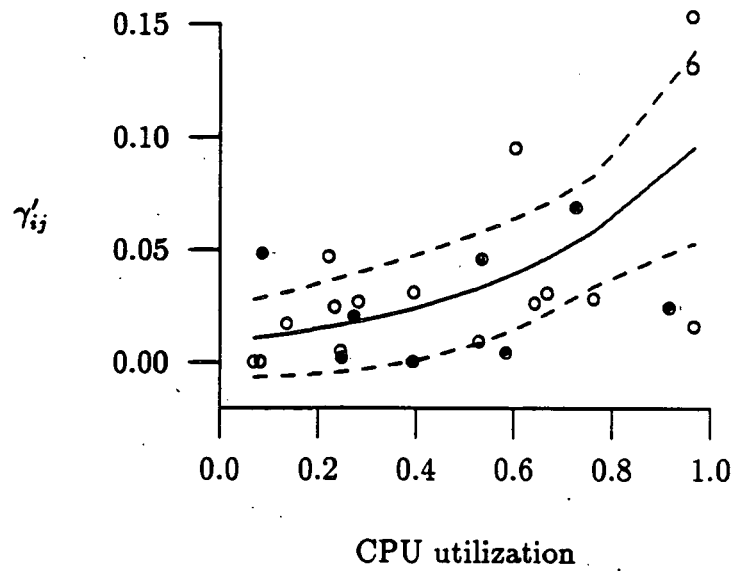
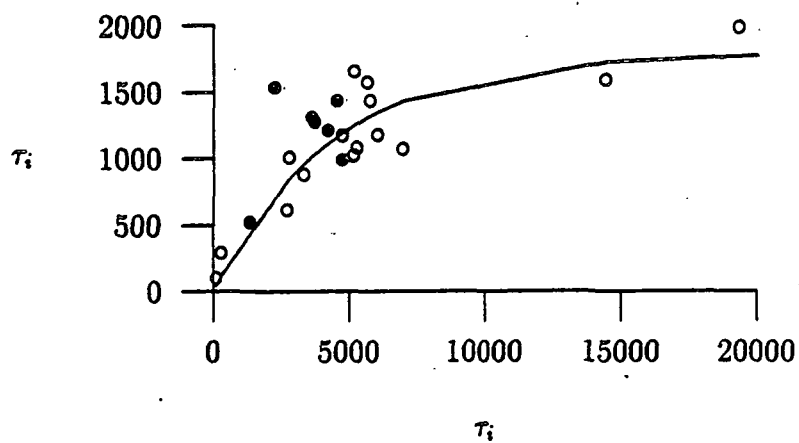
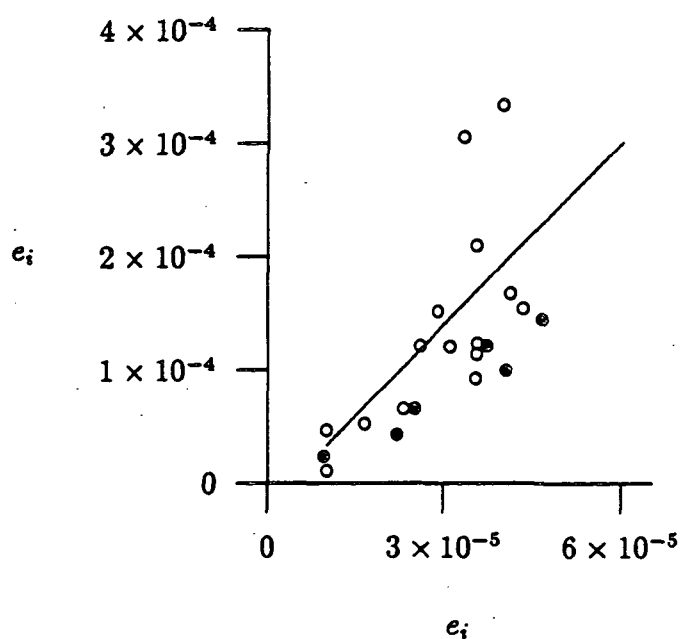
(a) γ_{ij} (b) γ'_{ij}

