Zabbix: Choice of 2 of Top 5 Largest French Banks

CORE IT PROJECT

2014 Sept

Presentation

Alain Ganuchaud

Core IT Project

http://www.coreit.fr

alain@coreit.fr

What is this presentation about ?

1. About the glue around Zabbix to make it compatible with large environments
2. Some examples about Large environments issues and how we solved them.
3. About DB Monitoring V7 for Large environments

Summary

Part I

-> The Problem

-> Environment

-> Architectures

Part II

-> Too many Alerts

-> Maintenance Granularity

-> Database size

Part III

-> DB Monitoring V7

Part I : Banking Environment

The Problem

Big companies (specially in the banking sector) have big issues because of :

- number of components
- complexity of human organization managing the servers
- Security, …

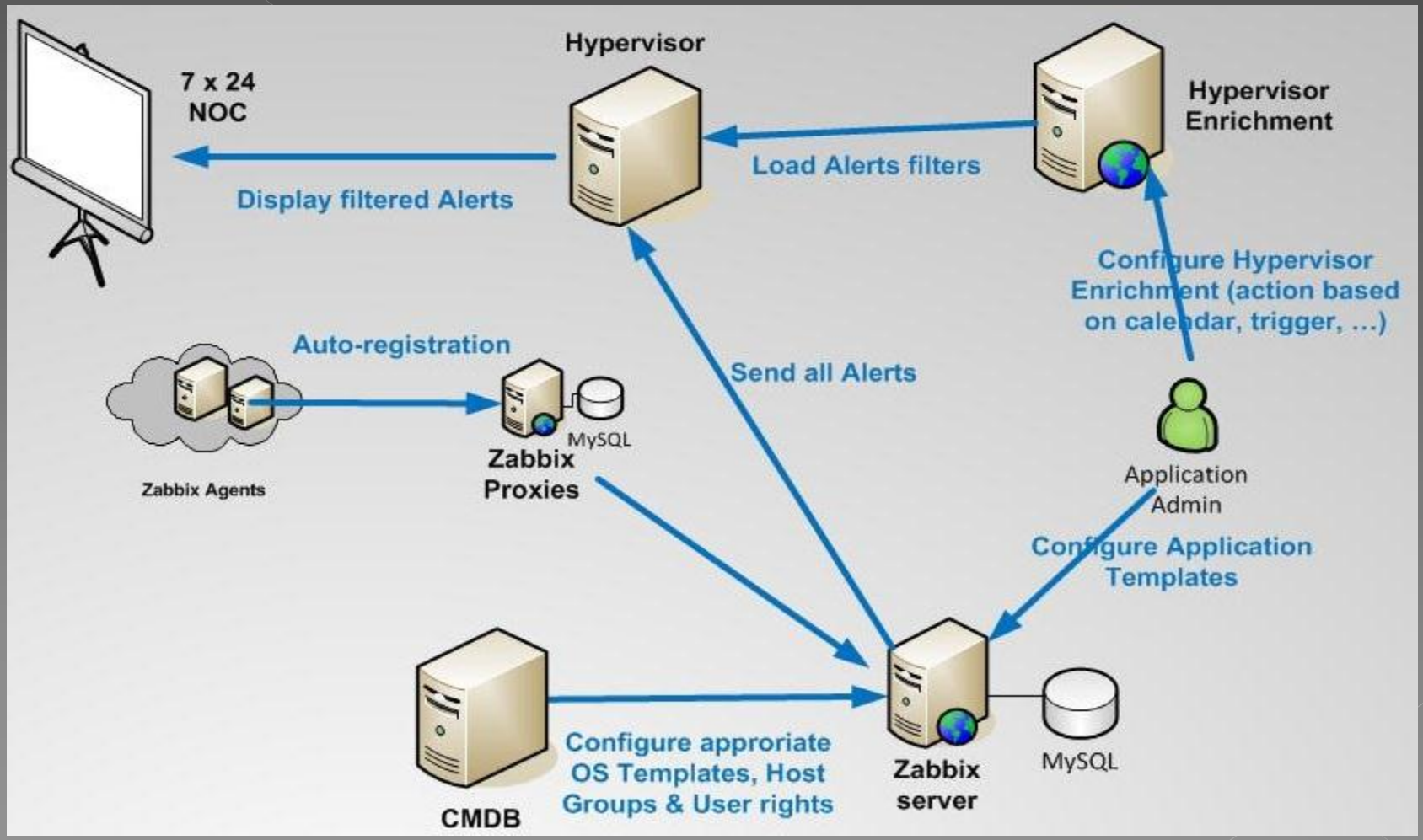
It implies strong constraints on Zabbix platform in terms of :

