rrdtool to the rescue

jq – fpw2007

time-series data

✓ time series data

- collected at regular interval
- aggregated to graph trends

✓ easy to gather…

- load average
- network i/o
- temperature
- etc.

➔very useful for sys-admin needs

tsd problems

- ✓ big amount of data…
  - … especially when:
    - x metrics
    - y servers
    - z years of retention
    - (5 min = 288 plots / metric / server / day)
- ✓ efficiency problems
  - storage
  - processing
- ✓ graphing
  - transforming raw data in a visually appealing graph

some solutions?

✓ plain text files

- slow (lots of files + find + update)
- no easy date handling

➔ discarded

✓ database

- what database schema?
- millions-line table unmanageable

➔ discarded

but in fact…

- ✓ why couldn't we change granularity?
- ✓ time-series data deprecate
  - recent values important
  - old values tend to be not so useful
- ✓ especially true for sys-admin needs

➔ *granularity change over time!*

rrdtool to the rescue

- ✓ rrdtool is **the** solution
  - mrtg successor
  - easy date handling
  - aggregate data over time
  - constant disk amount
- ✓ Perl bindings
  - do exist…
    - complete
    - fast
  - … but not on CPAN
  - see http://rrdtool.org

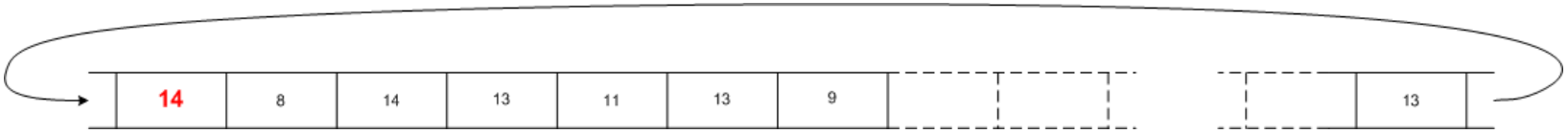
rrdtool principles

- array of timestamp / values

rrdtool principles

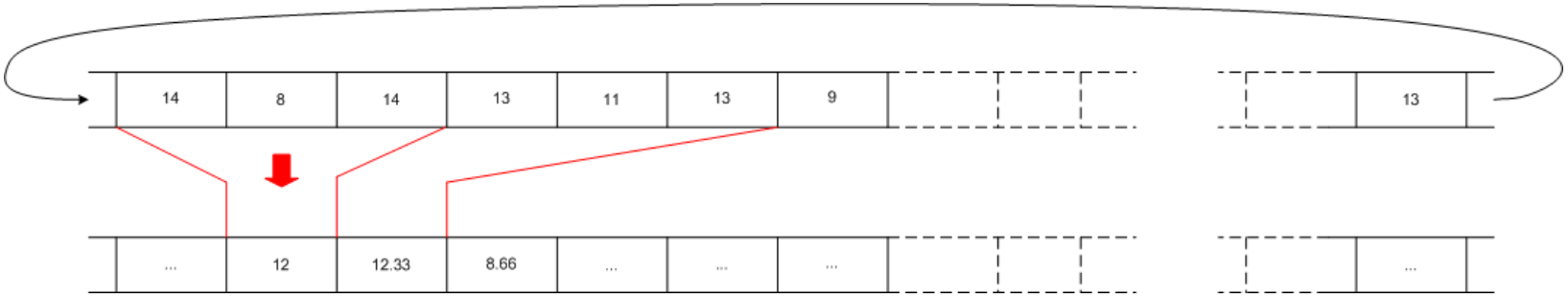
- array of timestamp / values

rrdtool principles



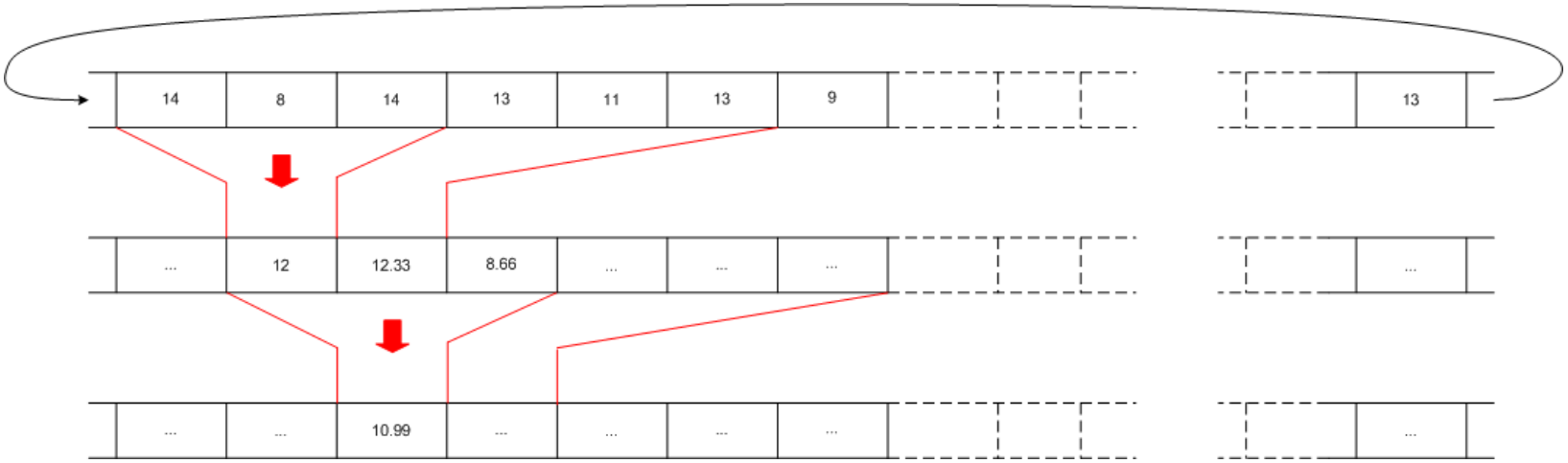
- array of timestamp / values
- wrap up when full ➔ constant size

rrdtool principles



- array of timestamp / values
- wrap up when full ➔ constant size
- aggregate values

rrdtool principles



