

SAR: past, present, and future?



Peter Tribble, ProQuest

What is sar anyway?

NAME

sar - system activity reporter

SYNOPSIS

sar [-aAbcdgkmpqruvwxy] [-o filename] t [n]

sar [-aAbcdgkmpqruvwxy] [-e time] [-f filename] [-i sec] [-s time]

Enabling sar

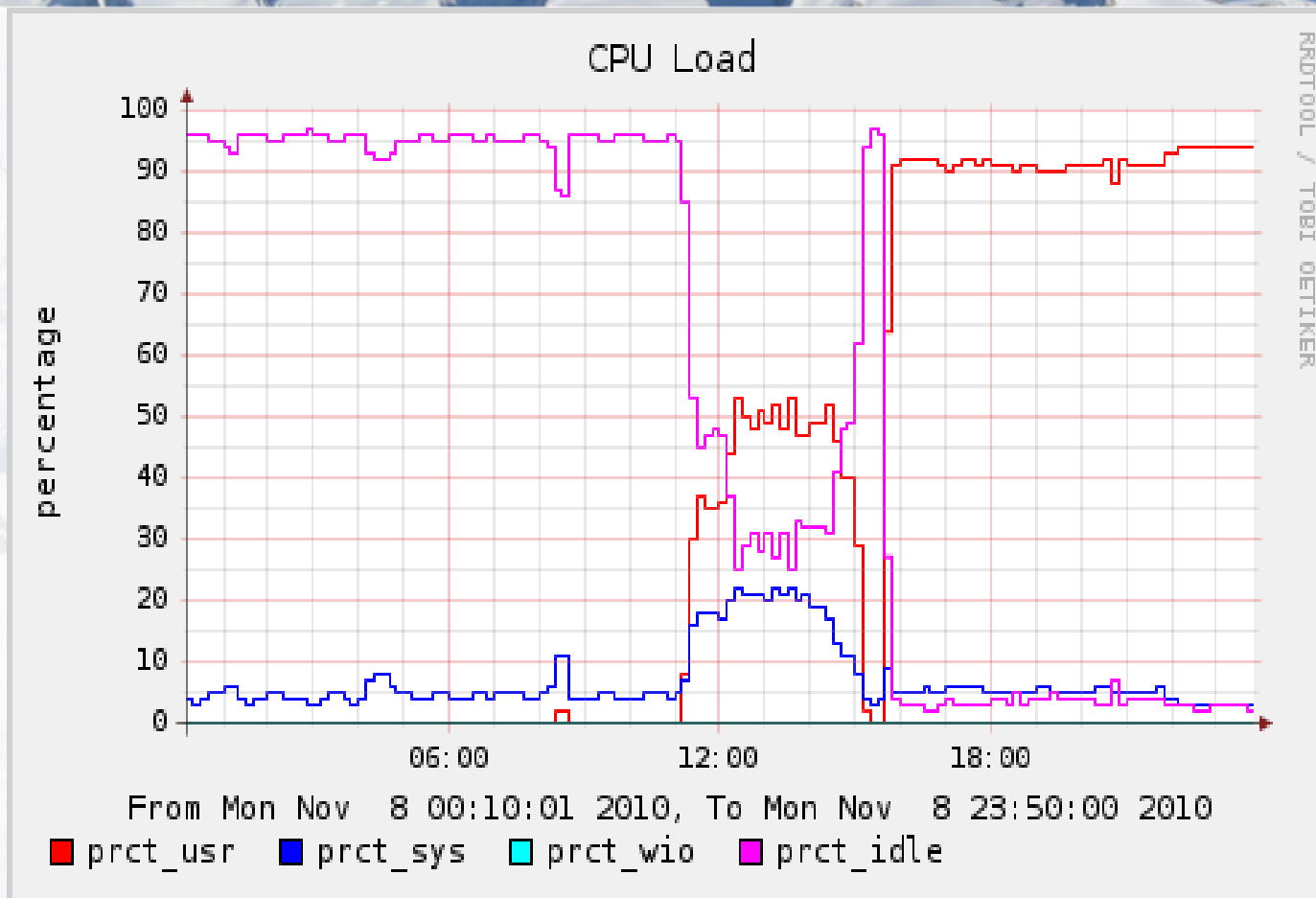
- First, enable the sar service
 - `svcadm enable sar`
- Second, configure the sys crontab

```
# My standard accounting  
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1  
55 23 * * * /usr/lib/sa/sa2 -s 00:00 -e 23:59 -i 600 -A
```

This gives you - what?

- Vanilla sar – cpu utilization summary
- -f to select different file (day)
- -d gives I/O statistics
- -c for syscalls, includes fork+exec
- -v includes process count
- Some other useless options

sar2rrd



So what's the problem?

