Solaris Performance monitoring - sar

System Activity Report Package

`sar` is part of the system activity reporter package that is included in the Solaris operating environment. This package consists of three commands that are involved in automatic system activity data collection: `sadc`, `sa1`, and `sa2`.

This facility stores a great deal of performance data about a system. This information is invaluable when attempting to identify the source of a performance problem.

The Report Package can be enabled by uncommenting the appropriate lines in the sys crontab. The `sa1` program stores performance data in the `/var/adm/sa` directory. `sa2` writes reports from this data, and `sadc` is a more general version of `sa1`.

In general, I have found the sa2 reports are not as useful in most debugging. Most of the time, the `sa1` run at intervals provides the information you require.

Setting up the `sa1` to extraction and store data is setup in the `sys` accounts crontab.

Alternatively, sar can be used on the command line to look at performance over different time slices or over a constricted period of time:

```
sar -A -o outfile 5 2000
```

(Here, "5" represents the time slice and "2000" represents the number of samples to be taken. "outfile" is the output file where the data will be stored.)

The data from this file can be read by using the "-f" option. `-f filename`: Uses filename as the source for the binary `sar` data. The default is to use today's file from `/var/adm/sa`.

Setting up sar

Configure performance monitoring at startup

Modify the `perf` startup script to initialize the performance counters then run the init script.

```
# id
uid=0(root) gid=1(other)
$ vi /etc/init.d/perf
```

Uncomment the entire if statement sections listed below in the `perf` startup script. This script initializes (zero's out) the counters during startup.

```
if [ -z "$_INIT_RUN_LEVEL" ]; then
      set -- `/usr/bin/who -r`
      _INIT_RUN_LEVEL="$7"
      _INIT_RUN_NPREV="$8"
      _INIT_PREV_LEVEL="$9"
fi

if [ $_INIT_RUN_LEVEL -ge 2 -a $_INIT_RUN_LEVEL -le 4 -a \
    $_INIT_RUN_NPREV -eq 0 -a \( $_INIT_PREV_LEVEL = 1 -o \
    $_INIT_PREV_LEVEL = S \) ]; then

       /usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
fi
:wq
# /etc/init.d/perf start
#
```

Configuring sys user crontab

Utilize the `crontab` command to modify the crontab configuration file for the `sys` user.

```
# id
uid=0(root) gid=1(other)
# crontab -l sys

#ident  "@(#)sys        1.5     92/07/14 SMI"   /* SVr4.0 1.2   */
#
# The sys crontab should be used to do performance collection. See cron
# and performance manual pages for details on startup.
#
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A

# crontab -e sys
(uncomment the first two entries then save the updates)
:wq

# crontab -l sys

#ident  "@(#)sys        1.5     92/07/14 SMI"   /* SVr4.0 1.2   */
#
# The sys crontab should be used to do performance collection. See cron
# and performance manual pages for details on startup.
#
0 * * * 0-6 /usr/lib/sa/sa1
```

```
15,30,45 7-18 * * 1-5 /usr/lib/sa/sa1
5 21 * * 1-5 /usr/lib/sa/sa2 -s 7:00 -e 18:01 -i 1200 -A
#
```

The first entry writes a record to `/var/adm/sa/sa<dd>` on the hour, every hour, seven days a week.

The second entry writes a record to `/var/adm/sa/sa<dd>` three time each hour during peak working hours: at 15 minutes, 30 minutes and 45 minutes past the hour, from 7am to 6pm, Monday through Friday.

Thus, these two crontab entries cause a record to be written to `/var/adm/sa/sa<dd>` every 15 minutes from 7am to 6pm, Monday through Friday, and every hour otherwise. Therefore, your records will contain 15 minute entries during business hours and hourly entries for after-hours and weekends.

The third entry would have run the shell script named `sa2` at 9:05 pm, Monday through Friday. The sa2 script writes reports from the binary data.

The shell script `sa2`, a variant of `sar`, writes a daily report in the file `/var/adm/sa/sar<dd>`. The report will summarize hourly activities for the day from 7:00 am to 6:01 pm.

Running sar

The syntax for the sar command is as follows:

```
sar [-aAbcdgkmpqruvwy] [-o <outputfile>] [t n ]
```

Options table for the sar command

Option	Action
a	Checks file access operations
b	Checks buffer activity
c	Checks system calls
d	Checks activity for each block device
g	Checks page-out and memory freeing
k	Checks kernel memory allocation
m	Checks interprocess communication
p	Checks swap and dispatch activity
q	Checks queue activity
r	Checks unused memory
u	Checks CPU utilization
nv	Checks system table status
w	Checks swapping and switching volume
y	Checks terminal activity
A	Reports overall system performance (same as entering all options)

```
# sar -u

SunOS ducttape 5.8 Generic_117350-25 sun4u    09/22/05

13:20:00    %usr    %sys    %wio   %idle
13:32:31        unix restarts
13:40:00       0       1       0      99
```

```
# sar -k

SunOS ducttape 5.8 Generic_117350-25 sun4u    09/22/05

13:20:00 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
13:32:31        unix restarts
13:40:00 6643776 3701631     0 64348160 55235912     0     5070848     0
```