Research Report

How Expert is Your Expert System for Performance Management?

Joseph L. Hellerstein

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, NY 10598

May 10, 1994

How Expert is Your Expert System for Performance Management?

Joseph L. Hellerstein

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, NY 10598

May 10, 1994

ABSTRACT: Diagnosing performance problems in computer systems is a complex task for which many expert systems have been developed. This paper studies the quality of the results produced by four techniques for diagnosing performance problems: threshold analysis, bottleneck analysis, what's-different analysis, and correlation analysis. All have strengths and weaknesses.

- Threshold analysis can be applied to variables with non-linear relationships such as the relationship between response times and utilizations. But threshold analysis requires extensive and carefully chosen reference data to characterize normal behavior.

- Bottleneck analysis does not require reference data. But bottleneck analysis does not identify what changed and is limited to variables with linear relationships such as between total CPU utilization and CPU utilization by individual users.

- What's-different analysis identifies variables that account for a change in performance. But what's-different analysis requires reference data (although not as much data as threshold analysis) and quantitative what's-different analysis assumes that there is a linear relationship between the variables being analyzed.

- Correlation analysis identifies what has changed without the use of reference data. But correlation analysis cannot be applied to constant-level performance

problems and is most effective if the relationship between measurement variables is linear.

Further issues arise when diagnostic techniques are employed in practice. For example, the above techniques were applied to a mainframe computer system that had a known performance problem; *none of the techniques produced the correct diagnosis.* These results suggest the need for a systematic study of diagnostic techniques. An approach to such a study is outlined in the conclusions of this paper.

1. Introduction

Computer performance evaluation (CPE) includes tasks such as capacity planning, performance management, and performance monitoring. Since these are time-consuming and knowledge-intensive activities, many expert system applications have been developed for their automation. However, to the best of our knowledge, no study has been undertaken to evaluate the quality of the results produced by these expert systems. This paper is a first attempt at such a study.

Herein the focus is on expert systems for performance management, in particular, diagnosing performance problems. Diagnosis is undertaken when a performance problem has been detected (e.g., a service level is violated); diagnosis characterizes a performance problem in terms of measurement variables so as to identify the appropriate corrective actions. For example, poor response time might be characterized by the CPU consumption of a specific user, which might lead to improving the efficiency of that user's application. A variety of expert systems have been developed. Examples include: [4], which makes recommendations on DASD tuning; [3], which analyzes MVS, CICS, and DB2 performance; and [6], which provides recommendations for MVS performance problems. Although many expert systems for diagnosing performance problems have been developed, we are unaware of any study that evaluates the techniques employed.

The remainder of this paper is organized as follows. Section 2 describes a framework for diagnosis that provides the basis for the comparisons presented in Section 3. Section 3 compares four commonly-used techniques for diagnosing performance problems: threshold analysis, bottleneck analysis, what's-different analysis, and correlation analysis. Section 4 contains concluding remarks and areas for future work.

2. Framework for Diagnosis

This section describes a framework for diagnosing performance problems in information systems.

The framework herein described is widely used by performance analysts. It starts with a service level indicator (e.g., response time), and then systematically examines measurement variables so as to identify the resources or workloads that caused the performance problem. To proceed from a service level indicator to the identification of an underlying cause, knowledge of relationships between measurement variables is employed. Sometimes this knowledge is expressed as rules. For example, Rule 1: If response time (R) is large, then examine the time spent in the CPU (R_{CPU}), IO (R_{IO}), paging (R_{PAGE}), and other (R_{OTHER}) subsystems

Rule 2: If R_{CPU} is large, then examine the CPU of individual users.

In general, decomposition (or "drill-downs") are used that are based on part-of relationships such as those indicated in rules 1 and 2. For example, applying Rule 1 might lead to determining that R_{CPU} should be examined more closely; this in turn would lead to applying Rule 2, which might result in determining that the problem was due to the CPU consumption of user A. The variables identified in this manner are called **causal** in that they explain the performance problem. In the preceding example, both R_{CPU} and the CPU utilization for user A are causal variables.

