ZABBIX SUMMIT 2023

Internal Changes and Improvements in ZABBIX 7.0

Kaspars Mednis

Chief Trainer, Zabbix, Latvia

ZABBIX SUMMIT 2023

Thread - based Preprocessing

Why Zabbix needs preprocessing

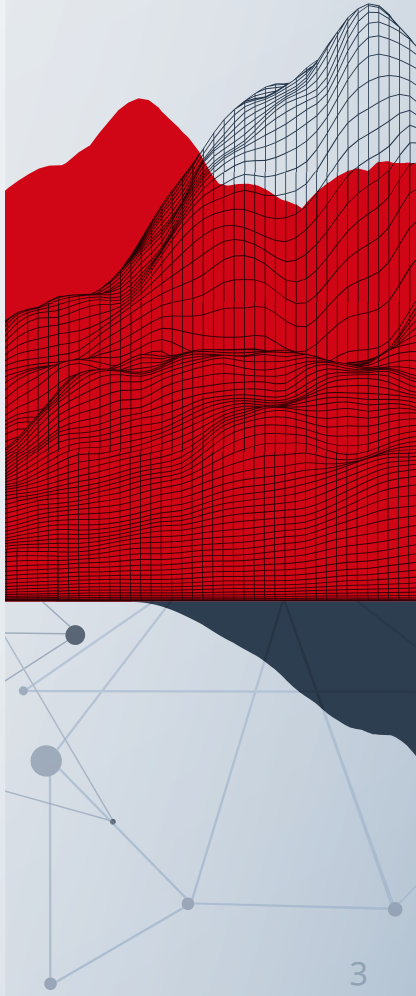
ZABBIX SUMMIT 2023

Collected values may need transformation before they can be used in monitoring:

- Convert bytes to bits and vice versa
- Extract values from JSON, XML, CSV or other data structures
- Create dependent items from bulk data collected by master items

Preprocessing is performed by both Zabbix server and proxies:

- Server can be unloaded by moving some hosts to proxies
- Load can be distributed between multiple proxies

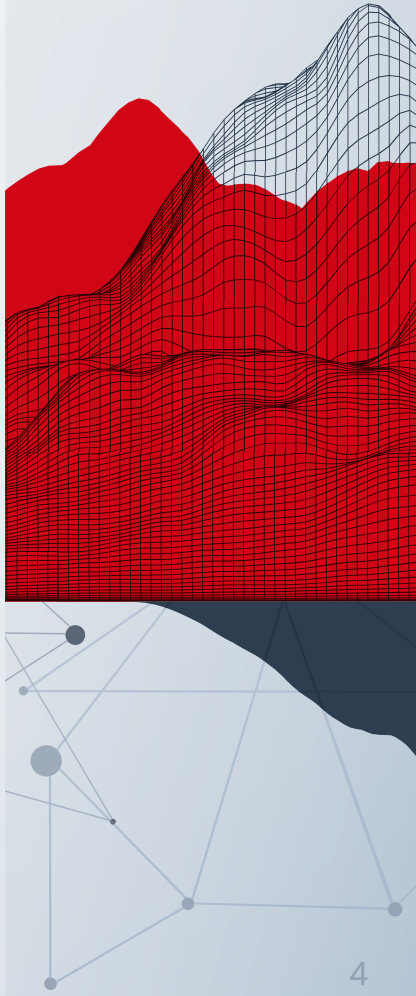


Preprocessing basics

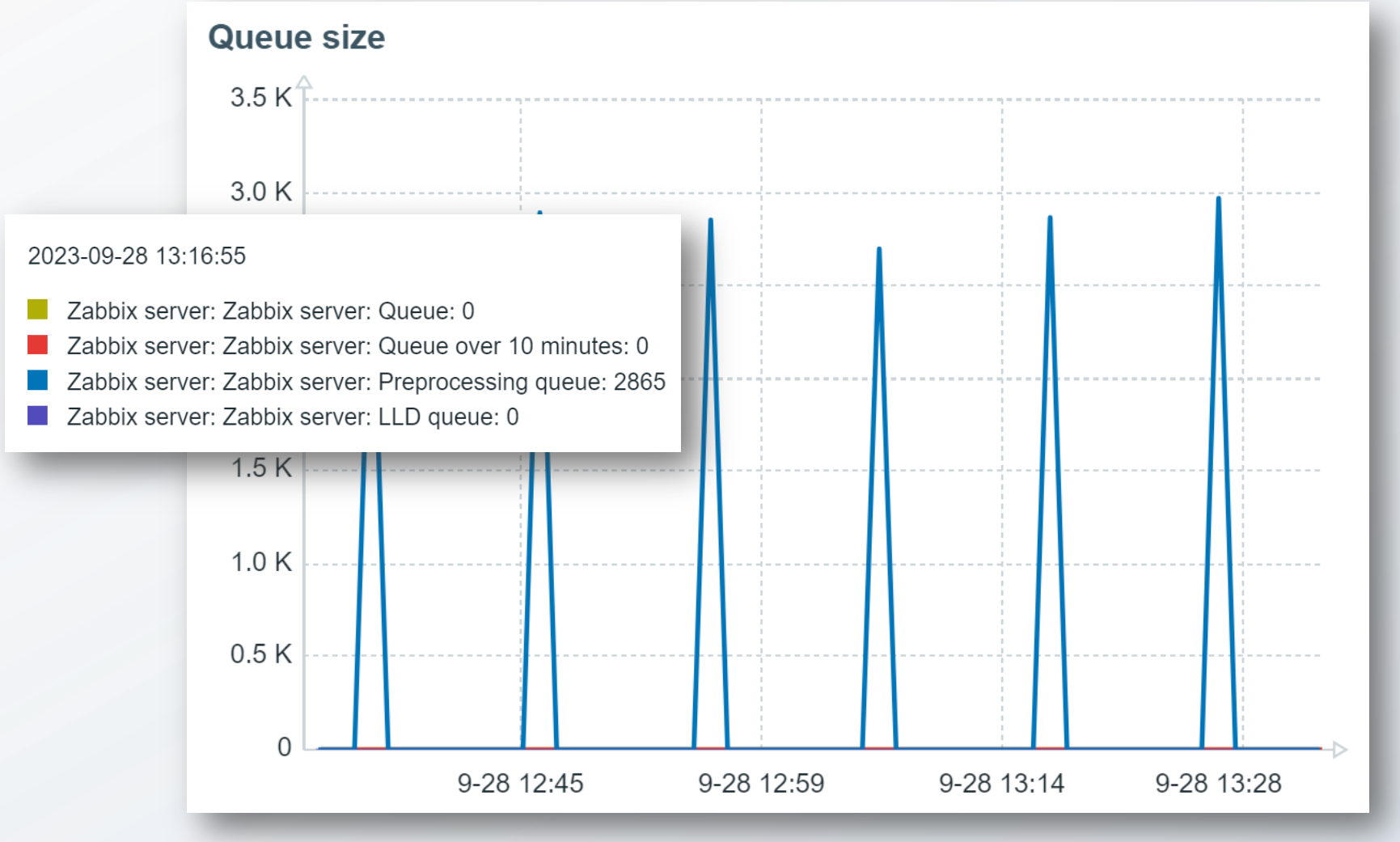
ZABBIX SUMMIT 2023

Every value collected by Zabbix goes through preprocessing

- Preprocessing queue is a FIFO stack
- Some items may have complex preprocessing steps
- It may require significant amount to time to preprocess them.
- In previous versions such items could block other items



Performance issues

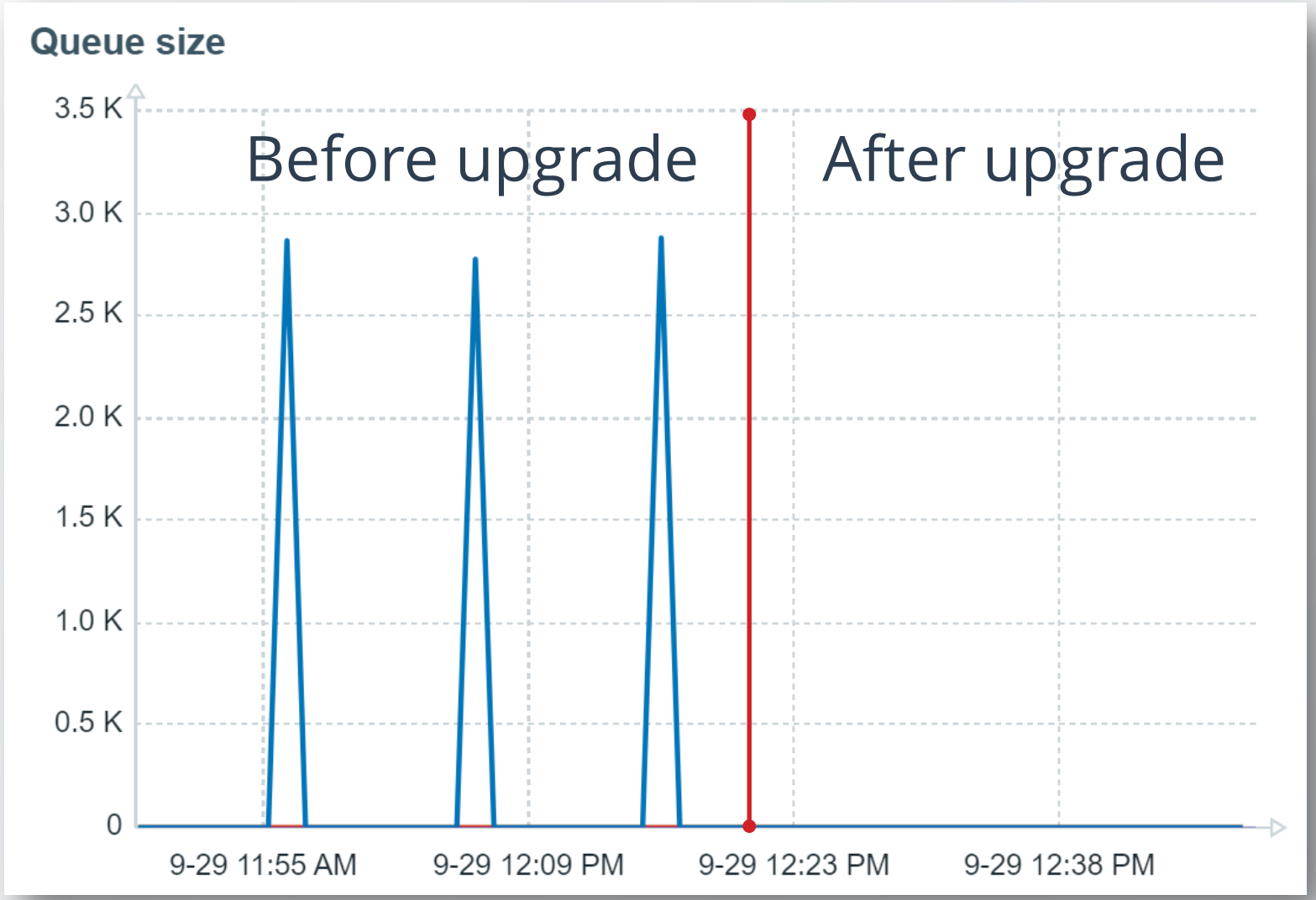


Separate queues

Separate preprocessing queues are introduced since Zabbix 6.4:

