55

AIX update

May 2000

In this issue

AIX Update

Published by

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

North American office

Xephon/QNA
Post Office Box 350100, Westminster CO 80035-0100, USA
Telephone: (303) 410 9344
Fax: (303) 438 0290

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, where you can download *Notes for contributors* in either text form or as an Adobe Acrobat file.

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 £180.00 in the UK; $275.00 in the USA and Canada; £186.00 in Europe; £192.00 in Australasia and Japan; and £190.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the November 1995 issue, are available separately to subscribers for £16.00 ($23.00) each including postage.

AIX Update on-line

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

Printed in England.

WLM and resource management in AIX v4.3.3

This month's instalment concludes this article on AIX's Workload Manager (the first part appeared in last month's issue of AIX Update).

Examples of each of the files in a property set are given below. Comment lines start with an asterisk '*', and the exclamation mark ('!', sometimes referred to as a 'bang') in the rule files represents a logical *NOT*.

/ETC/WLM/CURRENT/DESCRIPTION

```
Daytime production property set
```

/ETC/WLM/CURRENT/CLASSES

```
* IMPORTANT:
* IF THIS FILE IS MODIFIED WLM MUST BE RESTARTED AND A REBOOT IS
* ADVISABLE
*
Default:
description="WLM default class"
tier=0
System:
description="Default class for root processes"
tier=0
Database:
description="Oracle database"
tier=0
Intranet:
description="Intranet and Extranet web and application servers.
tier=0
```

/ETC/WLM/CURRENT/SHARES

```
*
* Last modified: 26/11/99 by wainj (jez wain)
*
Default:
CPU = 100
memory = 100
System:
CPU = 150
memory = 200
Database:
```

```
CPU = 500
memory = 1000
Intranet:
CPU = 100
memory = 100
```

/ETC/WLM/CURRENT/LIMITS

```
*
* Last modified: 26/11/99 by wainj (jez wain)
*
Default:
CPU = 0% - 100%
memory = 0% - 100%
System:
CPU = 5% - 100%
memory = 5% - 100%
Database:
CPU = 20% - 100%
memory = 30% - 100%
Intranet:
CPU = 0% - 100%
memory = 0% - 100%
```

/ETC/WLM/CURRENT/RULES

```
*
* Last modified: 26/11/99 by wainj (jez wain)
*
*
* class   reserved    user       group    application
* =======+=========+=========+==========+=============
System       -         -      root,admin        -
Database     -       oracle     oracle          -
Database     -         -          -      /usr/lpp/bin/oracle
Database     -         -       db_admin  !/bin/ksh
Intranet     -         -          -      /weblogic/bin/wls
Intranet     -         -          -      /usr/local/netscape/bin/httpd
Intranet     -       nobody       -      /bin/ksh,/usr/local/bin/perl
Default      -         -          -            -
```

STARTING WLM

The default AIX behaviour is not to use WLM, which must therefore be initiated by the system administrator. The usual method of doing this is to place a WLM entry in the *inittab* file so that WLM starts when the system is booted. A typical entry would be:

```
wlm:2:once:/usr/sbin/wlmctrl >dev/console 2>&1 # start WLM
```

This entry is automatically appended to the *inittab* file if WLM is started through either the WebSM or the **smit** interfaces. However, as processes that are already running are not included in WLM resource calculations, the WLM entry in *inittab* should be as high as possible on the list in order to capture information on the largest possible number of applications. A good place for the entry is just after filesystems are mounted.

WLM COMMANDS

WLM brings with it a number of new commands, including:

- **wlmcntrl**

  Starts and stops WLM.

- **wlmstat**

  Monitors WLM.

- **mkclass**

  Creates a new class.

- **chclass**

  Modifies a class's attributes.

- **rmclass**

  Deletes a class definition.

- **lsclass**

  Lists classes and their attributes.

Only the first two of the above commands are discussed here. For more information on the others, see Chapter 12 of *System Management Guide: Operating Systems and Devices*. This is available on-line at the address below (the continuation character, '➤', shows a formatting line break).

```
http://www.austin.ibm.com/doc_link/en_US/a_doc_lib/aixbman/
➤  baseadmn/toc.htm
```

Note that using any of the above commands, either directly or via the WebSM or **smit** interfaces, will result in the contents of the configuration files being reformatted. In particular, any comments in the files will be removed.

```
wlmcntrl [ -a | -p | -u | -o | -q ] [ -d configDir ]
    -a         (Activate) start WLM.
    -p         (Passive) collect statistical data without
               performing resource allocation.
    -u         (Update) update after modifying property files.
    -o         (Off) stop WLM.
    -q         (Query) show the current state of WLM (on or off).
```

The above command is usually run from *inittab* during system initialization. Users other than *root* can use only the **-q** switch.

```
wlmstat [ - l class] [-c | -m | -p | -i ] [ interval [ count ] ]
    -l         Display statistics for the given class.
    -c         Display CPU statistics.
    -m         Display memory statistics.
    -p         Display paging space statistics.
    -i         Display I/O statistics.
    interval   Specifies an interval in seconds.
```

count specifies how many reports **wlmstat** produces before the program exits. If *interval* is specified without *count*, the output continues until the program is killed (**^C**).

This command is similar to all the other ***stat** commands (**iostat**, **memstat**, etc). The command has a hidden, undocumented **-x** switch that produces a more detailed listing. However, this switch may change to **-v** or **-a** in a later release.

OTHER COMMANDS

The **ps** and **snap** commands have also been modified to take WLM into account. The **ps** command now includes the class format, which is used with the **-F** or the **-o** switches. For example:

```
ps -eo pid,class,args
```

WLM USAGE GUIDELINES

WLM provides a means of modifying AIX's default CPU and memory allocation algorithms. It should be understood that these algorithms have been fine-tuned over years of experimentation with thousands of systems in real-world configurations. This experimentation was carried out by system performance experts and AIX kernel engineers who understand all the issues and trade-offs. The efficiency of these algorithms is borne out by the fact that, at the time of writing, a mono-node 24-CPU AIX 4.3 SMP system hold the absolute TPC-C performance benchmark of over 130,000 tpm (more than other manufacturers have achieved with systems of up to 128 CPUs). The message here is that, before running WLM in a production environment, you should fully understand each application's behaviour and resource consumption characteristics, and perform benchmarking on a test system that closely matches the production system and its workload.

Assuming a performance problem has been identified in one or more applications, the first step is to allocate shares. Shares represent resource consumption goals for the operating system, and, while goals are there to be met, AIX and WLM don't worry too much if there are a few excesses from time to time. In this way, even if share allocation is not optimal, AIX is usually able to compensate and maintain acceptable response times and throughput. On the other hand, limits are never exceeded, so AIX is not able to compensate for a less than ideal configuration. Once shares are allocated, re-measure application performance to quantify the changes. These measurements should be made over a representative time period for the application mix. Additional tuning of share allocations will then probably suffice to meet the performance objectives.

Should some applications still fail to perform satisfactorily as a result of resource shortages, then consider setting the *WLM MINIMUM* resource limits to guarantee memory and CPU cycles to those applications. However, setting *MAXIMUM* limits is very risky and may quickly lead to disastrous system performance.

WLM provides a flexible and powerful tool for allocating system resources to specific applications so that they meet their performance objectives. However, it provides only a minimal amount of coherency checking for the configuration rules. Thus the system or performance administrator should be careful when deciding on the resource allocation rules, and always test any modifications prior to applying them to a production environment.

CONSOLIDATION, CLUSTERS, AND RM – KEEP IT SIMPLE

As discussed in the introduction to last month's instalment, the current trend towards server consolidation is a prime driving force behind the renewed interest in resource management. However, with more heterogeneous workloads, it's more likely that a cluster of two or more AIX servers, perhaps running on different hardware, such as Escalas and RS/6000s, will provide the most effective solution in terms of performance, configurability, and reliability.

For instance, a data warehouse usually requires dedicated hardware to prevent complex queries from taking resources away from more time-critical functions and on-line users. Application development and testing usually benefits from dedicated hardware. This avoids conflicts when allocating priority to production work or development. Different machines running different types of application may be clustered, providing a high-availability configuration without additional hardware. While this is resource management at its most rudimentary, it nonetheless offers a number of advantages, not least being the fact that keeping different applications apart eliminates complex interactions and greatly simplifies performance management. Other resource management strategies, such as WLM, may be applied when a node fails and several applications then run on a single node, competing for the now limited resources.

RESOURCE MANAGEMENT OUTSIDE AIX

The operating system is not the only place where resource management takes place. In typical stand-alone non-clustered servers running commercial applications, resource management also occurs both in the database and the transaction monitor. In the near future, it will also occur in Enterprise Java Beans (EJBs) and their successor, J2EE (Java 2 Enterprise Edition). The next sections take a quick look at resource management in Oracle 8*i* and Tuxedo.

RESOURCE MANAGEMENT BY THE DATABASE SERVER

In client/server systems, only servers have the information necessary to establish the resource usage patterns of connected clients. Some databases and database management systems, such as DB2 on IBM's S/390, make this information visible either to the operating system or another resource or workload management application. Oracle 8*i* uses a different approach – adding a resource management function to the database itself. The advantage of this is that the Oracle resource manager has access to and can operate on the database's internal structures and variables, thus providing close and accurate control over them. The downside is that this requires yet another user interface.

Resource management within the database provides a mechanism to control and limit the computing resources available to any given client (user) or group of clients. This makes it possible to:

- Guarantee minimum resources to a request, so it continues to progress regardless of system load.
- Share computing resources among competing processes and clients.
- Vary resource allocations by time.

ORACLE 8I'S RESOURCE ALLOCATION METHODS

While Oracle 8*i* includes resource management tools, only CPU can be managed using the current release, though one of its parameters also affects memory usage indirectly. It is not possible, however, to directly limit the amount of memory consumed by the database server on behalf of a client. The control mechanisms are:

- CPU usage
- Degree of parallelism.

