

collectd - the statistics collection daemon

Sebastian Harl

`<sh@tokkee.org>/<tokkee@lusc.de>`

Linux User Schwabach

13. Oktober 2007

Inhalt

Was ist collectd?

collectd - the statistics collection daemon

Geschichte

Ressourcen / Community

Wichtigste Features

Erweiterbarkeit / Bindings

Datenhaltung

Anwendungsfälle

Architektur

Überblick

Plugin-Aufbau

Monitoring-Integration

Zusammenfassung

Implementierung von neuen Plugins

collectd - the statistics collection daemon

- ▶ ursprünglich: reines sammeln von System-Informationen
- ▶ mittlerweile: modularer Aufbau mit vielen Erweiterungs- und Anwendungsmöglichkeiten, Perl-Bindings
- ▶ geplant: Integration einer Monitoring-Möglichkeit, weitere Bindings (z.B. Python), Windows Unterstützung (?) ...

Geschichte

- ▶ Juni/Juli 2005 von Florian Forster ins Leben gerufen
- ▶ erste Plugins: load, cpu, traffic
- ▶ aktuelle Version 4.0: 03. Juni 2007
- ▶ aktuell: > 20 Mitwirkende, 43 Plugins
- ▶ „Referenzen“: BBC UK, videobox.com, ... (ersetzt immer mehr Munin und Konsorten Setups)
- ▶ Zukunft: Weltherrschaft ;-)

Ressourcen / Community

- ▶ Homepage: `http://collectd.org/`
- ▶ SCM (Git): `git://git.verplant.org/collectd.git`
- ▶ Mailingliste: `collectd@verplant.org`
- ▶ IRC: `#collectd` auf Freenode

Wichtigste Features

- ▶ modularer Aufbau ⇒ mächtige Erweiterungsmöglichkeiten
- ▶ durchdachte Standard-Einstellungen ⇒ möglichst einfache Konfiguration
- ▶ hohe Auflösung der gesammelten Werte (Standard: 10s)
- ▶ mächtige Netzwerk-Unterstützung (u.a. Multicast, Unicast, IPv6, „Proxy“ /Weiterleitung)
- ▶ fein-granular skalierbar
- ▶ C99 (zumindest größtenteils) ⇒ gute Portierbarkeit (von Linux, *BSD, Solaris, MacOSX ist bekannt, dass es läuft)
- ▶ keine zwingenden Abhängigkeiten und geringe Ressourcen-Anforderungen ⇒ läuft auch auf kleinen System (z.B. OpenWRT)

Erweiterbarkeit / Bindings

- ▶ C-Plugins
- ▶ Perl-Plugins
- ▶ „Shell-Plugins“
- ▶ Unix-Domain-Socket
- ▶ SNMP-Unterstützung
- ▶ Nagios-Integration

Datenhaltung

- ▶ RRD (→ <http://oss.oetiker.ch/rrdtool/>)
- ▶ CSV
- ▶ Netzwerk

Anwendungsfälle

- ▶ Performance Analyse
- ▶ Kapazitätsplanung
- ▶ nette, bunte Graphen :-)

Architektur-Überblick

- ▶ schlanker Core
- ▶ Core verwaltet Plugins über Dispatch-Funktionen
- ▶ Plugins registrieren Callbacks für versch. Aufgaben (read, write, log) ⇒ im Prinzip alle Funktionalität in den Plugins

Plugin-Aufbau (C/Perl)

- ▶ `module_register()`
- ▶ „Arbeitsfunktionen“
- ▶ Plugin: `plugin_register_<type>()`
- ▶ Core: `plugin_read_all()`
- ▶ `plugin_dispatch_values()`, `plugin_log()`

Monitoring-Integration

- ▶ momentan in der Planungsphase
- ▶ Design-Überlegungen:
 - ▶ Überwachung von Schwellwert und Status-Indikatoren
 - ▶ Auslösen von notification-Callbacks

```

,----- .      ,----- .      ,----- .
,->! read  !---->! dispatch !--+->! notif. !
! ! plugin !    ! notif. !    ! ! plugin !
! '-----'    '-----'    ! '-----'
! ,----- .      A      ! ,----- .
,----- . +->! read  !-- .      ,-----!---- .      '->! notif. !
! plugin ! ! ! plugin ! ! ! check !    ! plugin !
! read  !-+ '-----' ! ! thresh. !    '-----'
! all  ! ! ,----- . ! '-----'    ,----- .
'-----' +->! read  !--+      A      , >! write !
! ! plugin ! ! ,-----!---- . ! ! plugin !
! '-----' +->! dispatch !--+ '-----'
! ,----- . ! ! values ! ! ,----- .
'->! read  !--' '-----' '->! write !
! plugin !    ! plugin !
'-----'    '-----'

```

Implementierung von neuen Plugins

- ▶ Demonstration...

Fragen?

History:

- ▶ 2007/10/13: Initial version (LUSC Workshop-Weekend 2007)