Figure 2.3: Regression Models for Software Error

(c) τ_i' (d) e_i'

- i =normal workload state
- j =software error state
- old environment (first three intervals)
- new environment (fourth interval)
- regression line
- - 95% confidence limits

Figure 2.3: (Continued)

increasing function of CPU utilization. The predicted values and the 95% confidence limits are shown in the figure. To determine whether models based on the first three intervals could accurately predict the error behavior in the fourth interval, the data points for the fourth interval are also shown in the same plot. The majority of the data from the fourth interval falls within the 95% confidence limits. Such findings are also true with destination probabilities to other types of error vs. CPU utilization functions. This indicates that models based on the measurements from the first three intervals can be used to predict the behavior in the fourth interval with reasonable accuracy; i.e., it is clear that the behavior of a new environment can be predicted by system operational information from old environments.

A closer look at γ_{ij} indicates the necessity of solving the equation

$$f(x, y)xy = b \quad (2.8)$$

Clearly, there is a lack of equations for a definite solution for x and y .

The answer is to make assumptions about two of the terms. Attempts are made to establish regression relations between p_{ij} and CPU utilization and between τ_{ij} and CPU utilization. Neither demonstrates strong uniform relations. Thus, a new model variable γ'_{ij} is defined as

$$\gamma'_{ij} = \hat{e}_i p_{ij} \hat{\tau}_i \quad (2.9)$$

where \hat{e}_i and $\hat{\tau}_i$ are the projected limiting entrance probability and the mean waiting time of the new resource-usage/error/recovery model, respectively. The projection is done by plotting e_i as a function of e'_i (Figure 2.3(d)) and τ_i as a function of τ'_i (Figure 2.3(c)),

where e_i and τ_i are the limiting entrance rate and mean waiting time, respectively, of the normal resource-usage states in the resource-usage/error/recovery models; e'_i and τ'_i are the limiting entrance rate and the mean waiting time of the normal resource-usage states in the resource-usage models. The new variable, γ'_{ij} , plotted as a function of CPU utilization exhibits similar characteristics as γ_{ij} , as shown in Figure 2.3(b). By introducing the additional regression models, the prediction of transitions to error states in the new resource-usage/error/recovery model becomes possible; the transitions are in turn defined by the transition probability p_{ij} and holding time τ_{ij} where i =normal resource-usage states and j =error states.

Before the prediction procedures are introduced, the regression construction process is outlined as follows.

1. Collect resource-usage and fault data for different intervals of time.
2. Identify resource-usage indices, and construct the resource-usage model and the resource-usage/error/recovery model for each interval.
3. Calculate the assortment of model parameters for each resource-usage model and resource-usage/error/recovery model.
4. Build the γ_{ij} regression models by plotting γ_{ij} as functions of resource-usage for each error state (j =error state).
5. Explore whether the p_{ij} 's to each error state are increasing functions of the resource-usage index.

6. Build the e_i regression model by plotting e_i vs. e'_i . The predicted line represents \hat{e}_i , which is the projected limiting entrance probability of a normal resource-usage state in the predicted resource-usage/error/recovery model.
7. If the p_{ij} 's are not increasing functions of the resource-usage index, build the τ_i regression model by plotting τ_i vs. τ'_i . The regression line represents $\hat{\tau}_i$, which is the projected mean waiting time of a normal resource-usage state in the predicted resource-usage/error/recovery model. Then, calculate γ'_{ij} and establish γ_{ij} regression models.

2.5 Prediction of the Resource-Usage/Error/Recovery Model

Using the regression models that developed above, and given the resource-usage information of a new environment, the new resource-usage/error/recovery model can be derived. The procedure to derive the new model is outlined below.

