The Collection and Integrated Access to Network Performance Information in a large scale Client/Server environment using SAS

Tom MacFarland. Andersen Consulting, Minneapolis, Mn.
Glen Bechtold. Andersen Consulting, Minneapolis, Mn.

ABSTRACT

Effectively managing and reporting the vast amount of data collected in a distributed Client/Server environment is not only complex but can be overwhelming. This paper will discuss how SAS® was used to integrate data from multiple vendor tools and provide an effective network management collection and reporting capability. Discussed will be how base SAS, SAS/CPE® and SAS/GRAPH®, integrated through scripts and the use of the World Wide Web (Web) tools, provided an effective reporting system for management, operations and support.

INTRODUCTION

Client/server distributed environments continue to be employed throughout the world as viable solutions to mission-critical business problems. As more and more of these environments emerge, the need to provide effective performance management becomes ever more critical. What follows will be one approach to the implementation of performance management tools into an integrated collection and reporting process.

Andersen Consulting (AC) operates a large internationally distributed client/server production and development environment located at its offices in Minneapolis, Minnesota. The Minneapolis office supports enterprises, solution centers and a software development environment. Enterprises are independent business unit solutions such as inventory management, sales, payroll, ticket revenue tracking, loan analysis, hospital administration, etc. Solution centers provide services and technical support to clients world wide. Solution centers usually concentrate on specific applications such as PeopleSoft, SAP, etc. Performance management in such an environment presents a formidable challenge; not only because of its widely distributed nature, but also due to the rapidly changing and diverse hardware and software requirements imposed by multiple development and production environments.

A major goal for performance management in this environment, as in most, is to utilize data and experience to provide rapid high quality information. This information guides the management and operations decision process in the day to day management and support of the networks and systems that support the service. This helps ensure that all components (servers, networks, workstations, applications, etc.) continue to deliver high quality, high performance service to the end users. The remainder of this paper will discuss the methodology and processes developed to support those goals.

WHAT IS MANAGED?

The Minneapolis center supports a range of enterprises and Solution Centers, from a large scale enterprise called Passenger Revenue Accounting Solutions (PRA), consisting of over 100 servers and 1000 workstations down to a small scale enterprise consisting of as few as 2 servers and slightly more then 20 workstations. In addition to the enterprise environment, there continue to be enhancements and new solution developments consisting of over 40 servers and 900 workstations. These environments continue to grow and change on a daily basis. The processing components include Sun Microsystems workstations and servers in addition to SGI, Sequent, IBM, HP and Compaq PC workstations and servers. UNIX and NT and Windows95 are the predominant operating systems. The networks consist of Ethernet and FDDI LAN's (Local Area Networks) with TLS (Transparent LAN Service), T1, fractional T1 -and Frame Relay providing the majority of WAN (Wide Area Network) support. The network component vendors are a combination of CISCO, Cabletron, BayNetwork and 3Com routers, hubs and switches.

THE METHODOLOGY

To manage performance in a distributed environment as described above presents a significant challenge. This challenge is to manage performance in a diverse, continually expanding and changing environment while the tools, resources and budgets remain limited. The goal of performance management is to provide **timely** information to diverse audiences. This information addresses the availability, reliability and overall performance of the networks, systems and applications that make up the service delivery. Client requirements from performance management come from three major areas: Management, Operations and Support.

Management level reporting provides information on customer satisfaction (i.e. the service provided) and it's cost effectiveness. Information presented at this level is designed as a means for tracking conformance to service levels. This reporting level usually contains information such as: business units processed (i.e. tickets, loans, merchandise, etc.), batch turnaround, response times, availability, transaction rates, etc.

Operations level reporting provides information which helps determine the effectiveness of the networks,

systems and overall service. This usually consists of information on resource utilization, component bottlenecks, mass storage levels, etc. The information is designed to provide an overall view of the health of all components such as networks, servers and workstations, and serves as an early warning of potential bottlenecks. This class of information also feeds trend analysis, modeling and capacity planning studies.

