

Performance-Analyse mit **collectd**

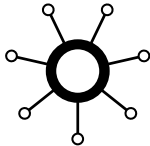
Wisse, was deine Rechner machen

Sebastian „tokkee“ Harl <sh@teamix.net>

team(ix) GmbH / **collectd** core team



12. Mai 2011



- gegründet 2001
- Ursprünge: Open-Source und Netzwerke
 - Debian
 - Nagios
 - Schulungen
 - u.v.m.
- Heute auch:
 - NetApp
 - VMWare
 - Riverbed (WAN-Beschleunigung)
 - Juniper
 - N-IX (Nürnberger Internet-eXchange)

Was ist collectd?

Wichtige Eigenschaften

Wichtige Plugins

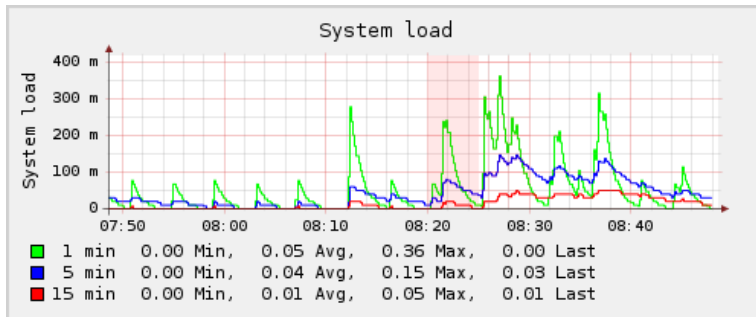
Über den Tellerrand

Optional

- **collectd** sammelt Leistungsdaten von Rechnern
- Leistungsdaten sind zum Beispiel:
 - CPU-Auslastung
 - Speichernutzung
 - Netzwerkverkehr
- Daten werden erhoben, verarbeitet und gespeichert
- Häufig: Darstellung als Graphen
- → Performance-Analyse, Kapazitätsplanung
- Nicht verwechseln mit *Monitoring!*
- Homepage: <http://collectd.org/>

- Daemon
- Freie Software (größtenteils GPLv2)
- Portierbar (Linux, *BSD, Solaris, ...)
- Skalierbar (OpenWrt, ..., Cluster / Cloud)
- Effizient (Default-Auflösung: 10 Sekunden)
- Modular (über 100 Plugins in Version 5.0)

- Daemon
- Freie Software (größtenteils GPLv2)
- Portierbar (Linux, *BSD, Solaris, ...)
- Skalierbar (OpenWrt, ..., Cluster / Cloud)
- **Effizient** (Default-Auflösung: 10 Sekunden)
- Modular (über 100 Plugins in Version 5.0)



- Daemon
- Freie Software (größtenteils GPLv2)
- Portierbar (Linux, *BSD, Solaris, ...)
- Skalierbar (OpenWrt, ..., Cluster / Cloud)
- Effizient (Default-Auflösung: 10 Sekunden)
- Modular (über 100 Plugins in Version 5.0)

- Daemon
- Freie Software (größtenteils GPLv2)
- Portierbar (Linux, *BSD, Solaris, ...)
- Skalierbar (OpenWrt, ..., Cluster / Cloud)
- Effizient (Default-Auflösung: 10 Sekunden)
- **Modular** (über 100 Plugins in Version 5.0)

apache	amqp	apcups	ascent	battery
bind	contrack	contextswitch	cpu	cpufreq
csv	curl	curl_json	dbi	df
disk	dns	email	entropy	exec
filecount	fscache	GenericJMX	gmond	hddtemp
interface	ipmi	iptables	ipvs	irq
java	libvirt	load	logfile	madwifi
match_regex	mbmon	memcachec	memcached	memory
Monitorus	multimeter	mysql	netapp	netlink
network	nfs	nginx	notify_email	ntpd
nut	olsrd	onewire	openvpn	OpenVZ
oracle	perl	ping	postgresql	powerdns
processes	protocols	python	routeros	rrdcached
rrdtool	sensors	serial	snmp	swap
syslog	table	tail	tape	target_scale
tcpconns	teamspeak2	ted	thermal	tokyotyrant
unixsock	uptime	users	uuid	vmem
vserver	wireless	write_http	xmms	zfs_arc

