

Zabbix Proxy Simulator: The Quest for the simulation of the large IT infrastructures.

Check your configurations, do the scalability testing and all other tidbits of the monitoring.

The Problem:

As a Zabbix Administrator, you are responsible for creation of the valid and reliable monitoring configurations. And while you working on the Groups, Templates, Items, Triggers, LLD's, Discoveries and Actions, it will be obvious to you, at least soon enough, that you are unable to verify, that your creation is actually working. There is no "out of box" solution, which is part of the Zabbix distribution, which can help you with your task.

Common Solutions

In order to assess, that the Monitoring configuration is actually working and doing what it suppose to do, Zabbix Administrators usually resort to do this kind of verifications while simulating failures on the existing infrastructure, temporarily, or permanently dedicated to that.

Common Solutions

In order to assess, that the Monitoring configuration is actually working and doing what it suppose to do, Zabbix Administrators usually resort to do this kind of verifications while simulating failures on the existing infrastructure, temporarily, or permanently dedicated to that.

Do you have a problem with that ?

Actually, I do. First, it is a question of the cost. Not everybody can allocate potentially expensive piece of the software and or hardware to be just the “testbed” for the monitoring configurations. Second, it is oftentimes really difficult to make your hardware, network device or an application to generate you “predictable failures” at your will.

The first step in right direction.

About a year ago, I've come with idea of “Zabbix Agent Simulator”.

This is fairly simple piece of Python code, which will behave exactly like regular Zabbix Passive Agent, and you can ingest your own simulated metrics from “the Agent” to the Zabbix Server

https://github.com/vulogov/zas_agent

The good ...

First, it is working. With little extra work, you can feed your Zabbix server with the data you are manually creating or through scenario generators and/or random number generators.

The good ...

First, it is working. With little extra work, you can feed your Zabbix server with the data you are manually creating or through scenario generators and/or random number generators.

The ugly ...

It is not scalable. If you do need to simulate 100 nodes, you do need to run 100 instances of the Agent simulator.

How to do it right ?

The right solution, for the large-scale Zabbix simulations is to implement simulations at the right logical point: Zabbix proxy.

How to do it right ?

The right solution, for the large-scale Zabbix simulations is to implement simulations at the right logical point: Zabbix proxy.

