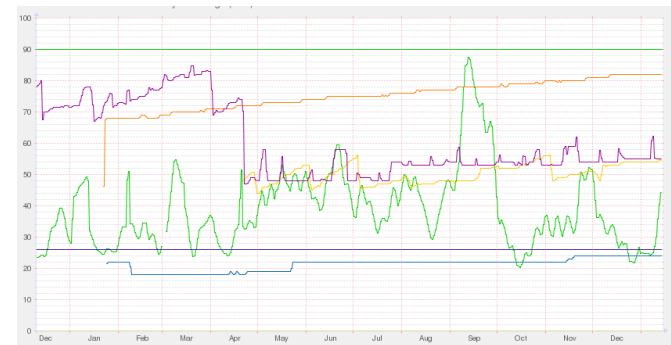
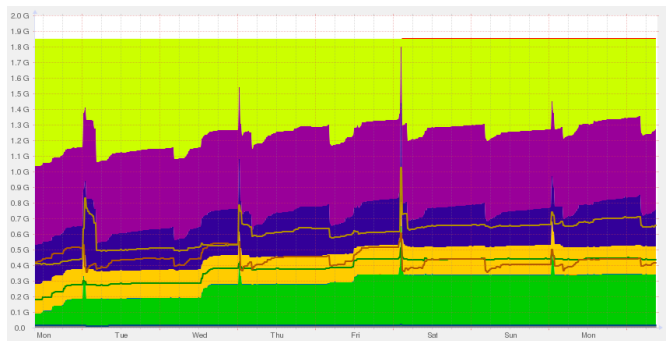


MUNIN - Langzeitmonitoring

Oder: was macht mein Server nachts um 3 Uhr ?

ATIS - Abteilung Technische Infrastruktur, Fakultät für Informatik



Gliederung

- Motivation
- Anforderungen
- Demo
- Anwendungen
- Plugins
- Sonstiges

Motivation

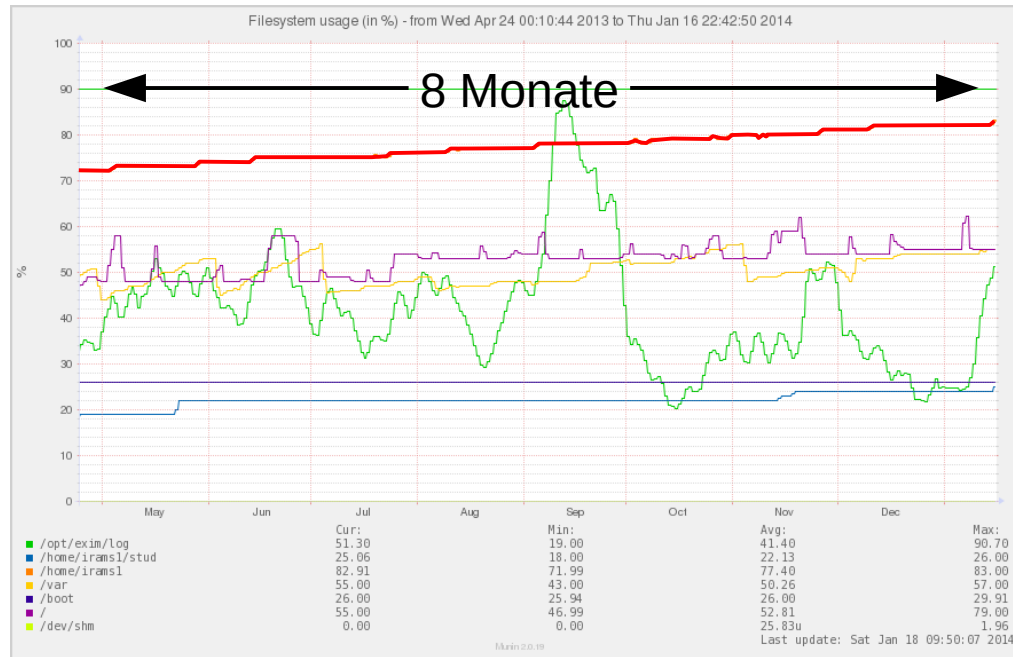
- „Der Server ist langsam, kannst Du mal schauen ?“
 - Wo / wie / wann schaut man ?
 - top, iostat, netstat, sar,

- Der Server soll ersetzt werden, was kauft man ?
 - Viel hilft viel ?
 -oder doch nicht ?