- CPU information limited to basic utilization
 - And aggregated
- No network data **at all**
- Nothing like mpstat
- Nothing like fsstat
- No NFS statistics
- No ZFS statistics

Problem summary

- Data limited
- Presumably useful once
 - when dinosaurs roamed the earth
- File format not extensible
 - So stuck in the past

Extend sar?

- Identify the useful statistics
 - How to choose?
 - Everyone will have different ideas
- Add them to sar to make v2.0
 - So next week have to do this all over again
 - Doesn't allow for future evolution
- Must do better!

Back to basics

- Underlying data is mostly kstats
- Soooooooooooooo.....
- Just save the kstats
- And do so indiscriminately
 - I'm not going to choose
 - Don't stop people doing new stuff

Enter KAR

- KAR – kstat activity reporter
- Originally, kstat -p
- Newer, custom collector
 - More metadata, more compact
- Saved in zip archives
 - 1.2x-5x old sar file size
 - @ 5 minutes, < 1GB/month

KAR tools

- Basic sar emulation
- An iostat lookalike
- An mpstat lookalike
- An fsstat lookalike
- Basic set of graphs
- Meta information

*stat

00:05:02 file system statistics

new	name	name	attr	attr	lookup	rddir	read	read	write	write		
file	remov	chng	get	set	ops	ops	ops	bytes	ops	bytes		
0	0	0	1206	0	4156	0	124	2206K	0	0	vopstats_nfs4	
0	0	0	1106K	0	721K	15K	43K	362M	0	0	vopstats_zfs	

00:25:02 processor statistics

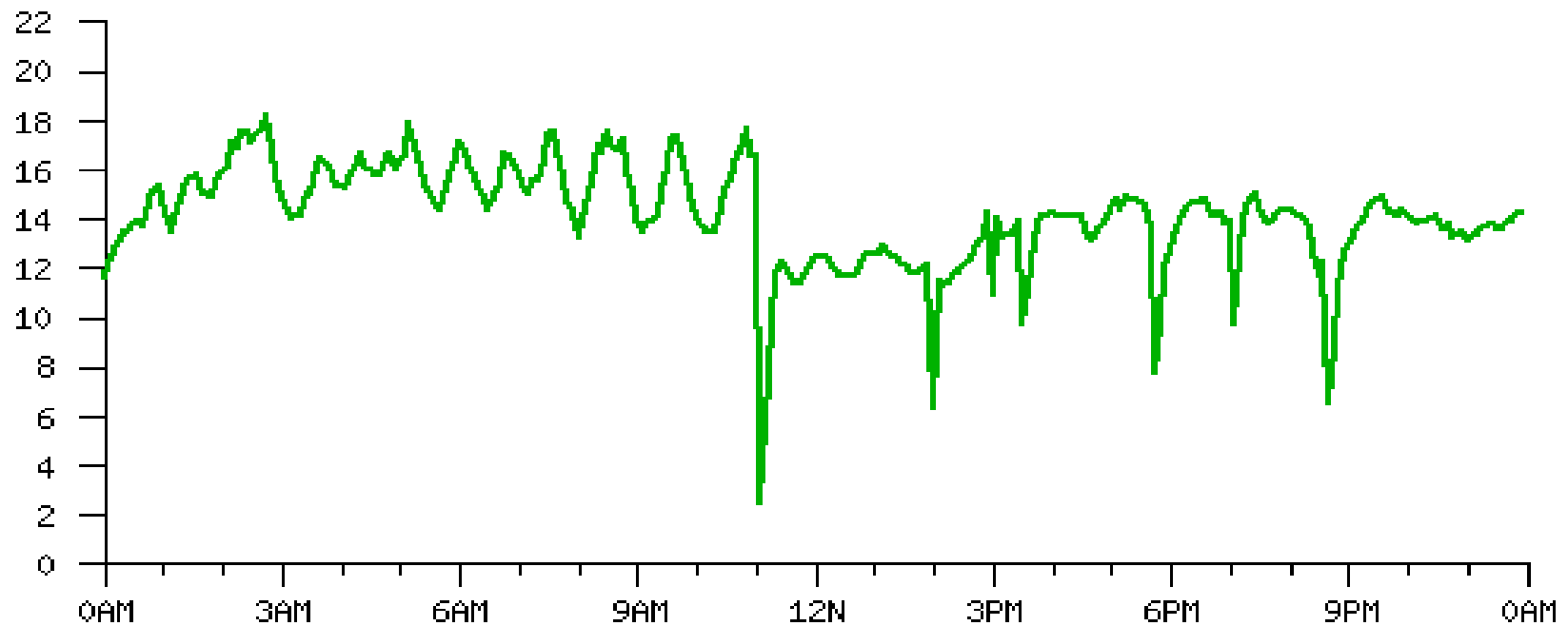
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	idl
0	3	0	221	2491	2062	830	1	53	164	4	39	0	5	95
1	1	0	0	646	6	1081	0	59	125	3	50	0	2	98
2	0	0	0	522	4	882	0	46	112	4	67	0	2	98
3	0	0	0	537	2	957	0	49	111	3	25	0	2	98

What's changing?

