

Monitoring @ \$HOME

Die Maschine beobachtet das schon eine ganze Weile.

Alexander Schreiber <als@thangorodrim.ch>

<http://www.thangorodrim.ch/>

Chemnitzer Linux-Tage 2015, 2015-03-22

Nicht alles was zählt, kann gezählt werden, und nicht alles was gezählt werden kann, zählt!

– Albert Einstein

Inhalt

- 1 Übersicht
- 2 Monitoring mit Alarm
- 3 Monitoring mit Graphen
- 4 Sensoren
- 5 Ende

Über den Autor

- beschäftigt sich seit fast 20 Jahren mit Linux
- tätig als Systemingenieur bei Google Switzerland
- hat den einen oder anderen Computer im Haus

Um was geht es?

- Monitoring
 - Werte/Zustände messen, ggf. Alarm schlagen
 - Werte messen, Graphen erzeugen
- Sensoren
 - Was kann ich wie messen ...
 - ... ohne ein Vermögen zu bezahlen.

Monitoring - Graphen

- kontinuierlich Messung und Aufzeichnung von Werten
- Erzeugung entsprechender aussagekräftiger Graphen
- ermöglichen schnellen Überblick und Trenderkennung
- Zielgruppe: Mensch

Monitoring - mit Alarm

- kontinuierliche Erfassung von Werten
- laufender Vergleich von Ist- mit Soll- und Schwellwerten
- ggf. Statusänderung (z.B. grün → rot)
- ggf. Auslösung von Alarmsignalen (Email, Pager, IM, ...)
- Zielgruppe: Mensch sowie ggf. Maschine (automatische Reparaturmassnahmen)

Allgemeines

- Für wen: Heimanwender, Grundlagenkenntnisse Linux, keine erfahrenen Admins
- Anforderung: einfach zu installieren, einfach zu konfigurieren
- “Auspacken, Einschalten, Geht”-Prinzip
- idealerweise einfach per `apt-get`
- Debian Monitoring: <http://debmon.org/>
- `deb http://debmon.org/debmon debmon-wheezy main`
- Konfiguration per einfacher (!) Textdatei

Nagios

- der Klassiker seit 15 Jahren, mittlerweile Nagios3
- startete als NetSaint, nach Namenskonflikt (Security Scanner) zu Nagios umbenannt
- aktive Community
- umfangreiche Pluginsammlung um alles zu überwachen
- relativ einfache Konfiguration über Textdateien
- kann jedoch auch komplexere Systeme abbilden
- <http://www.nagios.org/>
- `apt-get install nagios3`

Nagios, erweiterbar

- Nagios Checks als Plugins implementiert
- mehrere Plugin-Pakete bereits fertig installierbar
- zudem umfangreiche Pluginssammlungen im Netz
- wenn das nicht reicht:
 - sehr einfache Plugin API
 - (mindestens) eine Zeile Text ausgeben
 - exit code 0..3 (ok, warning, critical, unknown)
 - <http://sourceforge.net/projects/nagiosplug/>
 - damit Plugins in Sprache der Wahl realisierbar
- verschiedene Methoden für Remote-Systeme: ssh, nrpe, ...

Nagios, vielfältige Alarmierung

- Standard: EMail
- einfach erweiterbar, zum Beispiel:
 - via IRC (NagIRCBot)
 - direkt aufs Android Telefon (aNag)
 - via Jabber (nagios-notification-jabber)
 - SMS (verschiedene Möglichkeiten: GSM Modem, SMS Gateway, . . .)

