



# Behavioral Analysis Solutions

## Proactive IT Service Management for the Mainframe

### Table of Contents

Background.....	2
Introduction.....	2
ConicIT for Mainframes.....	3
ConicIT System Overview .....	6
System Components.....	8
Data Extraction and Cleansing.....	8
Analysis Rules Engine .....	9
Analysis Expert System and Self Learning Module.....	9
Outputs: Alerts, Reports, and Graphs .....	10
ConicIT for Cost Optimization .....	11
Summary.....	12



## Background

ConicIT is a software vendor specializing in the development of a unique Mainframe behavioral performance analysis solution for first fault problem resolution. The company's goal is to employ its sophisticated mathematical tools to help enterprises improve their service levels, prevent (recurring) service problems, and reduce costs.

## Introduction

For large enterprises the data center is a critical part of their business infrastructure. New market requirements – such as web based customer self service, globalization, and increased business agility are leading businesses to create new applications that are heavily reliant on existing legacy transaction computing infrastructure and applications. These new applications place heavy performance requirements on existing infrastructure and applications. This has made system performance even more business critical today than a decade ago. Since almost all of an enterprise's transactions interact with mainframes (which contain 70% of the world's business critical data) during processing, any mainframe system slowdown has an immediate effect on the business, and a severe, long duration mainframe system problem can be catastrophic. Over the years, system monitoring tools have evolved but they aren't enough to provide the level of support needed by these new usage paradigms. These new applications and usage paradigms are straining mission critical mainframe applications and IT staff to the limit.

IT staff is being called upon to provide flawless service response and immediate problem resolution for application issues. Most operational service degradation issues are related to unexpected and unplanned interferences between transactions on the system, making it impossible to provide flawless, 100% problem free operation. This leaves IT staff scrambling after the fact to discover and analyze the complex interactions that cause performance degradation – in most cases waiting for the problem to surface again and again before it can be fixed. Instead of after the fact detective work to solve performance problems, IT management and operations should focus on **first fault problem resolution**. That means both alerting IT staff and capturing all the relevant system data during a performance problem so that the IT staff will have a post-mortem detailed view of the system status right before, and during the problem's occurrence. Making this data available to the staff allows them to solve a problem the first time it occurs – providing a dramatic reduction in mean time to repair for mainframe transaction slowdowns and performance degradation.





as a result there will be over-usage of the memory, and poor response time. When all the facts are known, it is clear that the high memory usage is a symptom, and not the root cause of the problem. The DB2 lock is the root cause. Increasing the amount of memory available to the CICS will not solve this problem.

- **Data Complexity** – in order to identify a problem occurring within a system, it is necessary to understand what the normal behavior of the system is. Only with this knowledge will it be possible to identify the abnormal behavior that leads to a problem. Due to the large number of parameters in the system, each of which changes constantly, defining the normal behavior for each such parameter is a very difficult task. It is impossible to perform this task manually.
- **Lack of Context** – the definition of "normal behavior" for each parameter depends on the context. For example: the normal behavior of the dispatching wait (the amount of time a job is ready for CPU but cannot be allocated CPU time) is very different at different times of the day (think 11:00 a.m. vs. 11:00 p.m.), different days of the week (weekday vs. weekend, holiday, first working day of each month, etc.), and is affected by other activities within the system, and sometimes even between LPARs. Without taking the context into consideration, it is impossible to accurately define the normal behavior of each parameter, and thus it is impossible to perform reliable analysis to identify problems.  
Furthermore, most monitoring tools today collect data from a single component. Without looking at the bigger picture (the whole mainframe environment), it is usually very difficult to understand service problems.

To address those issues we developed ConicIT for Mainframes, a unique product that uses sophisticated mathematics and algorithms to solve the problem of alerts and root cause analysis for mainframe systems:

- State-of-the-art profiling algorithms to automatically assess the normal behavior of each one of the different parameters of the system according to historic data during different days of the week and different times of the day. There are thousands of parameters changing constantly throughout the day therefore performing this task manually is almost impossible.
- An advanced data-cleansing algorithm to address the non-stationary behavior and non-linearity of the monitored signals and to extract the underlying phenomena of interest from the noisy input.
- A holistic view of the mainframe system, taking into account the inter-correlation between its different parts.



