

Sensordaten mit SNMP verteilen

Axel Wachtler und Ralf Findeisen

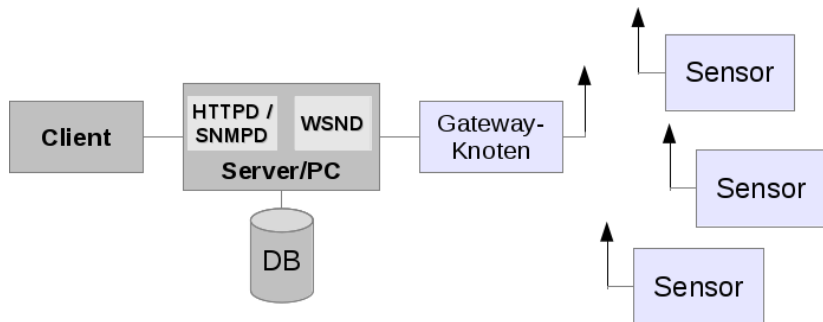
CHEMNITZER LINUX TAGE

17.03.2013

Einleitung

- ▶ Systembeschreibung
- ▶ Was ist SNMP?
- ▶ Implementierung
- ▶ Demo
- ▶ Ausblick

Systemüberblick



- ▶ Sensor- und Gatewayknoten z.B. über Funk verbunden
- ▶ serielles Interface zwischen Gatewayknoten und PC/Server
- ▶ WSN-Prozess speichert Daten vom seriellen IF in Datenbank
- ▶ Web- oder SNMP-Prozess liest angefragte Daten aus DB

Webserver-Ansatz

System CLT-2011

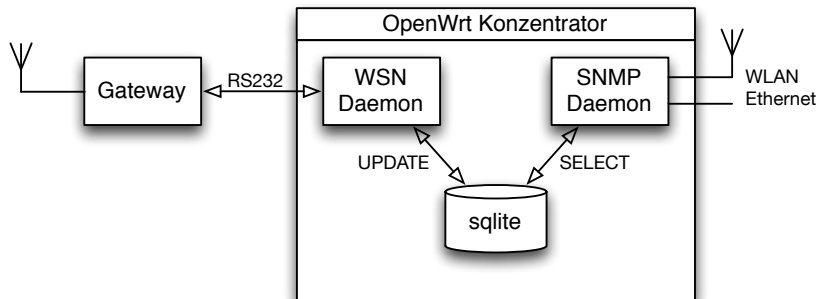
- ▶ Python Webserver/CGI-Script und Anbindung an RRD.
<http://cgi.uracoli.de/cgi-bin/uriah.cgi>

Bewertung

- ▶ aller Anfang ist *einfach*
- ▶ **ausreichend** für **kleine und einfache** Systeme
- ▶ aufwändige **Webprogrammierung**
- ▶ aufwändige **Layout-Gestaltung**
Anpassung für verschiedene Browser und Endgeräte
- ▶ nur **Webbrowser** als Client verfügbar

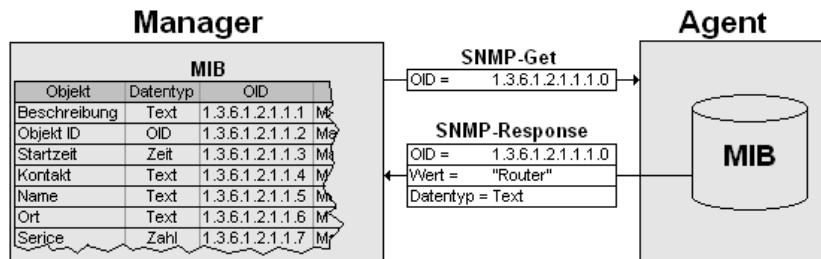
Vorteile eines SNMP basierten Systems

- ▶ SNMP wird als **generischer Zwischenserver** verwendet
- ▶ verschiedene Clienten von **Monitoringsystem** bis **Desktop-Programm** oder **App** verfügbar.
- ▶ Systeme sind einfach erweiterbar und selbst konfigurierend
- ▶ die Arbeit steckt in der **Konfiguration** statt Programmierung