Icinga

- basierend auf Fork von Nagios
- u.a. entstanden als Reaktion auf Stillstand von Nagios Entwicklung und andere Probleme
- mittlerweile Weiterentwicklung in verschiedenen Details
- Check Plugins von Nagios weiter verwendbar
- `apt-get install icinga2`

Xymon

- basierend auf BigBrother
- Namensgeschichte: bbgen toolkit → Hobbit → Xymon
- Namensverwirrung in Dateisystemstruktur:
/etc/hobbit/bb-hosts
- BigBrother, ein Monster in Shell, Xymon hat Reste davon
- benötigt eigenen Client für Remote-Systeme
- `apt-get install xymon`

SmokePing

- Netzwerklatenz in Graphen
- sehr einfach zu konfigurieren (Liste von Hosts, gruppierbar)
- generiert smoke plots
- verwendet ICMP ping → keine Software auf Zielen nötig
- `apt-get install smokeping`

SmokePing Konfiguration

- `/etc/smokeping/config.d/Targets`
- `+ Google_DNS`
 - `menu = Google DNS`
 - `title = Google DNS`
 - `++ 8_8_8_8`
 - `menu = 8.8.8.8`
 - `title = 8.8.8.8`
 - `host = 8.8.8.8`

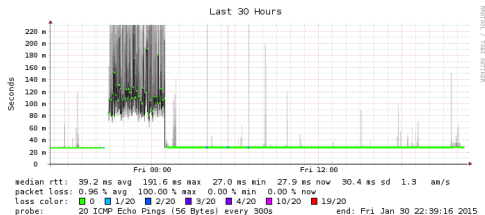
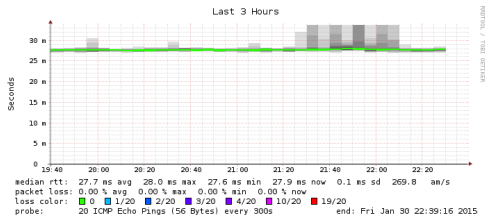
SmokePing Beispiel

SmokePing
Targets:

Filter:

- Charts
 - Local Wired
 - Local WLAN
 - external
 - Google DNS
 8.8.8.8
 8.8.4.4

8.8.8.8



Munin

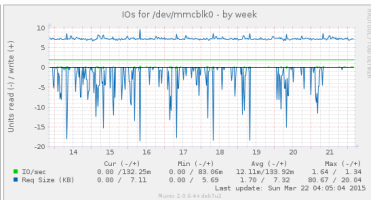
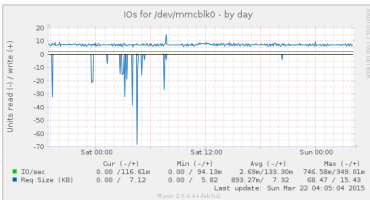
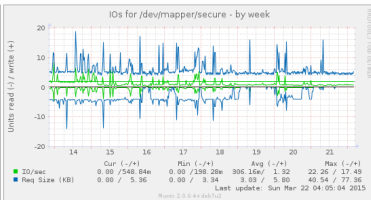
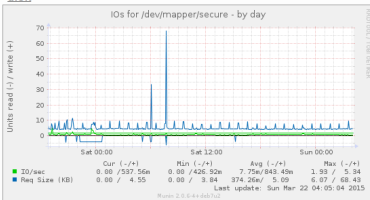
- generische Messwerterfassung und Graphenerzeugung
- Konzept:
 - Munin-Server für zentrale Datensammlung & Graphenerzeugung
 - Munin-Client (`munin-node`) für lokale Datenerfassung
 - 1 ... n Datensets/Graph (z.B: Füllstand aller Dateisysteme)
- Datenerfassung per Plugin; Plugin-Sammlungen existieren
- Erweiterbar: Plugin-API relativ einfach
- `apt-get install munin munin-node`

