

Einführung in **collectd**

Florian Forster

collectd-Entwickler

2009-11-15

Gliederung

Gliederung

Was ist collectd?

Gliederung

Gliederung

Was ist collectd?

Wichtige Plugins

Gliederung

Gliederung

Was ist collectd?

Wichtige Plugins

Ausblick

Gliederung

Gliederung

Was ist collectd?

Wichtige Plugins

Ausblick

Optional

Was ist **collectd**?

- ▶ **collectd** sammelt **Leistungsdaten** von Rechnern

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:
 - ▶ CPU-Auslastung

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:
 - ▶ CPU-Auslastung
 - ▶ Speichernutzung

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:
 - ▶ CPU-Auslastung
 - ▶ Speichernutzung
 - ▶ Netzwerkverkehr

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:
 - ▶ CPU-Auslastung
 - ▶ Speichernutzung
 - ▶ Netzwerkverkehr
- ▶ Daten werden **erhoben, verarbeitet und gespeichert**

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:
 - ▶ CPU-Auslastung
 - ▶ Speichernutzung
 - ▶ Netzwerkverkehr
- ▶ Daten werden erhoben, verarbeitet und gespeichert
- ▶ Häufig: Darstellung als **Graphen**

Was ist **collectd**?

- ▶ **collectd** sammelt Leistungsdaten von Rechnern
- ▶ Leistungsdaten sind zum Beispiel:
 - ▶ CPU-Auslastung
 - ▶ Speichernutzung
 - ▶ Netzwerkverkehr
- ▶ Daten werden erhoben, verarbeitet und gespeichert
- ▶ Häufig: Darstellung als Graphen
- ▶ **Nicht** verwechseln mit *Monitoring!*

Was ist **collectd**?

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

Wichtige Eigenschaften

- ▶ **Daemon**

Wichtige Eigenschaften

- ▶ Daemon
- ▶ **Freie Software** (größtenteils GPLv2)

Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ **Portierbar** (Linux, BSD, Solaris, ...)

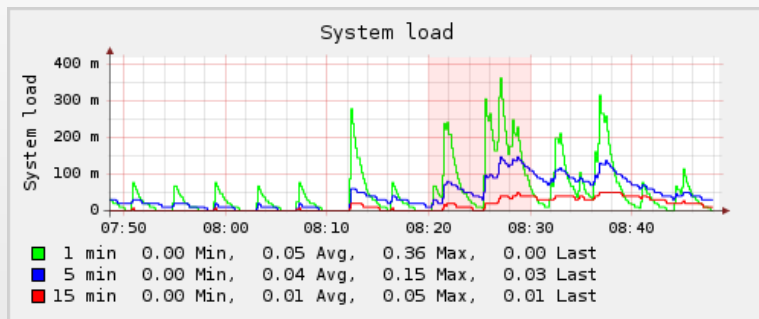
Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, BSD, Solaris, ...)
- ▶ **Skalierbar** (OpenWrt, ..., Cluster / Cloud)

Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, BSD, Solaris, ...)
- ▶ Skalierbar (OpenWrt, ..., Cluster / Cloud)
- ▶ **Effizient** (Default-Auflösung: 10 Sekunden)

Wichtige Eigenschaften: 10-Sekunden-Auflösung



Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, BSD, Solaris, ...)
- ▶ Skalierbar (OpenWrt, ..., Cluster / Cloud)
- ▶ **Effizient** (Default-Auflösung: 10 Sekunden)

Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, BSD, Solaris, ...)
- ▶ Skalierbar (OpenWrt, ..., Cluster / Cloud)
- ▶ Effizient (Default-Auflösung: 10 Sekunden)
- ▶ **Modular** (Über 80 Plugins)