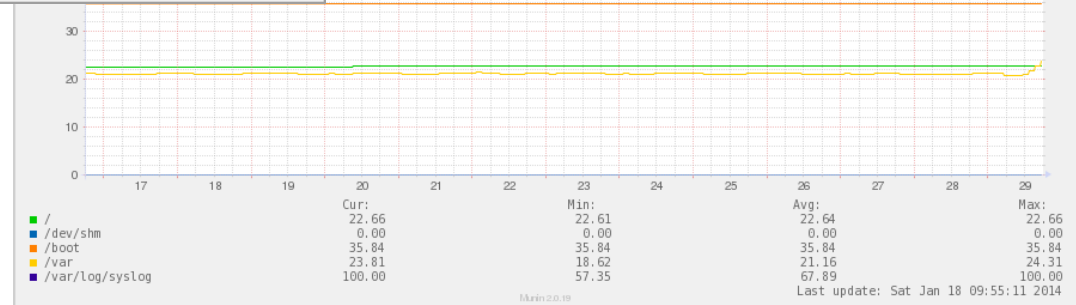
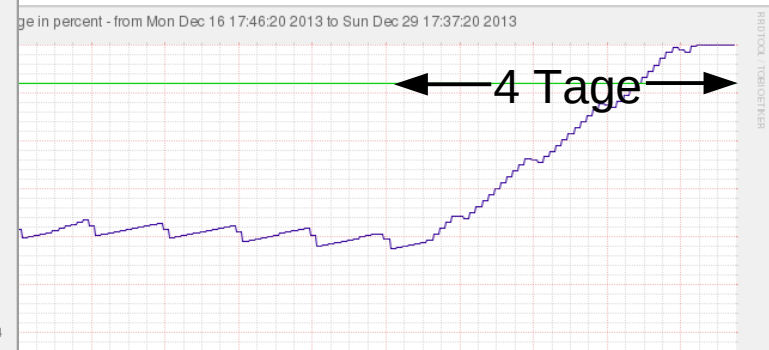
- Wieviele User nutzen den Dienst „XY“ ?

Motivation

Icinga / Nagios meldet Festplatte ist zu 90% voll



So ?



Oder so ?

Anforderungen

- Munin ist Open Source unter <http://munin-monitoring.org/>
- Munin erfordert einen **Munin-Server**
 - Webserver mit CGI und rrdtools (bekannt vom MRTG) (keine Datenbank!)
 - Der Server weiß zunächst nichts über den Client, außer dessen Namen !
- Die zu überwachenden Clients einen **Munin-Node**
 - Daemon der auf Port 4949 lauscht
 - munin-Plugins für die zu überwachenden Dienste / Kennwerte
 - Plugins sind beliebige Skripte, die unter **/etc/munin/plugins** liegen
- Die Server- und Node-Pakete sind in den Distributionen enthalten

Anforderungen: Wie macht man das konkret ?

- Auf dem Master:
 - `/etc/munin/munin.conf`
`[Webserver;host.example.com]`
`address 192.168.10.13`
`use_node_name yes`
`notify_alias host`
 - Cron-Job, der alle 5 Minuten die Clients abfragt

- Auf dem Client
 - `yum install munin-node` bzw. `apt install munin-node`
 - Öffne Port 4949/TCP
 - Erlaube IP des Munin-Servers
 - Lege plugins an (munin-node-configure)
 - Starte den Daemon
 - Warten !

Anforderungen: Datenspeicherung

- Munin speichert seine Daten in rrd-Datenbanken
 - Ältere Daten werden verdichtet
 - Und nach einem Jahr „hinten“ rausgeschoben
 - Dadurch endliche Größe für die Datensätze