- Queue for waiting items
- Queue for processed items

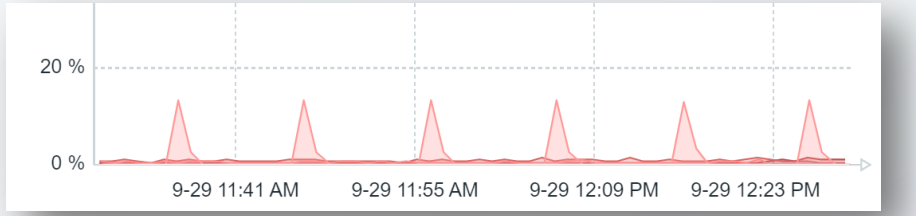
Performance after migration to 7.0



Performance tuning

Make sure you start enough preprocessors:

- The default number if 3
- Start more if required
- All workers work in parallel




```
### Option: StartPreprocessors
# Number of pre-forked instances of preprocessing workers.
# The preprocessing manager process is automatically started when preprocessor
# worker is started.
# Range: 1-1000
# Default: 3

StartPreprocessors=10
```

ZABBIX SUMMIT 2023

Asynchronous data polling

Data collectors

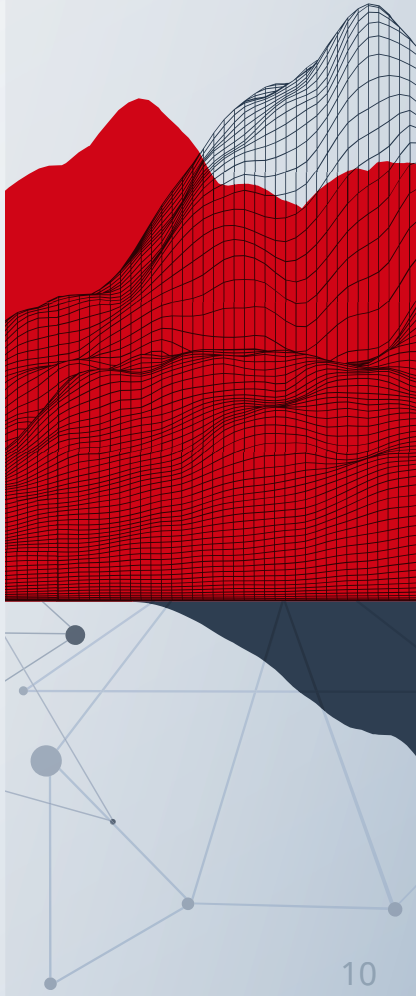
ZABBIX SUMMIT 2023

Data in Zabbix are collected by various data collectors:

- Pollers
- Trappers
- Pingers
- etc.

Poller processes can collect only a single metric at once:

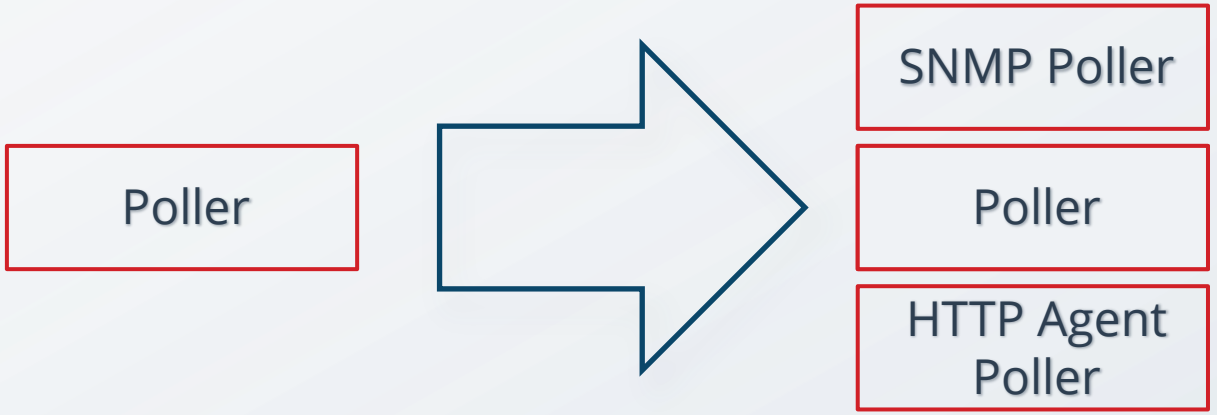
- As a result, hundreds of pollers are required in some scenarios
- Maximum number of pollers is limited to 1000



Specific poller types

In Zabbix 7.0 specific poller types are introduced

- SNMP poller
- Zabbix agent poller
- HTTP check poller

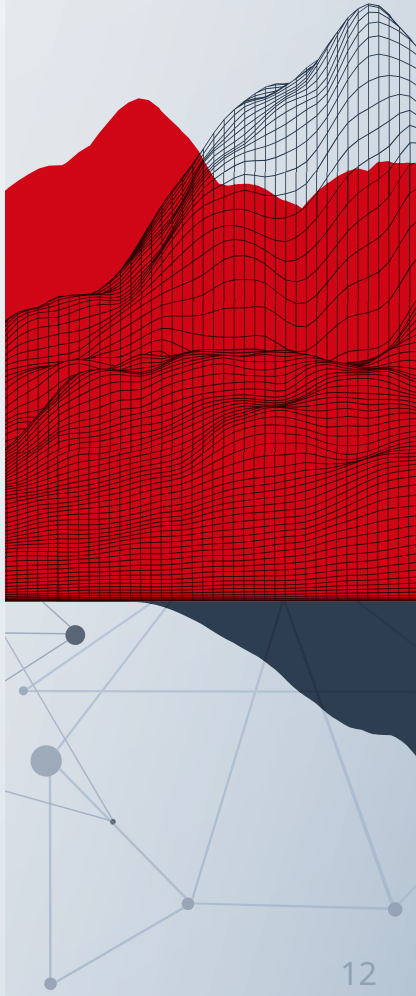


SNMP poller

SNMP poller collects data using the SNMP protocol:

- Uses SNMP protocol over UDP to collect data
- Optimized to work with network devices (SNMPGET, SNMPBULKGET)

```
### Option: StartSNMPPollers
#       Number of pre-forked instances of asynchronous SNMP pollers.
#
# Mandatory: no
# Range: 0-1000
# Default:

StartSNMPPollers=1
```



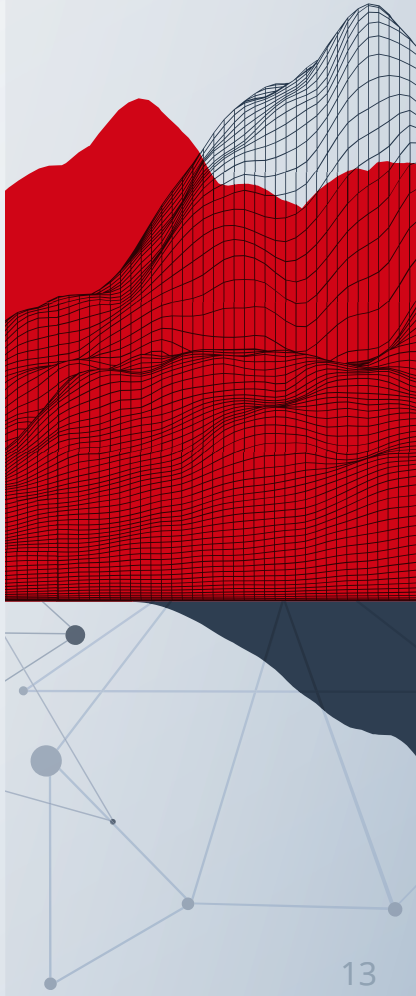
Zabbix agent poller

ZABBIX SUMMIT 2023

Zabbix agent poller collect data from passive Zabbix agents:

- Uses Zabbix communication protocol
- Optimized to communicate with Zabbix agents in passive mode

```
### Option: StartAgentPollers
#       Number of pre-forked instances of asynchronous Zabbix agent pollers
#
# Mandatory: no
# Range: 0-1000
# Default:

StartAgentPollers=1
```



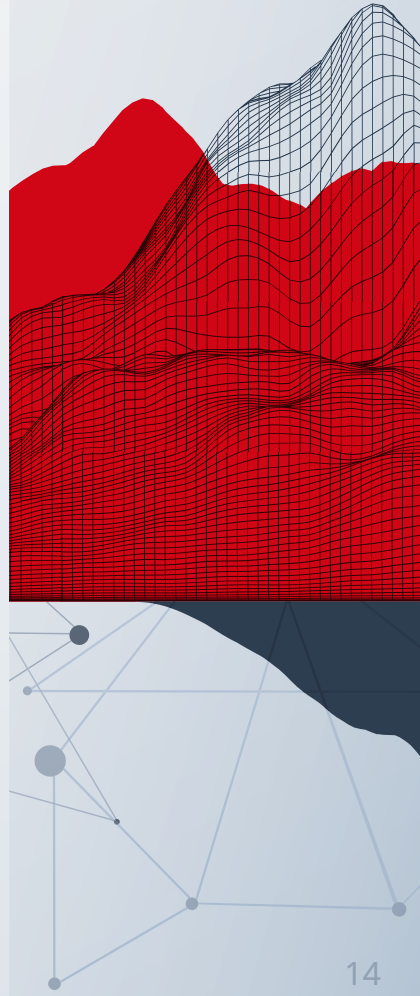
HTTP agent poller

ZABBIX SUMMIT 2023

HTTP agent poller collects data from the web pages:

- Uses libcurl for data collection
- Optimized to work with web requests

```
### Option: StartHTTPAgentPollers
#       Number of pre-forked instances of asynchronous HTTP agent pollers
#
# Mandatory: no
# Range: 0-1000
# Default:

StartHTTPAgentPollers=1
```