1. Construct the resource-usage model for the new environment.
2. Derive the p_{ij} 's using the γ'_{ij} or p_{ij} model.
3. Derive the τ_{ij} 's using the γ_{ij} model.
4. Normalize the p_{ij} 's such that

$$\sum_j p_{ij} = 1$$

5. Set $p_{ij} = \pi'_j$ where i =recovery state and j =normal workload states.

6. Let $\tau_{ij} = \bar{\tau}_i$ where i =recovery state and j =normal workload states.

First, a state transition model (similar to Figure 2.1) is formed, based on the resource-usage information. To calculate the predicted values of p_{ij} from the γ' regression, the values e'_i and τ'_i of the resource-usage model have to be derived. The projected \hat{e}_i and $\hat{\tau}_i$ are then extrapolated from the e_i and τ_i regression models. For example, a state in the new model with a CPU utilization of 0.571 is used; the associated e'_i and τ'_i are 0.0000342 and 29223.1. The γ'_{ij} 's from the resource-usage states to each error state are extrapolated from the γ' regression models (Figure 2.3(b)) with projected values of \hat{e}_i and $\hat{\tau}_i$. From Figure 2.3(b), the γ'_{ij} to software error is 0.0358. The values of p_{ij} to each error state are calculated from Equation (2.9) using the values for γ'_{ij} , \hat{e}_{ij} , and $\hat{\tau}_i$. For the state in the example, the p_{ij} to software error becomes 0.393.

Next, to derive the τ_{ij} 's of the new resource-usage/error/recovery model, the γ_{ij} 's from the resource-usage states to each error state are extrapolated from the γ_{ij} regression model (Figure 2.3(a)), and found to be 0.0324 for the state in the example. The values of p_{ij} and \hat{e}_i which have just been derived from the previous two regression models, are then applied to Equation (2.2) to determine τ_{ij} . The value of τ_{ij} to software error for the state in the example is 1619.7 seconds.

Note that the predicted p_{ij} 's from resource-usage states to error states must be normalized so that they sum to one. After normalization, p_{ij} for the state in the above example is 0.2704.

For the transitions from the recovery and failure states to normal resource-usage states, the state probability π'_i of the resource-usage model best estimates the transition probabilities, and the mean holding time is estimated by the average of the mean waiting time of each recovery states in the first three intervals.

The prediction of the transition probabilities and mean holding times from normal resource-usage states to error and from recovery states to normal resource-usage completes the new resource-usage/error/recovery model.

2.6 Results of the Prediction Methodology

Using the data from the first three intervals, the resource-usage/error/recovery model for the fourth interval is obtained. We compare the *predicted* transition probabilities and holding times of the transitions from the normal resource-usage states to the error states, with the ones *measured* for the fourth interval.

For simplicity, the resource-usage model for the fourth interval is modeled as a one-state model. Using the prediction method developed in the previous sections, the failure behavior in terms of state transition probabilities and mean holding times from the normal resource-usage state to error states is determined. The predicted resource-usage/error/failure model is shown in Figure 2.4. Figure 2.5 contains the resource-usage/error/failure model constructed by the actual failure data from the fourth interval.

Tables 2.4 and 2.5 list the predicted and actual transition probabilities and mean holding times from the normal resource-usage state to error states. To quantify the

