29

AIX

update

March 1998

In this issue

© Xephon plc 1998

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 550955
From USA: 01144 1635 33823
E-mail: HarryLewis@compuserve.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75067
USA
Telephone: 940 455 7050

Australian office

Xephon/RSM
GPO Box 6258
Halifax Street
Adelaide, SA 5000
Australia
Telephone: 08 223 1391

Contributions

If you have anything original to say about AIX, or any interesting experience to recount, why not spend an hour or two putting it on paper? The article need not be very long – two or three paragraphs could be sufficient. Not only will you actively be helping the free exchange of information, which benefits all AIX users, but you will also gain professional recognition for your expertise and that of your colleagues, as well as being paid a publication fee – Xephon pays at the rate of £170 ($250) per 1000 words for original material published in AIX Update. To find out more about contributing an article, see *Notes for contributors* on Xephon's Web site.

Editor

Harold Lewis

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, scripts, and other contents of this journal before making any use of it.

Subscriptions and back-issues

A year's subscription to *AIX Update*, comprising twelve monthly issues, costs £170.00 in the UK; $255.00 in the USA and Canada; £176.00 in Europe; £182.00 in Australasia and Japan; and £180.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the November 1995 issue, are available separately to subscribers for £15.00 ($22.50) each including postage.

AIX Update on-line

Code from *AIX Update* is available from Xephon's Web page at www.xephon.com (you'll need the user-id shown on your address label to access it).

© Xephon plc 1998. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs $36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

The AIX Performance Toolbox

INTRODUCTION

IBM's Performance Toolbox (PTX) is a set of utilities that provides comprehensive performance monitoring and tuning for RS/6000 systems. The product comprises two components, a manager and an agent, which can be purchased separately. Two types of manager are available: a *local manager*, for monitoring a single RS/6000 system, and a *network manager*, for monitoring multiple RS/6000s across a LAN using agents. The table below summarizes which managers are compatible with which agents.

AIX Version	Perfmgr Version	Perfagent Version
3.2.5	1.2.0.0	1.2.0.0
4.1.4	2.2.0.0	2.1.4.0
4.1.5	2.2.1.0	2.1.6.0
4.2.0	2.2.0.0	2.2.0.0
4.2.1	2.2.1.0	2.2.1.0
4.3	2.2.1.0	2.2.30.0

While managers are available only for RS/6000 systems, agents are available for non-IBM environments, thus enabling the unified performance management of all your Unix systems. The table below shows which non-IBM environments are supported:

Computer	Operating System
Hewlett-Packard 9000/700	HP-UX 9.01 and 9.03
Hewlett-Packard 9000/800	HP-UX 9.00
Sun SPARCstation 2	SunOS 4.1.3
Sun SPARC Classic	Solaris V 2.3
Sun SPARCstation 10	Solaris V 2.4
Sun SPARCstation 4	Solaris V 2.5

You should always install the version of the agent that corresponds with the operating system you're monitoring, even if other versions

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

appear to work. Also note that agents running on non-IBM platforms are able to collect only a subset of the statistics that are available for IBM systems.

It's also interesting to note that a substantial number of standard AIX system performance and monitoring tools, which used to be shipped free of charge with AIX, were then bundled with the performance agent component of PTX. The tools concerned are **rmss**, **filemon**, **netpmon**, **svmon**, **lockstat**, **tprof**, **fileplace**, **bf**, **bfprt**, **stem**, **syscalls**, **fdpr**, **genld**, **genkld**, **genkex**, and **stripnm**. Thankfully the decision to unbundle the tools from AIX has now been reversed, and the tools are once more available free of charge with AIX 4.3.

PTX FEATURES

The main features of PTX are as follows.

System performance monitoring

PTX enables system managers to see both recent and real-time performance characteristics of local and remote systems concurrently. A wide variety of statistics can be displayed simultaneously in various formats.

System performance analysis and control

Historical performance data can be compiled by both the manager and agent components of PTX. This data can be then displayed, graphed, printed, and filtered by various PTX tools.

The **filtd** component, which is part of the PTX agent, enables administrators to define performance-related conditions that trigger alerts and cause scripts to run that take remedial action to fix the cause of problem. In addition, the manager component has menu options that allow various standard AIX performance tools to be run from within PTX.

APIs for performance monitoring and reporting

The Application Programming Interface (API) library, which is bundled with the product, allows programmers to acquire performance

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

data from local and remote agents, and also to register custom data with, and report it to, local and remote managers.

SNMP INTERFACE

A PTX agent can both respond to SNMP requests and send 'traps' to SNMP managers (such as IBM's NetView). This facility is enabled by including the keyword '*dosmux*' in the agent's configuration file. The agent reports all available performance statistics to the local **snmpd** SNMP agent. This means that it is possible to monitor performance remotely without having to acquire a PTX manager by using an SNMP manager.

RS/6000 SP SYSTEMS SUPPORT

PSSP, which is the part of software that manages RS/6000 SP systems, contains the Performance Toolbox Parallel Extensions. Through this facility, agents running on individual nodes of an SP system report SP-specific performance data, such as statistics about switch and virtual shared disk activity, and about the Load Leveler program. The nodes themselves are organized into groups, each one containing a single manager node that collects statistics from all other nodes in the group.

USING AND CONFIGURING THE PRX MANAGER

Most of the information that the PTX manager displays takes the form of graphs. The basic entity that is displayed on these graphs – the *value* – is taken from one of the sets of statistics that a PTX agent is able to supply. One or more values are displayed as graphs in *instruments* (a combination of graphs with common axes). Instruments are combined into *consoles* that display collections of logically related statistics from various hosts.

The basic operation of the performance manager is very simple. Issuing the command **xmperf** returns a large message window which has a menu bar with the following six pull-down menus:

- *Files*

  Allows you to select 'recorded files' for playback, and also has

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

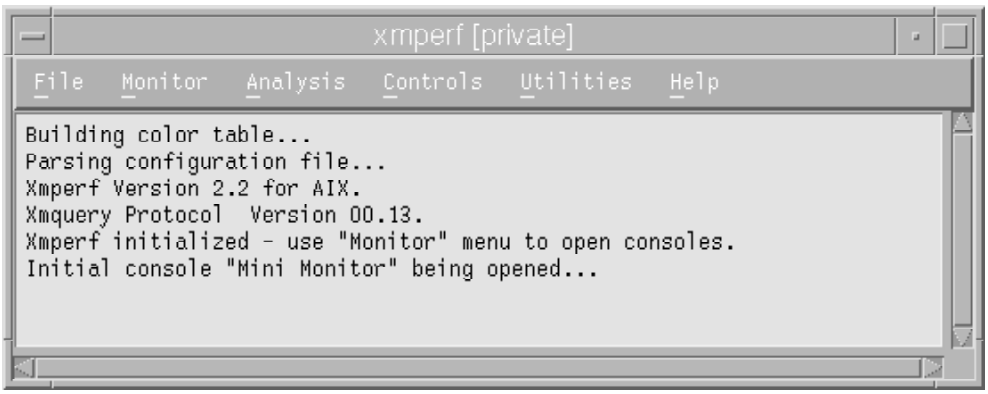

*Figure 1: The main **xmperf** window*

options for refreshing the list of reporting hosts and to exit from the program.

- *Monitor*

  This menu has options for instantiating one of the standard consoles, for adding a new console, and for starting one of the demo consoles.

- *Analysis*

  This menu provides an interface for invoking AIX and Unix commands that provide functions for monitoring system performance.

- *Controls*

  This menu provides a forms-based interface for managing processes running on the system, giving you the ability to change the priority of processes and, if necessary, to terminate them. The table displayed can be sorted according to the various columns of the **ps** report. Other selections allow you to alter network options, optimize programs, and edit logical volumes (the last option is available only on AIX 3.2.5).

- *Utilities*

  This menu's options allow you to display processes running on

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

remote hosts and to run various PTX tools, such as **3dmon**, the exception monitor, the recordings browsing tool, etc.

- *Help*

  In common with current practice, this menu's options allow you to display the version of the product and get help on PTX.

Figure 1 shows the main **xmperf** window.

Most of the menus are customizable; to customize a menu, create (or edit) a file called *xmperf.cf*. PTX searches for this file in the following order (which is worth bearing in mind if changes you made to *xmperf.cf* aren't reflected in the system):

1 The user's home directory

2 */etc/perf*

3 */usr/lpp/perfmgr*.

My advice, as always, is to keep the original intact and modify a copy. Another configuration file, *rsi.hosts*, controls the way in which PTX 'discovers' hosts that report statistics. The default is to broadcast invitations to the entire network, which can result in a lengthy initial set up time. An example of an alternative (commented) *rsi.hosts* file is shown below. This version limits the number of invitations sent to hosts, thus speeding up the initialization process.

```
#
# Sample rsi.hosts file
#
# Disable unlimited broadcasting
nobroadcast
# Broadcast to hosts on a specific subnet
9.192.1.123
# Broadcast to a specific host, specified by hostname
gold
#Broadcast to a specific host, specified by IP address
9.148.240.10
```

Various aspects of PTX's presentation, such as the colours, fonts, and labels used in graphs, can be changed by editing the file */usr/lib/x11/app-defaults/xmperf*.

The process of creating consoles and instruments, and of adding

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

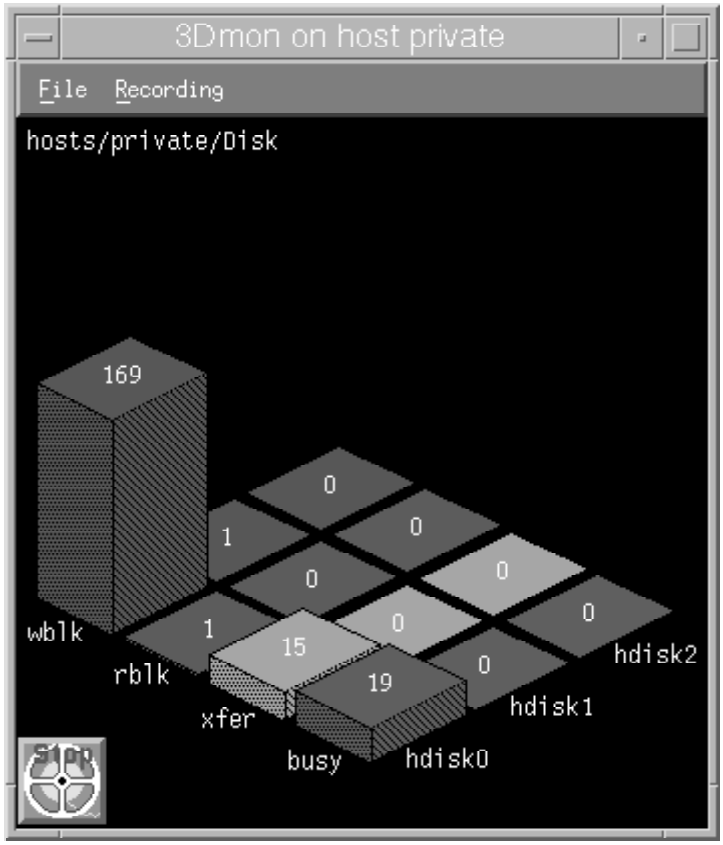

*Figure 2: **3dmon**'s interface showing hdisk0 I/O load*

values to them, is carried out by selecting items from submenus of **xmperf**'s *Monitor* menu. There are basically two types of value: *quantities*, such as the size of free memory and the number of processes waiting in the run queue, and *rates*, such as the number of packets received on the LAN interface per second and the rate at which data is written to a disk. Line, area, and bar charts are generally suitable for displaying quantities, while level, pie, and meter charts are suitable for displaying rates. The three dimensional monitor, **3dmon** (see Figure 2), is especially suitable for displaying a large number of identical statistics from similar sources, such as a group of remote hosts or a group of processes running on one host. Both regular and 3D consoles have options that enable the user to record data being displayed for off-line browsing and analysis. It is also possible to record data using the command line utility **ptxrlog**.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

CONFIGURATION OF PERFORMANCE TOOLBOX AGENT

The performance manager monitors statistics that are forwarded to it by the **xmservd** daemon. The **xmservd** daemon is started by the **inetd** daemon after it receives a UDP datagram from the PTX manager. The command to initiate the **xmservd** daemon is controlled by a line that appears in the file */etc/inetd.conf* after the keyword '*xmquery*'. If you intend that **xmservd** should continue running indefinitely after it's initiated, then you should add the following lines to the end of the file */etc/rc.tcpip* file once **xmservd** has been installed:

```
# Starting xmservd (PTX daemon)
sleep 2
/usr/bin/xmpeek.
```

Change the line in */etc/inetd.conf* that starts with the word *xmquery* to read:

```
xmquery dgram udp wait root /usr/bin/xmservd -p3 -l0
```

This tells **xmservd** to run indefinitely (as indicated by the parameter '*–l0*').