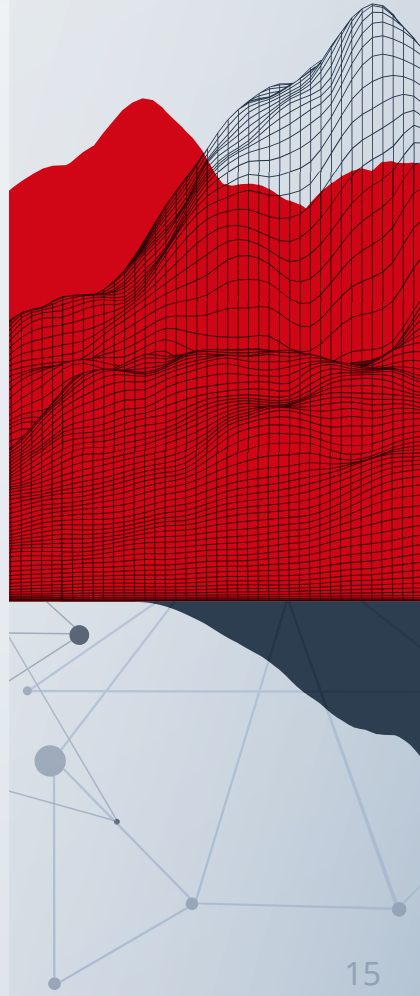
Number of concurrent checks

ZABBIX SUMMIT 2023

The maximum number of concurrent checks can be specified:

- Specified by the MaxConcurrentChecksPerPoller value
- The default value is 1000

```
### Option: MaxConcurrentChecksPerPoller
#       Maximum number of asynchronous checks that can be executed at once
#
# Mandatory: no
# Range: 1-1000
# Default:

MaxConcurrentChecksPerPoller=1000
```



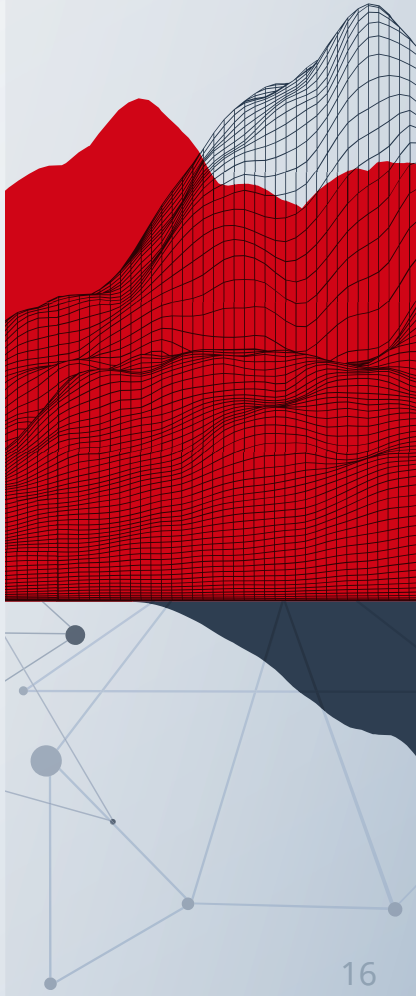
Asynchronous polling

ZABBIX SUMMIT 2023

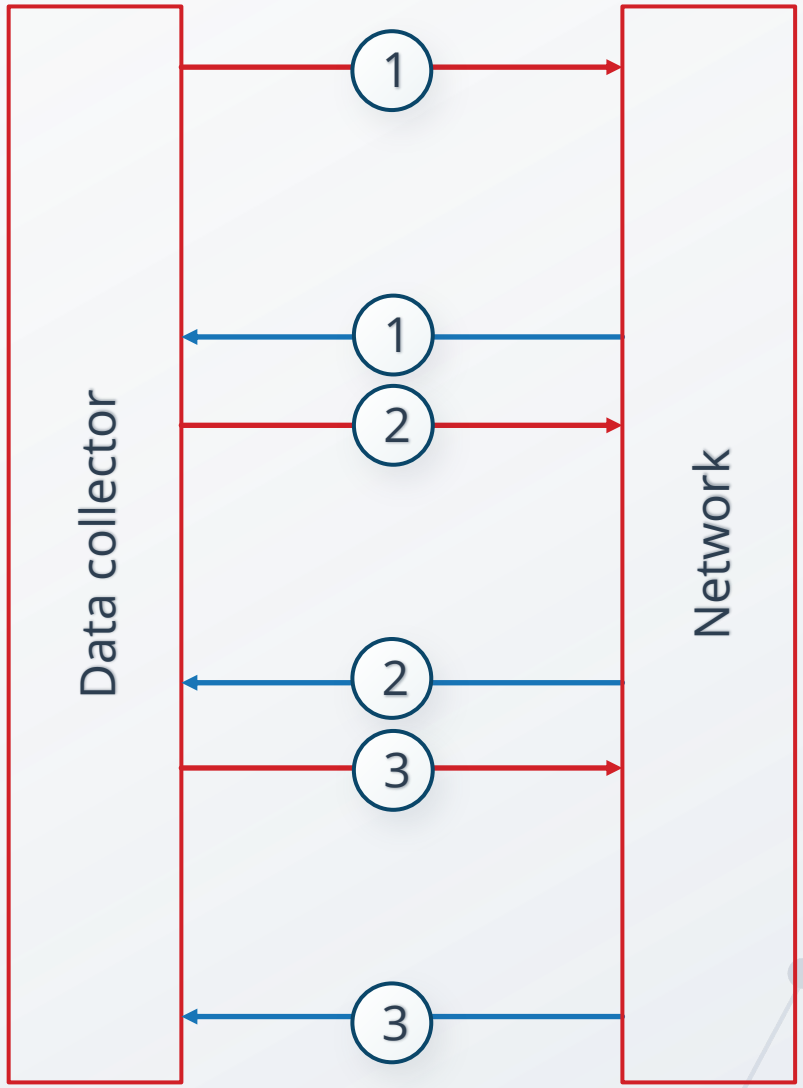
Each poller type now is an asynchronous process:

- Up to 1000 values can be collected by each process in a single cycle
- Separate thread is started to synchronize with configuration cache

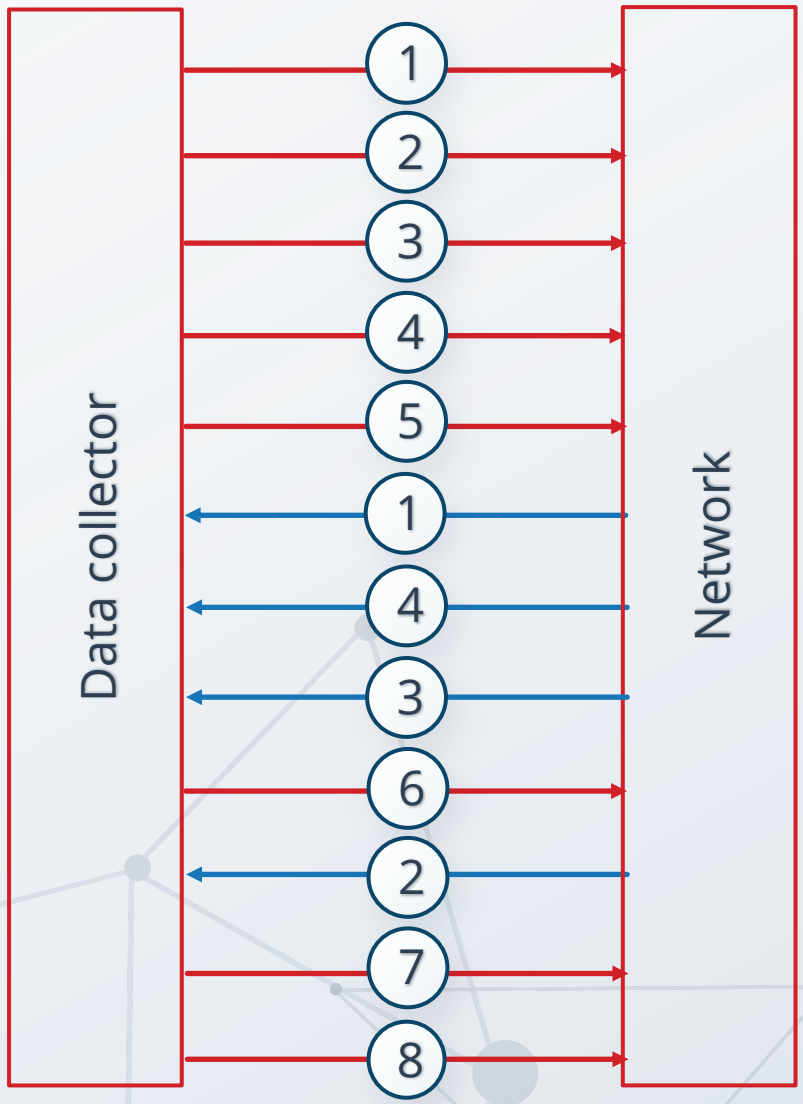
Asynchronous processes are a programming concept that allows tasks to be executed independently of each other and without blocking the main program's execution.



Synchronous data collection

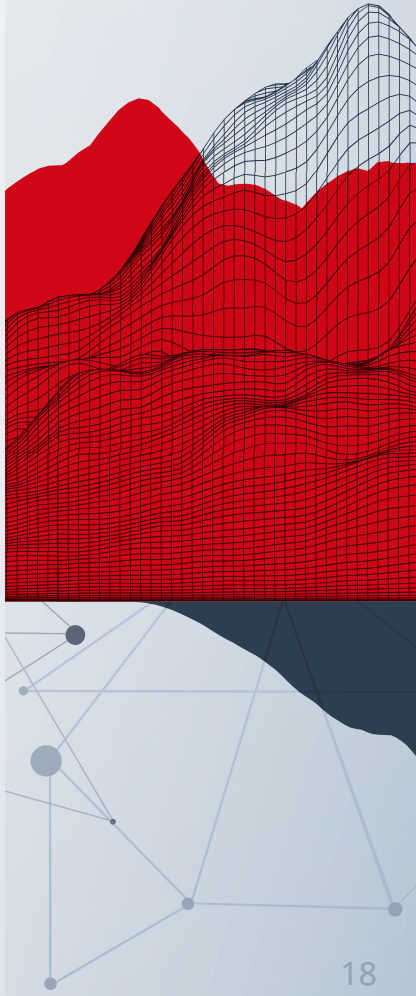


Asynchronous data collection



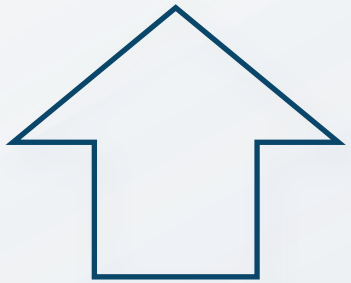
Zabbix 6.0 data collection

```
0:01 /usr/sbin/zabbix_server: poller #1 [got 97 values in 0.720154 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #2 [got 24 values in 0.480796 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #3 [got 24 values in 0.495460 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #4 [got 75 values in 0.734219 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #5 [got 24 values in 0.493454 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #6 [got 26 values in 0.493191 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #7 [got 25 values in 0.495608 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #8 [got 21 values in 0.480812 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #9 [got 25 values in 0.493319 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #10 [got 18 values in 0.489254 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #11 [got 20 values in 0.510633 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #12 [got 25 values in 0.495553 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #13 [got 21 values in 0.493304 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #14 [got 25 values in 0.493086 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #15 [got 90 values in 0.741500 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #16 [got 19 values in 0.492912 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #17 [got 17 values in 0.495651 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #18 [got 26 values in 0.492523 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #19 [got 25 values in 0.493129 sec, getting values]
0:01 /usr/sbin/zabbix_server: poller #20 [got 27 values in 0.493284 sec, getting values]
```