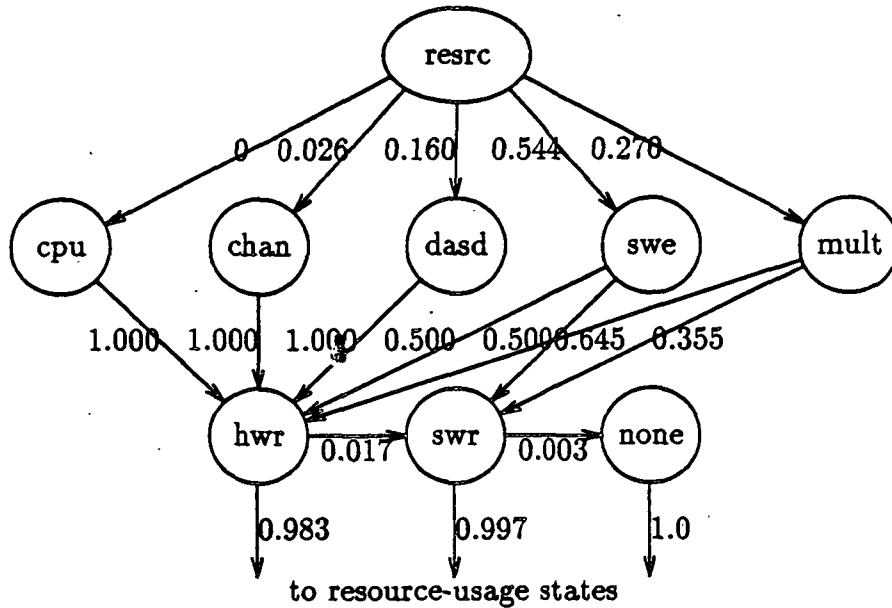


Figure 2.4: Predicted Resource-Usage/Error/Recovery Model for the Fourth Interval

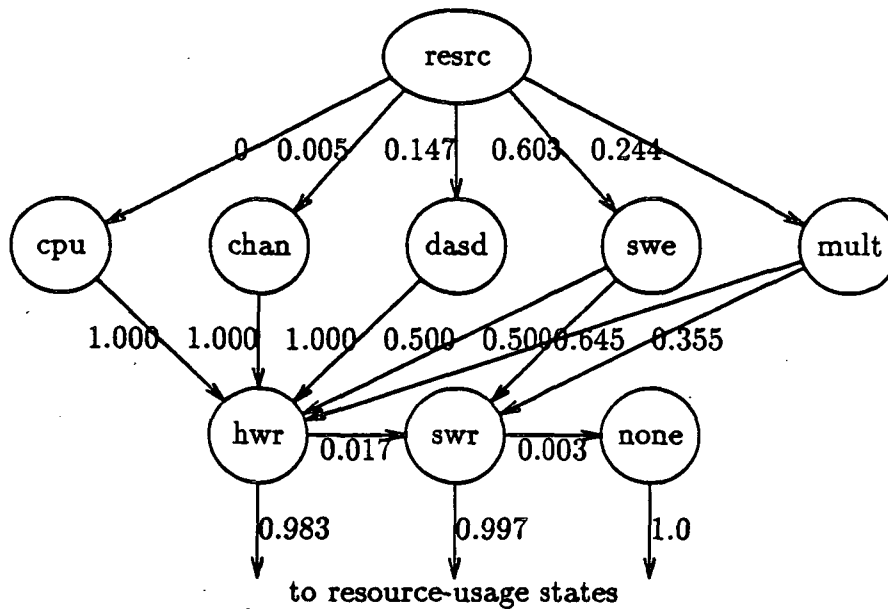


Figure 2.5: Actual Resource-Usage/Error/Recovery Model for the Fourth Interval

Table 2.4: Transition Probabilities

dest.	no. obs	expected	predicted	diff.	90%	99%
Channel	1	0.00543	0.02578	0.02035	0.00892	0.01397
Mult	27	0.14674	0.15999	0.01325	0.02609	0.04086
Disk	111	0.60326	0.54388	-0.05938	0.05933	0.09091
Software	45	0.24457	0.27036	0.02579	0.05213	0.8163

Table 2.5: Mean Holding Time

dest.	no. obs	expected	predicted	diff.	90%	99%
Channel	1	2210.68	1164.77	-1045.91	.	.
Mult	27	1550.90	1430.19	-120.71	517.75	810.77
Disk	111	1172.86	1421.95	249.09	176.04	275.67
Software	45	1708.40	1691.69	-16.71	547.70	857.67

goodness of the prediction method, consider the confidence intervals for the expected p_{ij} 's and τ_{ij} 's. Since the expected p_{ij} 's and τ_{ij} 's are estimates, there are confidence intervals associated with them. When the absolute difference between the predicted and expected is less than the $x\%$ confidence interval, the predicted value is said to fall within $x\%$. For example, for the transition probability to disk error, the absolute difference is 0.05938 is slightly greater than 0.05933, but less than 0.09091. Thus, the predicted value is close to the 90% interval. For the most part, the predicted values fall within the 90% interval. The large difference for channel error is due to the fact that the actual number of transitions to channel error is statistically insignificant.