Verfügbare Plugins (Auswahl; Stand: Version 4.8)

apache	dns	logfile	openvpn	syslog
apcups	email	mbmon	oracle	tail
apple_sensors	entropy	memcached	perl	tape
ascent	exec	memory	ping	tcpconns
battery	filecount	multimeter	postgresql	teamspeak2
bind	hddtemp	mysql	powerdns	thermal
cpu	interface	netlink	processes	unixsock
cpufreq	iptables	network	rrdcached	users
csv	ipmi	nfs	rrdtool	uuid
curl	ipvs	nginx	sensors	vmem
dbi	irq	ntpd	serial	vserver
df	libvirt	nut	snmp	wireless
disk	load	onewire	swap	xmms

Technische Details

- ▶ Aktuelle Version ist **4.8.1**
- ▶ Pakete für diverse Distributionen vorhanden
(Debian, RedHat, OpenSUSE, FreeBSD, OpenWrt, ...)
- ▶ Major-Version 3.* ist veraltet und inkompatibel
- ▶ Geschrieben in C
- ▶ Versionsverwaltung mit Git

Technische Details

- ▶ Aktuelle Version ist 4.8.1
- ▶ **Pakete** für diverse Distributionen vorhanden
(Debian, RedHat, OpenSUSE, FreeBSD, OpenWrt, ...)
- ▶ Major-Version 3.* ist veraltet und inkompatibel
- ▶ Geschrieben in C
- ▶ Versionsverwaltung mit Git

Technische Details

- ▶ Aktuelle Version ist 4.8.1
- ▶ Pakete für diverse Distributionen vorhanden (Debian, RedHat, OpenSUSE, FreeBSD, OpenWrt, ...)
- ▶ Major-Version **3.*** ist **veraltet** und inkompatibel
- ▶ Geschrieben in C
- ▶ Versionsverwaltung mit Git

Technische Details

- ▶ Aktuelle Version ist 4.8.1
- ▶ Pakete für diverse Distributionen vorhanden (Debian, RedHat, OpenSUSE, FreeBSD, OpenWrt, ...)
- ▶ Major-Version **3.*** ist **veraltet** und **inkompatibel**
- ▶ Geschrieben in C
- ▶ Versionsverwaltung mit Git

Technische Details

- ▶ Aktuelle Version ist 4.8.1
- ▶ Pakete für diverse Distributionen vorhanden (Debian, RedHat, OpenSUSE, FreeBSD, OpenWrt, ...)
- ▶ Major-Version 3.* ist veraltet und inkompatibel
- ▶ Geschrieben in **C**
- ▶ Versionsverwaltung mit Git

Technische Details

- ▶ Aktuelle Version ist 4.8.1
- ▶ Pakete für diverse Distributionen vorhanden
(Debian, RedHat, OpenSUSE, FreeBSD, OpenWrt, ...)
- ▶ Major-Version 3.* ist veraltet und inkompatibel
- ▶ Geschrieben in C
- ▶ Versionsverwaltung mit **Git**

Wichtige Plugins

- ▶ Spezielle **Lese-Plugins**
 - ▶ CPU, Speicher, Netzwerk-Schnittstellen
- ▶ Schreib- bzw. IO-Plugins
 - ▶ Network-Plugin
 - ▶ RRDtool (*falls Interesse*)
- ▶ Generische Plugins (*falls Interesse*)
 - ▶ SNMP
 - ▶ Tail

Wichtige Plugins

- ▶ Spezielle Lese-Plugins
 - ▶ CPU, Speicher, Netzwerk-Schnittstellen
- ▶ **Schreib-** bzw. **IO-Plugins**
 - ▶ Network-Plugin
 - ▶ RRDtool (*falls Interesse*)
- ▶ Generische Plugins (*falls Interesse*)
 - ▶ SNMP
 - ▶ Tail

Wichtige Plugins

- ▶ Spezielle Lese-Plugins
 - ▶ CPU, Speicher, Netzwerk-Schnittstellen
- ▶ **Schreib-** bzw. **IO-Plugins**
 - ▶ Network-Plugin
 - ▶ RRDtool (*falls Interesse*)
- ▶ Generische Plugins (*falls Interesse*)
 - ▶ SNMP
 - ▶ Tail

