A Methodology for Computer Performance Evaluation in the Apollo* Domain/OS Environment

Robert N. Bonham, SAS Institute Inc., Cary, NC

ABSTRACT

This paper describes a quantitative method for performing Computer Performance Evaluation (CPE) on the Apollo* token ring network. This method describes the monitoring, analysis, and tuning of a dynamic, tightly coupled, Local Area Network (LAN) using base SAS® software, SAS/GRAPH® software and the SQL procedure. As LANs grow, quantitative analysis is becoming essential to target future workstation acquisitions intelligently and to improve network performance and end user response time.

INTRODUCTION

Large homogeneous Local Area Networks (LANs) of hundreds of workstations are becoming increasingly common as as a cost effective solution to corporate computational needs. However, the discipline of workstation Computer Performance Evaluation (CPE) has not been developed enough to adequately analyze such large, complex LANs.

The principal design goal of workstation CPE is to provide network and system administrators the data necessary to manage and cost analyze a tightly coupled workstation network. To accomplish this, an historical performance database must be established to record the normal system state of affairs. A method must then be developed to process, analyze, and generate meaningful reports based on this performance database. This paper presents my experimental workstation CPE method and database design to analyze an Apollo LAN.

MOTIVATION

The need to develop a workstation CPE capability is driven by the amount of company resources committed to the workstation environment. For a small LAN (less than 50 workstations), an adequate performance evaluation can be obtained by gaining a thorough understanding of how the user community uses its network. In a large LAN, this becomes increasingly difficult as both the applications and the number of users grow. Consequently, a CPE methodology is essential to provide the necessary information to intelligently target future workstation acquisitions which in turn will improve network performance and end user response time.

BACKGROUND

Traditionally, CPE analysis has been broken into three components.

- Network Analysis:
  - Network Traffic
  - Hardware Errors
  - Performance Errors
  - Routing Statistics
  - Network Resource Availability
  - Network Exceptions Counts
- Operating System (OS) Analysis:
  - CPU Usage
  - Disk Activity
  - Network I/O
  - Virtual Memory Usage
  - Physical Memory Usage
- Resource Accounting:
  - Transaction Throughput
  - User Activity (Login/Logout)
  - User Response Time
  - Application/User Resource Consumption

In the following analysis, I will add an additional CPE component, which records the availability of objects within the network, Network Resources Availability.

I will not cover the analysis of Resource Accounting records, the third component of a CPE. Although these statistics represent one of the most significant parts of CPE, the current workstation OS implementation does not support the recording of Accounting transactions in a form suitable for automated analysis. Refer to **Future Development Areas** later in this paper for further discussion.

There are, however, several techniques available to aid in relating the aforementioned CPE components to transaction throughput or user response time. For example, one may use Berkeley SA Accounting Software (Tornheim and Williams 1987), the Apollo program Netlog, or one can model a typical user application, replay the recorded work application, track the changes in transaction throughput, and then compare the results to a chosen baseline system.

INTERNAL LAN DESCRIPTION

At SAS Institute, the primary software development environment consists of 337 Apollo workstations running Aegis OS SR9.7 or DOMAIN/OS SR10.1. The main network architecture is a 12 megabit per second baseband token passing ring (Leach et al. 1983) which connects three separate buildings. The internal configuration uses a "star-shaped ring" (Saltzer, Clark, and Pogran 1981) topology with approximately 500 Apollo DQC "quick connect" boxes and 115 Apollo network switches to enhance network reliability. The total cable length (one circle around the ring) is estimated to be 25,000 feet.

A breakdown of the Apollo LAN hardware configuration is shown in **Figure 1**.

DATA COLLECTION AND REDUCTION ARCHITECTURE

The first step in creating a CPE analysis is to decide what aspects of the computer system and LAN need to be characterized. Once this is determined, individual performance parameters can be collected for later analysis.

The second step requires the creation of a performance database. This database combines network diagnostic tools, log files, and a user-defined configuration file into a standardized set of validated performance parameters.

I chose SAS software, Release 6.06, to process and analyze the collected performance data. It was chosen primarily for its complete spectrum of data reduction and presentation capabilities. The following is a list highlighting some of the features used:

- Data Compression
- Database Indexing
- SQL Query Capability, Including SQL Views on the Database
- Tabulate Reports
- Plotting (Both Line and Graphic)
- High Resolution Graphics.

Data Collection

The Apollo Domain operating system provides a number of facilities to record LAN and OS performance data (Apollo 1988). Appendix A provides a listing and brief description of these facilities. I chose to combine the NETMAIN_SRVR performance log file and the NETSTAT -CONFIG report (Apollo 1988), along with a user-defined configuration file, into a LAN performance database.

The NETMAIN_SRVR collects and stores network performance statistics per workstation for a given token ring. It runs as a background server process and continuously logs the specified performance parameters sent to it by each workstation in the ring into a circular binary log file. A circular file is defined as a file having an absloute physical size that, when exceeded, will overwrite the earliest recorded information thereby maintaining its allocated size but having no true beginning or ending location. The NETMAIN server options file in use is shown below.

Note: I do not allow the log file to wrap and become circular. A shell script is run each day to change the name of the log file in the -LOG command by stopping and restarting the NETMAIN_SRVR program.

```
-LL       15000
-LOG      /cpe_scratch/netmain.1989-07-15
-NAPPEND
-SAMPLE est_topology      04:00:00
-SAMPLE topology          04:00:00
-SAMPLE hw_fail           never
-SAMPLE time_skew         never
-SAMPLE memory            never
-SAMPLE paging            never
-SAMPLE cpu_time          00:20
-SAMPLE net_service       00:20
-SAMPLE disk_errs         00:20
-SAMPLE err_counts        00:20
```

The second tool utilized is the NETSTAT -CONFIG command (at Domain/OS version SR10.X, the command name was changed to NODESTAT to prevent confusion with the Berkeley NETSTAT command). This command, when executed with the -CONFIG option, displays a report of node-specific hardware information: CPU type, display type, disk type, and so on (see below). The information received from each workstation, at 20 minute intervals, is then combined with the netmain performance log file by use of the unique Apollo workstation nodeid. The resulting performance database contains both the NETMAIN activity logs and the workstation configurations at the time of collection.