Resource consumers are grouped according to their relative importance. Should CPU resources be left over after all consumers at a given importance level have been serviced, the remaining CPU time is given to consumers in the next level. Within a given level, CPU time is divided among clients based on specified percentages. For example 80% of CPU time at a given level may be allocated to transaction-oriented clients, with the remaining 20% being allocated to decisionsupport queries. This degree of resource planning is possible as the database keeps a record of the time it spends servicing each client. This information is not, however, available outside the database.

The degree of parallelism is the number of threads that are used to handle the request. Since each thread has its own scratch area, this indirectly influences the memory resources consumed by the client.

An important improvement in Oracle 8*i* is the treatment of batch jobs that exceed their allotted time (for example, by overrunning into the OLTP window). These are no longer killed, but are suspended and can be restarted at an appropriate time. In earlier releases, any batch jobs running at the start of the OLTP session were killed, throwing away all the work already performed.

ORACLE CONCLUSION

Oracle 8*i* offers some useful resource management features without losing its conceptual and modelling simplicity. Given that Oracle systems are closed to the outside world, only the database itself can perform resource management. However, Oracle's view of the outside world is based on database connections. This is fine in simple two-tier environments, where there is generally a one-to-one mapping between clients and connections. However, when you add a transaction monitor between client and database, this mapping is lost.

Oracle's resource manager is not accessible to OPS, which means that it won't perform accounting across the nodes of a cluster.

RESOURCE MANAGEMENT BY THE TRANSACTION MONITOR

Transaction Processing (TP) monitors manage the entire transaction processing lifecycle. TP monitors know how to execute transactions, route them across systems, balance their workload, and restart them in case of failure. The management of all aspects of a distributed transaction, including system and resource management, is one of the main reasons why TP monitors are so widely used. TP monitors can manage resources on one or more servers and can cooperate with other TP monitors to implement a federated architecture.

TP monitors don’t really perform resource management, though they do optimize system performance through load balancing. By monitoring operation and performance, they balance the workload either statically or dynamically.

TP monitors sit between the client and the database server. One of the most important functions they perform is to multiplex client connections onto a much smaller number of database connections. This is required as each database connection consumes significant system resources, particularly memory and network bandwidth. By funnelling multiple clients into a much smaller number of database connections, the server requirement can be reduced. Multiplexing is feasible as end-users are willing to wait a short time before they receive the reply to a request.

TP monitors for AIX include Tuxedo and CICS. The following section discusses the load balancing functions in Tuxedo.

TUXEDO LOAD BALANCING

Tuxedo allows client requests to be serviced by more than one machine and/or more than one process on the same machine. Incoming requests are shared across all the available processes and machines. This avoids the situation where one process is overloaded, with a large queue of requests waiting to be serviced, while another is idle. This policy improves system response time and performance and has a secondary benefit of optimizing resource usage – as requests are balanced across servers, so the share of resources among operations is also balanced.

As the TP monitor has a global view of each transaction, it’s able to perform load balancing in situations where other components, such as the operating system or database, don’t have enough information to perform this task. For example, a database is aware only of the operations that it is performing – it doesn’t know about operations being performed by other databases, even if they are part of the same transaction.

Tuxedo load balancing uses a ‘load factor’ that is assigned to each service. This is the relative time taken to perform the requested operation. A default value is provided by the system that can be changed by the administrator. For example, an ‘average’ request may be assigned a load factor of 100, and other jobs are then given higher or lower load factors depending on their execution time. This load factor takes into account the relative speed of each machine in a multi-system environment. Note that execution time is the only factor taken into consideration here. To help administrators evaluate load factors, Tuxedo provides a reporting tool called **txrpt**, which lists the average execution time of each service.

The load factor is used differently depending on whether Tuxedo is in a single-system or multi-system environment. For example, consider a single-system environment in which four instances of Tuxedo server all provide the same three services. Each server has a queue of waiting service requests, and the queues are all visible to the Tuxedo scheduler. Each of the services has a different load factor, and incoming requests must be equitably shared among servers so as to minimize request time latency as perceived by the end-user. Each time a new service request is received, it is placed in the queue with the lowest load factor, and the load factor of the new job is added to the queue’s load factor. Each time a server removes a service request from a queue, the queue’s load factor is decreased by the load factor of the request retrieved.

Tuxedo offers a second technique for load balancing by allowing several servers to service a single request queue. This is given the fancy title of MSSQ, which simply stands for ‘Multiple Server Single Queue’. In this model, all incoming requests for services are placed in the same queue. When a server completes one request, it simply pops the next waiting request from the queue and services it. This is similar to multiple tellers serving a single queue in a bank or post office. It can be shown that the average waiting time of the MSSQ model is less than that of the multi-server multi-queue approach.

TUXEDO PRIORITIES

In addition to load balancing, Tuxedo also allows the prioritization of services and/or service requests. Every service is assigned a default priority, and the administrator can adjust every service’s and service request's priority. High-priority requests are handled first, though a request is dequeued every tenth service cycle in FIFO order, regardless of its priority, so as to guarantee the forward progress of low-priority jobs. It should be noted that no privileges are associated with the assigning of per-request priority, so care must be taken to avoid the deliberate or accidental abuse of the prioritization mechanism.

SUMMARY OF TUXEDO RESOURCE MANAGEMENT

The use of either of Tuxedo's load balancing methods will reduce average queue length and, in turn, minimize overall service time as perceived by the end-user. In addition, the prioritization of both requests and services means that important requests get an even faster response.

However, it can be seen that resource management in the transaction monitor is concerned only with the execution time of each service. While this execution time can be indirectly mapped to CPU occupation, in multi-server environments it may also include network delays. No account is taken of the consumption of other system resources, such as memory and I/O.

CONCLUSION

As all the resource and workload management tools and services discussed in this article are complementary, you'd expect that, together, they would provide a fairly comprehensive solution. The difficulty lies in the fact that each of systems discussed considers its own view to be the correct one. There is no information exchange between the various resource mangers, so that, for example, Oracle 8*i*'s resource manager doesn't use WLM to modify process priorities or limit memory usage. And therein lies the problem. In open systems, with hardware from one vendor, a database management system from another, a transaction monitor from a third, and applications from yet other vendors, the only means of providing integrated services is by all components adhering to an appropriate standard. This does not exist today, and (as far as I am aware) there is no movement in this direction.

All the existing Unix tools express resource management in terms of system resources rather than the performance objectives of the business. This places the burden of mapping business objectives to resource allocation on the administrator. There are a number of problems with this approach; for a start, the skills required to map business objectives to resource allocation are generally not available, as they require detailed knowledge of the behaviour of the kernel and applications. Furthermore, any mapping produced would be static, even if it includes rules for different times of day. Any change to the workload or system configuration would require a re-definition of the resource allocation policy.

Resource management is designed to handle multiple tasks dynamically, adjusting each one's share of CPU according to business requirements. With the current trend to consolidation, a smaller number of servers now handle the same number of applications, and so workloads are increasingly more mixed. In such environments, resource management can help get the most out of the existing computing infrastructure.

Complete end-to-end resource management in a heterogeneous environment is not realizable in the short-term. Nevertheless, today's resource management tools will continue to be refined and performance, resource, and workload management will be more closely integrated in future. Resource managers will monitor and manage additional sub-systems, including I/O, system, and storage area networks, and other interconnection mechanisms, such as non-uniform memory architecture (NUMA) and clusters. However I don't foresee the inclusion of simple devices, such as thin-clients, network fan-outs, and simple routers, in resource managers. This is because today's monitors and operating systems are often incapable of collecting the necessary statistics, let alone acting on or controlling the work being performed.

Jez Wain
System Architect
Bull SA (France)

cron administration

If an application uses **cron** heavily, the shell script below (**cronadm.sh**) may be useful. The script must be executed from *root* account and offers the following functionality:

- View any user's *crontab* file
- Create or modify any user's *crontab* file
- Revoke **cron** usage
- Grant **cron** usage
- View the *cron.allow* and *cron.deny* files
- View and edit the **cron** log file
- List valid **cron** users.

NOTE

1 The control of **cron** via the *cron.allow* and *cron.deny* files works with AIX version 4.3.2, but is unreliable with earlier versions.