Wichtige Plugins

- ▶ Spezielle Lese-Plugins
 - ▶ CPU, Speicher, Netzwerk-Schnittstellen
- ▶ Schreib- bzw. IO-Plugins
 - ▶ Network-Plugin
 - ▶ RRDtool (*falls Interesse*)
- ▶ **Generische Plugins** (*falls Interesse*)
 - ▶ SNMP
 - ▶ Tail

Wichtige Plugins

- ▶ Spezielle Lese-Plugins
 - ▶ CPU, Speicher, Netzwerk-Schnittstellen
- ▶ Schreib- bzw. IO-Plugins
 - ▶ Network-Plugin
 - ▶ RRDtool (*falls Interesse*)
- ▶ Generische Plugins (*falls Interesse*)
 - ▶ SNMP
 - ▶ Tail

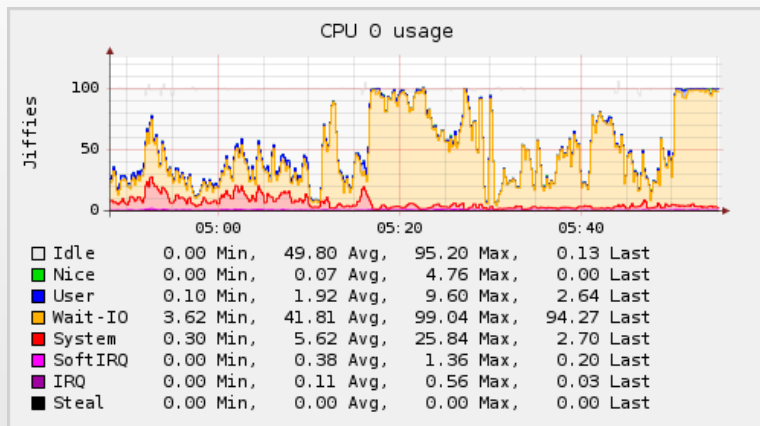
CPU, Speicher, Netzwerk-Schnittstellen

Synopsis

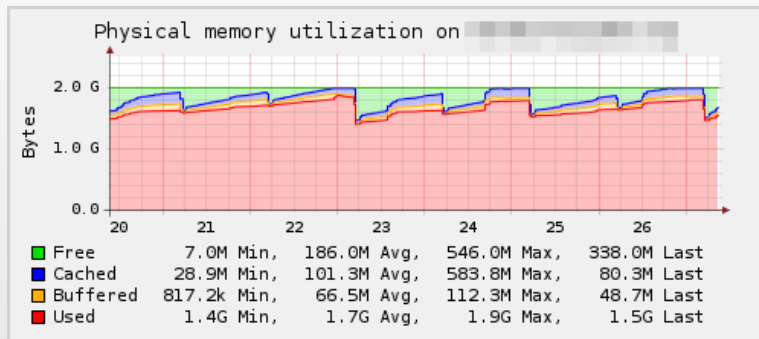
```
LoadPlugin "cpu"  
LoadPlugin "memory"  
LoadPlugin "interface"
```

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

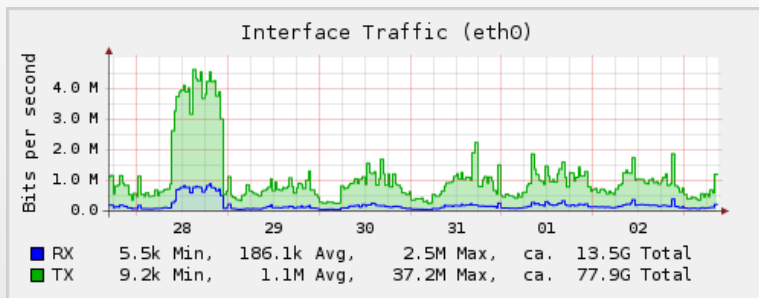
CPU, Speicher, Netzwerk-Schnittstellen



CPU, Speicher, Netzwerk-Schnittstellen



CPU, Speicher, Netzwerk-Schnittstellen



Network-Plugin

Betriebsarten

- ▶ Daten versenden („*Client*“)