The operation of the **xmservd** daemon is controlled by two configuration files: *xmservd.res* and *xmservd.cf*. Default templates for these two files can be found in the directory */usr/lpp/perfagent*; you are advised to copy these files to the directory */etc/perf*.

The inclusion of the keyword *dosmux* in the file *xmservd.res* tells **xmservd** to report performance statistics to the local SNMP daemon, **snmpd**. Other information that may be found in this file includes a description of any other programs that supply performance data, such as **filtd**, and the number and names of hosts that can request statistics from the local system. Below is an example (with comments) of a typical *xmservd.res* file.

```
#
# Sample /etc/perf/xmservd.res file
#
# Limit the number of hosts that may receive statistics
max: 3
# Always allow the reporting of statistics to the following hosts
always: gold silver
# Permit access only to the following hosts
only: gold silver copper iron lead
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

Figure 3: ***exmon*** *window showing a number of exceptions*

```
# Invoke the default exception-supplying daemon
supplier: /usr/bin/filtd -p5
# Report performance statistics to the SMNPD daemon
dosmux
```

The **filtd** component of the PTX agent can be used to detect performance problems and report them to a program called **exmon** (Figure 3), which is part of the performance manager, and to the SNMP daemon, **snmpd**. Below is an example of the file *filter.cf*, which controls the operation of the **filtd** daemon.

```
#
# Sample /etc/perf/filter.cf
# © IBM Corp
#
# define new statistic, called cpubusy
user = CPU_gluser * 100 / (CPU_glkern + CPU_gluser +
CPU_glidle+CPU_glwait)\
       "Percent User CPU"
allcpu = CPU_glkern + CPU_gluser + CPU_glwait "Percent CPU, excluding
idle"
cpubusy = CPU_gluser + CPU_glkern
# Use newly defined statistics to trigger an exception
# and SNMP trap
@cpubusyalert:{TRAP22}{EXCEPTION} DDS_IBM_Filters_cpubusy > 90 \
       DURATION 15 FREQUENCY 2 SEVERITY 3 \
       "CPU Busy More Than 90% "
# Define new statistics to be reported by agent to manager
rwratio = ((Disk_+_rblk)) * 100 / ((Disk_+_rblk) + Disk_+_wblk)\
       "Read/write ratio, all disks combined"
diskmax = Disk_>_busy "Busy percent - most busy disk"
diskmin = Disk_<_busy "Busy percent - least busy disk"
diskavg = Disk_+_busy / Disk_#_busy "Average disk busy percent"
readdistr = (Disk_+_rblk / Disk_#_rblk) * 100 / Disk_>_rblk \
            "Average disk reads in percent of most busy disk"
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
# Use predefined statistics to report exception and open
# aixterm with descriptive
# message
@pspacelowalert: \
           [aixterm -e ksh -c "banner $(hostname) PAGE LOW;
read"]{EXCEPTION} \
  PagSp_%totalfree <1500 DURATION 60 FREQUENCY 15 SEVERITY 1 \
  "Test for Page Free  < 1500KB"}
# Use predefined statistics to fix the problem automatically and to
produce
# exception and visible report
@tmpfull:\
 [chfs size=+1 /tmp; aixterm -bg white -fg red -e ksh -c "(banner
$(hostname)\
 /tmp full and increased automatically;read)"] {EXCEPTION}\
 FS_rootvg_hd3_%totusded>95 DURATION 60 FREQUENCY 15\
       SEVERITY 3 "/tmp>95%"
```

The *xmservd.cf* file controls the recording of data for **xmservd**. Below is a commented sample of *xmservd.cf*.

```
#
# sample /etc/perf/xmservd.cf file
#
# Keep last week's recordings (each day in a separate file)
retain 7 1
# Sample statistics once per minute
frequency  60000
# Statistics to record each minute
Proc/runque
Proc/swpque
CPU/gluser
CPU/glkern
CPU/glwait
# Statistics to record once every ten minutes
Mem/Real/%comp 600000
Mem/Real/%noncomp 600000
PagSp/%totalused 600000
# Perform recordings on working days only, 7:30 am to 7:30 pm
start 1-5 7 30 1-5 19 30
```

You can obtain a list of all statistics that are supplied by the host by issuing the command:

```
xmpeek -l
```

You should also check the syntax of the commands in the file *xmservd.cf* using the command:

```
xmscheck
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

Recently, two new features have been added to *xmserv.cf* file: the keyword *command* allows you to specify commands to be executed before recordings are deleted and *Hot Lines* allows you to trigger recordings by events that occur on the host.

DISPLAY AND ANALYSIS OF PTX RECORDINGS

Information recorded by **xmperf**, **3dmon**, and **ptxrlog** can be played back by selecting the appropriate option from **xmperf**'s *Files* menu. The playback interface has familiar VCR-like buttons that permit the viewer to navigate the recording easily. Additionally the following commands are available to manipulate recording files:

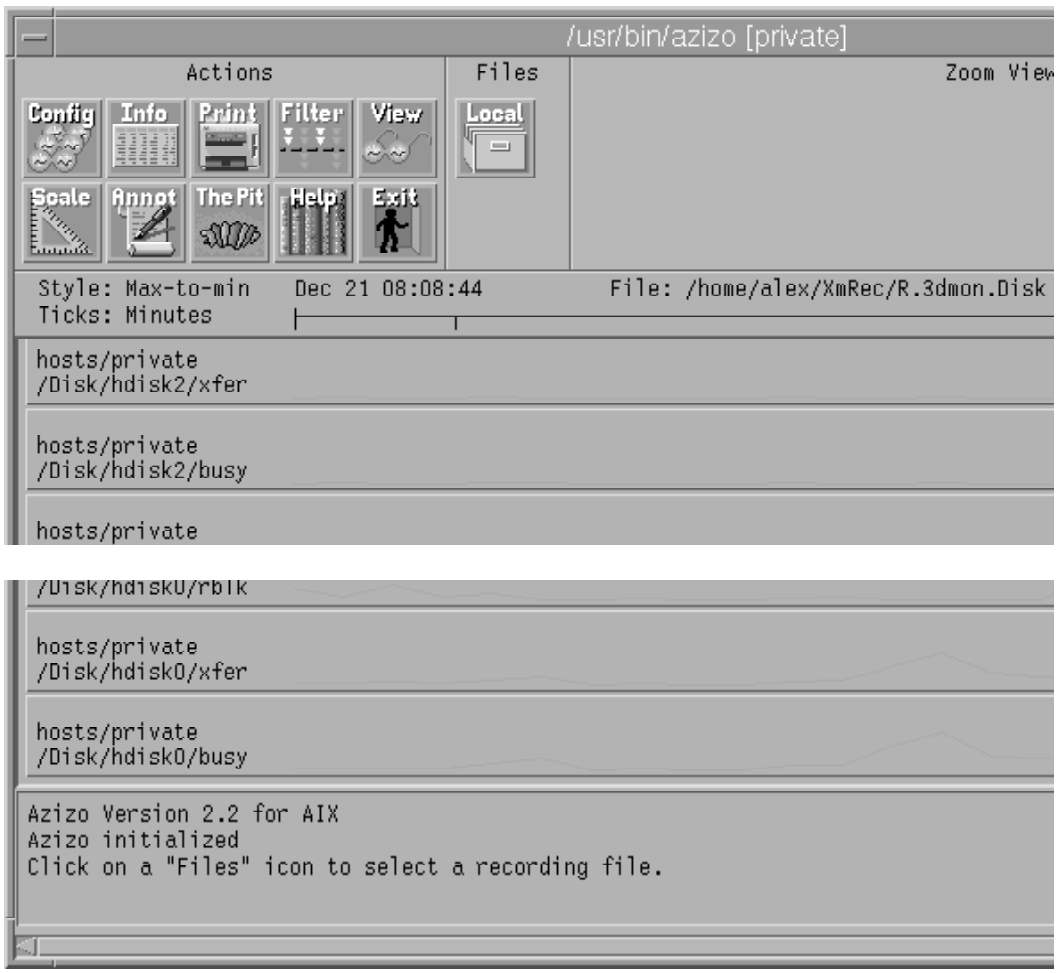

*Figure 4: **azizo**'s main window*

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

a2ptx Converts properly formatted ASCII files to PTX binary format

ptxmerge Merges a number of recording files into one file.

ptxsplit Divides a recording file into a number of other files, each containing statistics that relate to a particular named host, or a specific subset of statistics, or simply a subset of the whole that makes processing the statistics easier.

ptxconv Converts recording files from PTX 1.2 format to PTX 2.2 format and vice versa.

ptxtab Dumps recording files in tabular format – suitable for use by spreadsheets – according to various selection criteria.

ptxls Displays information about statistics in a recording.

Another graphical program, **azizo** (Figure 4 opposite), has more advanced facilities for processing recording files. When you start the program, it displays two windows – the main window and the top-level main graphs window. The top level main graphs window contains line charts of all statistics in the selected recording file. The main window contains (from top to bottom):

1 Icons for the utility's various functions

2 A metrics window displaying each of the metrics in the recordings file

3 A message window displaying program messages.

Most of **azizo**'s actions are performed by means of drag and drop – the user selects one of the graphs from the metrics window, drags it to one of the program icons with right mouse button, and drops it there.

To look for hot spots in recordings using **azizo**, use the following procedure:

1 Scan the metrics, looking for one that displays suspicious levels of activity.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

2 Select all other metrics and drop them on *ThePit* to reduce clutter.

3 Browse the main window from left to right, looking for graphs that display unusual performance-related events. Zoom in on the sections by selecting the appropriate graph area using the left mouse button. It is important at this point to select the *Keep Metrics* button in the *Zoom in* pop-up menu to keep the zoomed graph uncluttered.

At this point you can drag and drop the *Print* icon to the newly opened zoom window to get a hardcopy of the chart. Other handy options enable the user to save filtered portions of the recordings and to save 'configurations' (collections of filtered statistics).

SUMMARY AND FURTHER READING

This article describes only the basic configuration and operation of PTX. I didn't, for instance, cover the nitty gritty details of customization; my intent was to give the reader the basic tools necessary to deploy the tool quickly. For more information, consult the references below.

Standard IBM manuals (white books):

1 *Performance Toolbox for AIX: Guide and Reference*. SC23-2625

2 *Performance Toolbox Parallel Extensions for AIX: Guide and reference*. SC23-3997

The recent (October 1997) Red Book *Customizing Performance Toolbox and Performance Toolbox Parallel Extensions for AIX* (SG24-2011) discusses in detail the customization of PTX and PTPE. The Red Book *RS/6000 Performance Tools in Focus* (SG24-4989) gives an overview of both traditional Unix performance monitoring tools and AIX-specific tools, including PTX. Finally, the manual *Performance Tuning Guide* (SC23-2365) and the Red Book *Understanding IBM RS/6000 Performance and Sizing* (SG24-4810) discuss general principles of performance tuning.

A Polak
System Engineer
APS Ltd (Israel)

© Xephon 1998

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol, DHCP, is a client/server protocol for automating the configuration of systems that use TCP/IP. With the growth of TCP/IP intranets and the Internet, network managers are increasingly turning to DHCP (which is now implemented on all major operating systems) for TCP/IP management. In this article we'll find out how DHCP works, how it can simplify the task of managing your network, and how to configure it on AIX 4.2.

DHCP AND BOOTP

DHCP is described in the Internet Engineering Task Force (IETF) Request for Comments (RFC) documents 2131 and 2132. DHCP evolved from the BOOTP addressing protocol, so called because it allows a network host without an IP address to 'bootstrap' itself onto an IP network. The client/server model found in both DHCP and BOOTP is illustrated in Figure 1. Since the client doesn't know its own IP address or subnet until a server responds, it broadcasts a request for configuration. All servers listening on the BOOTP/DHCP port respond with network parameters (the DHCP and BOOTP daemons can't run simultaneously, as both use the same port).