2 The continuation character, '➤', indicates a formatting line break that's not present in the original source code.

CRONADM.SH

```
#! /bin/sh
######################################################################
# Name       : cronadm.sh (cron administration)
#
# Description: Manages cron administration.
#
# Notes      : 1. The script contains the following functions:
#                 o main
#                 o HandleInterrupt
#                 o MoveCursor
#                 o DefineVariables
#                 o DisplayListOfValues
#                 o RootUser
#                 o DisplayMessage
#                 o ProcessExit
#                 o DisplayCRONMenu
#                 o ViewUserCrontabFile
#                 o CreateOrModifyCrontabFile
#                 o GrantCronUsage
#                 o RevokeCronUsage
#                 o ViewCronLog
#                 o TrimCronLog
#                 o ListValidCronUsers
#                 o ViewCronAllowFile
#                 o ViewCronDenyFile
#                 o ProcessExit
########################################################################
# Name       : DefineVariables
#
# Description: Initializes all variables used in the script
########################################################################
DefineVariables ( )
{
#
# define default operating system and id
#
CURRENT_USERID=`id | tr "()" "::" | cut -d':' -f2`
DATETIME=`date "+%d/%m/%y at %H:%M:%S"`
#
# define escape sequences
#
ESC="\0033["
RVON=_[7m                  # reverse video on
RVOFF=_[27m                # reverse video off
BOLDON=_[1m                # bold on
BOLDOFF=_[22m              # bold off
BON=_[5m                   # blinking on
BOFF=_[25m                 # blinking off
TRUE=0
FALSE=1
FEC=1                   #  failure exit code
SEC=0                   #  succes exit code
OPTION=                 #  selected menu option
SLEEP_DURATION=5        #  seconds to sleep
ERROR="${RVON}${BON}cronadm.sh:ERROR:${BOFF}"
INFO="${RVON}cronadm.sh:INFO: "
#
# define menu title
#
CRON="${RVON}CRON ADMINISTRATION MENU (CRON)${RVOFF}"
#
# define variables for cron
#
AUTHORISED_USER=${CURRENT_USERID};export AUTHORISED_USER
CRON_ALLOWED_FILE=/var/adm/cron/cron.allow;export CRON_ALLOWED_FILE
```

```
CRON_DISALLOWED_FILE=/var/adm/cron/cron.deny;export CRON_DISALLOWED_FILE
CRONTAB_DIR=/usr/spool/cron/crontabs;export CRONTAB_DIR
CRONTAB_FILE=${CRONTAB_DIR}/${CRON_USER};export CRONTAB_FILE
CRON_LOG_DIR=/var/adm/cron; export CRON_LOG_DIR
CRON_LOG_FILE=log;export CRON_LOG_FILE
#
# define temporary files
#
TEMP_FILE="/tmp/cronadm_$$.tmp"
TEMP_FILE_1="/tmp/cronadm_$$_1.tmp"
#
# define signals
#
SIGHUP=1    ;  export SIGHUP     # used to refresh cron daemon
SIGINT=2    ;  export SIGINT     # ctrl-c
SIGTERM=15  ;  export SIGTERM    # kill command
SIGTSTP=18  ;  export SIGTSTP    # ctrl-z command
#
# PrintSpoolFile
#
DEFAULT_PRINTER=pr01    #  Unix printing
PRINTER_NAME=
FILE_TO_BE_PRINTED=
PRINT_REQUEST=
OUTPUT_FILE=
#
# messages
#
WORKING="Working...${RVOFF}"
INTERRUPTED="Program interrupted\; quitting...${RVOFF}"
INVALID_ENTRY="Invalid entry.${RVOFF}"
ROOT_USER="The script must be executed by root.${RVOFF}"
#
# define messages for cron management
#
CRON_DISALLOWED="User \${CRON_USER} is not authorized to use crontab
➤  command${RVOFF}"
NO_CRONTAB_FILE="crontab file \${CRONTAB_FILE} does not exist${RVOFF}"
MADE_CRON_GRANT="Granted cron usage to \$CRON_USER${RVOFF}"
NO_CRON_GRANT="Failed to grant cron usage.${RVOFF}"
CRON_GRANT_OK="\$CRON_USER already has permission to use cron
➤  facility${RVOFF}"
INVALID_CRON_USER="\${CRON_USER} does not exist in the system${RVOFF}"
INVALID_USER="Invalid user${RVOFF}"
REVOKED_CRON_GRANT="Successfully revoked cron usage${RVOFF}";export
➤  REVOKED_CRON_GRANT
CRON_GRANT_ALREADY_REVOKED="Cron usage is already revoked from
➤  \${CRON_USER}${RVOFF}"
REVOKE_CRON_GRANT_FAILED="failed to revoke cron usage${RVOFF}"
NO_CRON_LOG="Cron log file ${CRON_LOG_DIR}/${CRON_LOG_FILE} does
```

```
➤  not exist${RVOFF}"
NO_CRON_ALLOWED_FILE="File, \${CRON_ALLOWED_FILE} does not
➤  exist${RVOFF}"
NO_CRON_DISALLOWED_FILE="File, \${CRON_DISALLOWED_FILE} does not
➤  exist${RVOFF}"
}
########################################################################
# Name       : HandleInterrupt
#
# Description: Calls ProcessExit.
########################################################################
HandleInterrupt ()
{
DisplayMessage I "${INTERRUPTED}"
ProcessExit $FEC
}
########################################################################
# Name        : DisplayMessage
#
# Description : The function displays message
#
# Input       : 1. Message type (E = Error, I = Information)
#               2. Error code defined in DefineMessages().
########################################################################
DisplayMessage ( )
{
MESSAGE_TYPE=$1
MESSAGE_TEXT=`eval echo $2`
MoveCursor 24 1
if [ "${MESSAGE_TYPE}" = "E" ]
then
    echo "${ERROR}${MESSAGE_TEXT}\c"
else
    echo "${INFO}${MESSAGE_TEXT}\c"
fi
sleep ${SLEEP_DURATION}
return ${TRUE}
}
########################################################################
# Name       : ProcessExit
#
# Description: Removes all temporary files and makes a graceful exit.
#
# Input      : Exit code
########################################################################
ProcessExit ()
{
#
# assign parameter
#
```

```
EXIT_CODE="$1"
rm -f ${TEMP_FILE}
clear
exit $EXIT_CODE
}
#######################################################################
# Name        : DisplayListOfValues
#
# Description: Displays a list of valid system users and allows the
#              user to select one.
#######################################################################
DisplayListOfValues ()
{
echo "      List of valid System Users " > ${TEMP_FILE}
echo "      ========================== " >> ${TEMP_FILE}
echo " Delete the corresponding line and save the file " >> ${TEMP_FILE}
echo " in order to select a user from the list \n"       _>>
➤  ${TEMP_FILE}
cat /etc/passwd | cut -d':' -f1 >> ${TEMP_FILE}
cp ${TEMP_FILE} ${TEMP_FILE_1}
vi ${TEMP_FILE}
SELECTED_VALUE=`diff ${TEMP_FILE} ${TEMP_FILE_1} | tail -1 | \
               awk {'print $2'}`
}
#######################################################################
# Name        : RootUser
#
# Description: Checks whether the user is root.
#
# Returns     : TRUE if user is root
#              FALSE otherwise
#######################################################################
RootUser ()
{
USER=`id | cut -d'(' -f2 | cut -d')' -f1`
if [ "${USER}" = "root" ]
then
    return $TRUE
else
    return $FALSE
fi
}
#######################################################################
# Name        : PrintSpoolFile
#
# Description: Prints a spool file.
#
# Input       : Spool file name, spool file deletion indicator.
#
# Notes       : 1. Print job function calls this function with the name
#                 of the spool file.
#
#              2. The spool file is removed after being printed.
########################################################################
PrintSpoolFile ( )
{
#
# assign parameter
#
FILE_TO_BE_PRINTED=$1
DELETE_FILE=$2
if [ "${DELETE_FILE}" = "" ]
then
    DELETE_FILE=${TRUE}
fi
#
# ask user
#
while true
do
    echo "Do you wish to print the output (Y/N)?:\c"
    read PRINT_REQUEST
    case ${PRINT_REQUEST} in
        Y|y) break ;;
        N|n) chmod u+w ${FILE_TO_BE_PRINTED};
             if [ "${DELETE_FILE}" = "${TRUE}" ]
             then
                 rm ${FILE_TO_BE_PRINTED};
             fi
             return ${TRUE} ;;
          *) : ;;
    esac
done
#
# get the printer name
#
echo "Enter the printer name or press enter to accept default(${RVON}
➤  ${DEFAULT_PRINTER}${RVOFF}):\c"
read PRINTER_NAME
if [ -z "${PRINTER_NAME}" ]
then
    PRINTER_NAME=${DEFAULT_PRINTER}
fi
#
# print the file
#
lp -d${PRINTER_NAME} ${FILE_TO_BE_PRINTED} > /dev/null 2>&1
#
# check the return code
#
```

```
if [ $? -eq 0 ]
then
    :
else
    echo "${ERROR}${PRINT_NOT_OK}"
    sleep ${SLEEP_DURATION}
fi
}
########################################################################
# Name       : MoveCursor
#
# Input      : Y and X coordinates
#
# Returns    : None
#
# Description: Moves the cursor to the required location (Y, X).
#
# Notes      : 1. Function must be run in ksh.
########################################################################
MoveCursor ( )
{
YCOR=$1
XCOR=$2
# print -n "${ESC}${YCOR};${XCOR}H"
echo "${ESC}${YCOR};${XCOR}H"
}
########################################################################
# Name       : GetUserName
#
# Description: Gets a valid user name.
#
# Returns    : TRUE if a valid user is entered
#              FALSE if user wishes to quit
########################################################################
GetUserName ()
{
while true
do
    clear
    echo "Enter crontab username (l=list of values q=quit):\c":
    read CRON_USER
    case $CRON_USER in
         "") DisplayMessage E "${INVALID_ENTRY}" ;;
        l|L) DisplayListOfValues  ;
             if [ "${SELECTED_VALUE}" = "" ]
             then
                 continue ;
             else
                 CRON_USER="${SELECTED_VALUE}" ;
                 return $TRUE ;
             fi ;;
        q|Q) return $FALSE ;;
          *) # validate this user
             #
             if ValidSystemUser ${CRON_USER}
             then
                 return $TRUE ;
             else
                 DisplayMessage E "${INVALID_USER}" ;
             fi ;;
    esac
done
}
########################################################################
# Name       : DisplayCRONMenu
#
# Description: Displays menu for cron management
########################################################################
DisplayCRONMenu ( )
{
clear
echo ""
echo "#############################################"
echo "#                                          #"
echo "#      ${CRON}                              #"
echo "#                                          #"
echo "#       5. View crontab file                #"
echo "#      10. Create/modify crontab file       #"
echo "#      15. Grant use of cron facility       #"
echo "#      20. Revoke use of cron facility      #"
echo "#      25. View cron log file               #"
echo "#      30. Edit cron log file               #"
echo "#      35. List valid cron Users            #"
echo "#      40. View cron.allow file             #"
echo "#      45. View cron.deny file              #"
echo "#      99. Exit                             #"
echo "#                                          #"
echo "#############################################"
echo "      Please enter option -->\c"
}
########################################################################
# Name       : ValidSystemUser
#
# Description: Validates a user name received as input and establishes
#              whether the user name is valid.
#
# Input      : User name
#
# Returns    : TRUE if the user is valid
#              FALSE otherwise
```

```
#######################################################################
ValidSystemUser ( )
{
#
# assign parameter
#
P_NAME="$1"
#
# scan the passwd file
#
cat /etc/passwd |cut -d':' -f1 | while read UID
do
    if [ "${UID}" = "#" ]
    then
        continue
    elif [ "${P_NAME}" = "${UID}" ]
    then
        return $TRUE
    fi
done
#
# user does not exist in passwd file
#
return $FALSE
}
#######################################################################
# Name       : ValidCronUser
#
# Description: Checks whether a user in $CRON_USER is valid.
#
# Returns    : TRUE or FALSE
#######################################################################
ValidCronUser ( )
{
#
# check for existence of CRON_ALLOWED_FILE and
# CRON_DISALLOWED_FILE file
#
if [ ! -f "${CRON_ALLOWED_FILE}" ] && [ ! -f "${CRON_DISALLOWED_FILE}" ]
then
    #
    # no one except root is allowed to use cron
    #
    return ${FALSE}
fi
#
# check cron.allow file
#
if [ -f "${CRON_ALLOWED_FILE}" ]
then
```

```
    #
    # check the entries
    #
    cat ${CRON_ALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo "${LINE}" | cut -c1-1`
        if [ "${COMMENT}" = "#" ]
        then
            continue