- Aktuelle Version ist 5.0 (Release: März 2011)
- Pakete für diverse Distributionen vorhanden (Debian, RedHat, FreeBSD, OpenWrt, OpenSolaris [WIP], ...)
- Major-Version 4.x ist an einigen Stellen inkompatibel¹
→ v5upgrade Target
- Geschrieben in C
- Versionsverwaltung mit Git
→ `git://git.verplant.org/collectd.git`

¹http://collectd.org/wiki/index.php/V4_to_v5_migration_guide

- Daemon läuft auf jedem Client (Ausnahme: SNMP o.ä.)
- üblicherweise: ein oder mehrere zentrale Server, die Werte von Clients empfangen (Push-Modell)
- First steps: `install; select plugins; start daemon; enjoy ;-)`

C4

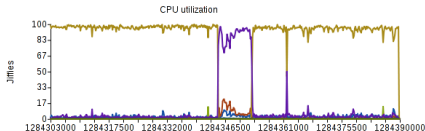
collection 4

- All instances
- All graphs
- Host "..."

Graph "CPU utilization"

Instance "..." / 0"

Instance: "..." / cpu - 0 / cpu - all"



Search

Hour

JSON (gRaphaël)

RRDtool

Go

collection 4.0.0

- Grundidee: Daten über, z. B., JSON zur Verfügung stellen
- verschiedene Frontends davor möglich
- effiziente Handhabung von vielen Datensätzen durch Caching
- flexible Konfiguration von Graphen

Was ist collectd?

Wichtige Plugins

- CPU, Speicher, Netzwerk-I/O
- Netzwerk-Plugin
- RRDtool-Plugin (Überblick)
- Generische Plugins (Überblick)
- Eigene Erweiterungen (Überblick)

Über den Tellerrand

Optional

- Spezielle Lese-Plugins
 - CPU, Speicher, Netzwerk-Schnittstellen
- Schreib- bzw. IO-Plugins
 - Netzwerk-Plugin
 - RRDtool, RRDCacheD
- Generische Plugins
 - SNMP
 - tail