The following is a NETSTAT -CONFIG report for Workstation HOBBES:

```
**** Node 176B9 ****  //hobbes
Time 1989/08/09.11:06:28  Up since 1989/07/21.14:19:05

System configured with 8.0 mb of memory.

  NODE CONFIGURATION
    Node Type:  DN3500
    Display type:  1280 x 1024 monochrome display
    68882 Floating Point Unit present.
    Peripheral configuration:
        Disks:  winchester
        Networks: Ring
        Peripheral bus:  AT-bus
        Tapes:  none
    Disk types:  MSD-170M
```

The final database component is a user configuration file containing site-specific information about each workstation on the network. Typically, user, department, and location information are needed to facilitate the data organization and results presentation. A sample configuration file is shown below:

```
ID     NODE_NAME  LOOP  ROOM  USER          DEPT SERVICE  SER_NUMBER
-----  ---------  ----  ----  ----          ---- -------  ----------
 3E17  JUMBO      E1E1  100   Backup        SSD  9.7      A00131133
19153  SNOOPY     E1E1  100   DC            ORD  PFS      A00261613
 41AB  CLYDE      E2E1  208   10.1 Testing  SSD  SR10     A00239280
1838A  TOSCA      E2E1  207   John Smith    SSD           EA84503320
 F615  DL_F615    J2E5  209   Ralph Wood    QA            A00197362
1C166  LEAR       J3E3  315   Jim Simmons   MCD           EA84901753
1B6BE  PADDY      J3W1  349   Ted Jones     ASM           EA84602168
17D66  DL_17D66   J3W4  383   Robert Brown  ASM           A00236135
176FD  EUNICE     J3W4  382   Ellen Wilson  PC            A00234049
```

Performance Database Organization

A critical aspect of performance database design is the ability to compress and summarize the performance data and still retain useful information in a condensed form.

A performance database should retain detailed information for ten to thirty days and summarize the information into various time-spans ranging from periods as short as ten or fifteen minutes, to monthly, quarterly, and yearly intervals (Cheeseman and Robinson 1986). For a LAN, the resulting amount of OS and network information can easily overwhelm most data processing and presentation techniques (the current implementation records three million discrete pieces of information a day). With this in mind, I chose the following file summarization process to age and retain performance data:

- Collect detailed OS and LAN information from every workstation every twenty minutes.
- Retain detailed information for a period of thirty days.
- Classify this detailed information into three shifts.
  - Prime Time
  - Off Hour Shift
  - Weekend Shift

- Summarize detailed information into workstation and network trend databases by shift and day, and retain for 90 days.
- Summarize daily shift information by day into monthly shift levels and retain for thirteen months.

Data Reduction

Several types of performance records can be recorded by an OS: a record containing counts over a specified time period, a record containing absolute bucket counts, or sample interval records. Each record type has data recording problems which must be addressed when collecting and processing the information. The DOMAIN/OS architecture and the NETMAIN_SRVR tool record all statistics using the absolute bucket counts. These counts require the processing code to handle several types of data integrity problems: bucket overflows, workstation shutdowns, and inconsistent data recording.

An additional multi-workstation data-reduction problem was also encountered: time skew. When a central data collection server requests information from three hundred autonomous workstations, the received data's sample date-time can be skewed as much as three minutes. In response, an algorithm was developed to scan the entire raw database and extract a unique, sample date-time event based on a given tolerance. Or, in otherwords, a process analyzes all date-times in the database and selects a subset of these based on a specified absolute difference. A separate step then reprocesses the performance database files, replacing the OS recorded sample date-time with the database unique sample date-time. This addition of a normailized sample value allows network and workstation workloads to be compared to each other without the scatter of a few seconds in the collection date-time.

With consideration to the data reduction problems discussed earlier, SAS code was written to interface directly with the binary NETMAIN_SRVR log file in a three phase process. Phase one reads the performance record exactly as recorded. Phase two extracts and reinserts a "fuzzless" unique date-time sample into the performance records as described above. Finally, phase three transforms the bucket counts into per second rates or dimensionless quantities, depending on the data type, and addresses any data inconsistency including workstation shutdown, bucket overflow, and inconsistent accounting records. A dataflow of the processing stage is shown in **Figure 2**.

After completing the reading and processing of the raw performance records, I used the SQL features of SAS software to combine the user-defined network configuration file and the NETSTAT -C hardware config file with the processed performance records. Two indexed SAS data sets are produced. The first is a twenty minute interval performance record database indexed on the Apollo nodeid and the workstation CPU type. The second data set has one observation per workstation and contains both the hardware and user-specified configuration files. SQL views are then created to allow easy access by combining both data sets by keying on the unique indexed workstation nodeid (see **Figure 2**).

REPORTING SYSTEM ARCHITECTURE

The primary goal of the reporting system architecture is to produce meaningful reports, while minimizing the number of report pages produced on a daily basis. At minimum, the reporting system should be required to display the following components:

- Network Traffic Time Plots
- Network Exceptions and Failures
- Workstation Usage Trends
- Workstation Exceptions and Failures

Furthermore, the unique characteristics of prime, off-hour, and weekend usage lead to the separation of reports into different time-dependent categories.

Figures 3 and **4** are graphs showing a four day average of global network traffic trends. **Figure 3** displays the total network traffic for the previous three days. **Figure 4** displays any network hardware or performance errors that occurred during the same time period based on a percentage of the total network traffic. Hardware errors are classified as any errors associated with the loss or corruption of the network carrier signal. Performance errors identify the loss of a packet due to excessive traffic loads into or out of the workstation.

The table below presents the offhour shift's ten heaviest utilized workstations on the network for the previous two weeks. Each workstation has been selected by Z-statistic scaling (Artis 1987) and weighting of three workstation performance parameters: mean CPU Usage, mean Net I/O, and mean Disk I/O. To Z-scale an observation, subtract the mean from each observation and divide the results by the standard deviation for the metric. Z-scaling yields a normalized value, ranging from -4.0 to +4.0, for each observation. A Z-scaled value of 0 indicates the observation was identical to the mean.

Workload Function of 50% CPU Usage, 30% Network I/O, & 20% Disk I/O
Offhour (12:01a.m. to 9:00a.m. and 5:00p.m. to 12:00a.m.)

ASSIGNED NODE NAME	DEPART -MENT	WKSTATION OR CPU TYPE	PHYSICAL MEMORY	Work Load Function	Total CPU Usage	Total Net I/O	Total Win I/O
FOZZIE	ASM	DSP4500	16	2.97	3.95	203.3	17.48
MSPIGGY	ASM	DSP4500	16	2.57	4.00	188.2	14.68
TIGGER	ASM	DSP3500	4	2.54	3.17	178.3	15.23
ERNIE	CD	DSP4500	16	2.49	5.75	288.8	5.34
MERLIN	CD	DSP3500	4	2.14	2.85	167.4	12.03
KERMIT	ASM	DSP4500	16	2.14	3.56	180.4	10.85
COOKIE	PC	DSP4500	16	2.03	3.14	141.5	13.03
ELVIRA	QA	DN3000	8	1.93	2.65	58.5	19.00
CHEF	SSD	DSP4500	16	1.91	3.36	126.6	13.03
RERUN	SSD	DSP3500	8	1.31	1.96	92.7	9.74

The Z-scaling procedure allows a weighted peak interval, consisting of 50% CPU Usage, 30% Net I/O, and 20% Disk I/O, to be calculated automatically over the performance database and thus reduces the number of performance tables requiring examination. The table above was produced by sorting on the Z-scaled statistic.