- array of values
- wrap up when full ➔ constant size
- aggregate values
- as many levels as wanted!

gory details

- ✓ round robin database
  - 1 file
  - one to many data sources
    - different types — gauge / counter / derive / etc.
    - heartbeat — plots frequency
    - optional min / max values
  - one to many rras
    - different algorithms — avg / min / max
    - consolidation interval — guestimate value if missing entries
    - steps — # of plots aggregated
    - rows — number of cells in the array

example

```
RRDs::create
 ( $rrdfile,
   "--start=$timestamp",
   "--step=300",                    # 300s = 5mn = default
   "DS:var1:GAUGE:600:U:U",         # heartbeat = 600s = default
   "DS:var2:GAUGE:600:U:U",         # GAUGE = free numbers
   "RRA:AVERAGE:0.5:1:8928",
                  # all plots kept for 1 month (60/5*24*31=8928)
   "RRA:AVERAGE:0.5:12:17520",   # 1 plot/hour for 2 years
   "RRA:AVERAGE:0.5:288:2555",   # 1 plot/day, 7 years
 );

[... later on ...]

RRDs::update( $rrdfile, "$timestamp:$val1:$val2" );
```

what about graphing?

- ✓ rrdtool designed for graphing
- ✓ easy & powerful interface
- ✓ timestamps / dates automatically handled a scaled
- ✓ aggregate source from different rrds
- ✓ on-the-fly computing with RPN

example

```
RRDs::graph
  ( $pngfile,
    "--imgformat==PNG",
    "--start=$start", "--end=$end",
    "--width=$w", "--height=$h", "--rigid",
    "--lower-limit=0", "--upper-limit=100",
    "DEF:usr=/path/to/usr.rrd:var1:AVERAGE",
    "DEF:sys=/path/to/sys.rrd:var2:AVERAGE",
    "CDEF:idle=usr,sys,+,100,-",
    "AREA:sys#483d8b:% kernel",
    "STACK:usr#4169e1:% user",
    "STACK:idle#b0c4de:% idle",
);
```