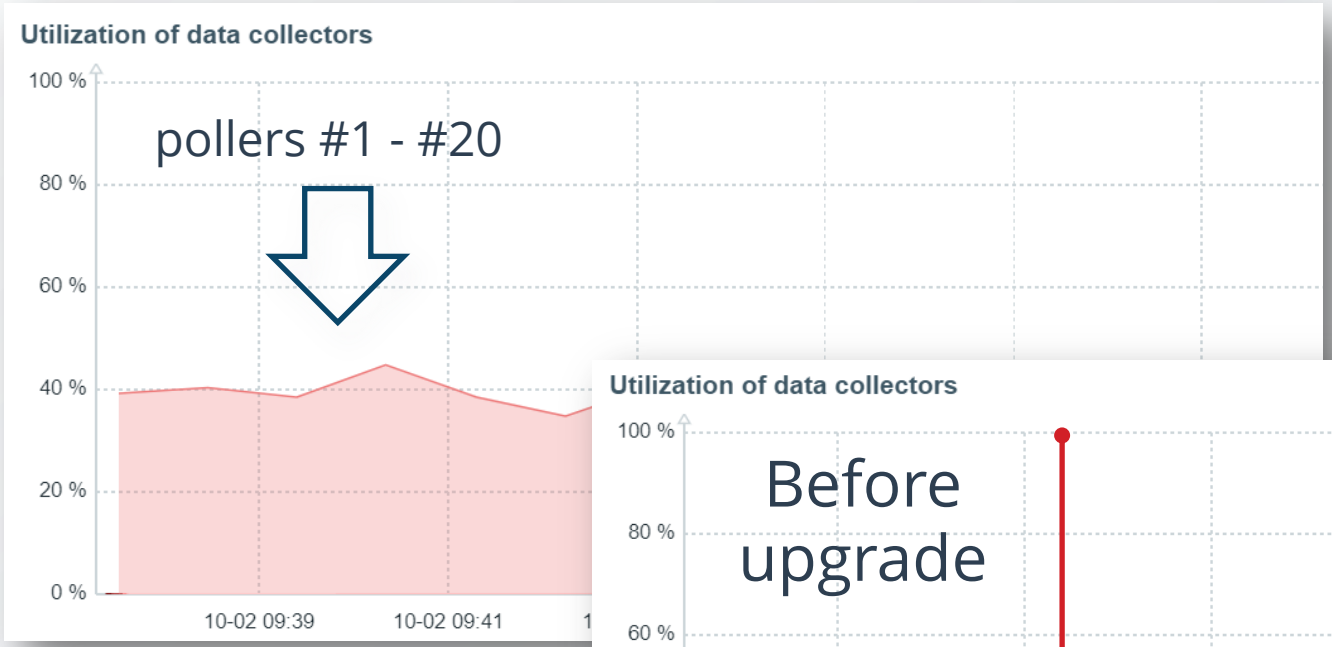
Zabbix 7.0 data collection

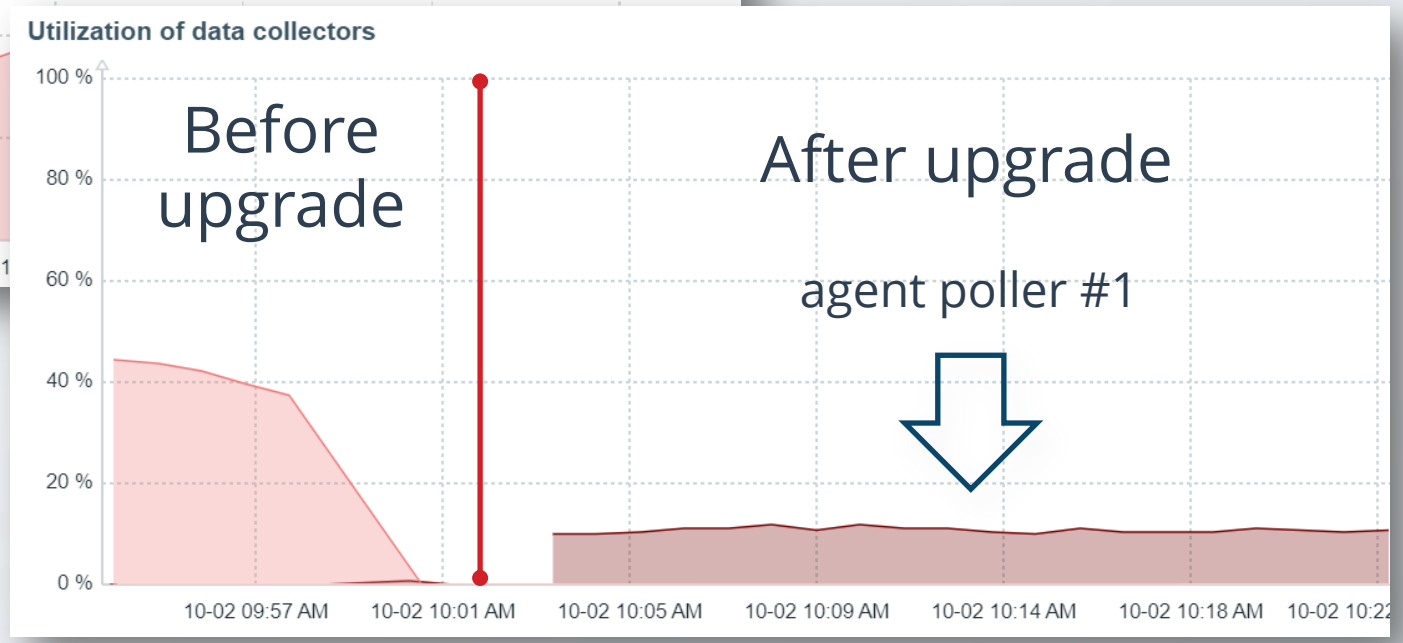
```
0:00 /usr/sbin/zabbix_server: poller #1 [got 0 values in 0.000015 sec, idle 1 sec]
0:00 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000018 sec, idle 1 sec]
0:00 /usr/sbin/zabbix_server: poller #3 [got 14 values in 0.000337 sec, idle 1 sec]
0:00 /usr/sbin/zabbix_server: http agent poller #1 [got 0 values, queued 0 in 5 sec]
0:06 /usr/sbin/zabbix_server: agent poller #1 [got 1497 values, queued 1108 in 5 sec]
0:00 /usr/sbin/zabbix_server: snmp poller #1 [got 0 values, queued 0 in 5 sec]
```



Reduce the number of processes

Zabbix 6.0 vs 7.0





ZABBIX SUMMIT 2023

Proxy memory buffer

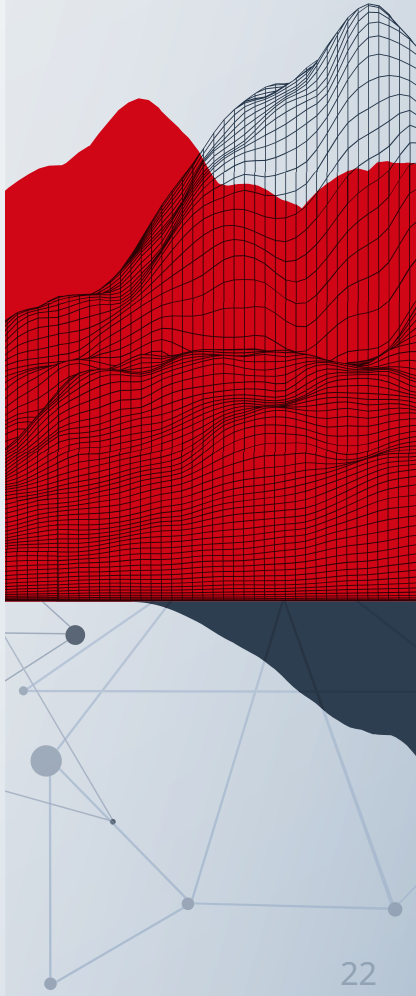
New proxy buffer modes

ZABBIX SUMMIT 2023

Zabbix 7.0 will introduce new proxy buffer modes:

- Disk — current behavior
- Memory — data stored only in shared memory
- Hybrid — buffer works in memory mode with DB as backup

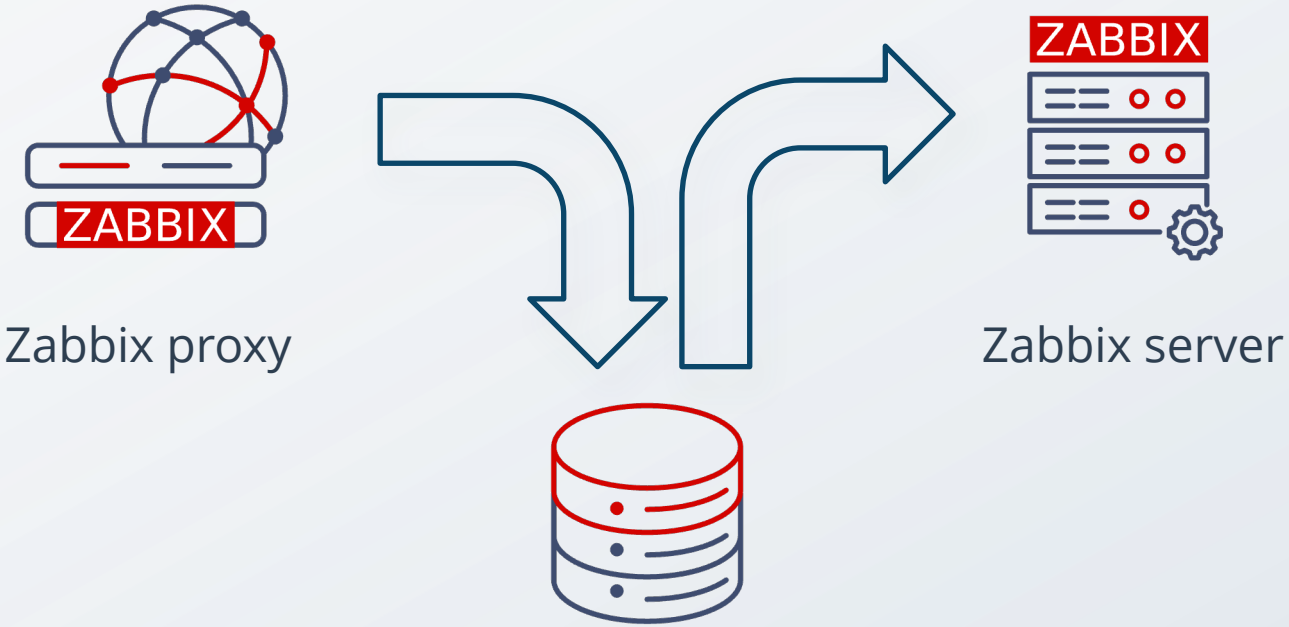
```
### Option: ProxyBufferMode
#     Specifies history, discovery and auto registration data storage mechanism:
#     disk    - data are stored in database and uploaded from database
#     memory  - data are stored in memory and uploaded from memory.
#     hybrid  - the proxy buffer normally works like in memory mode until it
#               runs out of memory or the oldest record exceeds the configured age
# Default:
# ProxyBufferMode=disk

ProxyBufferMode=hybrid
```



