A

UNIX shell scripts

This appendix includes scripts helpful to manage disk devices and monitor I/O for servers attached to the ESS. Implementation of these scripts is described in 6.5, “AIX-specific I/O monitoring commands” on page 184.

Introduction

The scripts presented in this appendix were written and tested on AIX servers, but could be modified to work with SUN Solaris and HP-UX.

By downloading the Acrobat PDF version of this publication, you should be able to copy and paste these scripts for easy installation on your host systems. To function properly, the scripts presented here rely on:

- An AIX host running AIX 4.3.3ML10+ or AIX 5L™
- Subsystem Device Driver (SDD) for AIX Version 1.3.1.0 or later
- ESS Utility package Version 1.0.6 or later

The scripts presented in this appendix are:

- vgmap
- lvmap
- vpath_iostat
- ess_iostat
- test_disk_speeds

Attention: These scripts are provided on an ‘as is’ basis. They are not supported or maintained by IBM in any formal way. No warranty is given or implied, and you cannot obtain help with these scripts from IBM.

VGMAP

The **vgmap** script displays which vpaths a volume group uses and also which rank each vpath belongs to. Use this script to determine if a volume group is made up of vpaths on several different ranks and which vpaths to use for creating striped logical volumes.

An example output of the **vgmap** command is shown in Example A-1.

Example: A-1 VGMAP output

```
# vgmap testvg

PV_NAME         RANK    PV STATE  TOTAL PPs   FREE PPs
testvg:
vpath0          1100    active      502        502
vpath2          1000    active      502        502
```

```
#!/bin/ksh
##############################################
# VGMAP
# usage: vgmap <vgname>
#
# Displays ESS logical disks and RANK ids for each
# disk in the volume group
#
# Note: the script depends on correct lssdd info in
# /tmp/lssdd.out
#
# Before running the first time, run:
# lssdd > /tmp/lssdd.out
#
# Author: Pablo Clifton pablo.clifton@usa.net
# Date: Feb 28, 2003
##############################################

lssddfile=/tmp/lssdd.out
workfile=/tmp/work.$0
sortfile=/tmp/sort.$0

# AIX
lsvg -p $1 | grep -v "PV_NAME" > $workfile
echo "\nPV_NAME   RANK          PV STATE        TOTAL PPs    FREE PPs    Free D"

for i in `cat $workfile | grep vpath | awk '{print $1}'`
do
        #echo "$i ... rank"
        rank=`grep -w $i $lssddfile | awk '{print $12}' | head -n 1`
        sed "s/$i /$i    $rank/g" $workfile > $sortfile
        cp $sortfile $workfile
done

cat $workfile
rm $workfile
rm $sortfile
########################## THE END ######################
```

LVMAP

The **lvmap** script displays which vpaths and ranks a logical volume uses. Use this script to determine if a logical volume spans vpaths on several different ranks. The script does not tell you if a logical volume is striped or not. Use **lslv <lv_name>** for that information or modify this script.

An example output of the **lvmap** command is shown in Example A-2.

Example: A-2   LVMAP output

```
# lvmap stripedlv

LV_NAME   RANK           COPIES        IN BAND        DISTRIBUTION
stripedlv:
vpath2    1000           004:000:000   100%          000:004:000:000:000
vpath4    1300           004:000:000   100%          000:004:000:000:000
vpath10   1400           004:000:000   100%          000:004:000:000:000
```

```
#!/bin/ksh
##############################################
# LVMAP
# usage: lvmap <lvname>
#
# displays logical disk and rank ids for each
# disk a logical volume resides on

# Note: the script depends on correct lssdd info in
# /tmp/lssdd.out
#
# Before running the first time, run:
# lssdd > /tmp/lssdd.out
#
# Author: Pablo Clifton pablo.clifton@usa.net
# Date: Feb 28, 2003
##############################################

lssddfile=/tmp/lssdd.out
workfile=/tmp/work.$0
sortfile=/tmp/sort.$0

lslv -l $1 | grep -v " COPIES " > $workfile

for i in `cat $workfile | grep vpath | awk '{print $1}'`
do
        #echo "$i ... rank"
        rank=`grep -w $i $lssddfile | awk '{print $12}' | head -n 1`
        sed "s/$i /$i    $rank/g" $workfile > $sortfile
        cp $sortfile $workfile
done

echo "\nLV_NAME   RANK           COPIES        IN BAND        DISTRIBUTION"
cat $workfile

rm $workfile
rm $sortfile
########################## THE END ######################
```

VPATH_IOSTAT

The **vpath_iostat** script is a a wrapper program for AIX that converts iostat information based on hdisk devices to vpaths instead.