- Plattenplatz
 - ~100 Clients
 - ~30.000 rrd-Dateien
 - ~3 GB

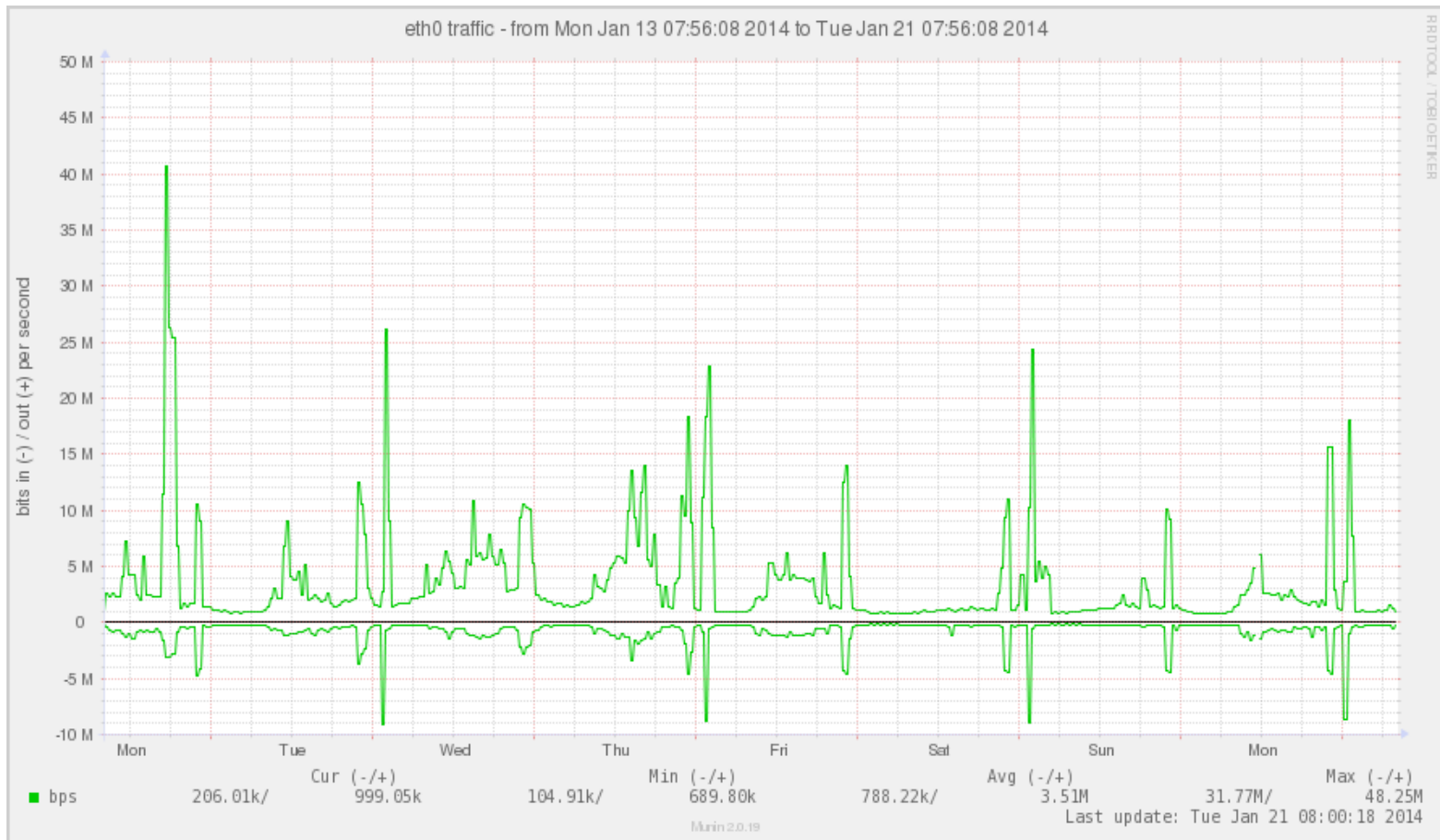
- IOPs !!!

