<u>The IT Manager's Nightmare</u>

Introduction

This article addresses the issue of controlling service levels in an organization's IT system: What are the problems and how the organization's pressures are applied on the people responsible to address and prevent future problems. This article also proposes a technical solution to substantially improve and prevent these problems.

The author is Yoram Kariv, CEO of ConicIT (www.conicit.biz)

A bank president receives a sarcastic message from an important and angry customer who writes "*Again, your X service from IT does not operate properly*" or "*Again the information on your Web site is out of date or the site responds slowly*."

The bank president immediately informs the IT manager, service managers, and the bank's quality control and customer service people. At this stage, when the customer has immediate access to most of the IT problems, attention is raised soon after the problem is identified.

Our primary assumption is that we are all professionals, we all want to solve the problem, and we each individually have excellent reasons, both professional and personal, to want the problem solved.

Despite this, in reality most of us do not have any methodology to deal with the problem!

In many cases we chase after the problem's symptoms and not after the problem itself. A problem in a computerized network escalates within seconds into a maze of unexplained symptoms, where finding reasonable operative measures is about as likely as a game of roulette (and sometimes Russian roulette).

Computerized systems hold countless spools and queues, including those that are not formally called "queues". For example: a single computer processor (CPU) serves a single request at any time, meaning that the other processing requests (applications, processes) wait for their turn. Access to databases and disks are limited by the redundancy, thus creating queues. Locks that are raised from logical access to databases create queues. Access memory to communication lines and operations on remote servers also create queues. All these and many others are various queues and tasks that compete against each other for access to limited resources.

A transaction or process for a certain operation can be completed within a fraction of a second when it operates as a single task in the computer. But in a regular work environment, under heavy work loads, the response time can rise to many seconds.

Where did the time go?

The time was simply consumed by waiting in queues and by the queue management itself. Parallel tasks rise considerably! These additional tasks obviously add more load to the resources, extending the queues even more. The actual program still required the same fraction of a second to perform, but the time is wasted in the queues and in the resources required to manage the queues. We therefore infer that queues are the primary reason for slow computer response time.

Assuming that we have access to all the relevant information for this problem, is the time when the information is retrieved of any importance to us? Does this type of information allow us to decipher the queue conflict?

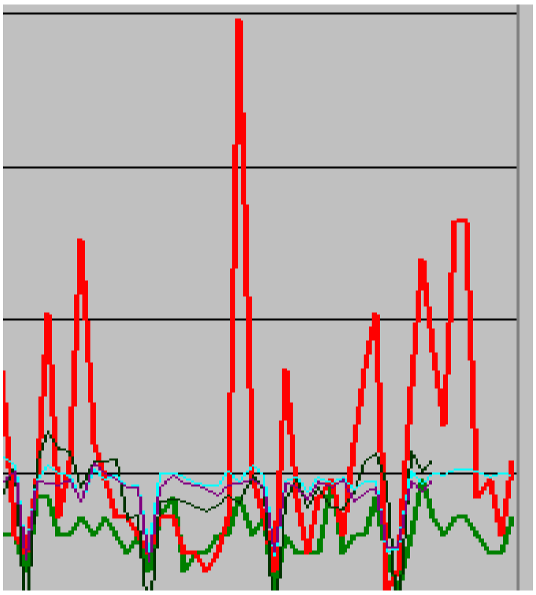
Let us analyze a familiar traffic problem - will a serious traffic jam occur in the intersection? Is the information "*There is a traffic jam*" relevant for the question "*Why is there a traffic jam?*"

We would obviously like to know why the traffic jam occurred or if it will occur again. IT managers ask these and similar questions on a daily basis.

To answer these problems, we need to ask secondary questions: "*Is there a correlation between computer resource consumption relative to the current response time?*" For example: "*Do changes in CPU consumption (more or less) explain something about a possible problem?*" "*Can we predict that we are approaching a problem soon?*"

There is no correlation between the computer resources consumption behavior and CPU and disk access consumption. This means that when the response time graph rises (response time is slower, and thus worse), it does not necessarily indicate a direct rise in CPU consumption or the number of disk access operations. In reality, it seems that each computer resource operates independently and its behavior is not necessary consistent.

For example, we found that CPU consumption rises with operational load, but suddenly drops from a decrease in the operational load, or it may cause tasks that are waiting to be processed to be locked out. What did we notice? All this time, *during peak loads - the processor is resting*…Confusing?



Graph 1: Average response times (red), Event entry rate (blue) CPU consumption (green), File access rate (black)

Graph 1 represents 30 minutes of operation, and illustrates the complexity of this problem. Regular human insight cannot find the correlation between the four parameters displayed above. Just the opposite, we can understand how difficult it is to predict what will happen within the next 5 minutes. "*Where will the problem arise?*" "*Where will a queue form?*" And many more complex questions.

To be more accurate, there is no known correlation between the loads on various resources within a single computer environment. This is even truer in an environment with many computers. There is no way to identify the problem and prevent its reoccurrence.

A Solution to the Problem:
ConicIT has generated a solution, ConicIT/MF. This product operates for us the existing data collection applications. Using a rules engine, it also performs problem detection in "*real time*." It does this while understanding and learning the problems that are integrated into the organization.
"Real time" investigation (detection) and database building provides the organization the following:

- Analyzed and processed data from all the data sources together. It provides a view of the entire computing system as if it is a "*standalone computer*."
- Analyzed data for a time span that starts before the problem started up until the event escalated. This means that the program can provide current and relevant details on the systems and queues within this time span, and thus converting the event from an impending "*problem*" that cannot be solved, but that can have grave consequences, to an understandable and analyzed process with clear symptoms.

The rationale behind the development of ConicIT/MF is that even though we cannot completely replace human intelligence (build a human-machine), we can at least provide the information to the professional who is to deal with the problem. This information will be current, consistent, and analyzed, allowing the system controllers to identify the problem early enough and thus prevent future occurrence.

Yoram Kariv: yoramk@conicit.biz