- Sizing
- Availability (DRP)
- Architecture & performance
- Organization (Human, nomenclature, procedures, …) …

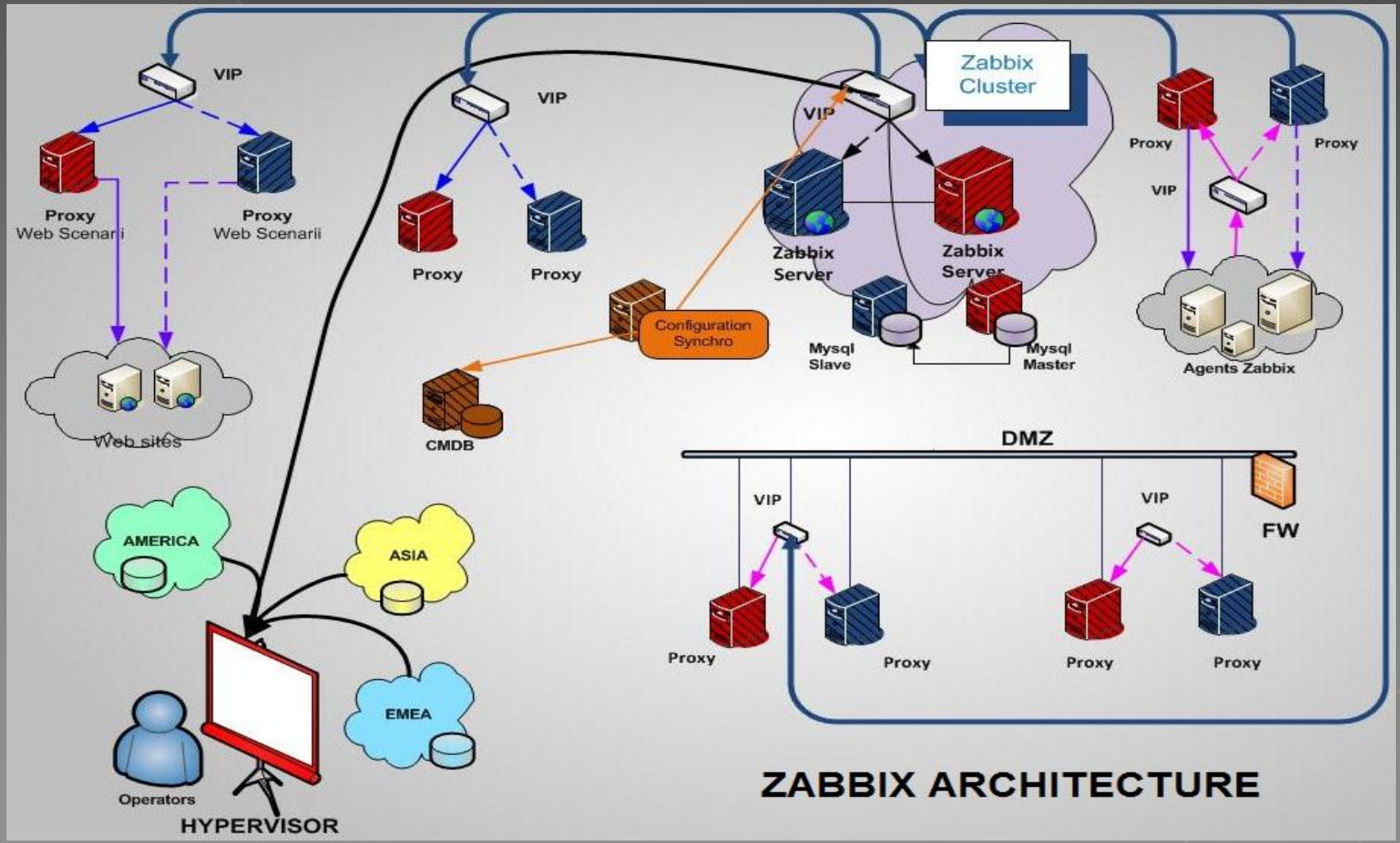
Environment

- More than 14 000 servers , 1 million items
- Some server with over 10 000 items
- Some server with over 10 000 triggers
- More than 10 Teams working on one server, average of 70 simultaneous users connected on the frontend
- More than 5 000 NVS
- Average of 40 000 events by day