Demo

<https://munin.informatik.kit.edu>

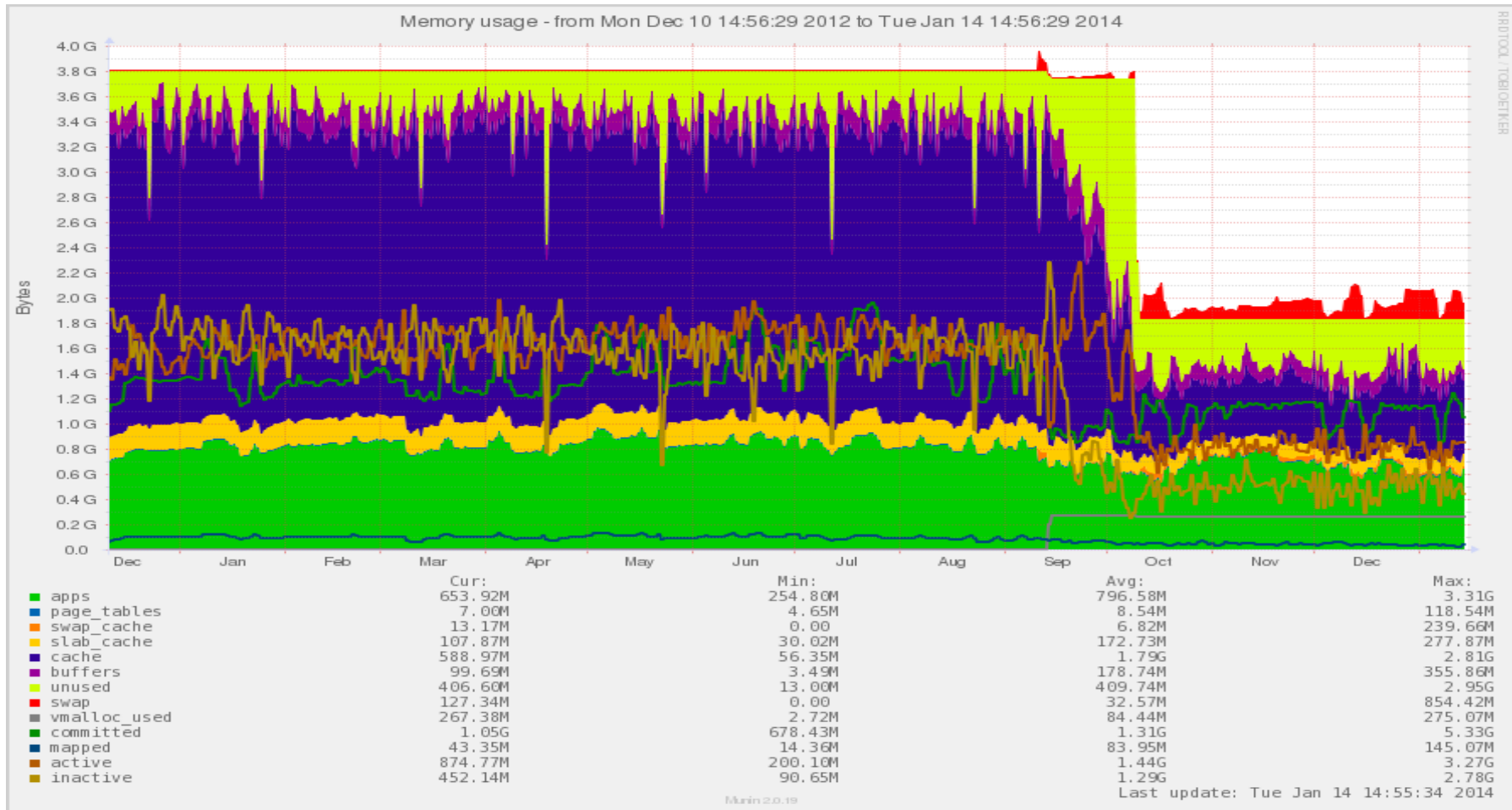
Anwendung

- Gigabit für den Mailserver ?
 - 10 Mbit würden schon fast reichen !