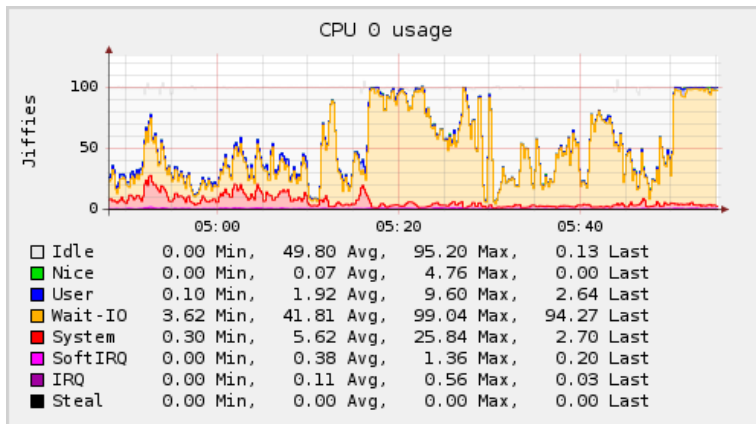
Synopsis

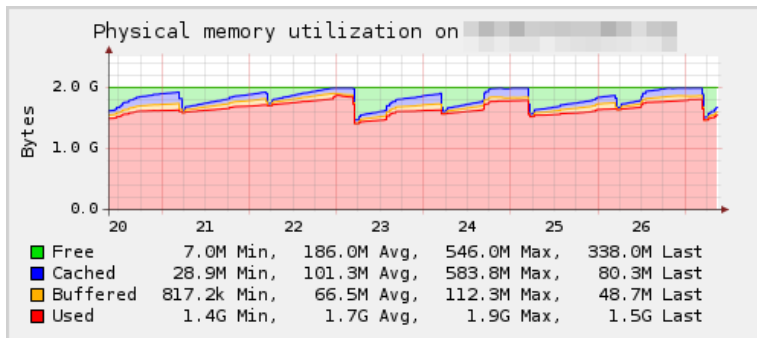
```
LoadPlugin "cpu"  
LoadPlugin "memory"  
LoadPlugin "interface"
```

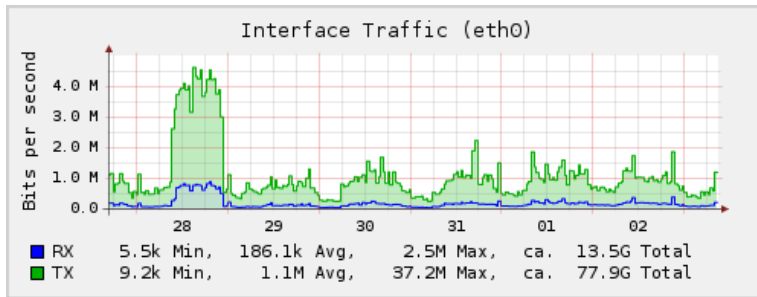

Synopsis

```
LoadPlugin "cpu"  
LoadPlugin "memory"  
LoadPlugin "interface"
```

```
<Plugin interface>  
  Interface lo  
  Interface sit0  
  IgnoreSelected true  
</Plugin>
```





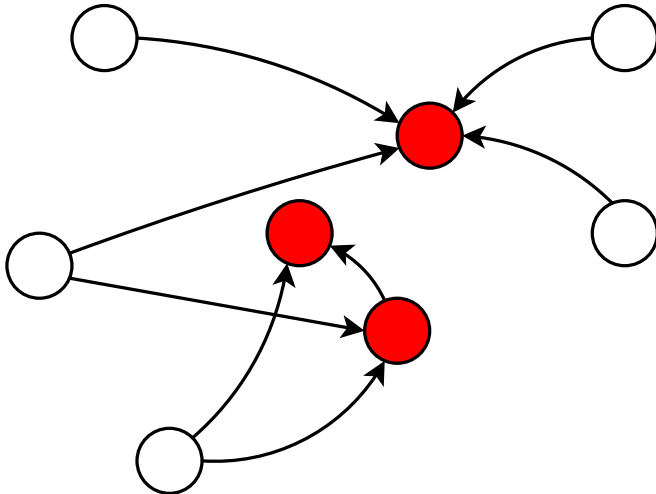


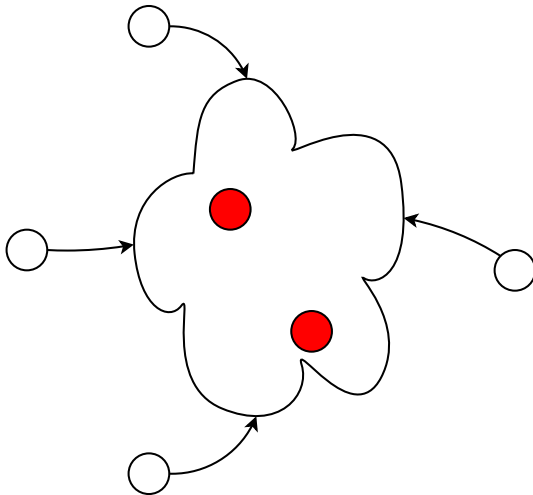
Betriebsarten

- Daten versenden („*Client*“)
- Daten empfangen („*Server*“)
- Weiterleiten („*Proxy*“)
- Unicast („*Punkt-zu-Punkt*“)
- Multicast („*Punkt-zu-Gruppe*“)
- IPv4 und IPv6

Ein Daemon für alles

Rolle des Daemon hängt von der Konfiguration ab.





