

# Performance-Analyse in großen Umgebungen mit **collectd**

Sebastian „tokkee“ Harl  
<tokkee@collectd.org>

FrOSCon 2009  
2009-08-22





# Gliederung

Was ist collectd?

Wichtige Eigenschaften

Wichtige Plugins

Eigene Erweiterungen

Über den Tellerrand



## 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!*



## Kontakt

- ▶ Homepage: <http://collectd.org/>
- ▶ Mailinglist: [collectd@verplant.org](mailto:collectd@verplant.org)
- ▶ IRC: #collectd auf [irc.freenode.net](http://irc.freenode.net)
- ▶ „Web 2.0“: <http://identi.ca/collectd>



## Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, \*BSD, Solaris, ...)
- ▶ Skalierbar (OpenWrt, ..., Cluster / Cloud)
- ▶ Effizient (Default-Auflösung: 10 Sekunden)
- ▶ Modular (Über 70 Plugins)

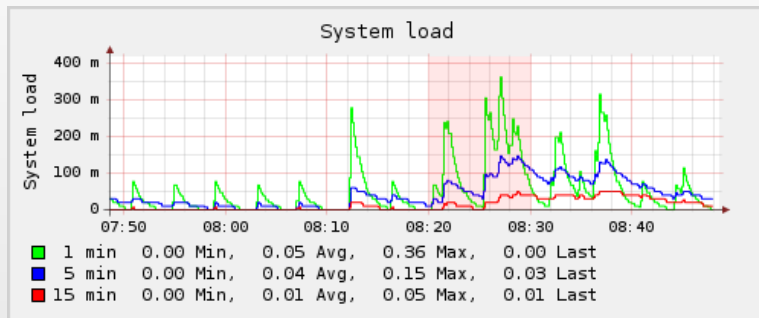


## Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, \*BSD, Solaris, ...)
- ▶ Skalierbar (OpenWrt, ..., Cluster / Cloud)
- ▶ **Effizient** (Default-Auflösung: 10 Sekunden)
- ▶ Modular (Über 70 Plugins)



## 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)
- ▶ Modular (Über 70 Plugins)





## Wichtige Eigenschaften

- ▶ Daemon
- ▶ Freie Software (größtenteils GPLv2)
- ▶ Portierbar (Linux, \*BSD, Solaris, ...)
- ▶ Skalierbar (OpenWrt, ..., Cluster / Cloud)
- ▶ Effizient (Default-Auflösung: 10 Sekunden)
- ▶ **Modular** (Über 70 Plugins)



└ Was ist collectd?

└ Wichtige Eigenschaften

## Verfügbare Plugins (Auswahl; Stand: Version 4.7)

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



## Technische Details

- ▶ Aktuelle Version ist 4.7.2
- ▶ Pakete für diverse Distributionen vorhanden (Debian, RedHat, FreeBSD, OpenWrt, ...)
- ▶ Major-Version 3.\* ist veraltet und inkompatibel
- ▶ Geschrieben in C
- ▶ Versionsverwaltung mit Git



# Gliederung

Was ist collectd?

## Wichtige Plugins

CPU, Speicher, Netzwerk-Schnittstellen

Network-Plugin

RRDtool- und RRDCacheD-Plugins

SNMP-Plugin

Tail-Plugin

Eigene Erweiterungen

Über den Tellerrand



## Wichtige Plugins

- ▶ Spezielle Lese-Plugins
  - ▶ CPU, Speicher, Netzwerk-Schnittstellen
- ▶ Schreib- bzw. IO-Plugins
  - ▶ Network-Plugin
  - ▶ RRDtool
  - ▶ RRDCached
- ▶ Generische Plugins
  - ▶ SNMP
  - ▶ Tail