Describing diagnosis in terms of applying rules is sometimes cumbersome. An alternative representation is a **measurement navigation graph (MNG)** [2]. Figure 1 depicts a part of a MNG for analyzing response time problems in a computer system. The arcs from R to R_{CPU}, R_{IO}, R_{PAGE}, and R_{OTHER} indicate that the latter are sub-parts of R. Here R is called the **parent variable**; R_{CPU}, R_{IO}, R_{PAGE}, and R_{OTHER} are the **child variables**. It may be that some of the child variables can be decomposed as well, as indicated by the arcs emanating from R_{CPU}, R_{IO}, and R_{PAGE}. Alternatively, there may be a very limited structure to the measurement data (e.g., Unix sar data); in these cases, the MNG may consist of a two-level hierarchy with the service level indicator (e.g., run queue length) as the root with arcs to all other measurement variables. Diagnosis using a MNG involves the following:

1. Start with the root variable (e.g., a service level indicator such as response time);

2. Identify the child variables that best characterize the performance problem and traverse the arcs to these variables;

3. Repeat steps 1 and 2 for each non-leaf child variable.

Key to diagnosis is the second step in the above procedure. Specifically, how is it determined which child variables best characterize the parent variable? One technique is to have thresholds for the values of measurement variables; an arc is traversed if the threshold for the destination variable is violated. For example, the thresholds might be: $R_{CPU}, R_{IO}, R_{PAGE}, R_{OTHER} < .2$. Now, suppose that $R_{CPU} =$

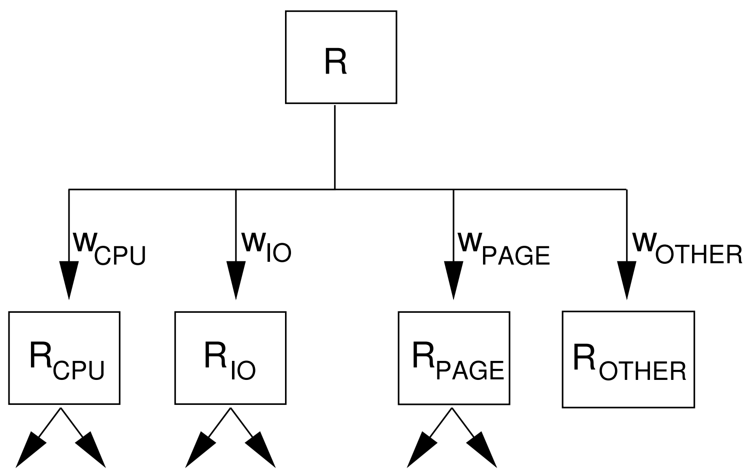

Figure 1: Measurement Navigation Graph (MNG) for a Response Time Performance Problem

8 and $R_{IO}, R_{PAGE}, R_{OTHER} = .1$; then, only the arc to R_{CPU} would be traversed, and R_{CPU} would be considered a causal variable. In general, multiple arcs from a parent variable may be traversed.

Threshold analysis is only one of several **diagnostic techniques** that determine which arcs to traverse in a MNG. The next section discusses several other diagnostic techniques as well. In order to compare these techniques, it is useful to express their operation in a common context. The perspective taken here is that arc traversal (a) uses a diagnostic technique to compute non-negative numbers or **weights** and then (b) traverses those arcs with the largest weights. For example, in Fig. 1, the weights for R_{CPU}, R_{IO}, R_{PAGE}, and R_{OTHER} are denoted by w_{CPU}, w_{IO}, w_{PAGE}, and w_{OTHER}. Suppose that $w_{CPU} = .52$, $w_{IO} = .37$, $w_{PAGE} = .19$, and $w_{OTHER} = .32$. Then, if only one arc from R were traversed, it would be the one to R_{CPU}.