Synopsis: Client

```
LoadPlugin "network"
```

```
<Plugin "network">
```

```
  Server "collectd0.musterfirma.de"
```

```
  Server "collectd1.musterfirma.de"
```

```
  Server "ff18::efc0:4a42"
```

```
</Plugin>
```


Synopsis: Server

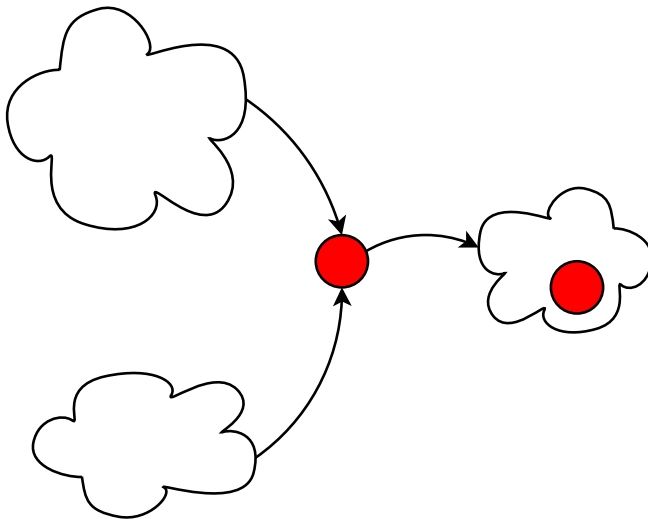
```
LoadPlugin "network"
```

```
<Plugin "network">
```

```
  Listen "collectd0.musterfirma.de"
```

```
  Listen "ff18::efc0:4a42"
```

```
</Plugin>
```



Synopsis: Proxy

```
LoadPlugin "network"
```

```
<Plugin "network">
```

```
  Listen "collectgw.extern.musterfirma.de"
```

```
  Server "collected1.intern.musterfirma.de"
```

```
  Forward true
```

```
</Plugin>
```

Authentifizierung / Verschlüsselung

- (seit Version 4.7.0)
- Authentifizierung via HMAC-SHA-256
- Verschlüsselung mit AES-256 (OFB)

Authentifizierung / Verschlüsselung

		Client		
		Nichts	Sign	Encrypt
Server	Nichts	akzeptiert	akzeptiert	nicht möglich
	AuthFile	akzeptiert	akzeptiert	akzeptiert
	Sign	nicht akzeptiert	akzeptiert	akzeptiert
	Encrypt	nicht akzeptiert	nicht akzeptiert	akzeptiert

- Schreibt Daten effizient in RRD-Dateien → Caching
- Funktionalität nun in RRDtool als RRD Caching Daemon verfügbar

Synopsis

```
LoadPlugin "rrdtool"
```

```
<Plugin "rrdtool">
```

```
  DataDir "/var/lib/collectd/rrd"
```

```
</Plugin>
```

Konfiguration

```
<Plugin "rrdtool">  
  DataDir "/var/lib/collectd/rrd"  
  
  CacheTimeout 3600 # 1 hour  
  CacheFlush 86400 # 1 day  
  
  WritesPerSecond 30  
</Plugin>
```

- FLUSH ermöglicht dennoch die graphische Darstellung von aktuellen Daten

- Idee: Generische Ansätze, statt Speziallösungen
- → Benutzerkonfiguration bestimmt das Verhalten
- ⇒ Unterstützung für neue Geräte braucht i.d.R. keine neue Version von **collectd**
- Beispiele: SNMP, tail, curl, DBI

- **collectd** API: C, Perl, Python, Java
- Externe Programme mittels unixsock- oder exec-Plugin