The **vpath_iostat** script depends on the **lssdd** command (included in the ESSUTIL package for AIX) and **iostat**. The script first builds a map file to list hdisk devices and their associated vpaths and then converts **iostat** information from hdisks to vpaths.

To run the script, make sure the ESS UTILITY **lssdd** and SDD are working properly—that is, all volume groups are using vpaths instead of hdisk devices and **lssdd** reports the correct information.

The command syntax is:

```
vpath_iostat (control c to break out)
```

or

```
vpath_iostat <interval> <iteration>
```

An example of the output **vpath_iostat** produces is shown in Example A-3.

Example: A-3 VPATH_IOSTAT output

```
garmo-aix: Total VPATHS used:        8    16:16 Wed 26 Feb 2003   5 sec interval
garmo-aix Vpath:          MBps            tps             KB/trans        MB_read         MB_wrtn
garmo-aix vpath0          12.698          63.0            201.5           0.0             63.5
garmo-aix vpath6          12.672          60.6            209.1           0.0             63.4
garmo-aix vpath14         11.238          59.8            187.9           0.0             56.2
garmo-aix vpath8          11.314          44.6            253.7           0.0             56.6
garmo-aix vpath2          6.963           44.2            157.5           0.0             34.8
garmo-aix vpath12         7.731           30.2            256.0           0.0             38.7
garmo-aix vpath4          3.840           29.4            130.6           0.0             19.2
garmo-aix vpath10         2.842           13.2            215.3           0.0             14.2
------------------------------------------------------------------------------------------
```

```
garmo-aix   TOTAL READ:   0.00    MB      TOTAL WRITTEN:  346.49  MB
garmo-aix   READ SPEED:   0.00    MB/sec   WRITE SPEED:   70.00   MB/sec
```

```
#!/bin/ksh
#####################################################################
# Usage:
#       vpath_iostat  (default: 5 second intervals, 1000 iterations)
#       vpath_iostat <interval> <count>
#
# Function:
#       Gather IOSTATS and report on ESS VPATHS instead of disk devices
#       AIX hdisks
#       HP-UX [under development ]
#       SUN   [under development ]
#       Linux [under development ]
#
# Note:
#       vpath_iostat depends on valid VPATH ids from the LSSDD command which
#       is part of the ESS Utilities
#
#       A small amount of free space < 1MB is required in /tmp
#
# Author:       Pablo Clifton   pablo.clifton@usa.net
# Date: Feb 28, 2003
#####################################################################

##########################################################
# set the default period for number of seconds to collect
# iostat data before calculating average
period=5
iterations=1000

essfile=/tmp/disk-vpath.out             # File to store output from lssdd command
ifile=/tmp/lssdd.out                    # Input file containing LSSDD info

ds=`date +%d%H%M%S`             # time stamp
hname=`hostname`                # get Hostname
ofile=/tmp/vstats               # raw iostats
wfile=/tmp/wvfile               # work file
wfile2=/tmp/wvfile2             # work file
pvcount=`iostat | grep hdisk | wc -l | awk '{print $1}'`

#############################################
# Create a list of the vpaths this system uses
# Format:       hdisk           ESS-vpath
# LSSDD output MUST BE correct or the IO stats reported
# will not be correct
#############################################
if [ ! -f $ifile ]
then
  echo "Collecting LSSDD info for disk to vpath map..."
  lssdd > $ifile
fi

cat $ifile  | awk '{print $4 "\t" $3}' > $essfile

#########################################
# ADD INTERNAL SCSI DISKS to RANKS list
#########################################
for internal in `lsdev -Cc disk | grep SCSI | awk '{print $1}'`
```

```
do
        echo "$internal $internal" >> $essfile
done

############################################
# Set interval value or leave as default
if [[ $# -ge 1 ]]
then
        period=$1
fi

#######################################
# Set <iteration> value
if [[ $# -eq 2 ]]
then
        iterations=$2
fi

#################################################################
#  ess_iostat <interval> <count>

i=0
while [[ $i -lt $iterations ]]
do
        iostat $period 2  > $ofile              # run 2 iterations of iostat
