Event Console Integration

03.05.2018, Sven Panne
Check_MK Conference #4
Two Kinds of Monitoring

Status-based

- Pushing/pulling states
- “Classic” Check_MK

Event-based

- Pushing of messages
- Independent of state
- Event Console

Both kinds are useful!
Design Considerations

- High frequency of incoming messages
- Human-manageable amount of actual events
- Not a high-performance syslog archive
Architectural Overview I

Basic Pipeline

- SNMP
- syslog
- spool
- local

Input

- Trap Parser
- Rule Engine
- syslog Parser

EC

Output

- Shell Script
- EMail
- Notifications
Architectural Overview II
Persistence & GUI

Monitoring Core

Livestatus

History
Events

SNMP
syslog
spool
local

Trap Parser
Rule Engine
Actions

GUI
Views
- Events
- History
- Details

Shell Script
EMAIL
Notifications
Architectural Overview IV
Host Information

Monitoring Core → Livestatus

History → Events → Hosts, Downtimes

SNMP → Trap Parser
syslog → syslog Parser
spool → local

Rule Engine → Actions

Check → Event Service

GUI
Views
- Events
- History
- Details

Event Service
- Shell Script
- EMail
- Notifications
- Logwatch
Events are useful, even during host downtimes

Still generated, but marked specially

Automatically archived after downtime

Tactical overview considers them “handled”

Actions can be skipped
Further Improvements

- Finer configuration of limits
- More powerful rewriting
- Export of rule packs via MKPs
Background jobs

03.05.2018, Tom Bärwinkel
Check_MK Conference #4
Examples of long running processes

Create Report  Bake Agents  Host renaming
Problem – Apache process timeouts

**Internal error:** Your request timed out after 110 seconds. This issue may be related to a local configuration problem or an object with a too large number of objects. But if you think this issue is a bug, please send a crash report.

An internal error occurred while processing your request. You can report this issue to the Check_MK team to help fixing this issue below for reporting.

**Crash Report**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>tb</td>
</tr>
<tr>
<td>Email Address</td>
<td></td>
</tr>
</tbody>
</table>

**Crash Report**

- **Crash Type:** gui
- **Time:** 2018-04-20 16:15:18
- **Operating System:** artful
The usual workaround

Let’s just increase the timeout

locate files  increase timeout  restart

update
Still a problem - What’s the status?

- No user feedback
- Blocked processes
- Apache restarts
Solution - separation of concerns

GUI
- configuration
- start jobs
- show progress

Common Interface

Background jobs
- do actual work
- send updates
How it looks in practice - Reports

Create new report schedule entry

- **Unique ID**
- **Report to create**: BI Availability
- **Schedule title**
- **Activation**: do not create this report
- **Generate report in background job**: run as background job
- **Period**: Every day
- **Time of day to create report at**: 00:00
- **Time range to create report for**: Last month
- **Owner**: cmkadmin - cmkadmin
- **Action**: Send via Email

Options for email:
How it looks in practice - Reports

Report Scheduler

Report generation started in background job reporting:1524736450.2

Scheduled reports

<table>
<thead>
<tr>
<th>Actions</th>
<th>ID</th>
<th>Title</th>
<th>Owner</th>
<th>Report</th>
<th>Action</th>
<th>Time Range</th>
<th>Context</th>
<th>Period</th>
<th>Time</th>
<th>Last Run</th>
<th>Last I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>background_job</td>
<td>Background Job</td>
<td>cmkadmin</td>
<td>Report of Host</td>
<td>Send via Email, Email-Addresses to mail PDF reports to: Options for report emails: (no parameters)</td>
<td>Last month</td>
<td>1 Filter settings</td>
<td>Every day</td>
<td>00:00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reporting

<table>
<thead>
<tr>
<th>Actions</th>
<th>Job ID</th>
<th>Job Title</th>
<th>State</th>
<th>Started</th>
<th>Owner</th>
<th>PID</th>
<th>Runtime</th>
<th>Last progress info</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>reporting-1524736450.2</td>
<td>Report generation of background_job / Background Job</td>
<td>initialized</td>
<td>2018-04-26 11:54.10</td>
<td>cmkadmin</td>
<td>19474</td>
<td>0:00:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>reporting-1524736392.6</td>
<td>Report generation of background_job / Background Job</td>
<td>finished</td>
<td>2018-04-26 11:53.12</td>
<td>cmkadmin</td>
<td>18329</td>
<td>0:00:00</td>
<td>Report generation complete.</td>
<td>Report generation finished</td>
</tr>
</tbody>
</table>
The Background jobs overview
Special rules for critical jobs

- Affected jobs:
  * bake agents
  * rename host
- Only one job allowed
- Error page with details
- Require acknowledgement
Upcoming background jobs

- Download of agent output
- Service discovery
- SLA calculations
Time-specific check parameters

03.05.2018, Konstantin Büttner
Check_MK Conference #4
Use case

• Recurring events lead to conditions that would be abnormal during other times

• Eg. backup jobs etc.
Predictive Levels?