- Plugins für zahlreiche Systemmetriken existieren (z. B. für CPU, Speicher und Netzwerk-Schnittstellen)

- Plugins für zahlreiche Systemmetriken existieren (z. B. für CPU, Speicher und Netzwerk-Schnittstellen)
- Vielfältige Netzwerk-Möglichkeiten (IPv4, IPv6, Unicast, Multicast, Proxies)

- Plugins für zahlreiche Systemmetriken existieren (z. B. für CPU, Speicher und Netzwerk-Schnittstellen)
- Vielfältige Netzwerk-Möglichkeiten (IPv4, IPv6, Unicast, Multicast, Proxies)
- Bewährtes Caching-Modell für RRD-Dateien

- Plugins für zahlreiche Systemmetriken existieren (z. B. für CPU, Speicher und Netzwerk-Schnittstellen)
- Vielfältige Netzwerk-Möglichkeiten (IPv4, IPv6, Unicast, Multicast, Proxies)
- Bewährtes Caching-Modell für RRD-Dateien
- Mächtige, generische Ansätze statt Speziallösungen (z. B. SNMP- und tail-Plugins)

Was ist collectd?

Wichtige Plugins

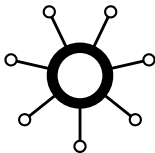
Über den Tellerrand

Optional

- `collectd-nagios`
Fragt Daten via `unixsock`-Plugin ab und erzeugt Nagios-kompatible Ausgabe
- `exec-nagios.px`
Perl-Skript welches *Nagios*-Plugins ausführt (→ *exec-Plugin*)
- `exec-munin.px`
Perl-Skript welches *Munin*-Plugins ausführt (→ *exec-Plugin*)
- `gmond-Plugin`
Empfängt und verarbeitet *Ganglia* Multicast-Pakete

Vielen Dank für die Aufmerksamkeit!

Gibt es Fragen?



Danke an Florian Forster für die initiale Version dieser Folien!

Kontakt:

Sebastian „tokkee“ Harl
team(ix) GmbH, Nürnberg
<sh@teamix.net>

<collectd@verplant.org> — irc.freenode.net/#collectd — <http://identi.ca/collectd>

Artikel zum Thema:

[http://linuxtechnicalreview.de/Vorschau/\(show\)
/Themen/Monitoring/Performance-Analyse-mit-Collectd](http://linuxtechnicalreview.de/Vorschau/(show)/Themen/Monitoring/Performance-Analyse-mit-Collectd)

Was ist collectd?

Wichtige Plugins

Über den Tellerrand

Optional

SNMP-Plugin

tail-Plugin

RRDCacheD-Plugin

Eigene Erweiterungen

Über den Tellerrand

Allgemeines

- Fragt Netzwerk-Zubehör via SNMP ab
- *Generisch*: Nicht für ein gestimmtes Gerät geschrieben
- Mehrere Geräte werden parallel abgefragt