3. A SECOND EXAMPLE

In this chapter the prediction methodology is further illustrated by applying it to a multicomputer system configuration. The data set used to develop the method in the previous chapter is referred to as Data Set I, and the new data set is referred to as Data Set II.

3.1 Data

Data Set II contains a system resource-usage and failure log collected on two IBM 370/168 mainframes over a period of three years [10]. Again the resource usage is measured by the SMF (System Management Facility). The recorded indices include CPU load, batch memory requests and usage, batch I/O wait time and load, and batch paging in and out. For every hour, the average values for each index are computed to depict the system resource usage for that hour. The clustering algorithm is later applied on the hourly averages. Again, the dependence of system reliability on CPU usage is investigated. The recorded CPU-related indices are TOTCPU, a measure of the total CPU

usage as a fraction between 0 and 1, and BATCPU, a measure of the batch CPU usage as a fraction between 0 and 1. An additional index, SYSCPU, a measure of the interactive and system workload, is calculated as the difference between TOTCPU and BATCPU. The failure data, recorded by the built-in failure detection facilities, contain error occurrences and recovery attempts. Errors occurring within five minutes of each other are coalesced into one error to eliminate the many manifestations of a single fault. As a result, a new type of error is created to account for different types of errors occurring close in time. Thus, four types of errors are defined: cpu-related errors in the central processor and storage (MCH), channel-related errors in I/O channels and associated interfaces (CCH), other errors in addition to MCH and CCH (OTH), and different types of error occurring in close succession (MUL). The choice of six clusters represents a compromise between two conflicting requirements. For regression, it is desirable to have a significant number of intervals (greater than seven). From the clustering point of view, the number of observations in each cluster has to be substantial for meaningful clustering results. The decision of six intervals is based on trial-and-error. Table 3.1 shows Data Set II divided into six intervals.

3.2 Modeling

First, the resource-usage data from the first five intervals are clustered by BATCPU and SYSCPU. Data from each interval are clustered into five clusters. Each of the

Table 3.1: IBM370 Data Intervals

interval	span	no. of obs.	no. of err.
1	2/79 - 6/79	3600	498
2	7/79 - 12/79	4416	311
3	1/80 - 6/80	4368	313
4	7/80 - 12/80	4416	721
5	1/81 - 6/81	4366	730
6	7/81 - 12/81	4416	338

resource-usage models is therefore a five-state model, with each state varying in mean BATCPU and mean SYSCPU.

Next, the corresponding resource-usage/error/recovery models are constructed. To simplify the model, recovery states are combined with the error states. In other words, instead of making transitions to recovery states and then to resource-usage states, the error states make transitions directly to the resource-usage states. Each of the resource-usage/error/recovery models, therefore, has a total of nine states—four error/recovery states and five normal resource-usage states.

After all of the resource-usage and failure models are constructed, the necessary model parameters are calculated for regression purposes. Again, the parameters of interest are those with i = each resource-usage state and j = each error state.

3.3 Regression

As a first step in finding regression relations, the γ_{ij} 's from the first five intervals are plotted versus CPU usage for $j = \{\text{MCH, CCH, OTH, MUL}\}$ (not shown). It is observed

that, for $j = \{\text{MCH, CCH}\}$, γ_{ij} vs. TOTCPU is the most pronounced increasing function, and for $j = \{\text{OTH, MUL}\}$, γ_{ij} vs. SYSCPU is the most pronounced increasing function. Regression relations are then established. Examples are shown in (Figure 3.1).

To solve the $f(x,y)xy = b$ dilemma, p_{ij} vs. CPU usage relations are plotted and found to be more pronounced increasing functions of CPU usage than γ'_{ij} vs. CPU usage relations. Thus, p_{ij} vs. CPU usages regression relations are established for the four error states. In addition, the e'_{ij} regression relation is determined.