Disk mode

Each value collected by Zabbix proxy goes through a database:

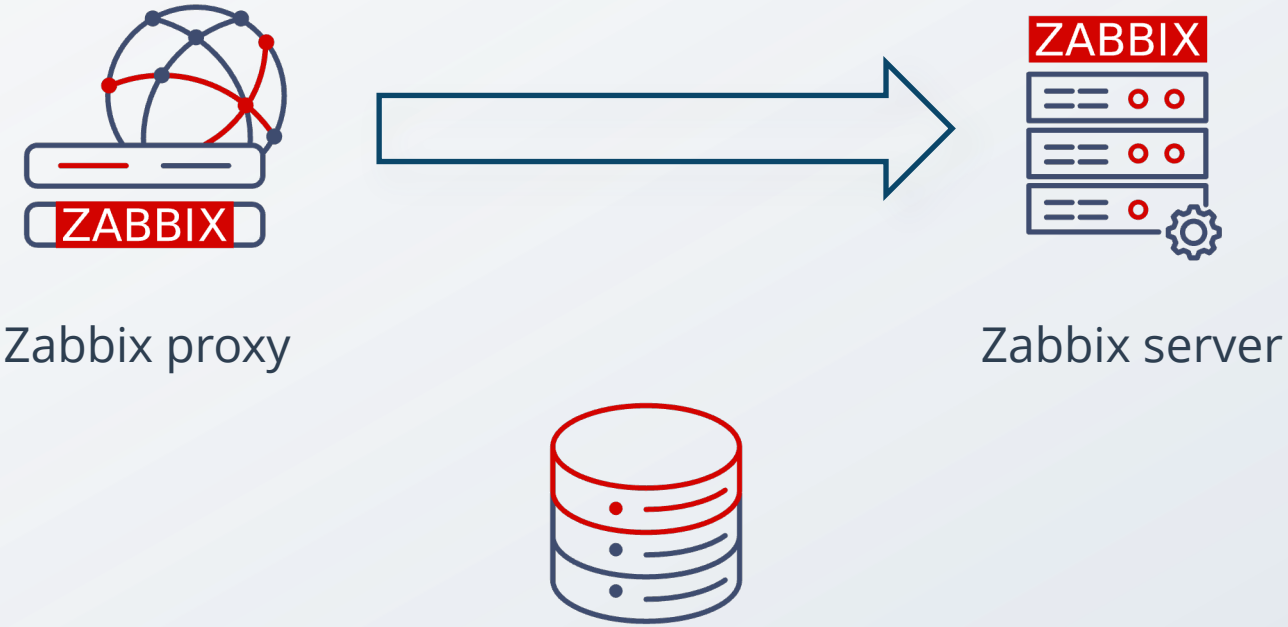
- Database (MySQL, Postgres or SQLITE) required on each proxy
- This may cause a bottleneck on large proxies



Memory mode

Data is sent to Zabbix server directly:

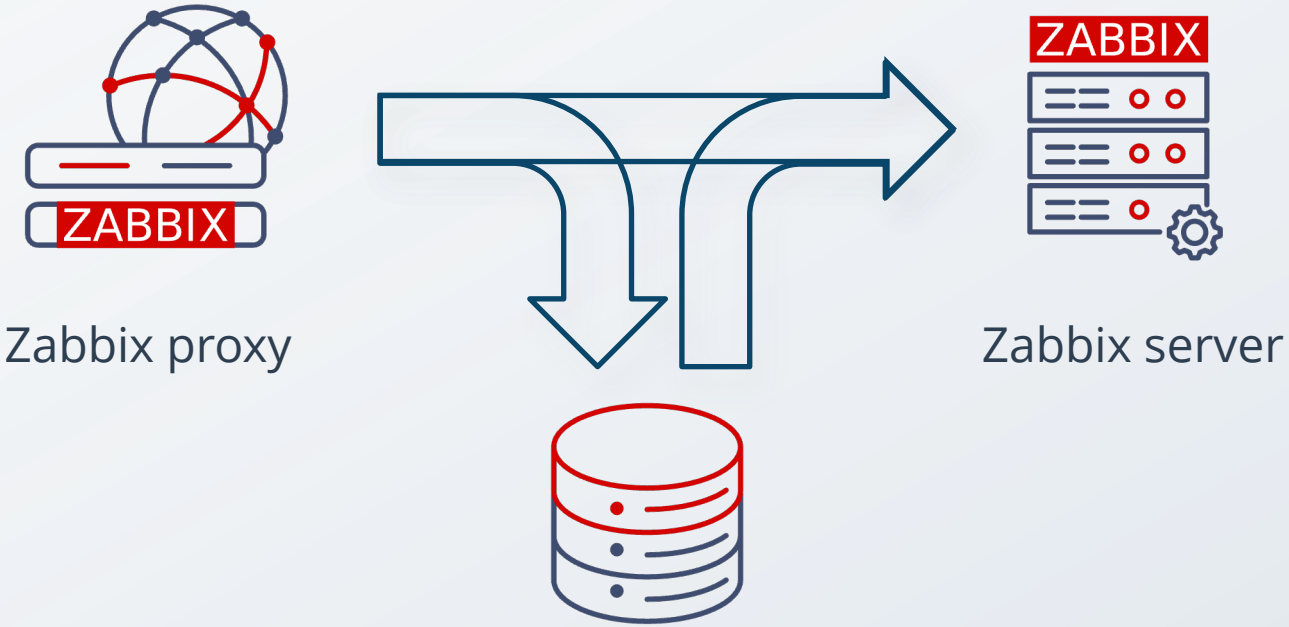
- the history data is being stored in shared memory and uploaded from it
- If buffer runs out of memory the old data will be discarded



Hybrid mode

Data is sent to Zabbix server directly:

- buffer normally works like in the memory mode
- the buffer is flushed in database if buffer does not have enough space



ZABBIX SUMMIT 2023

Item level timeout

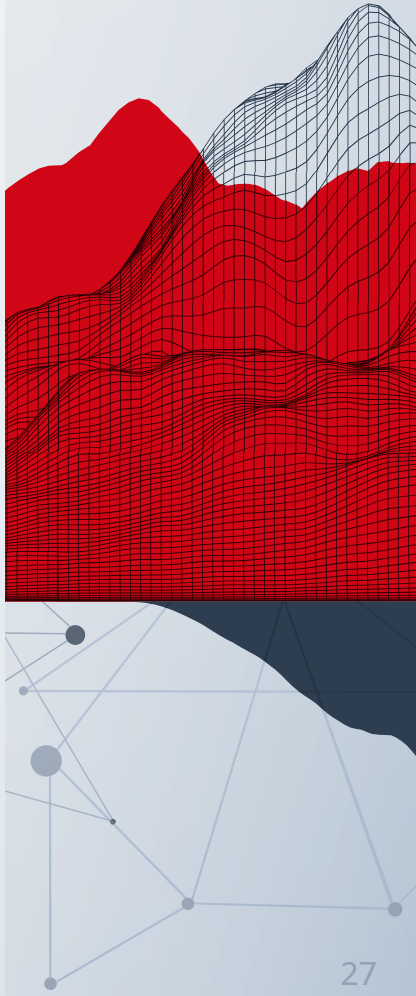
Zabbix timeout

ZABBIX SUMMIT 2023

Zabbix has a general timeout specified in the configuration file

- Timeout affects all data collection on Zabbix server or proxy
- Some item types have own timeout (HTTP agent, Script)

```
### Option: Timeout
#       Specifies how long we wait for agent, SNMP device or external check
#
# Mandatory: no
# Range: 1-30
# Default:

Timeout=4
```



Item level timeout

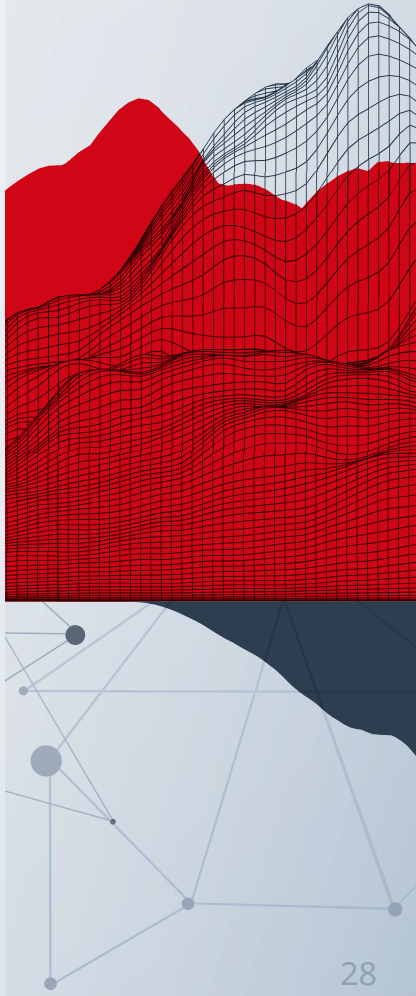
ZABBIX SUMMIT 2023

Zabbix 7.0 will introduce item level timeout for most checks:

- Timeout is defined using Zabbix graphical user interface
- Range is from 1 to 600 seconds (10 minutes)

