



<i>Launching kSar</i>	2
<i>Grabbing statistics</i>	5
<i>Managing graphs</i>	6
<i>Exporting graphs to PDF</i>	7
<i>Using trigger</i>	7
<i>Tips</i>	8
<i>Understanding Solaris graphs</i>	9
<i>Understanding AIX graphs</i>	11
<i>Understanding Linux graphs</i>	12
<i>Understanding Mac graphs</i>	13
<i>Enabling sar on your host</i>	14
<i>ANNEXES</i>	16
<i>ACKNOWLEDGEMENTS</i>	16

1. Launching kSar

1. Windows & Mac os X:

The simplest way is too double-click on the kSar-x.x.x.jar

2. Command line (unix & other):

You need a functional java installation and launch : `java -jar kSar-x.x.x.jar`

You can pass argument to command line to load a text file, to export a pdf and enable triggers. If you want to know what argument can be passed use -help options. eg.: `java -jar kSar-x.x.x.jar -help`

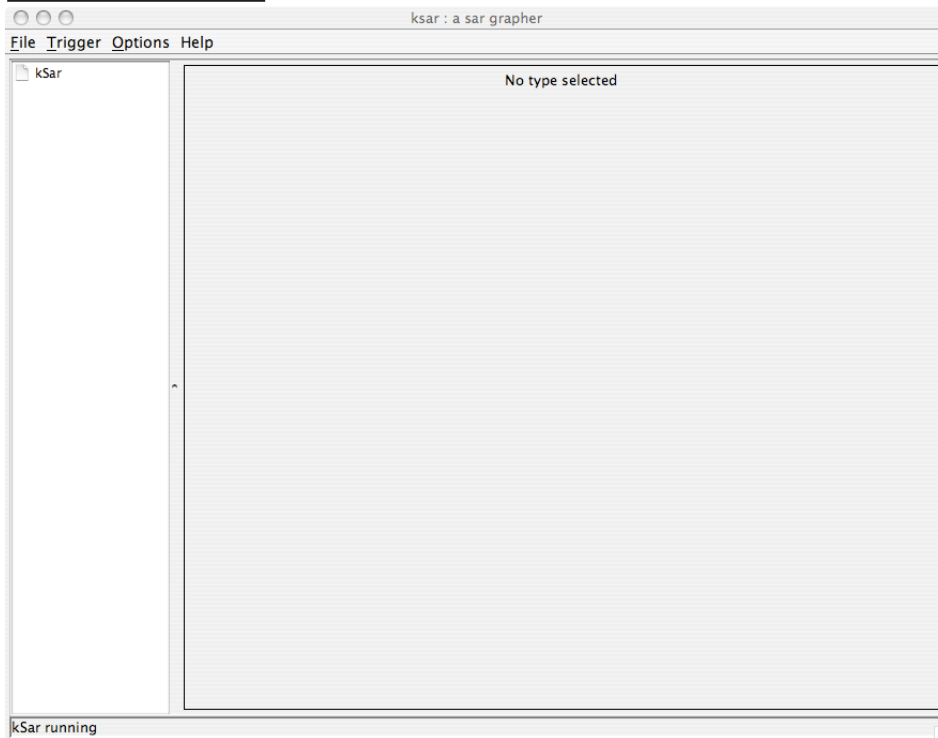
Argument that works either with the GUI or the command line:

- -input <sar output file> : parse the file specified.
- -showTrigger : will show trigger on graph (will toggle the menu)
- -noEmptydisk : will not export disk with no data (will toggle the menu)

Argument that works with the command line only (-input is mandatory or use those arguments) :

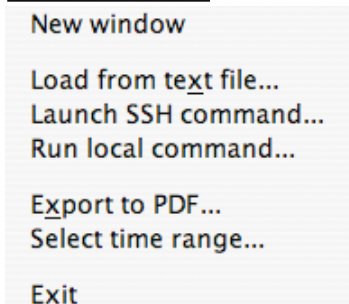
- -userPrefs : will use the userPrefs for outputting the pdf file
- -output <pdf file> : output the pdf report to file (backward compatibility)
- -outputPDF <pdf file> : output the pdf report to the pdf file
- -graph <String> : space separated list of graph you want to be output
- -outputPNG <base filename> : output the graphs to PNG file using argument as base filename
- -outputJPG <base filename> : out the graphs to JPG file using argument as base filename
- -width <size> : make JPG/PNG with specified width size (default: 800)
- -height <size> : make JPG/PNG with specified height size (default: 600)
- -addHTML : will create an html page with PNG/JPG image
- -noEmptydisk : will not export disk with no data

