

Inoffizielles Cacti - Handbuch

Stand 10.09.2009

Grundlagen zum Verständnis von Cacti

1. Überblick zu Cacti/rrdtool/snmp
2. *Installation von Cacti*
3. Verwendung des passenden Pollers
4. Allgemeine Konfiguration wie Benutzerverwaltung, Rechte, Poller, Pfade, usw.
5. *Grundlagen und Beispiele für snmp*
6. *Grundlagen und Beispiele für rrdtool, insbesondere die Konsolidierung*
7. Die Beziehung zwischen rrdtool-Kommandos und Cacti-Objekten

Praxis

8. Methoden um Daten zu sammeln (Data Queries, Data Input Methods)
9. Das Mapping zwischen snmp/Skripten und Cacti: die XML-Datei
10. *Die Cacti-Objekte im einzelnen: Devices, Data Sources, Graphen, usw.*
 - 10.1 **Data Templates mit "Get SNMP Data"**
(Data Template: einzelne OID per snmp)
 - 10.2 **Von der eigenen Data Input Method zum Graphen**
(einzelner skalarer Wert von einem Skript
Data Template: die neue Data Input Method)
 - 10.3 **Vom eigenen snmp Data Query zum Graphen**
(Data Template: Get SNMP data (indexed))
 - 10.4 **Vom eigenen script Data Query zum Graphen**
(Data Template: Get Script Data (indexed))
 - 10.5 **PHP script server Query**
(Data Query: Get Script Server Data (Indexed))
11. *Templates im allgemeinen, Import/Export*
12. *Weitergabe von Änderungen an Templates zu den konkreten Instanzen*
13. *Konkrete Abfragen: Advanced Ping, Prüfung ob TCP- und UDP-Ports offen sind*
14. *Fertige Templates für VMware, Microsoft, usw.*
15. *Anpassung, Darstellung und Anordnung der Graphen*

Debugging

16. *Fehleranalyse*
17. Online-Ressourcen: Xing-Gruppen, Cacti-Forum, Plugins

Advanced

18. *Einige interessante Plugins: Realtime, Thold (Notifications), Discovery, Weathermap*
19. Abschließende Bemerkungen
20. Ausblick
21. Skript-Vorlagen
22. Issues

2, 5, 6, 10-16, 18: Mit Beispielen zum Ausprobieren

1. Überblick zu Cacti/rrdtool/snmp

Cacti dient zum Monitoring von einzelnen Geräten oder ganzen Netzwerken.

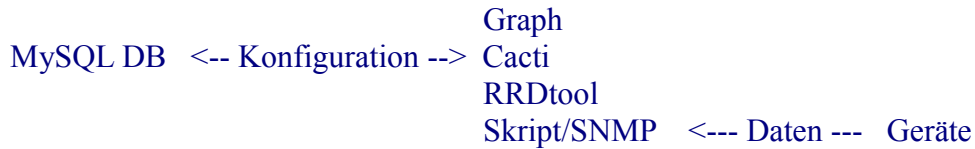
Mit Cacti schaut man WIE etwas läuft, mit Nagios OB es läuft.

Cacti verwendet rrdtool, um Daten über bestimmte Zeiträume graphisch darzustellen.

Daten werden mit Hilfe des Cacti-Pollers per snmp oder Skript geholt und in rrdtool abgelegt.

Die Graphen in Cacti werden dann mit Hilfe dieser Daten erzeugt und angezeigt.

Gesamtüberblick



Wie Werte in Cacti hinein gelangen

Skalare Werte: Data Input Methods: per snmp, Skript (oder PHP Skript Server)

Indexbasierte Werte: Data Queries: per snmp, Skript (oder PHP Skript Server)

Wie Werte in rrdtool abgespeichert werden

| | | | |
|---------|---------|------------------------|--|
| 5 Min. | 1. Wert | | |
| 10 Min. | 2. Wert | - CF.z.B. Durchschnitt | => konsolidierter Wert für den ganzen Zeitraum |
| 15 Min. | 3. Wert | | |

2. Installation von Cacti

NTP einrichten !!!

Cacti-Version 0.8.7e, mit Spine als Poller auf SLES
gemäß http://docs.cacti.net/manual:087:1_installation.1_install_unix

Software-Requirements:

apache2
mysql
php5
apache2-mod_php5
php5-mysql
php5-snmp
php5-xmlreader
php5-xmlrpc
php5-sockets
php5-ldap (nur falls LDAP-Authentifizierung benötigt wird)
rrdtool
net-snmp
autoconf, automake, mysql-devel, net-snmp-devel (für die Spine-Installation)

PHP-Konfiguration

Der Extension-Pfad in /etc/php5/apache2/php.ini ist bereits korrekt auf extension_dir = /usr/lib/php5/extensions gesetzt.

Die MySQL-Extension für PHP ist bereits in /etc/php5/conf.d/mysql.ini aktiviert.

Die SNMP-Extension für PHP ist bereits in /etc/php5/conf.d/snmp.ini aktiviert.

Templates können importiert werden, da bereits file_uploads = On in /etc/php5/apache2/php.ini gesetzt ist.

Weiterhin muss safe_mode = Off gesetzt sein, was ebenfalls bereits der Fall ist.

Apache-Konfiguration

/etc/init.d/apache2 start

MySQL-Konfiguration

Root-Passwort setzen:
/etc/init.d/mysql start
mysqladmin -u root password [PASSWORT]

SNMP-Konfiguration (lokal zum Testen)

/etc/init.d/snmpd start

Systeminformationen:

```
snmpget -v1 -c public localhost sysDescr.0  
snmpget -v1 -c public remotehost sysDescr.0
```

Gemountete Dateisysteme:

Man muss die auszugebenden Dateisystem einzeln angeben in snmpd.conf, da includeAllDisks nicht unterstützt wird.

```
disk /  
disk /home
```

...

```
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.9.1[.2]  
df
```

Netzwerkinterfaces:

```
snmpwalk -v 1 -c public localhost .1.3.6.1.2.1.2.2.1[.2]
```

Manchmal ist es hilfreich in /etc/snmp/snmpd.conf auf den zu überwachenden Systemen die Einheiten nicht mit ausgeben zu lassen, sodass man nur Werte erhält und nicht später mühsam mit einer Regex die Einheit wieder abschneiden muss:

```
dontPrintUnits true
```

```
/etc/init.d/snmpd restart
```

Erweiterung von SNMP durch eigene Skripte

```
/root/foo.sh:  
#/bin/bash  
echo "123"
```

```
/etc/snmp/snmpd.conf:  
exec foo /bin/bash /root/foo.sh
```

```
/etc/init.d/snmpd restart
```

```
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.8.1  
(Es gibt noch ein Beispielskript "echotest" ca. in der Mitte der snmpd.conf, daher 2 Einträge)  
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.8.1.101.1  
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.8.1.101.2
```

Cacti-Konfiguration

```
useradd cactiuser -p cactiuser
```

```
wget http://www.cacti.net/downloads/cacti-0.8.7e.tar.gz  
tar xfv cacti-0.8.7e.tar.gz
```

Cacti-Datenbank erstellen

```
mysqladmin -u root -p create cacti  
cd cacti-0.8.7e  
mysql -uroot -p cacti < cacti.sql  
Aufruf der MySQL-Shell:
```

```
mysql -uroot -p
mysql> grant all on cacti.* to 'cactiuser'@'localhost' identified by 'cactiuser';
(Ein flush privileges ist nicht notwendig, nur bei update der mysql.user-Tabelle)
```

```
include/config.php ergänzen falls notwendig
chown -R cactiuser rra/ log/
/etc/crontab:
*/5 * * * * cactiuser php /srv/www/htdocs/cacti/poller.php > /dev/null 2>&1
Prüfen, dass cron läuft
cd ..
mv cacti-0.8.7e /srv/www/htdocs/cacti
```

Die beteiligten Dienste in deren Standardrunlevels aktivieren

```
chkconfig apache2 on
chkconfig mysql on
chkconfig snmpd on
chkconfig -l apache2 mysql snmpd
```

Spine-Konfiguration (schneller als der Standard PHP-Poller cmd.php, da in C geschrieben)

```
wget http://www.cacti.net/downloads/spine/cacti-spine-0.8.7e.tar.gz
tar xfv cacti-spine-0.8.7e.tar.gz
cd cacti-spine-0.8.7e
Den Source entsprechend http://cacti.net/spine_download_patches.php patchen.
./bootstrap
./configure
make
make install
ln -s /usr/local/spine/etc/spine.conf /etc/spine.conf
/etc/spine.conf falls notwendig konfigurieren
```

Letzte Aktualisierungen installieren

Auf http://www.cacti.net/download_patches.php NACHSCHAUEN und entsprechend installieren.

```
cd /srv/www/htdocs/cacti
```

...

Patch-Dateien anschließend entfernen

```
ln -s /usr/bin/php5 /usr/bin/php
```

```
http://CACTIHOST/cacti
```

Achtung: richtige rrdtool-Version auswählen! (Bei Bedarf rrdtool manuell in der neuesten Version installieren)

Beim ersten Einlogversuch, kann man ein Passwort für den admin-Benutzer in Cacti setzen.

Dann als erstes in "System Utilities" -> "Technical Support" schauen ob etwas rot ist, z.B. das PHP memory_limit

Hier zum Üben für ein schnelles Restore, die Datenbank-Binaries und den Cacti-Ordner wegsichern (vorher sync ausführen und den MySQL-Dienst herunterfahren).

3. Verwendung des passenden Pollers

cmd.php, ursprünglicher Poller: PHP, langsam, stabiler
spine, neuerer Poller: C, schnell, aber unter Umständen problematisch

Zum Debuggen immer cmd.php nehmen.

Allerdings ist der Source von Spine sehr kurz, sodass man ohne weiteres auch direkt in den Source schauen kann bei Problemen.