SNMP

- ▶ SNMP = Simple Network Management Protocol
- ▶ IETF-Protokoll zur Überwachung/Steuerung von Netzwerkelementen (z.B. Router, Server, ...)
- ▶ Transportprotokoll UDP (Ports :161 und :162)
- ▶ Primitiven: snmpget, snmpgetnext, snmpset, snmptrap



SNMP-Abfrage, Quelle: Wikipedia:SNMP

Datenorganisation

- ▶ MIB = Management Information Base
- ▶ MIB ist virtuelle DB, wird vom Agenten implementiert
- ▶ Informationen in MIB werden über OIDs (Object Identifier) adressiert
- ▶ OIDs sind baumförmig organisiert

```
iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1)
entityMIBObjects          1.3.6.1.2.1.47
+-entityPhysical          1.3.6.1.2.1.47.1.1
  +-entPhysicalTable      1.3.6.1.2.1.47.1.1.1
    +-entPhysicalEntry    1.3.6.1.2.1.47.1.1.1.1
      +- entPhysicalIndex  1.3.6.1.2.1.47.1.1.1.1.1
      +- entPhysicalDescr  1.3.6.1.2.1.47.1.1.1.1.2
```

Zugriff auf die MIB

- ▶ `snmpget` (1)
liest einen OID-Wert
- ▶ `snmpgetnext` (1)
liest den nächsten OID-Wert
- ▶ `snmpset` (1)
beschreibt einen OID-Wert
- ▶ `snmpwalk` (1)
liest alle OID-Werte unterhalb des Start-OIDs
- ▶ `snmptrap` (1)
sendet Nachricht an SNMP-Manager

Beispiele

```
# OPTS=-0s -c pub -v2c 192.168.235.128

# snmpget ${OPTS} .1.3.6.1.2.1.47.1.1.1.1.2.1
mib-2.47.1.1.1.1.2.1 = STRING: "temperature at node 0x1"

# snmpset ${OPTS} .1.3.6.1.2.1.47.1.1.1.1.2.1 string foo
mib-2.47.1.1.1.1.2.1 = STRING: "foo"

# snmpwalk ${OPTS} ENTITY-SENSOR-MIB:entPhySensorTable
entPhySensorType.1 = INTEGER: celsius(8)
entPhySensorType.2 = INTEGER: voltsDC(4)
...
```

ENTITY-MIB

- ▶ Entity MIB (Version 3), RFC4133
- ▶ beschreibt die **physikalischen** Eigenschaften eines Sensors
- ▶ Prefix: .1.3.6.1.2.1.47.1.1.1.1.1.x.y
x = Variable, y = Device-Nummer

1	entPhysicalIndex	9	entPhysicalFirmwareRev
2	entPhysicalDescr	10	entPhysicalSoftwareRev
3	entPhysicalVendorType	11	entPhysicalSerialNum
4	entPhysicalContainedIn	12	entPhysicalMfgName
5	entPhysicalClass	13	entPhysicalModelName
6	entPhysicalParentRelPos	14	entPhysicalAlias
7	entPhysicalName	15	entPhysicalAssetID
8	entPhysicalHardwareRev	16	entPhysicalIsFRU

ENTITY-SENSOR-MIB

- ▶ Entity Sensor Management Information Base, RFC3433
- ▶ beschreibt die **Messwerte** eines Sensors
- ▶ Prefix: .1.3.6.1.2.1.99.1.1.1.x.y
x = Variable, y = Device-Nummer

