SysDB – System DataBase

a system management and inventory collection service

Sebastian 'tokkee' Harl
<**sh@tokkee.org**>

SysDB developer

Open Source Data Center Conference 2014
April, 10 2014
Berlin

Disclaimer

WARNING:
SysDB is still under heavy development.

Flaming, bashing or other forms of constructive feeback are very appreciated :-)

Background / Motivation

Motivation

Why SysDB?

- Who uses Nagios/Icinga/Naemon/OpenNMS/etc.?

Motivation

Why SysDB?

- Who uses Nagios/Icinga/Naemon/OpenNMS/etc.?
- Who also uses Puppet/Chef/etc.?

Motivation

Why SysDB?

- Who uses Nagios/Icinga/Naemon/OpenNMS/etc.?
- Who also uses Puppet/Chef/etc.?
- Who also uses collectd/Munin/etc.?

Motivation

Why SysDB?

- Who uses Nagios/Icinga/Naemon/OpenNMS/etc.?
- Who also uses Puppet/Chef/etc.?
- Who also uses collectd/Munin/etc.?
- Who also uses some inventory service?

Motivation

Why SysDB?

- Who uses Nagios/Icinga/Naemon/OpenNMS/etc.?
- Who also uses Puppet/Chef/etc.?
- Who also uses collectd/Munin/etc.?
- Who also uses some inventory service?
- Who also maintains a CMDB?

Motivation

Why SysDB?

- Who uses Nagios/Icinga/Naemon/OpenNMS/etc.?
- Who also uses Puppet/Chef/etc.?
- Who also uses collectd/Munin/etc.?
- Who also uses some inventory service?
- Who also maintains a CMDB?
- Who likes their setup? ;-)

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.
- First idea: group graphs using Puppet facts.

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.
- First idea: group graphs using Puppet facts.
- Second thought: oh, then we can also add monitoring information.

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.
- First idea: group graphs using Puppet facts.
- Second thought: oh, then we can also add monitoring information.
- Then: this should be generic and extensible.

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.
- First idea: group graphs using Puppet facts.
- Second thought: oh, then we can also add monitoring information.
- Then: this should be generic and extensible.
- Then I realized that I don't like web development.

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.
- First idea: group graphs using Puppet facts.
- Second thought: oh, then we can also add monitoring information.
- Then: this should be generic and extensible.
- Then I realized that I don't like web development.
- Also, a system like that would need a good back-end.

History

- Initially, I wanted a new front-end for collectd suitable for large, heterogeneous environments.
- First idea: group graphs using Puppet facts.
- Second thought: oh, then we can also add monitoring information.
- Then: this should be generic and extensible.
- Then I realized that I don't like web development.
- Also, a system like that would need a good back-end.
- → the idea for SysDB was born

The System DataBase

SysDB – Overview

- SysDB collects information about arbitrary hardware and software systems.
- The central idea is to get a central view of your infrastructure.
- Simple examples:
  - Hosts and their attributes (“facts”)
  - Services and their attributes
  - Monitoring information (e.g. current state)
- SysDB collects these information and correlates objects from various backends.

SysDB – core features

- `https://github.com/tokkee/sysdb`
  - CI: `https://travis-ci.org/tokkee/sysdb`
  - $>$ 60% code (function) unit-test coverage in the core
- BSD license
- Written in C
- Easy to extend (simple plugin API)
- Simple network protocol
- Most of the code implemented as a library (reusable)

SysDB – Plugins

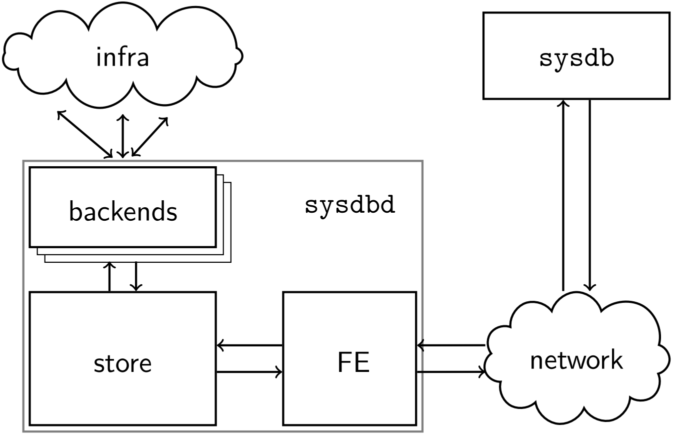
Currently available:

- `collectd::unixsock` – query collectd's UNIX socket interface
- `mk-livestatus` – query Monitoring systems (Nagios, Naemon, Icing, Shinken) using Check_MK Livestatus
- `puppet::store-configs` – query Puppet

- `cname::dns` – canonicalize host-names using DNS
- `syslog` – syslog logging

Planned: Passive data collection (e.g. using Gearman), Foreman, PuppetDB, $your_favorite_system (send patches!) :-)

