ZABBIX & High Availability

ZABBIX conference 2012 / Riga

Günther Sommer
IT Architect / „ZABIX Evangelist“
Business Unit Integration Projects

FREQUENTIS

Overview

ZABBIX and High Availability

- → Marketing ☺ – Who are we
- → Part I – The problem
- → Part II – The „standard“ way- Clustering
- → Part III – The „ZABBIX“ way - Distributed monitoring

Marketing ☺

Marketing ☺

About FREQUENTIS & ZABBIX

- → FREQUENTIS is a partner of ZABBIX
- → Using it as a monitoring solution for some of our systems
- → Certified for an ED109 – AL3 environment (with RHEL)

Company Overview

- → Established in 1947
- → 154 Mio. EUR Turnover 2010
- → Corporate headquarters in Vienna
  - Subsidiaries and regional offices in over 50 countries
- → about 980 Employees
- → Outstanding Engineering Capacity
  - more than 600 highly-qualified engineers (HW/SW/PM) at FREQUENTIS headquarter and subsidiaries
- → Export Quota > 90%
- → R&D Quota > 12%


First Air Traffic Control System in Austria, Vienna / Schwechat, 1955


Breakthrough in the US: FAA Command Centre / Herndon, Virginia, 2003


Company Headquarters on Wienerberg, relocation in 2006

Global Market Leader in ATC Voice Communication Systems

FREQUENTIS Worldwide References

[Excerpt 05/2011]

The problem

What means safety critical?

ZABBIX is used in safety critical environments:

→ Has an impact on person safety (ie. trains, airplanes, ...)

→ System is „not allowed“ to fail, this has to be mitigated by design & operation

→ You need to know the state of the system also for later analysis in case of an investigation

→ System being capable to view not just red/green, but also minor/major faults and complex status

Failures

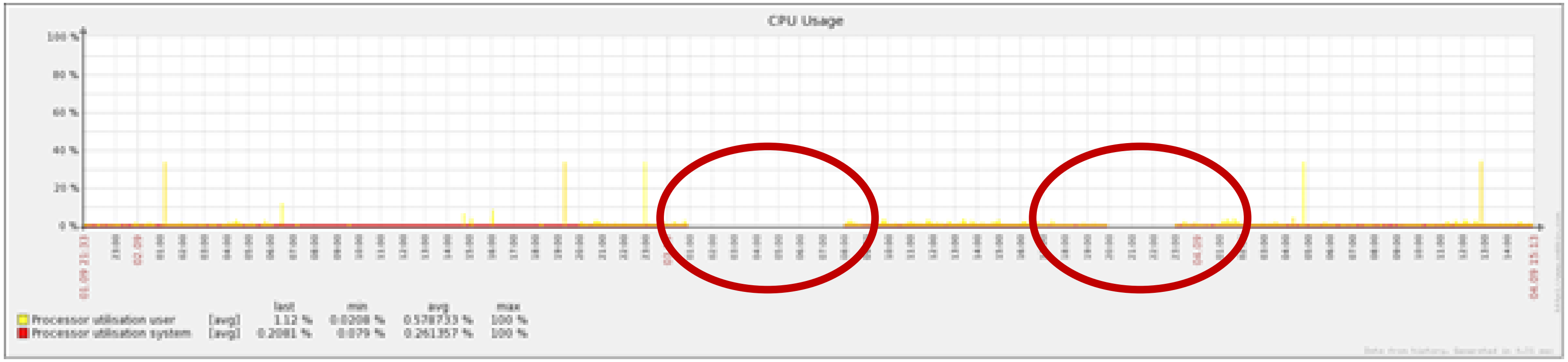
Typical failures:

- → HW fails
- → WAN links drops
- → Power outages

Effects:

- → Failure of monitoring makes system unusable
- → You are „flying blind“, you don‘t know whats effected
- → Can lead to shutdown of complete system, as not in a known state anymore

Monitoring gaps



→ Gaps are in the monitored items

→ What happend in there?