- 1 entPhySensorType
- 2 entPhySensorScale
- 3 entPhySensorPrecision
- 4 entPhySensorValue
- 5 entPhySensorOperStatus
- 6 entPhySensorUnitsDisplay
- 7 entPhySensorValueTimeStamp
- 8 entPhySensorValueUpdateRate

Installation

(an)getestete Systeme

- ▶ OpenWRT (auf Carambola)
- ▶ Debian Wheezy (Raspberry Pi)
- ▶ Ubuntu 12.04 (PC, VmWare)

Erforderliche Module

- ▶ snmpd
- ▶ Lua (5.1.4) und
 - ▶ lua-sqlite3
 - ▶ luars232
 - ▶ luaposix
- ▶ serieller Kernel-Treiber für den Gateway-Knoten

Software

`wsnmp.lua` - das SNMPD-Pass-Script

- ▶ wird für beide OID-Bäume in `snmpd.conf` eingetragen
- ▶ Aufruf mit `lua wsnmp.lua [-g|-s] OID [type, value]`
- ▶ Implementierung `snmpset`
 - ▶ `wsnmp.lua` schreibt Daten in File/Fifo und sendet Signal `SIGUSR1` and `wsnd`
 - ▶ `wsnd` empfängt das Signal und schreibt Daten in DB.

`wsnd.lua` - der WSN-Dämon

- ▶ wird als Systemprozess gestartet
- ▶ parst Zeilen von der seriellen Schnittstelle
- ▶ schreibt Daten in die sqlite3 Datenbank
- ▶ Verarbeitung von `SIGUSR1` (`snmpset`)

Die sqlite3-Datenbank

- ▶ realisiert den kollisionsfreien Zugriff von wsnd und snmpd
- ▶ der **letzte Messwert-Wert** sowie **statische Informationen** des Sensors werden gespeichert.
- ▶ Die *Langzeitarchivierung* der Messdaten erfolgt im **Managment-System**.
- ▶ sqlite3 hat nur **einen globalen Datenbank-Lock**
 - ▶ es kann nur ein Prozess in der DB schreiben (INSERT, UPDATE)
 - ▶ aufwändiges BUSY-Handling, deshalb snmpset Implementierung mittels File/Fifo Schnittstelle und Signal
- ▶ derzeit werden alle Daten in einer MIB-Tabelle gespeichert

SNMP-Frontends

Web-basierte Monitoringsysteme

- ▶ Ein standardisiertes Protokoll erlaubt die Nutzung in vielfältiger Umgebung
- ▶ Managementlösungen wie Nagios, Icinga, Zenoss, ...
- ▶ EMail, SMS-Benachrichtigung vom Managementserver
- ▶ Kombination mit anderen Dienstinformationen

Weitere Ideen und Applikationen

- ▶ SNMP in Scriptingumgebungen
- ▶ Benachrichtigungen mit Feeds, etc. auf mobile Geräte
- ▶ Desktopinformationen via Desktopwidgets (z.B. Conky)

Beispiel: Icinga - der Konzentrator als Host

The screenshot displays the Icinga web interface. At the top, there are status indicators for various states: 2 UP, 0 DOWN, 0 UNREACHABLE, 0 PENDING, 0/2 TOTAL; 0 OK, 0 WARNING, 0 CRITICAL, 0 UNKNOWN, 0 PENDING, 0/9 TOTAL. The Icinga logo is visible on the right, along with system statistics: 2/0/0, 0.01/0.02/0.015 s, 8/0/0, 0.01/0.03/0.015 s, 0.04/0.09/0.062 s, 0.10/0.23/0.170 s.

The left sidebar contains a navigation menu with sections: **All Services OK**, **General** (Home, Documentation, Search), **Status** (Tactical Overview, Host Detail, Service Detail, Hostgroup Overview, Hostgroup Summary, Servicegroup Overview, Servicegroup Summary, Status Map), **Problems** (Service Problems, Unhandled Services, Host Problems, Unhandled Hosts, All Unhandled Problems, All Problems, Network Outages), **System** (Comments, Downtime, Process Info, Performance Info, Scheduling Queue), **Reporting** (Trends, Availability, Alert Histogram, Alert History, Alert Summary, Notifications, Event Log), and **Configuration** (View Config).