After the examination of traffic and utilization trends, a concise and efficient procedure for detailed data analysis and reduction is necessary. Of prime importance is the pinpointing of excessive net I/O, the locating of resource utilization hotspots, and the exclusion of lightly loaded workstation performance data. As part of the design, it was considered unacceptable to produce even a single report for every workstation per day. This criterion results in a limit filter being written to scan the performance database and flag any observation exceeding a threshold set for it. In the second phase, relevant performance data is extracted, and a detailed series of plots describing the state of the workstation for the entire 24 hour period is produced.

Figures 5, and **6a**, **6b**, **6c**, and **6d** consist of a sampling of the detailed workstation exception reports for a single workstation, McCoy. **Figure 5** presents a tabulate chart analyzing the number and class of OS exceptions by workstation type and memory size. As presented in the table headings, this table reports any OS exceptions that have occurred during any twenty minute sample interval for the past 24 hours. Each are detailed below:

CPU_Usage
> 99.8% utilized

Network I/O
> 375 per second

Disk I/O
> 40 per second

Network Paging Server Overflows
> .001 per second

Network Remote File Server Overflows
> .001 per second

Received or Transmitted Network Performance Errors
> 2 % of total I/O

Received or Transmitted Network Hardware Errors
> 2 % of total I/O

Network Transmit Wait Acknowledgments (Wacks)
> 50 % of total transmit I/O

Network Transmit Negative Acknowledgments (Nacks)
> 50 % of total transmit I/O

Figures 6a, 6b, 6c, and 6d present several graphs of the detailed analysis per workstation. The following OS parameters are summarized by the detail reporting architecture:

- CPU usage (**Figure 6a** and Case Study 1)
- Net I/O (**Figure 6b** and Case Study 1)
- Remote Network file server and Paging server overflows (**Figure 6c** and Case Study 1) (Leach et al. 1985)
- Disk I/O (**Figure 6d**)
- Hardware, performance, and resource errors (Not Shown)
- Remote server paging activity (Not Shown)
- Network Transmit Waits and Receive Slow Paths (Not Shown)

INDIVIDUAL CASE STUDIES

The next two sections present and discuss two case studies of how the current CPE database was used to analyze two actual CPE related problems.

Case Study 1: Remote Network Request Server Overflows

A principal product of SAS Institute is software written in C. This software requires nightly rebuilds to support and couple the ever-changing software components. Failures of these rebuilds, especially when resulting from network or OS system problems, are considered unacceptable by the user community. Unfortunately, "My overnight rebuilds failed because of network problems," is a common user complaint.

This problem has not been easy to analyze or correct and provides very few clues as to what is actually happening. In response, I made a request to all major code builders to log the exact system error message given at the time of build failure. Once this feedback had been collected, several rebuild error message trends began to emerge and are listed below:

- A source file can't be accessed.
- An output file can't be opened.
- "Ghost files" have been created, files which the OS will respond to with the message "Attribute unavailable" if a user or program attempts to access them.

Any attempt to replicate the rebuild failures consistently or to extract and reproduce a failing component of the rebuilds was unsuccessful. Therefore, a detailed analysis was needed, and a single department was chosen for further study. The analysis would be accomplished by a detailed examination of the performance data records at the time of an actual rebuild failure.

On Thursday, the Compiler Department called. There had been rebuild failures overnight. The graph in **Figure 7** presents a network trend graph showing all remote file and OS Paging overflows that had occurred. Notice that at 4:00 a.m. on August 9, the File and Paging Queues overflows had spiked to over 5 I/O per second. Notice, also, that overflows of the File and Paging Queues had only occurred between 1:00 and 3:00 a.m., on Wednesday.

Examination of the daily exception chart, **Figure 8**, shows workstation BIGBIRD, the Compiler Department's DSP4500 disk server, to have recorded File and Paging queue overflows during the last twenty-four hour period.

Figures 9a, 9b, 9c, and **9d** present components of Wednesday's twenty-four hour detailed analysis for the workstation BIGBIRD. **Figure 9a** presents a summary of user and system CPU usage. **Figure 9b** shows network traffic including receives and transmits. **Figure 9c** presents any overflows of the remote file and paging server queues, and **Figure 9d** records disk reads and writes.

Notice the correlation of **Figures 9b**, Network I/O, and **9c**, Remote File Queue overflows, at 4:00 a.m. Notice, also, a high (greater than 95%) system CPU utilization as shown by **Figure 9a**. Although still unconfirmed by Apollo Computer Inc., I felt there was significant evidence to support the theory of BIGBIRD being an over-worked file server, and decided to redistribute some of the heaviest utilized resources to other workstations in the department.

In summary, **Figure 10** again presents the Network I/O for the workstation BIGBIRD after resource redistribution for an entire week. The I/O rate is equivalent to that in **Figure 9b**, but no File Queue overflows have occurred on the workstation, see **Figure 11**. These findings were consistent for several months.

Case Study 2: Workstation Restocking

As an administrator and performance analyst, my primary focus has been the determination and prediction of overutilized network resources. In a large LAN, there may also be an equal number of underutilized workstations. These machines are wasted company resources, and their relocation is a cost effective alternative to the purchase of additional hardware. To address this problem, the following selection criteria were used to recover ten workstations from the primary development network.

- All development ring workstations were monitored for CPU usage, Disk activity, and Network activity during prime time (9:00 a.m. to 5:00 p.m.) for two months. They were then sorted by total mean CPU usage. The lowest 15 were selected as candidates for possible relocation.
- Any workstation that had changed owners during the 2 month period was excluded.

- Any workstation being used as an SR10.1 test machine was excluded.
- Any workstation showing high Network or Disk I/O activity, when compared to the mean Network or Disk I/O activity for the 15 lowest workstations, was further screened to understand which resource or application was causing the abnormal activity.

As a result of the fourth criterion, two workstations were examined further. Both disks contained over 10 user accounts, and I felt a greater than average Disk I/O could be expected.