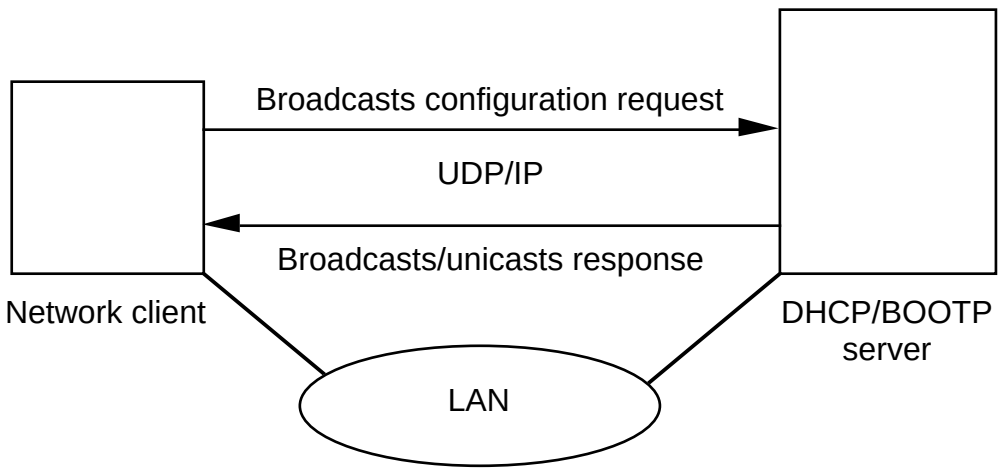

Figure 1: DHCP's and BOOTP's client/server model

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

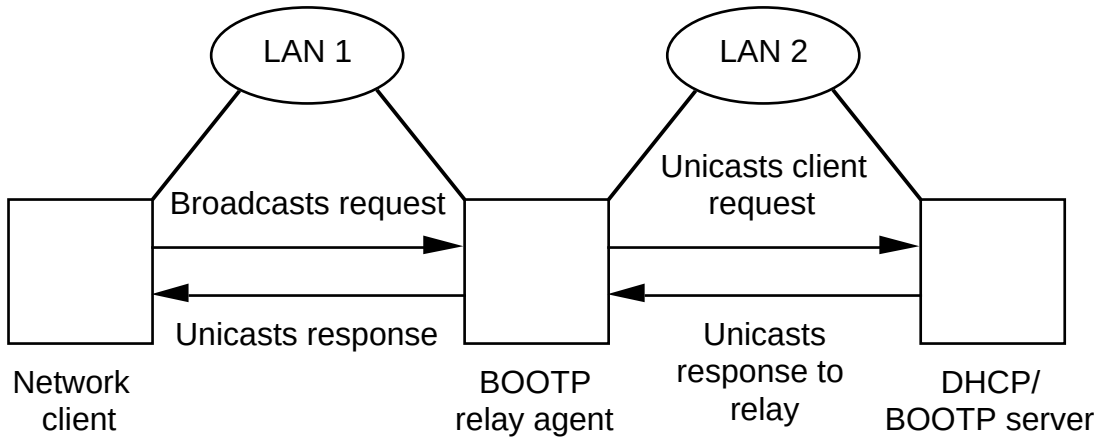

Figure 2: The operation of a relay agent

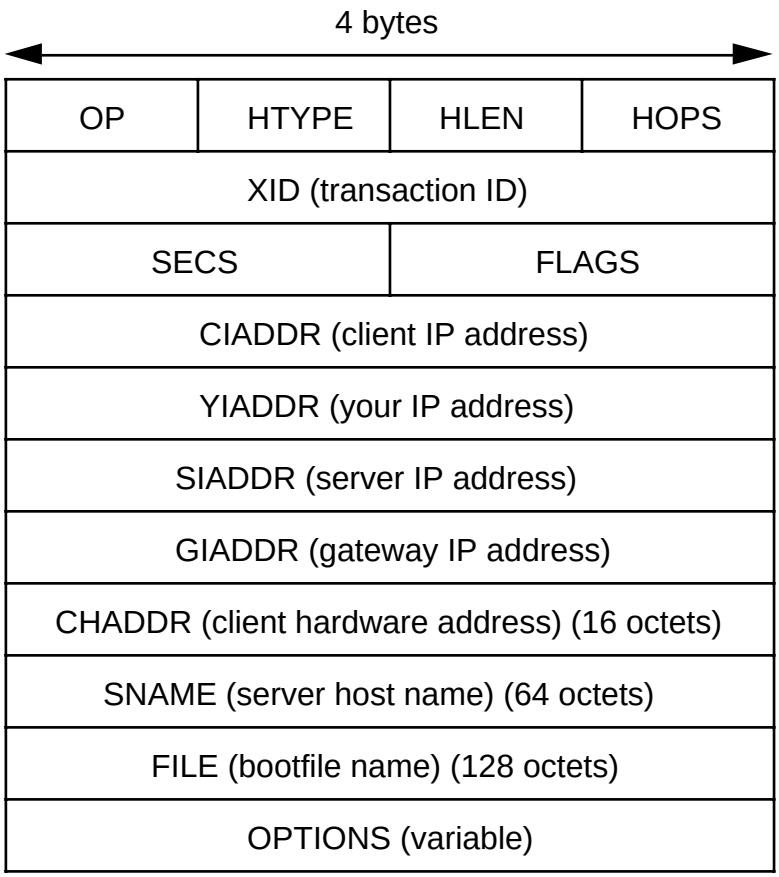

Figure 3: The DHCP packet format

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

One problem with this model is that broadcasting to servers off the local subnet may result in increased levels of network traffic, while maintaining a server on each subnet increases the workload of the network manager. BOOTP addresses this problem by using 'relay agents'. A relay agent is a router or other host that picks up the client's broadcasts and unicasts them off the subnet to one or more available protocol servers, as shown in Figure 2.

BOOTP relay agents can pass on DHCP messages too. In fact, the DHCP protocol specifies that all DHCP implementations must interoperate with BOOTP relays. That's why the DHCP packet format (see Figure 3) copies the BOOTP packet format. DHCP servers can be configured to work with BOOTP clients. But DHCP extends BOOTP in a number of important ways.

DHCP PROTOCOL BASICS

Let's watch while a newly installed network client goes through a protocol exchange with a DHCP server. An outline of the protocol flow appears in Figure 4 (overleaf). First, the client broadcasts a DHCPDISCOVER message, signalling to all available DHCP servers that the client wants to be configured. As discussed above, the broadcast is sent only on the local physical subnet – if there's no server on that segment, a relay agent forwards the request. The protocol packets are sent via the User Datagram Protocol (UDP) transport service, which in turn runs on the Internet Protocol (IP).

The DHCPDISCOVER message may include suggested values for various configuration 'options' such as name server or network address. I discuss these options in depth later in this article. A client that already has an address, but wants to obtain other configuration parameters (for instance, the address of the default gateway), can request this information from the server using a DHCPINFORM message.

OFFER, ACK, AND NAK

Let's assume there's no server on our client's subnet, and that a relay agent forwards the discover message to two different DHCP servers

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

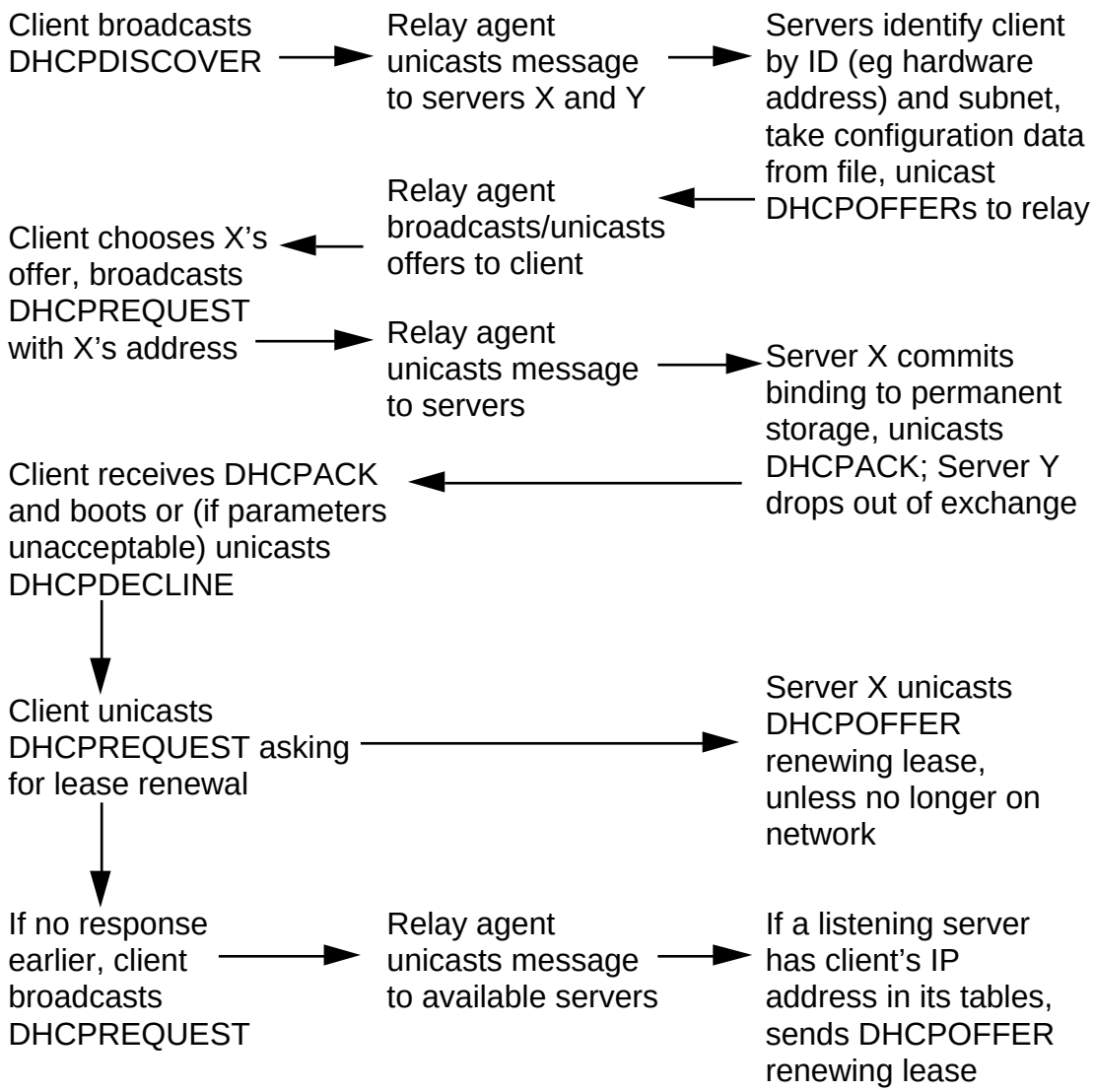

Figure 4: DHCP protocol flow

(*Server X* and *Server Y*) on the company's intranet. Each of these servers now sends the client a DHCPOFFER packet, containing an IP address and other configuration parameters. (The server will unicast the message to the client unless the client is unable to receive IP unicasts until it is configured – this is the 'bootstrap' problem mentioned above. In such cases, the server broadcasts on the client's subnet using the broadcast address *255.255.255.255*.)

Once the client has its two offers, it must choose between them,

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

according to network policy – for instance, it may favour a previously used server. Let's say that it chooses the offer from *Server X*. The client broadcasts a DHCPREQUEST, identifying *Server X* by its IP address. *Server Y* also receives the message, notes that *Server X* has been chosen, and drops out of the protocol exchange.

In the meantime, if the client spent too long dithering before accepting an offer, *Server X* may have given the address in the DHCPOFFER to another client – while the protocol discourages such conduct, it doesn't forbid it. In this event, *Server X* sends the disappointed client a DHCPNAK packet. The client then begins the DHCPDISCOVER cycle all over again.