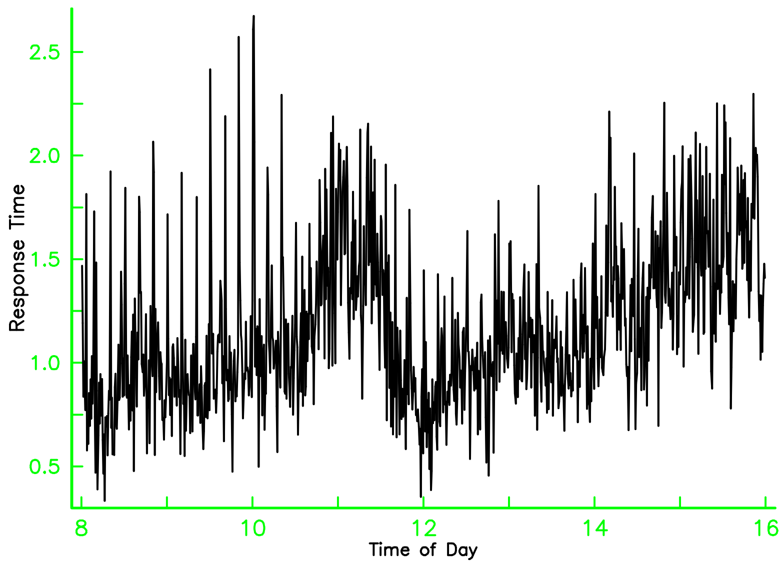

Figure 2: Response Times for System Used in Running Example

3. Comparisons of Diagnostic Techniques

This section describes and compares several techniques for diagnosing performance problems in information systems. In the language of Section 2, these diagnostic techniques compute weights for arcs in the measurement navigation graph (MNG). The techniques discussed are: threshold analysis, bottleneck analysis, what's-different analysis, and correlation analysis.

To facilitate the presentation, a running example is used in which the four techniques are applied to a performance problem in a VM/SP HPO Computer System running an engineering and text processing workload. Fig. 2 plots response times for this system versus time of day. The problem considered is the abrupt increase in response time that occurs at approximately 10:30 and continues until approximately 11:30. The components of response time in this example are the same as in Fig. 1: R_{CPU}, R_{IO}, R_{PAGE}, and R_{OTHER}.

Some notation is introduced to help in the comparisons presented. The k^{th} measurement variable is denoted by x_k; x_0 denotes the parent variable, and $x_1, \cdots, x_N$ denote the N child variables. For example, in Fig. 1, $x_0 = R$, $x_1 = R_{CPU}$, $x_2 = R_{IO}$, $x_3 = R_{PAGE}$, and $x_4 = R_{OTHER}$. The value of x_k at time t is denoted by $x_k(t)$. There are M measurements of each variable when the performance problem is present, which are referred to as the **problem data**. Last, to be consistent with the running example, it is assumed that

$$x_0(t) = x_1(t) + \cdots + x_N(t). \tag{3.1}$$

Generalizations to Eq. (3.1) are possible (as noted in the following discussion).

3.1. Threshold Analysis

Threshold analysis is probably the most widely used diagnostic technique. It has been employed in numerous expert systems for performance tuning (e.g., [1], [6], and [13]) as well as applications for assessing quality of service in commercial T1-lines [12] and diagnosing problems in power plants [7]. Threshold analysis requires specifying an acceptable range for values of each measurement variable; variables that lie outside their range are considered causal. Typically, threshold analysis is applied to mean values, where the k^{th} sample mean is

$$\bar{x}_k = \frac{\sum_{t=1}^{M} x_k(t)}{M}. \tag{3.2}$$

In general, thresholds take the form of a lower and an upper limit, although for most metrics only one limit is used. Let α_k be the lower limit for x_k and β_k be its upper limit. Then $w_k(T)$, the weight that threshold analysis assigns to x_k (for $k > 0$), is

$$w_k(T) = \begin{cases} 0 & \text{if } \alpha_k < \bar{x}_k < \beta_k \\ 1 & \text{otherwise} \end{cases} \tag{3.3}$$

For example, suppose that x_1 is R_{CPU} and x_2 is R_{IO}. Further, let $\alpha_1 = 0 = \alpha_2$, $\beta_1 = .26$, $\beta_2 = .30$, $\bar{x}_1 = .19$, and $\bar{x}_2 = .33$. Then, $w_1(T) = 0 < w_2(T)$. Hence, x_2 will be navigated to from x_0, but x_1 will not. Throughout, **(T)** is used to denote metrics that refer to threshold analysis. A **false positive** occurs if $w_k(T) = 1$ and the k-th variable is not causal; a **false negative** occurs if $w_k(T) = 0$ and the k-th variable is causal.