The main content area shows **Current Network Status** (Last Updated: Fri Mar 15 10:52:03 PDT 2013 - Update in 85 seconds [pause]), **Commands for checked services** (Select command dropdown, Submit button), and **Service Status Details For All Hosts** (Page 1 of 1, Results: 50).

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Concentrator	Disk Space	OK	2013-03-15 10:51:15	5d 0h 9m 42s	1/4	DISK OK
	Temperature	OK	2013-03-15 10:51:49	0d 1h 10m 14s	1/4	SNMP OK - 2210
localhost	Current Load	OK	2013-03-15 10:48:43	13d 2h 33m 0s	1/4	OK - load average: 0.00, 0.01, 0.05
	Current Users	OK	2013-03-15 10:49:57	13d 2h 32m 10s	1/4	USERS OK - 1 users currently logged in
	Disk Space	OK	2013-03-15 10:51:11	13d 2h 31m 20s	1/4	DISK OK
	HTTP	OK	2013-03-15 10:49:38	13d 2h 30m 30s	1/4	HTTP OK: HTTP/1.1 200 OK - 453 bytes in 0.001 second response time
	SSH	OK	2013-03-15 10:50:52	12d 23h 36m 40s	1/4	SSH OK - OpenSSH_5.5p1 Debian-Subuntu1 (protocol 2.0)
	Total Processes	OK	2013-03-15 10:51:05	13d 2h 28m 50s	1/4	PROCS OK: 88 processes
ido2db Process	OK	2013-03-15 10:45:34	13d 2h 29m 7s	1/3	PROCS OK: 2 processes with command name 'ido2db'	

At the bottom of the table area, it says "Displaying Result 1 - 9 of 9 Matching Services". The URL at the bottom of the browser window is `ubuntu/cgi-bin/icinga/status.cgi?host=all&style=detail&servicestatustypes=2`.

Icinga Hosts

Beispiel: Icinga - Temperaturservice

2 UP 0/0/0 DOWN 0/0/0 UNREACHABLE 0 PENDING 0/2 TOTAL

9 OK 0/0/0 WARNING 0/0/0 CRITICAL 0/0/0 UNKNOWN 0 PENDING 0/9 TOTAL

ICINGA 2/0/0 0.01/0.01/0.011 s 9/0/0 0.05/0.07/0.062 s 0.01/0.03/0.014 s 0.07/0.23/0.134 s

General

- Home
- Documentation
- Search:

Status

- Tactical Overview
- Host Detail
- Service Detail
- Hostgroup Overview
- Hostgroup Summary
- Servicegroup Overview
- Servicegroup Summary
- Status Map

Problems

- Service Problems
- Unhandled Services
- Host Problems
- Unhandled Hosts
- All Unhandled Problems
- All Problems
- Network Outages

System

- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue

Reporting

- Trends
- Availability
- Alert Histogram
- Alert History
- Alert Summary
- Notifications
- Event Log

Configuration

- View Config

Service Information

Last Updated: Fri Mar 15 10:54:01 PDT 2013 - Update in 74 seconds [jason]

Icinga 1.8.4 - Logged in as kingadmin

- View Information For This Host
- View Service Status Detail For This Host
- View Alert History For This Service
- View Trends For This Service
- View Alert Histogram For This Service
- View Availability Report For This Service
- View Notifications For This Service
- View Scheduling Queue For This Service
- View Config For This Service

Service State Information

Current Status: **OK** (for 0d 1h 2m 12s)

Status Information: SNMP OK - 2210

Performance Data: iso.3.6.1.2.1.99.1.1.1.4.1*2210

Current Attempt: 1/4 (HARD state)