Workstation CPU Usage

NODE	User	Dept	Type	CPU USAGE (%) * Total	System	User
DL_8454	Core Lab (PSVR)	CD	DN3000	0.64	0.3	0.3
DL_D31C	Beth Stone	CD	DN3000	0.78	0.3	0.5
HOOKIE	Gene Davis	DBI	DN3000	1.21	0.5	0.7
FORD	Tom Dean	DBI	DN3000	0.79	0.4	0.4
BRUTUS	Emily Jones	DBI	DN3000	1.02	0.4	0.6
RENEGADE	Steve Ray	COM	DN3010	1.07	0.6	0.5
BILBO	Kevin Furr	CMS	DN3000	1.06	0.3	0.7
GANDALF	Karl Johnson	IDB	DN3000	1.06	0.5	0.5
GARFIELD	Ken Scott	VMS	DN3000	1.10	0.4	0.7
POPEYE	Greg Brown	VSE	DN3000	0.82	0.3	0.5

* Note: Mean CPU usage for all DN3000 type machines was 5.6 % and the Mean CPU usage for all DN3500 type machines was 7.1 %.

FUTURE DEVELOPMENT AREAS

Vendor OS Enhancements for Transaction, and User Response

Case Studies 1 and 2 represent samples of the type of analysis the CPE method described above is capable of making. Questions such as

- Which disk servers are the most heavily utilized?
- When does the network experience the largest network traffic?
- When the new hardware was installed, did the network experience an increase in packet corruption?

can be answered quickly and effectively. Unfortunately, questions describing what effect a problem or change will have on the user cannot be directly answered; questions such as:

- Who is causing the resource bottleneck?
- Would more memory improve the user response time?
- Is the bottleneck caused by paging or file system activity?

cannot be answered with the current, inadequate set of OS performance tools.

To complete the analysis, what the CPE method requires is enhancements to OS and network instrumentation to provide the performance analyst the capability to trace LAN performance problems. This Measurement Interface (MI) will, at a minimum, contain published data structures and should describe the following workstation resource categories:

- Transaction definition and recording (OS trace records are too excessive)
- User response definition and recording
- CPU information including a breakdown of system usage, user usage, and process waiting lengths
- Paging and subsystem I/O rates, queue lengths, and wait times
- Virtual and physical memory queue lengths
- Network traffic, network performance, hardware errors, and resource errors (include who and how much)
- Internet routing traffic, queue lengths, and overflow information
- Winchester I/O, queue lengths, system disk activity, and disk errors (include who and how much), support of multiple disk servers
- Diskless traffic (who and how much)
- Node hardware configuration specifications
- Process load averages.

Interactive Menu Facility

Development of a user-friendly, interactive menu system using SAS/AF® software is under study. Features such as online help for each performance variable, integration of a charge back system for system usage, canned performance analysis programs, and a step-thru tutorial are under consideration.

CONCLUSIONS

My objectives in writing this paper are to describe a general method for a workstation CPE, to discuss some of the details involved in performing this analysis, and to apply these techniques to characterize and develop an understanding of the complex workstation interactions on a LAN. I have also sought to heighten the workstation vendors' awareness of the importance and marketing potential of workstation LAN CPE and to encourage a commitment to develop the necessary instrumentation to allow a complete LAN CPE capability. I feel a strategic opportunity exists for a market-conscious vendor in this arena.

The methodology presented provides administrators and performance analysts with an effective tool to allow intelligent and cost effective purchase decisions. Both over- and under-utilized network resources can be identified, global trends can be observed, and, perhaps, modeling and forecasting techniques can be applied to further enhance system performance.

In Case Study 1, the method provided a unique and otherwise unattainable view of the network. This allowed the isolation of an overutilized resource and the solution to a difficult user problem. The resolution also generated a feeling of respect from the user community as to the network administrator's capabilities. This respect can only be obtained by having the tools necessary to analyze the problem, to determine its cause, and to provide a plausible solution to the problem at hand.

In gaining an understanding of the workstation network interactions, we also gain insight about what drives them and how they interact. We can't expect to be able to determine how things will change in the future if we don't understand the present LAN interactions.

ACKNOWLEDGMENTS

The author wishes to acknowledge Dan Squillace and Bernie O'Connor, whose guidance, direction, and support make this project possible. I would also like to thank Michele Phillips and Helen Yamamoto for their review of this paper and for suggestions to improve its readability.

APPENDIX

List of Performance Tools on Domain/OS (Apollo 1988):

NETMAIN (NETMAIN_SRVR)
: functions as a server and interactive menu tool to collect and display information about the state of the network. It can be used to analyze individual node performance as well as overall network traffic. Data collected by the netmain server are classified into the following catagories: CPU_time, Disk_errors, Network Error_counts, Hardware_failures, Network Services, Topology Estimation, Paging Activity, and Software Diagnostic Messages.

NETSTAT (NODESTAT)
: displays Network bucket statistics including Network I/O, Disk I/O, Paging, and Network receive and transmit errors.

RTSTAT
: displays internet router information from workstation routing ports including packets routed, queue lengths, and routing destinations.

DSPST
: displays process statistics in a graphical, bar-chart fashion within the current process window. The chart is updated periodically. The default action of this command is to display the brief OS process list, all user processes, and all I/O information in a font size automatically selected based on window size.

PST
: lists internal state information for all processes in the system by name or UID.

/BSD4.3/PS
: displays process state information including CPU time, state, and an indication of which command is executing.

NETLOG
: functions as a tool that can be used to monitor Paging and Segment activity on a system.

SALVOL
: verifies and corrects allocation of disk blocks. It also displays disk allocation parameters and file system characteristics.

/BSD4.3/SA
: collects system accounting information for every process executing on the system. Output per process includes CPU usage, average number of I/O operations per execution, total I/O operations, and CPU storage usage.

REFERENCES

Apollo Computer Inc. (May 1988), *Managing Aegis System Software*, Chelmsfor, MA: Apollo Computer Inc., Appendix A.

Artis, P.H. (Dec. 1987), "A Systematic Approach for Identifying Peak Periods," *Computer Measurement Group Proceeding*, 287-289.

Cheeseman, J.T., and Robinson, E.C. (Dec. 1986), "A VAX Performance DataBase," *Computer Measurement Group Proceeding*, 284-290.

Leach, P.L., Levine, P.H., Pouros, B.P., Hamilton, J.A., Nelson, D., and Stumpf, B.L (Nov. 1983), "The Architecture of an Integrated Local Network," *IEEE Journal on Communications*, Vol. SAC-1, No. 5, 842-855.

Leach, P.L., Levine, P.H., B.P., Hamilton, J.A., and Stumpf, B.L. (March 1985), "The File System of an Integrated Local Network," *ACM Computer Science Conference*.

Saltzer, J.H., Clark, D.D., and Pogran, J.T. (Oct 1981), "Why a Ring," *7th Data Communication Symposium Proceeding*, 211-217.

Tornheim, J.J., and Williams, R.T. (Dec. 1987), "Workload Characterization Using UNIX System Accounting," *Computer Measurement Group Proceeding*, 351-359.