# CPU, Speicher, Netzwerk-Schnittstellen

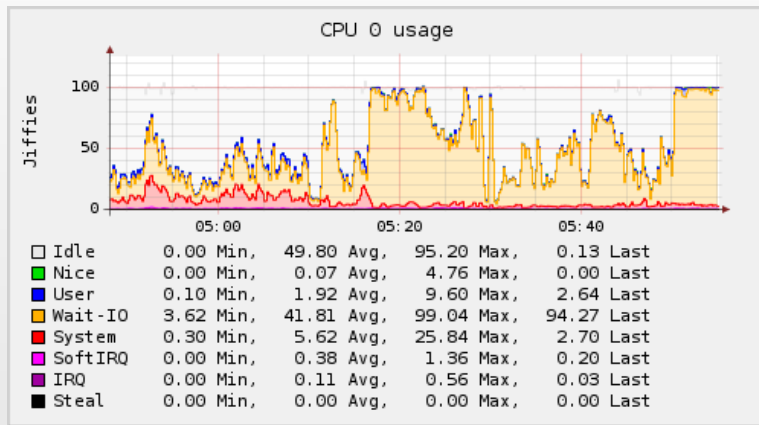
## Synopsis

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

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

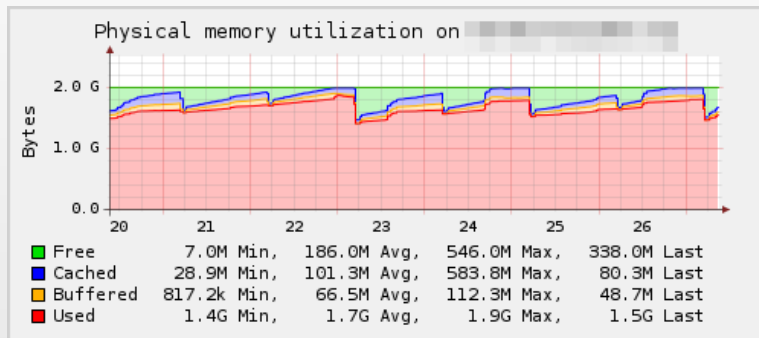


# CPU, Speicher, Netzwerk-Schnittstellen





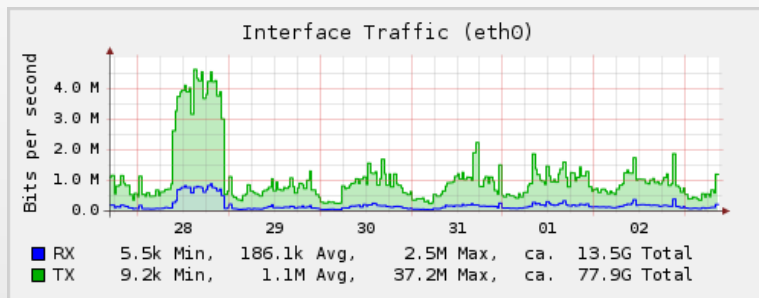
# CPU, Speicher, Netzwerk-Schnittstellen







# CPU, Speicher, Netzwerk-Schnittstellen





## Network-Plugin

### Betriebsarten

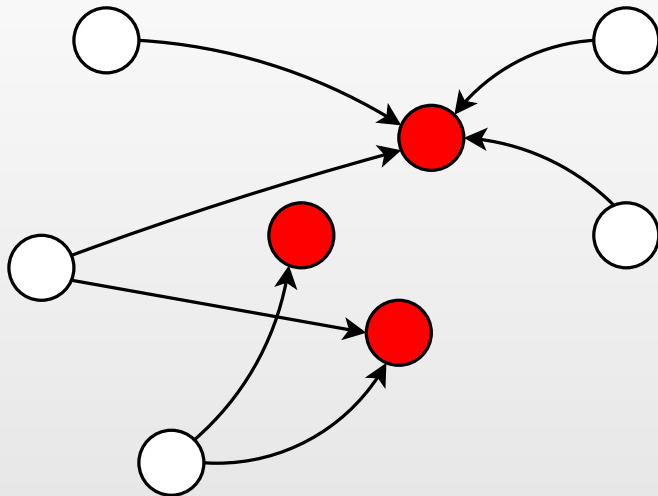
- ▶ Daten versenden („*Client*“)
- ▶ Daten empfangen („*Server*“)
- ▶ Weiterleiten („*Proxy*“)
- ▶ Unicast („*Punkt-zu-Punkt*“)
- ▶ Multicast („*Punkt-zu-Gruppe*“)
- ▶ IPv4 und IPv6

### Ein Daemon für alles

Rolle des Daemon hängt von der Konfiguration ab.