Support level reporting provides real-time information on the health of the networks, systems and service. This is where traps and events are employed to inform support personnel immediately (within minutes) of process and component failures. Information at this level not only provides the real-time alarms on component and application failures, but also provides automatic or manual "drill downs" to help determine the cause of the failure.

THE COLLECTION ARCHITECTURE

The data collection architecture shown in appendix A figure 1, is based on a clustered design which allows the data capture and collection to occur at points in the networks and systems that will minimize resource impacts. The architecture is designed to provide both event and trending data capture since neither one by itself is sufficient to meet the performance management requirements. All data is transferred to a centrally accessed repository (Network Operation Center). Most of this data is moved periodically and at times when the network and system resources are available to support the transfer with minimal impact. Traps and events are sent **immediately** to all designated network management areas. Events are then filtered, categorized and delivered via pager, trouble ticket, electronic mail, electronic voice, or Web based alarm processes.

SunNet Manager and HP OpenView are used as the main network management systems supporting SNMP (Simple Network Management Protocol) and RPC (Remote Procedure Calls) communications. Currently there are over 900 collection and event agent requests managed through a single SunNet Manager system. Additional consoles may be deployed throughout the network to provide a visual view on the current state of the networks and systems.

The performance data management system is a SAS suite of products. This suite consists primarily of base SAS, SAS/GRAPH and SAS/CPE. Data is distributed among multiple SAS datasets and, in some cases, multiple servers for data organization and security as well as performance considerations. SAS provides excellent multi-platform (UNIX, Windows, NT, etc.) data reduction and statistical features. SAS also supports a flexible native language and SQL based access capability. SAS/CPE provides for conversion of the raw data to SAS and the rollup of that data into hourly, weekly and monthly datasets. Custom SAS scripts have been written to perform a similar function provided by SAS/CPE for those clients which do not have the SAS/CPE product,

A "verify collection" process checks periodically as a fail-safe to insure all component collection processes are functioning. If the verify process detects that data collection has ceased for a particular component, it identifies the failing process and issues an event to the event management system.

SunNet Manager or HP OpenView manages the bulk of data collection on component (i.e., servers, workstations, routers, hubs, etc.) availability and resource utilization. Each "intelligent" component is "pinged" for availability. A subset of components are "pinged" for trip time to provide high-level network response trending.

The NT Monitor Service running on a single NT server is used to collect performance data from the local and remote NT servers. Currently in NT there are over 200 performance metrics available with more appearing every day. The performance monitor service on each NT server periodically captures its data into a file on the NT performance collection server. Nightly data from all NT servers is converted to a comma-separated file by a special batch utility automatically initiated via the "at" (automatic time/date) command. This comma-separated file is then transferred to the SAS server which loads the data into datasets.

The NT Event Log is monitored by a special utility to identify key events. On discovering a key event defined by a configuration file, the event is sent to the network management system. If the NT server is running SMS (**S**ystem **M**anagement **S**erver) then events in the event log can be sent as traps directly to the network management systems.

Three methods are used to collect data on network activity: LAN probes, router MIB's (Management Information Base) and unique software tools. Hardware probes attached to the networks provide LAN and node-to-node utilization as well as protocol profiles. Utilization metrics from this database are moved and processed into SAS datasets daily. Most CSU's/DSU's, routers, switches and smart hubs support MIB's which provide LAN and WAN utilization directly to SunNet Manager via SNMP protocols. A software package "NetMetrix" from Hewlett Packard also is used to collect LAN utilization, node-to-node NFS response and protocol profiles. This data is captured directly into flat files local to a workstation or server connected to the LAN. As in other locally collected data this information is then transferred daily into SAS datasets.