4. Allgemeine Konfiguration wie Benutzerverwaltung, Rechte, Poller, Pfade, usw.

Sensibilisierung für die Darstellung der Werte in Cacti an Hand des mitgelieferten "Graph Template: Unix - Load Average". Im Graphen den Abschnitt für genau einen Step auswählen, Vergleich mit Legende -> stacked !!

Settings -> General:
Poller Logging Level: Debug
SNMP-Konfiguration

Settings -> Paths:
RRDTool Default Font Path:/usr/share/rrdtool/fonts/DejaVuSansMono-Roman.ttf
Spine Poller File Path:/usr/local/spine/bin/spine

Settings -> Poller:
Poller Type: spine
Downed Host Detection: SNMP
Ping Type: UDP
(Mit ICMP Ping kommt es in der Regel zu Problemen, wegen mangelnder Rechte.)

Settings -> Graph Export:
Hier kann generierte Bilder automatisch exportieren lassen.

Settings -> Visuals:
Parameter für die Fonts und generierten Graphen

Settings -> Authentication:
LDAP-Konfiguration, für LDAP muss php5-ldap installiert sein.
Spezialbenutzer, z.B. Gastbenutzer, um Graphen anzusehen

System Utilities: PRÜFEN, DASS NICHTS ROT IST.

User Management
Zuerst dem Gastbenutzer noch ein Passwort geben.

Realms:
Welche Bereich der Benutzer betreten darf

Graph Permissions:
Tree Permissions bedeutet nur, ob man in der linken Spalte durch den Baum navigieren kann und welche Einträge dort somit sichtbar sind. Es hat nichts mit den Graphen auf der rechten Seite zu tun. Ist nur der Tree erlaubt, sieht man rechts dementsprechend keinerlei Graphen.

Man sollte die Default Policies in allen vier Bereichen pro Benutzer immer gleich lassen.
Geänderte Rechte werden erst mit dem nächsten Login des Benutzers wirksam.

Am besten so vorgehen:
Alle Default Policies pro Benutzer auf allow setzen.

Dann pro Tree sperren, pro Device sperren, dann pro Graph Template und schließlich pro Graph.
Das funktioniert auch wie erwartet.

Graph Settings:

Wie Graphen für diesen Benutzer angezeigt werden, hier geht es also nicht um Rechte.

Das sind übrigens die Einstellungen, die der Benutzer erreicht wenn er im "graphs"-Tab rechts oben auf Settings klickt, sofern er die Berechtigung dazu hat.

Ändern der Poller-Frequenz auf eine Minute

Den Eintrag in der /etc/crontab ändern auf:

```
* * * * * cactiuser php /srv/www/htdocs/cacti/poller.php > /dev/null 2>&1
```

Im Bereich Settings das Poller- und Cron-Interval auf Every Minute setzen.

Falls notwendig weitere RRAs anlegen.

Die gewünschten RRAs in den Data Templates aktivieren und den Step auf 60 Sekunden setzen.

Den Heartbeat noch anpassen und auf 120 setzen.

5. Grundlagen und Beispiele für snmp

Eine ausführlichere snmpd.conf verwenden:

```
cp /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf_orig  
cp /usr/share/doc/packages/net-snmp/EXAMPLE.conf /etc/snmp/snmpd.conf
```

Darin folgendes anpassen:

```
#   sec.name source      community  
com2sec local  localhost  public  
com2sec mynetwork 192.168.1.0/24 public
```

```
/etc/init.d/snmpd restart
```

Die geladenen MIBs finden sich in den folgenden Verzeichnissen.

```
$HOME/.snmp/mibs  
/usr/share/snmp/mibs/
```

Beispielabfragen von der Shell:

Ausgabe aller OIDs in Textform als Baum.

```
snmptranslate -Of -Tp | less
```

Ausgabe aller OIDs in Textform und numerisch.

```
snmptranslate -Of -Tl | less
```

```
snmpwalk -v1 -c public localhost interfaces
```

```
snmpwalk -v1 -c public localhost system
```

```
snmpwalk -v1 -c public localhost ipAddrTable
```

Abfrage eines einzelnen Wertes mit Angabe der MIB. Hier die Uptime des Systems

```
snmpwalk -v1 -c public localhost host.hrSystem.hrSystemUptime.0
```

Im folgenden Beispiel handelt es sich um die Uptime des snmpd, nicht um die Uptime des hosts.

```
snmpget -v1 -c public localhost sysUpTime.0
```

```
/etc/init.d/snmpd restart
```

```
snmpget -v1 -c public localhost sysUpTime.0
```

Um die numerische OID zu bekommen:

```
snmpget -v1 -On -c public localhost sysUpTime.0
```

Ausgabe in gewohnter Tabellenform, zeilenweise, mit Tabelleindex.

```
snmptable -v1 -c public -Os -Cw 200 -Ci localhost sysORTable
```

Blick in eine MIB am Beispiel InOctets:

```
snmpwalk -v1 -c public localhost interfaces | grep InOctets
```

```
vi /usr/share/snmp/mibs/IF-MIB.txt
```

Mehr Infos unter

<http://net-snmp.sourceforge.net/wiki/index.php/Tutorials>

<http://www.oid-info.com/>

<http://www.it-defender.com/content/view/231/29/>

6. Grundlagen und Beispiele für rrdtool, insbesondere die Konsolidierung

rrdtool dient zur Speicherung und grafischen Darstellung von Daten über Zeiträume, es generiert Grafiken. Aufruf über die Kommandozeile. Cacti ist ein Front-End für rrdtool!

Round-Robin-Database:

Es werden Dateien (.rrd) zum Speichern der Daten abgelegt. Diese Dateien wachsen NICHT mit der Zeit sondern haben von Anfang an eine unveränderliche Größe. Wenn die Datei mit Daten vollgeschrieben ist, wird am Anfang wieder begonnen und alte Daten überschrieben, so bleibt der Speicherverbrauch konstant (im Gegensatz zu herkömmlichen Datenbanken, die meist nur wachsen).

Allgemein werden Daten so für bestimmte festgelegte Zeiträume abgelegt, z.B. Tag, Woche, Monat und Jahr. Im Allgemeinen sind die Daten weniger "dicht" je länger der Zeitraum ist.

Daher ist der Speicherverbrauch für das Jahr nicht zwingend größer als der für z.B. für einen Tag.

Weitere Besonderheiten:

Es ist möglich statt der eigentlichen Werte, nur die Änderungsrate (Derive) zwischen ihnen abzuspeichern. rrdtool benötigt Daten in festgelegten Zeitintervallen. Stehen die Daten zu diesen Zeitpunkten nicht in ausreichendem Maße zur Verfügung wird UNKNOWN eingetragen. (Erscheint in den Logs als 'U' und im Graph als Lücke).

Beispiele (von oss.oetiker.ch):

Erstellung der RRD-Datei

```
rrdtool create target.rrd \  
  --start 1023654125 \  
  --step 300 \  
  DS:mem:GAUGE:600:0:671744 \  
  RRA:AVERAGE:0.5:12:24 \  
  RRA:AVERAGE:0.5:288:31
```

rrdtool create target.rrd

--start 1023654125

Sekunden seit 01.01.1970, der Zeitpunkt für ein Update der Daten MUSS später sein

--step 300

Es werden neue Werte alle 300 Sekunden (5 Minuten) erwartet. Es gibt also alle 5 Min. einen PDP.

DS:mem:GAUGE:600:0:671744

Data Source definiert eine Variable (hier mem) unter der die Daten abgelegt werden. Es kann mehrere DS geben.

GAUGE ist einer der möglichen Data Source Types (DST):

GAUGE: Speichert den tatsächlichen Wert.

COUNTER: Speichert die ÄNDERUNGSRATE in SEKUNDEN, der Wert muss sich immer erhöhen

DERIVE: Das gleiche wie COUNTER, erlaubt aber auch negative Werte, das heist der Wert darf auch abnehmen.

ABSOLUTE: Speichert die ÄNDERUNGSRATE als absolute Werte. Er geht davon aus, dass der

Zähler bei jedem Step zurückgesetzt wurde, das bedeutet dass er davon ausgeht, dass der vorherige Wert 0 ist.

Heartbeat (hier 600, sollte ein Faktor von Step sein):

Alle 300 Sekunden wird ein PDP erwartet. Steht 300 Sekunden lang kein Wert zur Verfügung, wird nochmal 300 Sekunden gewartet = 600 Sekunden. Steht bis dahin auch kein Wert zur Verfügung wird für das ganze Intervall UNKNOWN abgespeichert. Es ist eine Aussage darüber wie lange AM STÜCK keine Daten zur Verfügung stehen dürfen. Egal wie lange das ganze Intervall ist, wenn 2 Steps am Stück keine Daten da sind, wird für das ganze Intervall UNKNOWN eingetragen.

Minimum, Maximum: Alle Werte, die nicht in diesem Bereich sind werden als UNKNOWN abgespeichert.

RRA:AVERAGE:0.5:12:24

Round-Robin-Archive. Die Consolidation Function (CF) (hier AVERAGE) beschreibt wie aus den PDPs ein CDP erstellt wird. Es gibt außerdem noch MINIMUM, MAXIMUM und LAST.

Der x-files factor (hier 0.5) gibt an wieviel UNKNOWN-Einträge ein Intervall (hier die 12*300 Sekunden) haben darf, sodass der konsolidierte Wert noch gültig bleibt: Anzahl UNKNOWNs geteilt durch die Anzahl der PDPs im Intervall. Der Wert liegt also zwischen 0 und 1. 0.5 bedeutet also, maximal die Hälfte der PDPs darf UNKNOWN sein, dann wird immer noch ein gültiger CDP berechnet und eingetragen.

Der nächste Wert (12) gibt an, dass aus 12 PDPs mit Hilfe der CF ein CDP berechnet und in diesem RRA abgespeichert wird. Es werden maximal 24 dieser CDPs in diesem RRA abgelegt. Dies bedeutet: Der Step ist 300 Sekunden (5 Min.) mal 12 = 1 Stunde. Es wird also der Durchschnitt aus den 12 5-Minuten-Werten gebildet und als Ein-Stundenwert abgespeichert. 24 solcher CDPs werden gespeichert, repräsentieren also einen Tag.