        elif [ "${LINE}" = "${CRON_USER}" ]
        then
            return $TRUE
        fi
    done
    return $FALSE
fi
#
# check cron.deny file
#
if [ -f "${CRON_DISALLOWED_FILE}" ]
then
    #
    # check the entries
    #
    cat ${CRON_DISALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo "${LINE}" | cut -c1-1`
        if [ "${COMMENT}" = "#" ]
        then
            continue
        elif [ "${LINE}" = "${CRON_USER}" ]
        then
            return $FALSE
        fi
    done
    return $TRUE
fi
}
########################################################################
# Name       : ViewUserCrontabFile
#
# Description: View user crontab file.
#
# Returns    : $TRUE or
#              $FALSE
########################################################################
ViewUserCrontabFile ( )
{
#
# get user name in $CRON_USER
```

```
#
if ! GetUserName
then
    return $FALSE
fi
#
# display existing crontab file
#
CRONTAB_FILE="${CRONTAB_DIR}/${CRON_USER}"
#
# Check for crontab file existence
#
if [ ! -f ${CRONTAB_FILE} ]
then
    DisplayMessage E "${NO_CRONTAB_FILE}"
    return $FALSE
fi
#
# crontab file exists for the user copy it into /tmp
#
cp $CRONTAB_FILE ${TEMP_FILE}
clear
view ${TEMP_FILE}
return $TRUE
}
#########################################################################
# Name       : CreateOrModifyCrontabFile
#
# Description: Allows the user to modify an existing crontab file or
#              create a new one for a specified user.
#
# Returns    : $TRUE or
#              $FALSE
#
# Notes      : 1. The function calls the following functions:
#                 o GetUserName
#                 o DisplayMessage
#                 o ValidCronUser
#########################################################################
CreateOrModifyCrontabFile ( )
{
#
# get user name
#
if ! GetUserName
then
    return $FALSE
fi
#
# check for valid cron user
```

```
#
if ! ValidCronUser "${CRON_USER}"
then
    DisplayMessage E "${CRON_DISALLOWED}"
    return $FALSE
fi
#
# prepare crontab file name
#
CRONTAB_FILE="${CRONTAB_DIR}/${CRON_USER}"
#
if [ "${CRON_USER}" = "root" ]
then
    crontab -e 2> ${TEMP_FILE}
else
    su ${CRON_USER} -c "crontab -e"
fi
}
########################################################################
# Name       : GrantCronUsage
#
# Description: Grants cron usage to a user.
#
# Returns    : TRUE if successful
#              FALSE otherwise
#
# Notes      : 1. The function uses the global variable $CRON_USER.
########################################################################
GrantCronUsage ( )
{
#
# get the userid in $CRON_USER to whom grant must be made
#
if ! GetUserName
then
    return $FALSE
fi
#
# check existence of cron.allow and cron.deny files
#
if [ ! -f "${CRON_ALLOWED_FILE}" ] &&
➤  [ ! -f "${CRON_DISALLOWED_FILE}" ]
then
    #
    # initially, no one except root is allowed to use cron -
    # initialize cron.allow file with this user name
    #
    > ${CRON_ALLOWED_FILE}
    echo "${CRON_USER}" >> ${CRON_ALLOWED_FILE}
    DisplayMessage I "${MADE_CRON_GRANT}"
    return ${TRUE}
```

```
fi
#
#  check the existence of cron.allow file
#
if [ -f ${CRON_ALLOWED_FILE} ]
then
    #
    # file exists
    #
    cat ${CRON_ALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo "${LINE}" | cut -c 1-1`
        if [ "${COMMENT}" = "#" ]
        then
            continue
        elif [ "${LINE}" = "${CRON_USER}" ]
        then
            DisplayMessage I "${CRON_GRANT_OK}"
            return $TRUE
        fi
    done
    #
    # file exists, but entry is missing
    #
    echo ${CRON_USER} >> ${CRON_ALLOWED_FILE}
    DisplayMessage I "${MADE_CRON_GRANT}"
    return $TRUE
fi
#
# cron.allow file does not exist
# check the existence of cron.deny file
#
if [ -f ${CRON_DISALLOWED_FILE} ]
then
    #
    # file exists
    #
    cat ${CRON_DISALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo "${LINE}" | cut -c 1-1`
        if [ "${COMMENT}" = "#" ]
        then
            continue
        elif [ "${LINE}" = "${CRON_USER}" ]
        then
            #
            # remove this line
            #
            ed - ${CRON_DISALLOWED_FILE} <<!
/${CRON_USER}/d
w
```

```
q
!
            DisplayMessage I "${MADE_CRON_GRANT}"
            return $TRUE
        fi
    done
fi
#
# cron.deny file exists with no entry for required user
#
DisplayMessage I "${CRON_GRANT_OK}"
}
########################################################################
# Name       : ListValidCronUsers
#
# Description: Lists all the valid cron users.
########################################################################
ListValidCronUsers ( )
{
#
# write report header
#
echo "           List of valid cron users" > ${TEMP_FILE}
echo "           ========================\n" >> ${TEMP_FILE}
if [ ! -f "${CRON_ALLOWED_FILE}" ] &&
➤  [ ! -f "${CRON_DISALLOWED_FILE}" ]
then
    #
    # no one except root is allowed to use cron
    #
    echo "root" >> ${TEMP_FILE}
    view ${TEMP_FILE}
    PrintSpoolFile ${TEMP_FILE}
    return ${TRUE}
fi
#
# check for existence for CRON_ALLOWED_FILE
#
if [ -f "${CRON_ALLOWED_FILE}" ]
then
    cat ${CRON_ALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo $LINE | cut -c 1-1`
        if [ "${COMMENT}" = "#" ]
        then
            # ignore this entry
            contine
        fi
        NAME=`echo ${LINE} | awk '{print $1}'`
        echo "$NAME" >> ${TEMP_FILE}
    done
    view ${TEMP_FILE}
    PrintSpoolFile ${TEMP_FILE}
    return ${TRUE}
fi
# all users except those in CRON_DISALLOWED_FILE are allowed
# to use cron
#
DisplayMessage I "${WORKING}"
cat  /etc/passwd | while read LINE
do
    COMMENT=`echo $LINE | cut -c 1-1`
    if [ "${COMMENT}" = "#" ]
    then
        #
        # ignore this entry
        #
        continue
    fi
    NAME=`echo $LINE | cut -d':' -f1`
    #
    # does it exist in CRON_DISALLOWED_FILE
    #
    if [ -f "${CRON_DISALLOWED_FILE}" ]
    then
        COMMENT=`grep "${NAME}" ${CRON_DISALLOWED_FILE} | cut -c 1-1`
        if [ "${COMMENT}" = "#" -o "${COMMENT}" = "" ]
        then
            #
            # valid user of cron
            #
            echo "$NAME" >> ${TEMP_FILE}
        fi
    else
        #
        # valid user of cron
        #
        echo "$NAME" >> ${TEMP_FILE}
    fi
done
view ${TEMP_FILE}
PrintSpoolFile ${TEMP_FILE}
return ${TRUE}
}
#######################################################################
# Name       : RevokeCronUsage
#
# Description: Revokes cron permission from a user.
#
# Returns    : TRUE or
#               FALSE
########################################################################
RevokeCronUsage ( )
{
#
# get userid
#
if ! GetUserName
then
    return $FALSE
fi
#
# check the user's current grant
# check the userid in CRON_ALLOWED_FILE
#
if [ ! -f "${CRON_ALLOWED_FILE}" ] &&
➤  [ ! -f "${CRON_DISALLOWED_FILE}" ]
then
    #
    # no one except root is allowed to use cron
    # make an entry for $CRON_USER in ${CRON_DISALLOWED_FILE} file
    #
    echo "${CRON_USER}" > ${CRON_DISALLOWED_FILE}
    return ${TRUE}
fi
#
# check cron.allow file
#
if [ -f "${CRON_ALLOWED_FILE}" ]
then
    #
    # check the entries
    #
    cat ${CRON_ALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo "${LINE}" | cut -c1-1`
        if [ "${COMMENT}" = "#" ]
        then
            continue
        elif [ "${LINE}" = "${CRON_USER}" ]
        then
            #
            # delete this entry
            #
            ed - ${CRON_ALLOWED_FILE} <<!
/${CRON_USER}/d
w
q
!
            DisplayMessage I "${REVOKED_CRON_GRANT}"
            return $TRUE
        fi
    done
fi
#
# check cron.deny file
#
if [ -f "${CRON_DISALLOWED_FILE}" ]
then
    #
    # check the entries
    #
    cat ${CRON_DISALLOWED_FILE} | while read LINE
    do
        COMMENT=`echo "${LINE}" | cut -c1-1`
        if [ "${COMMENT}" = "#" ]
        then
            continue
        elif [ "${LINE}" = "${P_NAME}" ]
        then
            DisplayMessage I "${CRON_GRANT_ALREADY_REVOKED}"
            return $TRUE
        fi
    done
    #
    # put in an entry
    #
    echo "${P_NAME}" >> ${CRON_DISALLOWED_FILE}
    DisplayMessage I "${REVOKED_CRON_GRANT}"
    return $TRUE
fi
}
#######################################################################
# Name       : ViewCronLog
#
# Description: Allows user to view the cron log file.
#
# Returns    : $TRUE or
#              $FALSE
#######################################################################
ViewCronLog ( )
{
LOG_FILE=${CRON_LOG_DIR}/${CRON_LOG_FILE}
#
# check for existence of the file
#
if [ ! -f "${LOG_FILE}" ]
then
    DisplayMessage E "${NO_CRON_LOG}"
    return ${FALSE}
```

```
fi
#
# view the file
#
cp ${LOG_FILE} ${TEMP_FILE}
view ${TEMP_FILE}
rm ${TEMP_FILE}
return $TRUE
}
#######################################################################
# Name        : TrimCronLog
#
# Description: Allows the user to edit the cron log file.
#######################################################################
TrimCronLog ( )
{
LOG_FILE=${CRON_LOG_DIR}/${CRON_LOG_FILE}
#
# check for existence of the file
#
if [ ! -f "${LOG_FILE}" ]
then
    DisplayMessage E "${NO_CRON_LOG}"
    return ${FALSE}
fi
#
# view and edit file
#
vi ${LOG_FILE}
return $TRUE
}
#######################################################################
# Name        : ViewCronAllowFile
#
# Description: Allows the user to view the cron.allow file.
#######################################################################
ViewCronAllowFile ()
{
#
# check for existence of the file
#
if [ ! -f "${CRON_ALLOWED_FILE}" ]
then
    DisplayMessage E "${NO_CRON_ALLOWED_FILE}"
    return ${FALSE}
fi
#
# allow user to view and edit it
#
echo "             Details of cron.allow  File " > ${TEMP_FILE}
echo "              =========================== " >> ${TEMP_FILE}
cat ${CRON_ALLOWED_FILE} >> ${TEMP_FILE}
view ${TEMP_FILE}
return $TRUE
}
#######################################################################
# Name       : ViewCronDenyFile
#
# Description: Allows the user to view the cron.deny file.
#
# Returns    : $TRUE or
#              $FALSE
#######################################################################
ViewCronDenyFile ()
{
#
# check for existence of the file
#
if [ ! -f "${CRON_DISALLOWED_FILE}" ]
then
    DisplayMessage E "${NO_CRON_DISALLOWED_FILE}"
    return ${FALSE}
fi
#
# allow user to view and edit it
#
echo "              Details of cron.deny File " > ${TEMP_FILE}
echo "              ========================= " >> ${TEMP_FILE}
cat ${CRON_DISALLOWED_FILE} >> ${TEMP_FILE}
view ${TEMP_FILE}
return ${TRUE}
}
#######################################################################
# Name       : main
#
# Description: Displays and processes menu options.
#
# Notes      : 1. The function calls the following functions:
#                 o DefineVariables
#                 o RootUser
#                 o DisplayMessage
#                 o ProcessExit
#                 o DisplayCRONMenu
#                 o ViewUserCrontabFile
#                 o CreateOrModifyCrontabFile
#                 o GrantCronUsage
#                 o RevokeCronUsage
#                 o ViewCronLog
#                 o TrimCronLog
#                 o ListValidCronUsers
#                 o ViewCronAllowFile
#                 o ViewCronDenyFile
#                 o ProcessExit
########################################################################
main ()
{
DefineVariables
trap "HandleInterrupt " $SIGINT $SIGTERM $SIGHUP $SIGTSTP
if ! RootUser
then
    DisplayMessage I "${ROOT_USER}"
    ProcessExit $FEC
fi
while true
do
    DisplayCRONMenu
    read OPTION
    case $OPTION in
         5) clear ;
            ViewUserCrontabFile ;;
        10) clear ;
            CreateOrModifyCrontabFile ;;
        15) clear ;
            GrantCronUsage ;;
        20) clear ;
            RevokeCronUsage ;;
        25) clear ;
            ViewCronLog ;;
        30) clear ;
            TrimCronLog ;;
        35) clear ;
            ListValidCronUsers ;;
        40) clear ;
            ViewCronAllowFile ;;
        45) clear ;
            ViewCronDenyFile ;;
        99) ProcessExit $SEC ;;
         *) DisplayMessage E "${INVALID_ENTRY}" ;;
    esac
done
}
#
# invoke main
#
main
```