Two methods are used to report application performance: application logs and RTE's (Remote Terminal Emulators). Application logs are produced by many applications such as Sybase, Oracle, etc. These metrics are usually collected directly on the servers and/or workstations on which the applications are running. This data, which consists of response, turnaround, abend, resource and queue information, is also moved daily into the SAS datasets. The RTE's which currently run under Windows95 emulate actual users accessing the

applications. RTE's provide response, accessibility and reliability metrics via logs and SNMP traps. The RTE's are usually the first device to detect system or application outages.

Besides collecting all metrics into SAS datasets, much of the data is also available in real-time. Many of the collection tools such as SunNet Manager and HP OpenView provide their own GUI (Graphical User Interface) to browse this real-time data. In addition to these tools, Web based tools continue to be developed by vendors to provide access to performance based information.

"Trouble ticket" management is provided through Remedy ARS (Action Request System). Remedy is used as a notification and tracking vehicle for network and system events. Remedy also feeds statistics into SAS datasets for trend reporting and analysis. This information consists of metrics such as frequency of new and backlogged trouble tickets logged by type and priority.

At the Minneapolis facility backup of all datasets is provided nightly through a StorageTek 9710 Silo system supporting 6 DLT 4000 tape drives. The SAS dataset on-line sizes are managed using SAS/CPE's automatic data reduction and summarization features or in the absence of SAS/CPE, custom scripts.

THE TRAP/EVENT ARCHITECTURE

SunNet Manager and HP OpenView manage the events and traps received from the network and applications. These management systems process SNMP traps as well as events identified through a general purpose event handling facility. This facility monitors the system log files and on identification of unique message strings defined in an event message configuration file, it forwards the event to the network management system. This design supports the definition of almost any type or class of event desired. The only requirement is that a unique message string be logged into the systems log file. In most cases, it is a matter of minutes from the time a new event notification is required, until it is available and activated in the system. Once an event is posted to the network management system it is logged and notification via digitized voice, mail, trouble ticket and/or pager occurs. The methods of alarms and recipients are defined in a configuration file. This configuration file also provides multiple filters on the event, component and time or day the event occurred. These filters help determine the type, if any, of alarms to issue and the audience who should receive those alarms. Event log information is also placed daily into SAS datasets for exception and trend reporting and analysis.

THE REPORTING ARCHITECTURE

The reporting architecture is the most important area of performance management since it is the delivery vehicle for the service provided by performance management. The report architecture is based on SAS and the delivery architecture on the Web. The Web through the Internet/Intranet provides the user input and output for the selection and delivery of information. Use of the Web gives the flexibility, accessibility and usability features required for rapid deployment of performance information.

Within the Minneapolis office all service delivery components, such as workstations, servers, WAN/LAN, routers, etc., have well defined naming conventions. These conventions define the type of component, where it is located, and it's function. If the component is a connection device such as a router interface, the name identifies where it is and what it is connected to. Standards provide not only identification, but also allow for group reporting (i.e. all database servers in Dallas) using basic SAS or SQL wild cards. Since this architecture is also deployed to clients who may have little or no defined naming conventions, the implementation also supports an alternate method. In the absence of a consistent naming structure an optional feature provides the capability to associate components such as file servers, database servers, routers, etc. into meaningful named groups and or sub-groups (i.e. MINNEAPOLIS_FILE_SERVERS). Selecting a report for a given group is equivalent to selecting all components of that group from the SAS based reporting architecture.

The major goal of the reporting architecture is to provide a **common integrated interface** for creation and access to information. Even though there are several diverse sets of collection tools from multiple vendors having multiple formats, the **harvesting of all information into SAS datasets provided the basic framework for an integrated access and reporting facility.**