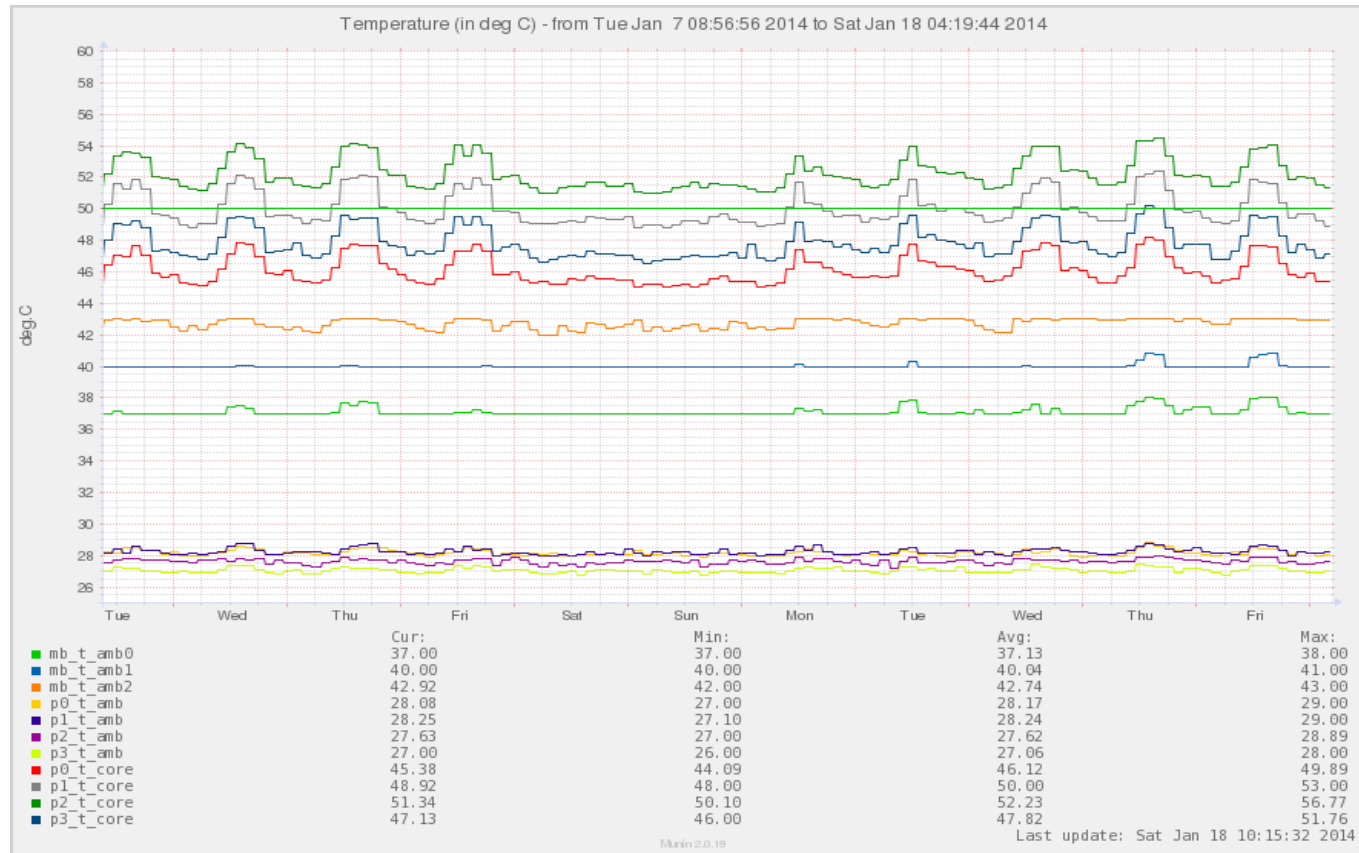
Anwendung

■ 2 GB RAM sind (fast) genug



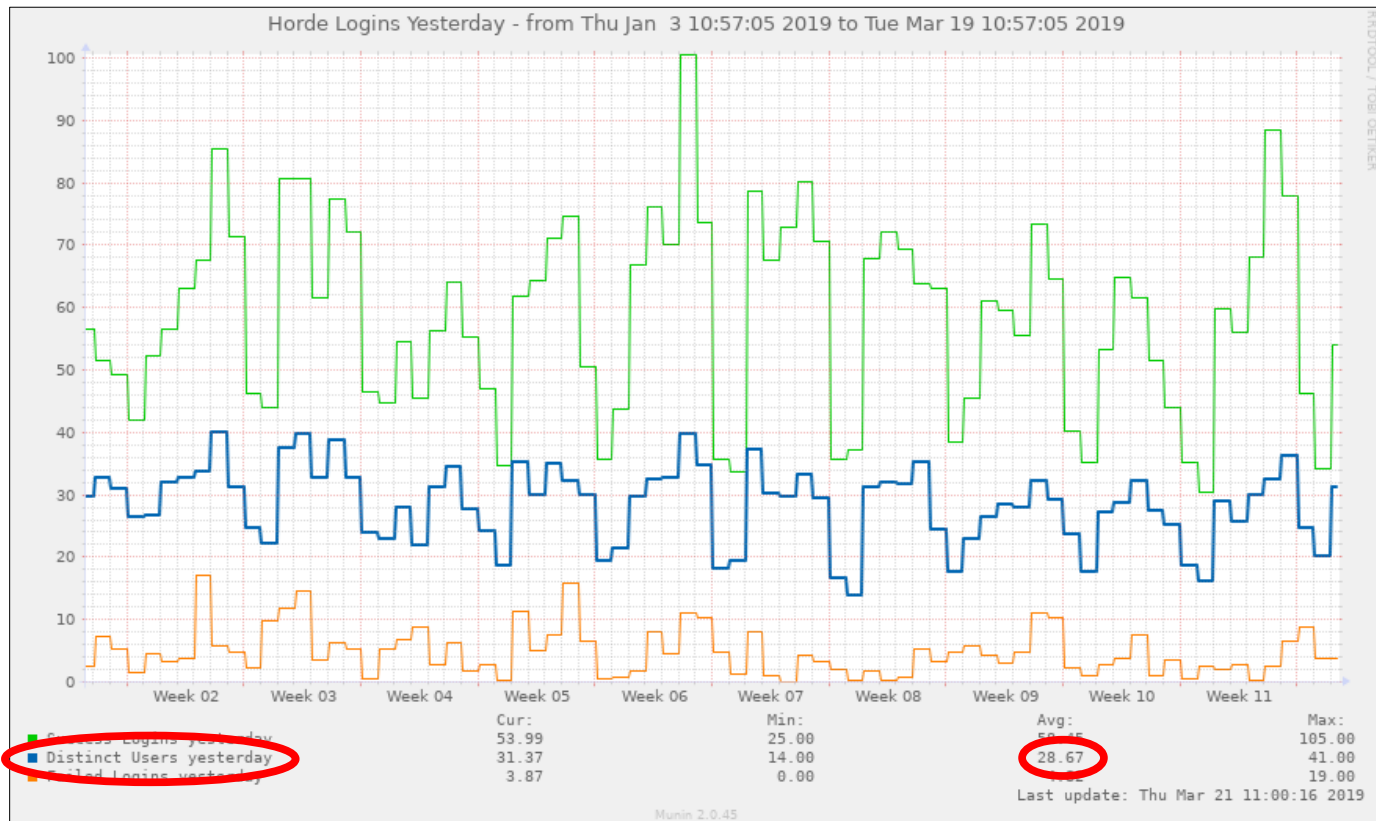
Anwendung

- Welchen Wochentag haben wir heute ?
 - An Hand der CPU-Temperatur !



Anwendung

- Wie viele verschiedene User nutzen Dienst XY ?