Hardware Configuration Report		WINCHESTER DISK DRIVE						% node count of network	total disk storage mbytes
		Disk-less	MSD-170M	MSD-380M	MSD-500M	MSD-86M	ALL		
CPU TYPE	Memory Size								
DN3000	4	7	1	.	.	1	9	3	256
	7	24	6	.	.	.	30	9	1020
	8	44	82	4	.	6	136	40	15976
DN3500	8	4	96	1	.	.	101	30	16700
	16	.	7	.	.	.	7	2	1190
DN4500	8	.	8	.	.	.	8	2	1360
DSP3500	4	.	.	7	.	.	7	2	2660
	8	.	1	5	.	.	6	2	2070
DSP4500	16	.	.	10	.	.	10	3	3800
DSP90	2	1	.	.	2	.	3	1	1000
ALL		79	201	27	2	7	317	100	46272

Figure 1 APOLLO NETWORK HARDWARE CONFIGURATION REPORT. Breakdown of Workstations by Type, Memory, and Disk (Note: a period indicates there are no machines in this category.)

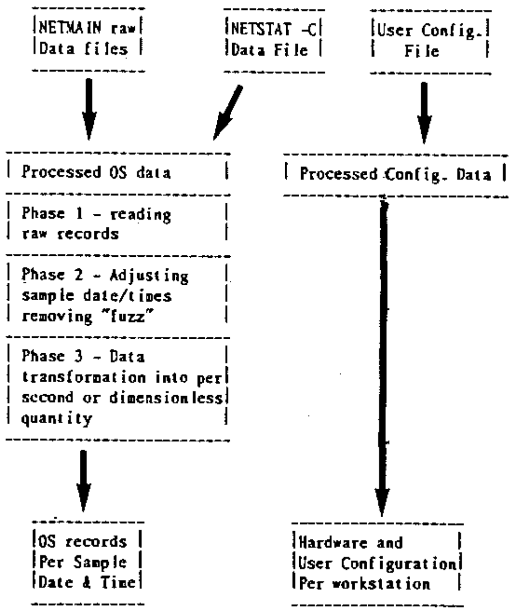


Figure 2 Dataflow of Performance Data Processing System