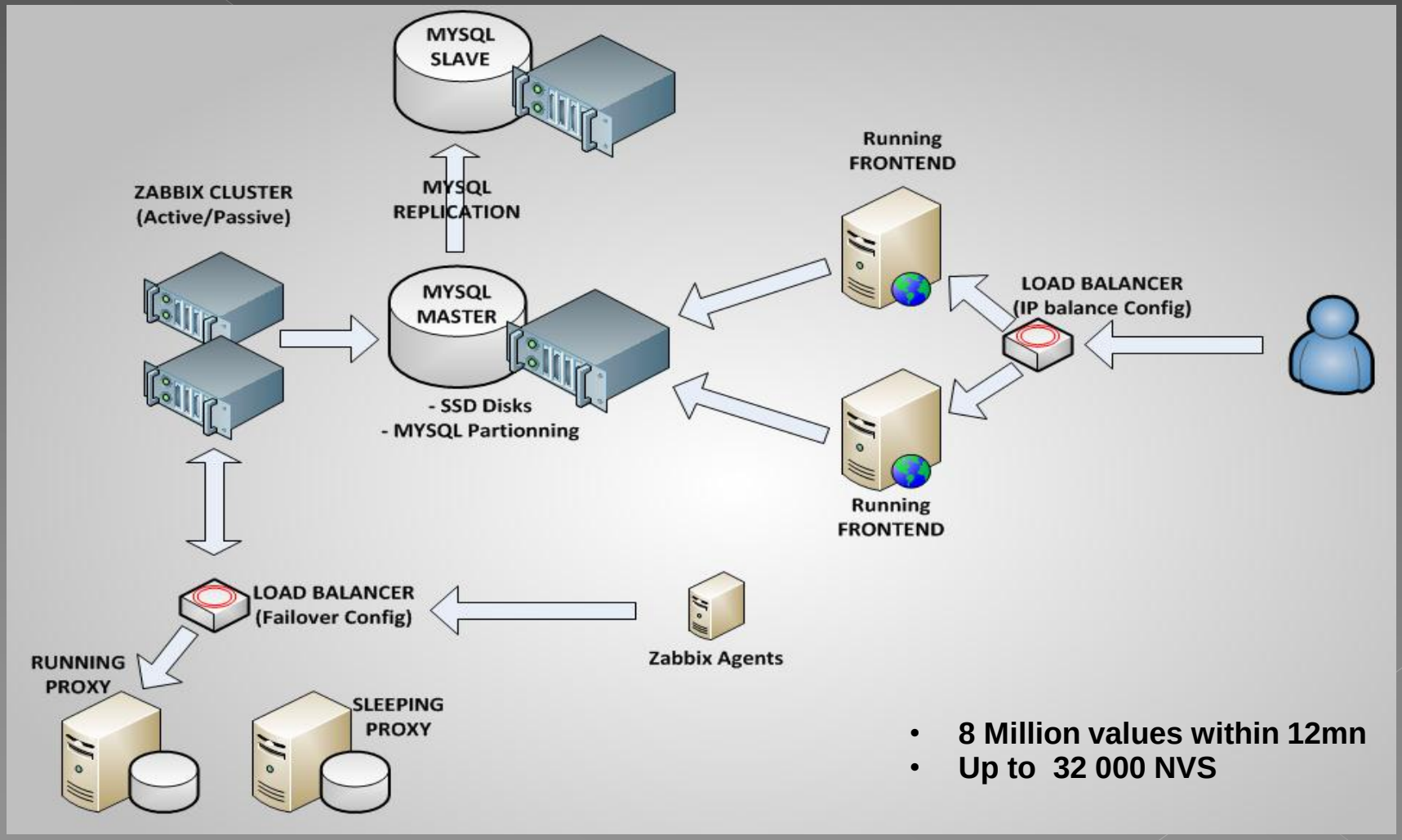
Environment : Big picture



Environment: Zabbix Architecture



Environment: System Architecture



- **8 Million values within 12mn**
- **Up to 32 000 NVS**

Part II : Issues

Issue 1 : Too Many Alerts

Issue 1: Too many alerts

Problem

A server with more than 10 000 triggers can fire too many alerts from several sources: OS, Databases, Applications, ...

