

Systemmonitoring unter Linux

CPU-Counter

B.Sc. Wirtsch.-Inform. Arno Sagawe, 29.06.10

Gliederung

- Systemmonitoring
- Protokolle und Dateien für das Systemmonitoring
- Datenzugriff
- Performance Counter
- Von Tools bis Enterprise Solutions
- Beispiel: Collectd
- Praxisbeispiele
- Fazit

Systemmonitoring

Systemmonitor (engl. *system monitor*, SM): einen Prozess der Daten über (s)ein System bzw. eine Anwendung sammelt, speichert und bereitstellt.

Notwendige Konfigurationen für das Systemmonitoring:

- Die *Anwendung* (SM) muss für das eigentliche Monitoren *konfiguriert* werden.
- Das *System* muss so *konfiguriert* werden, sodass es überwacht werden kann.

Protokolle und Dateien für das Systemmonitoring

- **SNMP**: Protokoll zum überwachen und steuern (verteilter) Systeme
- **CORBA**: Spezifikation für eine objektorientierte Middleware
- **JMX**: Spezifikation zur Verwaltung und Überwachung von Java-Anwendungen
- **TCP/IP**: Transport und Internetprotokoll
- **ICMP**: Austausch von Informations- und Fehlermeldungen über das Internet-Protokoll
- ...

- **/proc**: Virtuelles Dateisystem auf Unixsystemen; dient zur Ausgabe und Änderung von System- und Prozessinformation
- **/sys**: Stellt im Userspace Informationen über Geräte und Treiber bereit
- **Dateien**: /var/log/messages, /var/log/syslog

Datenzugriff

Sammeln der Daten

Monitor pull: (statistisches) Sammeln von Daten über den aktuellen Status einer (externen) Komponente durch eine Client-Software (engl. *Polling*).

Agent push: die Komponente sendet die (lokal) gesammelten Daten von sich aus an den SM. Das *Versenden* der Daten kann periodisch erfolgen oder via Request, also per Aufforderung des SM an die Komponente.

Hybrid: Kombination aus beidem.

Speichern und bereitstellen der gesammelten Daten

Interfaces: Wie wird auf die Daten zugegriffen?

Datenhaltung: Wie sollen die Daten gespeichert bzw. bereitgestellt werden?

Zugriffsrecht: Wer darf die gesammelten Daten einsehen?

Performance Counter

Performance Counter: Bestandteil moderner CPUs, welcher verschiedene Vorgänge misst, die ein Prozessor ausführt. Die CPU wird nicht zusätzlich belastet.

Beispiel: **CPU_CLK_UNHALTED** zählt Prozessorzyklen, in denen der Prozessor tatsächlich gerechnet hat (d.h. nicht im Zustand HALT war).

Zwei Arten des Systemmonitorings!?

Überwachen von Komponenten via SNMP passiert im *produktiven Einsatz*, um die Verfügbarkeit bzw. entstehende Engpässe und vorhandene Probleme anzuzeigen.

Das Überwachen von Programmen per Performance Counter kommt (eher) bei deren *Entwicklung und Wartung*, als im produktiven Einsatz vor.

Von Tools bis Enterprise Solutions

ps: Liefert eine detaillierte Prozessliste.

top: kontinuierliche Ausgabe der wichtigsten Informationen zur Ressourcenausnutzung.

iostat: liefert Input/Output Statistiken für Devices, Partitionen und Netzwerk-Dateisysteme (NFS), CPU.

vmstat: liefert sowohl Informationen zum Speichern als auch Informationen zum Block IO und zur CPU Aktivität.

pmap: zeigt den Speicherverbrauch an, den eine Anwendung generiert. Mit pmap können Bottlenecks aufgespürt werden.