To provide a consistent interface to the selection and generation of charts and reports **a single report utility** was developed. This utility shown in figure A below validates the users input and then automatically generates the required SAS code to process that request. Three basic output formats provide for gif or postscript charts, text reports and column formatted ASCII files. The ASCII output can be input directly to spreadsheet applications such as Microsoft Excel. Users may initiate the utility through a Web interface, as a line mode interactive command, or via a batch command. In batch mode all parameters are specified on the command line and the output routed to on-line (files, workstations, etc.) or off-line (printers, plotters, etc.) devices. Interactive mode is supported through direct line output/input or via a menu driven Web interface using HTML (hypertext Markup Language). The HTML form provides a simple and direct GUI (Graphical User Interface). In line mode the user is queried for the category, type, day or days and the specific chart or report format desired. In Web mode this information is entered via multiple menu and box input selections. This HTML GUI based input is then passed to a CGI (Common Gateway Interface) program to be processed. The CGI program validates the input and then issues a batch mode call to the report utility. Output is written to disk and also returned back to the user via a dynamically built HTML response page. This page may contain text as well as chart based information. A menu

definition file defines the category (i.e. CPU, Memory, Network, etc.) and type (i.e. CPU Queuing, CPU utilization, etc.) of information that will be presented to the user through the utility. Clients may customize their menus to present only options necessary to that installation. This minimizes the complexity and confusion that may occur when more options and reports are presented then are necessary.

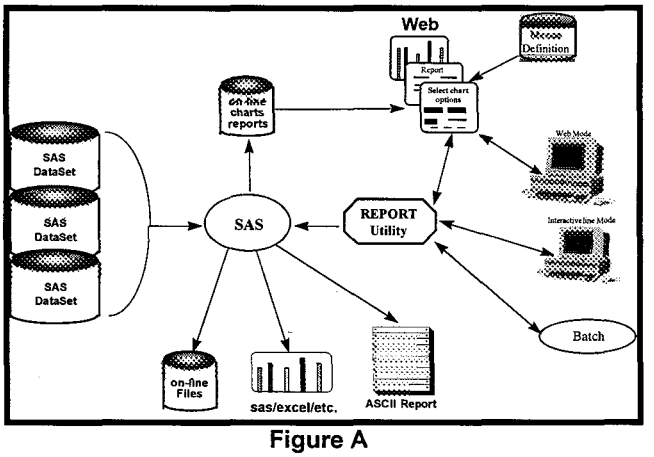


Figure A

Using the report utility, several pre-defined reports and charts are automatically generated each night and made available to the management and operations each morning via the Web. This information, usually in the form of charts, presents the previous days application, server and network performance metrics together with weekly, monthly and year to date trend information. If greater detail or "drill down" is required the user simply initiate the GUI based Web or the interactive line mode utility to enter their specific ad hoc requests.

The use of a single utility and the Web provides a consistent, quick and easy access, together with a high level of flexibility. It rarely takes longer then a few minutes for new or modified chart or report to be added to the menu and daily Web page. Use of the Web as an Executive Reporting System (ERS) also provides for local as well as remote access from a variety of different workstations (PC, UNIX, Macintosh, etc.).

Charts are the preferred reporting format since a significant amount of information can be provided in a single chart. As in the examples shown in appendix A, six types of charts are directly supported: whisker, spectrum, exception, bar, line and time span. Exception charts are a form of spectrum charts except that an exception chart is time based only and presents only two conditions, red for an exception and green for no exception. Whisker charts are used to show the distribution of data since averages rarely provide the true picture of sample based data. Spectrum and bar charts are used primarily to show load level or resource utilization relationships between multiple components or a single component over several days. Line charts are ideal for presenting detailed time based information. Lastly time span charts were implemented using annotated graphics and provide a visual overview of job or process turnaround times. Users through the report utility may select the type of chart or charts and how the information should be represented. Using the GUI interface, various chart formats can quickly and easily be explored until the most effective representation of the information is found.

SUMMARY

Performance and event management are becoming increasingly important components of the "infrastructure" that **must** be in place to support Client/Server environments. They are highly visible pieces required for the success of any client/server operation. Managing performance begins with understanding the requirements of the end users and then developing the architecture to collect and report the data that will provide the necessary information.

The implementation of client/server performance management in any environment small or large is a highly dynamic process. Client/server environments by nature continually change and information provided from performance management itself feeds those changes. Tools and processes developed to support performance management must therefore be extremely flexible to support easy and rapid changes.