Having more than 500 fired triggers in Monitoring / Triggers view for one server is not usable.

Each support Team want only to manage their own triggers: for example, OS Support Team is not interested by Database triggers.

Goal : Simplify the views for the end users which complain they are polluted by useless information.

Solution 1: Filter by inventory fields & application

Overview, triggers & Map Filters

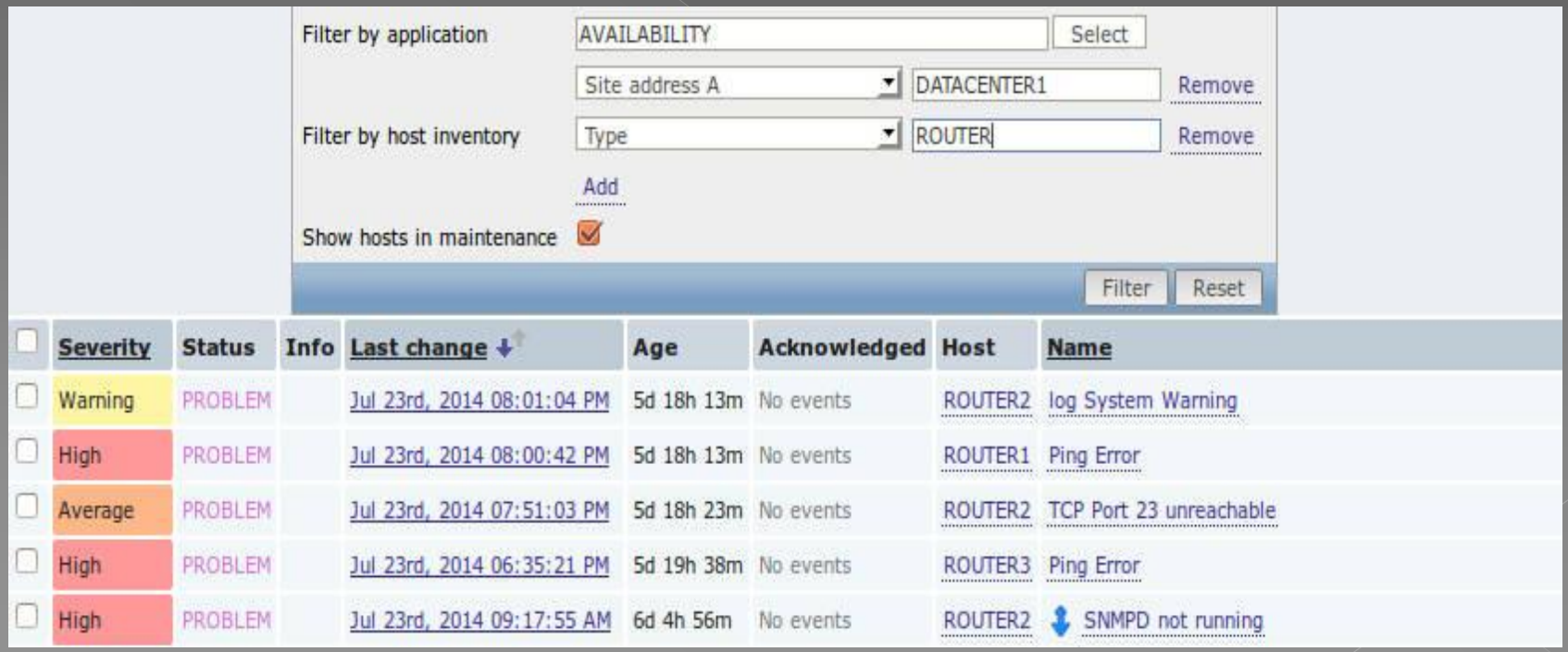
- Filter by host inventory fields
- Filter by Application