                                                # first run is IO history since boot

        grep hdisk $ofile > $ofile.temp # only gather hdisk info- not cd
                                                        # other devices

        tail -n $pvcount $ofile.temp | grep -v "0.0        0.0       0.0          0
0"  | sort +4 -n -r | head -n 100 > $wfil
e

        ###########################################
        #Converting hdisks to vpaths....           #
        ###########################################
        for j in `cat $wfile  | awk '{print $1}'`
        do
           vpath=`grep -w $j $essfile | awk '{print $2}'`
           sed "s/$j /$vpath/g" $wfile > $wfile2
           cp $wfile2 $wfile
        done


        ###########################################
        # Determine Number of different VPATHS used
        ###########################################
        numvpaths=`cat $wfile | awk '{print $1} ' | grep -v hdisk | sort -u | wc -l`

        dt=`date +"%H:%M %a %d %h %Y"`

        print "\n$hname: Total VPATHS used: $numvpaths  $dt     $period sec interval"
        printf "%s\t%s\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n"  "$hname" "Vpath:" "MBps" "tps"
"KB/trans" "MB_read" "MB_wrtn"


        ###########################################
        # Sum Usage for EACH VPATH and Internal Hdisk
        ###########################################
        for x in `cat $wfile |  awk '{ print $1}' | sort -u`
        do
        cat $wfile | grep -w $x | awk '{ printf ("%4d\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n" ,
$1, $2, $3, $4, $5, $6) }' | awk 'BEGIN {
 }
                { tmsum=tmsum+$2 }
                { kbpsum=kbpsum+$3 }
                { tpsum=tpsum+$4 }
                { kbreadsum=kbreadsum+$5 }
                { kwrtnsum=kwrtnsum+$6 }

        END {
                if ( tpsum > 0 )
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
vpath, kbpsum/1000, tpsum, kbpsum/tpsum , k
breadsum/1000, kwrtnsum/1000)
                else
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
vpath, kbpsum/1000, tpsum, "0", kbreadsum/1
000, kwrtnsum/1000)
             }' hname="$hname" vpath="$x" >> $wfile2.tmp

        done


        #############################################
        # Sort VPATHS/hdisks by NUMBER of TRANSACTIONS
        #############################################
        if [[ -f $wfile2.tmp ]]
        then
                cat $wfile2.tmp | sort +3 -n -r
                rm $wfile2.tmp
        fi


        ##############################################################
        # SUM TOTAL IO USAGE for ALL DISKS/LUNS over INTERVAL
        ##############################################################
        #Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
        # field 5 read field 6 written
        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | awk  'BEGIN { }
                { rsum=rsum+$5 }
                { wsum=wsum+$6 }

                END {
                rsum=rsum/1000
                wsum=wsum/1000

                printf
("-----------------------------------------------------------------------------------------
-\n")

                if ( divider > 1 )
                {
                printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n", hname, "TOTAL READ: ",
rsum, "MB", "TOTAL WRITTEN: ", wsum, "MB"
 ) }
```

```
                printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n\n\n", hname, "READ SPEED:
", rsum/divider, "MB/sec", "WRITE SPEED:
 ", wsum/divider, "MB/sec" )
                 }'  hname="$hname"  divider="$period"

        let i=$i+1

done

rm $ofile
rm $wfile
rm $wfile2
rm $essfile

################## THE END ##################
```

ESS_IOSTAT

The **ess_iostat** script is a a wrapper program for AIX that converts **iostat** information based on hdisk devices to ranks instead.

The **ess_iostat** script depends on the **lsess** command (included in the ESSUTIL package for AIX) and **iostat**. The script first builds a map file to list hdisk devices and their associated ranks and then converts **iostat** information from hdisks to ranks.

To run the script, make sure the ESS UTILITY **lsess** is working properly and enter:

```
ess_iostat (control c to break out)
```

or

```
ess_iostat <interval> <iteration>
```

An example of the **ess_iostat** output is shown in Example A-4.