In a fully-distributed client/server environment with multiple components from multiple vendors, there has yet to emerge a product which provides a fully integrated collection and reporting architecture. The overall architecture just described is driven with that in mind. SAS front ended with a flexible reporting tool became the base for this integration. It provides an integrated access to information, easy to use, customizable and extendible. The relational capabilities of SAS provide powerful analysis techniques required in client/server environments where multiple cooperative components continually interact. In addition the basic foundation of SAS brings the capability to employ the entire set of data management, statistical, analytical and reporting capabilities provided from the various SAS products such as SAS/CPE, SAS/INSIGHT®, SAS/STAT®, etc.

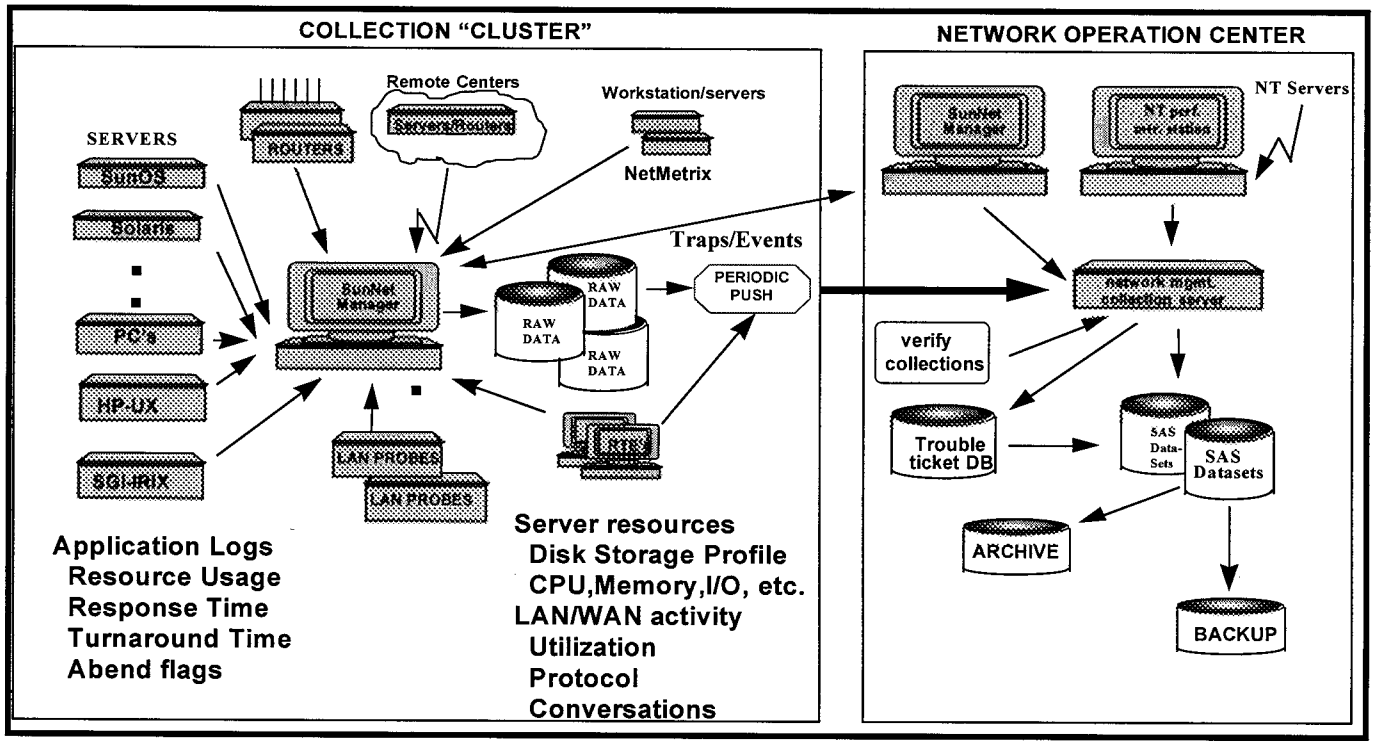
ACKNOWLEDGMENTS

The following people helped contribute to the implementation of this architecture and preparation of this paper.