RRA:AVERAGE:0.5:288:31

288 Steps = 5 Min. x 288 = 24 Stunden. Es wird also der Durchschnitt von 288 PDPs gebildet und als Tageswert abgespeichert. Das ganze 31 mal, repräsentiert also einen Monat.

Eine RRD-Datei kann auch mehrere DS haben. In jedem RRA werden die Daten für alle DS abgespeichert.

Zum Verständnis des DST:

Values = 300, 600, 900, 1200 (PDPs)

Step = 300 seconds

GAUGE DS = 300, 600, 900, 1200

COUNTER DS = 1, 1, 1, 1

DERIVE DS = 1, 1, 1, 1

ABSOLUTE DS = 1, 2, 3, 4

zur Erinnerung Änderungsrate/Sekunde

"

"

Anwendungsbeispiele:

GAUGE: Temperatur

COUNTER: iflnOctets

DERIVE: Wie ändert sich die Rate von Menschen die einen Raum betreten

ABSOLUTE: Anzahl Meldungen seit dem letzten Update

Erstellung von Graphen

```
rrdtool graph mem1.png \  
  --start 920804400 --end 920808000 \  
  --vertical-label KB \  
  DEF:mymem=target.rrd:mem:AVERAGE \  
  CDEF:realmem=mymem,1000,* \  
  LINE2:realmem#FF0000
```

CDEF:realmem=mymem,1000,*

Definiert die Variable realmem, gebildet durch das RRA mem aus der Datei target.rrd. (Es gibt im Beispiel mehrere RRAs. Welche Daten letztlich geholt werden hängt von start/end ab)

LINE2:realmem#FF0000

Multipliziert mit 1000 (der Stern muss escaped werden) 2Pixel-starke Linie, repräsentiert die Variable realmem, Einheit Kilobytes, gezeichnet in Rot.

Welche dieser Kommandos Cacti ausführt, kann man sich in Cacti auch direkt anzeigen lassen.

Weiter nützliche Kommandos:

rrdtune: Änderung der Konfiguration im Header einer rrd-Datei

rrdupdate: Wert in rrdtool-Dateien schreiben

rrdresize: Die Anzahl von Zeilen (letzter Wert in der RRA-Zeile) ändern

rrdfirst: Gibt die Timestamp des ersten Wertes aus

rrdlast: Gibt die Timestamp des letzten Wertes aus

rrdinfo: Gibt die Headerinformation aus

rrdfetch: Liest Daten aus der rrd-Datei

rrddump, rrdrestore: rrd-Daten in XML exportieren und von XML wieder importieren

Mehr Infos unter:

http://docs.cacti.net/manual:087:8_rrdtool (Konsolidierung)

<http://oss.oetiker.ch/rrdtool/tut/rrd-beginners.en.html>

<http://oss.oetiker.ch/rrdtool/tut/rrdtutorial.en.html>

<http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html>

<http://oss.oetiker.ch/rrdtool/tut/cdeftutorial.en.html>

<http://oss.oetiker.ch/rrdtool/doc/index.en.html>

Übung anhand des ersten Beispiels auf <http://oss.oetiker.ch/rrdtool/tut/rrdtutorial.en.html>

Achtung DS ist ein Counter, d.h. er speichert die Änderungsrate in Sekunden!

rrdtool create ...

```
ls -lha test.rrd
```

```
-rw-r--r-- 1 root root 1008 2009-09-03 07:55
```

rrdtool update ...

```
ls -lha test.rrd
```

```
-rw-r--r-- 1 root root 1008 2009-09-03 07:56
```

rrdtool fetch ...

rrdtool graph ...

RRAs: 2 Stunden, 5 Stunden.

Die Anzahl Steps im ersten RRA ist 1, daher ist der konsolidierte Wert gleich dem Originalwert, da aus nur einem Wert kein Durchschnitt gebildet werden kann.

Er legt folgendes ab (Counter gibt die Differenz pro Sekunde an):

(aktueller Wert - voriger Wert) / (Zeitpunkt jetzt - voriger Zeitpunkt) , z.B.:
(12357 - 12345) / (920805000 - 920804700) = 12 / 300 = 0.04

4.0000000000e-02 = $4 * 10^{-2} = 0.04$

0.02

0

0

0.0333

...

Der erste Wert ist NaN, da die Änderungsrate gespeichert wird, er aber hier noch keinen vorhergehenden Wert hat. Der letzte Wert ist NaN, da die Zeit ausserhalb des Bereichs liegt wo Daten vorliegen.

40 m = 40 / 1000 = 0.04

```
cp speed.png /srv/www/htdocs/
```

<http://IP/speed.png>

7. Die Beziehung zwischen rrdtool-Kommandos und Cacti-Objekten

http://www.infosecprojects.net/en/linuxutorials/images/cacti_rrdtool.png

8. Methoden um Daten zu sammeln (Data Queries, Data Input Methods)

Siehe Diagramm 2 aus dem 1. Kapitel am Anfang zum Überblick.

Liste von Data Input Methods in Data Templates:
Die 4 vorgegebenen + alle zusätzlich erstellten.

Die einzig sinnvollen Input Typen von Data Input Methods:
Script/Command
Script - Script Server (PHP)
(die anderen sind Altlasten, laut Cacti-Projektmitglied)

Data Input Methods in Data Queries

alle, die zum Abruf von indexbasierten Daten verwendet werden:

Get Script Data (indexed)

Ausführung eines Skriptes zum Holen von index-basierten Daten

Get Script Server Data (indexed)

Das gleiche, nur wird der Skriptaufruf vom Cacti-Skript-Server durchgeführt, statt eines normalen PHP-Aufrufes (ca. 20x schneller, da der PHP-Interpreter nur einmal aufgerufen wird und wiederholte Skriptaufrufe der Code nur einmal interpretiert wird.) Jedoch sind dafür Änderungen im Skript, der XML-Datei und im Data Query/Data Template notwendig.

Get SNMP Data (indexed)

Es wird SNMP direkt von Cacti aus verwendet für index-basierte Daten (snmpwalk). Dafür ist kein Skript erforderlich, lediglich die XML-Datei.

Weiterhin erscheinen alle Data Input Methods, die als Input Type "Script - Script Server (PHP)" haben.

Hinweise zu Data Input Methods

In das Feld "Input String" gehört im Fall des Typs Script/Command der ganze Pfad zum Skript sowie Parameter, falls man welche übergeben möchte, z.B. "<path_cacti>/scripts/diskfree.sh <partition>". Die Parameter, die man in dieser Zeile anfügt können nun anschließend im Bereich Input Fields -> Add ausgewählt und somit verwendet werden.

Ähnlich müssen für die Rückgabewerte, die das Skript zurückgibt im Bereich "Output Fields" Einträge angelegt werden.

Wenn man nun ein Data Template erstellt, muss man die gewünschte Data Input Method auswählen. Sobald man nun ein neues Data Source Item diesem Template hinzufügt erhält man im Feld "Output Field" genau die Einträge zur Auswahl, die in der Data Input Method als "Output Fields" definiert sind.

Hinweise zu Data Queries

Jedem Data Query ist eine XML-Datei zugeordnet sowie eine Data Input Method. Des weiteren muss jedem Data Query, mit dem man tatsächlich Graphen erstellen will mindestens ein Graph Template zugeordnet sein.

Die XML-Datei dient grob vereinfacht dazu die Daten, die in index-basierter Form von einer SNMP-Abfrage oder einem Skript kommen in Cacti zur Auswahl bereitzustellen. Dabei wird unterschieden zwischen Daten, die man bei der Graphenerstellung auswählen kann, wie z.B. die gemounteten Partitionen und den eigentlichen Werten. Die Unterscheidung wird durch das Tag "direction" in der XML-Datei erreicht.

Wenn man einem Data Query ein Graph Template zuweist (Associated Graph Templates -> Add), gibt man diesem nochmal einen extra Namen. Daher tauchen die Namen der Graph Templates, die man im Data Query sieht auch nicht zwingend in der eigentlichen Liste der Graph Templates auf. In der zweiten Spalte jedoch ist der eigentliche Name des Graph Templates mit angegeben.

Data Queries -> Associated Graph Templates

Erst wenn man hier ein Graph Template zuordnet wird auch ein Data Template mit Data Source und Suggested Values zugeordnet.

Data Queries -> Associated Data Templates

Erstmal kann man nur Graph Templates einem Data Query direkt zuordnen. Hat man dies jedoch getan und wählt man das Graph Template aus so sieht man Felder, die zum zugehörigen Data Template gehören. Wo kommen diese Felder her?
Über das Graph Template, bzw. dessen Graph Template Items wird das Data Template und damit dessen Data Source zugeordnet.

Data Queries -> Associated Data Templates -> Data Source

In dieser Zeile findet man den Namen der Data Source aus dem Data Template sowie eine Drop-down-Liste, die genau die Output-Felder der XML-Datei des Data Queries enthält.

Data Queries -> Suggested Values (Graph Template)

Das was hier steht bezieht sich auf die direkten Eigenschaften des Graph Templates, z.B. title. Es dient zur Unterscheidung von Graphen, die zum gleichen Host gehören.

Data Queries -> Suggested Values (Data Template)

Das was hier steht bezieht sich entweder auf die direkten Eigenschaften des Data Templates, z.B. name, oder auf die Eigenschaften des Data Source Items der Data Source des Data Templates, z.B. rrd_maximum. Das ganze dient dazu die verschiedene Data Sources die zum gleichen Host gehören zu unterscheiden.