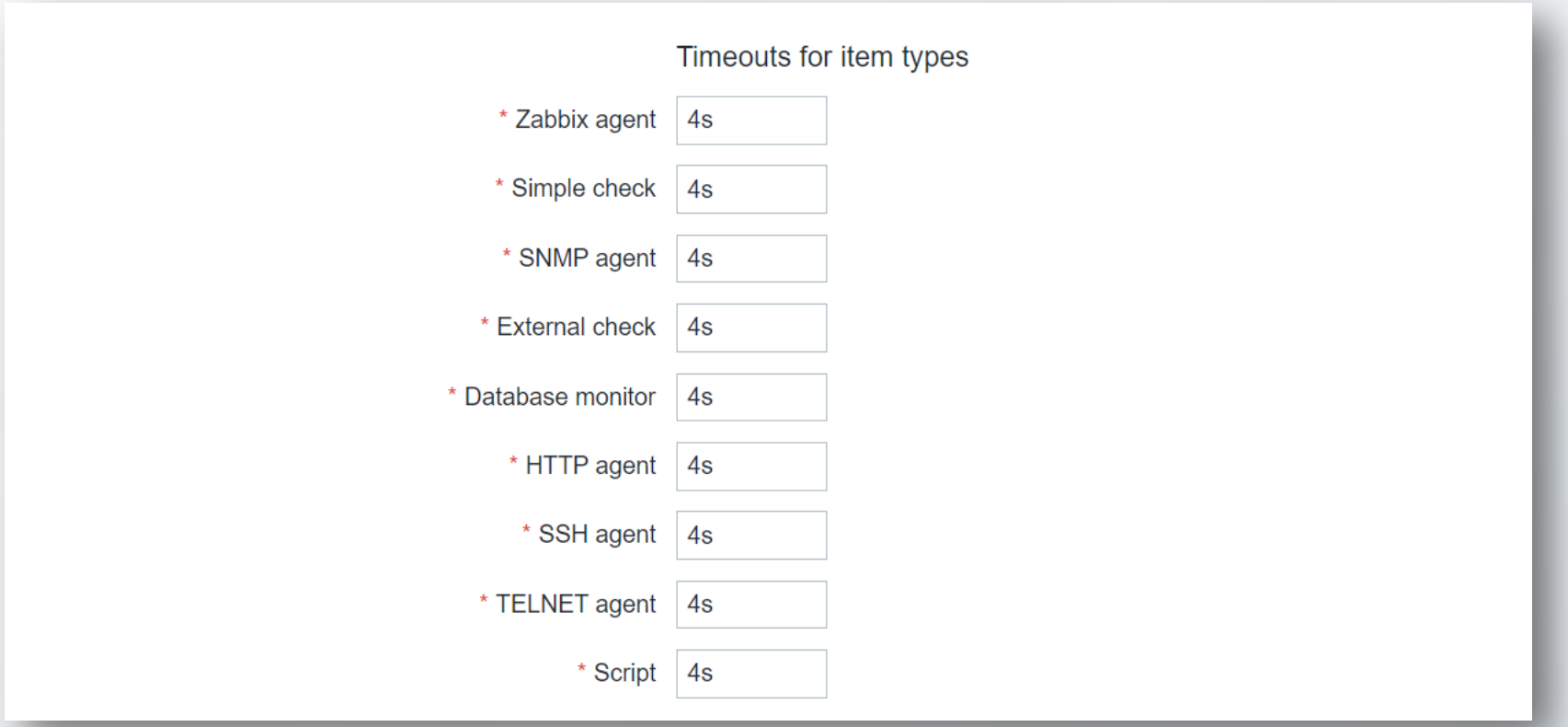
Timeout can be defined on multiple levels:

- On Zabbix server globally for all items
- Per proxy for items monitored by the proxy
- On each item individually



Global timeouts

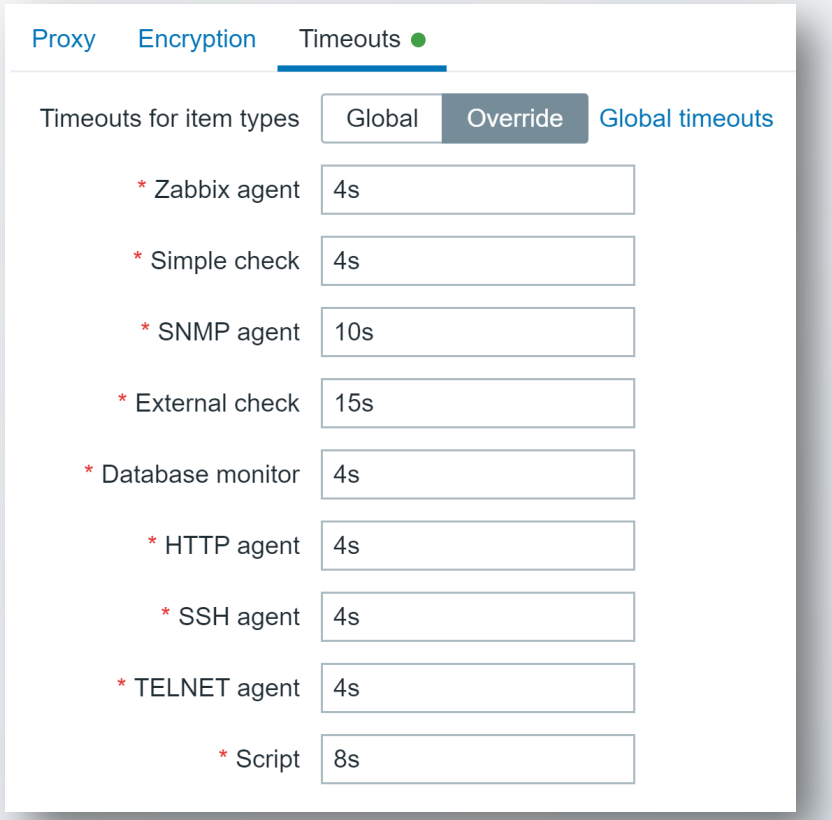
Global timeouts can be defined in Administration > General section



Proxy level timeouts

Proxy level timeout will affect all items collected by proxy

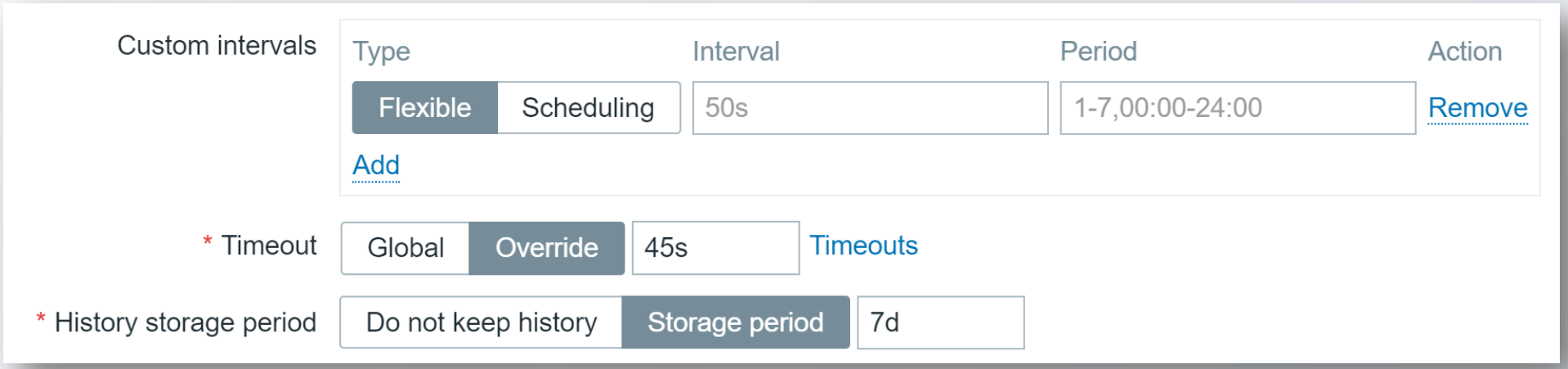
- Each type can be tuned individually
- Timeouts work as default values
- Forced override can be enabled



Individual timeouts

By using item level timeout, it is possible to tune individual items:

- Timeout defined globally or on proxy is used by default
- Can be adjusted if needed for a specific item



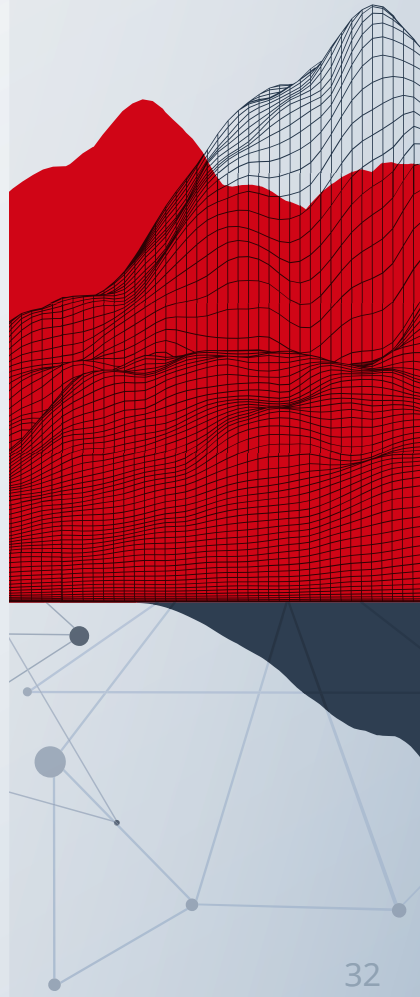
Timeout in the configuration file

ZABBIX SUMMIT 2023

The timeout setting from Zabbix server / proxy configuration file

- Will become a default value for item timeout during the upgrade process
- Will be used as a timeout for communication between server and proxy

```
### Option: Timeout
#       Specifies timeout for communications (in seconds).
#
# Mandatory: no
# Range: 1-30
# Default:

Timeout=4
```