- gemittelt über 10 Wochen knapp 30 verschiedene Nutzer pro Tag

Anwendung

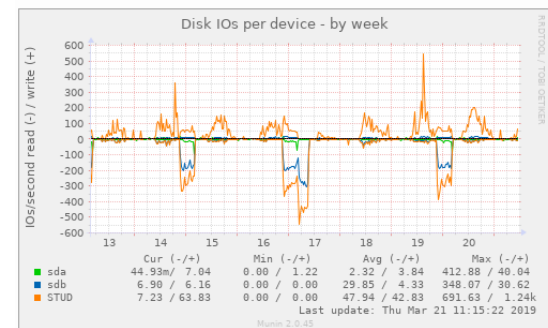
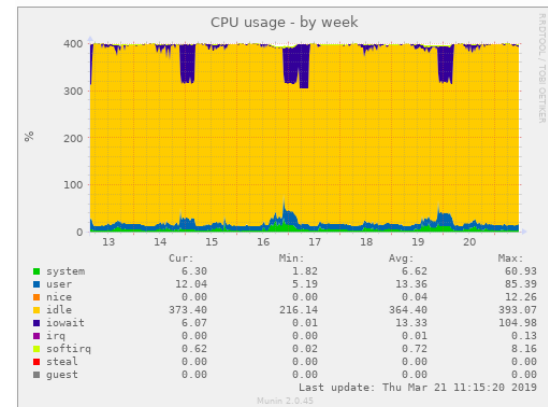
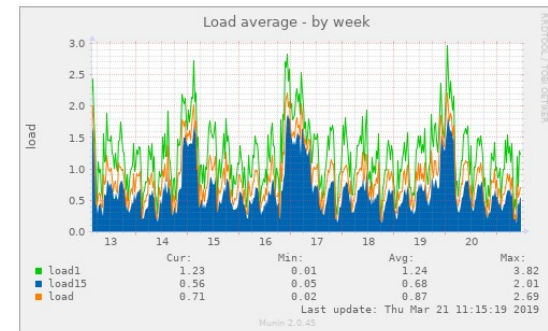
■ Was macht mein Server nachts um drei?