Network-Plugin

Betriebsarten

- ▶ Daten versenden („*Client*“)
- ▶ Daten empfangen („*Server*“)

Network-Plugin

Betriebsarten

- ▶ Daten versenden („*Client*“)
- ▶ Daten empfangen („*Server*“)
- ▶ Weiterleiten („*Proxy*“)

Network-Plugin

Betriebsarten

- ▶ Daten versenden („*Client*“)
- ▶ Daten empfangen („*Server*“)
- ▶ Weiterleiten („*Proxy*“)

Ein Daemon für alles

Rolle des Daemon hängt von der Konfiguration ab.

Network-Plugin

Synopsis: Client

```
LoadPlugin "network"
```

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

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

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

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

```
</Plugin>
```

Network-Plugin

Synopsis: Server

```
LoadPlugin "network"
```

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

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

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

```
</Plugin>
```

Network-Plugin

Synopsis: Proxy

```
LoadPlugin "network"
```

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

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

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

```
  Forward true
```

```
</Plugin>
```

Network-Plugin

Authentifizierung / Verschlüsselung

- ▶ Seit *Version 4.7*
- ▶ Authentifizierung via HMAC-SHA-256
- ▶ Verschlüsselung mit AES-256 (OFB)

Ausblick: Version 4.9

- ▶ Version 4.9

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features
 - ▶ NetApp-Plugin

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features
 - ▶ NetApp-Plugin
 - ▶ Hash-Match (→ Load-Balancing)

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features
 - ▶ NetApp-Plugin
 - ▶ Hash-Match (→ Load-Balancing)
 - ▶ Scale-Target

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features
 - ▶ NetApp-Plugin
 - ▶ Hash-Match (→ Load-Balancing)
 - ▶ Scale-Target
- ▶ In Arbeit

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features
 - ▶ NetApp-Plugin
 - ▶ Hash-Match (→ Load-Balancing)
 - ▶ Scale-Target
- ▶ In Arbeit
 - ▶ Python-Anbindung

Ausblick: Version 4.9

- ▶ Version 4.9
- ▶ Feature-Freeze am 9. Dezember
- ▶ Veröffentlichung am 21. Dezember
- ▶ Bereits implementierte neue Features
 - ▶ NetApp-Plugin
 - ▶ Hash-Match (→ Load-Balancing)
 - ▶ Scale-Target
- ▶ In Arbeit
 - ▶ Python-Anbindung
 - ▶ Erlang-Anbindung

Zusammenfassung

Zusammenfassung

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

Zusammenfassung

Zusammenfassung

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

Zusammenfassung

Zusammenfassung

- ▶ Plugins für zahlreiche Systemmetriken existieren (z. B. für CPU, Speicher und Netzwerk-Schnittstellen)
- ▶ Vielfältige Netzwerk-Möglichkeiten (IPv4, IPv6, Unicast, Multicast, Proxies)
- ▶ Mächtige, generische Ansätze statt Speziallösungen (z. B. SNMP- und Tail-Plugins)

Einführung in **collectd**

Vielen Dank für die Aufmerksamkeit!

Gibt es Fragen?

Optionales: Gliederung

Gliederung

Optional

RRDtool-Plugin

SNMP-Plugin

Tail-Plugin

Eigene Erweiterungen

 Perl-Plugin

 UnixSock-Plugin

 Exec-Plugin

 Java-Plugin

 Ausblick

Daten filtern

Benachrichtigungen

Über den Tellerrand: Tools

RRDtool-Plugin

Allgemeines

- ▶ Schreibt Daten in RRD-Dateien

RRDtool-Plugin

Allgemeines

- ▶ Schreibt Daten *effizient* in RRD-Dateien

RRDtool-Plugin

Allgemeines

- ▶ Schreibt Daten *effizient* in RRD-Dateien
- ▶ *Caching* um Performance-Problemen zu begegnen

RRDtool-Plugin

Allgemeines

- ▶ Schreibt Daten *effizient* in RRD-Dateien
- ▶ *Caching* um Performance-Problemen zu begegnen
- ▶ *Flushing* für aktuelle Daten