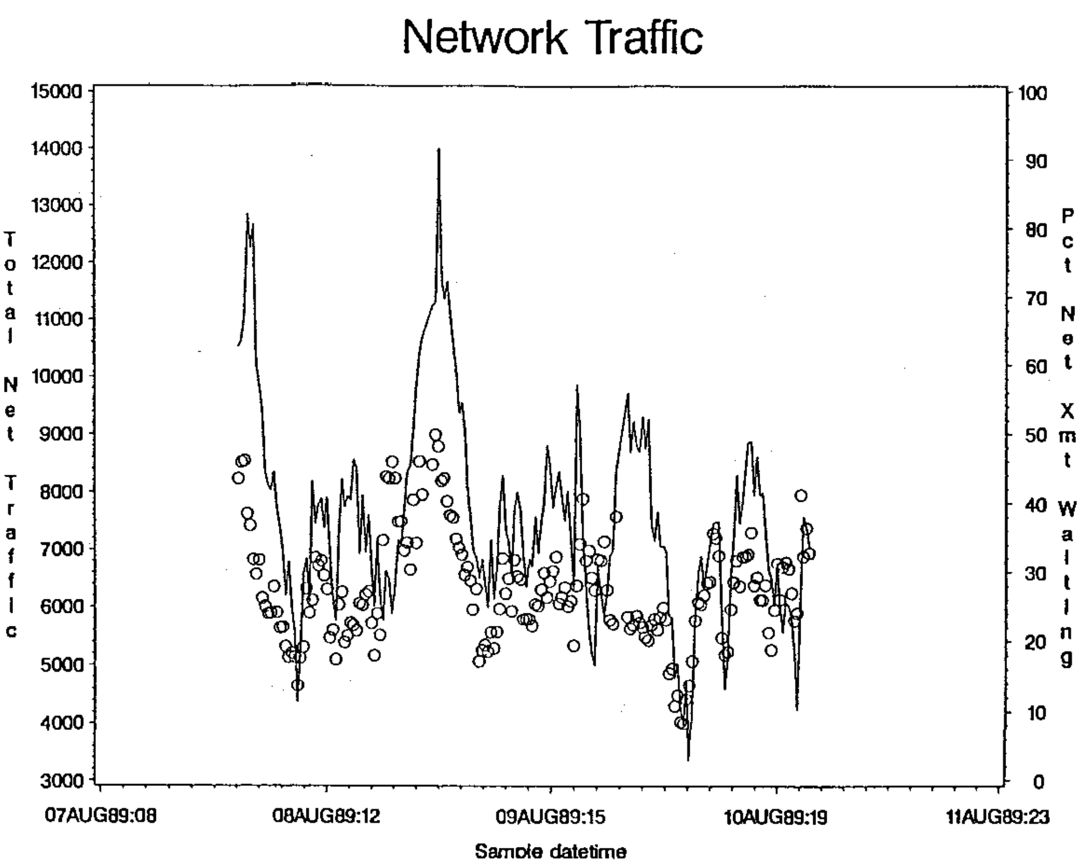


Figure 3

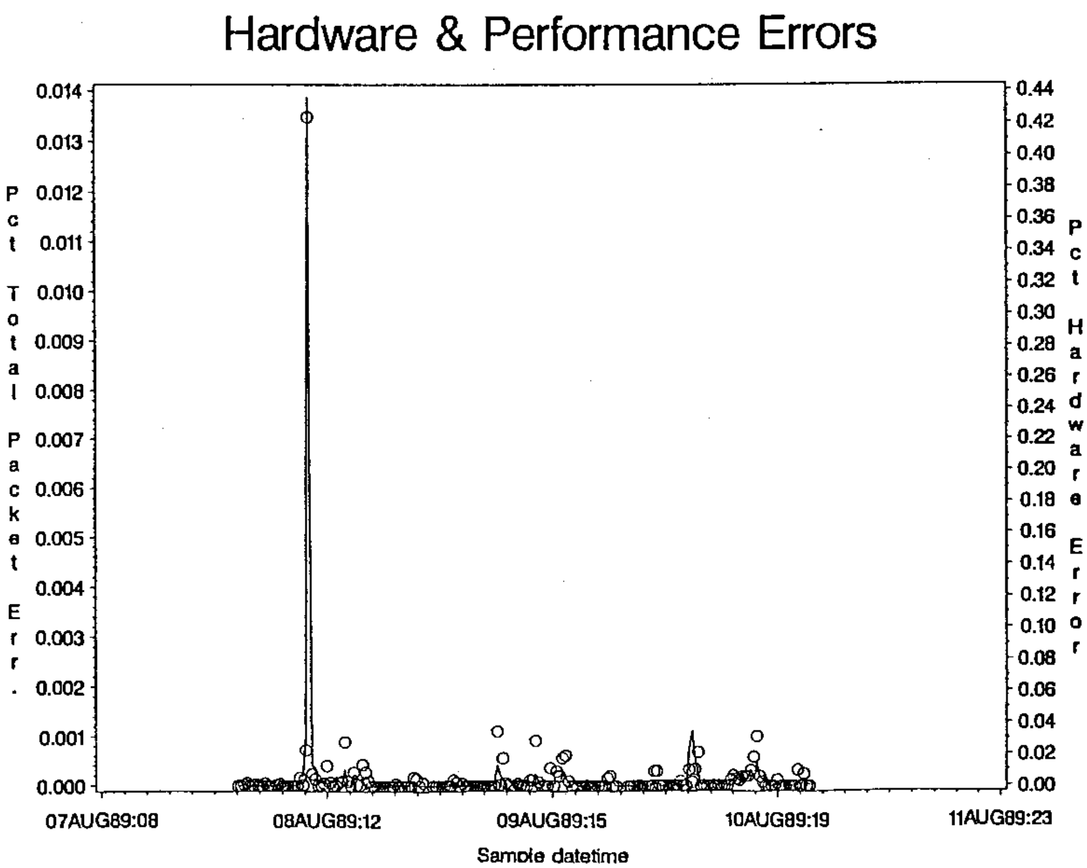


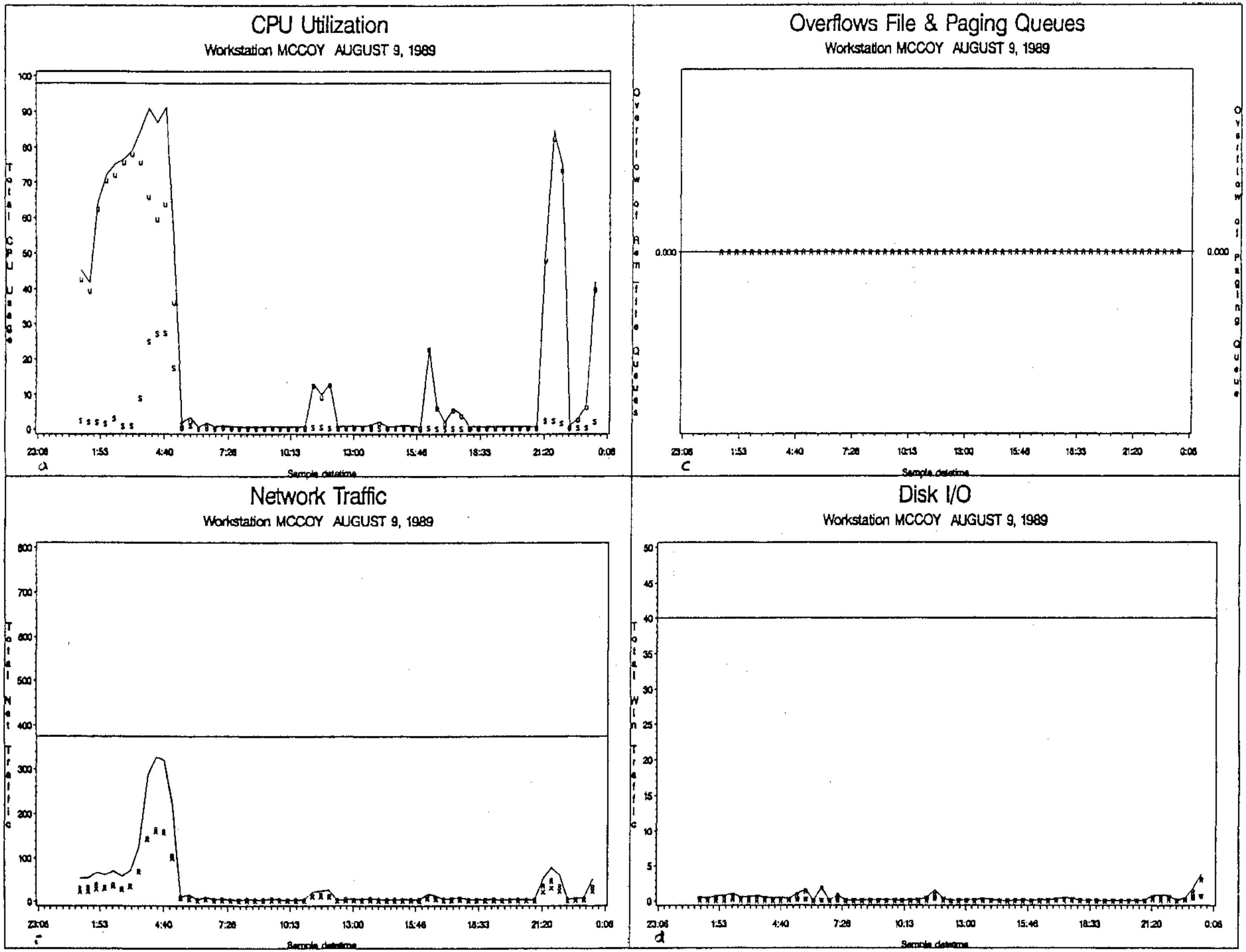
Figure 4

DETAILED DAY ANALYSIS: Logtime 12AUG89:00:36 to 12AUG89:23:47

Exception Analysis by Workstation Type and Memory

Workstation EXCEPTION STUDY: Breakdown by Nodetype and Memory		No. Incidents	CPU Usage 99.8	Net I/O 375	Disk I/O 40	Page of lo .001	Rem of lo .001	Perf Err. 2	Hdw Err. 2	Wack 50	Nack 50
WKSTATION OR CPU TYPE	PHYSICAL MEMORY										
DN3000	4	7	.	.	.	.	.	.	7	.	.
	8	171	.	.	2	9	10	13	94	14	57
DN3xx	2	1	.	.	.	.	.	.	1	.	.
DN3500	8	75	.	.	1	12	1	7	62	4	6
	16	9	.	.	.	.	.	.	6	2	1
DN4500	8	7	.	.	.	1	.	1	5	2	.
DSP3500	4	11	.	5	4	.	.	.	.	2	.
	8	4	.	.	.	2	1	1	1	.	.
DSP4500	16	38	.	18	16	1	9	.	3	1	.
ALL		347	.	23	23	25	21	26	203	25	64

