

Getting the most from your free system activity reporter: Sar

Jeff Kubler

Kubler Consulting

jrkubler@proaxis.com

www.proaxis.com/~jrkubler





SAR - Introduction

- Sar - Stands for System Activity Reporter.
- Originated with system V Unix
- Retrieves data via /dev/kmem or pstat(2) interface
- logs data in background using /usr/lib/sa/sadc program



Sar - Introduction

- Logged data kept in binary format disk file
- Useful for long term analysis and baseline pictures
- sar can be used for interactive and historical collection of data.



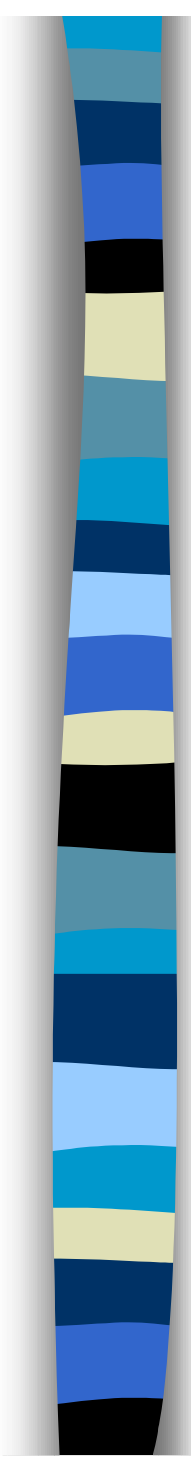
Sar - Introduction

- Many have written scripts to collect, manage and output the data
- At least one 3rd party tool exists to analyze sar data



Sar - option description

- `sar [-ubdycwaqvmAMS] [-o file] t [n]`
 - This form records data or outputs
- `sar [-ubdycwaqvmAMS] [-s time] [-e time] [-i sec] [-f file]`
 - This form reads from a recorded file



Sar (- u CPU, -b buffer, -d device, etc.)

```
# sar -u 1 10
```

```
HP-UX pointman B.10.20 E 9000/831 03/01/99
```

21:09:44	%usr	%sys	%wio	%idle
21:09:45	1	0	0	99
21:09:46	0	0	0	100
21:09:47	0	0	0	100
21:09:48	18	0	0	82
21:09:49	0	3	0	97
21:09:50	0	0	0	100
21:09:51	0	0	1	99
21:09:52	0	0	0	100
21:09:53	0	0	0	100
21:09:54	0	4	0	96
Average	2	1	0	97



Sar

- -v Report status of text, process, inode and file tables:
- -q Report average queue length while occupied, and percent of time occupied.
- -w Report system swapping and switching activity:



sar

- -c Report system calls:
- -y Report tty device activity:
- -d Report activity for each block device, e.g., disk or tape drive.
- -b Report buffer activity:



sar

- -u Report CPU utilization (the default)
- -m Report message and semaphore activities:
- -A Report all data. Equivalent to -udqbwcaym.
- -o followed by file name sends data to file.



Sar command structure

- Simplest format - sar -option interval time period
 - sar -u 60 60
 - every 60 seconds for 60 minutes
- send info to file
 - sar -u -o outputfile 20 5
 - sends output to file and screen
 - file has everything regardless of option



Sar command structure

- Sar format to read recorded data:
 - `sar -f filename`
 - This prints the -u default information from the file directed by -f
 - `sar -A -f filename`
 - This prints everything if the collection was made with -A



Sar setup

- Needed programs sar and sadc
 - /usr/lib/sa/sadc -this collects in background
 - /usr/bin/sar - performs interactive
 - daily system activity goes to /usr/adm/sa/saxx - xx is the date of the month



Executing sar in the background

```
# more sa1
#! /usr/bin/sh
# @(#) $Revision: 72.3 $
#      sa1.sh

DATE=`date +%d`
ENDIR=/usr/libin/sa
DFILE=/var/adm/sa/sa$DATE
cd $ENDIR
if [ $# = 0 ]
then
    exec $ENDIR/sadc 1 1 $DFILE
else
    exec $ENDIR/sadc $* $DFILE
fi
```