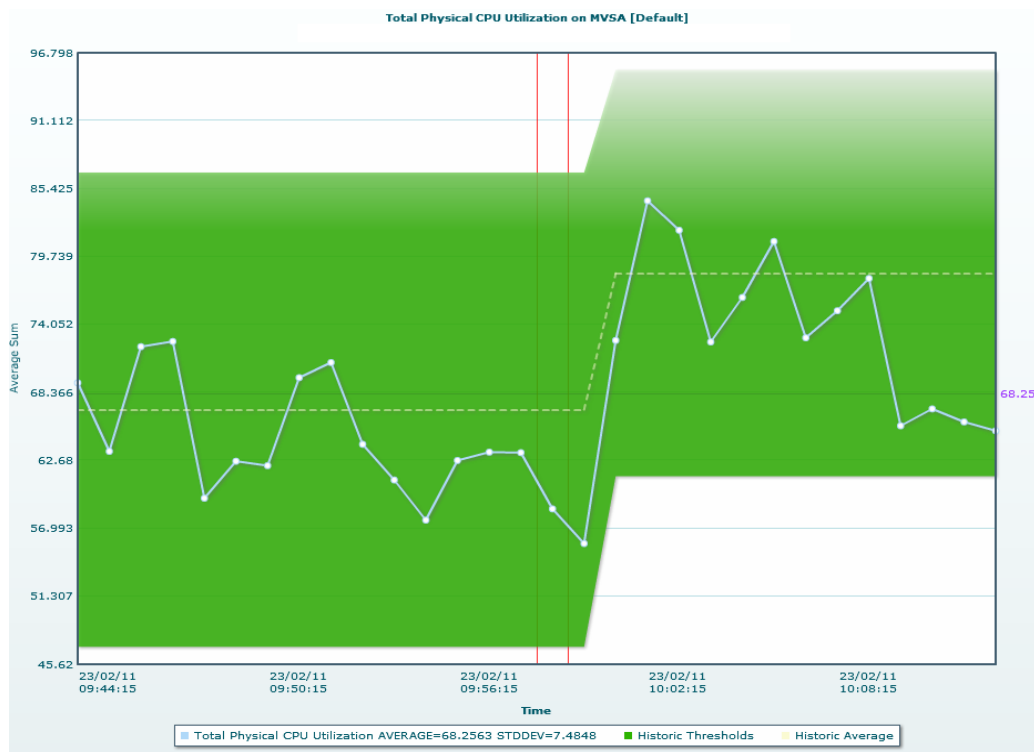
- A combination of data-agnostic mathematical models and domain knowledge of the mainframe environment to create an expert system that can identify service problems before they start.



## ConicIT System Overview

ConicIT runs on a separate Linux system external to the system being monitored. ConicIT is a completely agentless architecture which doesn't require installation on the system being monitored. It receives data from existing monitors (e.g. Omegamon, TMon, Sysview, Mainview) through their standard interfaces. User emulations enable ConicIT to appear as just another operator to the existing monitor and adds no more load to the monitored system than would adding an additional human operator.

The monitored data stream is retrieved by parsing the data from the various data sources. This raw data is first sent to ConicIT's data cleansing component. Data from existing monitors is very "noisy", since various system parameters values can fluctuate widely even when the system is running perfectly. **Graph 1** shows sample CPU utilization measurements over time from a perfectly normal system. Even though the CPU utilization is deviating greatly from the mean – it is not necessarily indicative of a system problem. The job of the data cleansing algorithm is to find meaningful features from the fluctuating data. Without an appropriate data cleansing algorithm it is very difficult or impossible for any useful analysis to take place. Such cleansing is a simple visual task for a trained operator, but is very tricky for an automated algorithm.



**Graph 1: Normal CPU Fluctuations**



The relevant features found by the data cleansing algorithm are then processed to create appropriate variables. These variables are created by a set of rules that can process the data and apply transformations to the data (e.g. combine single data points into a new synthesized variable, aggregate data points) to better describe the relevant state of the system.

The variables are then put into context – comparing each value to the normal value of that variable and looking at the system as a whole. The normal behavior profiles of each and every one of the variables are calculated automatically for different days and times of the day, taking into account holidays and special days of the month (e.g. the first working day of the month). They are updated frequently to adapt to the seasonal changes in the data. Only by putting the variables in context is it possible to identify which parts of the system are behaving abnormally in a way that could cause a performance problem.

When a possible performance problem is predicted the system issues an alert to the appropriate consoles. No guesswork is needed to define the thresholds of these alerts and no ongoing work is needed to update them because they are based on deviations from the normal behavior of the system. This baseline of normal behavior is calculated automatically and constantly updated.