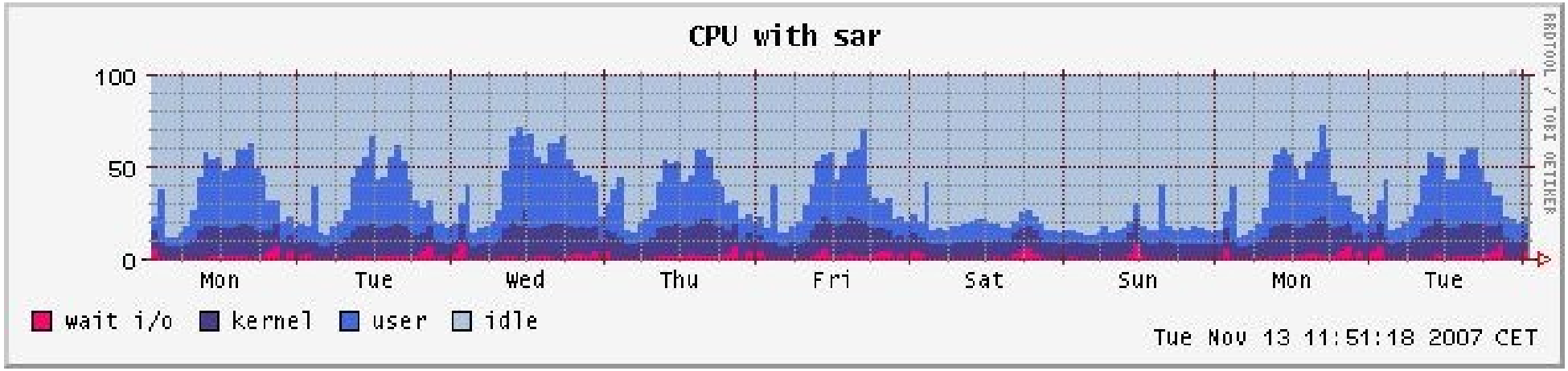
some gotchas

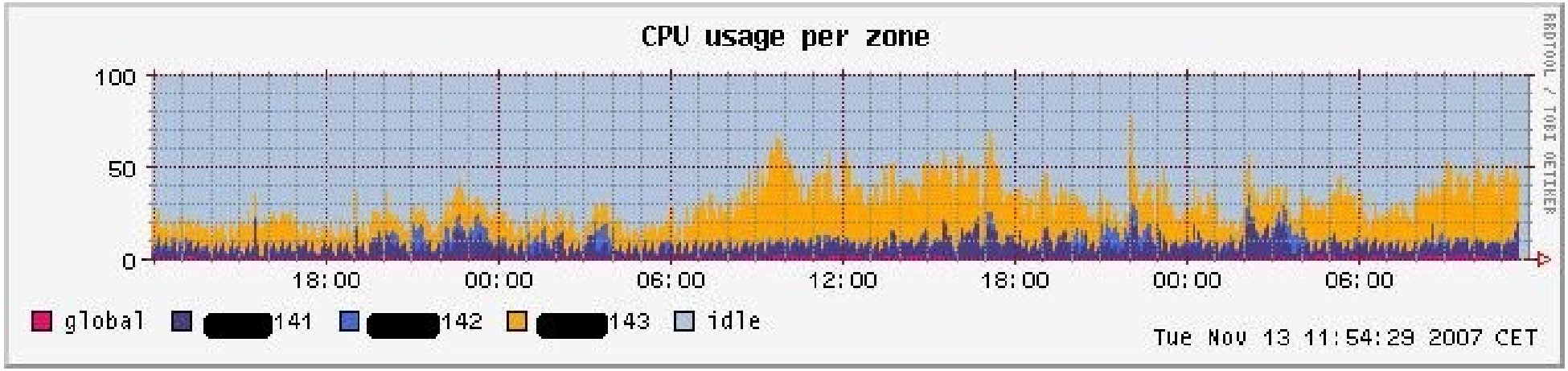
- ✓ beware of unknown value
  - value + U = U
- ✓ interpolation
  - plots don't always happen on exact regular interval
  - when not exact, rrdtool interpolates