Two issues arise when using threshold analysis in practice. Foremost is the matter of determining the α_k and β_k. In some diagnostic applications (e.g., the analysis of

$metric$	R	R_{CPU}	R_{IO}	R_{PAGE}	R_{OTHER}
$x_{k,.9}$	1.35	0.26	0.30	0.17	0.85
$x_{k,.95}$	1.48	0.33	0.32	0.22	0.91
$x_{k,.99}$	1.59	0.38	0.38	0.25	1.11
$\bar{x}_k$	1.43	0.19	0.33	0.15	0.76
$\bar{x}_k - \bar{x}_k^r$	0.37	0.04	0.14	0.07	0.12
$r_{xx}(0,k)$	1.00	0.47	0.46	0.39	0.74

- x_{kp} = p-th quantile of the k-th variable in the reference data
- $\bar{x}_k$ = sample mean of the k-th variable in the problem data
- $\bar{x}_k^r$ = sample mean of the k-th variable in the reference data
- $r_{xx}(0,k)$ = sample correlation between the 0-th and the k-th variable in the problem data

Figure 3: Key Statistics for Running Example

w_k	R_{CPU}	R_{IO}	R_{PAGE}	R_{OTHER}
$w_k(T)$, $p = .9$	0	1	0	0
$w_k(T)$, $p = .95$	0	1	0	0
$w_k(T)$, $p = .99$	0	0	0	0
$w_k(B)$	0.13	0.23	0.10	0.54
$w_k(W)$	0.12	0.37	0.19	0.32
$w_k(C)$	0.47	0.46	0.39	0.74

- T = threshold analysis
- B = bottleneck analysis
- W = what's-different analysis
- C = correlation analysis

Figure 4: Weights Computed for Running Example

power plants in [17]), the α_k and β_k are based on physical tolerances. However, such tolerances are rare in information systems. Instead, a common approach is to use quantiles obtained from historical or **reference data** (as in [5]) that has been collected when no problem is present. Ideally, these should be quantiles of *mean* values averaged over M samples (as in Eq. (3.2)). However, in practice it is common to use a reference distribution of *individual* samples since obtaining an empirical distribution that is representative of $\bar{x}_k$ requires a factor of M more data, which may require several months of data instead of just the previous day. Because installation workloads and configurations change frequently, it is very difficult to obtain such volumes of data under the same conditions.

Unfortunately, approximating the distribution of $\bar{x}_k$ by the distribution of the $x_k(t)$ can result in selecting thresholds that are too large since $\bar{x}_k$ has a variance that is a factor of $\sqrt{M}$ smaller than $x_k(t)$; hence, quantiles of the $x_k(t)$ distribution tend to be considerably larger than those in the $\bar{x}_k$ distribution. Having an excessively large threshold results in a higher than expected level of false negatives. Let x_{kp} denote the p^{th} quantile of x_k obtained from the reference data. In general, p should be chosen so as to achieve an acceptable balance between false negatives and false positives. Fig. 3 displays data obtained from the system used in the running example. Included in the table are values of x_{kp} for $p = .90, .95, .99$. Fig. 4 displays the $w_k(T)$ for each value of p with $\alpha_k = 0$ and $\beta_k = x_{kp}$. For example, let x_2 be R_{IO}, $x_{2,.9} = .30 = \beta_2$, and $\bar{x}_2 = .33$. Since $\bar{x}_2 > x_{2,.9}$, $w_2(T) = 1$.

A second issue with threshold analysis is that it does not quantify the relative effect of child variables on the parent. In part, this is due to the use of binary weights; that is, $w_k(T)$ is either 0 or 1. Although using binary weights simplifies matters, it also results in the loss of significant information. For example, suppose that the following thresholds are used: $R_{CPU}, R_{IO}, R_{PAGE}, R_{OTHER} < .2$. Now suppose that $R_{CPU} = 8$, $R_{IO} = 1$, and $R_{PAGE} = R_{OTHER} = .1$. Clearly, both R_{CPU} and R_{IO} violate their thresholds; however, the magnitude of the violation for R_{CPU} is considerable greater, an insight that is not conveyed by binary weights. One way to avoid this deficiency is to use the quantile of $\bar{x}_k$ in the reference distribution as the value of w_k. For example, it may be that 70% of the values of R_{IO} in its reference distribution are less than or equal to 1, and 95% of the values of R_{CPU} in the reference distribution are less than or equal to 8; thus, $w_{IO} = 70$ and $w_{CPU} = 95$. However, this approach does not address the inability of threshold analysis to quantify the impact of the child variable on the parent variable. Specifically, while a quantile of 70% for R_{IO} may seem large, R_{IO} accounts for only 11% of the value of R.