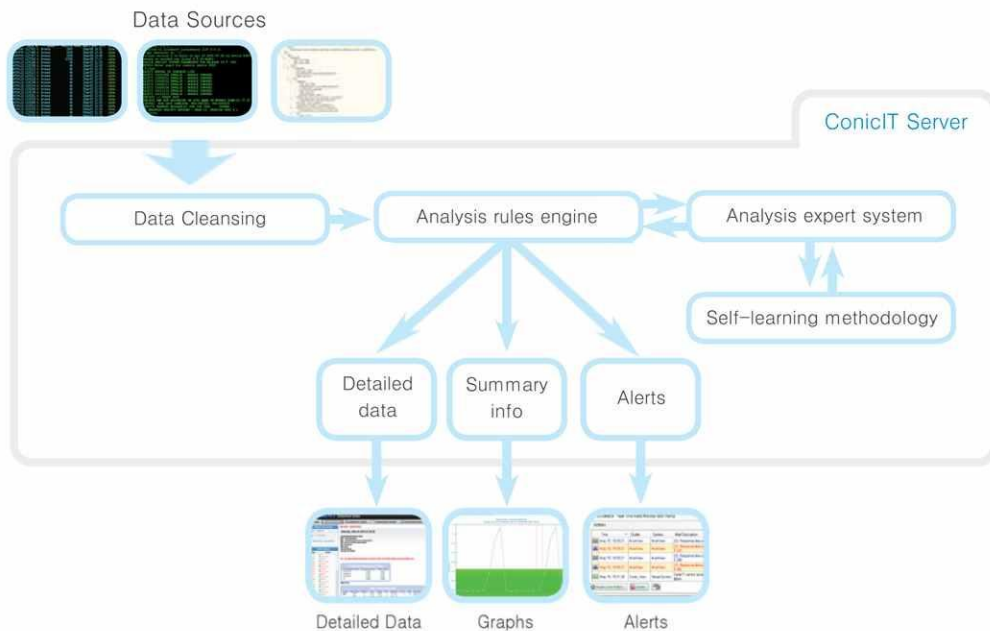
On top of the alert that is issued, ConicIT creates a detailed report describing the state of the different parts of the monitored system and all of the anomalies that were detected. This report can be used to quickly identify which components are behaving abnormally, when they started to behave this way, and in which component the problem originated. Pinpointing the root cause of the problem becomes an easy task, making it unnecessary to assemble large teams of experts to investigate and waste time analyzing after-the-fact data.

The result is that when service problems start brewing IT personnel already have a heads up, it is clear what the root cause of the problem is, who is responsible for solving it, and the problem can be fixed correctly the first time.

## System Components

As shown in **Figure 1**, ConicIT's system architecture consists of 3 major components:

1. A data cleansing component
2. An analysis rules engine
3. An expert system and self-learning component



**Figure 1: ConicIT Major Component Overview**

## Data Extraction and Cleansing

The system interfaces with the system monitors and other data sources through standard (3270) emulation. The system appears as just another user to the monitors, and requests data from monitors on a regular basis. Data can also be extracted from SNMP traps, command-line tools, and files. The extracted data is parsed into a form usable by the data cleansing component.

This raw stream of data enters the data cleansing component as a set of simple and complex variables obtained by preprocessing of the input data stream. The data cleansing algorithm then uses various sophisticated mathematical models to decide what data should be passed the next stage for analysis. The enormous fluctuations in systems data make standard filters and



statistical analysis irrelevant. For example, looking at the data in **Graph 1** it is clear that static thresholding won't work – it will cause either too few or too many alerts, depending on how the threshold is set. Using the mean is equally uninformative.

ConicIT uses a proprietary data cleansing algorithm that is fast enough to analyze the real time data stream while ensuring that the analysis engine is not swamped with irrelevant data.

### **Analysis Rules Engine**

This component takes relevant data provided by the data cleansing component and uses it to create specific variables appropriate for performance analysis. This allows the system to easily support new monitors, since when adding a new monitor only the preprocessing needs to be changed, not the data cleansing component. ConicIT configuration set up also allows for easy mapping between the 3270 data stream and variables needed by expert system, and it also enables the use of multivariate data (creating a synthetic variable that represents the merged behavior of more than just a single measured variable) which enables the models to use correlated variables to find anomalies.

The analysis rule engine is also responsible for data manipulation related to end user needs (e.g. data aggregation for user display) and for creating all of the user notifications and reports. This engine uses the relevant data it obtains from the data cleansing component and results from the expert system to create end user screens, reports and alerts.

The analysis rule engine is implemented as rules in a standard scripting language (Perl) and can be easily extended.

### **Analysis Expert System and Self Learning Module**

These are the modules that are responsible for creating and updating the behavior profiles for each and every one of the variables collected from the monitored systems.