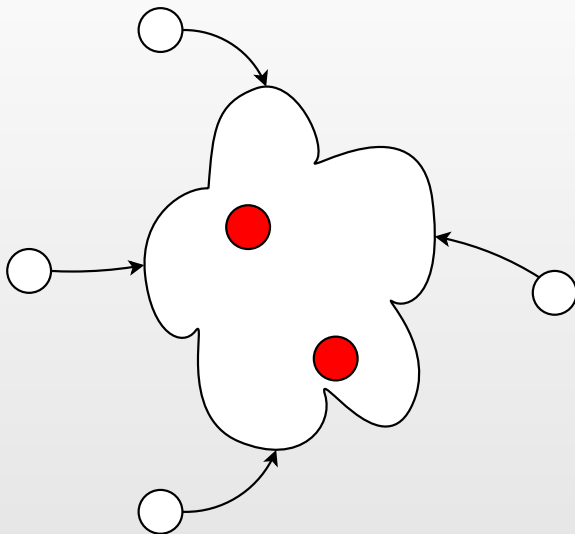


## Network-Plugin



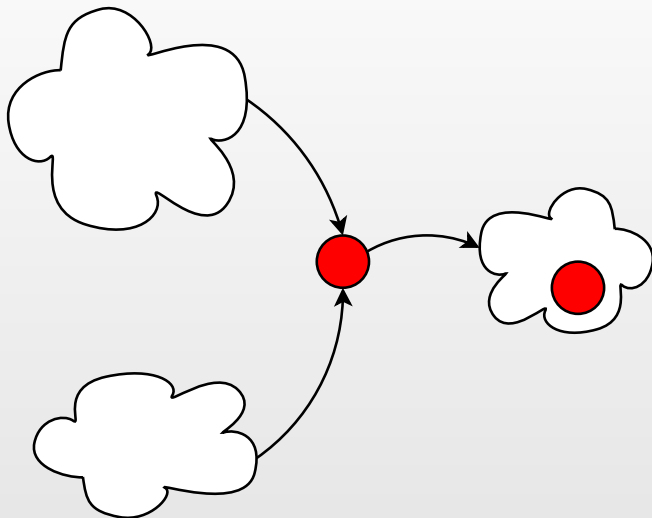


## Network-Plugin





## Network-Plugin





# 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 4.7.0)
- ▶ Authentifizierung via HMAC-SHA-256
- ▶ Verschlüsselung mit AES-256 (OFB)



## Network-Plugin

		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



# 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
```

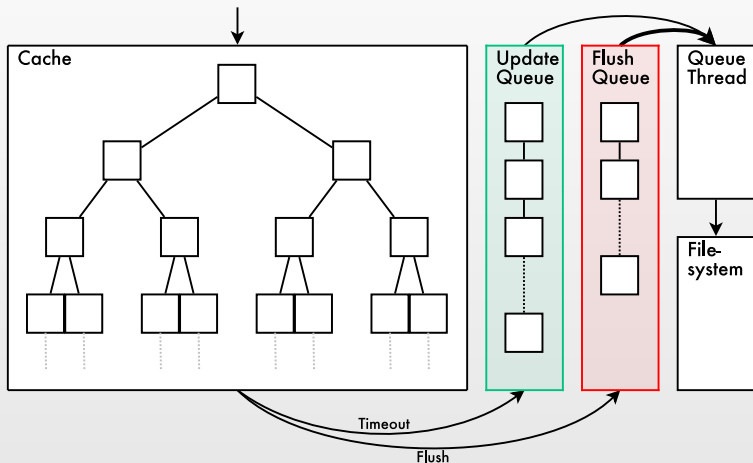
```
  WritesPerSecond 30
```

```
</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>
```



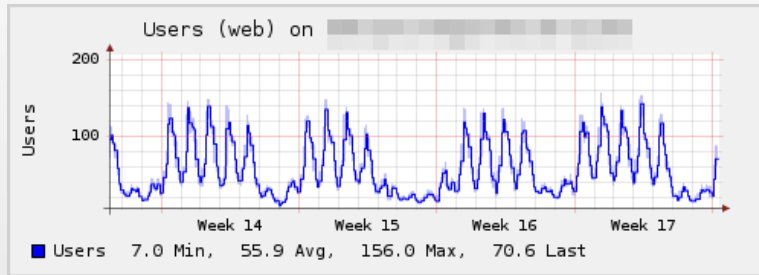
# SNMP-Plugin

## 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>
```