Data Queries -> Data Input Method

Hier sind nur die index-basierten Data Input Methods zur Auswahl, da mit Data Queries nur index-basierte Daten geholt werden:

Achtung:

Cacti kann Abfrageergebnisse in Ziffern (Strings) nur selbstständig in numerische Werte in Cacti umwandeln wenn es sich um ganzzahlige Werte handelt, bei Gleitkommazahlen geht es nicht. Dies kann man dann z.B. in einem PHP-Skript mit der Funktion intval machen:

<http://www.php.net/manual/en/function.intval.php>

9. Das Mapping zwischen snmp/Skripten und Cacti: die XML-Datei

Die XML-Datei dient zum Mapping zwischen snmp/Skript und Cacti für indexbasierte Daten.

Ein Beispiel für ein SNMP Data Query:

(http://docs.cacti.net/manual:087:3a_advanced_topics.3a_snmp_data_queries)

10. Die Cacti-Objekte im einzelnen: Devices, Data Sources, Graphen, usw.

Ein neues Device bzw. neuen Host anlegen

Benötigte Informationen:

Beschreibung

IP oder Hostname

Als Hosttemplate "ucd/net SNMP Host" verwenden

Welche Methode soll verwendet werden, um die Verfügbarkeit zu prüfen?

SNMP-Konfiguration

Anschließend

Woran sieht man, ob der Host UP ist und ob der SNMP-Zugriff funktioniert?

Welche Aussagen lassen sich über das verwendete Host-Template machen?

Graphen für diesen Host anlegen

für CPU, load und memory

snmp stats für ein Netzwerkinterface in der Einheit In/Out Bytes

Woran sieht man direkt ob die Erstellung der Graphen erfolgreich war?

Das Ergebnis auch noch einmal unter "Graph Management" prüfen.

Erzeugte Graphen in den Graph Tree einfügen

Im Abschnitt "Devices" alle Graphen in den Tree einfügen.

Das Ergebnis im Tab "graphs" anschauen.

Im Bereich "New Graphs" für diesen Host einen weiteren Graphen erstellen.

Das Ergebnis auch noch einmal unter "Graph Management" prüfen.

Im Tab "graphs" ist zu sehen, dass der neue Graph automatisch eingefügt wurde.

Im Bereich "Graph Trees" den Eintrag für den "Default Tree" ansehen.

Einen weiteren Tree mit dem Namen "DC" erstellen.

Den Tree Item Type auf "Host" setzen und anschließend den Host auswählen.

Und nun den ersten Eintrag im Tree wieder entfernen, sodass der Host nur noch im Tree DC angezeigt wird.

Das Ergebnis auch noch einmal unter "Graph Management" prüfen.

Welche rrd-Dateien wurde für diesen Host von Cacti erstellt und wo liegen sie?

Siehe dazu im Bereich "Data Sources".

Welches rrdtool-Kommando wurde jeweils verwendet?

Welche Round-Robin-Archives sind in Cacti vordefiniert?

Welche davon enthalten eine tatsächliche Konsolidierung der Daten?

Den Aufbau von Graph Templates anschauen.

10.1 Data Templates mit "Get SNMP Data"

Am Beispiel des Data Templates "Host MIB - Logged in Users".
Hier wird eine OID direkt im Data Template eingetragen.
(snmpget -v1 -c public localhost .1.3.6.1.2.1.25.1.5.0)

Übung mit der numerischen OID der Uptime des snmd

10.2 Von der eigenen Data Input Method zum Graphen

Beispiel an Hand eines Skriptes für einen skalaren Wert als Data Input Method:

Erzeuge eine neue Data Input Method mit dem Namen Temperature

die die Rückgabewerte eines einfachen Shell-Skriptes entgegen nimmt.

Das Skript am besten in "<path_cacti>/scripts/test.sh" abgelegt .

```
#!/bin/sh
```

```
printf "123"
```

Es sind auch mehrere Rückgabewerte in einem String möglich, z.B.

```
printf "temp1:123 temp2:104"
```

Die Data Input Method soll die Werte unter dem Namen "temp" an Cacti weiterreichen (Output Field). Für das Beispiel mit mehreren Werten, müssen dann entsprechend 2 Outputfields hinzugefügt werden. Dabei ist wichtig dass die Namen exakt dem Rückgabestring entsprechen (hier temp1 und temp2).

Entweder

```
chown cactiuser test.sh
```

```
chmod u+x test.sh
```

oder in der Data Input Method `"/bin/sh <path_cacti>/scripts/test.sh"` eingeben.

Erzeuge ein neues Data Template mit dem Namen Temperature, welches diese Data Input Method benutzt

Data Source

Name (Häkchen nicht setzen): |host_description| - Temperature

Die richtige Data Input Method auswählen

Internal Data Source Name: temp

(Siehe auch http://www.infosecprojects.net/en/linuxutorials/images/cacti_rrdtool.png)

Data Source Type: GAUGE

Erzeuge ein neues Graph Template mit dem Namen Temperature, welches dieses Data Template verwendet

Im Abschnitt Graph Template:

Name (Häkchen nicht setzen): |host_description| - Temperature

(sonst keine Änderungen)

Füge nun ein Graph Template Item hinzu, um die Werte darzustellen

Die richtige Data Source auswählen

Color: FFF200

Graph Item Type: AREA

CF: MAX

Text Format: Temperature in our Data Center

Den automatisch erzeugten neuen Graph Item Input anschauen

Füge ein weiteres Graph Template Item hinzu, um die Legende darzustellen

Die richtige Data Source auswählen

Graph Item Type: LEGEND

Den neuen Graph Item Input nochmal anschauen und die Verknüpfungen mit den anderen Graph Template Items (Legende) beachten.

Hinweis:

Erst mit einem Graph Template werden tatsächlich rrd-dateien aus einem Data Template erzeugt, daher auch hier die Abfrage nach der Consolidation Function bei den Graph Template Items.

Falls man das möchte kann man dieses Graph Template nun einem bestehenden Host Template zuweisen oder auch dafür ein neues Host Template anlegen.

Wir legen ein neues Host Template an mit dem Namen "VMware Host"

und weisen ihm das neue Graph Template zu.

Dieses Host Template weisen wir nun dem ursprünglichen Host zu.

Dabei ist zu erkennen, dass die Graphen des vorherigen Host Templates (ucd/net SNMP Host) nicht verloren gehen, sondern der neue Temperatur-Graph einfach hinzugefügt wird :-)

Bei der anschließenden Erstellung des Graphen wird keine weitere individuelle Abfrage für den Data Source Name und den Graph Title gemacht, da wir die beiden Häkchen nicht aktiviert haben. Zum Ausprobieren, einfach im Data Template und im Graph Template die Häkchen aktivieren, den Graph und die Data Sources löschen und den Graph neu erstellen.

10.3 Vom eigenen snmp Data Query zum Graphen

Beispiel an Hand eines SNMP Data Queries. Hierbei wird kein Skript benötigt, da die Daten per snmp geholt werden. Allerdings muss man Cacti mitteilen, wie die Daten strukturiert sind, z.B. welche Tabellenspalten usw.

http://docs.cacti.net/manual:087:3a_advanced_topics.3a_snmp_data_queries#snmp_data_queries

<oid_index>:

```
snmpwalk -v1 -c public localhost .1.3.6.1.2.1.2.2.1.1
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
```

<oid_index_parse>:

Falls angegeben extrahiert aus den vom obigen Kommando zurückgelieferten OIDs die Indizes:

```
.1.3.6.1.2.1.2.2.1.1.1
.1.3.6.1.2.1.2.2.1.1.2
```

...

Extrahiert die hinteren beiden Segmente

(am Anfang der Regex ist ein gefrästiger Quantifizierer):

```
1.1
1.2
```

<oid_num_indexes>:

Falls man hier eine OID angibt wird sie verwendet um die Gesamtzahl der Einträge zu ermitteln:

```
snmpwalk -v1 -c public localhost .1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 5
```

<source>value</source>:

value bedeutet einfach das Ergebnis der Abfrage. Hier ist auch eine Regex möglich.

<direction>input</direction>:

Definiert eine Spalte die Informationen enthält, die in Cacti als Spaltenüberschrift bei diesem Data Query angezeigt werden sollen, z.B. der Name eines Netzwerkinterfaces.

<direction>output</direction>:

Definiert eine Spalte, die die eigentlichen Werte für die Graphen enthält, z.B. InOctets

"Successfully located XML file" bedeutet nur, dass Cacti die Datei gefunden hat.

Im Data Template wird als Data Input Method "Get SNMP data (indexed)" ausgewählt und entsprechende Data Source Items hinzugefügt. Wie aber wird eine Verbindung zwischen dem Data Template und dem Data Query hergestellt?

Antwort:

Über die Graph Template Items (Data Source) wird eine Verbindung zu den Data Sources im Data Template hergestellt, denn diese Data Sources können dort im Graph Template bereits ausgewählt werden.

Im Data Query kann man nun dieses Graph Template auswählen. Dabei sieht man auch das zugeordnete Data Template (denn im Graph Template wurden ja dessen Data Sources ausgewählt)

und erhält somit die Verbindung zwischen den Feldern der XML-Datei und den Data Sources des Data Templates.

Als Suggested Values kann man alle XML-Felder nehmen die auf "input" gesetzt sind, also keine Werte für die Graphen enthalten.

Im Beispiel für das Data Template (name) wird dann z.B. folgende rrd-Datei erzeugt:
localhost - Traffic - 192.168.1.35 - bond0

und der Titel für den Graph lautet entsprechend:
localhost - Traffic - bond0

Anschließend kann man das Data Query einem Device hinzufügen.

Alle Graph Templates, die auf Data Templates basieren, die als Data Input Method "get SNMP data (indexed)" eingestellt haben, benötigen eine entsprechende Data Query, da sie sonst zwar einem Host Template jedoch nicht einem Host zugeordnet werden können (woher sollen die Daten kommen?). Für viele der mitgelieferten Data Templates werden leider keine dazugehörigen Data Queries mitgeliefert.

Graph Templates, die auf einem solchen Data Template basieren können keinem Device zugeordnet werden. Es muss das entsprechende Data Query zugeordnet werden.