However, if the address is still available, *Server X* enters the client-configuration binding into its files and sends the client a DHCPACK message. If the configuration package is acceptable to the client, it initializes and begins operation. The client may now take the precaution of using ARP to test its new address; if it discovers the address is already in use by another host, it sends the *Server X* a DHCPDECLINE message and begins the DHCPDISCOVER cycle again.

Note that if a client 'remembers' its last network address, it may skip the first part of the protocol cycle, and, rather than send a DHCPDISCOVER message, it may include the desired address in a DHCPREQEST message unicast to the appropriate server.

LEASE RENEWAL

With BOOTP, a client receives a permanent IP address – that is, until the administrator manually alters the server's configuration files. With DHCP, however, addresses are 'leased' to clients for a length of time up to infinity.

Let's assume that our client has an address on a six-hour lease. At the end of a predetermined time – the RFC suggests half of the lease period – the client unicasts a DHCPREQUEST to the server to say it wants to renew the lease. (Unless, of course, the client has crashed or for whatever reason come off the network.)

If *Server X* doesn't respond – say it's been moved to a new network segment – the client tries again at a suggested time in the lease term

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

(the suggested time being when one eighth, 12.5%, of the lease term remains). At this time it broadcasts a query, which the BOOTP relay picks up and forwards off the local segment. If *Server X* still doesn't get the message, the client loses the address when the lease expires, and it has to restart the protocol sequence. Meanwhile, what happens to our client's old address? As the binding is still in the server's files, the server won't allocate the address to another client unless there are no other addresses available. However, it notes that the lease has expired, and reallocates the address if necessary.

In some implementations of DHCP, a client that is ready to retire from the network prior to lease expiration can voluntarily relinquish its IP address using a DHCPRELEASE message. This facility is not, however, a requirement of the DHCP protocol.

CONFIGURATION PARAMETER ASSIGNMENT

We stated earlier that DHCP allows a client to get a wide variety of configuration options, as well as an IP address. RFC 2132 defines more than seventy different configuration options that may be given to a network client – including DNS domain name, subnet mask, location of default routers, NetBIOS information, and X-Windows information. These options are identified by a code number in the options field of the DHCP message.

Different packages of configuration options can be created for various subnets, client 'classes', and even individual clients. (Some implementations, though not AIX's, refer to these profiles as 'scopes'.) The network manager simply defines the attributes for the various subnets, classes, etc, on the server. When a client asks to be configured, the server notes the client's identifying information, such as subnet of origin and hardware address, and searches its tables for the matching configuration profile. We'll see later exactly how these configuration profiles are set up with AIX.

This facility allows great control and flexibility in managing the network. A client or type of client that needs specific configuration parameters can be treated differently from other hosts on the network. This 'class' feature can be used to handle network set-ups that include multiple logical subnets on a single physical LAN. The manager

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

defines each logical subnet as a separate class, identified in the server's configuration files by a string. Clients that are members of each subnet are then configured to present the class string as an identifier when they ask the server for configuration.

ADDRESS ASSIGNMENT

BOOTP requires the network manager to pre-configure the server with an address for each client. With DHCP, three different models of address assignment are available. As with configuration options, addressing can be custom-tailored to meet the needs of your network.

MANUAL ADDRESS ASSIGNMENT

Manual address assignment is similar to the old BOOTP model. A specific address is chosen and assigned on an infinite lease; the address is paired with the client's MAC address in the server's configuration file. (A 'MAC address' is a unique hardware address assigned by the manufacturer of the Token Ring or Ethernet card.) This type of assignment is useful if you have a file server or other host that requires a fixed address.

However, this type of address assignment begs the question: why not just walk over to the host and enter the IP address there, rather than use the server? The answer is that, for a start, using DHCP saves time and effort – you don't have to travel around the site to all your hosts. Added to that, the DHCP server can deliver other configuration parameters to the host at the same time as it delivers the IP address. Note that the infinite lease means you'll have to remove the binding from the server's tables manually if you want to reassign the address.

AUTOMATIC ADDRESS ASSIGNMENT

While DHCP can save time and effort, even when you use manually assigned addresses, it really comes into its own when using automatic address assignment. In this scheme, you define a range of IP addresses for the server to allocate. When clients boot up for the first time, the DHCP server selects an address for that client and delivers it, along with other configuration options. The server then stores the IP

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

address/MAC address pair for that host. The MAC address is normally used by the server as the 'key' that uniquely identifies the client.

DYNAMIC ADDRESS ASSIGNMENT

DHCP makes a distinction between 'automatic' assignment and 'dynamic' assignment. Both types involve assignment from a pre-defined pool of addresses, but 'automatic' assignment involves infinite leases, while 'dynamic' assignment involves short leases that allow addresses to be clawed back when clients no longer need them.

Dynamic address assignment is ideal for 'roving users', as it lets them plug into the intranet at any location and receive an appropriate configuration for the subnet. The fact that the lease expires after a period means the roving user doesn't leave a trail of un-needed addresses around the network.

This capability is especially valuable if you have more users that need to log on to the network than addresses. Say you have 25 sales representatives out in the field, each one equipped with a notebook PC. Each one logs on to the company intranet an average of eight hours a week, but they're not likely all to be logged on at any one time. You don't want to waste an IP address on each machine, so you create a pool of 15 addresses and deliver them on a four-hour lease. When machine *A* logs off, it will only keep its address if the address is not needed by other machines. If *A* comes back onto the network while its address is in use by someone else, it is simply assigned a new address.

A relatively short lease can be useful even if you have plenty of network addresses. For one thing, it prevents 'address hoarding', whereby a user jealously hangs on to every IP address he's ever had his hands on.

DYNAMIC DNS

Clearly, the automatic allocation of addresses has implications for the Domain Name System (DNS), which controls the translation of host names to IP addresses. In most cases, where client systems are not accessed by name from other systems, host names can be assigned arbitrarily. On the other hand, for 'well-known hosts' such as print and

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

file servers, the name cannot change and must always point to the host's current IP address. The manager must either hand-pick an address for this type of host or arrange for the DHCP servers to communicate with DNS servers.

The DHCP protocol does not provide a way to update DNS records automatically. In the absence of an industry standard, vendors have developed a number of proprietary DNS update methods. For example AIX's DHCP implementation includes a function called *Dynamic DNS*. This is implemented via the command **nsupdate**.

In April 1997, however, IETF released RFC 2136, which specifies a standard means for DHCP servers to send update messages to DNS servers, using either UDP or TCP. While it's too early to assess how widely this standard will be implemented, the growth of TCP/IP networks has made integrated management of DNS and IP addressing increasingly important.

IMPLEMENTING DHCP ON AIX

We've looked at the basics of DHCP operation, so now we'll see how it works with AIX. We'll use the Skillco Company's network as an example, so let's watch while its network administrator implements DHCP on the company network. The Skillco network is made up of three LANs connected by two routers, one of which is an AIX system. Each LAN is a subnet of the class B network *172.16.0.0*, and each has a number of AIX and other hosts. (See Figure 5.)

Skillco's network administrator plans to put a DHCP server on subnet *172.16.1.0*. The AIX router acts as a gateway between subnets *172.16.3.0* and *172.16.2.0*, and forwards protocol packets from those networks to the server. All network hosts need to be configured with IP addresses, subnet masks, the address of the default router and nameserver, and the domain name of the network.

Addresses are to be assigned dynamically. Although there are enough addresses to go around, the administrator wants to prevent address hoarding, so rather than assign addresses on an infinite lease, hosts receive a six-month lease. If a machine leaves the network, its address goes back into the pool within six months.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

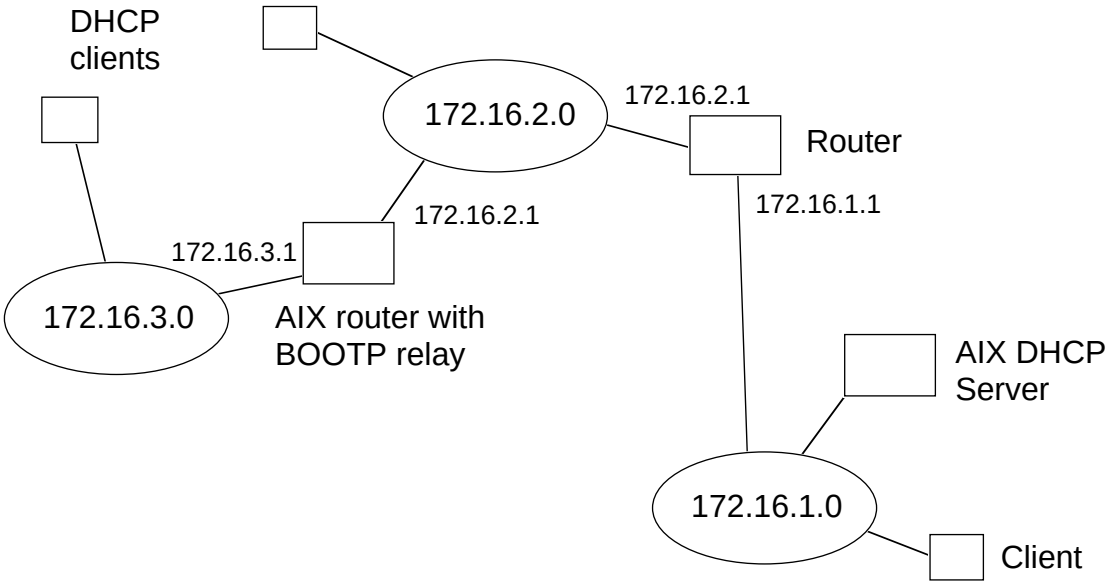

Figure 5: Skillco's LANs

Now that the administrator has created the master plan, the next step is to configure and start the AIX DHCP server, client, and relay daemons (**dhcpsd**, **dhcpcd**, and **dhcprd** respectively).

SERVER DAEMON

You can start the AIX server daemon using the command:

```
startsrc -s dhcpsd
```

Alternatively, use the command **dhcpsconf** to initiate an X-Windows GUI that lets you start, stop, and retrieve statistics from a running server, as well as allowing you to manipulate configuration files.

The default configuration file for the server daemon is */etc/dhcpsd.cnf* – specify an alternate file by using the command **dhcpsd -f**. The server's client-address bindings database is kept in */etc/dhcpsd.ar* and */etc/dhcpsd.cr*.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

SKILLCO SERVER DAEMON CONFIGURATION FILE

Here's the configuration file used to define Skillco's network:

```
logItem      SYSERR
logItem      OBJERR
logItem      PROTERR
logItem      WARNING
numLogFiles  6
logFileSize  100
logFileName  /usr/tmp/dhcpserver.log

leaseTimeDefault               6 months
leaseExpireInterval            1 day


network 172.16.0.0 24
{
  option 6 172.16.1.3            # Name server
  option 15 skillco.com.         # Domain name
  subnet 172.16.1.0  172.16.1.10-172.16.1.254
   {
     option 1 255.255.255.0      # Subnet mask
     option 3 172.16.1.1         # Default router
    }
  subnet 172.16.2.0  172.16.2.10-172.16.2.254
   {
     option 1 255.255.255.0    # Subnet mask
     option 3 172.16.2.1       # Default router
   }
  subnet 172.16.3.0  172.16.3.10-172.16.3.254
   {
     option 1 255.255.255.0    # Subnet mask
     option 3 172.16.3.1       # Default router
   }
}
```

Let's look at these entries one at a time. Note that the log entries and lease entries are global, affecting the entire file. By contrast, the entries that follow are local, relating only to a particular network or subnet. These local options are always contained within a pair of braces ('{' and '}').

- *logItem*, *numLongFiles*, *logFileSize*, and *logFileName*.

  These statements define the options, number, maximum size, and name of log files. The administrator has chosen several items to log in log files: SYSERR (system error); OBJERR (object error,

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

between objects in the daemon); EVENT (event occurred in the process); and TRACE (code flow for debugging); others are available.

- *network.*

  This entry identifies a network administered by this server, using its dotted decimal IP address. This entry may be followed by a subnet mask, specified either in dotted decimal notation, or (as here) by the number of bits in the mask. The network address may also be followed by the range of addresses to be assigned to hosts in this network. In our example, we have three subnets, so the address ranges are specified for each rather than at the network level.