Munin, Konfiguration

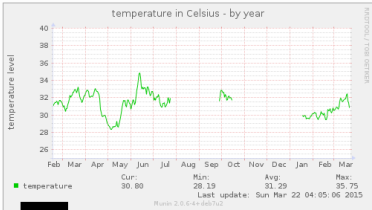
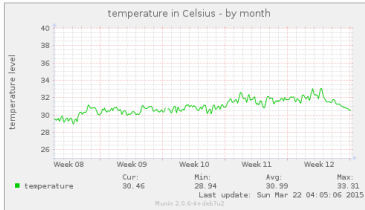
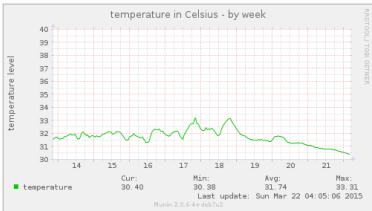
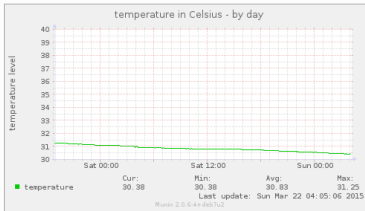
- Munin-Client wird von Distribution bei Installation automatisch konfiguriert
- weitere Sensor-Plugins einfach aktivierbar per Symlink von `/usr/share/munin/plugins/$PLUGIN` nach `/etc/munin/plugins/`
- Konfiguration auf Serverseite:
[hostname]
address 10.1.2.3
use_node_name yes

Munin, Graphen 1/2

disk



Munin, Graphen 2/2



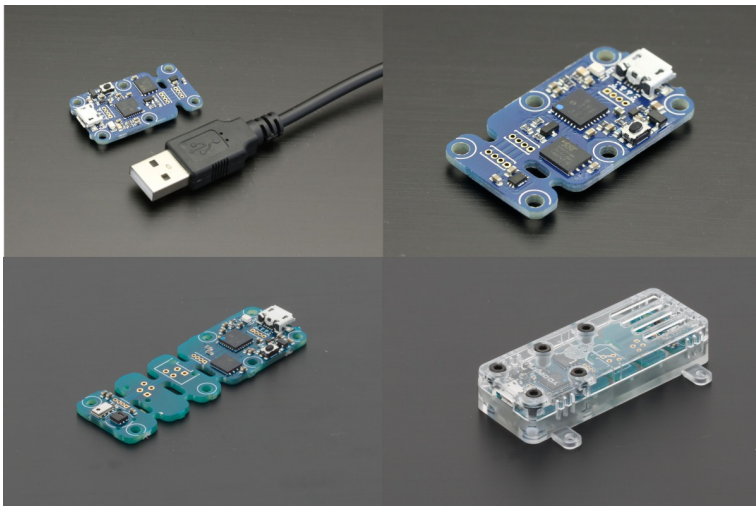
Sensoren

- Überwachung nicht nur von IT-Systemen, sondern Umgebung
- zahlreiche Möglichkeiten:
 - OneWire Sensoren
 - via GPIO
 - mit USB Interface
 - fertige Sensormodule mit Ethernet Interface
 - direkt (z.B. via GPIO Pins) angesprochene Sensoren
 - Sensoren mit USB Interface
 - Sensoren via I2C angebunden
- fertige industrielle Lösung meist teuer (typ. >\$200)
- typische Billiglösungen meist mit viel Bastelarbeit

Yoctopuce

- Sensormodule/-netzwerke mit USB Interface
- Userspace Treiber für x86 & ARM (libusb)
- keine Industriepreise
- verschiedene Sensoren
 - Lichtsensor, liefert Helligkeit in Lux (CHF 25)
 - Meteosensor, Temperatur, Luftfeuchte, Druck (CHF 65)
 - Elektro-Sensoren (V, A, mV), ...
- verschiedene Aktoren
 - Relais
 - Servosteuerung
 - kleine OLED displays
- <http://www.yoctopuce.com/>

Yoctopuce Sensorbeispiel



Weitere Sensoren

- `lm_sensors`: Hardwaremonitoring im Computer
- OneWire-Sensoren, z.B. DS1820 Temperatursensor
- USB-Sensors
<https://code.google.com/p/usb-sensors-linux/>
- <http://www.sparkfun.com> viele Sensoren, oft I2C

Zusammenfassung

- Monitoring @ \$HOME machbar
- sowohl Graphen als auch Alarmierung
- mit moderatem Aufwand
- Umgebungsmonitoring zu zivilen Preisen ebenfalls

Fragen?

Vielen Dank für Euer Interesse!