Checkmk Conference #11

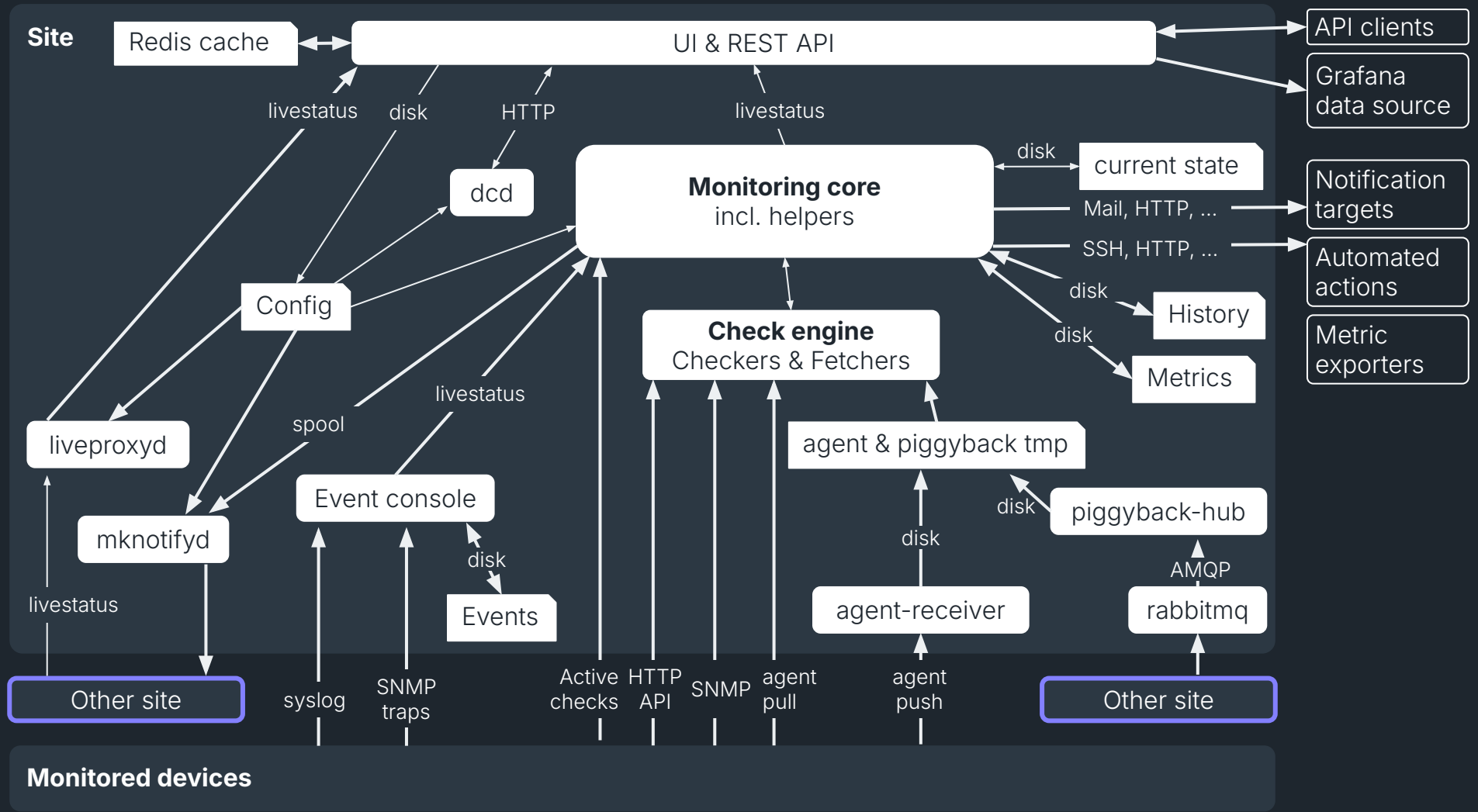
Performance tuning and more — Checkmk under the hood