3.2. Bottleneck Analysis

Considered next is bottleneck analysis. This technique has been used in many contexts, such as expert systems for performance tuning (e.g., [4] and [8]), general methodologies for performance analysis (e.g., [18]), and automating assistance with linear programming models [9]. The guiding principle of bottleneck analysis is to reduce the level of the largest component so as to obtain the biggest reduction in the parent variable. As such, bottleneck analysis constrains the semantics of the parent variable in that a larger value is assumed to indicate that performance is worse. However, this constraint is not always appropriate. For example, it is *desirable* to have larger values of throughput metrics.

Bottleneck analysis calculates weights as follows:

$$w_k(B) = \frac{\bar{x}_k}{\bar{x}_0}. \tag{3.4}$$

Throughout, **(B)** is used to denote metrics that refer to bottleneck analysis. Applying bottleneck analysis to the data in Fig. 3 yields the weights shown in Fig. 4. For example, let x_0 be R and x_2 be R_{IO}. $w_2(B) = \bar{x}_2/\bar{x}_0 = .33/1.43 = .23$.

Bottleneck analysis does not suffer from any of the deficiencies of threshold analysis: it does not require the use of historical data, and it calculates arc weights in a manner that reflects the effect child variables have on their parent. To illustrate the latter, recall the numerical example used for threshold analysis in which $R_{CPU} = 8$, $R_{IO} = 1$, and $R_{PAGE} = R_{OTHER} = .1$; bottleneck analysis reflects the differing impacts on R in that $w_{CPU} = .87$, $w_{IO} = .11$, and $w_{PAGE} = w_{OTHER} = .01$.

Bottleneck analysis does have two shortcomings. First, as used in practice, bottleneck analysis assumes that the value of the parent variable is equal to the sum of the values of the child variables. It is possible to generalize this somewhat so that the parent variable is a linear function of the child variables. Specifically, it suffices that

$$x_0(t) = a_1 x_1(t) + \cdots + a_N x_N(t),$$

where the a_k are non-negative constants. In many cases, this linearity assumption is reasonable, such as the decomposition of CPU utilization by user. However, in other cases, linearity is unreasonable, such as the relationship between CPU delays and CPU utilizations.

A second drawback of bottleneck analysis is that it does not addressed what *caused* the performance problem; rather, it focuses on what contributes to poor performance. This is a consequence of bottleneck analysis only considering a single value for each measurement variable; hence, there is no way to determine what has changed. For example, having $R_{CPU} = 8$ may be normal; however, having $R_{IO} = 1$ may be the change that prompted user complaints. In contrast, threshold analysis provides insights as to what has changed by comparing problem data with reference data.

3.3. What's-Different Analysis

What's-different analysis provides a technique for diagnosis that combines characteristics of threshold analysis and bottleneck analysis. Central to what's-different analysis is determining how the *change* in each child variable affects the parent variable. This approach can be used to answer questions such as "Why is performance today worse than yesterday?" or "Why are the CICS1 response times so much larger than those for CICS2?"

Like threshold analysis, what's-different analysis requires reference data to assess if variables have changed in a manner that is causal. Herein, $\bar{x}_k^r$ denotes the average value of x_k in the reference data. What's-different analysis is concerned with the difference between the value of each x_k in the problem and reference data; that is, $\bar{x}_k - \bar{x}_k^r$. Applications incorporating this approach include expert systems for performance tuning [10], tools that assist with linear programming models [9], and knowledge-based systems that analyze corporate financial statements [14].

What's-different analysis can be either qualitative or quantitative. Qualitative what's-different analysis is concerned only with directional effects. For example, suppose that both R and R_{CPU} increase, but R_{PAGE} decreases; qualitative what's-different analysis sets w_{CPU} to 1 (since it changed in a manner that can explain the increase in R), and it sets w_{PAGE} to 0 (since a decrease in R_{PAGE} cannot explain an increase in R). Unfortunately, a qualitative analysis may not lead to succinct characterizations of a performance problems. Indeed, in Fig. 3, all child variables increase, and so all child variables would be considered causal by a qualitative analysis.