RRDtool-Plugin

Allgemeines

- ▶ Schreibt Daten *effizient* in RRD-Dateien
- ▶ *Caching* um Performance-Problemen zu begegnen
- ▶ *Flushing* für aktuelle Daten
- ▶ *Throttling* für gleichmäßige Last

RRDtool-Plugin

Synopsis

```
LoadPlugin "rrdtool"
```

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

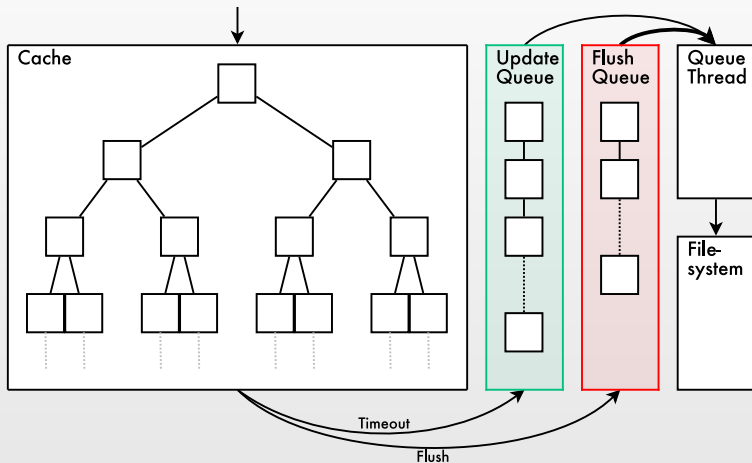
```
  DataDir "/var/lib/collectd/rrd"
```

```
  CacheTimeout 300
```

```
  WritesPerSecond 30
```

```
</Plugin>
```

RRDtool-Plugin



RRDCacheD-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

SNMP-Plugin

Allgemeines

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

Konfiguration

- ▶ „Data“-Blöcke
- ▶ „Host“-Blöcke

SNMP-Plugin

Synopsis: Data-Block

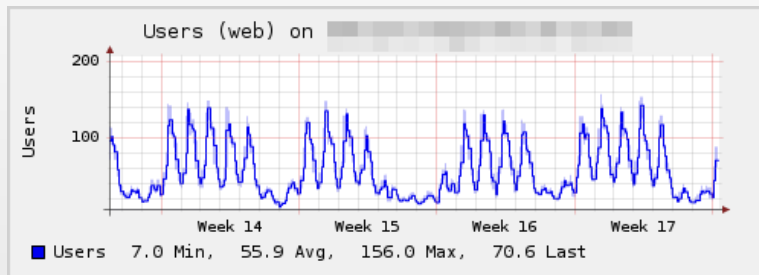
```
<Plugin "snmp">
  <Data "ifmib_if_octets64">
    Type "if_octets"
    Table true
    Instance "IF-MIB::ifName"
    Values "IF-MIB::ifHCInOctets" \
          "IF-MIB::ifHCOutOctets"
  </Data>
</Plugin>
```