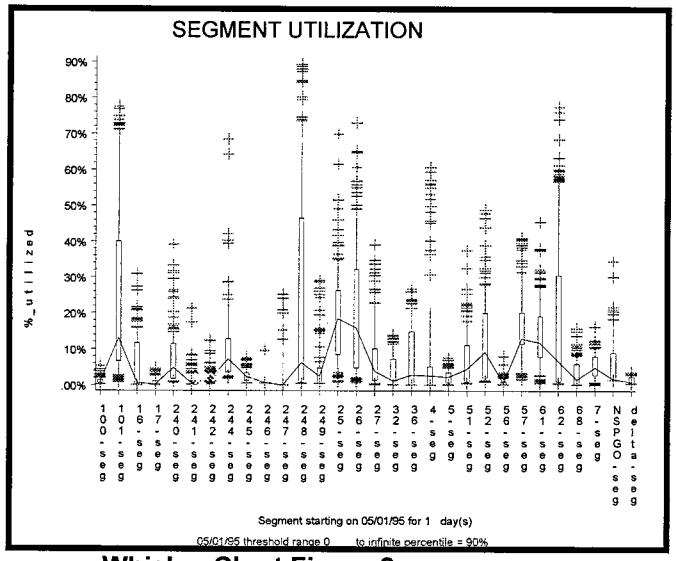
Robert K. Newman Wendy J. Haney
J.P. Young

Tom MacFarland, 333 South 7th street, Minneapolis, Mn. 55402-2460
Tom.MacFarland@cscoe.ac.com

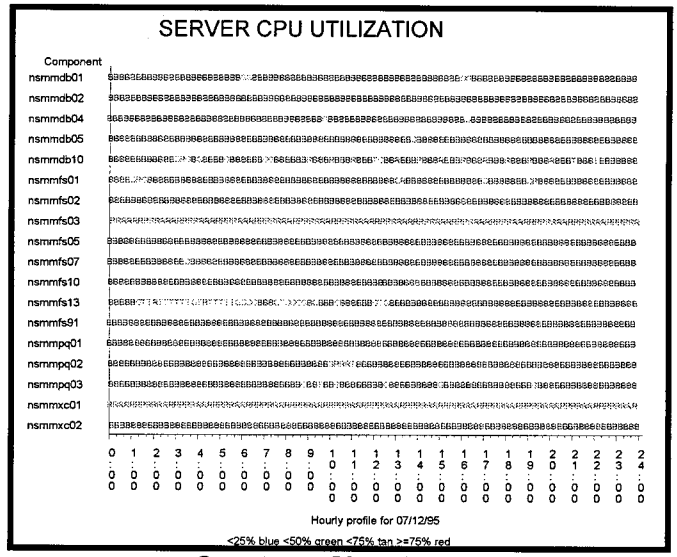
APPENDIX A



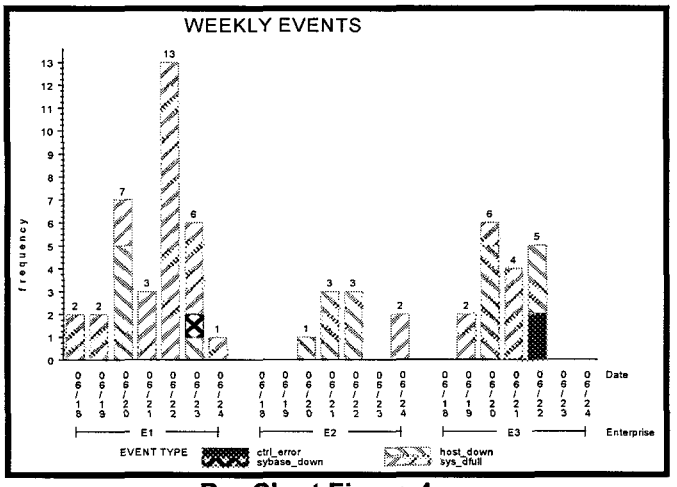
Date Collection Architecture Overview Figure 1