Example: A-4 ESS_IOSTAT output

```
# ess_iostat 5 1

garmo-aix: Total RANKS used:       12      20:01 Sun 16 Feb 2003   5 sec interval
garmo-aix  Ranks:       MBps             tps           KB/trans       MB_read        MB_wrtn
garmo-aix  1403         9.552           71.2            134.2           47.8            0.0
garmo-aix  1603         6.779           53.8            126.0           34.0            0.0
garmo-aix  1703         5.743           43.0            133.6           28.8            0.0
garmo-aix  1503         5.809           42.8            135.7           29.1            0.0
garmo-aix  1301         3.665           32.4            113.1           18.4            0.0
garmo-aix  1601         3.206           27.2            117.9           16.1            0.0
garmo-aix  1201         2.734           22.8            119.9           13.7            0.0
garmo-aix  1101         2.479           22.0            112.7           12.4            0.0
garmo-aix  1401         2.299           20.4            112.7           11.5            0.0
garmo-aix  1501         2.180           19.8            110.1           10.9            0.0
garmo-aix  1001         2.246           19.4            115.8           11.3            0.0
garmo-aix  1701         2.088           18.8            111.1           10.5            0.0
------------------------------------------------------------------------------------------
garmo-aix    TOTAL READ:   430.88  MB        TOTAL WRITTEN:  0.06    MB
garmo-aix    READ SPEED:    86.18 MB/sec      WRITE SPEED:    0.01   MB/sec
```

```
#!/bin/ksh
```

```
#set -x
#####################################################################
# Usage:
#       ess_iostat  (default: 5 second intervals, 1000 iterations)
#       ess_iostat <interval> <count>
#
# Function:
#       Gather IOSTATS and report on ESS RANKS instead of disk devices
#       AIX hdisks
#       HP-UX
#       SUN
#       Linux
#
# Note:
#       ess_iostat depends on valid rank ids from the LSESS command which
#       is part of the ESS Utilities
#
#       A small amount of free space < 1MB is required in /tmp
#
# Author:       Pablo Clifton   pablo.clifton@usa.net
# Date: Feb 28, 2003
#####################################################################

##########################################################
# set the default period for number of seconds to collect
# iostat data before calculating average
period=5
iterations=1000

essfile=/tmp/lsess.out          # File to store output from lsess command

ds=`date +%d%H%M%S`             # time stamp
hname=`hostname`                        # get Hostname
ofile=/tmp/rstats                       # raw iostats
wfile=/tmp/wfile                        # work file
wfile2=/tmp/wfile2              # work file
pvcount=`iostat | grep hdisk | wc -l | awk '{print $1}'`

#############################################
# Create a list of the ranks this system uses
# Format:       hdisk           ESS-rank
# LSESS output MUST BE correct or the IO stats reported
# will not be correct
#############################################
lsess | awk '{print $1 "\t" $9}' > $essfile


#########################################
# ADD INTERNAL SCSI DISKS to RANKS list
#########################################
for internal in `lsdev -Cc disk | grep SCSI | awk '{print $1}'`
do
        echo "$internal $internal" >> $essfile
done


###############################################
# Set interval value or leave as default
if [[ $# -ge 1 ]]
then
```

```
        period=$1
fi


##########################################
# Set <iteration> value
if [[ $# -eq 2 ]]
then
        iterations=$2
fi


#################################################################
#  ess_iostat <interval> <count>

i=0
while [[ $i -lt $iterations ]]
do
        iostat $period 2  > $ofile              # run 2 iterations of iostat
                                                        # first run is IO history since
boot

        grep hdisk $ofile > $ofile.temp # only gather hdisk info- not cd
                                                        # other devices

        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | sort +4 -n -r | head -n 100 > $wfil
e

        ###########################################
        #Converting hdisks to ranks....           #
        ###########################################
        for j in `cat $wfile  | awk '{print $1}'`
        do
           rank=`grep -w $j $essfile | awk '{print $2}'`
           sed "s/$j /$rank/g" $wfile > $wfile2
           cp $wfile2 $wfile
        done


        ###########################################
        # Determine Number of different ranks used
        ###########################################
        numranks=`cat $wfile | awk '{print $1} ' | grep -v hdisk | cut -c 1-4| sort -u -n |
wc -l`
        dt=`date +"%H:%M %a %d %h %Y"`

        print "\n$hname: Total RANKS used: $numranks    $dt     $period sec interval"
        printf "%s\t%s\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n"  "$hname" "Ranks:" "MBps" "tps"
"KB/trans" "MB_read" "MB_wrtn"


        ###########################################
        # Sum Usage for EACH RANK and Internal Hdisk
        ###########################################
        for x in `cat $wfile |  awk '{ print $1}' | sort -u`
        do
        cat $wfile | grep -w $x | awk '{ printf ("%4d\t\t%-9s\t%-9s\t%-9s\t%-9s\t%-9s\n" ,
$1, $2, $3, $4, $5, $6) }' | awk 'BEGIN {
 }
```

```
                { tmsum=tmsum+$2 }
                { kbpsum=kbpsum+$3 }
                { tpsum=tpsum+$4 }
                { kbreadsum=kbreadsum+$5 }
                { kwrtnsum=kwrtnsum+$6 }

        END {
                if ( tpsum > 0 )
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
rank, kbpsum/1000, tpsum, kbpsum/tpsum , kb
readsum/1000, kwrtnsum/1000)
                else
                printf ("%-7s\t%4s\t\t%-9.3f\t%-9.1f\t%-9.1f\t%-9.1f\t%-9.1f\n" , hname,
rank, kbpsum/1000, tpsum, "0", kbreadsum/10
00, kwrtnsum/1000)
             }' hname="$hname" rank="$x" >> $wfile2.tmp

        done


        #############################################
        # Sort RANKS/hdisks by NUMBER of TRANSACTIONS
        #############################################
        if [[ -f $wfile2.tmp ]]
        then
                cat $wfile2.tmp | sort +3 -n -r
                rm $wfile2.tmp
        fi


        ##############################################################
        # SUM TOTAL IO USAGE for ALL DISKS/LUNS over INTERVAL
        ##############################################################
        #Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
        # field 5 read field 6 written
        tail -n $pvcount $ofile.temp | grep -v "0.0       0.0       0.0          0
0"  | awk  'BEGIN { }
                { rsum=rsum+$5 }
                { wsum=wsum+$6 }

                END {
                rsum=rsum/1000
                wsum=wsum/1000

                printf
("------------------------------------------------------------------------------------------
-\n")

                if ( divider > 1 )
                {
                printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n", hname, "TOTAL READ: ",
rsum, "MB", "TOTAL WRITTEN: ", wsum, "MB"
 ) }

               printf ("%-7s\t%14s\t%4.2f\t%s\t%14s\t%4.2f\t%s\n\n\n", hname, "READ SPEED:
", rsum/divider, "MB/sec", "WRITE SPEED:
 ", wsum/divider, "MB/sec" )
                 }'  hname="$hname"  divider="$period"

        let i=$i+1
```

```
done

rm $ofile
rm $wfile
rm $wfile2
rm $essfile
################################## THE END ##########################
```