- *leaseTimeDefault.*

  This entry defines the default lease duration, expressed as a decimal number, ranging from seconds to years. While the AIX default lease term is one hour, the Skillco lease term has been set at six months, as discussed above.

- *leaseExpireInterval*.

  This entry sets the time interval at which the server examines the lease expiration condition. The AIX default is one minute, though the Skillco server has been configured to examine leases once a day.

- The next three entries are options that apply to all hosts on network *172.16.0.0*:

  – DNS name server address *172.16.1.3*.

  – Domain name *skillco.com*.

  – Subnet mask *255.255.255.0.*

- *subnet*

  These entries specify the three subnets administered by this server, followed by their address pools for dynamic address assignment. Options specific to each subnet are contained within curly braces. In our example:

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

  - Subnet *172.16.1.0* is assigned addresses *172.16.1.10* through *172.16.1.254*. (The lower-numbered addresses are reserved for hosts that use manually-assigned permanent addresses.)
  - Subnet *172.16.2.0* is assigned addresses *172.16.2.10* through *172.16.2.254*.
  - Subnet *172.16.3.0* is assigned addresses *172.16.3.10* through *172.16.3.254*.

- *Client*/*Class*.

  While these entries don't appear in our example, as discussed earlier, specific clients can receive customized configuration profiles using the client statement.

- *Class*.

  This entry (which also doesn't appear in our example) can be used to define a specific group of machines that need special treatment. The argument is an ASCII string, followed (if desired) by an address range and options inside curly braces.

- *updateDNS*.

  This is another entry that doesn't appear in our example. It uses an ASCII string to specify an executable program to use for updating the DNS server, such as the command **nsupdate**.

CLIENT DAEMON

The usual way to start the **dhcpcd** daemon is using a line in the */etc/rc.net* script, which runs at boot time. This line is commented out by default. You can use **smit** to enable the daemon at start-up.

The command **startsrc -s dhcpcd** can be used to start the client manually. Three different flags may be used with this command:

- **-f** specifies the configuration file, the default being */etc/dhcpcd.ini.*
- **-t** specifies the time in seconds that the daemon waits before placing itself in the background (this lets a machine boot even if the client can't find a DHCP server).

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

- **-i** specifies that the daemon will run in Inform mode, which allows the client to get configuration information from the server without being assigned an IP address.

SKILLCO CLIENT DAEMON CONFIGURATION FILE

The **dhcpcd** configuration file is less complicated than the server file. It contains the same type of log file entries as the server configuration file, as well as options to be requested by this client and interfaces to configure. The Skillco example configuration looks like this:

```
numLogFiles     4
logFileSize     100
logFileName     /usr/tmp/dhcpcd.log
logItem         SYSERR
interface       en0
```

- *numLogFiles*, *logFileSize*, *logFileName*, and *logItem.*

  These are as described above for the server daemon.

- *interface*.

  This entry specifies the interface on which DHCP should be configured (you may specify more than one). The string causes the DHCP client to configure the first interface it finds and completes successfully. Options for each interface are specified within curly braces.

Entries that don't appear in our example include the following:

- *clientid.*

  This entry specifies the identifier that network clients should use in protocol exchanges with the DHCP server. This ID is one of the 'keys' that allows the server to store the configuration information for each client. The default *clientid* is the MAC address of the client, though the *hostname* is a common alternative.

- *reject.*

  This entry comprises one or more numbered option codes; if these options are returned by a server, the client should reject them.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

RELAY DAEMON

The relay daemon can be started using the command **startsrc -s**. The **dhcprd** configuration file contains logging entries, just as with the files for the server and client daemons. In addition it contains the addresses of the servers to which it will relay DHCP messages. The Skillco example file is as follows:

```
numLogFiles  6
logFileSize  100
logFileName  /usr/tmp/dhcpserver.log
logItem      SYSERR

server 172.16.1.1
```

- *numLogFiles*, *logFileSize*, *logFileName*, and *logItem*.

  These are as described for the server daemon above.

- *server*.

  This entry specifies the IP address of a server to which BOOTP or DHCP packets should be forwarded. More than one server may be included – all packets are forwarded to all specified servers.

CONCLUSION

DHCP is increasingly seen as a necessary adjunct to managing complex networked systems. DHCP can help you automate the configuration of hosts that use TCP/IP, saving time and reducing costly configuration errors. The protocol's ability to serve custom configuration profiles gives the manager complete control over the network, especially when combined with an automated DNS update tool such as AIX DDNS.

Nora and Larry Bennett are independent consultants specializing in TCP/IP, Unix, and the Internet. They can be contacted at ncb@mu-networks.com and lcb@mu-networks.com.

Nora Bennett and Larry Bennett
Consultants
mu Networks Limited (UK)

© Xephon 1998

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

AIX performance in client/server systems

This is the concluding part of this article, the first part of which appeared in last month's issue of *AIX Update*.

ON-LINE TRANSACTION PROCESSING

Although transaction processing has been established for many years on mainframes, it has only recently been introduced to AIX and other departmental systems. IBM's CICS is by far the best known transaction processing system, though others are emerging, such as Tuxedo.

CICS/6000 is built on top of DCE, the Distributed Computing Environment, and Transarc's Encina. DCE is a highly sophisticated framework that allows remote facilities to be accessed as if they were local. It also provides features such as time synchronization, security, and printing services. DCE's server components require substantial hardware resources if they are to provide acceptable performance; this is a result of having to load and run DCE, Encina, CICS, and business applications on one system.

The tuning of this environment is somewhat complex. CICS provides a wide range of transaction statistics that can be used to time each transaction. Where one transaction runs a number of applications on other systems, this can be very useful to identify which application is slowing down the overall response. Once this has been established, traditional techniques can be applied to determine the cause of the delay on that system.

CLIENT

The client workstation is a frequently overlooked component of client/server systems that, nevertheless, can easily be the source of performance problems. The current trend is to run ever more sophisticated applications, which have a voracious appetite for disk space, memory, and processor power. This trend can be seen clearly in the evolution of personal productivity applications such as Microsoft Word, which has grown from a modest DOS application that would

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

run in 640 KB of memory to a Windows application that requires at least 8 MB of memory and a fast i486 or Pentium-based PC to run effectively. When powerful client/server tools are being run on PCs and workstations, the client system needs to be sized accordingly. There are a number of areas where performance issues tend to occur:

- *Network stack*. There are a number of common network protocols in use, such as TCP/IP and IPX/SPX. Complex environments, with a mixture of systems and protocols, are common in many organizations. These, in turn, require that workstations run more than one protocol. A protocol stack is the software used to implement the communications protocol, and a stack is required for each protocol in use. Each stack needs memory, and also incurs a processor overhead on the workstation to service data packets that are received or sent by the system.

  Each protocol has its own strengths and weaknesses, so it's important, where possible, to select the most appropriate one for the application. For instance, although most popular file servers support multiple protocols, which protocol is chosen can result in large differences in performance. During a recent set of tests on an imaging system, a 300% performance difference was recorded in the system's performance when running two popular proprietary protocols. This equates to an application being perceived as either very quick or very slow by users.

- *Memory*. As is the case with servers, the performance of applications being run on clients is very memory-dependent. Today's Windows applications are especially memory-hungry, and are becoming increasingly more so. However, as the price of PC memory has fallen to under $10 per megabyte, a fraction of its price just a few years ago, it's usually worth investing in additional memory.

- *LAN adapter*. Network adapters have a limited bandwidth. While it's possible to buy cheap Ethernet adapters for $20 or less apiece, these are often clone cards based on the NE1000 or NE2000 Ethernet cards, and are thus based on an old design that's reflected in their throughput. Most modern designs are capable of much higher throughput – as much as 400% higher in some cases.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

Equipping workstations with good quality network adapters is a good investment, especially as it's now possible to buy 100BASE-T adapter cards for less than $100.

SECURITY

The security framework of client/server systems is becoming more complex. While this is necessary to ensure the acceptance of such systems in environments where sensitive data is handled, it has an inevitable impact on performance. DCE, for instance, provides a much higher level of security than is the norm in 'traditional' client/server systems. Some of the typical security techniques implemented in client/server systems are:

- *Logon*. When the user logs on to the computer system, initial access rights are determined against a security database. The database may be no more sophisticated than the *etc/password* file, or it can be a much more sophisticated set-up involving DCE.
- *Initial access*. When a user tries to open a file, his or her access rights are checked to see if the user is allowed access. This is normally checked against the file's attributes, but can be more sophisticated. For instance, CA-Unicenter uses a relational database to validate access rights.
- *Every access*. Some systems check each read or write operation to ensure that the user may perform that operation. While this has a significant performance impact, it nonetheless allows security to be enforced where access rights may change after the user has opened the file.
- *Encryption*. When sensitive files need to be accessed in an environment where security cannot be guaranteed, encryption can be used to prevent unauthorized access. Encryption takes two forms: *file level* encryption and *on-the-fly* encryption. File level encryption is where files are stored in a scrambled format; on-the-fly encryption is used to encrypt data prior to it being sent over a communications link. As there is usually a delay associated with the encryption process, security and performance requirements need to be balanced.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

- *Authentication*. DCE/Kerberos uses a sophisticated authentication scheme to ensure that users are who they claim to be. This is a two-way process, where the user and the security system exchange encrypted tokens, which they check to ensure they can decode them. This is used to get around 'spoofing' techniques, often used by hackers, in which an unauthorized user pretends to be a genuine user. Again there is an overhead associated with this process, which introduces a noticeable delay in simpler systems. Once more there is a trade-off between security and performance.

It is important to select the level of security that is consistent with the organization's needs, while also considering the performance implications of the security systems. When tuning systems that use third-party security packages, bear in mind that these are often shipped with default settings that are stricter than necessary.

PRINTING AND PRINT SERVERS

Nowadays, printing is one area of client/server systems that invariably causes performance problems. As client applications become more graphically orientated, the demand for sophisticated printing facilities increases. Users expect printing of colour graphics at the same speed as character data. Although this is possible, it can be achieved only after considerable expense. There is a wide range of performance issues with printing that need to be considered in order to reduce bottlenecks.

- *Graphics resolution and colour*. This is the main factor in determining print speed. A typical A4 page printed at a resolution of 300 dots per inch (dpi) generates a print file of about 2 MB. At 600 dpi, the figure is four times higher. If colour is added, the effect is similar. 24-bit colour requires three bytes (24 bits) to represent every dot on the page. I recently scanned a colour photograph for a high-quality catalogue at a resolution of 600 dpi using 24-bit colour, and the resulting scan file was over 48 MB in size. A file of this size is going to take a long time to print on any printer. The bottom line is that the lowest resolution and colour depth that is consistent with the user's requirements should be used. High resolution and colour are both nice to have,

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

but rarely needed. Recently 1200 dpi printers have become more widespread, and this will make printing performance even more of an issue than it has been up till now, as the size of print files grows.

- *LAN speed*. LAN speed is a critical factor in ensuring effective printing performance, as frequent print jobs result in many large files being transferred on a regular basis. The design of the LAN needs to allow for this.

- *Bridges*. I've often come across old PCs being used as LAN bridges or routers. While this is a cost-effective way of using old, redundant hardware, and it is usually easy to configure a PC as a gateway, the throughput of these devices is usually adequate only for interactive traffic, with poor performance on file transfers. The topology of the LAN needs to be studied to determine the route used between workstations and printers; where print jobs are normally routed through old PCs configured as routers, upgrading them to purpose-built routers is an effective (and not altogether obvious) way of improving print performance.

- *Printer datastreams*. There are two main formats used to send information to printers. Raster format allows bitmapped graphics to be sent straight to the printer. Vector format allows the printer to be sent a description of the required printed output (for instance, a command could be sent to print a circle of a given size and colour at specified coordinates). Vector format produces far smaller print files, which have less impact on the LAN. The printer is then required to interpret vector streams and convert them to raster format for printing. The speed of this process, known as 'rasterization', depends on the processor in the printer. Printers generally accept a variety of datastreams, the most common for desktop laser printers being H-P PCL format, while Postscript is frequently used on high resolution printers in the publishing industry. Printers can usually accept both vector and raster data to print. For Windows client applications, the datastream is configured via Windows and the printer driver.