Pre-requisites

- Load CMDB infos into Zabbix Host inventory

Solution 1: Filter by inventory fields & application

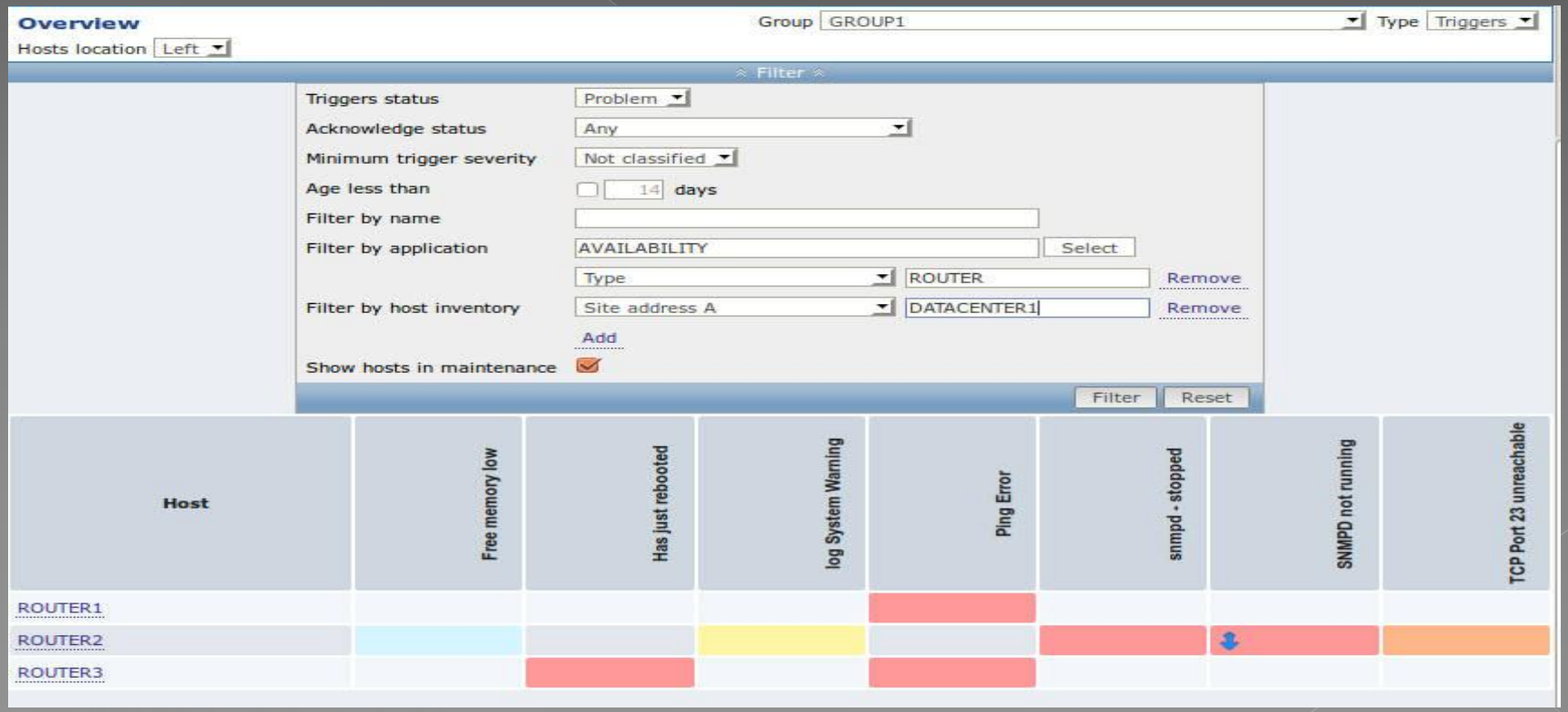
Triggers View: we see all triggers for hosts located in Datacenter1 and which type is Router and which Application is Availability..



Severity	Status	Info	Last change	Age	Acknowledged	Host	Name
Warning	PROBLEM		Jul 23rd, 2014 08:01:04 PM	5d 18h 13m	No events	ROUTER2	log System Warning
High	PROBLEM		Jul 23rd, 2014 08:00:42 PM	5d 18h 13m	No events	ROUTER1	Ping Error
Average	PROBLEM		Jul 23rd, 2014 07:51:03 PM	5d 18h 23m	No events	ROUTER2	TCP Port 23 unreachable
High	PROBLEM		Jul 23rd, 2014 06:35:21 PM	5d 19h 38m	No events	ROUTER3	Ping Error
High	PROBLEM		Jul 23rd, 2014 09:17:55 AM	6d 4h 56m	No events	ROUTER2	SNMPD not running

Solution 1: Filter by inventory fields & application

Overview: we see all triggers for hosts located in Datacenter1 and which type is Router and which Application is Availability..