Arif Zaman
DBA/Administrator
High-Tech Software (UK)

System and network back-up

AIX is clearly among the most complete and flexible versions of the Unix operating system, but along with that flexibility comes complexity. It took us quite some time to create the volume groups, logical volumes, filesystems, and paging space that are required by the operating system. Then came the painful task of performance and availability tuning through the selection of disk placement, mirroring and striping of logical volumes, and tuning of network and system parameters.

So now for the big question: if the entire system, or any part of it, goes belly-up, how do you get it all back to where it was? This question has stumped many AIX administrators over the years. Unfortunately, in many cases, the question has not been adequately answered until the 'if' was no longer necessary.

IS MKSYSB ENOUGH?

Don't be fooled! **mksysb** is a standard back-up utility that has always been a part of AIX, but don't assume that as a 'system' back-up utility it backs up (and – more importantly – recovers) everything. **mksysb** only backs up the *rootvg* volume group. With AIX Version 3, the rest is up to you to figure out. With AIX Version 4, a separate utility, **savevg**, is used to back up other volume groups.

In AIX Version 3.2 (don't even ask about 3.1!), **mksysb** does little more than take a 'back-up' of filesystems in the root volume group to a bootable tape. While it does manage, in most cases, to recover the device configuration of the system, everything gets reinstalled in default locations, and any mirrored and non-filesystem (raw logical volume) data is not included in the back-up.

AIX Version 4 includes new versions of **mksysb**, as well as the **savevg** utility. These do a much better job of saving logical volume definitions and retaining paging space, mirroring, striping, partition placement, etc, though they still don't back up raw logical volume data.

If you reinstall from a back-up made with the latest version of

mksysb, you're likely to get the system back the way you want it, as long as you're installing on a system with the same hardware configuration. Any change in configuration, and the result is often 'roughly the same', as though you had reinstalled from a back-up made using AIX Version 3.2's **mksysb**. There are ways around this – by altering some text files prior to backing up, or creating files on diskette and using them during the boot process – but this is a bit too complex to get into here.

SYSBACK TO THE RESCUE

Originally known as 'AIX System Back-up and Recovery/6000', and now officially known as 'AIX Operational Services Back-up and Recovery', this tool will forever be known as Sysback. Sysback was originally developed in AIX Technical Support in the USA in response to customer complaints over the lack of recovery features in **mksysb** (this was back in the days of AIX 3.1, when **mksysb** 3.1 couldn't make a bootable tape or recover device configuration). As new features were added to AIX, such as disk mirroring, Sysback quickly incorporated the ability to capture this information, while **mksysb** lagged behind for quite some time.

Although **mksysb** eventually (in AIX 4.1) incorporated many of the features of Sysback, Sysback continued to progress rapidly, and has gained many advanced features, such as the ability to back up to (and install from) a remote tape drive and stripe back-ups across tape drives. Sysback also added a fully customizable installation menu, allowing users to change the disk placement and virtually any other characteristic of a volume group, logical volume, or filesystem. In addition, the tool includes all volume groups in a single back-up, also including raw logical volume data that's still missing from **mksysb** back-ups today. Sysback also added the ability to perform back-ups of selected filesystems, logical volumes, and volume groups, allowing it to serve as a single back-up product for most installations.

Sounds perfect! So what's the downside? Well, Sysback is not included in AIX, but must be purchased separately. While this is not an issue for anyone who knows the importance of a reliable back-up product, the fact that it is available only through IBM Direct (1-800-

CALL-IBM) and is supported only through AIX SupportLine (1-800-CALL-AIX) in the USA probably is. Customers outside the USA may purchase Sysback from IBM as an AISPO (‘As-Is’ Software Product Offering), though availability and support is country-dependent and sometimes non-existent.

Secondly, while Sysback’s ability to back up and restore various types and levels of data make it very popular in single-system or small network environments, it becomes increasingly unfriendly as more systems must be managed by a single administrator. With Sysback, you may perform a back-up to a remote server’s back-up media only by initiating the back-up from the client. Sysback also provides no scheduling, reporting, tape re-tension, or error notification facilities, as are usually provided in network back-up products. For an administrator, this means creating a lot of shell scripts, redirecting error messages to files, and setting up **cron** jobs. Set-up and maintenance of these back-ups can become increasingly complex and tiring.

WHAT ABOUT ADSM, NETWORKER, AND OTHERS?

There are many good back-up products, IBM’s ADSM and Legato’s Networker being two popular ones. ADSM is a heterogeneous tape library management product, giving you the ability, from an IBM PS/2, to back up and catalog files on different systems to your mainframe. Networker will back up files on any type of Unix system, providing many of the same tape library management and file cataloguing features as ADSM. The product has many back-up management features, far too many to mention here, that can automate back-ups, tape re-tension, file archiving, etc, in a large network environment.

The drawback is that these products can be quite expensive, which is not surprising when you consider the complexity of the task they perform – cataloguing multiple copies of every file on every system, filing them somewhere in a tape library, then quickly and automatically retrieving a specific file from that library when called upon to do so.

In addition, while products designed for multiple platforms will back up files and directories, they know little or nothing about the elements of your AIX ‘system’, such as volume groups, paging space, logical volumes, etc. These back-ups allow you to restore files, but not your whole operating system. This being the case, these products are almost always supplemented by another program that takes care of the system stuff.

SYSBACK AND STORIX BACK-UP ADMINISTRATOR

Storix Back-up Administrator for AIX (SBA) is not only designed by the creator of Sysback but also includes Sysback and all its features. In addition, it also provides a graphical interface for administering back-ups, restores, and system installs from a central server. It is also available in a stand-alone version for systems that do not require network back-up features.