■ Die „load“ ist erhöht

■ Die CPU steht macht „iowait“

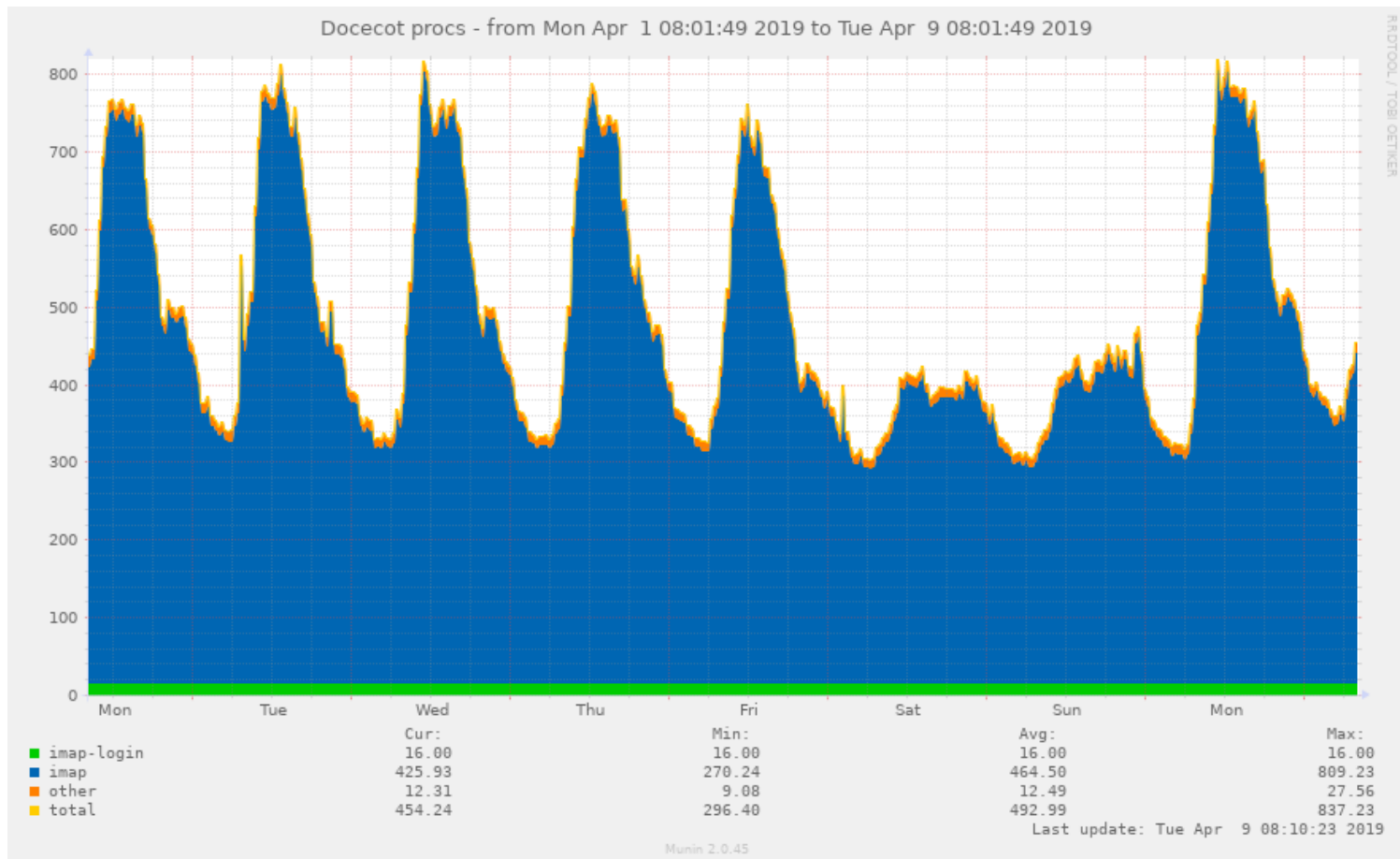
■ Lesezugriffe auf die Platte mit den Heimatverzeichnissen

■ Klar - das ist der Backup-Job !



Anwendung

- Wann gehen die User in die Mensa und ins Bett ?
 - oder wann bootet man den Server ?



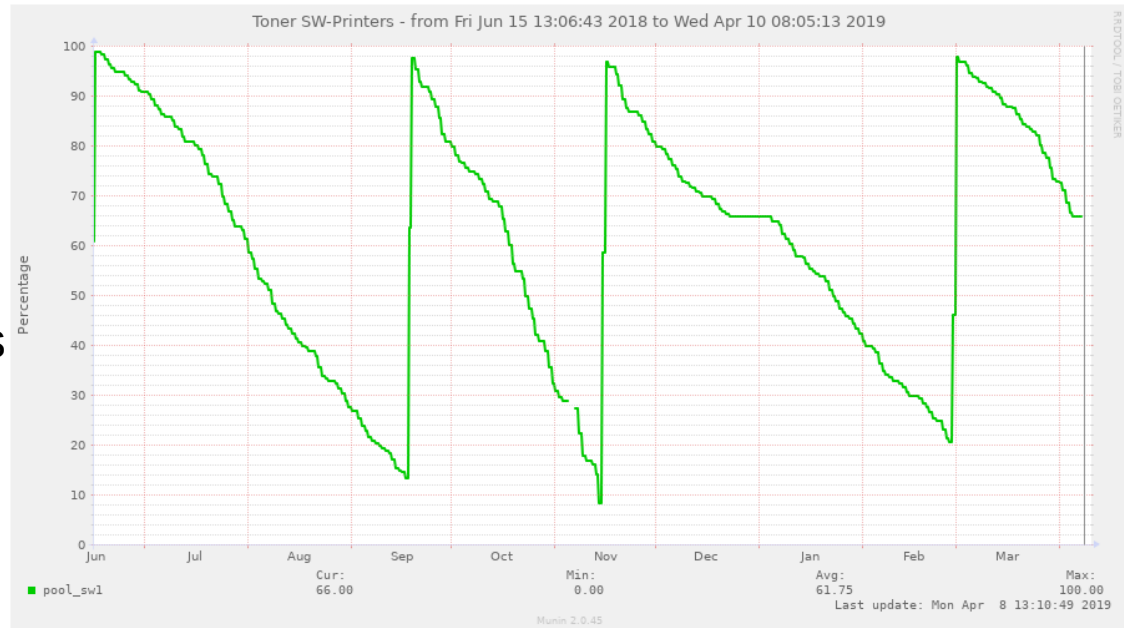
Wie lange reicht der Toner im Schrank?