Solution 1: Filter by inventory fields & application

Overview, triggers & Map Filtering

- Table host_inventory is loaded from CMDB (nightly SQL scripts, fields are matched between CMDB fields and host_inventory fields).
- Triggers View & Overview are filtered by Host_inventory fields and/or Application (Frontend Dev done by Zabbix SIA originally for my customer, release in version 2.4)
- Map component is filtered by Application (Frontend Dev done by Zabbix SIA originally for my customer, release in version 2.4)

Issue 2 : Maintenance Granularity

Issue 2: Maintenance Granularity

Problem

On a server with 20 applications running on it and 20 teams supporting them, maintenance can not be at host level (too high level).

If you put your host in maintenance just only for 1 application, you lose the visibility on the 19 other ones.

Solution 2: Low Level Maintenance

We developped maintenance at trigger level. A Frontend page was developped to manage triggers maintenance, triggers that enter maintenance periods are disabled, triggers that exit maintenance periods are released into unknown state with original status at start date.

Solution 2: Low level Maintenance

host's triggers

Group GROUP1 Host ROUTER1

Caution, status are updated every minute

Filter

	Status	Severity	Description	Status	Maintenance	Maintenance since	Maintenance till
☐	OK	Warning	Disk I/O is overloaded on {HOST.NAME}	enabled	Delete	2014-10-01 18:00:00	2014-10-31 18:00:00
☐	OK	Information	Hostname was changed on {HOST.NAME}	enabled	Schedule	N/A	N/A
☐	OK	Information	Host information was changed on {HOST.NAME}	enabled	Delete	2014-11-11 13:00:00	2014-11-11 14:00:00
☐	OK	Information	{HOST.NAME} has just been restarted	enabled	Schedule	N/A	N/A
☐	OK	Average	Lack of available memory on server {HOST.NAME}	enabled	Schedule	N/A	N/A
☐	OK	Warning	/etc/passwd has been changed on {HOST.NAME}	enabled	Schedule	N/A	N/A

Solution 2: Low Level Maintenance

- A new Frontend Tab (Low Level Maintenance) fills in a new table (trigger_maintenance) with triggers infos: start_status, stop_status, start_date, stop_date, ...
- Triggers are updated according their status at start and stop dates via APIs requests and SQLs
- Triggers are released at stop_date in unknown state and original status at start_date with a comment « Trigger exits maintenance period, will be updated shortly »

Solution 2: Low level Maintenance

Online Demo

Issue 3 : Database size

Issue 3: Database Size

Problem

If you intend to store history & trends over a long period : You are dead! ... because database will contain too much datas and will be overloaded by frontend requests to get back the graphs for long periods.

Solution 3: Dedicated Archiving

We split the trends on another server (has only a RO frontend and no running zabbix_server daemon), we reduced the history store period. We simplified the Archive frontend, RW access are forbidden for admin users.

Solution 3: Dedicated Archiving

Archiving (readonly Tabs)

Solution 3: Dedicated Archiving

Archiving

- On Zabbix platform, trends are kept 3 months, history datas are kept 5 days
- On Archiving platform, trends are copied from Zabbix platform and kept 3 years. History tables are empty. Readonly Frontend is simplified.
- Trends are copied nightly by SQL scripts.
- Deleted hosts on Zabbix platform are not deleted on Archiving platform.

Other Needs

In Frontend Area

- Friendly & concise Frontend for user
- Could be … Saveable custom filters at user or usergroup level, embeddable in dashboard

In Architecture Area

- High Availability solution at Application level
- Distributed Mode Architecture

Part III : Database Monitoring

DB Monitoring V7

Database Monitoring for big companies.

Was designed to monitor Oracle, DB2, MSSQL, Mysql & Postgresql through external checks from zabbix proxies that are configured with database clients.

DB Monitoring V7

Database Monitoring for big companies

- Easy to maintain (SQLs scripts are centralized on proxies, not on 10 000 servers)
- Simple (1 php script and 1 template per engine type)
- V7 brings discovery of Databases instances by LLD
- MSSQL partially based on perfmon

DB Monitoring V7

Pros

- Easy to maintain (centralized)
- Simple (1 php script per engine type)
- Was originally developped with databases experts

Cons

- Externalscripts are big zabbix ressources consummers

DB Monitoring V7

Next versions (beginning of 2015)

- Externalscripts replaced by modules
- Sybase support

Download

- Download on http://www.coreit.fr
- About 2 000 downloads in 2013

Merci pour votre attention !

Alain Ganuchaud

Core IT Project

alain@coreit.fr

Questions?

Follow us on Twitter: @zabbix_fr
@AlainGanuchaud