3. The main window:



4. The main window has two panels. The left one will have a list of graphs available depending on the data kSar has parsed. The right window will show you the graph you have selected.

5. The file menu:



New window : a new window to graph a different host/time will show up

Load from text file : see chapter 2 section 1

Launch SSH command : see chapter 2 section 2

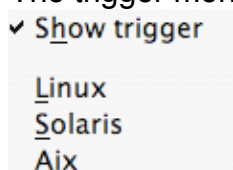
Run local command : see chapter 2 section 3

Export to PDF : see chapter 4

Select time range : see chapter 3 section 2

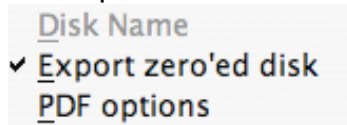
Exit: i am not sure what this menu is doing.

6. The trigger menu:



Show trigger : is a toggle menu, to enable/disable the trigger
Linux/Solaris/AIX : are menus to define trigger value see chapter 5

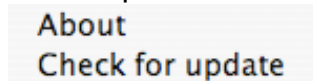
7. The Options menu:



The Disk name submenu will permit you to map awful device name to any chosen text. For example if you device is sd3,a ; you can tell kSar that it should use the comment you have put to name it.

The Export zero'ed disk is a toggle option that will print or not disks which have not data in it. You can toggle this options also via command line

8. The Help menu:



About will show you some information about version.

The check for update sub-menu will connect to kSar web server to see if there a new version available.

2. Grabbing statistics

1. Using a local text file of a sar output:

This will parse a output text file of the sar command. you need to have a special precaution to make this file, since the parser is not locale time aware. The parser is waiting for C locale output file

eg. : `LC_ALL=C sar -A -f /var/adm/sa/sa22 > mysar22.txt`

this command will use the default locale to output all statistics from sar using the file put after the -f argument to a file called mysar22.txt

After generating the file, you can use the file chooser dialog to load it into kSar.

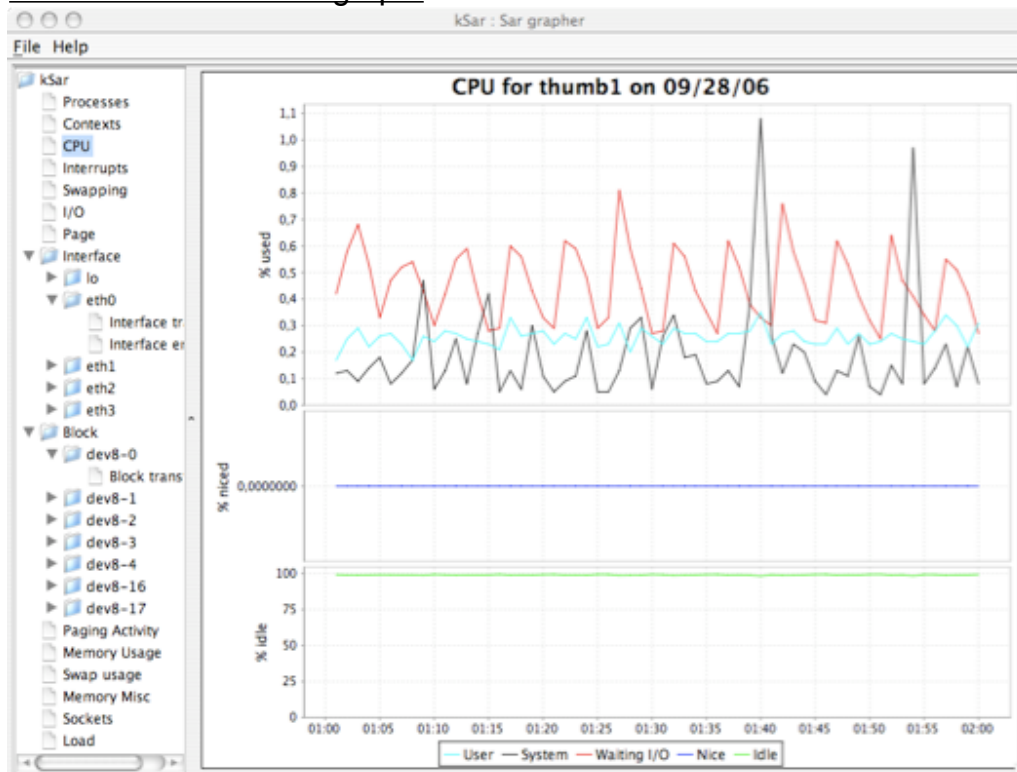
2. Using ssh to connect to a remote host:

If you provide valid username, remote hostname and password, you will be able to execute a remote command. The result of the command will be parse by kSar to make graphs. The environment variable `LC_ALL=C` is already exported for the first command but if you need to specify more than one command be sure that output is locale independent.

3. Using a local command:

After typing your command and hitting the OK button. kSar will execute the command using a pre-defined locale (`LC_ALL=C`) on the host. Again be sure that the output of the command is locale independent if you have more than one command in a row.

4. The main window with graph:

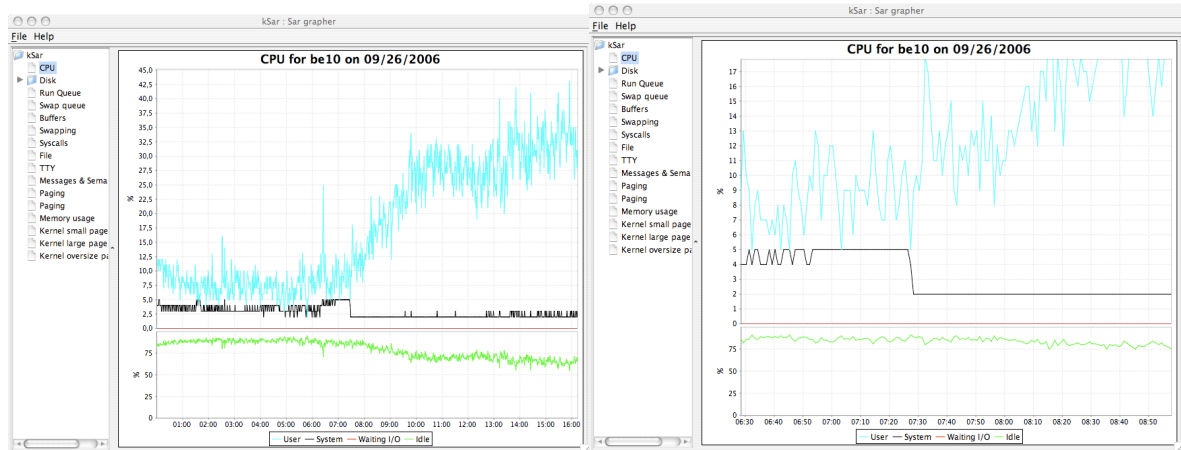


As you can see, the graph type tree is deployed here and a graph has been selected (this is not the case when kSar finish the parsing, tree is undeployed and no graph is shown/selected)

3. Managing graphs

1. Interactive zooming:

Using the move, you can interactively zoom onto up a part of a graph. To select a zone to zoom, click on the upper left corner and while still holding the mouse button move to the lower-right of the zone you want to zoom. To come back to un-zoomed view click and drag the mouse to any corner location except a lower-right one

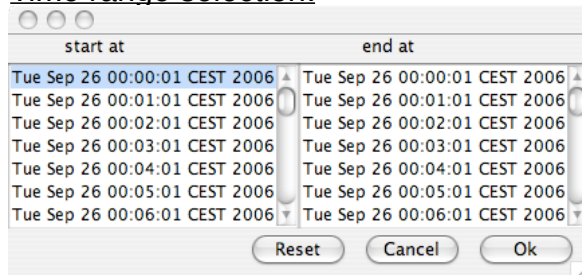


Unzoomed

Zoomed

The zooming selection will disappear if you select another graph.