May 20, 2025 - Live from Paulaner am Nockherberg, Munich

Lars Michelsen
Chief Technology Officer
Checkmk GmbH

Checkmk architecture

Tuning Checkmk

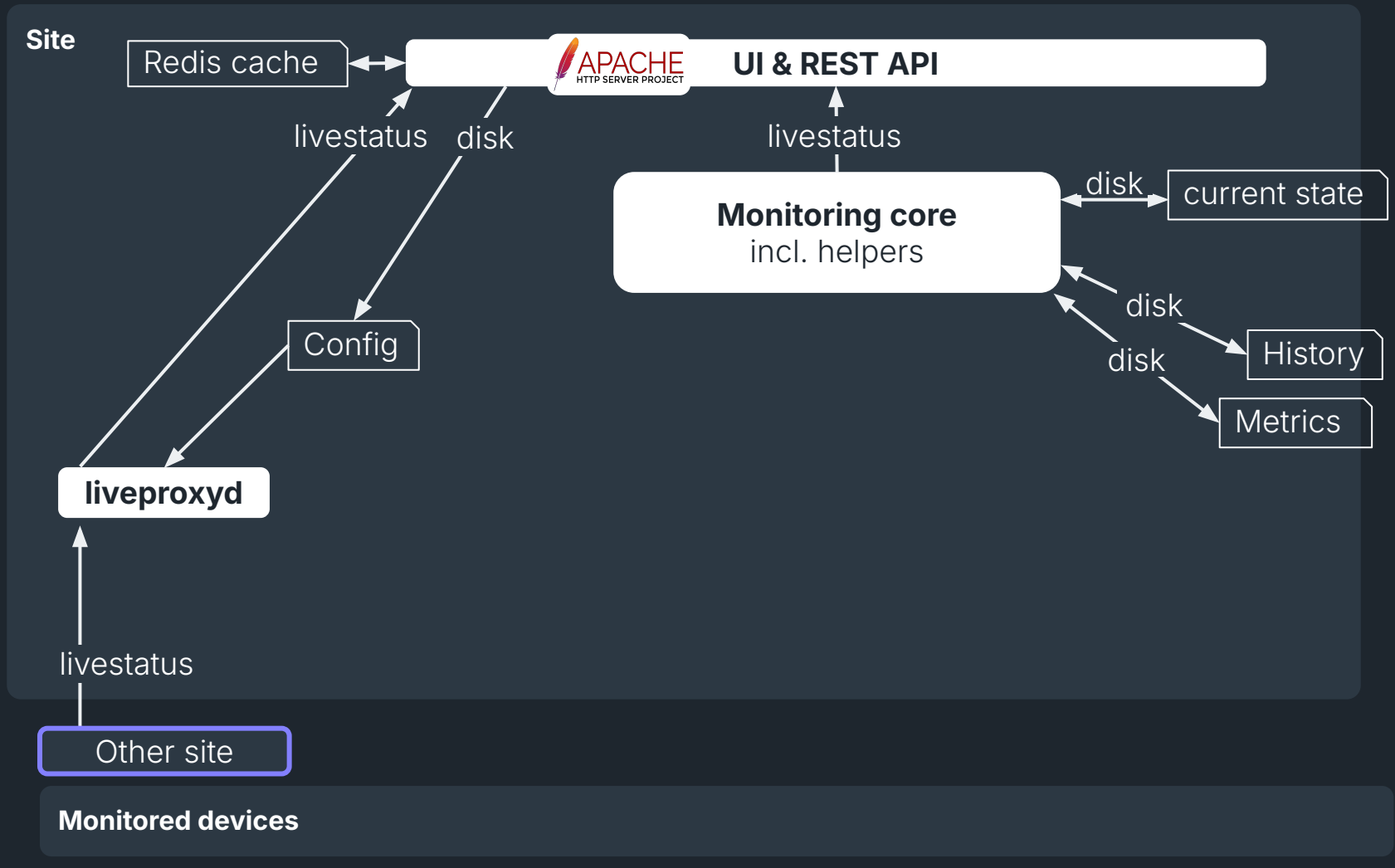

Site
Redis cache
UI & REST API
API clients
Grafana data source
livestatus
disk
HTTP
livestatus
dcd
Monitoring core
incl. helpers
disk
current state
Mail, HTTP, ...
Notification targets
SSH, HTTP, ...
Automated actions
Config
disk
History
disk
Metrics
Metric exporters
Check engine
Checkers & Fetchers
livestatus
spool
liveproxyd
agent & piggyback tmp
Event console
mknotifyd
disk
piggyback-hub
disk
disk
Events
AMQP
livestatus
agent-receiver
rabbitmq
Other site
syslog
SNMP traps
Active checks
HTTP API
SNMP
agent pull
agent push
Other site
Monitored devices