- *Print engine speed*. The processor in a laser printer is often referred to as the 'print engine', and it determines the performance

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

of the printer. The speed of most printers is quoted in pages per minute (ppm). However, this is a very unreliable measure of the speed that can be obtained when printing graphics, which slow down printers. Printers, in common with other components of computer systems, should be evaluated prior to purchase by running them under a realistic workload. This is the only way to ensure that their performance will meet the requirements of users.

- *Accelerators*. Printing performance has long been a problem, and so companies like Xionics have produced a range of hardware-based solutions for improving print performance. Xip-Print is an accelerator card that lets many H-P printers produce graphics at the printer's nominal ppm speed. The cost of the accelerator is normally less than that of upgrading to a faster printer. Such products are worth investigating, especially as some also support compressed raster datastreams, thus minimizing network loading. (They uncompress files at the printer, while still maintaining a high throughput.)

- *Printer memory*. When the printer receives a raster file or has to rasterize an incoming vector file, it needs a considerable amount of memory to process the workload. The bit map is built in memory and then sent to the print mechanism. The vast majority of printers need to assemble the complete page in memory prior to printing. If there is insufficient memory, the printer usually prints as much of the page as it can handle in memory. This results in either incomplete pages or in print being split between two separate pages. Some printers can handle such problems by printing a partial page and holding the page for the remainder of the print. However, adequate memory ensures best performance, and 4 MB should be considered the absolute minimum for printers. If resolution greater than 600 DPI is being used, then 16 MB is appropriate.

- *Print servers*. Most networks use a dedicated print server that enables the client application to create the print spool file without having to wait for the printer. The print server can become heavily loaded and require considerable amounts of disk space. Again, old PCs are often press-ganged into use as print servers, but – as

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

with routers, this is not usually a good solution, and a higher specification PC is recommended where response times are an issue.

- *Soft fonts and font servers*. Word processors allow a wide range of fonts to be used in a single document. Printers themselves hold a number of standard fonts, and these require just a character datastream to print. When the document contains fonts that are not resident in the printer, either the fonts need to be sent to the printer or the document needs to be rasterized at the workstation. X-Windows provides font servers that enable fonts to be automatically downloaded to either X-Windows displays or printers. The downside to this flexibility is performance. If a document uses several fonts at different pitches, then each font's font tables need to be sent to the printer. The same is true for X-Windows displays. Raster font tables can be as large as 1 MB or more, and it may be necessary to send a number of them to the printer. This explains why some seemingly normal word processor files take so long to print. Microsoft's TrueType printing system also provides this on-demand font download capability. Ideally, try to use the fonts that are resident in the printer or buy a printer font cartridge to avoid this lengthy download.

NEW APPROACHES FOR CLIENT/SERVER TUNING

One of the issues facing support staff is that the IT environment is becoming more complex, and consequently problem determination is no longer straightforward (was it ever?). Coupled with the drive to provide a better service to end users, this places considerable pressure on support teams. In order to manage complexity, there is a need to simplify the way that diagnostic information is collected. Proactively anticipating performance problems is preferable to reacting to failures in the system. Most commercial tools for AIX (and Unix in general) have focused on providing graphical presentation of an increasing number of statistics. While this is undoubtedly very useful in determining the causes of problems, facilities are also needed to warn of potential problems, such as disks that are filling up rapidly and are in danger of running out of space.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

The biggest issue at most installations, though, is how quickly a service can be reinstated in the event of a failure or slowdown. For the rapid solution of a problem it is necessary to determine its root cause as quickly as possible, so that appropriate action can be taken. Being presented with a wide range of options and statistics can slow down the diagnosis process. There are a number of other areas that are worthy of investigation.

Response measurement and statistics

Understanding AIX statistics requires a good knowledge of Unix principles, as well as those of computer science in general. Many statistics are by themselves meaningless, and it is often the analysis of several statistics together that enables correct problem determination. For instance, if the number of queued processes is high, this should only raise concerns about CPU loading if the time spent by each process is also high. A much simpler approach to determining whether there are performance issues to be addressed is to consider only end-user response times. There are a number of commercial tools that allow you to measure this, and it can also be measured by running timed transactions from workstations at regular intervals.

When the system is running poorly, a useful way of determining the cause of the problem is to compare current statistics against a set of figures taken when the system was functioning satisfactorily. This technique allows problems to be pinpointed quickly simply by comparing two sets of figures. You can implement such a system using either commercial tools or standard Unix performance commands, such as **sar**. Spreadsheets, such as Microsoft Excel, provide a number of powerful functions that can be used to compare figures and present results graphically. Spreadsheets also allow you to analyse trends and predict future demand.

In a client/server environment, where middleware and the network can have a major impact on performance, it is possible to build a set of origin/destination matrices to calculate the typical time it takes for a transaction to transfer data over the network. Timed transactions can then be used to compare actual response against hypothetical response, thus enabling you to identify non-server performance factors. Although it is likely that future commercial products will enable an accurate

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

breakdown of server versus network times, it is currently necessary to compile this information using this approach. While this method requires some initial effort, it is an extremely effective approach.

If it is established that the bottleneck lies somewhere in the network, then you may need additional information to determine its exact location. Tools such as NetView/6000 and Sniffers can be invaluable in collecting this sort of information. The more complex the environment, the more important these tools become.

COMMERCIAL TOOLS

There is a growing range of tools available to assist in performance monitoring. While these currently allow you to monitor a number of distributed systems, they haven't as yet become fully integrated, and still don't offer automated problem diagnosis. Another shortcoming is that they don't effectively resolve dependencies between the server, network, and middleware. Despite this, these tools are much more effective than native AIX and Unix tools, particularly when it comes to their presentation and alert facilities.

The tools available can be classified as belonging to two distinct generations. The ones from IBM and HP are not as sophisticated as those from performance specialists, such as Landmark, BGS, and Candle. There is a considerable difference in the price of tools in the two generations, and also in that of tools from different vendors. Given the choice of tools available, my advice is to shop around to get the right ones for the right price.

Steve Butler (UK) © Xephon 1998

Leaving? You don't have to give up *AIX Update*.

You don't have to lose your subscription when you move to another location – let use know your new address, and the name of your successor, and we'll send *AIX Update* to both of you for the duration of your subscription. There's no charge for the additional copies.

 © 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

Monitoring AIX with PCs, revisited

This month's instalment concludes this article on monitoring AIX systems with PCs, the first part of which was published in the February 1998 issue of *AIX Update* (Issue 28).

THE DATA_HOME AND MONTH_FILES DIRECTORIES

Running the command **control_load l -n -zz** results in 21 files being placed in the directory *DATA_HOME* (*$HOME/system_stats*). The files in this directory conform to the following naming convention.

Files containing **sar** information, excluding averages, are named:

```
${DATA_HOME}/${MACHINE_ID}SAR${SEL}.TXT
```

while files containing only **sar** averages are named:

```
${DATA_HOME}/${MACHINE_ID}AVE${SEL}.TXT
```

${DATA_HOME}* is the directory that contains daily files, *${MACHINE_ID} is the RS/6000 system's identifier, and *${SEL}* is the **sar** parameter being collected (either *A*, *B*, *C*, *K*, *M*, *Q*, *R*, *U*, *V*, *W*, or *Y*). The strings '*SAR*' and '*AVE*' in the filename distinguish files of averages from normal collection files.

A total of 21 files are also placed in the monthly collection directory *MONTH_FILE* (*$DATA_HOME/month_files*) as a result of running the above command. The files in this directory conform to the following naming convention.

Files containing **sar** data (except average values) are named:

```
${MONTH_FILE}/${MACHINE_ID}${SEL}${FILE_TAG}.TXT
```

while files containing only **sar** averages are named:

```
${MONTH_FILE}/${MACHINE_ID}${SEL}${FILE_TAG}A.TXT
```

${MONTH_FILE}* is the directory containing monthly collections, *${MACHINE_ID} is the RS/6000 system's identifier, and *${SEL}* is the **sar** parameter being collected (either *A*, *B*, *C*, *K*, *M*, *Q*, *R*, *U*, *V*, *W*, or *Y*). *${FILE_TAG}* is the month in *mmm* format. The strings '*.TXT*'

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

and '*A.TXT*' distinguish text files of averaged data from ones of normal collection data.

SCHEDULING OF THE PROCESS.

The following is an example of a typical **crontab** table for scheduling the generation of the text files:

```
00 07 * * 2-5 /usr/home/it032x/control_load 1 -n -zz
00 08 * * 1 /usr/home/it032x/control_load 3 -n -zz
```

The first line generates files between Tuesdays and Fridays using the previous day's **sar** file. The second line generates files on Mondays based on the previous three days' files. One result of the above process is that daily files are overwritten daily. Thus, if files are required, they need to be processed daily.

TRANSFERRING THE FILES

The FTP script below can be used to transfer data files to the relevant PC server for use by the PC package.

CONTROL_TRANSFER SCRIPT

```
HOME=/usr/home/it032x
print "start transfer `date` ">>$HOME/ftplog
ftp -ivn <$HOME/ftpcommands >>$HOME/ftplog
print "finished `date` \n\n">>$HOME/ftplog
```

FTPCOMMANDS FILE

```
open servername
user username password
lcd /directory/where/files/are/located/on/the/host
cd /directory/where/files/are/kept/on/remote/host
mget *.TXT
quit
```

The above script is just an example of how to automate the transfer of data – you are the best judge of which files you require and when you require them, so exact details of how the transfer is implemented are left up to you.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

HOUSEKEEPING

The only housekeeping that's required results from the fact that daily files are overwritten each time the **control_load** script is run. This means that the transfer of these files should be scheduled to take place prior to running **control_load**. Also consider the housekeeping necessary for monthly files, which are appended to each month. These files can grow fairly large, so it may be necessary to archive them frequently.

SYSTEM COMMANDS

The process described above can be extended to encompass commands that report on system performance other than **sar**. This is useful if you don't have **sar** on your system, and also allows you to collect a wider range of statistics than **sar** provides. The scripts for using other system commands (they are listed later in this article) are similar to **control_load** and its associated formatting scripts. The scripts are:

- **control_load_command**
- **format_command_output**.

It is necessary to keep these scripts' output separate, as they use the same file names as the **sar** script. Their main use is to collect data at regular intervals throughout the day. Each run generates data that's appended to a monthly file and is also stored in another file that's overwritten each time the script is run. This gives you the option of transferring and processing the collected data after each run or at the end of the month. Feel free to change the script if you require only one of these approaches. Note that, if you use the script to collect data generated by the **ps** command, the resulting data files can grow large rapidly, depending on which options are used. Care is therefore needed when deciding what information to collect until you have a good idea of how much data is being generated.

Both files generated contain a unique identifier associated with the command used to generate the data. The monthly file contains the short month name (eg *DEC*) in its name, and you can customize the format of file names by altering the last two lines of the **format_command_output** script. The commands used in this example

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

are as follows: at the start of the **control_load_command** script, the commands **iostat** and **vmstat** are executed, with the resulting output being redirected into separate files (which are referred to in the options below). The interval times selected for the **iostat** and **vmstat** can be changed to suit your needs. The following options are available to the **control_load_command** script:

p This runs the command:

```
ps -eF "%C,%u,%c,%p,%t"|grep -v "%CPU" > $FORMLOG
```

This generates a file with the following data about processes:

– The %CPU time

– The user ID

– The name of the command

– The process ID

– The total elapsed time.