Total kstats: 1336
Total statistics: 28858
Numeric statistics: 28029
String statistics: 829
Statistics zero: 15867
Statistics Changed: 5324
Kstats changed: 577

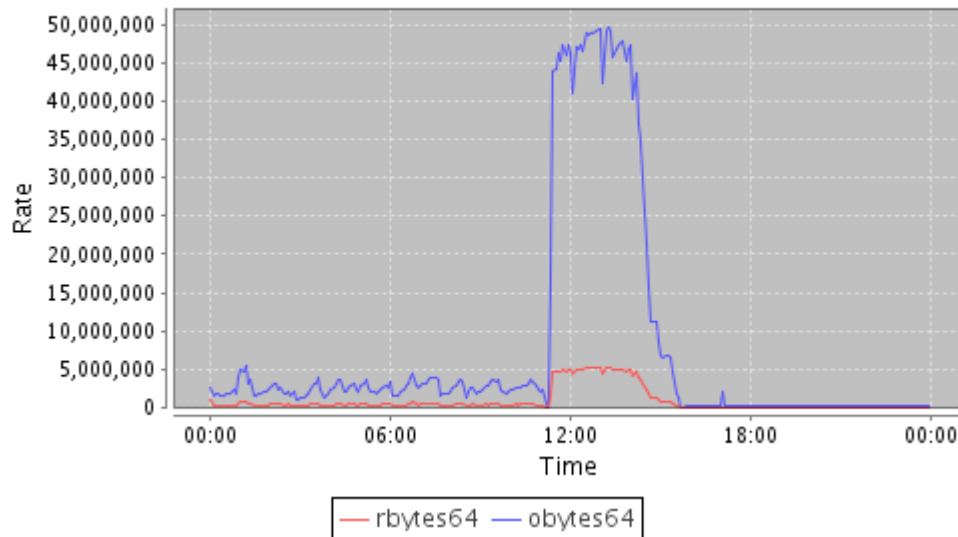
print to extract data

```
kar print -f ~/aldebaran.zip -M zfs -S size  
pl -prefab chron data=x1 unittype=time mode=line x=1 y=3 yinc="1 1000000000"
```

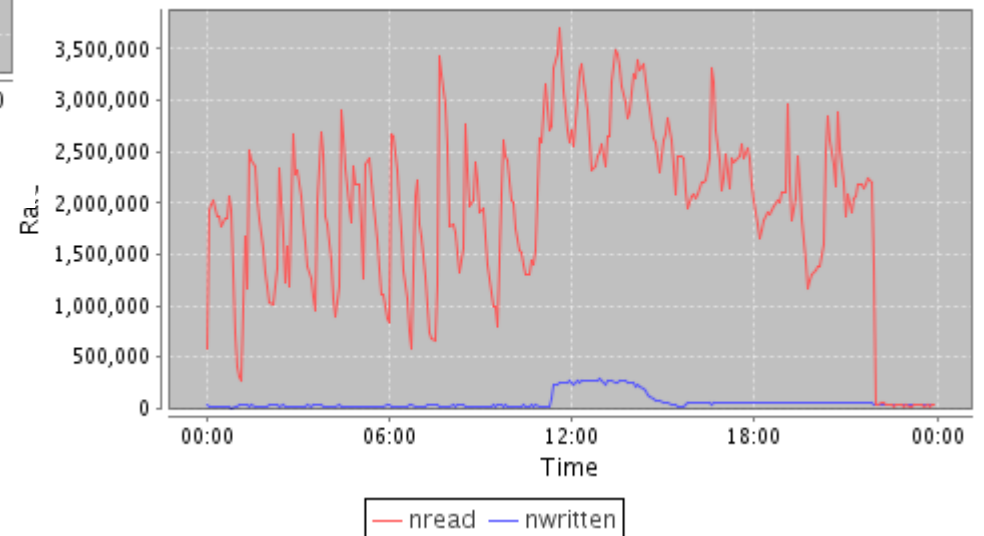


Network/disk graph

nge:0:mac



sd:18:sd18



Problems...

- Relies on jkstat
 - Parser is ****slow****
 - Implementation or fundamental?
- Missing metadata
 - Not in kstats at all (or sar)
 - Such as device, mountpoint mappings

Implementation alternatives

- KAR just an experiment
 - Basic idea of saving kstats sound
- Just files, so standard tools can grok the data
 - KISS rules
- Database to store the data?
- Dump into rrd files?
 - Useful as secondary format

Alternative Tools

- Why keep sar or its ilk at all?
- SNMP
 - Needs management infrastructure
- Zabbix, nagios, et al
 - Still need the management infrastructure
 - Ridiculously hard to set up
- Still limited by fixed set of statistics



Thank You!

Questions?

<http://www.petertribble.co.uk/Solaris/kar.html>