Figure 5 Exception Analysis by a Workstation Type



Figures 6a, 6b, 6c and 6d

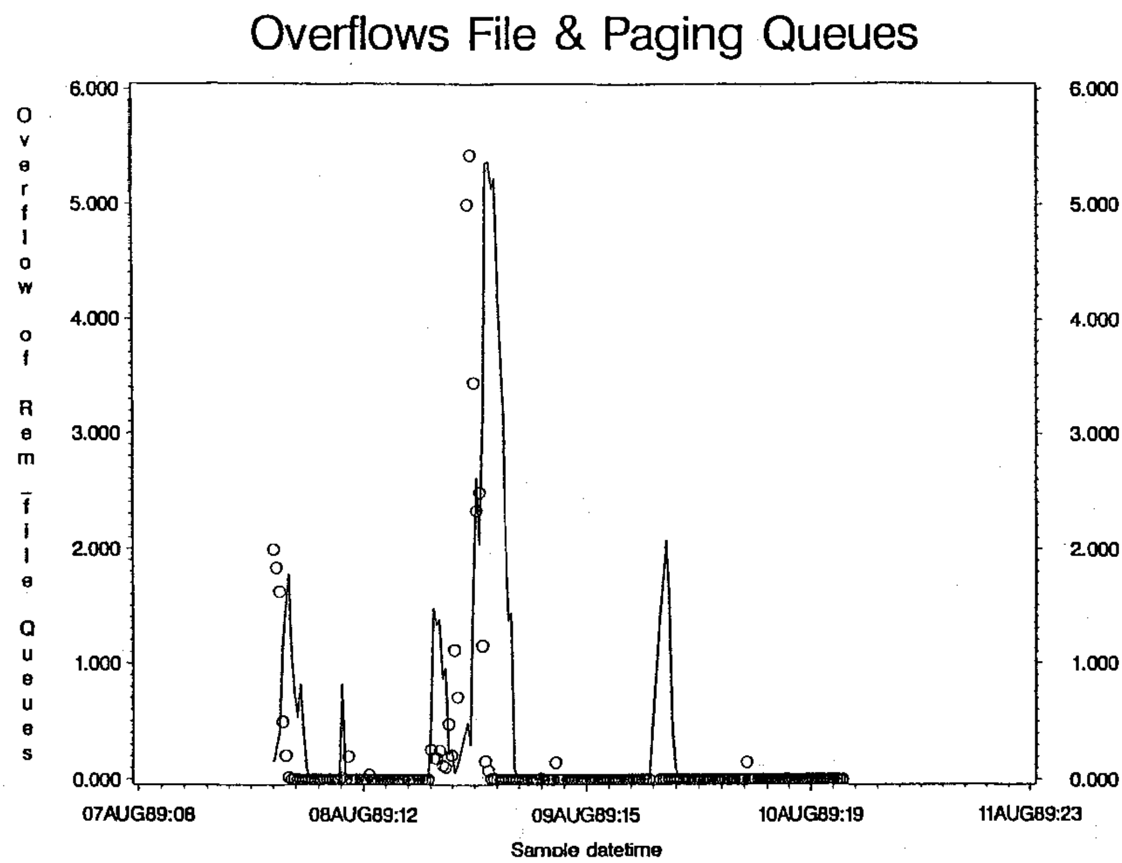


Figure 7

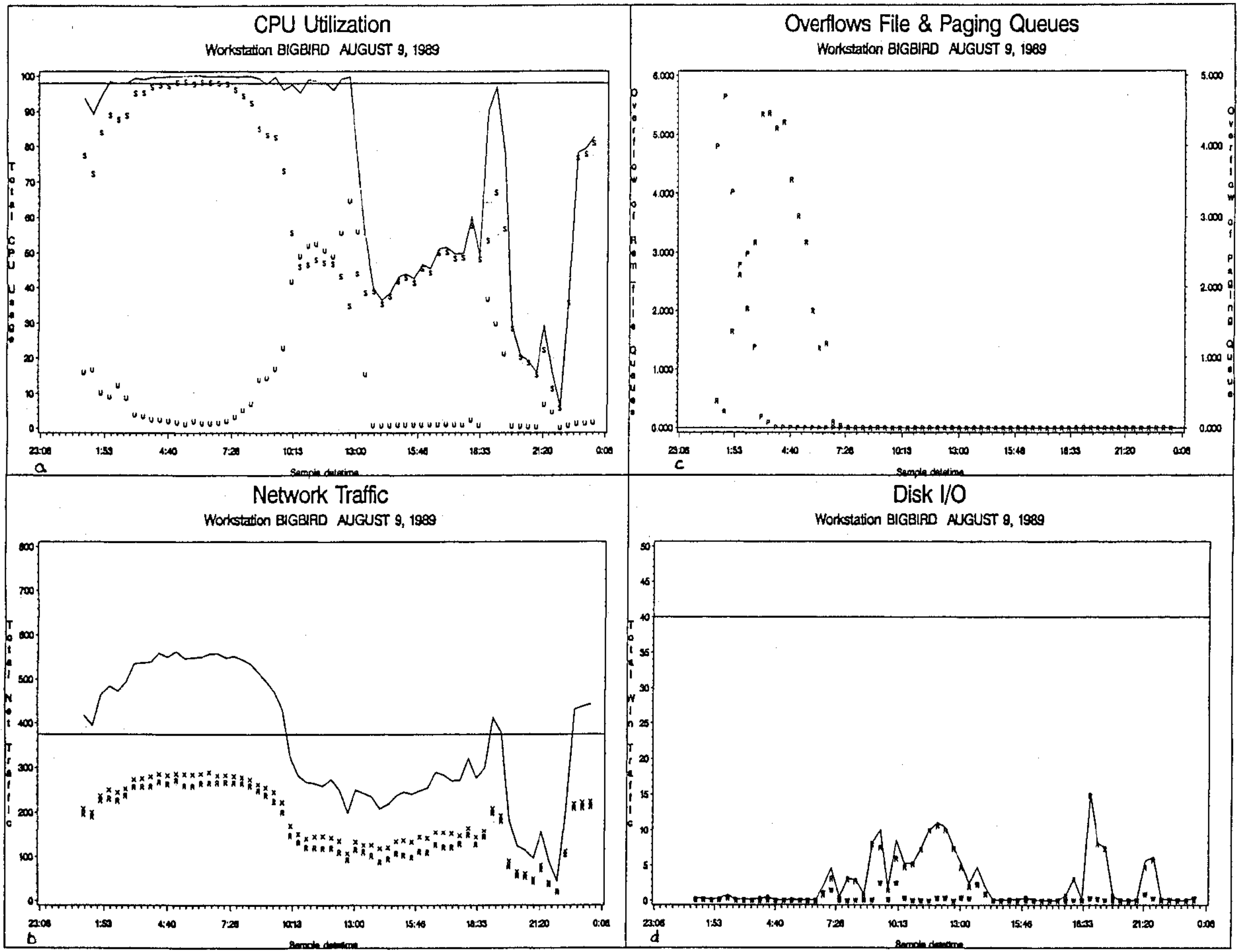
SAS COMPUTER PERFORMANCE EVALUATION for APOLLOS
DETAILED DAY ANALYSIS: Logtime 18AUG89:00:36 to 18AUG89:23:30
Exception Analysis.