Last Check Time: 2013-03-15 10:51:49

Check Type: ACTIVE

Check Latency / Duration: 0.183 / 0.032 seconds

Next Scheduled Check: 2013-03-15 10:56:49

Last State Change: 2013-03-15 09:41:49

Last Notification: N/A (notification 0)

Is This Service Flapping? **NO** (4.01% state change)

In Scheduled Downtime? **NO**

Last Update: 2013-03-15 10:53:53 (0d 0h 0m 8s ago)

Modified Attributes: None

Active Checks: **ENABLED**

Passive Checks: **ENABLED**

Obsessing: **ENABLED**

Notifications: **ENABLED**

Event Handler: **ENABLED**

Service

Temperature

On Host Concentrator

(Concentrator)

Member of **No servicegroups.**

Service Dependencies

192.168.109.1

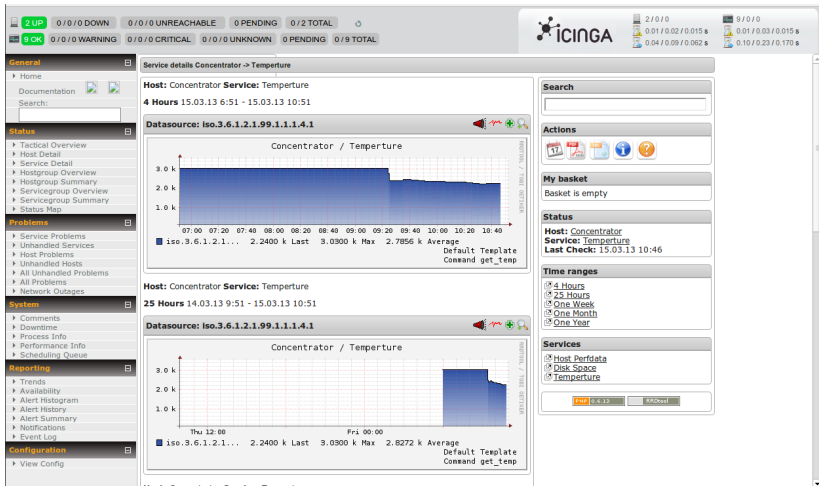
Service Commands

- Disable active checks of this service
- Re-schedule the next check of this service
- Submit passive check result for this service
- Stop accepting passive checks for this service
- Stop obsessing over this service
- Disable notifications for this service
- Send custom service notification
- Schedule downtime for this service
- Disable event handler for this service
- Disable flap detection for this service
- Add a new service comment
- Reset Modified Attributes

Extra Actions

Icinga Service

Beispiel: Icinga Graph mit PNP4Nagios



PNP4Nagios Graph

Demo

- ▶ Daten auf der seriellen Schnittstelle
`cat /dev/ttyUSB0`
- ▶ Starten wsnd
`tail -f wsnd.log`
- ▶ direkter Aufruf wsnmp.lua
`tail -f wsnmp.log`
- ▶ `snmpget`, `snmpset`, `snmpwalk`
- ▶ Icinga

Zusammenfassung

Erreichtes

- ▶ initiales Framework für die SNMP-Anbindung eines Sensornetzwerkes
- ▶ einige 100 Zeilen Lua und Python Code und jede Menge gelesene Dokumentation

Ausblick

- ▶ Reduzierung der Systemlast auf embedded Plattformen
- ▶ Implementierung eines pass_persist Scripts für ENTITY-SENSOR-MIB
- ▶ Nutzung von SNMP-Traps (z.B. bei Anmeldung oder Reboot von Sensoren)
- ▶ Test weiterer Frontends
- ▶ Untersuchung der Skalierbarkeit in grossen Sensornetzen

Zum Schluß ...

Vielen Dank für die Aufmerksamkeit!

Weitere Fragen?

Die Folien gibt es auf <http://www.uracoli.de/clt2013>.