ZABBIX QUEUE
SOLVE DELAYED METRICS

AIGARS KADIKIS
TECHNICAL SUPPORT ENGINEER
ZABBIX
SERVER QUEUE

- All devices are running all the time
- Each device are reachable by using desired protocol (passive checks)
- Devices are capable to reach the Zabbix component, deliver metric (active checks)
PASSIVE CHECKS

- Time correct between master and all Zabbix proxy servers

$ timedatectl

```
Local time: Thu 2021-03-18 19:38:14 EET
Universal time: Thu 2021-03-18 17:38:14 UTC
RTC time: Thu 2021-03-18 17:38:14
Time zone: Europe/Riga (EET, +0200)
System clock synchronized: no
NTP service: inactive
RTC in local TZ: no
```

ACTIVE CHECKS

- Time is correct on agent locally

```
date
date /T
time /T
```
HOW TO KEEP TIME CORRECT?

If host is using passive checks, then easy!
Compare a local time of machine with the execution time of this check.

```{Linux by Zabbix agent:system.localtime.fuzzytime(3m)}=0```

If hosts is using only active checks, then starting with version 4.2+ we can determine the offset by the help of JavaScript preprocessing step because JavaScript engine the knows server time:

```return Math.round((new Date()).getTime() / 1000 - value);```

More info if agent is using active checks only:
https://blog.zabbix.com/time-is-off-active-zabbix-agent/11676/
HOW QUEUE WORKS?

Uptime!

By restarting the component, it will clear up queue.

Restart does not solve queue. It creates an illusion that everything is OK.
Queue details are represented in level of item:

<table>
<thead>
<tr>
<th>Schedule check</th>
<th>Delayed by</th>
<th>Host</th>
<th>Name</th>
<th>Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-09-10 19:21:22</td>
<td></td>
<td></td>
<td>Kubernetes cluster</td>
<td></td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used cache memory grafana</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Container tomatotomat_1 is running</td>
<td></td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used RSS memory MSSQL</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used cache memory oracle-xe-11g</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used swap grafana</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used cache memory tomatotomat_1</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used RSS memory grafana</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used cache memory MSSQL</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>Used swap MSSQL</td>
</tr>
<tr>
<td>2021-09-10 19:21:38</td>
<td></td>
<td></td>
<td>Docker 02</td>
<td>CPU utilization tomatotomat_1</td>
</tr>
</tbody>
</table>

Section limited to 500 records 😞

Q: What if I have tens of thousands missing metrics?
A: Most likely some devices are completely offline
FORGET QUEUE, SOLVE ALL RED HOSTS

For hosts which are using at least one passive checks, we can use API to query error massage.
FORGET QUEUE, SOLVE ALL RED HOSTS

Solution via Windows PowerShell:

More about solution:

https://blog.zabbix.com/summarize-devices-that-are-not-reachable/13219/
Obtain session token by reaching out frontend:

```
curl http://127.0.0.1/api_jsonrpc.php -s -X POST -H 'Content-Type: application/json' -d \
'{"jsonrpc":"2.0","method":"user.login","params":{"user":"Admin","password":"zabbix"},"id":1,"auth":null}' | \
grep -E -o "([0-9a-f]{32,32})"
```

```
0cf5f7e39de997d9a02593ab973acf85
```

Query itemid's by using queue.get method:

```
zabbix_get -s 127.0.0.1 -p 10051 \
-k '{"request":"queue.get","sid":"0cf5f7e39de997d9a02593ab973acf85","type":"details","limit":"9999999"}
```

Output:

```
{"response":"success","data":[{"itemid":322185,"nextcheck":1614589895}, 
{"itemid":322191,"nextcheck":1614589895}, 
{"itemid":322188,"nextcheck":1614589895}, 
{"itemid":322187,"nextcheck":1614590435} ],"total":4}
```
PUT JSON IN EXTERNAL FILE

Put item IDs in external file:

```
zabbix_get -s 127.0.0.1 -p 10051
-k '{"request":"queue.get","sid":"0cf5f7e39de997d9a02593ab973acf85","type":"details","limit":"9999999"}' >
/tmp/queue.json
```
PASSWORDLESS ACCESS TO DATABASE

MySQL:

```bash
# cat ~/.my.cnf
[client]
host=127.0.0.1
user=zabbix
password=zabbix
```

Postgres:

```bash
# cat ~/.pgpass
127.0.0.1:5432:*:zabbix:zabbix
```
SUMMARIZE

MySQL:

grep -oP 'itemid":\K\d+" /tmp/queue.json | tr "\n" ',' | sed 's|$.||' | xargs -i echo "
SELECT p.host AS proxy, hosts.host, items.key_
FROM hosts
JOIN items ON (hosts.hostid = items.hostid)
JOIN hosts proxy ON (hosts.proxy_hostid=proxy.hostid)
LEFT JOIN hosts p ON (hosts.proxy_hostid=p.hostid)
WHERE items.itemid IN ({});
"
| mysql --table zabbixDB > /tmp/queue.txt

Postgres:

grep -oP 'itemid":\K\d+" /tmp/queue.json | tr "\n" ',' | sed 's|$.||' | xargs -i echo "
SELECT p.host AS proxy, hosts.host, items.key_
FROM hosts
JOIN items ON (hosts.hostid = items.hostid)
JOIN hosts proxy ON (hosts.proxy_hostid=proxy.hostid)
LEFT JOIN hosts p ON (hosts.proxy_hostid=p.hostid)
WHERE items.itemid IN ({});
"
| psql zabbixDB > /tmp/queue.txt
<table>
<thead>
<tr>
<th>proxy</th>
<th>host</th>
<th>key_</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GUNTA-NOTE</td>
<td>RebootRequired</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>proc_info[msiexec.exe]</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>vfs.dir.count[C:,^Windows.old$,,,0]</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>vfs.file.exists[C:\Windows\Memory.dmp]</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>vfs.file.exists[C:\hiberfil.sys]</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>agent.ping</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>wmi.getall[root\cimv2,select * from Win32_OperatingSystem]</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>system.localtime</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>agent.version</td>
</tr>
<tr>
<td>broceni</td>
<td>GUNTA-NOTE</td>
<td>system.localtime[utc]</td>
</tr>
</tbody>
</table>
THANK YOU!

AIGARS KADIKIS
TECHNICAL SUPPORT ENGINEER
ZABBIX