SNMP-Plugin

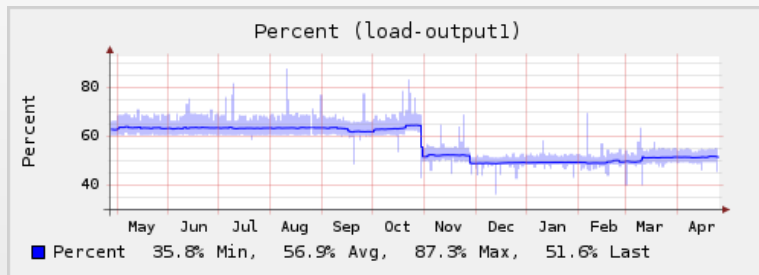
Synopsis: Host-Block

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

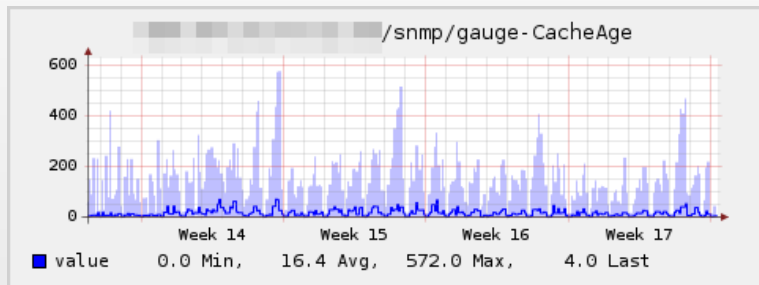

SNMP-Plugin: Users



SNMP-Plugin: USV-Last



SNMP-Plugin: Cache-Alder

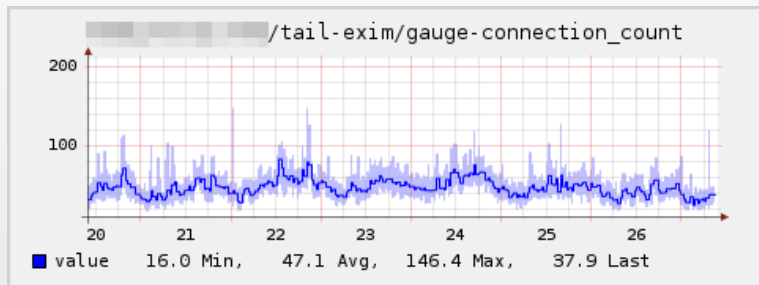


Tail-Plugin

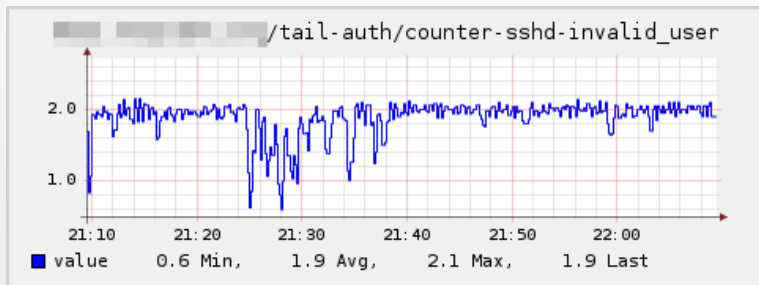
Allgemeines

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

Tail-Plugin: Verbindungen von Exim



Tail-Plugin: SSH Brute-Force-Angriffe



Perl-Plugin

Allgemeines

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

Perl-Plugin: Beispiel

Beispiel-Code

```
package Collectd::Plugin::Magic;
use Collectd qw( :all );

sub magic_read
{
    my $vl = { plugin => 'magic',
              values => [Magic->getCurrentLevel ()] };
    plugin_dispatch_values ('magic_level', $vl);
}
plugin_register (TYPE_READ, 'magic', 'magic_read');
```


UnixSock-Plugin

Allgemeines

- ▶ Öffnet einen UNIX-Domain-Socket
- ▶ Kennt mehrere Befehle
(z. B. PUTVAL, FLUSH, LISTVAL)
- ▶ Interaktion mit externen Programmen möglich
- ▶ `cussh.pl`: „*collectd UNIX socket shell*“
- ▶ Mehrere Abstraktionsschichten existieren
(z. B. für C, Perl und Python)

UnixSock-Plugin: Beispiel

Kommunikation

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

Exec-Plugin

Allgemeines

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

Exec-Plugin: Beispiel

Beispiel-Skript

```
#!/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
```

Java-Plugin

Allgemeines

- ▶ Verfügbar seit *Version 4.7*
- ▶ Integriert eine „Java Virtual Maschine“ (JVM)
- ▶ Exportiert die API
(→ nicht nur Lese-Plugins möglich)
- ▶ Prinzipielle Ähnlichkeit zum Perl-Plugin