2. Time range selection:

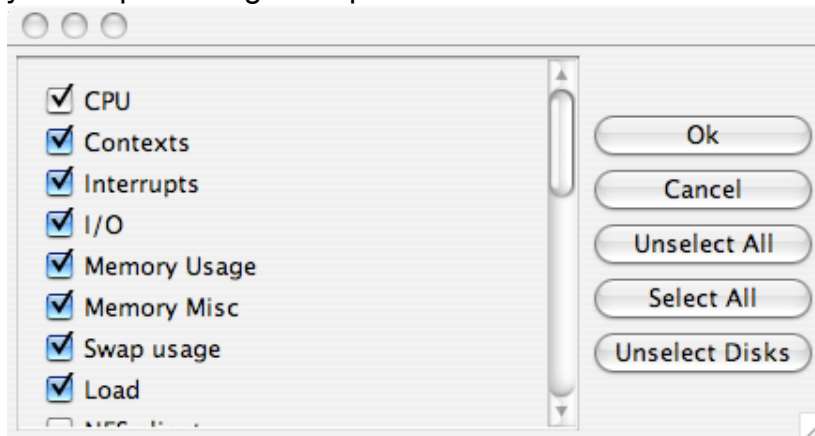


With this function you can select part of the data you have parsed into kSar. The time selection will be for all graphs and apply to PDF output also. This can be useful if you have parsed the all day but want just a specify time range to be displayed/printed.

4. Exporting graphs to PDF

1. Export graphs to PDF:

After choosing the file location and name, a box will show up to let you select the graphs you want to include in the PDF report. You have short-cut button to unselect all or just unselect the disks. After hitting the OK button a progress bar will notify you the percentage completed.



5. Using trigger

1. Understanding trigger:

When some value of a graph go over or under some value, you can get background greyed to notify the period when it happens. it permits to make finding trouble more easily. Those triggers are shown on the GUI and the PDF exported. You can toggle this feature by using the menu Trigger 'Show trigger'.

6. Tips

1. Live statistics:

Since kSar is waiting for the end of the command. It is possible to ask sar to output its statistics at a specified interval during a number of times. Doing this, kSar is able to graph live value. Be aware that if you don't specify the number of time, sar might stay running on your host until it get killed by someone. You should also noted that due to sar Solaris not showing at every interval the sar header, live statistics on Solaris is only available one counter at a time. You need to make a click and drag in the graph as if you would disable some zooming. The live functionality has been almost tested but some error might raised.

eg. linux: `sar -A 5 10` (will graph live value for 10 times with an interval of 5 seconds

eg.solaire: `sar -u 12 7` (CPU usage for 7 times every 12 seconds)

7. Understanding Solaris graphs

1. CPU:

The panel of the cpu graphs is divided into two parts. The upper part is how the cpu has been used by the host. The lower part of the graph is the cpu time left idle that is available to process instruction.

If your CPU idle graph is at 0 for a long period of time, this probably means that your host needs to have more cpu power, but you should have a look at the time wasted waiting I/O, and time spent in system. If you are running a nfs server and your system time is very high, maybe you could have a look at your nfs parameter and try to tune some parameters to get things running better. If you got many waiting I/O you should have a look at the disks performance to see if your box is really wasting time on disk utilisation. Warning on a multi-processor platform some waiting I/O cpu cycle can be assumed as idle time. Furthermore Waiting I/O has been disabled on Solaris 10+

2. Disk (Disk Xfer/Disk Wait):

Disk have two panels. "Disk Xfer" are graphs about data transferred from/to the disk. "Disk Wait" are statistics about queue and waiting I/O statistics.

There are three "disk xfer" graphs. bytes/s which are how many bytes are read/write to the disk. read+write/s which is the number of write and read command issued (one command can span many blocks). And finally avserv is the average time spend to handle one request (including seek, rotational latency, and data transfer).