  ➔ load avg=3.0 at 298s may show as 3.02

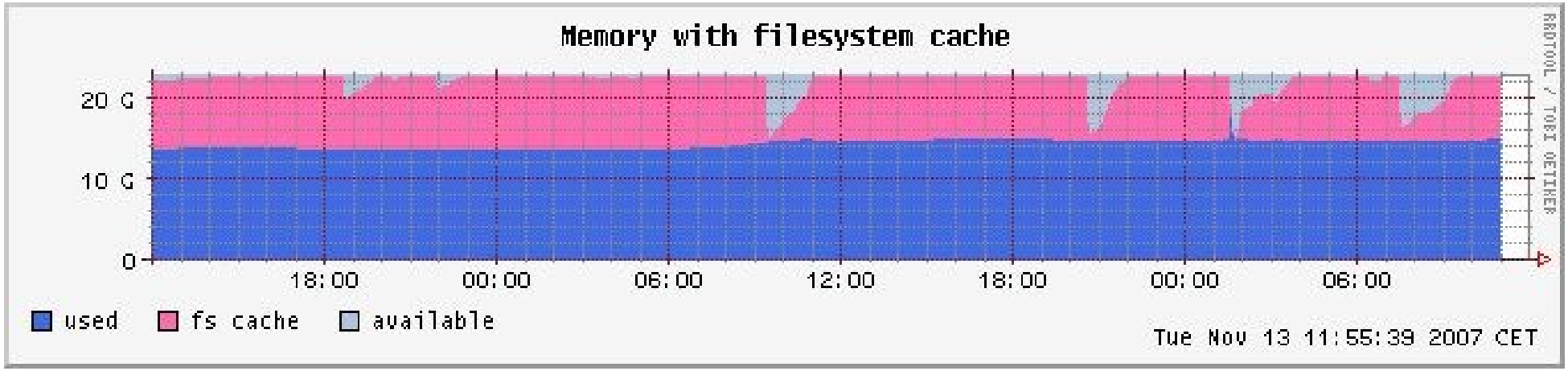
  ➔ takes a while for the plots to show up (need multiple samples)
- ✓ beware of autoscaling on y axis ()
  - compare oranges to apples: use 0-100% scale for non-technical ppl
  - min/max needed to avoid spikes
- ✓ a plot on the graph a trend does not make

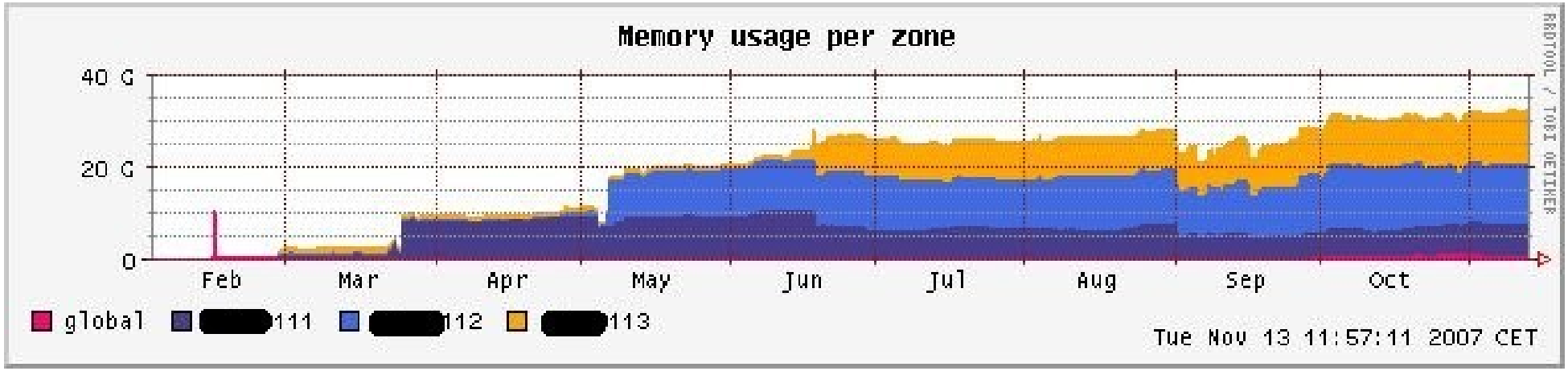
CPU usage


CPU with sar
100
50
0
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Mon
Tue
wait i/o
kernel
user
idle
Tue Nov 13 11:51:18 2007 CET
RRDTOOL / TOBI OETIKER