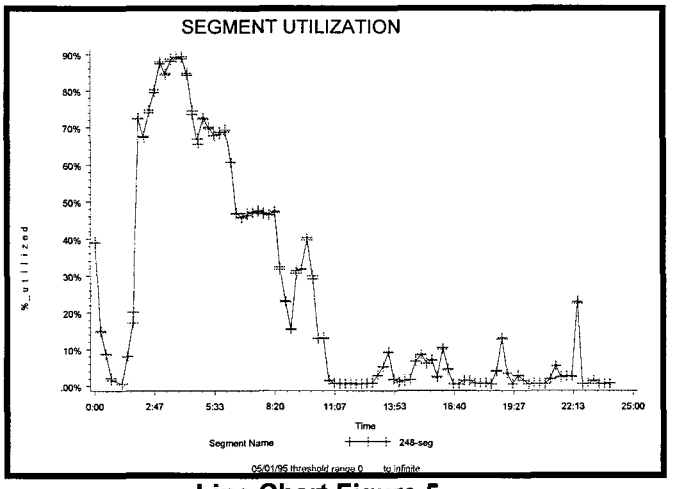
Whisker Chart Figure 2



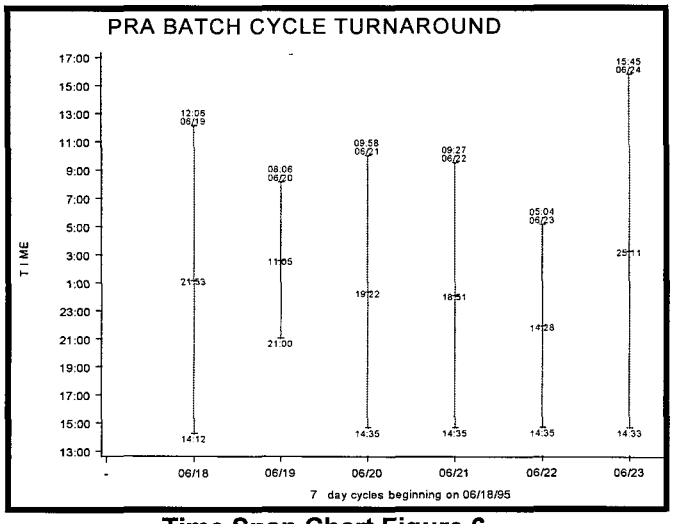
Spectrum Chart Figure 3



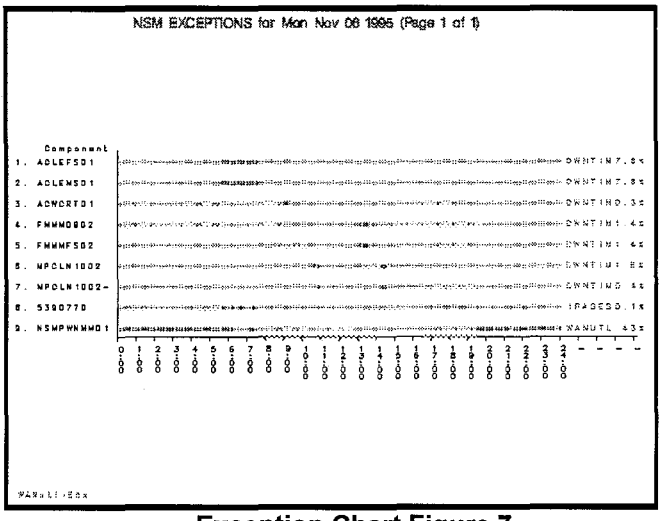
Bar Chart Figure 4



Line Chart Figure 5



Time Span Chart Figure 6



Exception Chart Figure 7