Konfiguration

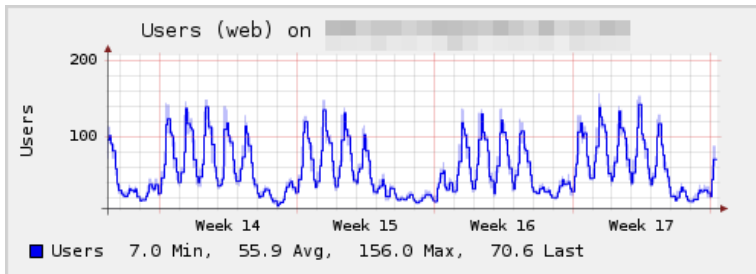
- „Data“-Blöcke
- „Host“-Blöcke

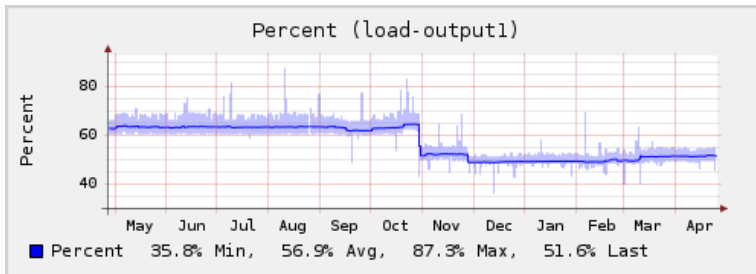
Synopsis: Data-Block

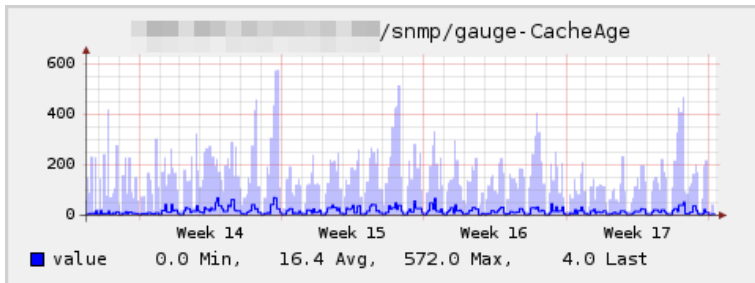
```
<Plugin "snmp">
  <Data "ifmib_if_octets64">
    Type "if_octets"
    Table true
    Instance "IF-MIB::ifName"
    Values "IF-MIB::ifHCInOctets" \
          "IF-MIB::ifHCOutOctets"
  </Data>
</Plugin>
```

Synopsis: Host-Block

```
<Plugin "snmp">  
  <Host "switch0.intern.musterfirma.de">  
    Address "10.0.42.2"  
    Version 1  
    Community "public"  
    Collect "ifmib_if_octets64"  
    Interval 60  
  </Host>  
</Plugin>
```





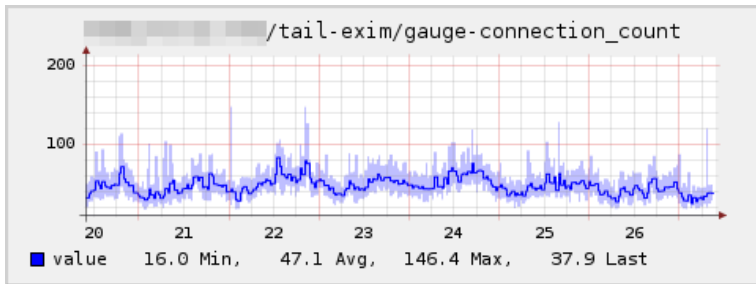


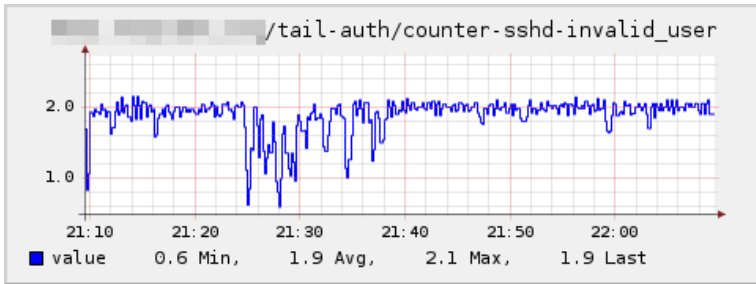
Allgemeines

- Verfolgt Log-Dateien
- Extrahiert Werte oder zählt Ereignisse
- Selektion der Zeilen / Werte mit regulären Ausdrücken
- Verwendbar für MTAs, Web-Server, ...