Changing the lower case '*c*' in the above command to an '*a*' produces additional statistics. Adding the string '*,%x*' to the end of the command's list of flags results in the accumulated CPU time also being recorded. This is given by the formula:

```
%CPU = (CPU time / Elapsed time) * 100
```

f This runs the command (it's just one line):

```
df -vk `mount|grep jfs|awk '{print $2}'`|grep -v Filesystem|awk
'{r=$6+$7};{print $9","$2","$3","r","$6}'>$FORMLOG
```

This generates a file containing the following data for all mounted filesystems:

– The total number of kilobytes

– The number of kilobytes used

– The total number of inodes

– The number of inodes.

c This runs the command:

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
cat $VMFILE|awk '{print $14","$15","$16","$17}'> $FORMLOG
```

This generates a file containing the following data about CPU consumption:

– The percentage user time

– The percentage system time

– The percentage idle time

– The percentage wait time.

s This runs the command:

```
cat $VMFILE|awk '{print $5","$6","$7","$8","$9","$10}'> $FORMLOG
```

This generates a file containing the following data about paging activity:

– The pager input/output list

– The number of pages paged in from paging space

– The number of pages paged out from paging space

– The number of pages freed

– The number of pages scanned by the page replacement algorithm

– The number of clock cycles consumed by the page replacement algorithm.

m This runs the command:

```
 cat $VMFILE|awk '{print $3","$4}'> $FORMLOG
```

This generates a file containing the following data about active and virtual memory:

– The number of actual virtual pages

– The size of free page list.

k This runs the command:

```
cat $VMFILE|awk '{print $1","$2}'> $FORMLOG
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

This generates a file with the following data about kernel threads:

– The number of kernel threads placed in the run queue

– The number of kernel threads placed in the wait queue.

t This runs the command:

```
cat $VMFILE|awk '{print $11","$12","$13}'> $FORMLOG
```

This generates a file with the following data about faults:

– Device interrupts

– System calls

– Kernel thread context switches.

w This runs the command:

```
who -u|sed "s/[ ][ ]*[ ]/,/g" > $FORMLOG
```

This generates a file with user information.

i This runs the following code segment:

```
for i in `lsdev -Cc disk|awk '{print $1}'`
do
cat $IOFILE |grep $i |tail -1|awk '{print
$1","$2","$3","$4","$5","$6}' >>$FORMLOG
done
```

This generates a file containing the following data on disks in the system:

– The percentage of time that the physical disk was active

– The amount of data transferred (read or written) to the drive in KB per second

– The number of transfers per second

– The total KB read

– The total KB written.

$VMFILE refers to the file created by the command **vmstat** in the **control_load_command** script. The **df** command's **-k** option is supported only in AIX 4 or later.

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

ADDING YOUR OWN COMMANDS TO THE SCRIPT

You can add your own command to the **control_load_command** script if you want to. All that's necessary is that you adhere to the following rules:

1 Each field must be separated by a comma.

2 The output must not contain any header information, though it's useful to note down any header information for later use by the PC package.

3 Each command must be given a unique lower case identifier, which is subsequently used as part of the output file's name.

Below is an example taken from the script:

```
case "$1" in
    p)
    ps -eF "%C,%u,%c,%p,%t"|grep -v "%CPU" > $FORMLOG
        ;;
```

The command that is executed by the segment of code above is **ps**, which is associated with the unique identifier *p*. The output of the **ps** command is formatted by removing the header and ensuring that each field is separated with a comma.

The general method for using the **control_load_command** script is similar to that used earlier (described in the first part of this article), and the command line for the script is as follows:

```
control_load_command -zz
```

The first parameter controls which commands are to be processed:

-zz Results in all commands being processed.

"p f" Results in only the **ps** and **df** commands being processed.

To get the best from these scripts, it's necessary to run them throughout the day at regular intervals, and it's also a good idea to set this interval in line with your **sar** report interval. Given the amount of output produced by the scripts, it may not be a good idea to use the **-zz** option every time the script is run, but to run different commands at different times – the choice is yours.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

CONTROL_LOAD_COMMAND SCRIPT

```
#!/bin/ksh

HOME=/usr/home/it032x
VMFILE=$HOME/system_stats/vmfile
IOFILE=$HOME/system_stats/iofile

STR_LOOP=$1

vmstat 5 2|tail -1 >$VMFILE &
iostat -d 30 2 >$IOFILE

if [[ "$STR_LOOP" = "-zz" ]]
then
    STR_LOOP="p f c s m k t w i"
fi
for i in $STR_LOOP
do

    print " $i started `date +"%H:%M"`"

    $HOME/format_command_output $i

    print " $i completed `date +"%H:%M"`"

done

rm -f $VMFILE
rm -f $IOFILE
#end

{format_command_output script}
#!/bin/ksh

DATA_HOME=/usr/home/it032x/system_stats
MACHINE_ID="MN"
FORMLOG=${DATA_HOME}/form1
VMFILE=$DATA_HOME/vmfile
IOFILE=$DATA_HOME/iofile
VAR_SPLIT=`date +"%y%j %d %d\/%m\/%Y`
YEAR_JUL=`echo $VAR_SPLIT|awk '{print $1}'`
SAR_DATE=`echo $VAR_SPLIT|awk '{print $2}'`
YES_DATE=`echo $VAR_SPLIT|awk '{print $3}'`
HOURMIN=`date +"%H%M`
HM_ENTRY=`date +"%H:%M"`

testfile ()
{
if [[ ! -s $VMFILE ]]
```


© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

```
then
    print "vm file is missing or not complete "
    exit 2
fi
}
testfileio ()
{
if [[ ! -s $IOFILE ]]
then
    print "io file is missing or not complete "
    exit 2
fi
}

>$DATA_HOME/Datafile2

case "$1" in
    p)
    ps -eF "%C,%u,%c,%p,%t"|grep -v "%CPU" > $FORMLOG
    ;;
    f)
    df -vk `mount|grep jfs|awk '{print $2}'`|grep -v Filesystem|awk
'{r=$6+$7};{print $9","$2","$3","r","$6}'>$FORMLOG
    ;;
    c)
    testfile
    cat $VMFILE|awk '{print $14","$15","$16","$17}'> $FORMLOG
    ;;
    s)
    testfile
    cat $VMFILE|awk '{print $5","$6","$7","$8","$9","$10}'> $FORMLOG
    ;;
    m)
    testfile
    cat $VMFILE|awk '{print $3","$4}'> $FORMLOG
    ;;
    k)
    testfile
    cat $VMFILE|awk '{print $1","$2}'> $FORMLOG
    ;;
    t)
    testfile
    cat $VMFILE|awk '{print $11","$12","$13}'> $FORMLOG
    ;;
    w)
    who -u|sed "s/[ ][ ]*[ ]/,/g" > $FORMLOG
    ;;
    i)
    testfileio
    >$FORMLOG
```

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

```
    for i in `lsdev -Cc disk|awk '{print $1}'`
    do
    cat $IOFILE |grep $i |tail -1|awk '{print
$1","$2","$3","$4","$5","$6}'>>$FORMLOG
    done
    ;;
    *)
    print "Not correct options"
    exit 2
    ;;
esac

NO_LINES=`cat ${FORMLOG}|wc -l`

let TAIL_CHOP=NO_LINES
let HEAD_CHOP=TAIL_CHOP

cat ${FORMLOG}>$DATA_HOME/Datafile1

COUNT=0

while [ $COUNT -lt $HEAD_CHOP ]
do

    let ROTATE=HEAD_CHOP-COUNT
    let IDENT=COUNT+1
    KEY="k"$YEAR_JUL"."$HOURMIN$IDENT

    tail -$ROTATE $DATA_HOME/Datafile1 | head -1| sed "s/^/
$KEY\,$YES_DATE\,$HM_ENTRY\,/g">>$DATA_HOME/Datafile2

    let COUNT=COUNT+1

done

sed "s/ //g" $DATA_HOME/Datafile2 > $DATA_HOME/Datafile3

SEL=`echo $1|tr "[a-z]" "[A-Z]"`


cat $DATA_HOME/Datafile3 > $DATA_HOME/${MACHINE_ID}${SEL}.TXT
cat $DATA_HOME/Datafile3 >> $DATA_HOME/${MACHINE_ID}`date +%b|tr "[a-z]"
"[A-Z]"`${SEL}.TXT
#end
```

HOW TO USE THE TEXT FILES

Suppose you needed to know which commands are being used by a particular user throughout the day, and to identify their CPU

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

consumption. This could easily be achieved using the scripts in this article. For instance, the file generated by the **control_load_command** script using the '**p**' option includes details of all processes. By

```
Key           Date     Time  %CPU UID  Command  PID  Elapsed
k97253.08001 10/09/97 08:00  2.1 root init     1    11-09:13:55
k97253.08002 10/09/97 08:00  0.3 root syncd    2250 11-09:13:22
k97253.08003 10/09/97 08:00  0.0 root tsschctl 2692 11-09:13:13
k97253.08004 10/09/97 08:00  0.0 root biod     3044 11-09:12:52
k97253.08005 10/09/97 08:00  0.0 root errdemon 3282 11-09:13:22
```

Figure 1: Sample of the file generated

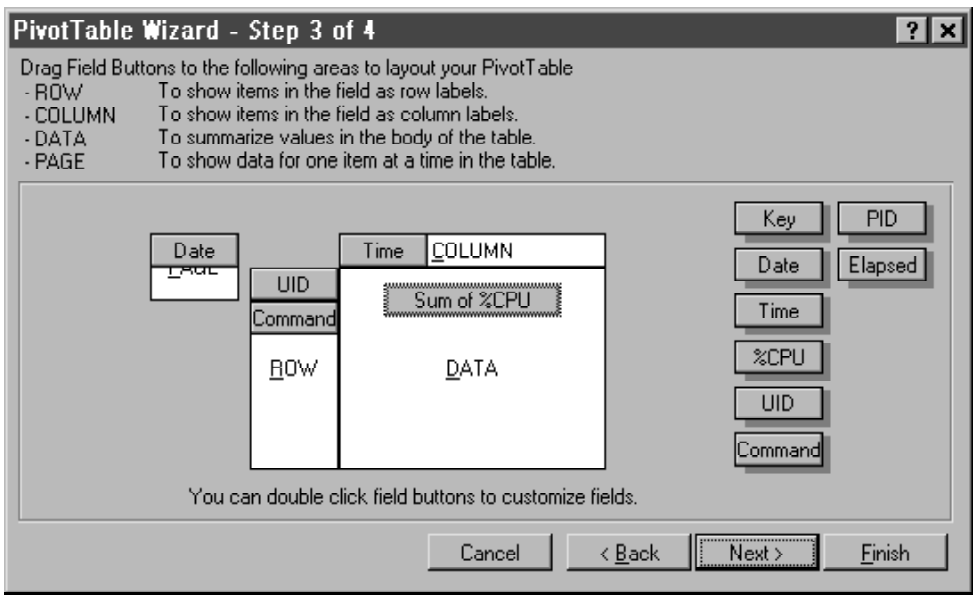

Figure 2: Creating a pivot table

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

transferring the file to a PC, it can then be analysed using a package such as Microsoft Excel 7. Figure 1 shows an example of such a file, showing details of processes, including process ids (*PID*s). Note that the headings for each field in the file have been added using Excel.

An ideal way to analyse this type of data is using a *pivot table*. To produce a pivot table using Excel, select cell *A1*, and then select the *Pivot Table Report…* option on the *Data* menu, and follow the defaults until you get to the following step (the procedure is essentially the same using Excel from Office 97, though you have to click the *Pivot Table Wizard* button after you've chosen the data source to get the dialogue below). Select all options as shown in Figure 2 (you can drag the headings on the right into the areas on the left).

Following the wizard to completion generates a pivot table as shown in Figure 3.