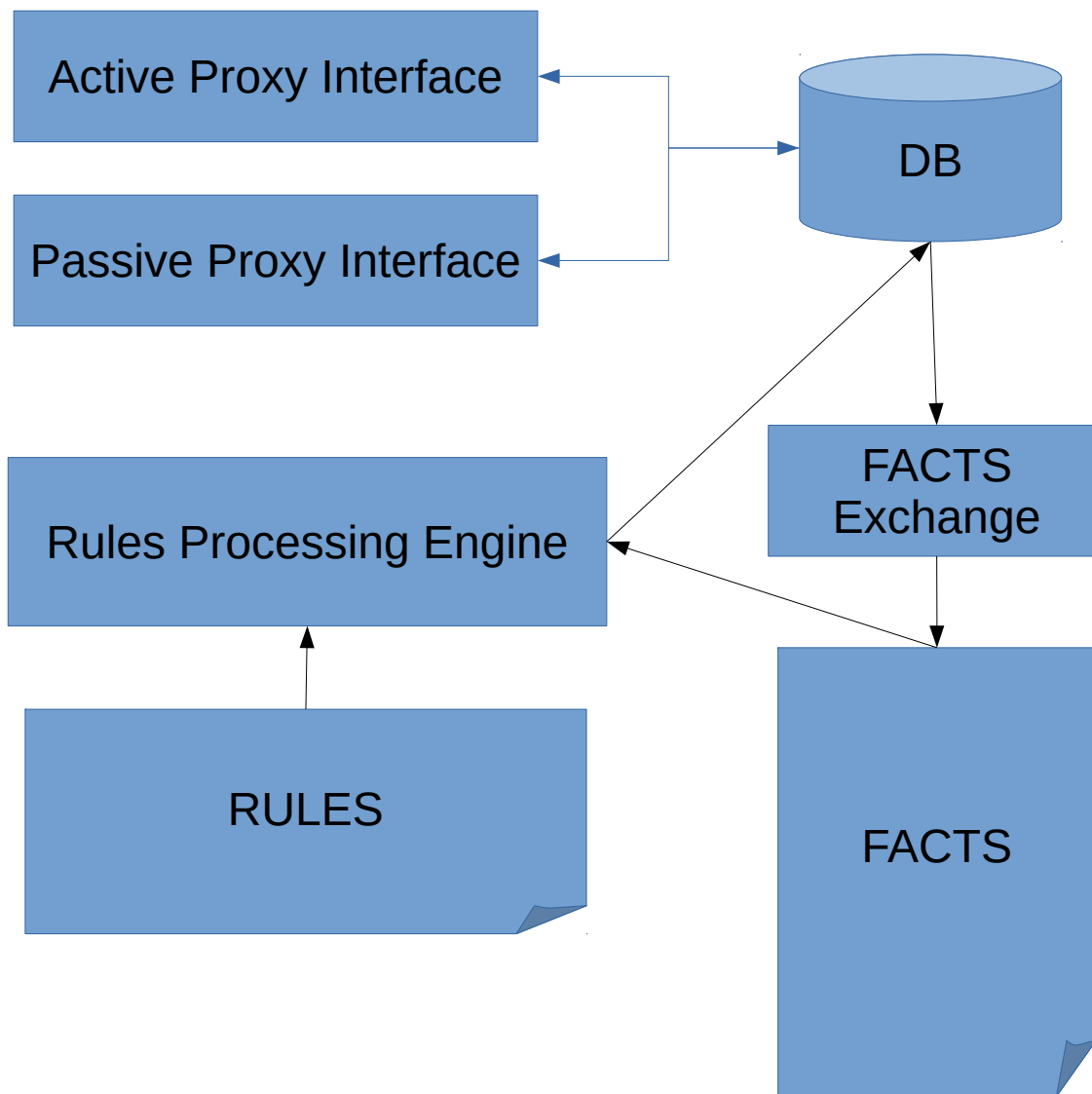
But Zabbix Proxy doesn't support any simulations...

That's why we do need a “Zabbix Proxy Simulator”

Zabbix Proxy Simulator will interface with Zabbix Server through either Active or Passive proxy Interface Module, and store the configuration, allocated for this "Proxy" in the Database.

Then, "FACTS Exchange" process will normalize the configuration and previous data, stored in the database to the list of FACTS

Rules processing Engine will apply the set of the Rules to the list of Facts.





The library and the language CLIPS is at the core of the simulator functionality (although, simulator itself is written in Python). In order to efficiently configure and use simulator, please take a time to read CLIPS documentation. CLIPS will provide you a well thought and well designed way of applying rules to a facts, develop the Expert System and Application using functional and OOP method.

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

What's is the FACT looks like ?

```
(deftemplate ITEM
  (slot host)
  (slot key)
  (slot type)
  (slot oid)
  (slot value)
  (slot stamp)
  (slot freq)
)
```

Example:

```
(assert (ITEM
  (host "Zabbix Server")
  (key "vfs.fs.size[/,pfree]")
  (type "Zabbix Agent")
  (freq 60)
))
```

What's does the RULE looks like ?

```
(defrule set-pfree-dribble-1
  ?item ← (ITEM
    (key "vfs.fs.size[/,pfree]")
    (host ?host)
  )
=>
  (modify ?item
    (value (random 70.0 80.0))
    (stamp (getclock))
  )
)
```

Good things are worth waiting ...

At the time of the Zabbix Conference 2016, the Zabbix Proxy Simulator is not finished as “ready to use” product. There are parts which is still missing, but you can experiment and use Python-CLIPS integration and extension module, which could be found on the GitHub

<https://github.com/vulogov/Clips>