3.4 Prediction

To predict the resource-usage/error/recovery model for the last interval, the resource-usage model is constructed as a five-state model. Using the regression relations determined in the previous section, p_{ij} and τ_{ij} from the resource-usage states to the error states can be predicted. Transition probabilities p_{ij} 's are determined directly from the p_{ij} regression models, and mean holding times τ_{ij} 's are determined by the e'_{ij} and γ_{ij} regression relations along with the calculated p_{ij} 's. The resulting values for p_{ij} 's are shown in Table 3.2, and for τ_{ij} 's in Table 3.3.

3.5 Discussion

Observe Tables 3.2 and 3.3. For both the p_{ij} 's and τ_{ij} 's, most of the predicted values fall within the 90% confidence interval. All of the predicted values fall within the 99% confidence interval with very few exceptions. All of the exceptions are transitions of very

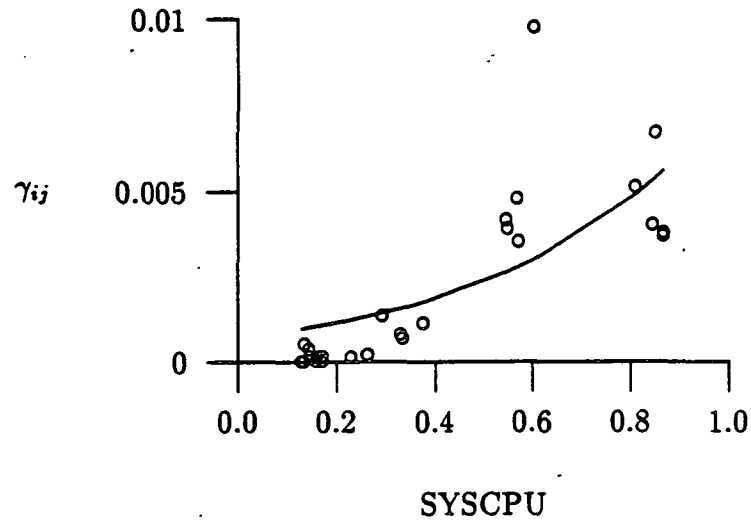
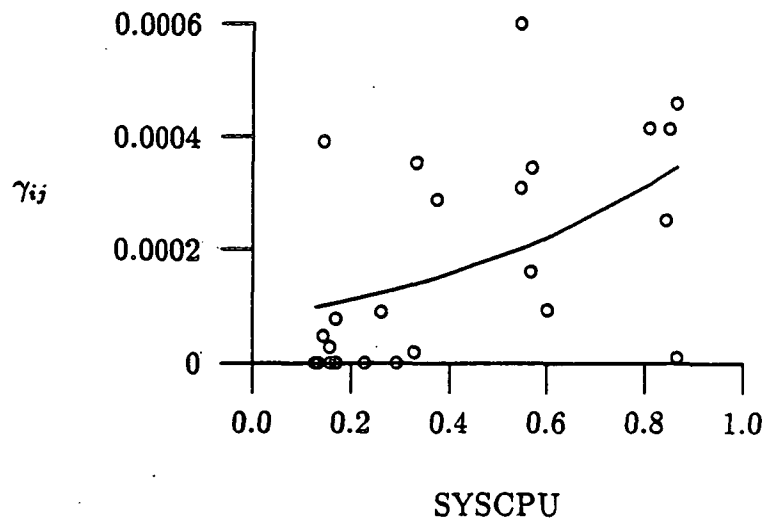
(a) γ_{ij} for $j = \text{OTH}$ (b) γ_{ij} for $j = \text{MUL}$

Figure 3.1: Limiting Destination Probability from Normal Resource-Usage States to Error States