Unlike IBM's ADSM, SBA is not a tape library management product. Randomly backing up and cataloging thousands of files requires a complex and expensive feature that is needed only in very large (and usually multi-platform) installations. SBA was developed for use exclusively in an AIX environment and includes most of the features that the majority of customers require without imposing on them the high cost of features many don't need.

In addition to the central administration of network back-ups, SBA also provides Sysback with the ability to schedule unattended back-up jobs, to create tape overwrite and re-tension policies, to send back-up notification, report errors, create historical reports, and more.

Best of all, Storix Back-up Administrator for AIX is available worldwide, making the powerful features of Sysback available to everyone (nevertheless, most customers will still have to purchase directly from Storix in the USA, though resellers are available in some countries). More information on SBA, as well as a free evaluation copy, can be obtained from Storix' Web site, at: *http://www.storix.net*.

BACK-UP AND RECOVERY OF SP NODES

The system back-up and recovery of SP nodes is a long-standing problem at SP installations. SP customers have long complained about the difficult task of backing up and restoring nodes, as they have no console or front panel. IBM's PSSP (Parallel System Support Product) software is included with the SP system, providing utilities for the installation and management of nodes from the control workstation (CWS). Until recently, these have been the only tools available to aid customers with the recovery of nodes. The list below shows all the options available today for backing up and recovering SP nodes.

PSSP software

IBM's PSSP, which is used for SP system management, provides a node installation process using NIM (Network Installation Management), installing nodes from a pre-configured **mksysb** disk back-up. Although this is useful for the initial installation of SP nodes, it proves very cumbersome in either periodic system back-ups or the reinstallation of nodes for the following reasons:

1 Nodes must be backed up to disk as NIM does not support remote tape installation. This is very costly in disk space when storing back-ups of several nodes.

2 The back-up and installation includes only the *rootvg* volume group. All other volume groups must be created manually and restored from a separate back-up.

3 The installation process provides no prompts and is not customizable.

mksysb

By itself, a **mksysb** installation is not an option, unless a tape drive containing the required back-up is attached directly to the node. If so, the node must be booted using manual node conditioning from the control workstation. However, when installation is complete, the customized node data is not recovered from the back-up, so the node will not be customized using data in the System Data Repository (SDR) on the control workstation.

Sysback

This provides a system back-up that may be restored across the network from a tape drive attached to the server. It also allows the customization of volume group, logical volume, and filesystem attributes when installing the back-up. Like **mksysb**, however, the node must be booted using manual node conditioning, and the customized node data is neither recovered from the back-up nor re-applied from the System Data Repository (SDR).

SBA

Storix Back-up Administrator for AIX is designed to ease the process of booting and installing SP nodes, allowing the node to be completely reinstalled from a remote disk image or tape drive and automatically customizing the node when the restore is complete. Since the node is customized using SDR data on the control workstation, the node may be cloned from a back-up that originated from another node, even if the hardware configuration of the two nodes is different. The following SP system features are built into SBA:

1 SBA provides a graphical node 'front panel' from which the target node may be booted, installed, and monitored by pressing a single 'network boot' button. The node's system LED is also displayed on the front panel to monitor the progress of the boot process, and a console window is opened for the node in order to monitor the progress of both the network boot and installation.

2 If the node was configured for a prompted installation, all of the features of the Sysback installation process, allowing for full customization of volume group and logical volume placement, sizes, and other attributes.

3 Once the installation process is complete, the node is automatically customized using data in the System Data Repository (SDR).

PRODUCT COMPARISON AT A GLANCE

Figure 1 shows the most common back-up and recovery features required by users in AIX networked environments and the features provided by products from IBM, Legato, and Storix. Note that IBM Network Installation Manager (NIM) provides more than just back-up and restore functionality, also handling product installation and updates. IBM's ADSM and Legato's Networker are examples of tape library management and file and tape cataloguing products, each of

Feature	mksysb	NIM	Sysback	ADSM/Legato	Storix (SBA)
Interface	**smit** (ASCII)	**smit** (ASCII)	**smit** (ASCII)	Graphical	Graphical
Network back-up	None	Back-up manually started from server	Back-up manually started from the client	Centrally managed	Centrally managed
Full system back-up/ install	Only for rootvg (use **savevg** command for other VGs)	Uses **mksysb**, must back up and restore from disk file (no support for tape)	Yes	N/A	Yes
Custom install from system back-up	Yes, edit text files before backing up	N/A	Yes	N/A	Yes
Cloning systems (custom install)	Yes, edit text files before backing up	No	Yes, all configuration options are customizable on installation	N/A	Yes, all configuration options are customizable on installation
Install/ clone SP nodes	No	Uses **mksysb** back-up on disk, with automatic customization	No	N/A	Installs from remote tape or disk file, with automatic customization
Unattended system installation	User must edit config files before back-up	Yes, but prompts not supported	Yes	N/A	Yes
Data backed up	rootvg only	rootvg only	System, VGs, filesystems, LVs, and directories	Directories and files only	System, VGs, filesystems, LVs, and directories
Incremental back-ups	No	No	Files and logical volumes	Files only	Files and logical volumes
Tape capacity	One system back-up per tape	N/A	Manually append multiple back-ups	Random placement of different back-ups	Automatically appends multiple back-ups to tape

Figure 1: Product feature summary

Feature	mksysb	NIM	Sysback	ADSM/Legato	Storix (SBA)
Track tape contents	No	N/A	No	Yes, catalogs all files and tapes in libraries	Yes, tracks back-ups on tape
Enhanced perfor-mance	Software compression	Software compression	Software compression, parallel tape writes	Software compression	Software compression, parallel tape writes
Unattended multi-tape back-up	No, user changes volumes	N/A	Yes, uses sequential autoloader or multiple tape drives	Yes, uses random tape libraries	Yes, uses sequential autoloader or multiple tape drives
Multi-copy back-ups	No	No	Yes	No	Yes
Back-up schedules	No	No	No	Yes	Yes
Overwrite protection	No	N/A	No	Yes	Yes
Back-up retention	No	N/A	No	Yes	Yes
Back-up history	No	No	No	Yes	Yes

Figure 1: Product feature summary (continued)

which provides several features beyond the topics discussed in this article.

Anthony Johnson (USA) © Storix Software 2000

‘more’ exercises

This short article complements an article of mine published in Issue 52 of *AIX Update* (*Understanding the* ***more*** *command*). The earlier article describes how the command **more** is used, while this one presents two exercises to test your knowledge of the command.

EXERCISE 1

For this exercise, choose a relatively long file with which you're familiar, preferably one in which you would easily recognize the location of most parts of the file. A *README* file can work well, though a file that contains at least 100 numbered lines would be best. For the sake of the exercise, the file is assumed to be called *readme.now*. Change to the directory containing the file or copy the file into a test directory, then carry out the following steps:

1 Enter **more readme.now**. Note that the first screen of text is displayed. The **more** prompt appears at the bottom, followed by a designation of the percentage of file displayed so far.

2 Observe how the screens change as you press the spacebar. Use the **f** and **b** keys to scroll forwards and backwards one screen at a time. Notice how the percentages change.

3 Observe how the screens change as you press the *ENTER* key. Use the **j** and **k** keys to scroll forwards and backwards line by line. Type **3j** and watch the file scroll down three lines.

4 Use the **d** and **u** keys to scroll down and up by half a screen. Type **5d** and notice that, whenever you press **d** or **u**, the text now scrolls down or up by five lines.

5 Use the lowercase and uppercase **g** keys to scroll to the first and to the last lines of the file. Type a number followed by **g** (with no space), such as **65g**, to go directly (in this case) to line 65.

6 Type **50%** to move to midway through the file. Type **10%** and note that the focus is positioned about ten percent from the top of the file.

7 Type **g** to return to the top of the file, then search for a word that you know occurs a number of times in the file by entering a forward slash ('/') followed by the word, such as **/develop**. The first occurrence of the word is displayed. Each time you press **n**, the next occurrence of the word is displayed.

8 Enter **?develop** to see the previous occurrence of the word. Now, as you press **n**, you move backwards through the file to the previous occurrence of the word.

9 Note the line that is currently ‘in focus’ and type **v** at the **more** prompt. Observe that the **vi** editor is invoked and the cursor remains positioned at the same line. Type **:q** within the **vi** editing session to leave **vi** and return to the **more** session.

Note that, if you had entered **more -N readme.now** in step 1, the **vi** editor would initialize at line 1 no matter where you were in the document when the **v** command was entered.

10 Type **q** to end the **more** session.

EXERCISE 2

For this exercise, we pipe the output of a command to **more**. For this exercise, we choose a directory that contains many files, preferably ones with names that span an alphabetically broad range. So, we choose the directory */usr/bin*.

1 Enter **ls -l /usr/bin | more** and note that the first group of files contained in the directory are displayed. There is no indication of the percentage of output already displayed, as you’d see if you were viewing a file.

2 Observe how the screen changes as you use the spacebar, *ENTER* key, **f**, **b**, **d**, **u**, **j**, and **k** keys.

3 Use the positioning keys above to locate the first file name that begins with the letter ‘h’. With that line in focus, type **mh**, which sets a marker at that point called ‘h’. Now position the focus at the first file with a name that begins with the letter ‘p’ and type **mp**, which sets a marker there called ‘p’. Type **'h** (a single quote followed by an ‘h’) and you should find yourself back at the first file name that begins with an ‘h’. Type **'p** and the first file name beginning with ‘p’ should be displayed. If you type **''** (two single quotes, not a double quote) you should find yourself where you were before your most recent move (in this case, at the marker ‘h’).

4 Enter **!ls**, and the **more** session is suspended, so that the results of the original AIX command are displayed and you see a list of the contents of the current directory. Press the *ENTER* key to return to the **more** session.

5 Type the ‘h’ key to display the **more** command’s help screens. Since the help file has more than one screen, you can use **more** commands to view the help file itself. Type **q** followed by *ENTER* to leave the help display and return to the file list.