Checkmk architecture

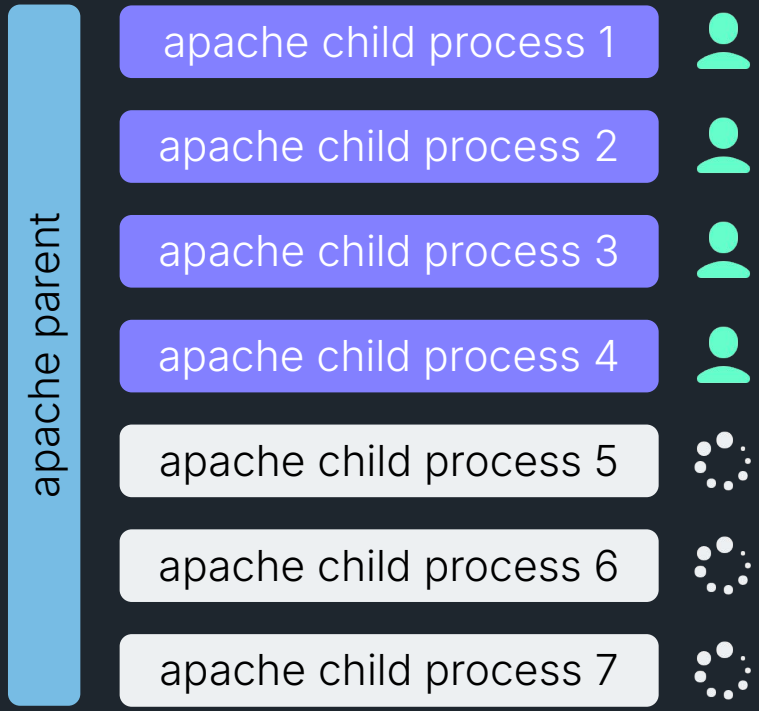
Tuning Checkmk

Tuning for best performance

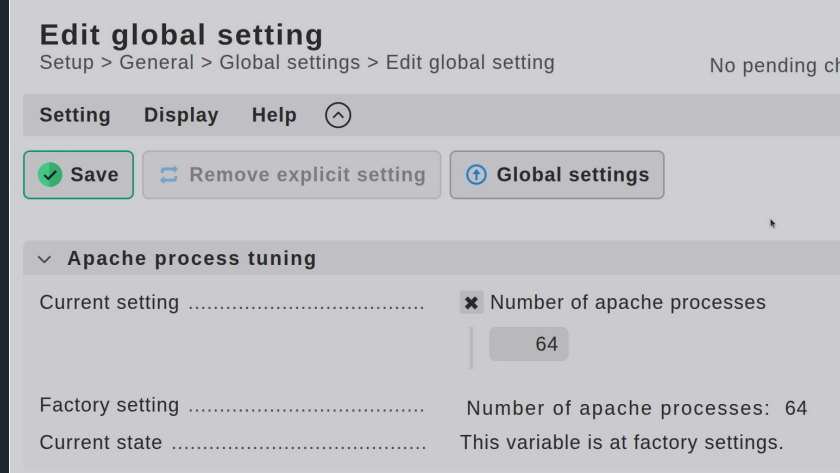
UI &
REST API


Site
Redis cache
APACHE
HTTP SERVER PROJECT
UI & REST API
livestatus
disk
livestatus
Monitoring core
incl. helpers
disk
current state
disk
History
disk
Metrics
Config
liveproxyd
livestatus
Other site
Monitored devices

Checkmk Apache 101: prefork



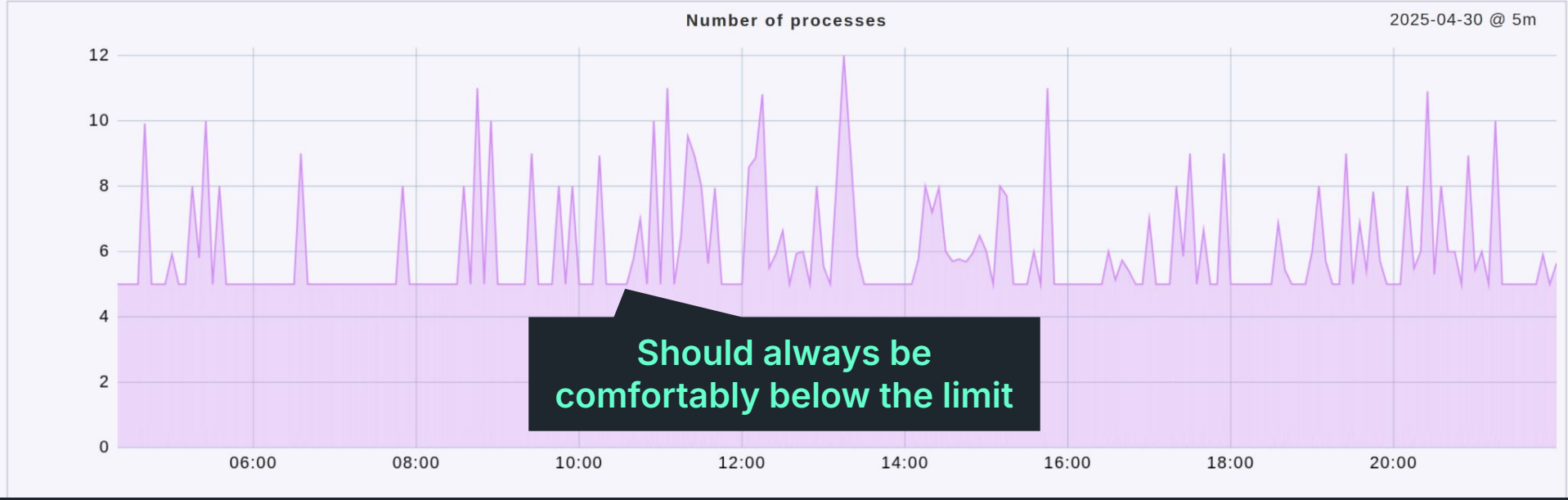
●●● *up to 64 (default)*



Default too high in most cases!

Process monitoring for Apache (built-in since 2.4.0)

OK	Process stable apache		Processes: 5, Virtual memory: 969 MiB, Resident memory: 748 MiB, CPU: 1.52%, Youngest running for: 43 minutes 16 seconds, Oldest running for: 7 hours 23 minutes	5