Java-Plugin: Beispiel

Beispiel-Code (1/3): Überblick

```
import org.collectd.api.Collectd;
import org.collectd.api.CollectdReadInterface;

public class MagicPlugin
    implements CollectdReadInterface
{
    public int read ();    /* Callback-Funktion */
    public MagicPlugin (); /* Konstruktor */
}
```

Java-Plugin: Beispiel

Beispiel-Code (2/3): Callback-Funktion

```
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));
}
```

Java-Plugin: Beispiel

Beispiel-Code (3/3): Konstruktor

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


Ausblick

Python

- ▶ Voraussichtlich in *Version 4.9*
- ▶ Entwickelt von Sven Trenkel (noris network AG)
- ▶ Exportiert die API
(→ nicht nur Lese-Plugins möglich)
- ▶ Cool: Interaktive Debugging-Shell

Ausblick

Erlang

- ▶ Veröffentlichungszeitpunkt unbekannt (“when it’s ready”)
- ▶ Exportiert die API
(→ nicht nur Lese-Plugins möglich)
- ▶ Verhält sich wie eine “distributed Erlang-Node”
- ▶ Alternative: erlang-collectd von „Astro“
(Gibt’s auf github)

Eigene Erweiterungen: Zusammenfassung

Zusammenfassung

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

Eigene Erweiterungen: Zusammenfassung

Zusammenfassung

- ▶ **collectd** API nutzen

C, Perl und Java möglich, demnächst Python und Erlang

Eigene Erweiterungen: Zusammenfassung

Zusammenfassung

- ▶ **collectd** API nutzen
C, Perl und Java möglich, demnächst Python und Erlang
- ▶ Externe Programme erweitern
UnixSock-Plugin ermöglicht Kommunikation

Eigene Erweiterungen: Zusammenfassung

Zusammenfassung

- ▶ **collectd** API nutzen
C, Perl und Java möglich, demnächst Python und Erlang
- ▶ Externe Programme erweitern
UnixSock-Plugin ermöglicht Kommunikation
- ▶ Eigenes Programm / Skript schreiben
→ Exec-Plugin

Daten filtern

- ▶ Auswahl: „Matches“

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert
 - ▶ Zeit

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert
 - ▶ Zeit
- ▶ Aktion: „Targets“

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert
 - ▶ Zeit
- ▶ Aktion: „Targets“
 - ▶ Name ändern

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert
 - ▶ Zeit
- ▶ Aktion: „Targets“
 - ▶ Name ändern
 - ▶ Wert ändern

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert
 - ▶ Zeit
- ▶ Aktion: „Targets“
 - ▶ Name ändern
 - ▶ Wert ändern
 - ▶ Verwerfen

Daten filtern

- ▶ Auswahl: „Matches“
 - ▶ Name
 - ▶ Wert
 - ▶ Zeit
- ▶ Aktion: „Targets“
 - ▶ Name ändern
 - ▶ Wert ändern
 - ▶ Verwerfen
- ▶ Konzept bei IP-Tables geklaut

Daten filtern: Anwendungsgebiete

- ▶ Wenn Hostname keinen Punkt enthält, verwerfe Daten (Einfacher FQDN-Check)

Daten filtern: Anwendungsgebiete

- ▶ Wenn Hostname keinen Punkt enthält, verwerfe Daten (Einfacher FQDN-Check)
- ▶ Wenn Zählerstand null, verwerfe Daten („inaktive“ Zähler ignorieren, z. B. bei SNMP)

Daten filtern: Anwendungsgebiete

- ▶ Wenn Hostname keinen Punkt enthält, verwirfe Daten (Einfacher FQDN-Check)
- ▶ Wenn Zählerstand null, verwirfe Daten („inaktive“ Zähler ignorieren, z. B. bei SNMP)
- ▶ Umbenennen von Daten (Wechsel auf neue `1m_sensors`-Version)