More insight is often possible using quantitative what's-different analysis. This technique computes arc weights by apportioning the change in the value of the parent variable (i.e., $\bar{x}_0 - \bar{x}_0^r$) amongst its children. For example, the weight for R_{CPU} is computed by dividing the change in R_{CPU} (.04) by the change in R (.37). Thus, based on Eq. (3.1)

$$w_k(W) = \frac{\psi(\bar{x}_k - \bar{x}_k^r)}{\sum_{i=1}^{N} \psi(\bar{x}_i - \bar{x}_i^r)}, \tag{3.5}$$

where $\psi(\bar{x}_k - \bar{x}_k^r) = \bar{x}_k - \bar{x}_k^r$ if $\bar{x}_k - \bar{x}_k^r \geq 0$ and is 0 otherwise (since a decrease in x_k cannot explain an increase in x_0). Throughout, **(W)** is used to denote metrics that refer to what's-different analysis. Fig. 4 reports the $w_k(W)$ calculated for the measurements in Fig. 3. Note that what's-different analysis can be considered a generalization of bottleneck analysis in that $w_k(B) = w_k(W)$ if the reference data are zero.

Two considerations are important when what's-different analysis is employed. First, like threshold analysis, historical data are required. However, the requirements imposed by what's-different analysis are considerably less demanding than those of threshold analysis in that what's-different analysis operates with average values while threshold analysis must estimate distribution quantiles. A second consideration is that like bottleneck analysis, quantitative what's-different analysis assumes there is a known linear relationship between the parent and child variables (although generalizations are possible). As noted previously, this is appropriate in some cases, but not in others.

3.4. Correlation Analysis

Some of the shortcomings of the foregoing techniques can be overcome by using a fourth approach to diagnosing performance problems: correlation analysis. Correlation analysis offers the opportunity to identify the underlying cause of a performance problem without the use of reference data. Doing so involves quantifying the degree to which changes in child variables track changes in the parent variable. Correlation analysis has been incorporated into several tools for performance analysis (e.g., NaviGraph [15] and the Performance Analysis Facility/VM [11]).

Intuitively, two variables track each other if they always change in the same direction or if they always change in opposite directions. This is quantified by the sample covariance and the sample correlation. Let x_m and x_n be two variables. The **sample covariance** of x_m and x_n is

$$c_{mn} = \sum_{t=1}^{M} \frac{(x_m(t) - \bar{x}_m)(x_n(t) - \bar{x}_n)}{M}. \tag{3.6}$$

Note that c_{mn} is positive if x_n and x_m always change in the same direction, and c_{mn} is negative if the two variables always change in opposite directions. The **sample correlation** between x_m and x_n is

$$r_{mn} = \frac{c_{mn}}{\sqrt{c_{nn} c_{mm}}}. \tag{3.7}$$

It turns out that $-1 \leq r_{mn} \leq 1$ [16]; $r_{mn} = -1$ implies that x_m and x_n always change in opposite directions; $r_{mn} = 1$ implies that x_m and x_n always change in the same direction. Either case suggests a causal relationship.

Correlation analysis uses the sample correlation between the parent and child variables to identify causal variables. If r_{0k} is very close to 1 or to -1, x_k is considered causal. Thus,

$$w_k(C) = \mid r_{0k} \mid . \tag{3.8}$$

Throughout, **(C)** is used to denote metrics that refer to correlation analysis. As before, Fig. 4 contains the weights computed for correlation analysis based on the data in Fig. 3.

Employing correlation analysis raises three issues. First, the problem data must contain multiple measurements of each variable in order to compute the correlation function. Thus, either the problem must exist for a sufficiently long time or the measurements must be fairly fine grain. Second, correlation analysis can give misleading

Desirable Features	*Diagnostic Technique*			
	T	B	W	C
Applies if non-linearities are present	Yes	No	No	No[2]
Only requires problem data	No[1]	Yes	No	Yes
Reflects effect of child variable on the parent variable	No	Yes	Yes	Yes
Handles constant-level performance problems	Yes	Yes	Yes	No
Average values suffice	No	Yes	Yes	No
Identifies what changed	Yes	No	Yes	Yes

1. Historical data are required to obtain threshold values.

2. Can be applied to non-linear function, but the results may be misleading.

Figure 5: Summary of Desirable Features of Diagnostic Techniques

results if there is a non-linear relationship between the parent and child variables. Specifically, if x_0 is a deterministic function of only x_k, $w_k(C)$ may be quite modest if this function is non-linear; hence, x_k might not be identified as a causal variable.