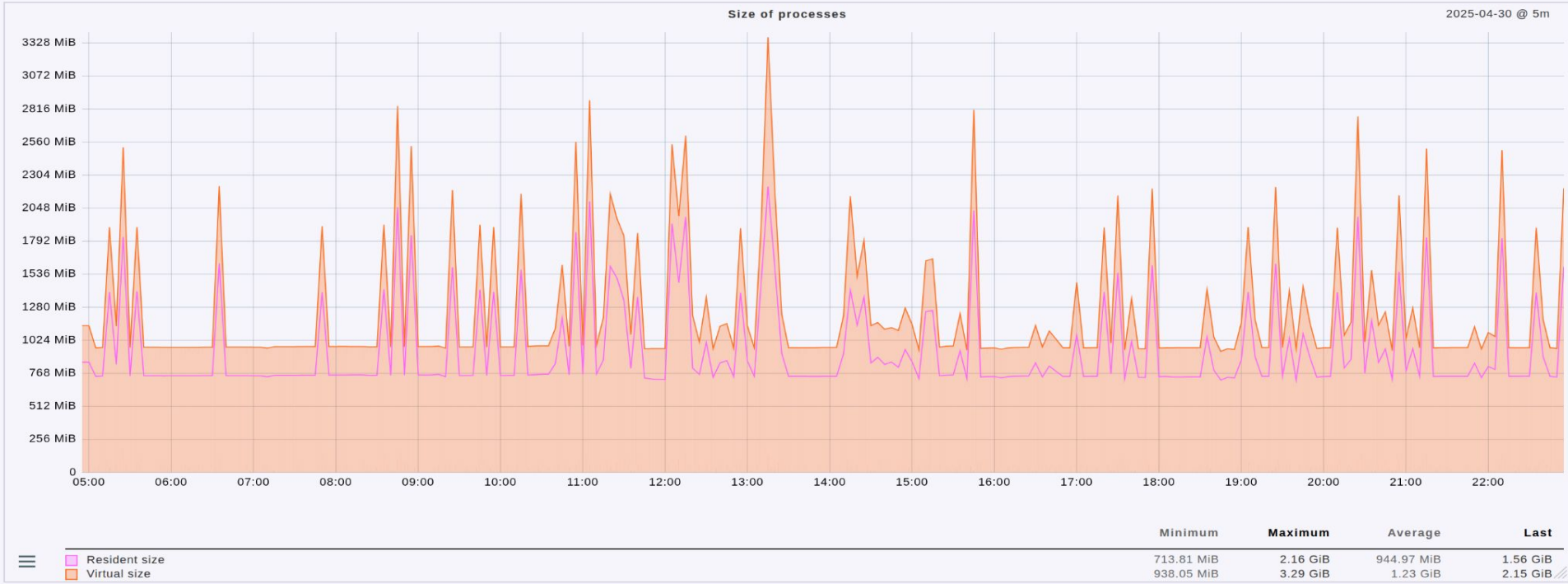
Process monitoring added for all key components

During update from 2.3 to 2.4

```
-|  14/36 Process discovery for self monitoring...
-| Adding: Shipped rule to monitor sites rrdcached
-| Adding: Shipped rule to monitor sites rrd helper
-| Adding: Shipped rule to monitor sites redis-server
-| Adding: Shipped rule to monitor sites real-time helper
-| Adding: Shipped rule to monitor sites rabbitmq
-| Adding: Shipped rule to monitor sites piggyback hub
-| Adding: Shipped rule to monitor sites notify helper
-| Adding: Shipped rule to monitor sites notification spooler
-| Adding: Shipped rule to monitor sites livestatus proxy
-| Adding: Shipped rule to monitor sites jaeger
-| Adding: Shipped rule to monitor sites fetcher helpers
-| Adding: Shipped rule to monitor sites event console
-| Adding: Shipped rule to monitor sites dcd
-| Adding: Shipped rule to monitor sites cmc
-| Adding: Shipped rule to monitor sites checker helpers
-| Adding: Shipped rule to monitor sites automation helpers
-| Adding: Shipped rule to monitor sites apache
-| Adding: Shipped rule to monitor sites alert helper
-| Adding: Shipped rule to monitor sites agent receiver
-| Adding: Shipped rule to monitor sites active check helpers
```

Apache is memory hungry

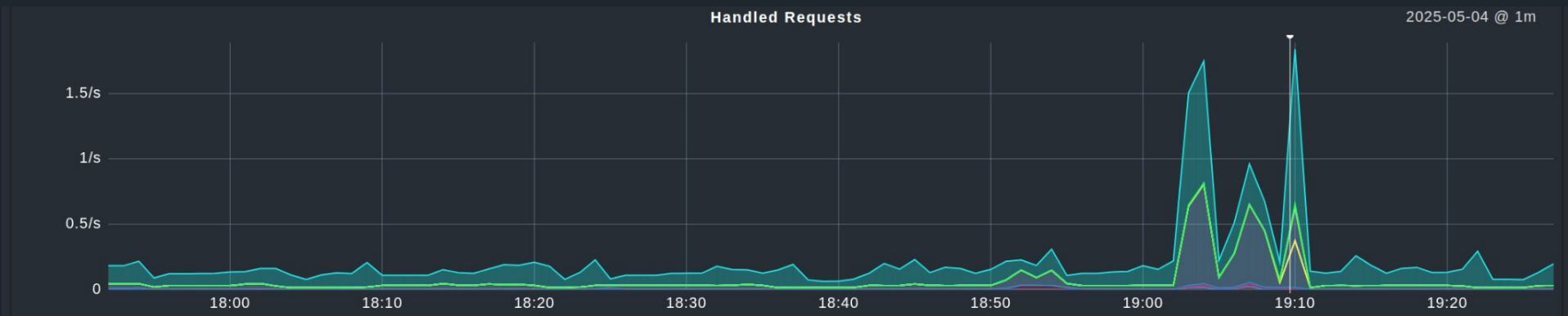
Avoid OOM. Also check your normal Memory service!