![Graph showing predictive levels with different areas: Reference, OK area, Warning area, Critical area, and current value: 0.40. The graph shows data points over a 16-hour period.]
Solution – Timespecific parameters

• Define parameters for time periods

• Generic solution – works for all check plugins
How does it work?

Parameters

- Fixed Levels
- Enable timespecific parameters

- Warning at 5.00 per core
- Critical at 10.00 per core
How does it work?

- Configured timeperiod parameters
- Disable timespecific parameters

Default parameters when no timeperiod matches
- Fixed Levels
- Warning at 5.00 per core
- Critical at 10.00 per core
Customizable graph layouts

03.05.2018, Konstantin Büttner
Check_MK Conference #4
Our objective in graphing

• Traditional graphing systems are for customizing graphs

• But: day-to-day, we’d rather just have graphs
  Tweaking graphs is fun, but time consuming

• Our objective: Make our data useful out of the box
# A bit of history - PNP4Nagios

<table>
<thead>
<tr>
<th>PNP graph</th>
<th>In the Olden Days</th>
</tr>
</thead>
</table>
| ![Graph](image) | - Graphs were defined for each check type  
- Templates in PHP (ノ samo益
   ぞ)ノ 町 町
- No semantics, no uniformity*  
- No customizability** |

*except for checks sharing templates  
**unless you wrote your own templates
A bit of history - Metrics

Metrics graph

A new dawn

- Most visible: New look
- Most important: Semantics
- Checks no longer have graphs...
- ...but values have semantics
  - Automatic graphs
  - Custom graphs
Now: More customizability

• Unified options for dashlets, views, reports (Werk #5569)
Now: More customizability

• Unified options for dashlets, views, reports (Werk #5569)

• Vertical scaling and mirroring customizable (Werk #3920)
Graph ......................................................... Heap and non-heap memory

Apply to
- Specific destination
- Template graphs
  - Site
  - Hostname
  - Service description
- Combined graph
  - Apply to combined graphs

Vertical axis scaling
- Explicit range
  - Lower: 0.00  Upper: 209715200.0

Mirror the vertical axis

Add new element
Performance improvements

03.05.2018, Konstantin Büttner
Check_MK Conference #4
Why is a performance focus critical?

• The typical Check_MK setup keeps growing – more services, more sites, more users

• Better scalability makes new things possible
How to make things go faster

Avoid unnecessary work

Optimize

Make things feel faster
Avoid unnecessary work - Tabs

• A typical browser: 50 open tabs

• You don’t use them all... but they all make regular livestatus queries, though

• New: Only update active tabs/windows (Werk #4753)

• Benefit: No more wasted data transfers
Avoid unnecessary work – Emails

• Graphs can make emails reasonably large

• Multiply that by the number of recipients ...

• What’s a mail server for? (Werk #4813)

• Benefit: Notification system becomes more scalable
Avoid unnecessary work – Sites

• Frequent scenario: multi-site setup

• For many users, only few sites are actually relevant

• Yet, GUI in the past contacted all sites for every user

• No more: Now configureable for each user (Werk #4921)
Optimize – BI

• Large aggregations now compile faster (Werk #5142)

• Magic - “it’s the algorithm, stupid”

• Further reworking already planned
Optimize – liveproxyd

• Previously: # The main loop of the daemon goes here

• One python process – GIL limits scalability

• Now what?
Optimize – liveproxyd

• Now: One process per site, master process to manage
• Process view:

  OMD[heute]::$ ps -ef | grep liveproxyd
  UID  PID  PPID  C STIME  TTY   TIME CMD
  heute 9261   1  0 11:40 ? 00:00:00 liveproxyd[master]
  heute 9262 9261  0 11:40 ? 00:00:00 liveproxyd[heute_slave_1]
  heute 9263 9261  0 11:40 ? 00:00:00 liveproxyd[heute_slave_2]

• More resilient, more performant through use of multiple CPUs (cf. Werk #4901)
Make things feel faster - Graphs

• Previously: View with many Graphs? Better wait for all the data

• Now: Load the view with placeholders, update graphs asynchronously
Take-home message

- Attention to detail pays off
- Continuously improving
- If you have lots of sites and lots of users, Werk #4921 may help you