Das ganze nochmal etwas anders formuliert:

Graph Templates, die auf Data Sources von index-basierten Werten basieren, können einem Host nicht direkt zugewiesen werden. Man muss dem Host das entsprechende Data Query zuweisen und dann das gewünschte Graph Template (Graph Type) auswählen.

Warum ist das so?

Weil man über Data Queries mehrere "Datensätze" auf einmal bekommt und für jeden dieser Datensätze unter Umständen mehrere Graphen generieren möchte. Siehe z.B. Data Query "SNMP - Interface Statistiks".

Man kann die output-Spalten einer XML-Datei auch auf mehrerer Data Templates aufteilen. In diesem Fall erstellt man darin nur Data Source Items für die gewünschten Spalten.

Entsprechend kann man dann für jedes Data Template auch ein eigenes Graph Template erstellen, indem man darin nur die gewünschten Data Sources auswählt.

Was genau bedeutet die Re-Index Method?

Sie gibt an bei welchem Ereignis die Daten des Data Queries erneut eingelesen werden, sodass man neue Einträge in der Tabelle in Cacti angezeigt bekommt. Z.B. wenn ein zusätzliches NIC eingebaut wird und sich so der Index-Zählerstand (Index Count) ändert.

10.4 Vom eigenen script Data Query zum Graphen

Hierbei wird ein Skript verwendet um index-basierte Daten abzurufen, die nicht per snmp verfügbar sind oder fehlerhaft zurückgeliefert werden, aber z.B. per ssh oder nrpe abgerufen werden können.

Es funktioniert wie das snmp Data Query, nur dass das Skript die Werte zurückliefert, statt snmp und dass das Skript in der XML-Datei eingetragen sein muss (<script_path>).

http://docs.cacti.net/manual:087:3a_advanced_topics.3d_script_data_query_walkthrough#script_data_query_walkthrough

10.5 PHP script server Query

http://docs.cacti.net/manual:087:3a_advanced_topics.2_php_script_server#php_script_server

11. Templates im allgemeinen, Import/Export

Graph Templates

Graph Template Items definieren welche Data Sources auf dem Graph dargestellt werden sollen. Fügt man ein Graph Template Item einem Graph Template hinzu und wählt oben eine Data Source aus, so wird automatisch diese Data Source als Graph Item Input hinzugefügt.

Graph Item Inputs erlauben einem beim konkreten Erstellen von Graphen für bestimmte Graph Items abweichend vom Template Eingaben zu machen. Ein Beispiel: Man wählt als Graph Item Input eine Color und ordnet es dem Item1 zu (oder auch mehreren Items). Erstellt man nun für ein Device einen konkreten Graphen auf Basis dieses Graph Templates so erkennt man im oberen Bereich die vorherigen Graph Item Inputs wieder. Im Fall der Color hat man nun automatisch eine Farbliste zur Auswahl und kann bzw. muss sie für diesen speziellen Graph setzen.

Klickt man im Graph Template auf das Graph Template Item so sieht man im Bereich Color einen zusätzlichen Eintrag: [Field Not Templated]. Klickt man darauf landet man genau in der Konfiguration des entsprechenden Graph Item Inputs.

Die Drop-down-Liste Field Type bei den Graph Item Inputs enthält die möglichen Felder der Graph Template Items. Wenn man sich ein Graph Template Item anschaut, sieht man die Felder von oben nach unten: Data Source, Color, Opacity/Alpha Channel,...

Wenn man in einem Graph Template Item welches Werte im Graphen selbst darstellen soll, z.B. Line2 als Consolidation Function z.B. LAST auswählt, dann muss das natürlich vorher in den RRAs aktiviert sein. Cacti legt die rrd-Dateien gemäß der Konfiguration in Data Sources -> RRAs an. Per default ist dort aber nur AVERAGE und MAX aktiviert. Wenn man nun in einem Graph Template auf LAST zugreifen möchte sind die Daten dafür nicht vorhanden. Daher im Bereich Data Sources den Debug-Mode anschalten und schauen welche RRAs mit welcher Konsolidierung tatsächlich angelegt wurden.

Die Consolidation Function in einem GRPINT bedeutet aber etwas anderes!

Damit ist nicht die Konsolidierungsfunktion in den RRAs gemeint! Sondern hier wird konsolidiert in Bezug auf den Zeitraum, für den der Graph erstellt wird!

Z.B.:

```
/usr/bin/rrdtool graph - \
```

```
...
```

```
--start=1252570961 \
```

```
--end=1252574501 \
```

```
...
```

```
GPRINT:a:AVERAGE:"Current\:%8.2lf %s"
```

Host Templates

Hierbei handelt es sich um eine Zusammenstellung von Graph Templates und Data Queries, die dann als ganzes einem Device zugeordnet werden können. So kann man Devices der gleichen Gruppe immer das gleiche Host Template zuweisen und so die entsprechenden Graphen generieren lassen ohne sie jedesmal neu definieren zu müssen.

Ein Host Template zwingt einem Device keine feste Struktur auf. Man kann immer für ein Device individuell Graph Templates oder Data Queries hinzufügen oder entfernen.

Data Templates

Data Templates stellen ein Skelett für eine rrdtool create Anweisung dar. Die Data Source Items entsprechen den DS-Zeilen in der rrdtool create Anweisung. Die auswählbaren RRAs entsprechen den RRA-Zeilen in der rrdtool create Anweisung.

Wenn man ein neues Data Template erstellt, sollte man beim Namen der Data Source das Häkchen (Use Per-Data Source Value (Ignore this Value)) setzen und den Namen mit |host_description| - beginnen lassen. Sonst kann man die daraus abgeleiteten Data Sources (rrd-dateien) keinem Host zuordnen bzw. alle aus diesem Data Template erzeugten Data Sources hätten den gleichen Namen.

Ein Data Template kann mehrere Data Source Items enthalten. Dies ist notwendig in Fällen wo ein Skript Daten parallel anliefert oder eben bei Data Queries.

Data Template -> Output Field

Die Drop-down-Liste Output Field erscheint nur im Bereich Data Source Item wenn die Data Input Method auch entsprechende Felder definiert hat, siehe z.B. "Unix - Load Average".

Data Template -> Custom Data

Was hier erscheint ist abhängig von der Data Input Method. Es sind die Felder, die bei der Data Input Method als Input Fields definiert wurden. Ist dem Data Template z.B. eine Data Input Method zugewiesen die als Typ SNMP hat kann man hier unter anderm die OID, SNMP Version, etc. eingeben. Ist der Typ Script/Command kann man hier Werte an das Skript bzw. das Kommando übergeben, sofern dies in der Data Input Method definiert ist.

Hat man bei der Data Input Method in den Input Fields einen Special Type Code angegeben, z.B. snmp_community, so erscheint hier "Value will be derived from the host if this field is left empty". Dies bedeutet, dass der Wert für dieses Feld von den Daten des Hosts ermittelt und automatisch eingefügt wird, sofern man das Feld leer lässt.

Use Per-Data Source Value (Ignore this Value) sollte bei Data Source -> Name immer aktiviert sein, sodass jede Data Source, die auf diesem Template basiert ihren individuellen Namen hat. Generell kann man damit individuell pro Feld sagen ob es getemplatet wird(unchecked) oder nicht(checked).

Es geht beim Import/Export meist um Graph Templates und deren Abhängigkeiten.

Exportiere das Template "Unix - Logged in Users".

Löschen DANACH die zugehörige Data Input Method "Unix - Get Logged In Users".

Ändere anschließend manuell im exportierten Template die Höhe (Height) von 120 auf 200 und importiere das Template wieder.

[new]: das Element war nicht vorhanden und wurde neu angelegt.

[update]: das Element wurde überschrieben

Beim Import werden Templates und Objekte mit gleichem Namen ohne nachzufragen überschrieben.

Weitere Erläuterungen:

New Graphs

Hier kann man zu einem Device einzelne oder mehrere Graphen erstellen. (Nach dem Erstellen der Graphen braucht es in der Regel zwei Pollingzyklen bis die Graphen sichtbar werden, das sind ca. 10 Minuten standardmäßig.)

Erklärung zur Drop-down-Liste "Graph Types":

Über der Liste sieht man in Grau das dem Device zugeordnete Host Template. In der Liste selbst kann man nun auswählen welche Graph Typen zur Auswahl dargestellt werden. Der Inhalt dieser Liste basiert auf den Graphen und Data Queries des DEVICES, nicht auf denen des Host Templates, wie man vermuten könnte. Dies bedeutet, hat man dem Device individuell Graph Templates oder Data Queries hinzugefügt oder entfernt so erscheinen diese entsprechend auch hier, was nicht der Fall wäre, wenn nur die des Host Templates dargestellt würden.

All:

Zeigt alle dem Device zugeordneten Graph Templates und Data Queries an.

Graph Template Based:

Zeigt nur die Graph Templates an.

Restliche Einträge:

Die dem Device zugeordneten Data Queries.

Um das ganze zu verstehen, schaut man sich die Einträge der Drop-Down-Liste an und geht dann anschließend auf console -> Devices, wählt das Device aus und scrollt nach unten. Hier sieht man nun genau das, was auch bei New Graphs erscheint, wenn man in der Drop-Down-Liste "All" auswählt.

Um ein Data Query konkret zu verwenden, muss ihm immer mindestens ein Graph Template zugeordnet sein. Sonst kann man es zwar einem Device zuordnen, jedoch nicht zur Graphenerstellung verwenden.

Mit der Drop-down-Liste unter den Graph Templates hat man die Möglichkeit zusätzliche Graphen erzeugen zu lassen. Man wählt hier also nur etwas aus, wenn man Graphen haben möchte, die in der Liste darüber noch nicht vorhanden sind.