Verify that: Total apache memory demand < estimated RAM for new processes

Can’t scale further? Look at OMD Apache monitoring

OK | OMD stable apache | Requests: 0.02/s, Seconds serving: 0.01/s, Sent: 8.56 kB/s



	Minimum	Maximum	Average	Last	2025-05-04 19:09
Other: Requests	0.05/s	1.19/s	0.15/s	0.17/s	1.19/s
Scripts: Requests	0/s	0.02/s	0.0004/s	0/s	0.02/s
Styles: Requests	0/s	0.02/s	0.0002/s	0/s	0.02/s
Image: Requests	0/s	0.25/s	0.003/s	0/s	0.25/s
NagVis: Other: Requests	0/s	0/s	0/s	0/s	0/s
NagVis: AJAX: Requests	0/s	0/s	0/s	0/s	0/s
NagVis: Sidebar element: Requests	0/s	0/s	0/s	0/s	0/s
Checkmk: Other: Requests	0.02/s	0.76/s	0.06/s	0.03/s	0.36/s
Checkmk: Dashboards: Requests	0/s	0.02/s	0.0005/s	0/s	0.02/s
Checkmk: Sidebar elements: Requests	0/s	0.03/s	0.002/s	0/s	0/s

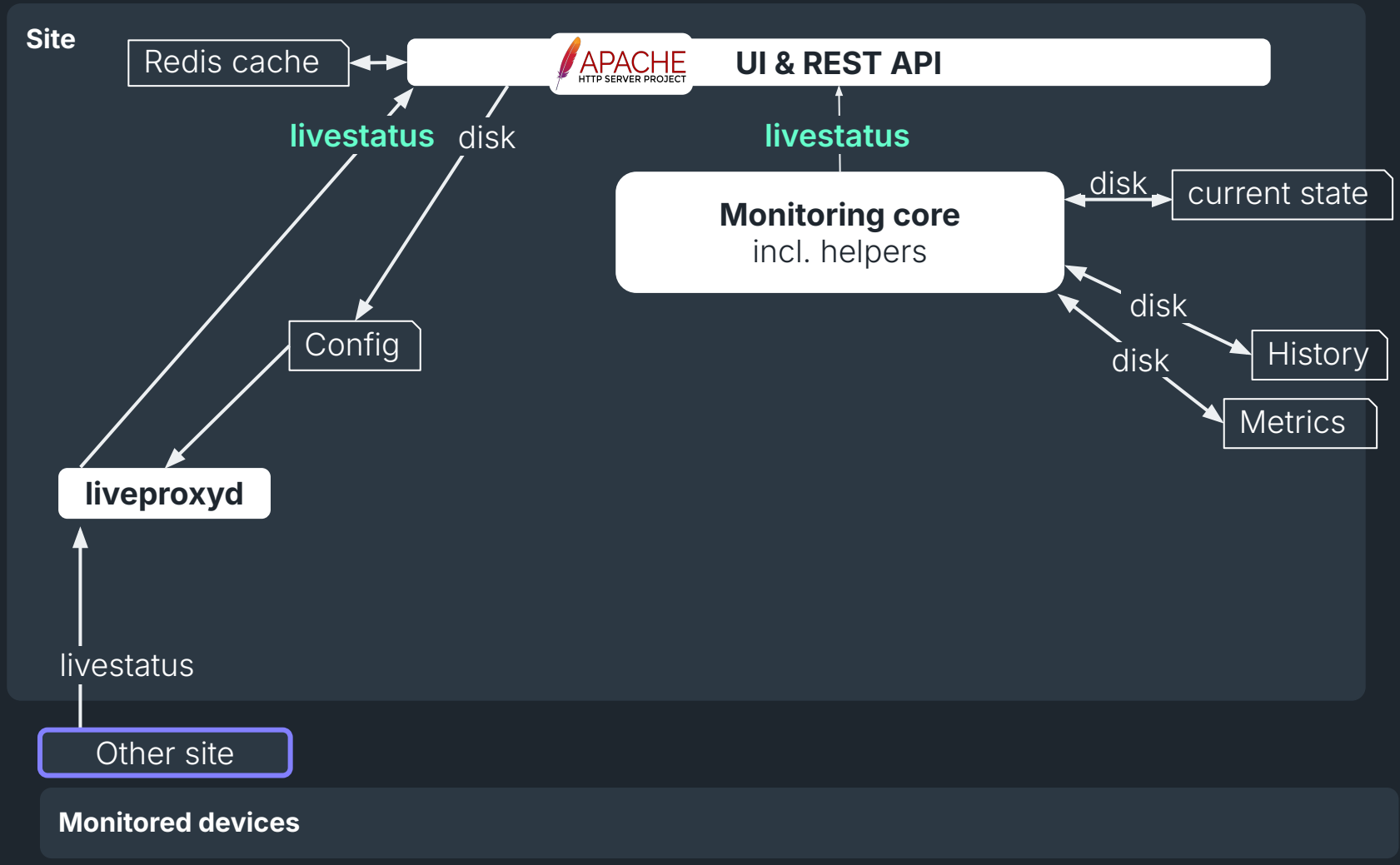
Slow view log

May help to identify slow (custom) views

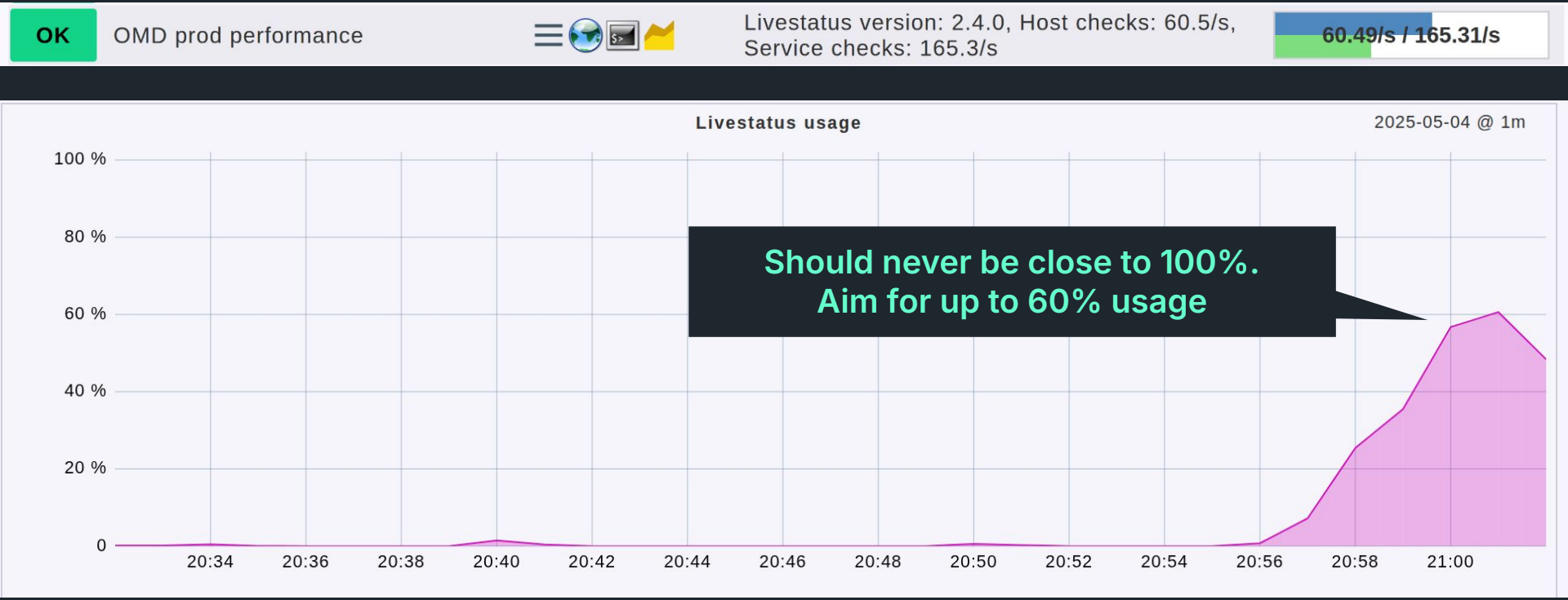
- When enabled, one log entry per view rendering that exceeds the configured threshold is logged to `var/log/web.log` and contains the following information
- See werk #11517 for further information
- Possible ways to improve view load times:
  - Tune regexes for efficiency
  - Reduce data to be fetched
    (limit number of rows, increase update interval)

```
OMD[lm]:~$ tail -f
var/log/web.log

2025-05-06 14:33:25,918 [10]
[cmk.web.slow-views 793137]
View name: host,
User: cmkadmin,
Row limit: 1000,
Limit type: soft,
URL variables: ['host=beta',
'site=beta', 'view_name=host',
'_display_options=htbfcoderu',
'_ajaxid=0'],
View context: {'siteopt': {},
'svcstate': {}, 'serviceregex':
{}, 'host': {'host': 'beta',
'neg_host': ''}, 'site': {'site':
'beta'}},
Unfiltered rows: 1337,
Filtered rows: 1337,
Rows after limit: 0,
Duration fetching rows: 15.03s,
Duration filtering rows: 15.00s,
Duration rendering view: 15.23s,
Rendering page exceeds 0s: 15.29s
```