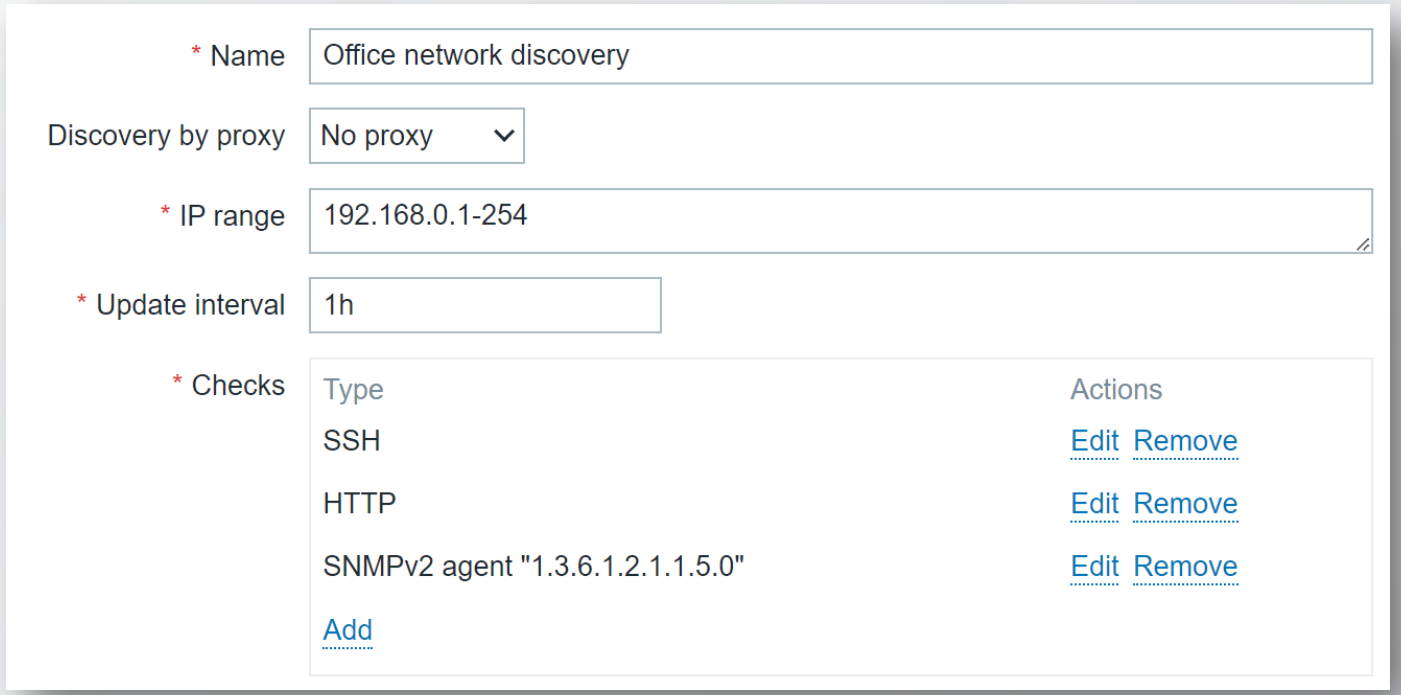
ZABBIX SUMMIT 2023

Concurrent network discovery

Network discovery

Network discovery can be quite slow on large segments:

- Each segment is processed by a single discoverer only
- Each IP address and service is processed sequentially



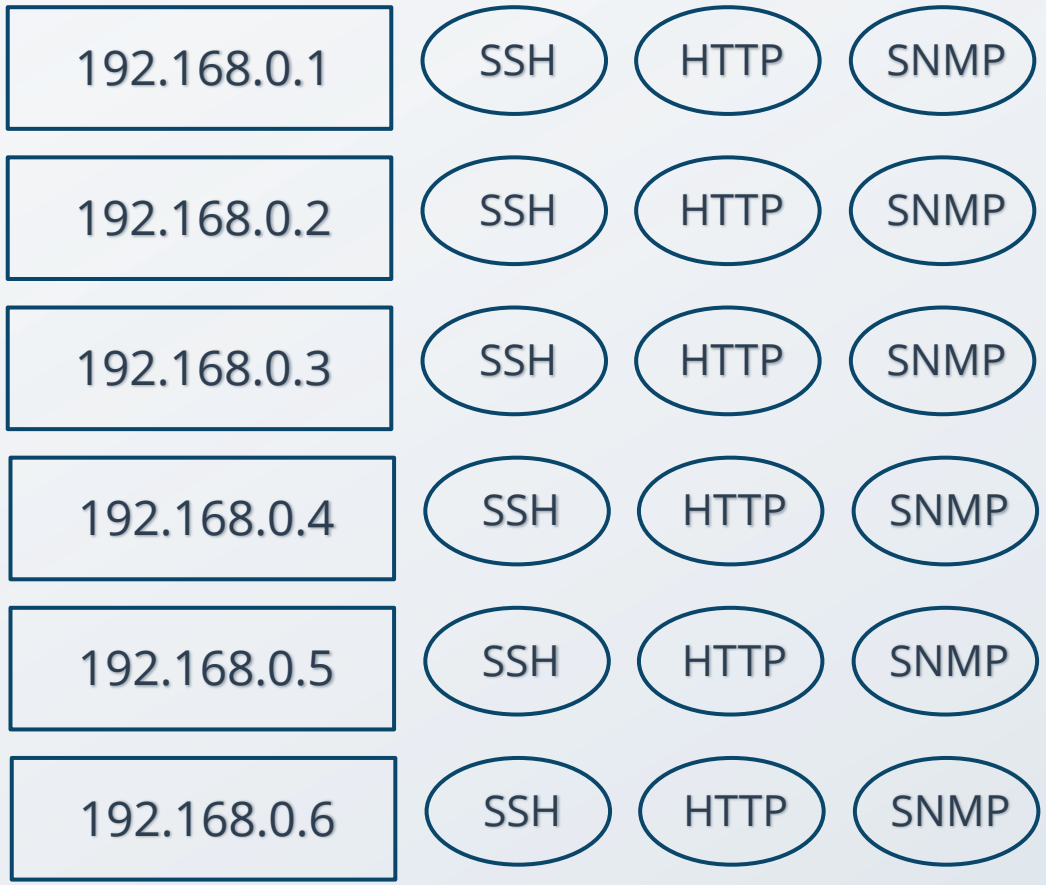
Sequential network discovery

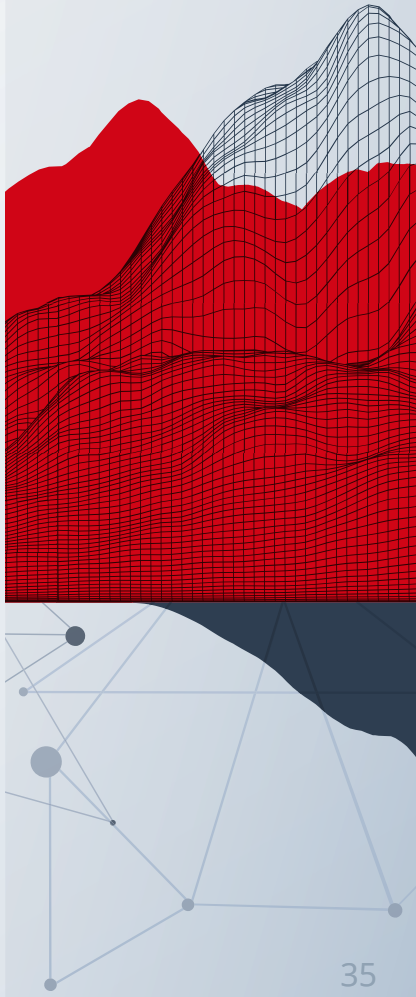
ZABBIX SUMMIT 2023

* Name: Office network discovery
Discovery by proxy: No proxy
* IP range: 192.168.0.1-254
* Update interval: 1h
* Checks:
Type
SSH
HTTP
SNMPv2 agent "1.3.6.1.2.1.1.5.0"
Add

```
### Option: Timeout
Timeout=3
```

Single process





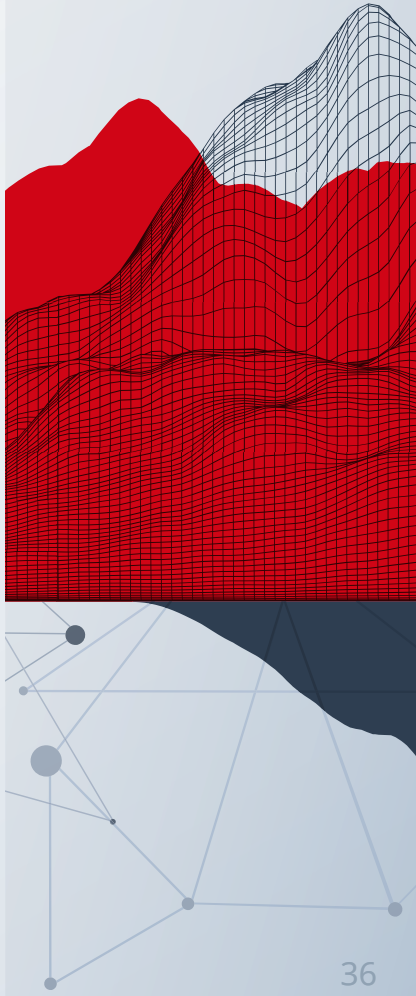
New processes

ZABBIX SUMMIT 2023

Zabbix 7.0 introduces new processes

- Discovery manager
- Discovery worker (previously known as discoverer process)

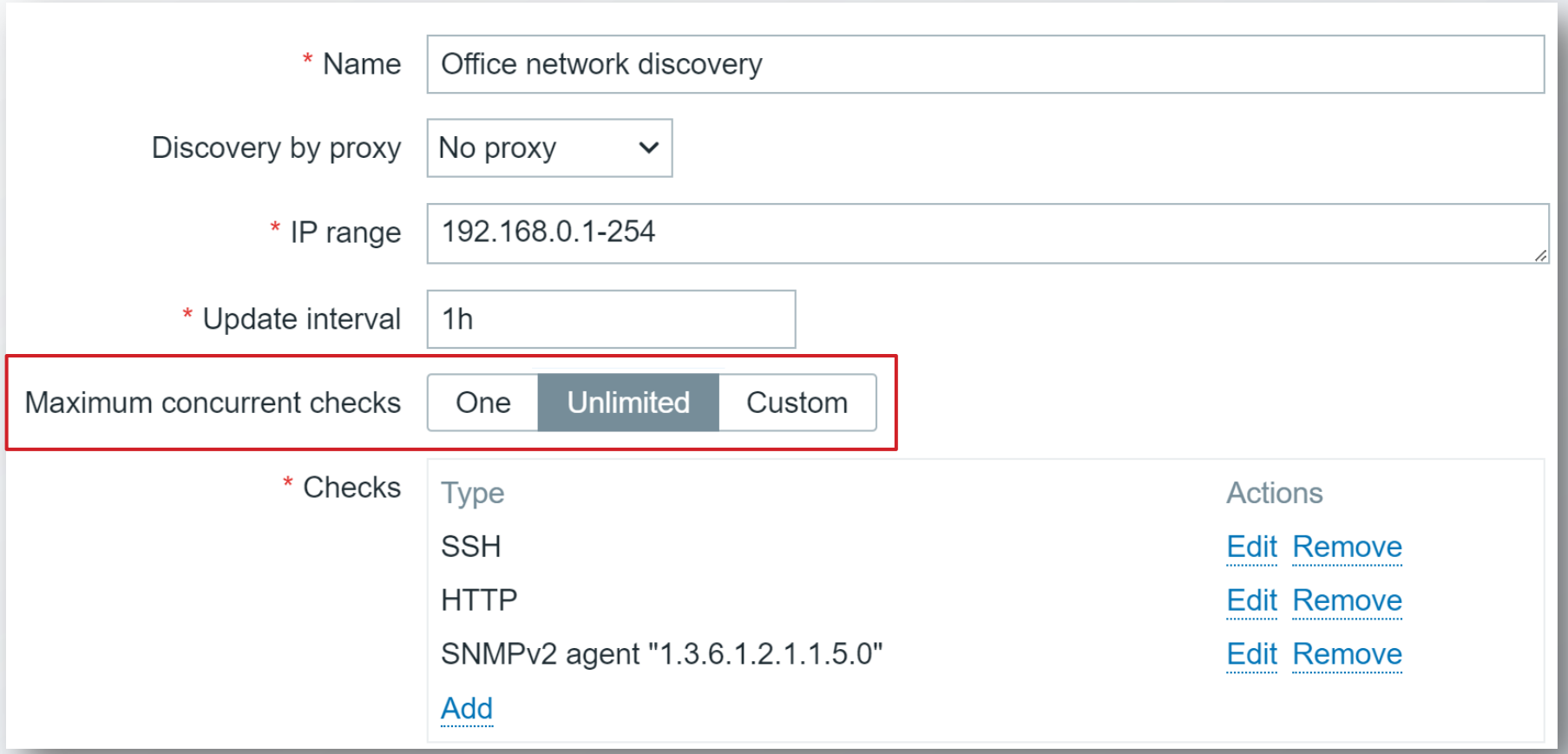
```
### Option: StartDiscoverers
#       Number of pre-started instances of discovery workers.
#
# Mandatory: no
# Range: 0-1000
# Default:

StartDiscoverers=5
```