Konfiguration

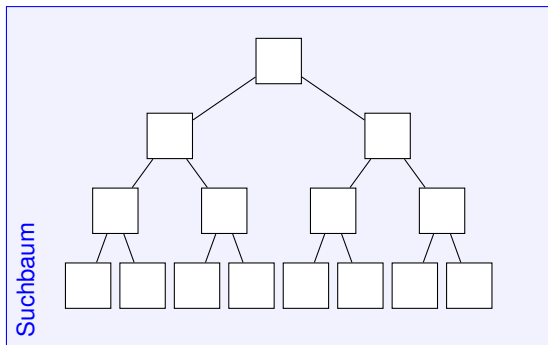
```
<Plugin "tail">
  <File "/var/log/exim4/mainlog">
    Instance "exim"
    <Match>
      Regex "S=([1-9] [0-9]*)"
      DSType "CounterAdd"
      Type "ipt_bytes"
      Instance "total"
    </Match>
  </File>
</Plugin>
```

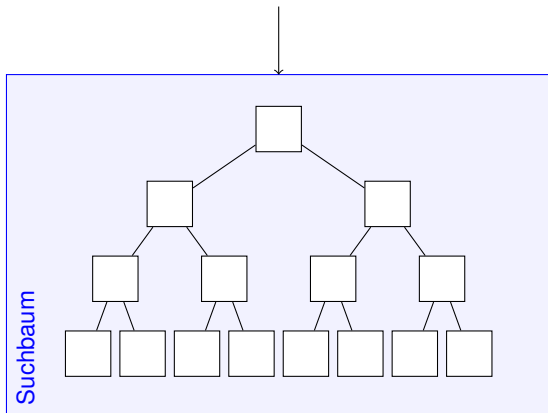


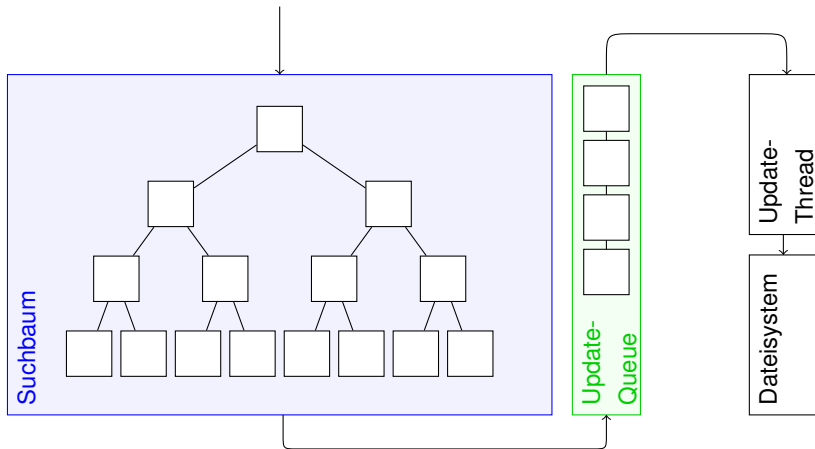


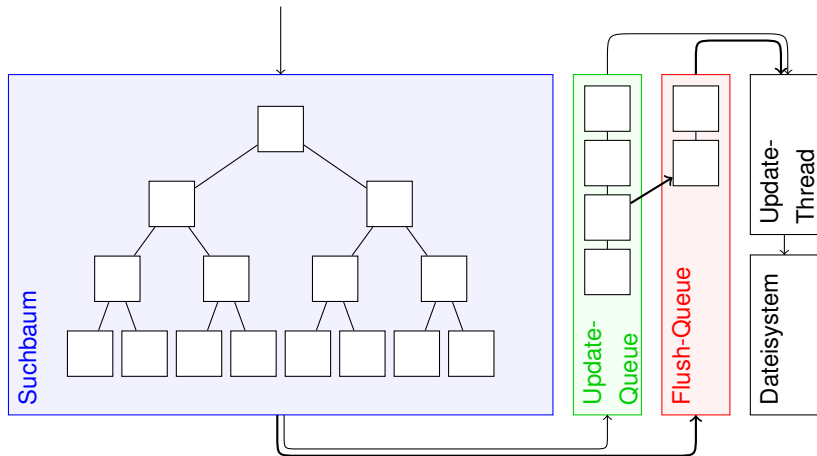
Allgemeines

- Update-Prinzip des RRDtool-Plugins
- Eigenständiger Daemon
- Integration in RRDtool 1.4
- Weitere Funktionen, z. B. Journaling
- Vorteil: Neustart von **collectd** ohne Cache-Verlust