Third, correlation analysis cannot be applied to constant-level performance problems; that is, problems for which $x_0(t)$ is a constant. For example, a constant-level performance problem is present if the parent variable is CPU utilization, and CPU utilization is always 100% in the problem data (e.g., there is always a queue of batch jobs). The effect of a constant-level performance problem on $w_k(C)$ can be seen in Eq. (3.6) and Eq. (3.7). When $x_0(t)$ is a constant, $x_0(t) = \bar{x}_0$ and so $c_{0k} = 0 = c_{00}$; hence, $w_k(C) = \mid r_{0k} \mid = 0/0$.

3.5. Summary of Comparisons

Fig. 5 summarizes the comparisons presented in this section. It is clear that no technique is uniformly preferred to the others, although each has appeal under specific circumstances. In particular, while threshold analysis has many shortcomings, it is the only technique discussed that is broadly applicable when there is a non-linear relationship between parent and child variables. The appeal of bottleneck analysis is that it quantifies the effect of child variables on the parent, and it does not require historical data. What's-different analysis has the advantage of identifying what changed (including constant-level performance problems) while requiring only a modest amount of historical data (since it uses average values). Correlation analysis has appeal if sufficient problem data are available, relevant reference data cannot be obtained, and there is variability in the level of the performance problem.

How good are these techniques in practice? While a complete answer is beyond the scope of this paper, some insights can be obtained from the data in Fig. 4. Note that the techniques produce different orderings of the weights assigned to child variables in the running example; hence, different variables would be identified as causal. For example, bottleneck analysis and correlation analysis assign their largest weight to R_{OTHER}, while what's-different analysis and threshold analysis assign their largest weight to R_{IO}. Also, correlation analysis ranks R_{CPU} second; but for bottleneck analysis this variable is third, and for what's-different analysis it is fourth. Which ordering is correct? In general, this is a difficult question to answer. However, for these data, the available information strongly suggests that R_{PAGE} and R_{CPU} are the causal variables since the performance problem related to difficulties with managing one type of memory pool. Unfortunately, none of the techniques assigned their largest weight to either of these variables; that is, **none of the techniques produced the correct diagnosis!**

That none of the techniques identified the most likely causal variables is somewhat unsettling in that, to the best of our knowledge, all existing expert systems for diagnosing performance problems employ at least one of the four techniques herein discussed. Clearly, further study is required.

4. Conclusions

This paper is a first attempt at studying the quality of the results produced by expert systems for CPE. The study focuses on diagnosing performance problems; that is, characterizing performance problems in terms of causal variables. Four techniques are studied: threshold analysis, bottleneck analysis, what's-different analysis, and correlation analysis. All have strengths and weaknesses. For example,

- Threshold analysis can be applied to variables with non-linear relationships such as the relationship between response times and utilizations. However, threshold analysis requires extensive and carefully chosen historical data to characterize normal behavior. Also, threshold analysis does not quantify how much variables contribute to the performance problem.

- Bottleneck analysis does not require historical data, and it quantifies the impact of variables on the performance problem. However, bottleneck analysis does not characterize the changes that led to the performance problem. Further, bottleneck analysis is limited to variables with linear relationships such as between total CPU utilization and CPU utilization by individual users.

- What's-different analysis incorporates both the notion of quantifying impact and identifying variables that account for a change in performance. However, this technique requires historical data (although not as much data as threshold analysis), and it assumes that there is a linear relationship between the variables being analyzed.

- Correlation analysis provides a way to characterize causal variables by quantifying the extent to which variables track the changes in the level of the performance problem. However, correlation analysis cannot be applied to constant-level performance problems, requires multiple measurements of each variable in the problem data, and is most effective if the relationship between measurement variables is linear.

To the best of our knowledge, all expert systems for diagnosing performance problems employ one or more of the techniques studied in this paper. Thus, the results reported here are unsettling in that all of the techniques have significant limitations. Understanding the seriousness of these limitations will require further study. In particular, efforts are needed in three areas: characterizing performance problems, constructing metrics for evaluating the quality of the diagnoses produced by expert systems for CPE, and developing approaches for the quantitative analysis of the diagnoses produced. A starting point for the characterizing work is to study systems with performance problems. For example, the system in Fig. 2 contains three kinds of performance problems: periodic spikes prior to 10:00; a level shift between 10:30 and 11:30; and a ramp towards the end of the day. Such observations could be used to characterize the values of both causal variables and service level indicators.