Daten filtern: Verfügbare Matches

- ▶ Empty Counter
- ▶ Hashed (*Version 4.9*)
- ▶ RegEx
- ▶ TimeDiff
- ▶ Value

Daten filtern: Verfügbare Targets

- ▶ **Plugins**
 - ▶ Notification
 - ▶ Replace
 - ▶ Scale (*Version 4.9*)
 - ▶ Set
- ▶ **Built-in**
 - ▶ Jump
 - ▶ Return
 - ▶ Stop
 - ▶ Write

Daten filtern: Verfügbare Targets

- ▶ Plugins
 - ▶ Notification
 - ▶ Replace
 - ▶ Scale (*Version 4.9*)
 - ▶ Set
- ▶ **Built-in**
 - ▶ Jump
 - ▶ Return
 - ▶ Stop
 - ▶ Write

Benachrichtigungen

- ▶ Text-Nachrichten

Benachrichtigungen

- ▶ Text-Nachrichten
- ▶ Für einen Benutzer

Benachrichtigungen

- ▶ Text-Nachrichten
- ▶ Für einen Benutzer
- ▶ → *nicht* maschinenlesbar!

Benachrichtigungen

- ▶ Text-Nachrichten
- ▶ Für einen Benutzer
- ▶ → *nicht* maschinenlesbar!
- ▶ Assoziierte Meta-Daten

Benachrichtigungen

- ▶ Text-Nachrichten
- ▶ Für einen Benutzer
- ▶ → *nicht* maschinenlesbar!
- ▶ Assoziierte Meta-Daten
- ▶ → automatisch verarbeitbar

Plugins: Erzeugende

- ▶ `target_notification`

Plugins: Erzeugende

- ▶ `target_notification`
- ▶ IPMI

Plugins: Erzeugende

- ▶ `target_notification`
- ▶ IPMI
- ▶ MySQL

Plugins: Erzeugende

- ▶ `target_notification`
- ▶ IPMI
- ▶ MySQL
- ▶ Network

Plugins: Erzeugende

- ▶ target_notification
- ▶ IPMI
- ▶ MySQL
- ▶ Network
- ▶ Exec, Java, Perl, ...

Plugins: Konsumierende

- ▶ Notify Desktop

Plugins: Konsumierende

- ▶ Notify Desktop
- ▶ Notify Email

Plugins: Konsumierende

- ▶ Notify Desktop
- ▶ Notify Email
- ▶ LogFile, SysLog

Plugins: Konsumierende

- ▶ Notify Desktop
- ▶ Notify Email
- ▶ LogFile, SysLog
- ▶ Network

Plugins: Konsumierende

- ▶ Notify Desktop
- ▶ Notify Email
- ▶ LogFile, SysLog
- ▶ Network
- ▶ Exec, Java, Perl, ...

Schwellwerte

- ▶ *Einfache* Monitoring-Funktionalität

Schwellwerte

- ▶ *Einfache* Monitoring-Funktionalität
- ▶ Minimum, Maximum

Schwellwerte

- ▶ *Einfache* Monitoring-Funktionalität
- ▶ Minimum, Maximum
- ▶ Fehlende Daten

Schwellwerte

- ▶ *Einfache* Monitoring-Funktionalität
- ▶ Minimum, Maximum
- ▶ Fehlende Daten
- ▶ Wiederholt / einmalig

Schwellwerte

- ▶ *Einfache* Monitoring-Funktionalität
- ▶ Minimum, Maximum
- ▶ Fehlende Daten
- ▶ Wiederholt / einmalig
- ▶ Für komplexe(re) Checks / Aktionen: Eigenes Plugin

Schwellwerte

- ▶ *Einfache* Monitoring-Funktionalität
- ▶ Minimum, Maximum
- ▶ Fehlende Daten
- ▶ Wiederholt / einmalig
- ▶ Für komplexe(re) Checks / Aktionen: Eigenes Plugin
- ▶ `collectd-nagios`: **collectd**↔Nagios-Interaktion

Über den Tellerrand: Zubehör

- ▶ `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
- ▶ `Collectd::Unixsock`
Perl-Modul für die Kommunikation mit dem UnixSock-Plugin

Über den Tellerrand: Interaktion

- ▶ `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* (ab 4.7.0)