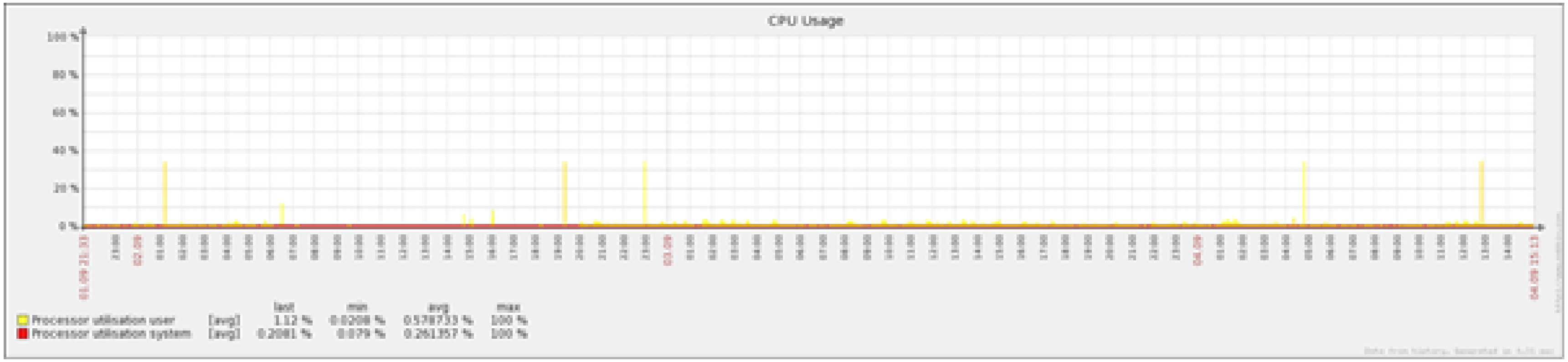
– The fault itself?

– Consequence of fault

– Double fault possible

– Monitoring failure

No gap in monitoring



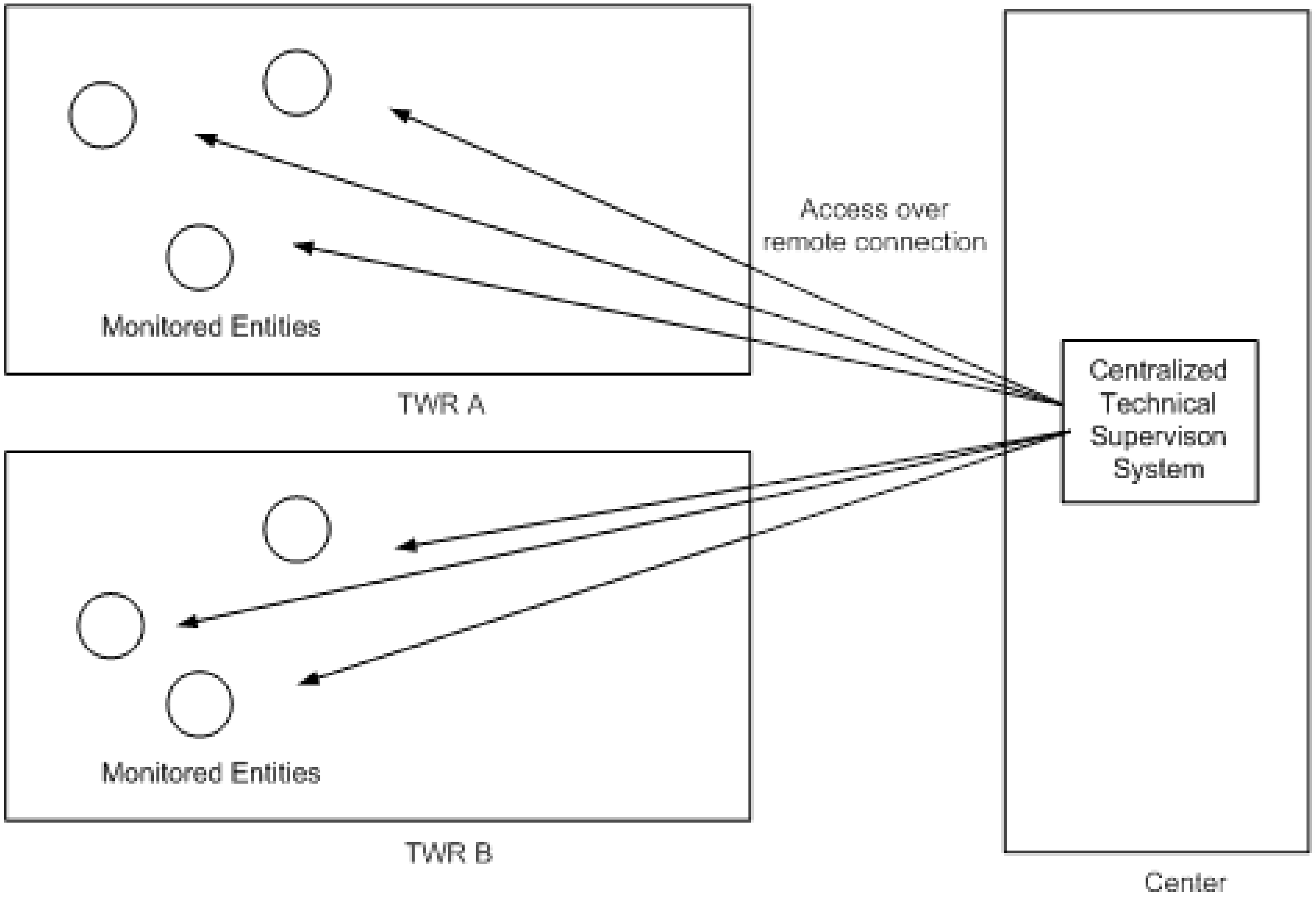
- → The target is to have no gaps at all
- → Doesn‘t have to be gap free immedeatily but at some point in time after a resync
- → Allows a failure analysis „post-mortem“ and to see what was the failure and what where consequences of the failure