On the "Disk Wait" panel, there are three graphs : avque,avwait,%busy. avque is the average number of queries in the queue waiting the disk to be processed. avwait is how long a request stay idle in the queue, the more it stay means that the disk have trouble to empty the queue. And %busy is the percentage of the disk utilisation. if you hit for a long period of time 100% getting a faster disk and/or changing data stripping might help you.

3. Run Queue:

The "Run Queue" panel has two graphs. the runq-sz (run queue size) graph is the number of process/thread that are ready to run. the runqocc is the percentage of time that run queue has at least one process. if you run queue has more that 2 process/thread by processor then maybe you need to add cpu because process are waiting for idle cpu.

4. Swap Queue:

The Swap Queue is pretty always at zero, if you got some value, then you got a problem or you have had a problem. In typical situation the swap queue is only used when there is a big memory exhaustion The swpq-sz (swap queue size) is the number of process the system had to swap to the disk for freeing some memory. To find out which process has been swapped to the disk, you can search for process where rss size is 0 either with prstat,ps or top.

5. Buffers:

The panel "Buffers" got four graphs. The first two graph from the top are read from the system buffers, raw disk read and disk read. The second one is the same as the previous for writing operations. The third graphs is the buffer write cache, the value from this graph are not very useful. And the last one, maybe the most important is the buffer read cache, if you value fall down 99% then your buffers is probably not very useful adding some memory can get your application faster to work.

6. Swapping:

The Swapping panel report information about lightweight process. the graph report the number of 512 bytes pages swapped out to disk or swapped in to memory and the the number of swap request in and out. There is also a graph reporting the number of process switch (actually LWP switching) done by the processor. the last graph must be used with the “Run Queue” graph. If you have many process and many process switching may be your host is just going from one process to another without doing something useful.

7. Syscalls:

The Syscalls panel has four graphs. The amount of read/write call per second is on the first graph. The second graph is the number of syscalls. The third graph is fork/exec per second. The last graph is the amount of character read/write issued by the read/write system call.

8. File:

The “File” panel report information about file-system utilisation. iget is the number of inode request done by second. The namei is the number of name resolution per second. The last one dirbk is the number of directory block read by second.

9. TTY:

This panel report TTY usage.

10. Messages & Sempaphores:

This panel has one graph. it shows shared memory activity. Note that oracle is using semaphore a lot.

11. Paging (first page):

This Paging panel has four graphs. The first one is the number of page that a process ask for and is already in memory (shared lib/fork,..). The second graph is the number of pages and the number of page request transferred from the swap file system. The third graph is the number of page that should be in memory but either was paged out or a copy-on-write process want to modify it(vflt). The pflt is actually the number page request that cannot be satisfied due to an error. The last graph (fourth) is the number of page that was waiting to acquire a lock before going into memory.

12. Paging (second page):

This panel has four graphs, it report mostly what the page scanner do. The first graph is the number of page that has been freed in one second by the page scanner. The second graph is the number of page request and the number of page put into the swap file system. The third graph is how fast the page scanner search page to be freed, if you got high value that mean that the page scanner is frenetically search for page to freed...you maybe got a memory shortage.

13. Memory Usage:

The “Memory usage” panel has two graphs showing the free swap and the free memory available.

14. Kernel (small page/large page/oversize page):

This Three panels are the memory pool usage.

8. Understanding AIX graphs

1. To be done

9. Understanding Linux graphs

1. Depending on your sysstat version some graphs may have different types of data. This data may not be collected (reported as zero) depending on your kernel version.

2. CPU

The CPU panel may contain from two to three graphs. The first one is CPU used during the statistics interval. The last one is the percentage of time the CPU was idle (that is doing nothing). The middle reported CPU consumed by nice'd process.

If your CPU idle graph is at 0 for a long period of time, this probably means that your host needs to have more CPU power, but you should have a look at the time wasted waiting I/O, and time spent in system. If you are running a NFS server and your system time is very high, maybe you could have a look at your NFS parameter and try to tune some parameters to get things running better. If you got many waiting I/O you should have a look at the disks performance to see if your box is really wasting time on disk utilisation.