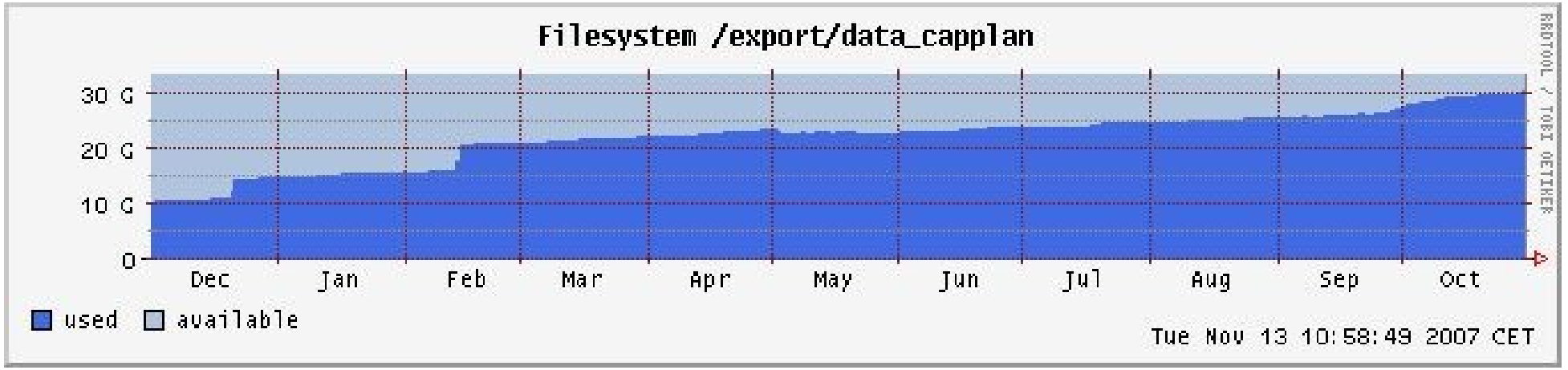

CPU usage per zone
100
50
0
18:00
00:00
06:00
12:00
18:00
00:00
06:00
global
[redacted]141
[redacted]142
[redacted]143
idle
Tue Nov 13 11:54:29 2007 CET
RRDTOOL / TOBI OETIKER

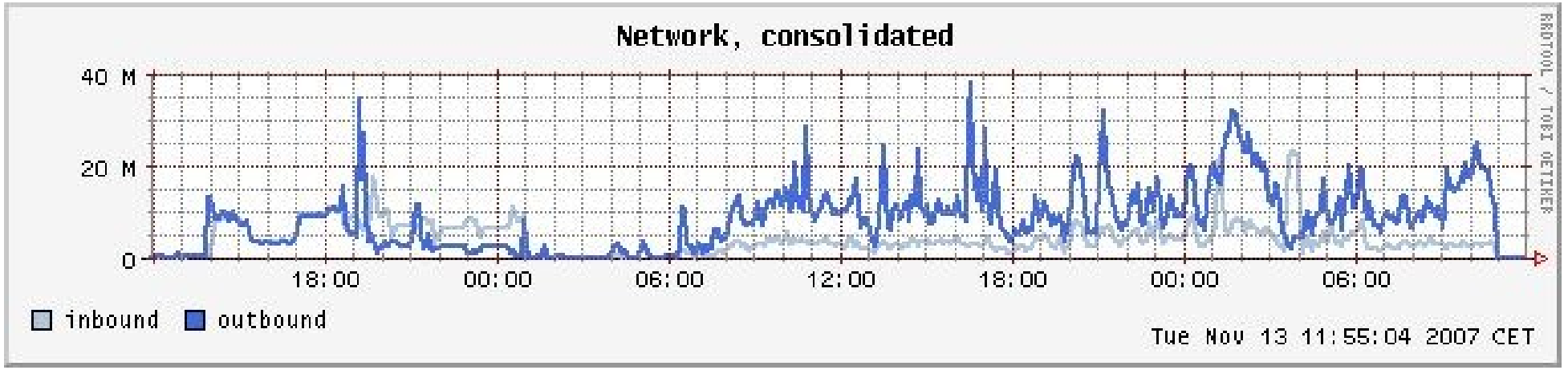
Memory usage





I/O usage


Filesystem /export/data_capplan
30 G
20 G
10 G
0
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
used
available
Tue Nov 13 10:58:49 2007 CET
RRDTOOL / TOBI OETIKER


Network, consolidated
40 M
20 M
0
18:00
00:00
06:00
12:00
18:00
00:00
06:00
inbound
outbound
Tue Nov 13 11:55:04 2007 CET
RRDTOOL / TOBI OETIKER

conclusions

✓ rrdtool

- very useful for time-series-data
- light-weight
- easy graphing

✓ but…

- may not be convenient for everything
- some gotchas

➔ **GIVE IT A TRY!**