SysDB – Architecture

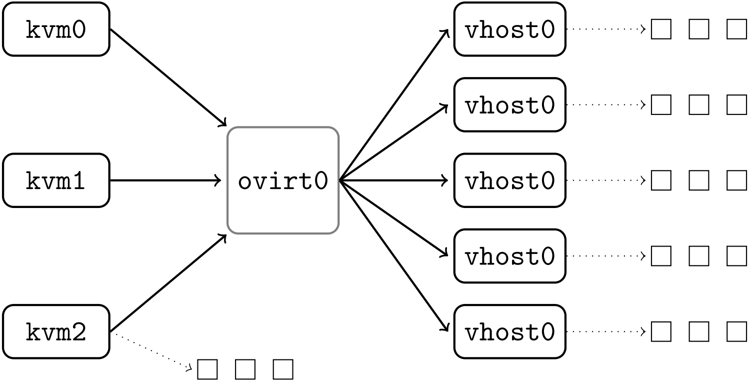


The SysDB Store

- The actual database implementation (in memory)
- Stores generic objects:
  - Host – any kind of physical resource
  - Service – any kind of service
  - Attribute – attributes of hosts and services
- Canonicalization of hosts
- Each object stores the timestamp of the last update and the (automatically calculated) update interval

- Interface to query data
- JSON is the external data representation

The SysDB Store – Examples



Internal Data Structures

```
message Host {
    required string name;
    required DateTime last_update;
    required Interval update_interval;

    repeated Attribute attributes;
    repeated Object children; // Host or Service
    optional Object parent;
}
```

Internal Data Structures

```
message Service {
    required string name;
    required DateTime last_update;
    required Interval update_interval;

    repeated Attribute attributes;
    repeated Object children; // Host or Service
    repeated Object parent;   // multiple!
}
```

Internal Data Structures

```
message Attribute {
    required string name;
    required Type value;
    required DateTime last_update;
    required Interval update_interval;

    optional Object parent;
}
```

Type is either an integer, floating point number, DateTime or binary data.

The SysDB Client

- Interactive client program for SysDB
- Connects to a SysDB daemon
- Interactive command shell
- Receives and displays asynchronous log messages

The SysDB Query Language

- Remotely similar to SQL
- Meant to be easily usable in RPCs
- Still under heavy development ;-)
- Currently supported commands:
  - `LIST` – returns all hosts
  - `FETCH <hostname>` – returns details for a host
  - `LOOKUP hosts WHERE <expression>` – query details for multiple hosts based on their attributes

The SysDB Query Language – Example(I)

```
sysdb=> LIST;
{"hosts":[{
      "name": "nagios.lxc.tokkee.net",
      "last_update": "2014-04-03 10:26:41 +0200",
      "update_interval": "5m4s"
    },{
      "name": "puppet.lxc.tokkee.net",
      "last_update": "2014-04-05 11:04:08 +0200",
      "update_interval": "5m2s"
    },{
      "name": "whisky.mobile.tokkee.net",
      "last_update": "2014-04-05 11:09:15 +0200",
      "update_interval": "10s"
    }]}
```

The SysDB Query Language – Example(II)

```
sysdb=> LOOKUP hosts WHERE attribute.architecture = ’amd64’
                           AND service.name =~ ’postgres’;
{ "name": "whisky.mobile.tokkee.net",
  "last_update": "2014-04-05 11:09:15 +0200",
  "update_interval": "10s",
  "attributes": [{
      "name": "architecture", "value": "amd64",
      "last_update": "2014-04-03 10:26:41 +0200",
      "update_interval": "5m3s" },{ ... }],
  "services": [{
    "name": "cpu-0/cpu-idle",
    "last_update": "2014-03-31 23:34:06 +0200",
    "update_interval": "9.716754301s" },{ ... }]
}
```

Use Cases

Use Cases

- Extend (replace?) your CMDB
  → put the CMDB behind and query it through SysDB
- Compare the back-ends (monitoring)
  → Which hosts / services are missing in a back-end?
  → What is the global status of all Windows systems in some data-center?
- More flexible web-frontend combining multiple back-ends
  → central dashboard

Future Directions

Future Directions

- More documentation!
- RDBMS store back-end (?)
- Interface to query live data (monitoring status, metrics) from backends
- Distributed architecture (HA and load-balancing)
- Web-Interface
- Extend the type system and filters
- Add support for related informationen (?):
  - persons / groups (ACLs, . . .)
  - Events / calendar (Change-Requests, . . .)
- . . .

SysDB – System DataBase

Thank you for your attention!

Questions, comments, rants?

SysDB – System DataBase

Contakt:
Sebastian “tokkee” Harl
<sh@tokkee.org>

https://github.com/tokkee/sysdb

Please send patches! :-)