The Analysis Expert System is the real-time component used by the Rules Engine to understand what the normal behavior of each variable is for the specific date and time at which the data was extracted. The expert system can say how "abnormal" the data is, meaning how far it deviates from normal behavior. This information is used by the Rules Engine, along with domain knowledge about the mainframe environment, to decide if a problem is brewing and whether or not an alert should be issued.

The self-learning mechanism is an offline algorithm that uses historical data to periodically calculate and update the normal behavior profiles. It calculates the profiles of each variable for different dates and times, taking into account holidays, special days of the month, and seasonal changes in the data.



## Outputs: Alerts, Reports, and Graphs

The results of the analyses performed by the system are given to the users in three forms: alerts, reports, and graphs.

Alerts are issued when a problem is detected. These alerts are smart alerts because they are not issued according to a fixed threshold, but rather according to the deviation of variables from their normal behavior. They are much more reliable than fixed threshold based alerts. The alerts can be shown on the user's PC using ConicIT's alert system, or on central alert management system consoles (like IBM™ Tivoli Console or BMC™ Patrol Console). They notify the user that a problem is occurring in one or more of the system components and contain a brief explanation about the detected problem.

Reports are created every time data from the monitored system is analyzed, regardless of whether a problem was detected or not. The reports show the status of the different parts of the system. If an anomaly was detected in a certain part of the system there will be additional explanations about the specific anomaly detected and any additional data necessary to understand this problem.

Graphs can be created for any variable collected by the system, as well as all of the synthetic variables generated by the system. These graphs can be created for any time frame whether it be 30 minutes or one year. Comparative graphs can also be generated to compare the behavior of a certain variable during different times. These graphs assist in the problem analysis, giving a visual way to understand how severe a problem is, when it started, and what the normal behavior of the variable is.

The users are alerted if and only if there is a real problem brewing, and all of the information necessary to pinpoint the root cause of the problem is given to them in the reports and graphs. This drastically shortens the time required to solve service problems, as well as making it very simple to understand who needs to solve the problem, thus saving precious time and money for the organization.



## ConicIT for Cost Optimization

While talking about service management issues with our customers we learned that one of the most important things to them is the price of maintaining their mainframe SLA. Mainframe costs are rising and the current economic environment means increased scrutiny of the spending needed to maintain SLAs. The increasing use of dynamic vendor pricing models makes it very difficult to plan (or even understand) the exact costs associated with maintaining a mainframe SLA. For example a peak CPU usage overage for 20 minutes can affect the charges of the whole month, and underutilization of a parallel sysplex LPAR can cancel the discount given to PSLC licenses. This tension has created the need for cost aware real time monitoring tools – tools that can factor dynamic cost models into the management environment, and issue real-time alerts when service related problems affect the bill paid to software vendors.

We have learned that the same core technology developed by ConicIT for service management can be used to help customers perform cost-aware SLA management. This means that customers will be able to understand the price of maintaining their SLA and take action to reduce this price. ConicIT can alert ahead of time when expected CPU usage peaks will affect sub-capacity and other usage based pricing models. It is then possible to make a conscious decision concerning the SLA and the price of maintaining it. It is also possible to take action in order to prevent or minimize the expected CPU usage peak. ConicIT provides a way to see how predicted usage patterns will affect mainframes costs, enabling a proactive approach to managing capacity related costs.



## Summary

As opposed to the assumptions of the 1990's - mainframes are here to stay and mainframe usage continues to grow and evolve. Performance issues continue to plague even the most efficient mainframe installations, and the costs associated with mainframe performance issues are of great concern to CIOs. There are more than 200 billion lines of mainframe code running today's businesses worldwide. Fortune 500 companies maintain 35 million lines of COBOL code and add 10 percent more new lines of mainframe code annually just to keep up with business needs, changing conditions, and regulations. These applications still manage and process most customer, product, supply-chain, and critical business data and most are a mix of screen processing, data I/O, and the core of the application—the enterprise's business rules.

The performance and reliability requirements from these mainframe applications are growing while the number of developers who truly know the applications and systems is decreasing. Even though mainframe systems have robust monitoring and management capabilities, mainframe IT personnel are buckling under the strain of supporting all the business IT requirements, and there is a growing gap between the number of mainframe personnel needed and those available. The existing IT personnel need tools that augment existing management and monitoring tools to support them in the analysis and repair of system performance issues in less time and the first time they happen.

Additionally, the costs of mainframes are constantly on the rise and organizations are trying to save money every way possible. The price of maintaining the organization's SLA is usually unknown to the IT personnel, especially before problems occur. A real time tool is required to assist in cost-aware SLA management – giving IT personnel an early warning about CPU usage peaks being created and making the decision concerning the price of maintaining the SLA an informed one.

ConicIT's technology is uniquely positioned to solve these problems.