The „A“ and the „B“

- Need to avoid the SPOF (single point of failure)
- To solve that problem, the system gets duplicated
- One system is called the A system, the other one the B system
- In case of a failure in the A system, the system automatically switches over to the B system

Part II – the „standard“ way / Clustering

The standard solution



The standard solution (II)

- → Monitoring system is seperate system
- → Make the monitoring system redundant as well
- → High bandwidth usage
- → If system is remote, than a WAN link failure will drop whole site

The simple way? NO!

→ Setup of two ZABBIX instances

→ Both are monitoring, if one fails, the other one still monitors

But:

→ Only allows passive checks (not sure with ZABBIX 2.0)

→ You have to acknowledge it on two systems

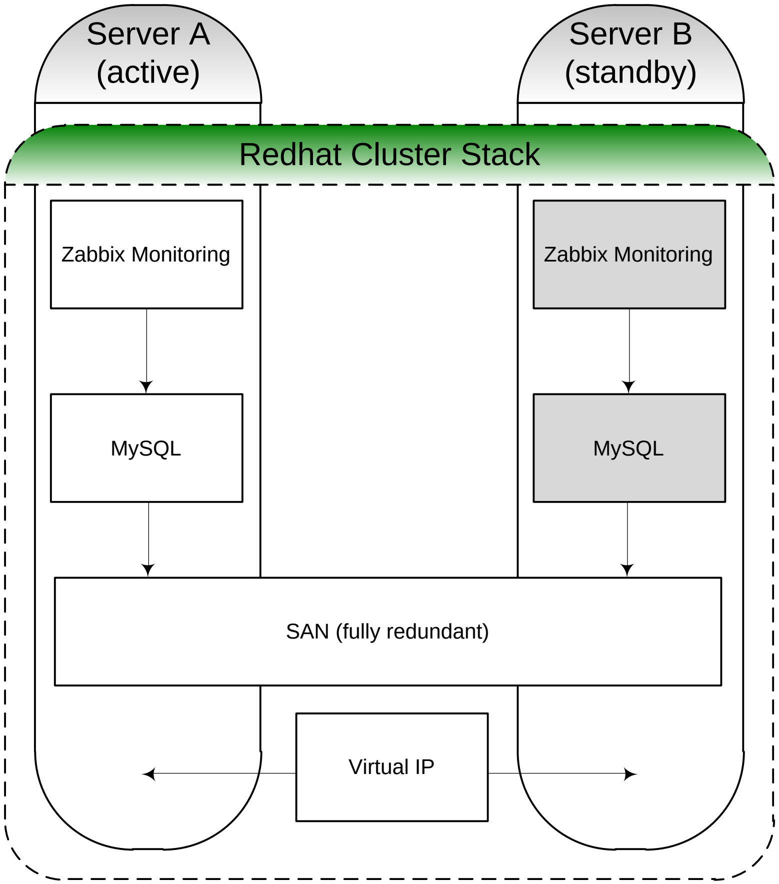
→ They can have different states (as checking on different timestamps)