Executing sar in the background

```
# more sa2
#! /usr/bin/sh
# @(#) $Revision: 72.1 $
#      sa2.sh

DATE=`date +%d`
RPT=/var/adm/sa/sar$DATE
DFILE=/var/adm/sa/sa$DATE
ENDIR=/usr/sbin
cd $ENDIR
$ENDIR/sar $* -f $DFILE > $RPT
find /var/adm/sa \( -name 'sar*' -o -name 'sa*' \) -
mtime +7 -exec rm {} \;
#
```



Sar simple output

HP-UX fred B. 10.20 A 9000/712 02/09/100

```
23: 12: 23      %usr      %sys      %wi o      %i dl e
                devi ce      %busy      avque      r+w/s      bl ks/s      awwai t      avserv
runq-sz %runocc swpq-sz %swpocc
bread/s l read/s %rcache bwrit/s l writ/s %wcache pread/s pwrit/s
swpin/s bswin/s swpot/s bswot/s pswch/s
scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
iget/s namei /s di rbk/s
rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
text-sz ov proc-sz ov inod-sz ov file-sz ov
msg/s sema/s
23: 12: 53      0          0          0          100
                0.0          0          0.0          0
                0          4          100          0          0          100          0          0
                1.00          0.0          1.47          0.0          20
                34          1          0          0.00          0.00          21982          0
                0          0          0
                0          0          19          0          0          0
N/A      N/A      84/276      0      475/476      0      251/800      0
                0.00          0.00
```



Setting up sar collection

- Use cron for collection
 - #collect sar data
 - 0 * * * * /usr/lbin/sa/sa1
 - 20,40 8-17 * * 1-5 /usr/lbin/sa/sa1
- This collects data once an hour at non-peak times every 20 minutes during the busy times.



Setting up sar data reduction

■ Use cron to reduce the data

– #reduce the sar data

- 5 18 * * * /usr/lbin/sa/sa2 -s 8:00 -e 18:01 -l 900
-A {one line}

■ HP-UX 11.0 users may need to reduce

– #reduce the sar data

- 5 18 * * * /usr/lbin/sa/sa2 -A



Using sar data

- Data can be exported to an excel spreadsheet for graphing.
 - Use script to add “”s and ,’s to the file.
- SARCHECK will examine the data and make conclusions based on the measurements.



Rules of Thumb

- Commonly held theorems about levels of acceptable usage.
 - CPU - total busy above 85 % is not good
 - CPU Queue Length of 5 is going to begin to show poor performance, 15 and above is very bad.
 - Memory - buffer cache read hit rate and write hit of less than 90 % is not good.



Rules of Thumb (continued)

- Capture Ratio (this is a ratio of the user processing / by system activity or (User + Real + Nice)/(System + Interrupt + Context Switch) = Capture ratio)). Should be greater than 3 and will definitely reflect a problem when 1.0.
- Real processing, System, Interrupt, Context Switching should not exceed 10 % as individual measures.



Rules of Thumb (continued)

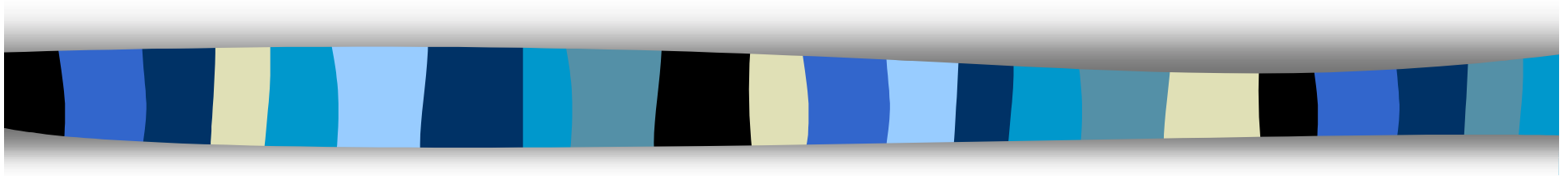
- Memory % used should not exceed 80 to 90 %.
- Virtual % used should not exceed 50 to 80 %.
- Disk I/O Queue length should not exceed 1.0.
- Total reads & writes per drive should not exceed 50 to 60.



Rules of Thumb (continued)

- Page out rate of 10 per second is not good and an indicator of memory shortage.
- Deactivation rate of 5 processes is an indicator of memory issues.

The End



Thanks for coming!