Workstation EXCEPTION STUDY: Breakdown by nodetype and Memory		No. Incidents	Total No. Exceptions	CPU Usage (99.8)	Net I/O (375)	Win I/O (40)	Page oflo (0.001)	Rem oflo (0.001)	Perf Error (2)	Hdw Errors (2)
WKSTATION OR CPU TYPE	PHYSICAL MEMORY									
DN3000	4	2	2	.	.	.	.	.	.	2
	7	13	16	.	.	.	.	4	.	12
	8	101	115	.	6	5	3	33	.	47
DN3500	8	72	93	.	10	9	9	32	.	26
DN4500	8	2	3	.	.	.	.	1	.	2
DSP3500	4	16	20	.	.	3	1	10	.	5
	8	20	26	.	.	3	1	16	.	6
DSP4500	16	74	127	.	32	28	25	34	.	8
ALL		300	402	.	48	48	39	130	.	108

Workstation EXCEPTION STUDY: Breakdown by nodename	No. Incidents	Total Exceptions	CPU Usage (99.8)	Net I/O (375)	Win I/O (40)	Page oflo (0.001)	Rem oflo (0.001)	Perf Error (2)	Hdw Errors (2)
ASSIGNED NODE NAME									
ERNIE	13	35	.	8	5	9	12	.	1
FOZZIE	10	18	.	7	3	5	2	.	1
CHUD	18	20	.	.	.	1	2	.	1
BIGBIRD	15	32	.	8	.	10	13	.	1
KERMIT	5	7	.	.	2	1	3	.	1
ANTARES	4	11	.	.	4	4	3	.	.
GODZILLA	12	18	.	3	.	5	9	.	1
KIRK	2	3	.	.	.	.	2	.	1
MONTY	3	4	.	.	.	.	3	.	1
FUDD	2	3	.	.	.	.	2	.	1
ALF	1	2	.	.	.	.	1	.	1
...	...	...	.	.	.	.	.	.	.
ALL	300	402	.	48	48	39	130	.	108

Reported on 18AUG89
Data rate io/second or percentage for cpu usage

Figure 8 Individual Workstation Exception Analysis



Figures 9a, 9b, 9c, and 9d

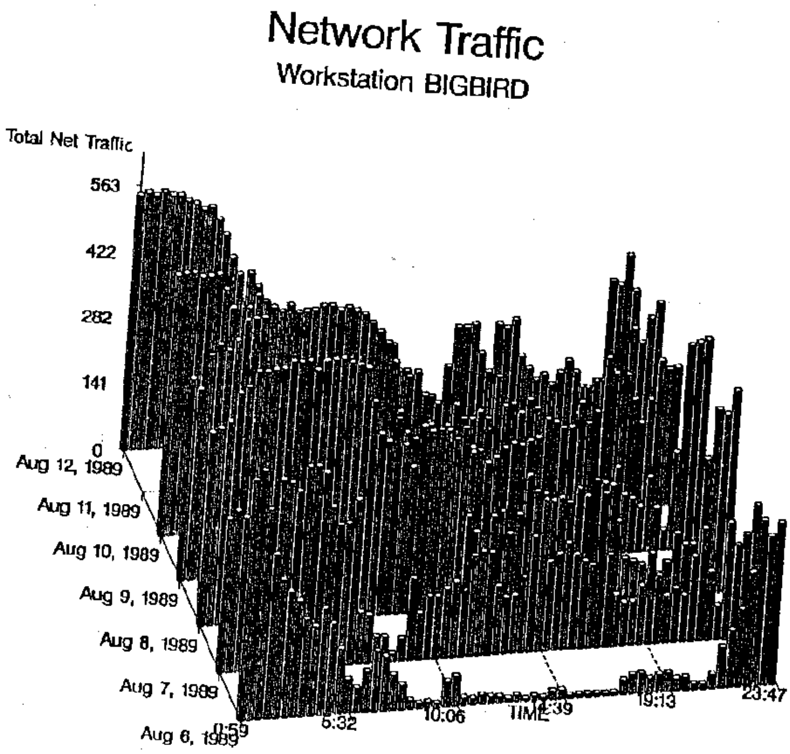


Figure 10

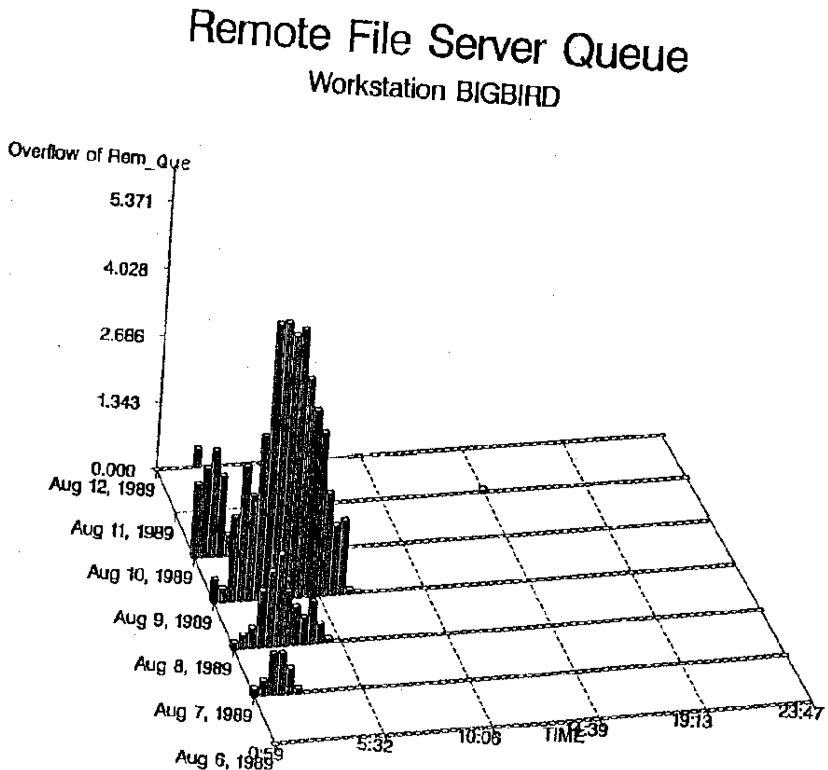


Figure 11