→ You always look on the wrong one ☺

→ SO: DON‘T DO THAT AT HOME! ☺

SAN based redundancy

- Using a full redundant SAN

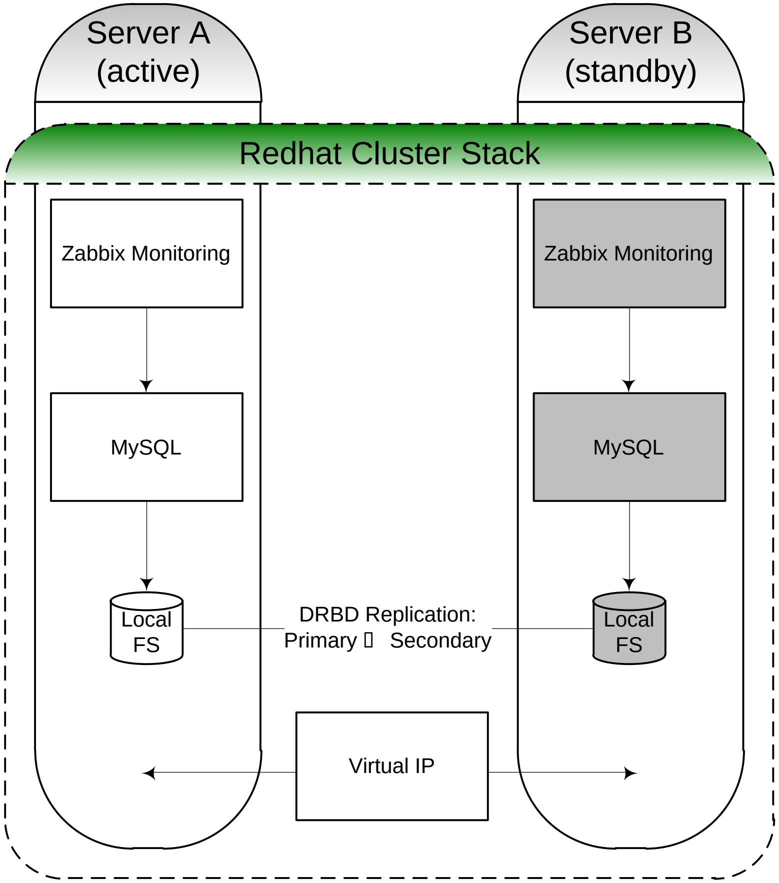


SAN based redundancy (II)

- → No single point of failure, as common SAN storages are now internally fully redundant
- → No sync and resync problems
- → Can have almost have any amount of data
- → But not geo-redundant