Discovery rule configuration

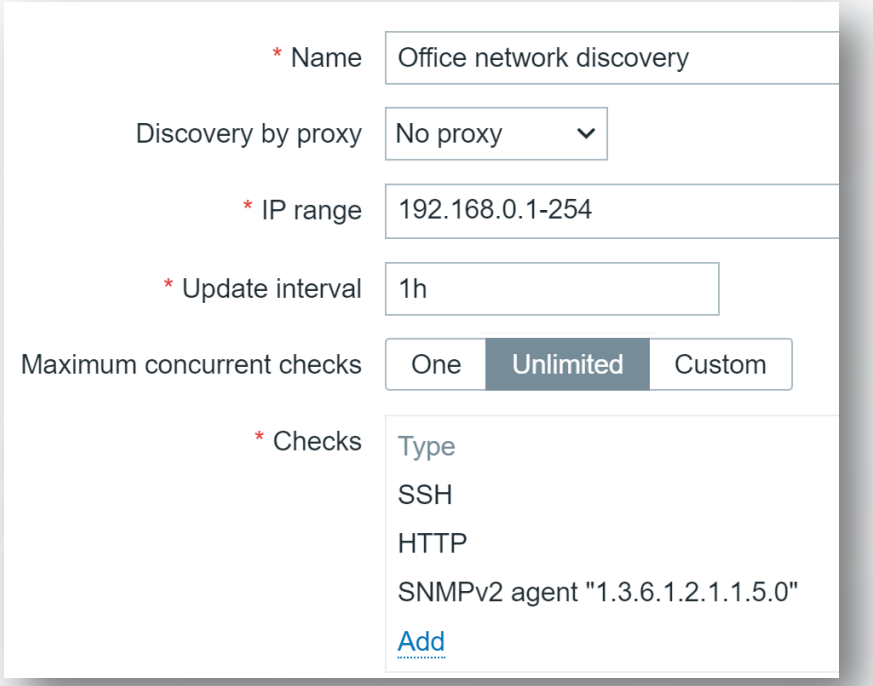
ZABBIX SUMMIT 2023

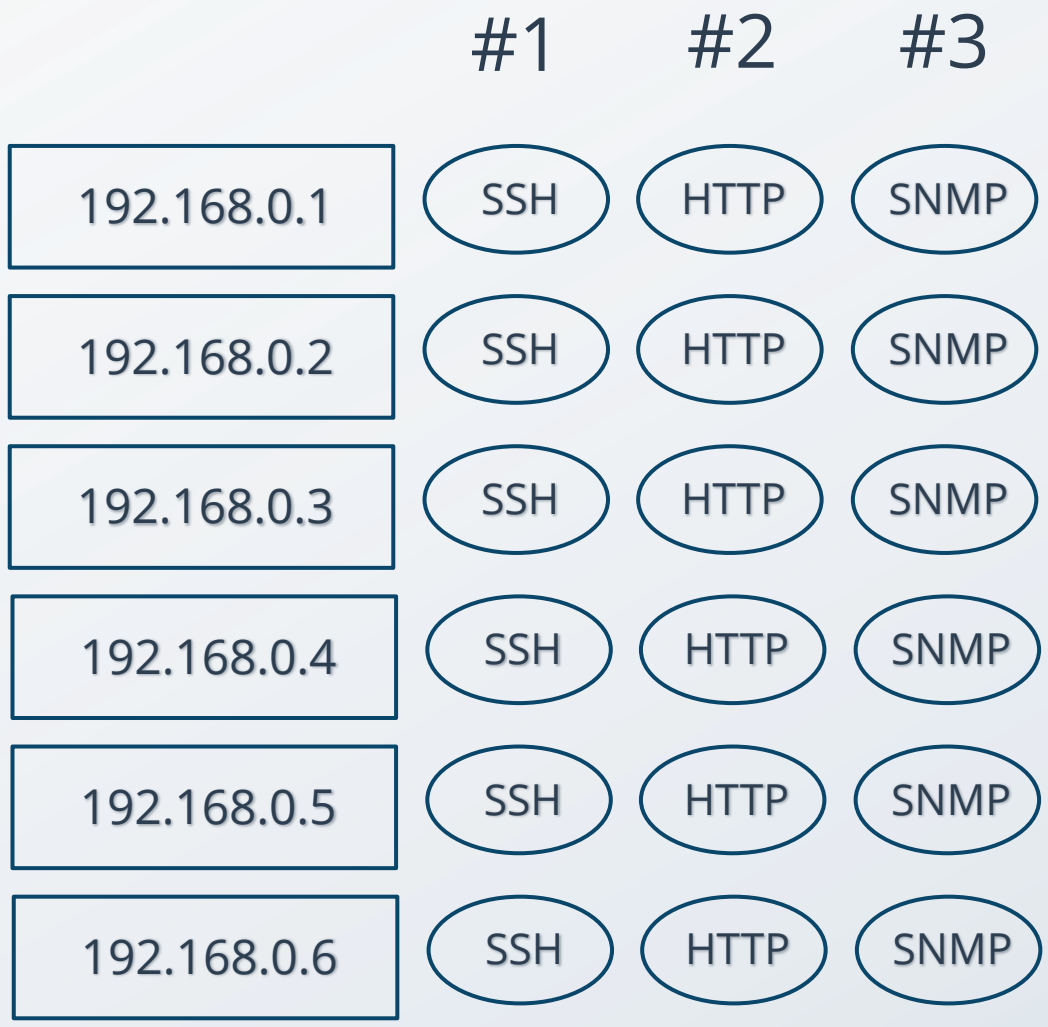
Concurrency is configured on discovery rule level

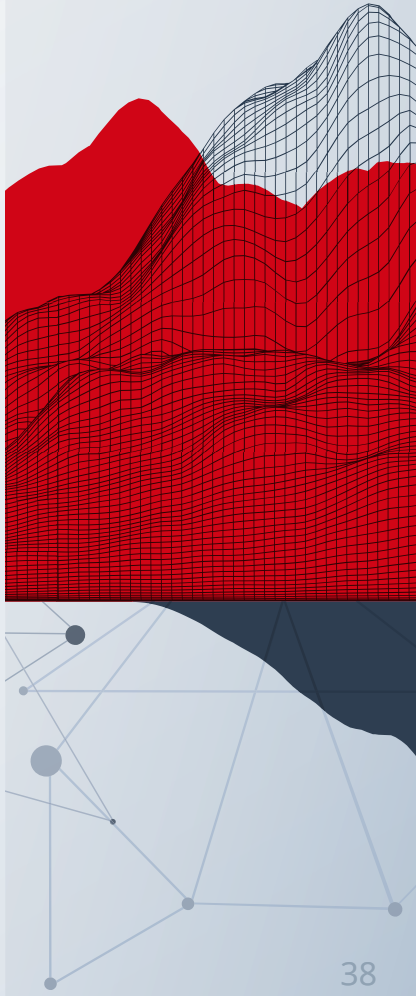


Concurrent network discovery (in DEV)

ZABBIX SUMMIT 2023







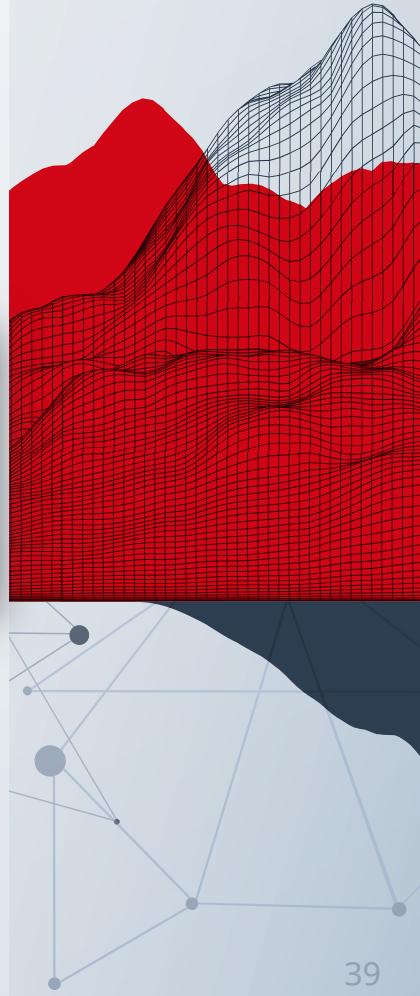
Internal monitoring

ZABBIX SUMMIT 2023

New internal items are introduced:

- The count of network checks enqueued monitored in the discovery queue
- Utilization of discovery manager and workers

Name ▲	Triggers	Key	Interval	History	Trends	Type
Zabbix server: Discovery queue		zabbix[discovery_queue]	1m	1w	365d	Zabbix internal
Zabbix server: Utilization of discovery manager internal processes, in %	Triggers 1	zabbix[process,discovery manager,avg,busy]	1m	1w	365d	Zabbix internal
Zabbix server: Utilization of discovery worker internal processes, in %	Triggers 1	zabbix[process,discovery worker,avg,busy]	1m	1w	365d	Zabbix internal



ZABBIX SUMMIT 2023

Thank You!

Kaspars Mednis

Chief trainer