Zu der Drop-down-Liste unter den Data Queries (Select a graph type):

Sofern man ein Data Query anklickt, muss hier immer der richtige/gewünschte Graph Type dazu ausgewählt werden. Mit Graph Typ sind hier die Graph Templates gemeint, die dem Data Query zugewiesen wurden (Associated Graph Templates). Diese Namen tauchen somit nicht zwangsläufig in der generellen Liste der Graph Templates auf. Wurden bereits Data Queries zum ausgewählten Graph Type erstellt erscheinen sie grau hinterlegt, damit man nicht versehentlich Graphen doppelt erstellt. Möchte man nun aber zusätzlich einen anderen Graph für so ein bereits verwendetes Data Query erstellen, wählt man zuerst den gewünschten Graph Type in der Drop-down-Liste aus. Sofort lässt sich das Data Query wieder anklicken (sofern eben noch nicht verwendet für diesen Graph Typ).

Graph Management

--- CDEFs

--- Colors

--- GPRINT Presets

Hier werden wie zu erwarten die einzelnen konkreten Graphen aufgelistet. Außerdem kann man

hier Graphen individuell oder basierend auf einem Graph Template erstellen. Wählt man einen Graph aus so sieht man den Graph selbst sowie welches Graph Template für diesen Graph verwendet wurde und zu welchem Host der Graph gehört. Weiterhin sieht man ergänzende Informationen: Graph Item Fields: Die Data Sources, die die Daten für den Graph liefern. Graph Item Fields entspricht hier den Graph Item Inputs im Graph Template (siehe console -> Graph Templates, ein Graph Template auswählen).

Graph Management -> CDEFs

Mit CDEFs können die Daten, die für die Graph-Erstellung aus rrd-Dateien abgefragt werden verändert werden, sodass zum Beispiel im Graph statt bits/second bytes/second dargestellt werden. DEFs sind Variablen in rrdtool-graph-Kommandos, die die Werte zum gewählten Abfragezeitpunkt enthalten. CDEFs sind Beschreibungen von mathematischen Operationen, die auf die Werte von DEFs angewendet werden.

(Siehe auch cdeftutorial von Tobi Oetiker: <http://oss.oetiker.ch/rrdtool/tut/cdeftutorial.en.html>)

Hier kann man sich nun ganz einfach beliebige CDEFs zusammenklicken und sie anschließend bei den Graph Template Items in den Graphen oder Graph Templates verwenden.

Graph Management -> Colors

Hier kann man sich Farben zusammenstellen oder die vorhandenen ändern. Man gibt einfach den gewünschten HEX-Code ein und bekommt die Farbe dann dargestellt. Diese können dann für Graph Items verwendet werden, z.B. Linien, Legenden, Kommentare, etc.

Graph Management -> GPRINT Presets

Mit GPRINT Presets kann man Werte des Graphen in der Legende im Graph darstellen und zwar in gewünschter Formatierung (ähnlich printf). Ein Beispiel wäre das Graph Template "UNIX - Load Average". Klickt man es an sieht man oben die einzelnen Graph Template Items. Ungefähr in der Mitte der Tabelle ist eine Spalte mit dem Namen "Graph Item Type". Dort sieht man, dass Item 2, 4 und 6 vom Typ GPRINT sind. Diese 3 Items verwenden den GPRINT Type "Load Average". Sie bilden die Legende unterhalb des Graphen mit den aktuellen Werten der Load Average. Schaut man sich nun das GPRINT Preset "Load Average" direkt an sieht man den Namen und die Angabe zur Formatierung: %8.2lf (2 Nachkommastellen).

Graph Trees

Die Darstellung von Graphen auf dem graphs-Tab erfolgt in Form von Bäumen. Hier kann man Graphen an beliebiger Stelle in einen Baum einfügen oder entfernen. Es ist auch möglich mehrere Bäume zu erstellen.

Es stehen 3 Elemente zum Einfügen in den Baum zur Verfügung: Header, Host oder Graph.

Header: Bildet einen Unterbaum

Host: Fügt alle Graphen eines Hosts ein

Graph: Fügt nur einen einzigen Graph ein

Für die Darstellung der Baumelemente stehen verschiedene Sortierungsfunktionen zur Verfügung:

Manual Ordering (No Sorting)

Alphabetic Ordering

Natural Ordering

Numeric Ordering

[root] bedeutet root im ausgewählten Tree, nicht root gegenüber allen Trees.

Data Sources

--- RRAs

Data Sources entsprechen realen rrd-Dateien. Wählt man eine Data Source aus erfährt man welches Data Template für diese Datei verwendet wurde und zu welchem Host sie gehört. Weiterhin sieht man den exakten Pfad im Dateisystem zu dieser rrd-Datei. Man kann hier auch neue rrd-Dateien erstellen, indem man das gewünschte Data Template und den Host angibt.

Mit RRAs wird definiert wie Daten gespeichert werden sollen, z.B. ob der Durchschnitt oder das Maximum abgelegt werden soll, wieviele PDPs (Primary Data Points) für einen Consolidated Data Point (CDP) verwendet werden sollen, usw. Siehe rrdtool für Details.

Devices

Hier sind alle eingetragenen Hosts aufgelistet und man kann neue hinzufügen oder entfernen.

Möchte man einen Host hinzufügen, muss der Hostname oder die IP-Adresse eingegeben werden. Ordnet man dem Host ein Host Template zu, sieht man unten die dem Host Template zugehörigen Graph Templates und Data Queries und kann somit sehr einfach gleiche Graphen für Hosts, die das gleiche Host Template haben erstellen. Ordnet man kein Host Template zu bleiben die Listen für Graph Templates und Data Queries leer und man kann manuell welche hinzufügen. In beiden Fällen kann man immer individuell Graph Templates und Data Queries pro Host hinzufügen oder entfernen.

Möchte man die Re-Index Method für ein bereits eingetragenes Data Query ändern, muss man es wieder löschen und entsprechend mit anderer Re-Index Method wieder hinzufügen.

Schaut man sich das Host Template direkt an, welches man für den Host ausgewählt hat sieht man dort genau die unten aufgelisteten Graph Templates und Data Queries.

12. Weitergabe von Änderungen an Templates zu den konkreten Instanzen

Für einen beliebigen Graph aus dem Tab "graphs" in Graph Management nachschauen, auf welchem Graph Template er basiert.
In diesem Template dann für ein Graph Item die Darstellung ändern, z.B. von AREA auf LINE2.

Die Änderungen werden sofort an alle Graphen, die auf diesem Template basieren angewendet.

13. Konkrete Abfragen: Advanced Ping, Prüfung ob TCP- und UDP-Ports offen sind

Download:

<http://forums.cacti.net/about10049.html&highlight=advancedping>

Entpacken:

`unzip cacti_graph_template_ping_advanced_ping_v1_3.zip`

In Cacti das Graph Template zum Import auswählen:

`cacti_graph_template_ping_advanced_ping_v1_3.xml`

14. Fertige Templates für VMware, Microsoft, usw.

VMware ESX

<http://forums.cacti.net/about3730-0-asc-0.html>

Download der aktuellsten Version: <http://forums.cacti.net/about3730-0-asc-45.html>

"import the host template into cacti (i'm using 0.8.6d for this), drop the vmware_*.xml into your resource/script_queries directory and the php files into scripts directory."

IIS

Download:

<http://forums.cacti.net/about12464.html>

Docu,Skripts:

<http://forums.cacti.net/about12464-0-asc-15.html>

Weitere fertige Templates

<http://www.debianhelp.co.uk/cactitemplates.htm>

Forum für Skripte und Templates

<http://forums.cacti.net/forum-12.html>

15. Anpassung, Darstellung und Anordnung der Graphen

Titel eines einzelnen Graphen in Graph Management ändern.

Höhe und Breite von Graphen über ein Graph Template ändern, z.B. Höhe auf 220, Breite auf 1000.

Die Farbe eines Graph Template Items ändern.

Den Legenden-Text eines Graph Template Items ändern.

Beispiel für ein CDEFs: "Multiply by 1024"

Beispiel für ein GPRINT Preset: "Load Average"

Beim Ändern von Graph Template Items, wenn es Probleme gibt, den Graph bzw. die Graphen einzeln löschen und in Graph Management den Debug-Mode anschalten und schauen, ob alles ok ist.

Tab "graphs":

Settings, Tree View, List View, Preview View (Zeiträume)

16. Fehleranalyse

Die Reihenfolge, in der man nach dem Fehler suchen sollte.

Stehen die Daten auf dem zu überwachenden Gerät per snmp/Skript zur Verfügung?

Werden die rrd-Dateien angelegt, bzw. sind sie vorhanden?

Werden die Daten oder wird stattdessen 'U' in die rrd-Dateien geschrieben?

(Sind die Daten aus den rrd-Dateien abrufbar und wie erwartet?)

Sind die Einstellungen in Cacti passend zu den Werten, z.B. Minimum und Maximum?

System Utilities -> Rebuild Poller Cache

Spine-Debugging, für einen Host laufen lassen (Verbosity, first device id, last device id).

```
/usr/local/spine/bin/spine --verbosity=5 --first=1 --last=1
```

Croneintrag manuell laufen lassen (am besten als cactiuser, /etc/passwd)

```
php /srv/www/htdocs/cacti/poller.php --force --debug
```

```
/var/log/apache2/error_log
```

```
/srv/www/htdocs/cacti/log/cacti.log
```

Offizielle Anleitung zum Debugging

http://docs.cacti.net/manual:087:4_help.2_debugging#debugging

Tipp: Bei unklaren Parametern in Cacti, nicht in der Cacti-Dokumentation schauen, sondern in der Dokumentation von rrdtool.

Z.B. der slope-mode im Graph Template.

Dies führt zu einer glatteren Darstellung der Linien im Graph sofern man weit genug hinausgezoomt ist. Wenn man so nah ist, dass man die einzelnen Treppenstufen deutlich unterscheiden kann, wird nicht geklättet.