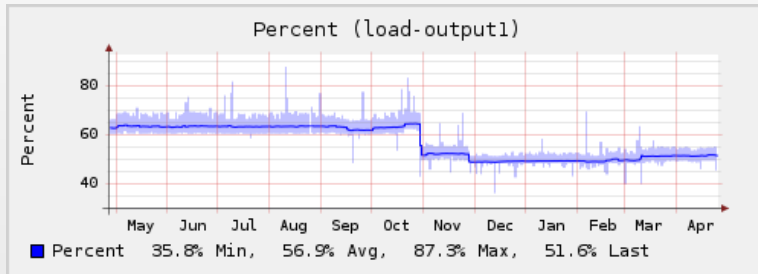


## SNMP-Plugin: Users



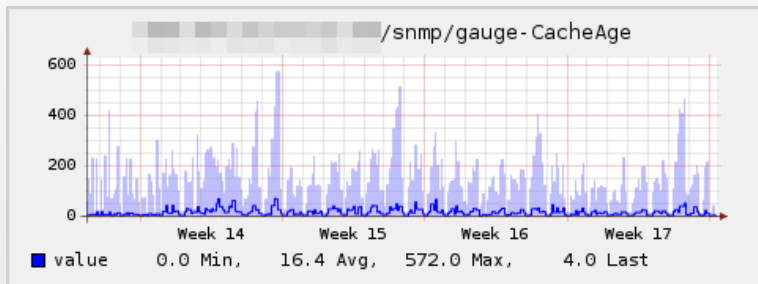


## SNMP-Plugin: USV-Last





## SNMP-Plugin: Cache-Alder







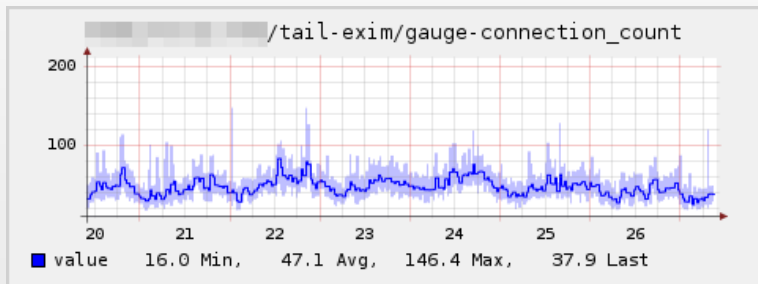
# Tail-Plugin

## Allgemeines

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

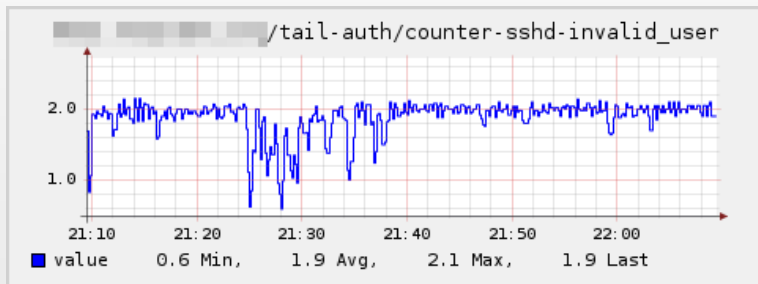


## Tail-Plugin: Verbindungen von Exim





## Tail-Plugin: SSH Brute-Force-Angriffe





## Wichtige Plugins: Zusammenfassung

### Zusammenfassung

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



# Wichtige Plugins: Zusammenfassung

## Zusammenfassung

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



# Wichtige Plugins: Zusammenfassung

## Zusammenfassung

- ▶ 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



## Wichtige Plugins: Zusammenfassung

### Zusammenfassung

- ▶ 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)



# Gliederung

Was ist collectd?

Wichtige Plugins

**Eigene Erweiterungen**

Perl-Plugin

UnixSock-Plugin

Exec-Plugin

Java-Plugin

Über den Tellerrand





# 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)



# 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*“



# Exec-Plugin

## Allgemeines

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



# Java-Plugin

## Allgemeines

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



## Eigene Erweiterungen: Zusammenfassung

### Zusammenfassung

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



## Eigene Erweiterungen: Zusammenfassung

### Zusammenfassung

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



## Eigene Erweiterungen: Zusammenfassung

### Zusammenfassung

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



# Gliederung

Was ist collectd?

Wichtige Plugins

Eigene Erweiterungen

Über den Tellerrand





## 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



# Performance-Analyse in großen Umgebungen mit **collectd**

Vielen Dank für die Aufmerksamkeit!

Gibt es Fragen?

Danke an Florian Forster für die initiale Version dieser Folien!