Allgemeines

- Integriert einen Perl-Interpreter
(vergleichbar zu Apaches `mod_perl`)
- Instanziierung und Syntax-Analyse nur einmal
- Exportiert die API
(→ nicht nur Lese-Plugins möglich)

```
package Collectd::Plugin::Magic;
use Collectd qw( :all );
sub magic_read
{
    my $v1 = { plugin => 'magic',
               values => [Magic->getCurrentLevel ()] };
    plugin_dispatch_values ('magic_level', $v1);
}
plugin_register (TYPE_READ, 'magic', 'magic_read');
```

Allgemeines

- Öffnet einen UNIX-Domain-Socket
- Kennt mehrere Befehle
(z. B. PUTVAL, FLUSH, LISTVAL)
- Interaktion mit externen Programmen möglich
- `collectdctl` (ab Version 5.0, eta: dieses Jahr ;-))
- `cussh.pl`: „*collectd UNIX socket shell*“

```
-> | PUTVAL "testhost/magic/magic_level" \  
    interval=10 1179574444:42  
<- | 0 Success
```

Allgemeines

- Führt Programme aus
- Liest von deren Standard-Ausgabe
- Können über längere Perioden laufen
(vgl. `init`)

```
#!/bin/sh
INTVL=${COLLECTD_INTERVAL:-10}
CHOST="${COLLECTD_HOSTNAME:-localhost}"
IDENT="$CHOST/magic/magic_level"
while sleep $INTVL
do
    VALUE='magic --level'
    echo "PUTVAL \"$IDENT\" interval=$INTVL N:$VALUE"
done
```

Allgemeines

- Integriert eine „Java Virtual Maschine“ (JVM)
- Exportiert die API
(→ nicht nur Lese-Plugins möglich)
- Prinzipielle Ähnlichkeit zum Perl-Plugin


```
import org.collectd.api.Collectd;  
import org.collectd.api.CollectdReadInterface;  
public class MagicPlugin  
    implements CollectdReadInterface  
{  
    public int read ();    /* Callback-Funktion */  
    public MagicPlugin (); /* Konstruktor */  
}
```

```
public int read ()  
{  
    ValueList vl = new ValueList ();  
    vl.setHost ("testhost");  
    vl.setPlugin ("magic");  
    vl.setType ("magic_level");  
    vl.addValue (Magic.getCurrentLevel ());  
    return (Collectd.dispatchValues (vl));  
}
```

```
public MagicPlugin ()  
{  
    /* Callback-Funktion anmelden */  
    Collectd.registerRead ("MagicPlugin", this);  
}
```

- **collectd** API nutzen
C, Perl, Python und Java möglich

- **collectd** API nutzen
C, Perl, Python und Java möglich
- Externe Programme erweitern
unixsock-Plugin ermöglicht Kommunikation

- **collectd** API nutzen
C, Perl, Python und Java möglich
- Externe Programme erweitern
unixsock-Plugin ermöglicht Kommunikation
- Eigenes Programm / Skript schreiben
→ exec-Plugin

- `snmp-probe-host.px`
Erzeugt semi-automatisch `<Host />`-Blöcke für das SNMP-Plugin
- `jcollected`
Java-Implementierung des Netzwerk-Protokolls (→ *JMX*)
- `kcollected`
KDE-Programm zur Near-Realtime-Anzeige von Graphen
- Perl, Ruby, Python Module und C-Bibliothek für die Kommunikation mit dem `unixsock`-Plugin verfügbar