Das admin-Kennwort neusetzen:

```
mysql> use cacti;
```

```
mysql> select * from user_auth where username='admin' \G
```

```
mysql> update user_auth set password=md5('thepassword') where username='admin';
```

Weitere relevante Tabellen:

```
mysql> select * from host\G
```

```
mysql> select * from settings;
```

Wenn folgende oder ähnliche Meldungen im Graph Management kommen kommen:

RRDTool Command:

GRAPH ACCESS DENIED

RRDTool Says:

GRAPH ACCESS DENIED

Dann zuerst bei User Management schauen, ob der eingeloggte Benutzer die Graphen überhaupt sehen darf.

17. Online-Ressourcen: Xing-Gruppen, Cacti-Forum, Plugins

<http://forums.cacti.net/>

<http://www.cacti.net/downloads/>

<https://www.xing.com/net/cacti/> (Reinhard Scheck, Verantwortlicher für die Cacti-Dokumentation)

Plugins

<http://cactiusers.org/downloads/>

Eine ganze Appliance für Cacti, basierend auf CentOS mit vielen Plugins fertig installiert.

<http://cactiez.cactiusers.org/>

FAQ

http://docs.cacti.net/manual:087:4_help.1_faq#faq

Maintenance, z.B. logrotate, tägliche MySQL-dumps

http://docs.cacti.net/manual:087:4_help.3_maintenance

Variablen wie "|host_description|":

http://docs.cacti.net/manual:087:6_reference.variables#variables

Ältere Dokumentation

<http://www.cacti.net/downloads/docs/html/>

18. Einige interessante Plugins: Realtime, Thold (Notifications), Discovery, Weathermap

Zuerst die Plugin Architecture installieren.

Anleitung: <http://cactiusers.org/wiki/PluginArchitectureInstall>

Am besten direkt nach der Cacti-Installation machen, bevor man irgendwas in Cacti ändert.

```
Plugin Architecture downloaden
http://cactiusers.org/downloads/
wget http://mirror.cactiusers.org/downloads/plugins/cacti-plugin-0.8.7e-PA-v2.5.zip
unzip cacti-plugin-0.8.7e-PA-v2.5.zip
cp cacti-plugin-0.8.7e-PA-v2.5.diff /srv/www/htdocs/cacti/
cd /srv/www/htdocs/cacti/
patch -p1 -N --dry-run < cacti-plugin-0.8.7e-PA-v2.5.diff      (simulation)
Falls keine Fehler, dann:
patch -p1 -N < cacti-plugin-0.8.7e-PA-v2.5.diff
vi include/global.php
$config['url_path'] = "/cacti/";
```

Das "Plugin Management" aktivieren (steht in keiner Doku !!!)

```
mysql -uroot -p cacti < pa.sql
```

Jetzt muss noch in den Realm Permissions das Häkchen bei "Plugin Management" gemacht werden. Dann erscheint in console->Configuration ein neuer Link "Plugin Management". Manche Plugins können nur hierüber installiert werden, z.B. thold.

Jetzt kann man Plugins installieren,

über den "alten Weg":

Anleitung: <http://cactiusers.org/wiki/PluginsInstall>

oder über das Plugin Management

Beispiel "realtime"

```
wget http://cactiusers.org/downloads/realtime.tar.gz
tar xfvp realtime-0.32.tar.gz
mv realtime /srv/www/htdocs/cacti/plugins
vi /srv/www/htdocs/cacti/include/global.php
$plugins = array();
$plugins[] = 'realtime';
//$plugins[] = 'discovery';
...
```

Für jeden Benutzer (auch admin) muss das Recht für dieses Plugin im Bereich "User Management" gegeben werden.

Weiterhin muss für die meisten Plugins in Cacti noch etwas konfiguriert werden. Dies findet sich entweder unter Settings -> Misc oder unter einem eigenen Tab in Settings.

Für das Realtime-Plugin muss dort noch das Image-Cache-Directory gesetzt werden und für den Webserver-Benutzer schreibbar gemacht werden.

```
/srv/www/htdocs/cacti/plugins/realtime/cache
und auf der Shell angelegt werden:
cd /srv/www/htdocs/cacti/plugins/realtime
```

```
mkdir cache  
chown wwwrun cache  
chmod 755 cache
```

Jetzt findet man rechts neben jedem Graph ein kleines Symbol in Form eines "Zickzack-Graphen" mit dem man das Plugin ausführen kann.

Um ein Plugin zu deaktivieren reicht es die entsprechende \$plugins-Zeile in der global.php auszukommentieren. Um ein Plugin zu deinstallieren, diese Zeile entfernen und das Verzeichnis des Plugins unterhalb von /cacti/plugins/ löschen.

19. Abschließende Bemerkungen

Wenn möglich in Produktionsumgebungen alle Erweiterungen weglassen, wobei Templates weniger problematisch sind als Plugins.

Manche mitgelieferten Beispiel-Skripte funktionieren nur für localhost, obwohl das so nicht direkt sichtbar ist!

Reguläre Ausdrücke (innerhalb von Cacti) lassen sich nur für indizierte Daten in Cacti verwenden, da es nur dafür eine XML-Datei gibt. Möchte man dies für skalare Werte verwenden, muss man die Regex in ein eigenes Skript einbauen.

Das Hinzufügen von Data Source Items zu einem Data Template wirkt sich nur auf NEUE Data Sources (rrd-Dateien) aus, also solche, die nach dieser Änderung mit diesem Data Template erzeugt werden.

Die mit diesem Data Template erzeugten und bereits bestehenden Data Sources erhalten diese neue Data Source Items NICHT, da es in rrdtool keine Möglichkeit gibt zu bestehenden rrd-Dateien ein DS nachträglich hinzuzufügen.

Ordnerstruktur in Cacti:

```
/cacti/cli/  
/cacti/docs/  
/cacti/log/  
/cacti/ressource/script_queries/  
/cacti/ressource/script_server/  
/cacti/ressource/snmp_queries/  
/cacti/rra/  
/cacti/scripts/
```

Auf **Debian** ist der rrdtool default true type font nicht Bestandteil des rrdtool-Paketes, sondern des Paketes ttf-dejavu. Der Pfad ist:

```
/usr/share/fonts/truetype/ttf-dejavu/DejaVuSansMono.ttf
```

Inzwischen gibt es glücklicherweise auch ein richtiges Buch zu Cacti

http://www.amazon.de/Cacti-Network-Monitoring-Dinankur-Kundu/dp/1847195962/ref=sr_1_1?ie=UTF8&s=books-intl-de&qid=1252402485&sr=8-1

Und hier die englische Originalausgabe mit Beschreibung:

http://www.amazon.com/Cacti-Network-Monitoring-Ibrahim-Lavlu/dp/1847195962/ref=sr_1_7?ie=UTF8&s=books&qid=1252402518&sr=8-7

20. Ausblick

Zuordnung von mehreren IP-Adressen zu einem Host

Dieses Szenario kommt z.B. vor, wenn ein Server mehrere IPs hat und die Daten für Cacti von verschiedenen IPs kommen, z.B. einem "Remote Insight Board".

Das ganze geht, wenn man eine Data Input Method verwendet, die einen hostname als input field benötigt. Im Data Template erscheint dann der hostname als Eingabefeld im Bereich Custom Data. Hier aktiviert man dann das Häkchen für "Use per Data Source Value". Wenn es zum Erstellen der Data Sources kommt gibt man einfach als hostname die IP der Remote Console an statt die des Hosts. Man hat dann zwar nicht wirklich Devices mit unterschiedlichen IPs zusammengefasst, jedoch kommen die Daten von unterschiedlichen IPs obwohl alles einem einzigen Device zugeordnet ist.

Die Kommandozeilenschnittstelle (CLI)

Ein neues Device mit der IP 192.168.1.35 anlegen, mit dem Host-Template 3 (id). Diese Id kann man ermitteln indem man in Cacti auf den Link eines Host-Templates geht sich unten im Browser die genaue URL ansieht. Z.B. in der Standardinstallation hat das Host-Template "ucd/net SNMP Host" die ID 3: http://cactiserver/cacti/host_templates.php?action=edit&id=3

```
php /srv/www/htdocs/cacti/cli/add_device.php --description='Webserver_1' --ip='192.168.1.35' \
--template=3 --avail=snmp --community='public'
```

http://docs.cacti.net/manual:087:6_reference.4_cli_script#command_line_scripts

Monitoring von Clustern

...

Disaster Recovery

...

Der Aufbau der Cacti-Datenbank

...

Eigene Plugins programmieren

...

21. Skript-Vorlagen

PHP-Skript zur Abfrage eines einzelnen Wertes per snmpget.

```
<?php

# PHP script template for a single value.

# deactivate http headers
$no_http_headers = true;

# include some cacti files for ease of use
include(dirname(__FILE__) . '/../include/config.php');
include(dirname(__FILE__) . '/../lib/snmp.php');

# OID for a single value: uptime snmpd
$oid_uptime = "DISMAN-EVENT-MIB::sysUpTimeInstance";

# all required input parms
$hostname     = $_SERVER["argv"][1];
$snmp_community = $_SERVER["argv"][2];
$snmp_version  = $_SERVER["argv"][3];
$snmp_port     = $_SERVER["argv"][4];
$snmp_timeout  = $_SERVER["argv"][5];
$snmp_user     = "";
$snmp_pw       = "";
$cmd           = "get";
$snmp_retries  = 3;

# snmp v3
$auth_proto = "";
$priv_pass  = "";
$priv_proto = "";
$context    = "";

if ($cmd == "get") {
    $uptime = cacti_snmp_get($hostname, $snmp_community, $oid_uptime, $snmp_version,
        $snmp_user, $snmp_pw, $auth_proto, $priv_pass, $priv_proto, $context, $snmp_port,
        $snmp_timeout, $snmp_retries);
} else {
    print "Invalid use of script query, required parameters:\n\n";
    print "  <hostname> <community> <oid> <version> <user> <pw> <auth_proto> <priv_pass>
<priv_proto> <context> <port> <timeout> <retries>\n";
    return 1;
}

# Example to extract numerical characters and turn the string into a float:
#if (preg_match('/^ *?(\d+\.\d+)/', $uptime, $temp) != 1) {
#     print "Error, could not get snmp value\n";
#     return 1;
# }
```

```
#}  
#$uptime = floatval($temp[0]);  
  
print $uptime;  
  
?>
```