6 Type **q** to leave the more session. Then enter **ls -l /usr/bin | more -d -p 30G** and note that the **more** command displays the results of the list command and immediately scrolls to line 30. You should also notice that a help line appears with the **more** prompt.

7 Type **q** to end the **more** session.

David Chakmakian
Programmer (USA)

Using top to identify CPU performance problems

top is a Unix utility that generates continuous reports on the state of the system, including a list of the fifteen processes that are consuming the most CPU time. Version 3 of **top** has three primary design goals:

- To provide an accurate snapshot of the state of both the system and processes that are currently running.
- Not to be one of the top CPU-consuming processes itself.
- To be as portable as possible.

top was inspired by a VMS command that lists the five ‘busiest’ processes along with a bar chart of their CPU utilization. While the VMS command just shows the name of the process and its CPU consumption as a percentage, when the utility was ported to Unix its developers decided that the output of the Unix version should be as informationally rich as the **ps** command.

top displays information about a system and the processes that it’s running. By default, the processes are sorted by CPU utilization, which is expressed as percentages. The display is updated at regular intervals to reflect the current state of the system. **top** is a handy tool for determining what the system is doing.

SAMPLE OUTPUT

The following diagram shows sample output from my system running AIX 4.2.1 (the output has been cosmetically changed to allow it to fit the page width):

```
load averages:  0.97,  0.77,  0.74                         20:20:53
71 processes:  65 idle, 2 zombie, 4 running
CPU states:  0.0% idle, 70.5% user, 27.2% kernel,  2.1% wait
Memory: 320M Total. Real: 192M, 20M Buffers. Virtual: 128M, 31M Free

  PID USERNAME PRI NICE SIZE   RES STATE   TIME   WCPU    CPU COMMAND
18604 root      0   20   48M   34M run     4:47 58.22% 87.59% SW95
 3538 root      0   20   11M 5380K run   494:45  0.61%  6.20% X
14248 werner    0   20 3364K 1716K sleep   8:55  0.01%  0.84% dtwm
18814 werner    0   20   25M 3068K run     1:10  0.02%  0.42% netscape
15034 werner    0   20  924K  680K sleep   1:23  0.00%  0.37% dtterm
21248 root      0   20  444K  420K run     0:00  0.00%  0.23% top
 4278 werner    0   20  816K  480K sleep   0:30  0.00%  0.15% dtterm
10926 nobody    0   20 4992K  604K sleep  47:56  0.06%  0.11% httpd
20552 root      0   20 1228K  284K sleep   3:05  0.01%  0.05% nmbd
16056 werner    0   20  992K  696K sleep   0:55  0.00%  0.05% dtterm
 6604 root      0   20  196K   72K sleep   0:15  0.00%  0.04% syslogd
    1 root      0   20  348K  216K sleep 124:31  0.15%  0.04% init
 3614 root      0   20   88K   28K sleep  43:40  0.05%  0.03% syncd
20762 root      0   20  968K  364K sleep   0:03  0.00%  0.03% ksh
12534 root      0   20  308K   88K sleep   2:02  0.00%  0.02% pcimapsvr
```

SoftWindows95 (*SW95*) is basically a Windows 95 emulator. The PowerPC processor in my workstation and AIX are pushed to the limit to emulate an Intel processor running Microsoft's hefty operating system. I never log off my workstation, which is why the *X* process has such a large aggregate time of 494 minutes and 45 seconds (remember, this is CPU time, not time since the process was initiated). Something else the output shows is that *X* uses a lot of CPU time. While I was previously aware of this, I didn't realize that it uses so much more than Netscape. As you can see, there are times when my RS/6000 workstation could use more CPU power.

COMPILATION AND INSTALLATION

top comes with a configuration script called *Configure*. This script was developed specifically for **top** – it prompts for various bits of information, then creates the files needed for successful compilation. After successfully running *Configure*, the command **make** can be used to compile the program. If compilation is successful, then 'make install' should install both the executable and the manual page in the correct locations as specified during configuration. Installation usually needs root authority.

top is very sensitive to changes in the data structures maintained by the kernel. Typically, an executable created under one version of the operating system will not run successfully on any other version. On operating systems that run on a variety of hardware platforms, such as IBM's AIX, a separate executable may need to be created for each platform type. When a machine is upgraded to a newer version of the operating system, **top** should be recompiled and reinstalled.

AVAILABILITY

The official home for **top** is *www.groupsys.com/top/*. This site is also provided via FTP at *ftp.groupsys.com*. Mirror sites have been established to provide a wider distribution of new versions as they become available. The download section has details on how one can obtain the source package for **top**.

MONITOR

In some respects the **monitor** tool found on Bull's AIX freeware and shareware Web site and described in the July 1999 issue of *AIX Update* provides more system performance information than **top**. However, an advantage of using **top** is that its freely available source supports almost every combination of Unix and processor.

Werner Klauser
Klauser Informatik (Switzerland)

RS/6000 and AIX announcements

IBM's latest RS/6000 announcements contain new offerings in both low- and high-end systems. In addition, a number of new I/O and communication adapters, as well as improvements to the AIX operating system, were introduced.

This article presents a technical review of the products and new features announced.

RS/6000 7044 MODEL 170

The new RS/6000 44P Model 170 is the successor to the previously available 43P Model 150 workstation. This machine is designed as an entry-level workstation or workgroup server. In fact, this is the first 64-bit uniprocessor-based machine in the RS/6000 range. The Model 170 features either a 333 MHz or 400 MHz POWER3-II microprocessor with improved memory controller, SDRAM memory, and a memory bus speed of 95 MHz on 333 MHz processor systems and 100Mhz on 400 MHz processor systems (this compares with an 83 MHz bus on the Model 150).

System memory can be expanded up to 2 GB. The machine includes an integrated Fast Ethernet adapter, internal and external Ultra SCSI adapters, and a service processor. In addition a tablet, keyboard, and mouse as well as two serial, one parallel, and stereo audio connectors are available as standard. The system provides three disk drive bays, one of which is occupied by the system disk supplied with the machine (9.1 or 18.2 GB). Additional disks can be installed in available media bays. The system is expandable up to 72.8 GB of internal disk storage. Two of the media bays are occupied by 1.44-MB 3.5-inch diskette drive and a 32x maximum speed CD-ROM, leaving one media bay available for an additional peripheral device. The system also includes four 32-bit and two 64-bit PCI slots.

As can be seen from the table overleaf, which compares the performance of newly announced machines with Model 43P 150, floating point and OLTP performance has improved significantly, though integer performance has not grown in line with the other benchmarks.

		7044-170		% Improvement	
Benchmark	7043-150	333 MHz	400 MHz	333 MHz	400 MHz
SPECint_base95	14.5	28.6	23.5	11%	62%
SPECfp_base95	9.8	34	46	250%	371%
SPECint95	15.1	19.8	25.3	31%	68%
SPECfp95	10.1	35.6	47.9	248%	374%
Relative OLTP	6.0	10.4	14.5	73%	142%

RS/6000 7044 MODEL 270

The Model 270 supports one, two, three, or four 375 MHz POWER3-II processors with 4 MB of Level 2 (L2) cache per processor. System memory can be expanded up to 8 GB. The machine comes with an integral Fast Ethernet adapter, internal and external Ultra SCSI adapter, and service processor. Additionally tablet, keyboard, mouse, two serial, one parallel, and stereo audio connectors are available as standard. The system provides two disk drive bays, one of which is taken by the system disk supplied with the machine (9.1 or 18.2 GB). An additional disk can be installed in the available media bay. The system is expandable up to 54.6 GB of internal disk storage when three 18.2 GB disks are installed. Two of the media bays are occupied by a 1.44 MB 3.5-inch diskette drive and a 32x maximum speed CD-ROM. Five PCI slots are available (two are 64-bit and run at 50 MHz and three are 32-bit and run at 33 MHz). All PCI slots are long.

The table below compares the performance of systems with the same number of processors, showing how system performance scales proportionally with processor speed. Notice how the addition of two processors delivers more than twice the OLTP and SPECfp_base_rate95 performance of the Model 260 (IBM has announced the availability of an upgrade from Model 7043 260 to Model 270).

	7043-260	7044-270	
Benchmark	200 MHz	375 MHz	Improvement
SPECint_base95	11.5	22.6	97%
SPECfp_base95	25.1	46	83%
SPECint95	12.5	24.3	94%

	7043-260	7044-270	
Benchmark	200 MHz	375 MHz	Improvement
SPECfp95	27	48.2	79%
SPECint_base_rate95			
1 processor	104	203	95%
2 processors	205	406	98%
3 processors	-	-	-
4 processors	-	811	296% (relative to 2 processors)
SPECfp_base_rate95			
1 processor	225	414	84%
2 processors	434	749	73%
3 processors	-	-	-
4 processors	-	1142	161% (relative to 2 processors)
Relative OLTP			
1 processor	10.5	21	100%
2 processors	21	40.3	92%
3 processors	-	-	-
4 processors	-	68	224% (relative to 2 processors)

RS/6000 SP 375 MHZ POWER3 SMP NODES

New SMP SP nodes based on the same processors as the RS/6000 7044 Model 270 are available in two configurations: a thin node with two disk storage bays and two PCI slots and a wide node with four disk storage bays and ten PCI slots. Both support two or four processors, have an integrated Fast Ethernet adapter, an Ultra2 SCSI interface for internal disks, a high-speed serial port, and a service processor. Connection to the SP switch is by installing the SP Switch MX2 adapter, which attaches directly to the processor mezzanine (MX) bus. Up to 8 GB of ECC SDRAM can be installed using two memory card slots. The table overleaf compares the performance of the new node in wide configuration with that of the previously available POWER3 SMP high node with 200 MHz POWER3 processors. The improvement is relative to the 4-processor configuration, and ranges from 51% to 85%. It’s easy to see that ‘SPEC_rate’ and OLTP performance of the new two and four-way machines are almost equal to that of the previously available four and eight-way models.