Two kinds of metrics should be considered for quantifying the results produced by expert systems for CPE. The first take into account the specific performance problem. An example of such a metric is the probability of correctly identifying the causal variables given the kind of performance problem present. A second class of metrics deals with theoretical properties of diagnostic techniques. For example, a desirable property is consistency in the sense that if the performance problem is made sufficiently severe the technique will identify correctly the set of causal variables. Not all techniques possess this property. For example, threshold analysis is *not* consistent. To see this, recall that threshold analysis identifies a variable as causal if it violates its threshold. Thus, no matter how severe the performance problem is made, a non-causal variable that violates its threshold is still classified as a causal variable. In contrast, bottleneck analysis is consistent since as a performance problem becomes more severe, the fraction of the service level indicator that is attributed to non-causal variables approaches 0; hence, non-causal variables are not identified as causal.

Last, quantitative analyses of expert systems are required. In part this will involve empirical studies, such as creating performance problems in a computer system and then comparing the results produced by different diagnostic techniques. However, just as queueing theory forms the basis for CPE, an underlying theory is needed for the analysis of expert systems for CPE. In so far as the metrics employed for this analysis are statistical in nature, this theory will have a statistical flavor. For example, the probability of detecting a performance problem given that one exists is very similar to the concept of the power of a statistical test.

Acknowledgements

The author thanks Robert Berry, Norbert Lenz, and William White for their thoughtful comments.

References

[1] **B. Arinze, M. Igbaria, and L.F. Young**: "A Knowledge Based Decision Support System for Computer Performance Management," *Decision Support Systems* **8**, 501-515, 1992.

[2] **Robert F. Berry and Joseph L. Hellerstein**: "A Unified Approach to Interpreting Measurement Data in Performance Management Applications," *First IEEE Conference on Systems Management*, University of California, Los Angeles, May, 1993.

[3] **Bernard Domanski**: "A PROLOG-based Expert System for Tuning MVS/XA," *Proceedings of the Computer Measurement Group*, 160-166, 1987.

[4] **Boole & Babbage**: *DASD ADVISOR Technical Backgrounder*, Boole & Babbage, Inc., 10BAB34.BGR, December, 1987.

[5] **Tom Bowen and Liz Payling**: "Expert Systems for Performance Review," *Journal of Operations Research Society*, **38** 929-934, 1987.

[6] **Computer Associates**: "CA-MINDOVER," Computer Associates, 711 Stewart Avenue, Garden City, New York, 1993.

[7] **A. D'Ambrosio, M. Oriati, and A. Serventi**: "AI in the Power Plants: PERF-EXS, a PERFormance diagnostics EXpert System," *IEEE Conference on AI Applications*, 11-17, 1988.

[8] **Siegmund Gralla** "An Expert System for VM Performance Analysis," *Research Report*, RC 12784, IBM Corporation, 1987.

[9] **Harvey J. Greenberg**: "Rule-based Intelligence to Support Linear Program Analysis," *Decision Support Systems*, **9**, 425-447, 1993.

[10] **J.L. Hellerstein, David A. Klein, and Keith R. Milliken**: *Expert Systems in Data Processing* Addison-Wesley, 1989.

[11] **IBM**: *VMPAF User's Guide and Reference Manual*, (SC23-0564-00) IBM Corporation, 1993.

[12] **D. R. Irwin**: "Monitoring the Performance of Commercial T1-rate Transmission Service," *IBM Journal of Research and Development*, 805-814, 1991.

[13] **Kiyoshi Itoh, and Takaaki Konno**: "An Integrated Method for Parameter Tuning on Synchronized Queueing Network Bottlenecks by Qualitative and Quantitative Reasoning," *IEICE Transactions on Information and Systems* **5**, 635-647, 1992.

[14] **Donald W. Kosy and Ben P. Wise**: "Self-Explanatory Financial Planning Models," *Proceedings of the National Conference on Artificial Intelligence*, 176-181, 1984.

[15] **Landmark Systems Corporation**: "NaviGraph," Landmark Systems Corporation, 8000 Towers Crescent Drive, Vienna, Virginia, 1993.

[16] **B. W. Lindgren**: **Statistical Theory**, The MacMillian Company, 1968.

[17] **Douglas Smith**: "Monitoring/diagnostic Systems Enhance Plant Asset Management," *Power Engineering*, 23-28, June 1992.

[18] **William Tetzlaff and Thomas Beretvas**: "A New Approach to VM Performance Analysis," *Proceedings of the Computer Measurement Group*, 339-350, 1982.