PHP-Skript zur Abfrage mehrere Werte(Tabelle) mit snmpget, z.B. für den Fall, dass snmpwalk nicht geht (Loops, Fehler, keine Indexspalte)

```
<?php  
  
# PHP script template for script data queries in Cacti  
  
# deactivate http headers  
$no_http_headers = true;  
  
# include some cacti files for ease of use  
include(dirname(__FILE__) . '/../include/config.php');  
include(dirname(__FILE__) . '/../lib/snmp.php');  
  
# OIDs for requesting several lines, for example if snmpwalk doesn't work for any reason (errors,  
loops, no index column)  
$oid_inoctets = array(1 => "IF-MIB::ifInOctets.1",  
2 => "IF-MIB::ifInOctets.2",  
3 => "IF-MIB::ifInOctets.3");  
  
# all required input parms  
$hostname = $_SERVER["argv"][1];  
$snmp_community = $_SERVER["argv"][2];  
$snmp_version = $_SERVER["argv"][3];  
$snmp_port = $_SERVER["argv"][4];  
$snmp_timeout = $_SERVER["argv"][5];  
$snmp_user = "";  
$snmp_pw = "";  
$cmd = "get";  
$snmp_retries = 3;  
  
# snmp v3  
$auth_proto = "";  
$priv_pass = "";  
$priv_proto = "";  
$context = "";  
  
$inoctetsi=array();  
$nr=1;  
$result="";  
  
# Execute snmpget on every entry in the array
```

```

foreach ($oid_inoctets as $oid){
    if ($cmd == "get") {
        $inoctets = cacti_snmp_get($hostname, $snmp_community, $oid, $snmp_version,
        $snmp_user, $snmp_pw, $auth_proto, $priv_pass, $priv_proto, $context, $snmp_port,
        $snmp_timeout, $snmp_retries);

# Example to extract numerical characters and turn the string into an integer:
#     if (preg_match('/^ *?(\d+)/', $inoctets, $oc) != 1) {
#         print "Error, could not get snmp value\n";
#         return 1;
#     }
#     $inoctetsi[$nr] = intval($oc[0]);
#     $nr++;
# } else {
#     print "Invalid use of script query, required parameters:\n\n";
#     print "   <hostname> <community> <oid> <version> <user> <pw> <auth_proto>
<priv_pass> <priv_proto> <context> <port> <timeout> <retries>\n";
#     return 1;
# }

        $inoctetsi[$nr] = $inoctets;
        $nr++;
    }

unset($oid);
$nr=1;

# Concatenate all results to one single string like Cacti wants it
foreach ($octetsi as $val){
    $result = $result . "nic${nr}inoctets:$val ";
    $nr++;
}

unset($str);

# Remove last blank
$result=substr("$result",0,-1);

print "$result";

?>

```

22. Issues

Es gibt unter Umständen Unterschiede zwischen dem Anlegen von Graphen über ein Device oder direkt über Graph Management. Wenn man in einem Data Template (für skalare Werte) bei custom data bei einem Feld das Häkchen anklickt wird wie erwartet beim Erstellen des Graphen über das Device dieses Feld abgefragt und man kann eine individuelle Eingabe machen. Wenn man jedoch nicht über das Device geht sondern direkt in Graph Management den Graph anlegt kommt diese Abfrage nicht und der Graph wird dann später auch nicht dargestellt.

Im Bereich User Management -> Graph Permissions funktionieren die Regeln mit der Default Policy deny nicht richtig:

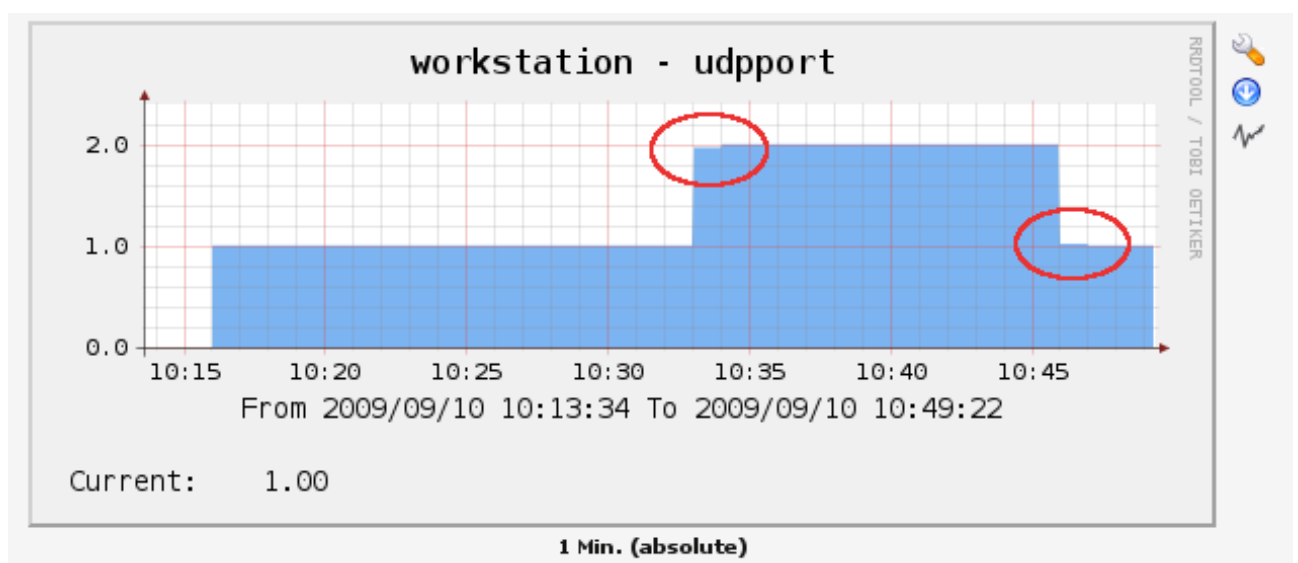
Z.B. wenn man nur das Device X erlaubt sieht man wie erwartet alle Graphen nur von Device X. Erlaubt man nun zusätzlich einen einzelnen Graph für einen anderen Host im darüberliegenden Abschnitt, sieht man diesen Graphen aber nicht, sondern wie vorher nur alles für das device X!
<http://bugs.cacti.net/view.php?id=1530>

Möchte man spine von der Shell direkt aufrufen muss man die ausführlichen Parameter (z.B. --first) angeben, die kurzen funktionieren nicht.

Bei der Installation des Weathermap-Plugins auf Debian führt ein Bug in einem GD-Paket dazu, dass das Plugin nicht funktioniert. Hierzu gibt es folgenden Foreneintrag mit entsprechender Lösung:

<http://forums.cacti.net/viewtopic.php?t=21517>
dpkg -i ./libgd2-xpm_2.0.35.dfsg-1_i386.deb

Möchte man Integerwerte darstellen kann es sein, dass unerwartet Kommazahlen auftreten. Dies kann man z.B. beobachten wenn man einen Parameter nimmt der ausschließlich zwei Integerwerte, z.B. 1 und 2 zurückliefert. Im Graph erwartet man daher nur 1 und 2 als mögliche Werte. Solange sich der Wert nicht ändert ist auch alles wie erwartet. Aber sobald der Wert von 1 auf 2 oder umgekehrt springt, hat man seltsame Zwischenwerte. Das kann z.B. so aussehen:



In den roten Ellipsen sieht man kleine Abweichungen.

Die Ursache ist, dass der Poller fast nie hundertprozentig exakt zu den Zeitintervallen die Daten

liefert wie sie rrdtool erwartet. Es muss immerhin auf die Sekunde stimmen. Und wenn nun der aktuelle Wert 1 ist und 1 Sekunde zu spät geliefert wird und der vorherige Wert 2 ist, rechnet rrdtool das zurück und gibt dann nicht 1 ein sondern den interpolierten Wert für den exakten Zeitpunkt ein. Siehe auch:

http://docs.cacti.net/manual:087:8_rrdtool.02a_non_integers#non_integer_values_shown
und

<http://oss.oetiker.ch/rrdtool/tut/rrdtutorial.en.html>

Abschnitt Data Resampling

Gegenprobe mit Timestamps die exakt 60 Sekunden auseinanderliegen und genau dem Minutentakt der Uhrzeit entsprechen. Nimmt man irgendwelche Timestamps, selbst wenn sie genau 60 Sekunden auseinanderliegen, tritt der oben beschriebene Effekt auf.

```
/usr/bin/rrdtool create ./test.rrd --step 60 DS:udpport:GAUGE:120:0:U RRA:AVERAGE:0.5:1:60  
RRA:AVERAGE:0.5:5:288 RRA:AVERAGE:0.5:30:336 RRA:AVERAGE:0.5:120:372  
RRA:AVERAGE:0.5:1440:365 RRA:AVERAGE:0.5:60:8760 RRA:LAST:0.5:1:60
```

```
rrdtool update ./test.rrd 1252581480:1 1252581540:1 1252581600:1
```

```
rrdtool update ./test.rrd 1252581660:2 1252581720:2 1252581780:1
```

```
rrdtool fetch ./test.rrd LAST --start 1252581480 --end 1252581780  
udpport
```

```
1252581540: 1.0000000000e+00  
1252581600: 1.0000000000e+00  
1252581660: 2.0000000000e+00  
1252581720: 2.0000000000e+00  
1252581780: 1.0000000000e+00
```

Hier sieht man also, dass der Effekt nicht auftritt, es stehen tatsächlich nur Integerwerte in der rrd-Datei.

Feedback ist willkommen:

Norbert Klein

norbert.klein@infosecprojects.net

www.infosecprojects.net