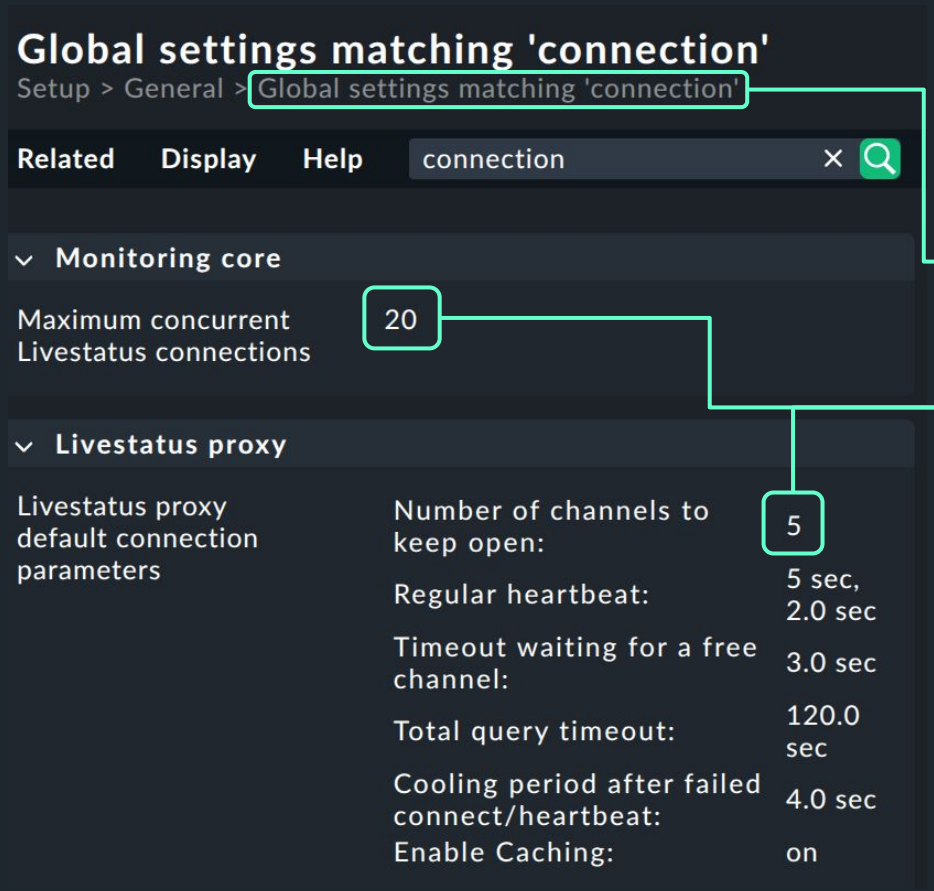

Site
Redis cache
APACHE
HTTP SERVER PROJECT
UI & REST API
livestatus
disk
livestatus
Monitoring core
incl. helpers
disk
current state
disk
History
disk
Metrics
Config
liveproxyd
livestatus
Other site
Monitored devices

Livestatus performance monitoring



Configure via global setting: Maximum concurrent Livestatus connections

Balance with livestatus proxy channels



Livestatus proxy establishes N fixed connections, known as channels

- Configure in global settings or site connection
- Must be lower than the number of maximum concurrent livestatus (here 5<20)
  - This buffer allows for ad-hoc connections

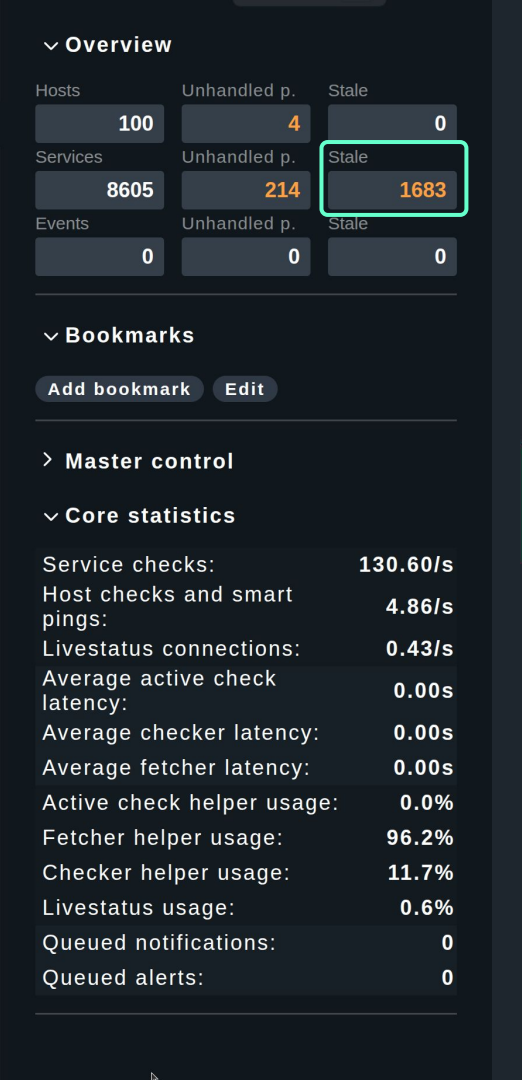
Use `cat var/log/liveproxyd.state` to get liveproxyd usage details