TEST_DISK_SPEEDS

Use the **test_disk_speeds** script to test a vpath and record the speed at different times throughout the day to get an *average* read speed a rank is capable of in your environment. Use **lsess** or **lssdd** to determine which ranks the vpaths reside on.

You can change the amount of data read, the block size, and the vpath by editing the script and changing the variables:

```
tsize=100 # MB
bs=128 # KB
vpath=vpath0 # disk to test
```

An example of the output for **test_disk_speeds** is shown in Example A-5.

Example: A-5   TEST_DISK_SPEEDS example

```
# test_disk_speeds
vpath0 43.0 MB/sec  100 MB      bs=128k
```

```
#!/bin/ksh
##########################################################
# test_disk_speeds
# Measure disk speeds using dd
#
# tsize = total test size in MB
# bs    = block size in KB
# testsize= total test size in KB; tsize*1000

# count = equal to the number of test blocks to read which is
#         testsize/bsize
# Author: Pablo Clifton pablo.clifton@usa.net
# Date: February 28, 2003
#########################################################
# SET these 2 variables to change the block size and total
# amount of data read. Set the vpath to test
tsize=100 # MB
bs=128 # KB
vpath=vpath0 # disk to test
#########################################################
let testsize=$tsize*1000
let count=$testsize/$bs

# calculate start time, dd file, calculate end time
stime=`perl -e "print time"`
dd if=/dev/$vpath of=/dev/null bs="$bs"k count=$count
etime=`perl -e "print time"`
```

```
# get total run time in seconds
let totalt=$etime-$stime
let speed=$tsize/$totalt

printf "$vpath\t%4.1f\tMB/sec\t$tsize\tMB\tbs="$bs"k\n" $speed
########################## THE END ###############################
```