Shared nothing architecture

→ Shared nothing architecture



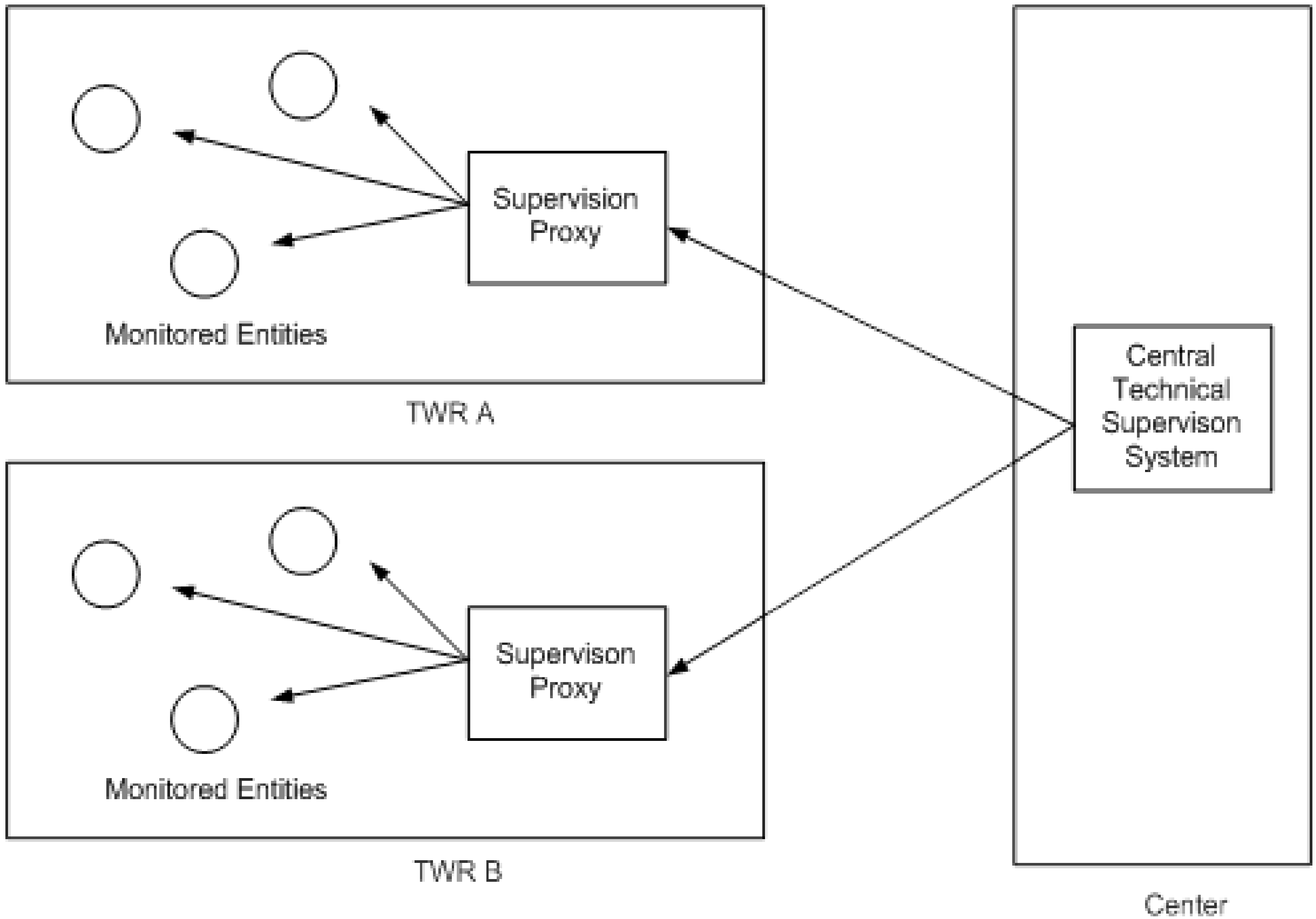
Shared nothing architecture (II)

- Allows operation in two different locations without any common piece of hardware (geo-redundant)
- No single point of failure
- Most complex setup
- Recovery can be tricky (split brain, resync, ...)
- Size of database is limtied due to sync speed
- Requires a lot, lot, lot of testing and tuning

Part III – the „ZABBIX“ way /

Distributed monitoring

The distributed solution



FREQUENTIS

The distributed solution (II)

- Low bandwidth usage as data gets accumulated
- WAN link failure will stop delivering data to central node, BUT it gets queued and stored
- As soon as link comes back, data goes into central data storage
- You have all of your data in one place
- Still each system has it‘s own monitoring system and you can connect to it or use the master node

Node vs. Proxy

ZABBIX has two ways of distributed monitoring:

→ Node – the heavyweight

- „Networked“ full ZABBIX systems which have a master node

→ Proxy – the lightweight

- Only data collector to offload/distribute ZABBIX monitoring item queries

Node – The heavyweight solution

- → The node allows you to have a full ZABBIX server (including web interface) running on the remote site
- → Setup is more complex
- → Needs DB schema changes on all databases
- → Can do everything „on it‘s own“
- → Has it‘s own fully fledged GUI

Proxy – The lightweight solution

- The proxy is only a small piece of SW running, can be co-located on servers
- Easy to install, needs no database, no local configuration
- No node-setup in ZABBIX necessary
- Queues all the data
- But has no GUI

Q & A

→ Any questions ?

Thank you


COMMUNICATION
AND INFORMATION
SOLUTIONS FOR
A SAFER WORLD