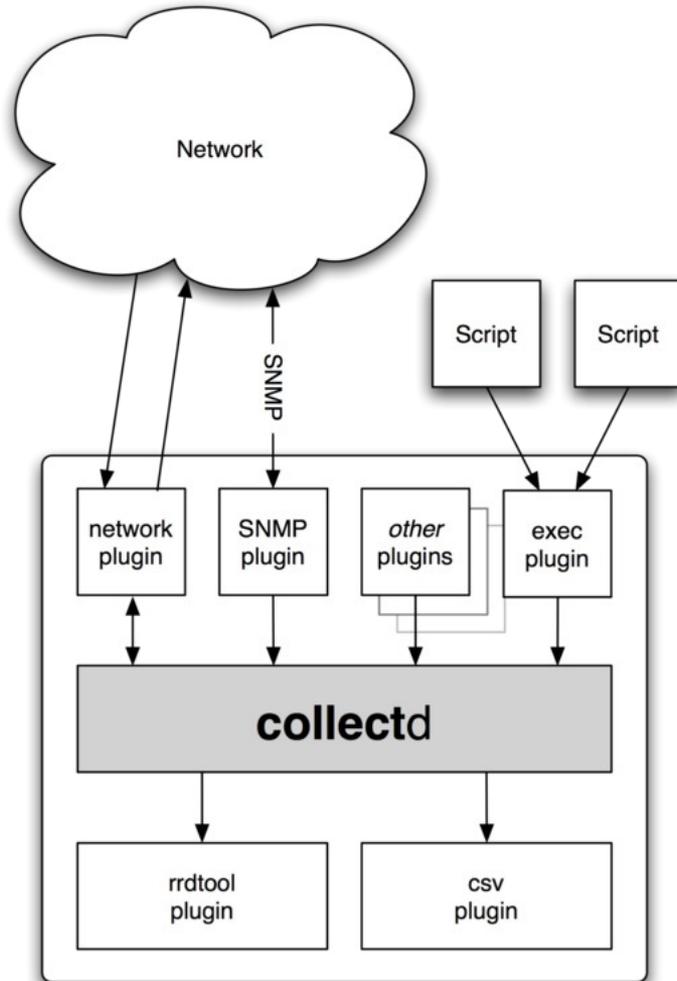
collectd: Sammelt Daten (netzwerkweit) und kann diese als Graphen visualisiert ausgeben.

oprofile/perf: Liefert detaillierte Informationen über die CPU-Nutzung von Programmen und Programmteilen:

nagios: Enterprise (Netzwerk-)Monitoring.

Beispiel: collectd

Quelle: <http://www.collectd.org/images/architecture-schematic.png>



Praxisbeispiele

- `collectd`
- `oprofile`
- `perf`

Fazit

collectd

- *Graphische Darstellung* von Systemaktivitäten
- Performance Counter werden *nicht unterstützt*
- *Nicht geeignet* um Engpässe in Anwendungen aufzuspüren (kein Profiler)
- Geeignet für das Überwachen des produktiven Einsatz

oprofile

- Ist ein vollwertiger Profiler
- Kann bei der *Entwicklung und Wartung* von Softwarekomponenten genutzt werden
- Es ist *eine GUI* vorhanden, was die Benutzbarkeit erhöht
- Unterstützt Performance Counter *direkt* und erzeugt dabei *minimalen Overhead*
- Nachteilig ist, noch *keinen 1.x Status* erreicht

perf

- Der offizielle Profiler für den GNU/Linux Kernel ($\geq 2.6.31$).
- *Unterstützt* Performance Counter direkt
- Kann bei der *Entwicklung und Wartung* von (verteilten) Anwendungen genutzt werden
- Wird mit dem GNU/Linux-Kernel entwickelt
- Ist noch in der *ersten Entwicklungsphase*
- Eine *GUI fehlt*, jedoch entstehen *schon APIs* für Programmiersprachen

Literatur

Artikel

Edge, J.: Scripting support for perf

<http://lwn.net/Articles/373842>

Edge, J.: Perfcounters added to the mainline

<http://lwn.net/Articles/339361/>

Rebe, R.: Optimierung mit Profil

<http://www.linux-magazin.de/Heft-Abo/Ausgaben/2006/07/Optimierung-mit-Profil>

Projekthompage

collectd: <http://collectd.org>

oprofile: <http://oprofile.sourceforge.net>

perf: <https://perf.wiki.kernel.org/>

Internet

[GZG01]

Grelk, K.; Zilm, T.; Günther, K.: *Linux-Befehlsreferenz zur Systemadministration*, mitp-Verlag, Bonn, 2001.

Danke!