10. Understanding Mac graphs

1. CPU:

The CPU panel has two graphs, the upper graph is how cpu has been consumed, the second graph is for idle cpu. If your cpu idle percentage is blocked at zero for a long time and the user cpu time is high, considering buying a faster cpu can help you. But since Mac hasn't so much info is pretty hard to said something about cpu usage.

2. Page Out:

The page out graph show how many page a stored into swap space. This is done either your system doesn't have enough memory or system has used this page for a moment. Having high value continuously report that the system doesn't have enough memory.

3. Page In:

This panel has two graphs. The first graph is the number of page taken out of the swap space. The lower graphs has two values pflt (page fault) which is the number of page that were asked but there was an error or page than need to be duplicated for the "copy on write". the vflt (validity fault) counter is the number of page that should be in the memory but is not (eg. shared lib,...).

4. Block:

Block panel has two graphs. The upper graphs is the number of read/write per second. The lower is the number of disk block read per second (block side is device dependant). Beware that Diskimage are seen as a disk and blockname is set in order of mount command. this means that if you box has two disks disk0 and disk1, if you mount the diskimage A (this will be disk3), unmount it, and mount diskimage B (this will be also disk3), you will see only 1 disk3. The removable media problem appear also with usb/firewire disk.

5. Interface:

Each Interface has two panel. One for the traffic of the interface, the other for error.

1. Interface traffic:

The upper graph is the number of packet per second. The lower graph is the number of byte per second which arrive/leave the interface.

2. Interface errors:

The first graph is the number of errors per second. The last graph is the number of collision (coll) and the number of packet dropped per second.

11. Enabling sar on your host

1. The little unix script i used

```
#!/bin/sh

## sarl (
# written by FBA/XCH
#

PATH=/usr/lib64/sa:/usr/lib/sa:/bin:/usr/bin
LANG=C
TEMP=$1
SEC=${TEMP:-60}

DATE=`date +%d`
for i in /var/adm/sa /var/log/sa ; do
    if [ -d $i ] ; then DDIR=$i ; break ; fi
done

DFILE=$DDIR/sa$DATE
cd $DDIR

if [ -f "$DFILE" ] ; then
    find $DFILE -mtime +1 >/dev/null 2>&1 && rm -f$DFILE >/dev/null 2>&1
fi

H=`date '+%H'`
M=`date '+%M'`

LEFT=`expr \( 3600 \* 24 - 1 - $H \* 3600 - $M \* 60 \) / $SEC`
LEFT=`expr \( 60 \* 24 - 1 - $H \* 60 - $M \) `

( exec sadc $SEC $LEFT $DFILE </dev/null >/dev/null 2>&1 ) &
```

This shell script will collect statistics for your box until midnight, you can specify the interval time in second as a parameter. By default the script will collect system statistics every minutes.

2. Using my script on Mac.

The only thing to do is to cron the script a midnight. Please note that the Mac sar will rewrite the current file, so if you relaunch sar in the middle of the day previous data will be lost !!!

3. Using my script on Solaris

There are two things to do on Solaris to make this script runs smarter. The first thing is to enable the /etc/init.d/perf shell script installed with Solaris (you need to uncomment the file), at the end of the file add the path to sarl shell script. The second thing is to cron the sarl script via the sys account.

4. Using my script on Linux

By default the sysstat package install a default statistics collector in /etc/cron.d/sysstat. The only to do is to comment out the value and cron the script at midnight. Be aware that if the box reboot during the day, the collector will stop. If you want to avoid this put a little script in your run-level start-up directory to launch it. **** WARNING **** on linux you should have the sdc program have the -d options pass to have disk statistics collected (by default seems to be off)

ANNEXES

Solaris:

sar packages:

SUNWaccr	System Accounting, (Root)
SUNWaccu	System Accounting, (Usr)

Linux:

sar packages:

Sysstat (<http://perso.orange.fr/sebastien.godard/index.html>)

Developpement tools:

JFreechart : <http://www.jfreechart.org/>

iText : <http://www.lowagie.com/iText/>

Jsch : <http://www.jcraft.com/jsch/>

ACKNOWLEDGEMENTS

Thanks to : FBA,FSE,DPA,LRA,COZ,JB,ETH,AOL