This makes it easier to establish what individual users are doing and to calculate the average CPU usage. The pivot table displays the sum

Date	(All)					
Sum of %CPU		Time				
UID	Command	8:00:00	8:10:00	8:20:00	8:30:00	8:40:00
User1	aixterm	0	0	0	0	0
	ftpd	0	0	0	0	0
	ksh	0	0	0	0	0
	manu	3.3	1.5	0.8	1	0.8
	mwm	0	0	0	0	0
	oracle	1.6	0.7	0.5	0.4	0.3
	rlogin	0	0	0	0	0
	sh	0	0	0	0	0

Figure 3: Pivot table

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

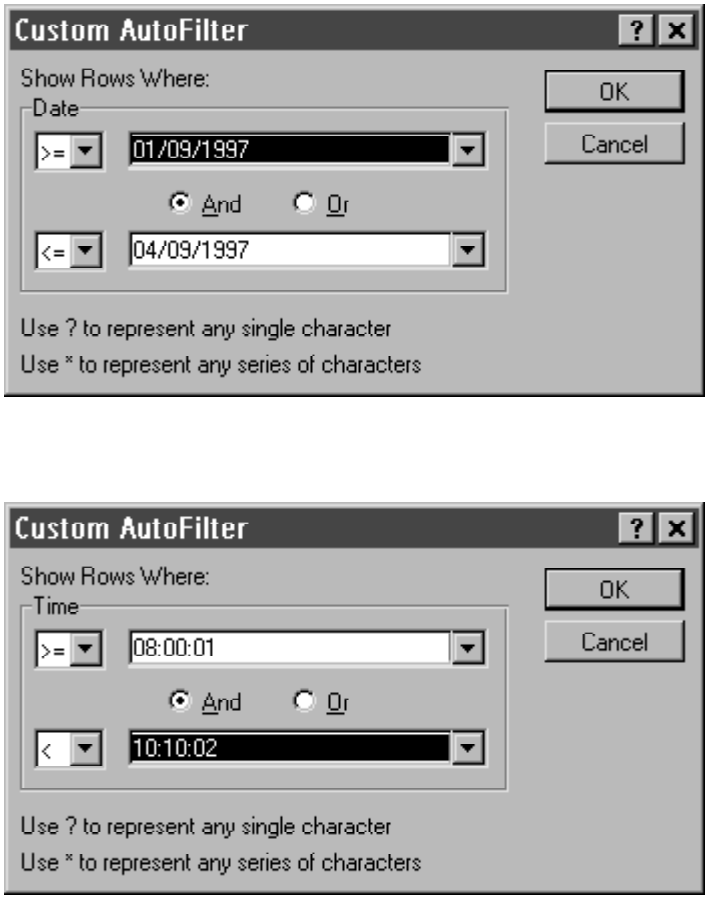

Figure 4: Setting up filtering

of *%CPU* for individual commands from the time the command was initiated. This measure of CPU usage doesn't represent the process's CPU consumption at the time shown – it shows the average since the process started. You can investigate CPU usage further by adding *PID* to the *ROW* data in the Pivot Table Wizard screen, as this shows each instance of the command separately. A further refinement is to use an '**a**' instead of a '**c**' in the **ps** command's formatting options, as this results in the full name of the command being shown (though it has the drawback of increasing the size of the data files).

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

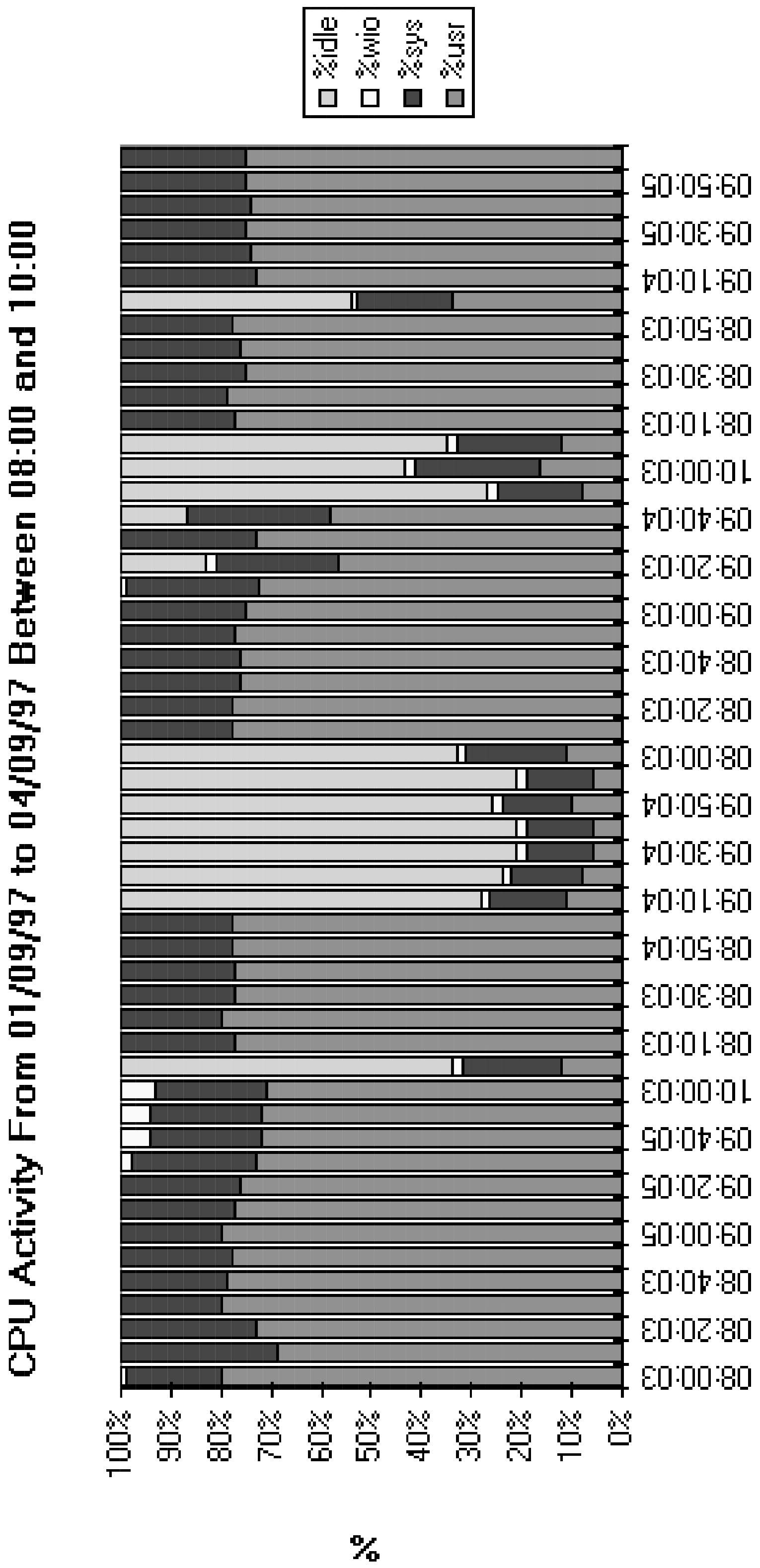

Figure 5: Charting system performance

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

To analyse **sar** reports using Excel, it's a good idea to start by filtering them using Excel's built-in *Data Filter*. This enables a more selective use of the monthly files. For example, if you need to analyse the performance of your system on certain days and between specified times, load the monthly **sar** files into Excel and, after selecting cell A1, select *Filter|Autofilter* from the *Data* menu.

Each column is then presented with a pull down menu. If you select the *Custom* option on the *Date* column, you'll be presented with the dialogue shown in the top half of Figure 4, which allows you to filter the data you require. Selecting the *Custom* option in the *Time* column allows you to filter particular times (the bottom half of Figure 4).

The data is then filtered to include only that which conforms with the date and times set; this then could be charted to highlight the reports. A chart resulting from the above operation is shown in Figure 5.

PERFORMANCE MONITORING

By generating text files of system performance data and analysing them on a PC it is possible to determine where the problems lie with systems. This procedure aids in identifying bottlenecks, and also points the way to further research and analysis. The usual suspects when performance problems are encountered are CPU, memory, network, and disk usage. Reports on these can easily be generated using **sar**'s **-q** and **-u** options and the **ps** command. Use **vmstat** to examine memory usage and **iostat** to analyse disk performance. Using **sar** it is possible to build a historical picture of system performance. AIX 4 provides more extensive commands for analysing performance in the Performance Toolbox/6000 product (which is also available in AIX 3.2), though this is not a standard installation product.

ERROR REPORTS

A corollary to the development of the above scripts was that we found other uses for the date calculation script. Our systems are regularly monitored, and error logs are routinely checked. We used the date script to develop a report that could be printed off or displayed to the

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

screen, and which made the process of system monitoring more thorough and much quicker. The script is listed below. If you make the necessary changes, the script could easily be implemented at your installation.

ERROR REPORT SCRIPT

```
#!/bin/ksh

if [[ "$1" = "" ]]
then
    BACK=1
else
    BACK=$1
fi

HOME=/usr/home/it032x
PRINTER=lp1
HOUR_MIN=`date +"%H%M"`
MONTH_DAY=`date +"%m%d"`
YEAR=`date +"%y"`
LONG_VAR=`$HOME/calcdate $BACK -s -m -d -y`
STR_MONTH_DAY=`echo $LONG_VAR|awk '{print $1 $2}'`
STR_YEAR=`echo $LONG_VAR|awk '{print $3}'`
REPORT_STR=`$HOME/calcdate $BACK -/ -d -m -y`
REPORT_END=`date +"%d/%m/%y`


START=$STR_MONTH_DAY"0000"$YEAR
END=$MONTH_DAY$HOUR_MIN$YEAR

print " ERROR LOG REPORT"| tee /tmp/error.rob
print " ****************\n"| tee -a /tmp/error.rob
print " From $REPORT_STR 0000 to $REPORT_END $HOUR_MIN\n"| tee -a /tmp/
error.rob
errpt -s $START | tee -a /tmp/error.rob
#Uncomment the following errpt line if only hardware errors are
required.
#Also delete the original errpt command
#errpt -d H -s$START | tee -a /tmp/error.rob
print "\n########END\n"| tee -a /tmp/error.rob
lp -d$PRINTER /tmp/error.rob
#end
```

Robert Russell (UK) © Xephon 1998

© 1998. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.

Contributing to *AIX Update*

AIX Update is primarily written by practising AIX specialists in user organizations – not journalists, or consultants, or marketing people. In our view, the information and advice provided by such people – people like you and your colleagues – are far more valuable to their fellow professionals than the alternatives available from other sources.

We don't expect you to write an original article from scratch – just send us listings or specifications of any relevant programs, utilities, scripts, user modifications, or other code that might be of use to other installations, with a short explanation of why it was developed and what it does. And most IS departments produce a great many internal technical reports and other documents, many of which can easily be adapted for publication. Xephon's editorial staff will transform even the most informal listing or document into a polished article fit for publication. So you don't have to spend any time on reformatting or rewriting your contribution – just send it as it is and we'll do the rest.

Contributors aren't just helping their fellow professionals – they also receive a significant material reward themselves. We pay good rates for the articles we publish: $250 (£170) per 1000 words if we get copyright, and $140 (£90) per 100 lines of code.

A copy of *Notes for contributors* can be downloaded from Xephon's Web site at *www.xephon.com*. Articles for submission to *AIX Update* can be sent to the editor, Harry Lewis, at *HarryLewis@compuserve.com*.

© 1998. Reproduction prohibited. Please inform Xephon of any infringement.

Rational Software has announced ClearCase version 3.2, a software configuration management product that now features wizards and applets and a Visual SourceSafe converter. The product features Snapshot Views that enable developers to work remotely, letting them check files, modify them off-line, and merge them back to the main line of code at a later date. The Visual SourceSafe Converter provides a means to migrate SourceSafe information into ClearCase. The product is expected soon, it runs on AIX and a number of other Unixes (and also on NT), and costs US$3,000.

For further information contact:
Rational Software, 18,880 Homestead Road, Cupertino, CA 95014, USA
Tel: +1 408 862 9900
Web: www.rational.com

Rational Software, Olivier House, 18 Marine Parade, Brighton BN2 1TL, UK
Tel: +44 1273 624814
Fax: +44 1273 624364

* * *

Platinum has launched AutoAnswer, a Web-based automated Help Desk package aimed at helping users to solve their own problems. It integrates existing problem resolution technologies, based on the company's Apriori product, with call centre management software from Quintus. Included are modules for problem resolution, call entry, and call managementt.

AutoAnswer supports all the major databases (though DB2 wasn't explicitly mentioned). The server component runs on AIX and other flavours of Unix.

For further information contact:
Platinum Technology, 1815 S Meyers Road, Oakbrook Terrace, IL 60181, USA
Tel: +1 630 620 5000
Fax: +1 630 691 0718
Web: www.platinum.com

Platinum Technology, Platinum House, North Second Street, Central Milton Keynes MK9 1BZ, UK
Tel: +44 1908 248400
Fax: +44 1908 248444

* * *

IBM has announced its TXSeries transaction processing middleware, which combines IBM and Transarc products along with new technologies aimed at providing secure, reliable, and scalable transactions.

The product is geared towards extending existing applications and building new ones that use the Internet, providing multiple Java-enabled gateways, supporting the CORBA IIOP, and enabling standard Web browsers to serve as front-ends to the resultant TXSeries applications.

Out now for AIX and Solaris, the product includes Encina, distributed CICS, MQSeries, DSSeries, and Lotus Domino GO Webserver with new features for Web-based transaction processing.

For further details contact your local IBM representative.