Table 3.2: Expected vs. Predicted p_{ij} for i = Resource-Usage States and j = Error States for the Sixth Interval

j	i	obs.	exp. p_{ij}	pred. p_{ij}	diff.	90%	99%
MCH	1	13	0.0253	0.0136	-0.0118	0.0114	0.0179
	2	3	0.0110	0.0133	0.0023	0.0104	0.0163
	3	11	0.0266	0.0156	-0.0110	0.0130	0.0204
	4	1	0.0038	0.0189	0.0151	0.0062	0.0097
	5	5	0.0144	0.0072	-0.0072	0.0105	0.0165
CCH	1	35	0.0682	0.0724	0.0042	0.0183	0.0287
	2	29	0.1062	0.0633	-0.0430	0.0307	0.0480
	3	46	0.1114	0.0714	-0.0400	0.0255	0.0399
	4	24	0.0906	0.0797	-0.0108	0.0290	0.0454
	5	22	0.0634	0.0606	-0.0029	0.0215	0.0337
OTH	1	14	0.0273	0.0427	0.0154	0.0118	0.0185
	2	50	0.1832	0.1897	0.0065	0.0385	0.0603
	3	47	0.1138	0.1050	-0.0089	0.0257	0.0403
	4	3	0.0113	0.0347	0.0234	0.0107	0.0167
	5	23	0.0663	0.0439	-0.0224	0.0220	0.0344
MUL	1	4	0.0078	0.0041	-0.0037	0.0064	0.0100
	2	3	0.0110	0.0134	0.0024	0.0104	0.0163
	3	4	0.0097	0.0084	-0.0013	0.0079	0.0124
	4	0	0.0000	0.0035			
	5	1	0.0029	0.0042	0.0014	0.0047	0.0074

Table 3.3: Expected vs. Predicted τ_{ij} for i = Resource-Usage States and j = Error States for the Sixth Interval

j	i	obs.	exp. τ_{ij}	pred. τ_{ij}	diff.	90%	99%
MCH	1	13	2736.4	3319.1	582.8	1775.5	2780.4
	2	3	2587.3	6456.2	3868.9	2816.0	4409.7
	3	11	4752.4	4471.9	-280.5	2628.2	4115.7
	4	1	373.0	5372.2	4999.2	.	.
	5	5	3862.6	5130.3	1267.7	5093.4	7976.0
CCH	1	35	4289.1	4360.0	70.9	1080.3	1691.7
	2	29	6629.9	9226.6	2596.7	2529.6	3961.2
	3	46	4938.6	6562.0	1623.5	2001.9	3134.9
	4	24	14985.6	8353.0	-6632.6	7968.2	12477.9
	5	22	8561.1	4846.4	-3714.7	3486.5	5459.7
OTH	1	1	2662.2	4344.2	1682.0	1526.9	2391.0
	2	50	7107.5	5807.6	-1300.0	2372.6	3715.4
	3	47	4251.8	4776.6	524.9	1256.4	1967.5
	4	3	1175.0	7711.6	6536.6	1637.2	2563.8
	5	23	5349.8	6152.4	802.6	3396.1	5318.1
MUL	1	4	1328.2	4271.6	2943.4	1216.6	1905.2
	2	3	3474.7	5534.8	2060.1	1964.5	3076.3
	3	4	11325.5	4611.8	-6713.7	9661.4	15129.3
	4	0	0.0	7611.8	7611.8	.	.
	5	1	980.0	6048.9	5068.9	.	.

few observations which again make the result statistically insignificant. For example, for both p_{ij} 's and τ_{ij} 's, the exceptions are ($i=4, j=MCH$), one observation, ($i=4, j=OTH$), three observations, and ($i=4, j=MUL$), zero observations.

The inaccuracies in prediction are explained by the long time span of an interval and by the low error rate. In general, the longer the time that a data set spans, the more variation is contained in the data set. Simply stated, exceptional phenomena are much more likely to occur over a period of three years than over three months. Low error rate makes the regression less statistically significant. Data Set I is a better candidate for the method, because Data Set I has an average of 323 errors per month while Data Set II has 83 errors per month. The inaccuracies occur when the transitions are few, as discussed in the previous two paragraphs. When errors are rare or infrequent, any statistical inferences made based on the few observations can be unsound. Predicting the rare and infrequent is almost impossible and probably unrewarding.

4. CONCLUSIONS

In the past, analytical and measurement-based models have been developed to characterize system behavior. An open issue has been how these models can be used, if at all, for system design improvement or system tuning. This thesis has attempted to address this issue. Past studies have shown the resource-usage dependency of system failure behavior. The current study shows that it is possible to predict the system error/failure rate when given resource-usage information. Previous fault prediction schemes have lacked the use of real measurements; with the proposed method, measurement-based modeling finds its place in fault prediction application.