Benchmark	POWER3 SMP High node 2-way/4-way	375 MHZ POWER3 SMP Wide node 4-way/8-way	Improvement
SPECint_base95	12.9	22.6	75%
SPECfp_base95	26.3	47.1	80%
SPECint95	14	24.4	74%
SPECfp95	28.6	50.9	78%
SPECint_base_rate95			
2 processors	239	407	-
4 processors	465	812	-
6 processors	690	-	70%
8 processors	905	-	75% (relative to 2/4 processors)
SPECfp_base_rate95			
2 processors	465	804	-
4 processors	900	1359	-
6 processors	1315	-	73%
8 processors	1760	-	51% (relative to 2/4 processors)
Relative OLTP			
2 processors	23	44	-
4 processors	43.3	80	-
6 processors	64	-	91%
8 processors	81.3	-	85% (relative to 2/4 processors)

10,000 RPM SCSI DISK DRIVES

The internal 9.1 GB and 18.2 GB 10,000 RPM SCSI drives, which were announced as internal disk options for selected RS/6000 models, support transfer rates of 800 MBps with the Ultra2 SCSI interface and 40 MBps with the Ultra SCSI interface, also supporting a transfer rate of 20 MBps with the Fast/Wide SCSI interface. The 3.5" by 1" form factor of these disk drives allows them to be used in dense system configurations where 1" drives were previously used. The drives feature an average seek time of 5.1 milliseconds, average latency of 2.99 milliseconds, and maximum media data transfer rate of 44.3 MBps.

PCI UNIVERSAL DIFFERENTIAL ULTRA SCSI ADAPTER

This SCSI-2 differential adapter, which features a maximum data transfer rate of 40 MBps, can connect to disk drives using SCSI-2 F/W or Ultra SCSI interfaces located up to 25 metres from the adapter. The adapter negotiates with each connected device to determine the fastest mutually-supported transfer rate, and it may be installed in either a 3.3 V PCI slot or an older 5 V slot. The adapter features an industry-standard 68-pin connector.

PCI ADVANCED SERIALRAID PLUS ADAPTER

This is a 4-port (2-loop) Serial Storage Architecture (SSA) adapter with an instantaneous data transfer rate of up to 160 MBps per loop. The following RAID levels and initiator combinations are supported:

Raid Level	Number of Initiators
JBOD (non-RAID)	Eight-initiator
RAID 1	Two-initiator
RAID 0 + 1	Two-initiator
RAID 5	Two-initiator
RAID 0	One-initiator

The adapter supports data and parity disk configurations in RAID 5 mode of ‘(2+P)’ up to ‘(15+P)’. Up to six (15+P) RAID 5 arrays can be defined. When operating in RAID 0+1 mode, the adapter supports up to eight mirrored drives. You may also have hot-spares in RAID 5 and RAID 0+1 modes. Any supported RS/6000 system can be set up to boot from an Advanced SerialRAID Plus adapter as long as at least one non-RAID disk is connected to the adapter.

The adapter can be extended using an IBM 32 MB Fast-Write Cache Option card, which improves write performance in RAID 0+1, RAID 5, and JBOD configurations. When this card is installed, the adapter can be configured in single- or dual-initiator fast-write cache mode. In dual-initiator fast-write cache mode, if one of the adapters fails, control is transferred to the other one. In this set-up, the fast-write cache size is 16 MB for each adapter.

An optional 128 MB DRAM card is recommended for dual-initiator, fail-over configurations. This option provides the additional capacity needed to keep a copy of each adapter's 32 MB fast-write cache, and also provides additional working space.

The previously announced Advanced SerialRAID adapter can be updated to the latest features by installing microcode that can be downloaded from the following URL:

```
http://www.hursley.ibm.com/~ssa/rs6k/
```

4-PORT 10/100 BASE-TX ETHERNET PCI ADAPTER

This adapter provides four Ethernet ports on one PCI slot, effectively replacing four separate adapters. The adapter is available only on the following RS/6000 models, which have a limited number of PCI slots: 43P-140, 43P-150, 7046-B50. The adapter communicates at 10 or 100 Mbps, supports full- and half-duplex modes of communication, and has auto-negotiation capability. Both 32- and 64-bit data widths are supported in full-size PCI slots. Each port has an RJ-45 connector for attachment to standard CAT-3/5 Unshielded Twisted Pair (UTP) cable.

ARTIC960RXF DIGITAL TRUNK RESOURCE PCI ADAPTER

This adapter is used in conjunction with the previously announced ARTIC960RxD Quad Digital Trunk PCI adapter (DTXA). It provides additional Digital Signal Processing (DSP) resources to support entry-level voice recognition software, such as IBM's Voice Response for AIX V2.2. It's a single-slot, long format, 32/64-bit PCI adapter operating at a PCI bus speed of 33 MHz.

GIGABIT ETHERNET MEDIA CARDS FOR SP SWITCH ROUTERS

The SP Switch Router is a high-performance I/O gateway for RS/6000 SP systems, providing the fastest means of communication between an SP system and other systems, including other SP systems. This augments existing methods of communication, including 10/100 Ethernet, ATM, HIPPI, and FDDI. The new adapter adds Gigabit Ethernet to this list. A 1000BaseSX media card supports cable lengths of up to 550 metres, and a 1000BaseLX media card supports cable lengths of up to 5 kilometres. Both cards support 1518-byte standard Ethernet packets and 900-byte ‘Jumbo Frames’.

OTHER ENHANCEMENTS

Several AIX OS enhancements were announced concurrently with the new hardware. All the new features, as well as recent PTFs, are available as AIX 4.3.3 maintenance level 02.

AIX 4.3.3 Workload Manager

In addition to the changes to AIX’s internal resource control algorithms that have improved maximum limits and tier enforcement, the new maintenance level provides the following functional enhancements to WLM:

- *Classification of existing processes*

  All processes in existence when WLM starts are now classified, and it’s no longer necessary to stop and re-start applications when WLM is stopped and re-started.

- *Passive mode WLM*

  In the new ‘passive mode’, WLM classifies all processes and gathers statistics about their resource usage without regulating resource allocation. This allows system administrators to gather data on applications’ resource usage without enforcing resource control.

- *Management of application file names*

  When some applications listed in the *rules* file cannot be accessed, **wlmcntrl** issues warnings but still starts WLM, ignoring the missing files.

Dynamic CPU de-allocation

Dynamic CPU de-allocation is a RAS enhancement that monitors recoverable system processor errors and dynamically deconfigures an affected processor when a pre-defined threshold is breached. The way it works is that, when a processor's internal recoverable errors reach a predefined threshold, the firmware notifies AIX, which then migrates all software processes and interrupts to another processor and puts the faulty processor in 'stopped state'. The problematic CPU is also removed from the system configuration for all subsequent system boots until it is replaced.

The user can enable or disable the CPU de-allocation feature via AIX's system management services. Dynamic CPU de-allocation is implemented in the system firmware, service processor firmware, and AIX (4.3.3+APAR IY06844), and is supported by the following hardware:

- RS/6000 Enterprise Server S80
- RS/6000 SP 375 MHz POWER3 SMP thin and wide nodes
- RS/6000 44P Model 270.

PCI hot-plug

Hot-plug capabilities are supported by RS/6000 SP expansion I/O units for selected PCI adapters. This new functionality enables the maintenance of these adapters without powering off the SP expansion I/O unit or the associated SP POWER3 SMP high node, and it requires AIX version 4.3.3 with APAR IY06844 as well as update to the SP expansion unit's firmware, which can be downloaded from:

```
http://www.austin.ibm.com/support/micro/download.html
```

Alex Polak
System Engineer
APS (Israel)

AIX news

IBM has announced Tivoli NetView Version 6.0, the latest version of its popular network management product. NetView can be used to 'discover' TCP/IP networks, display network configuration, correlate and manage network events, monitor network health, and gather network performance data.

NetView ships with a platform-independent Java-based console that is claimed to be more scalable and robust than the previous Web-based client for NetView. The company has also improved NetView's 'Smartsets', which enable users to group resources by type, location, etc, by allowing the grouping of services by type, so that servers offering similar services, like HTTP and POP3 mail boxes, can be grouped for viewing and policy-based management.

NetView runs on the Tivoli Framework and is available on AIX, Solaris, Tru64 Unix, and Windows NT. It's out now, and details on pricing are available on request from IBM.

For further information, contact your local IBM representative.

* * *

Compaq, in association with Brocade, CNT, and Nortel, has launched SANworks, a suite of products for replicating SAN-based data between remote sites. Data Replication Manager (DRM) can replicate data over thousands of miles using ATM lines.

SANworks Storage Resource Manager, described as an 'open SAN management tool', provides automated reporting for storage capacity, consumption, and availability. The final component is a Web-based management tool for analysis of SAN-based data storage on AIX (and also Windows 2000/NT, Solaris, HP-UX, Compaq Tru64 Unix, and Red Hat Linux).

The products are available now and prices can be obtained on request from the vendor.

For further information contact:
Compaq Computer Corp, PO Box 692000, Houston, TX 77269, USA
Tel: +1 281 370 0670
Fax: +1 281 514 4583
Web: http://www.compaq.com

Compaq Computer, Hotham House, 1 Heron Square, Richmond, Surrey TW9 1EJ, UK
Tel: +44 181 332 3000
Fax: +44 181 332 1961

* * *

Microport has announced FieldSight 1.0, which monitors Unix systems-level functions and applications and notifies service personnel when preset thresholds are breached. FieldSight uses Microport's GHOST technology to assure accessibility and maintenance, even during a system panic or lock-up. The company claims that, in many instances, the software continues to work even when it's impossible for a user to log in and run programs normally.

FieldSight is out now; prices are available on request from the vendor.

For further information contact:
Microport Inc, 108 Whispering Pines Dr, Scotts Valley, CA 95066, USA
Tel: +1 831 438 8649
Fax: +1 831 438 7560
Web: http://www.microport.com