■ Zu Fuß:

- Zettel an der Wand neben dem Drucker
- Jeder der Toner wechselt, schreibt das Datum und Druckerzählerstand auf
- ... und das funktioniert ?

■ Oder Munin:

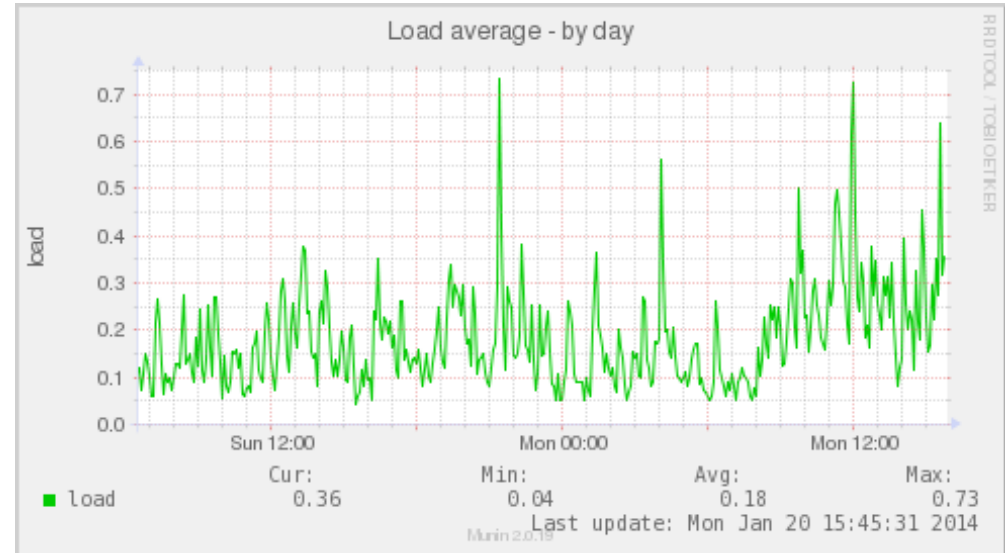
- Ein Blick genügt um zu sehen, dass eine Kartusche etwa zwei bis drei Monate reicht.



Plugins

- Viele Standard-Plugins sind im munin-node-Paket enthalten
 - `munin-node-configure`
 - `ls /usr/share/munin/plugins`
 - oder von <https://github.com/munin-monitoring/contrib/>
 - oder bei „Suchmaschine“

- oder selber schreiben !
 - Beispiel: „load“-Plugin



Plugins: Server-Client-Dialog

■ # telnet host.example.com 4949

```
Trying 192.168.10.13...
Connected to host.example.com.
Escape character is '^]'.
# munin node at host.example.com
```

■ list

```
cpu df forks if_eth0 load memory uptime users [...]
```

■ config load

```
graph_title Load average
graph_args --base 1000 -l 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes ....
load.info 5 minute load average
```

■ fetch load

```
load.value 0.43
```

■ quit

Plugins: Beispiel für „load“

```
#!/bin/bash
if [ "$1" = "config" ]; then
    echo 'graph_title Load average'
    echo 'graph_args --base 1000 -l 0 '
    echo 'graph_vlabel load'
    echo 'graph_scale no'
    echo 'graph_category system'
    echo 'load.label load'
    echo 'graph_info The load average of the ....'
    echo 'load.info 5 minutes load average'
    exit 0
fi
echo -n "load.value "
cut -f2 -d' ' < /proc/loadavg
```

Sonstiges

- `logtail` eignet sich gut bei sehr großen Logdateien
 - „tail -f für Cronjobs“
- Munin kann auch Benachrichtigungen versenden
- Munin kann man auch mit dem nrpe von Nagios / Icinga koppeln
- Windows-Clients
 - es gibt kaum Plugins
 - probing von außen per SNMP
- Munin kommt nur alle 5 Minuten, öfter geht nicht
 - Man hat keinerlei Aussage, was in den 4'59" dazwischen stattfindet
- Daten älter als ein Jahr gehen verloren !

Wer viel misst, misst Mist !

Vielen Dank für Ihre Aufmerksamkeit

Olaf.Hopp@kit.edu