This thesis has proposed a combined statistical/analytical approach to use measurements from existing environments to forecast the system failure behavior in a new environment. Using regression, the method makes generalizations about the system failure behavior from models extracted from the measurements of the existing environments. When the resource usage of a new environment is known, predictions can be made from the generalized system failure behavior. Comparisons of the predicted results with the

actual data from the new environment show a close correspondence for both data sets on which the method has been applied.

However, the method is somewhat data-sensitive, while mostly system-independent. Improved accuracy is possible when the errors are frequent. In the case of very low error rates, the method attempts the almost impossible — to predict accidental and infrequent events, and hence may not succeed.

During the course of the research, many fascinating ideas that have arisen which deserve future investigation. Some of these are

1. Other standardizing variables in addition to the steady-state destination probability can be explored.
2. The effect of other measured resource-usage variables can be studied.
3. Regression on multiple resource-usage indices can be explored.

REFERENCES

- [1] S. E. Butner and R. K. Iyer, "A statistical study of reliability and system load at slac," in *Proceedings of 10th International Symposium on Fault-Tolerant Computing*, 1980.
- [2] X. Castillo and D. P. Siewoerk, "A workload dependent software reliability prediction model," in *Proceedings of 12th International Symposium on Fault-Tolerant Computing*, pp. 279-285, June 1982.
- [3] K. S. Trivedi, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [4] J. F. Meyer, "Closed-form solutions of performability," *IEEE Transaction on Computer*, July 1982.
- [5] A. Goyal et al., "The system availability estimator," in *Proceedings of 16th International Symposium on Fault-Tolerant Computing*, 1986.
- [6] A. Goyal, S. S. Lavenberg, and K. S. Trivedi, "Probabilistic modeling of computer system availability," *Annals of Operations Research*, vol. 8, March 1987.
- [7] X. Castillo and D. P. Siewiorek, "A performance-reliability model for computing systems," in *Proceedings of 10th International Symposium on Fault-Tolerant Computing*, 1980.
- [8] R. K. Iyer, S. E. Butner, and E. J. McCluskey, "A statistical failure/load relationship: Results of a multicomputer study," *IEEE Transaction on Computers*, vol. C-31, no. 7, pp. 697-706, July 1982.
- [9] R. K. Iyer and D. J. Rossetti, "A statistical load dependency model for CPU errors at SLAC," in *Proceedings of the 12th International Symposium Fault-Tolerant Computing*, pp. 363-372, June 1982.
- [10] R. K. Iyer, D. J. Rossetti, and M. C. Hsueh, "Measurement and modeling of computer reliability as affected by system activity," *ACM Transactions on Computer Systems*, vol. 4, pp. 214-237, August 1986.

- [11] M. M. Tsao, *Trend Analysis and fault prediction*. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, 1983.
- [12] M. C. Hsueh, R. K. Iyer, and K. S. Trivedi, "Performability modeling based on real data: A case study," *IEEE Transaction on Computers*, April 1988.
- [13] R. A. Maxion, "Anomaly detection for diagnosis," in *Proceedings of 20th International Symposium on Fault-Tolerant Computing*, 1990.
- [14] T.-T. Lin and D. P. Siewiorek, "Error log analysis: statistical modeling and heuristic trend analysis," *IEEE Transaction on Reliability*, vol. 39, October 1990.
- [15] O. Schoen, "On a class of integrated performance/reliability models based on queuing networks," in *Proceedings of 16th International Symposium on Fault-Tolerant Computing*, pp. 214-237, July 1986.
- [16] B. E. Aupperle, J. F. Meyer, and L. Wei, "Evaluation of fault-tolerant systems with nonhomogeneous workloads," in *Proceedings of 19th International Symposium on Fault-Tolerant Computing*, pp. 159-166, June 1989.
- [17] R. A. Howard, *Dynamic Probability Systems*. New York: John Wiley & Sons, Inc., 1971.