Tuning for best performance

UI & REST API

Stale services

Checkmk staleness 101

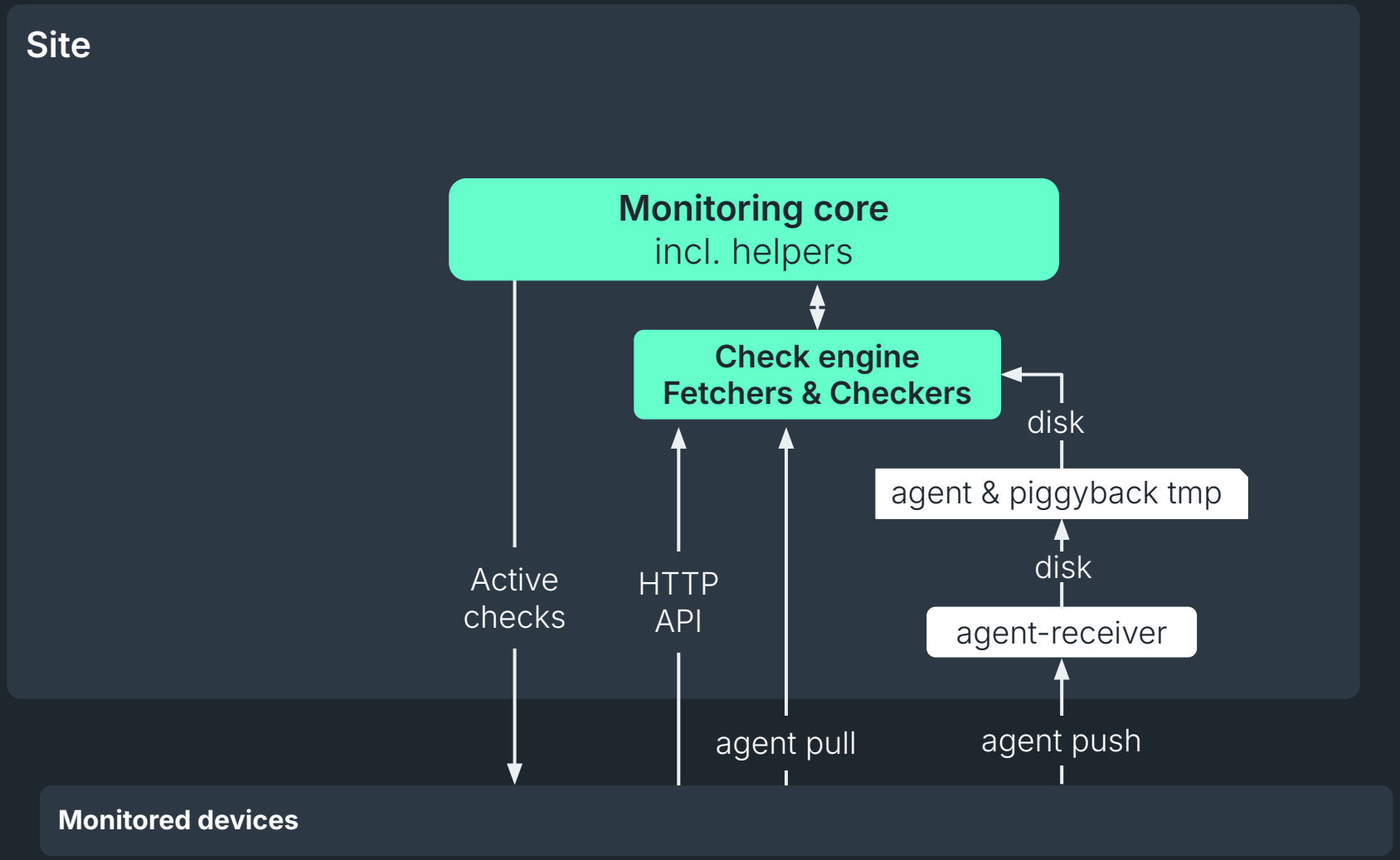


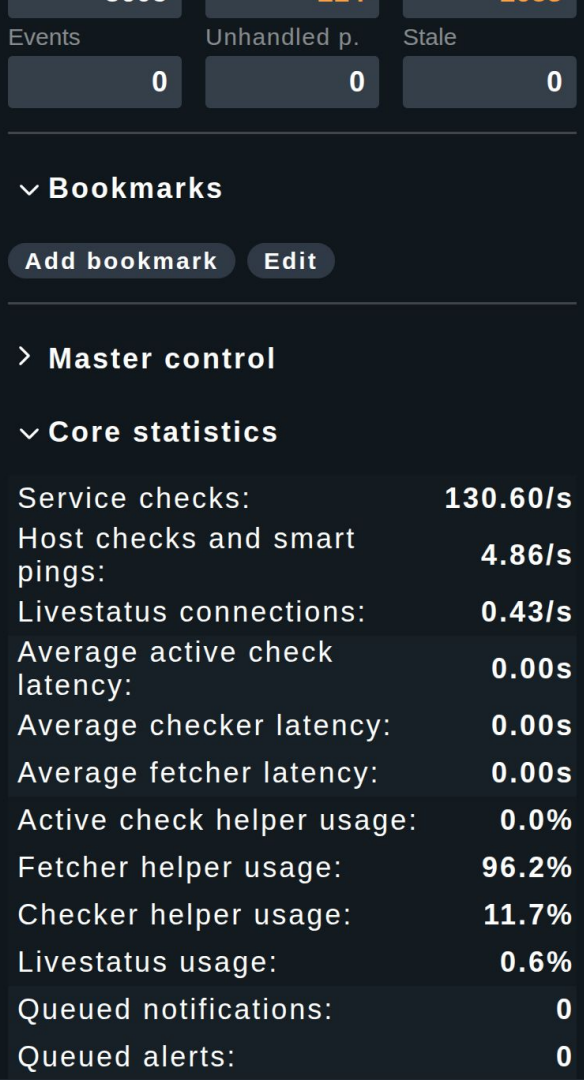
State	Service	Icons	Summary	Age	Checked
OK	Memory		Total virtual memory: 7.90% - 3.12 GiB of 39.6 GiB, 9 additional details available	10 m	102 s

Stale: not refreshed for the last 1.5 check intervals

Example: 90s, if check interval = 60s.

Note: Configurable via global setting: Staleness value to mark hosts / services stale


Site
Monitoring core
incl. helpers
Check engine
Fetchers & Checkers
disk
agent & piggyback tmp
disk
agent-receiver
Active
checks
HTTP
API
agent pull
agent push
Monitored devices



Reasons for staleness

Check_MK and Check_MK discovery services

- Highly utilized fetcher helpers
- Highly utilized checker helpers

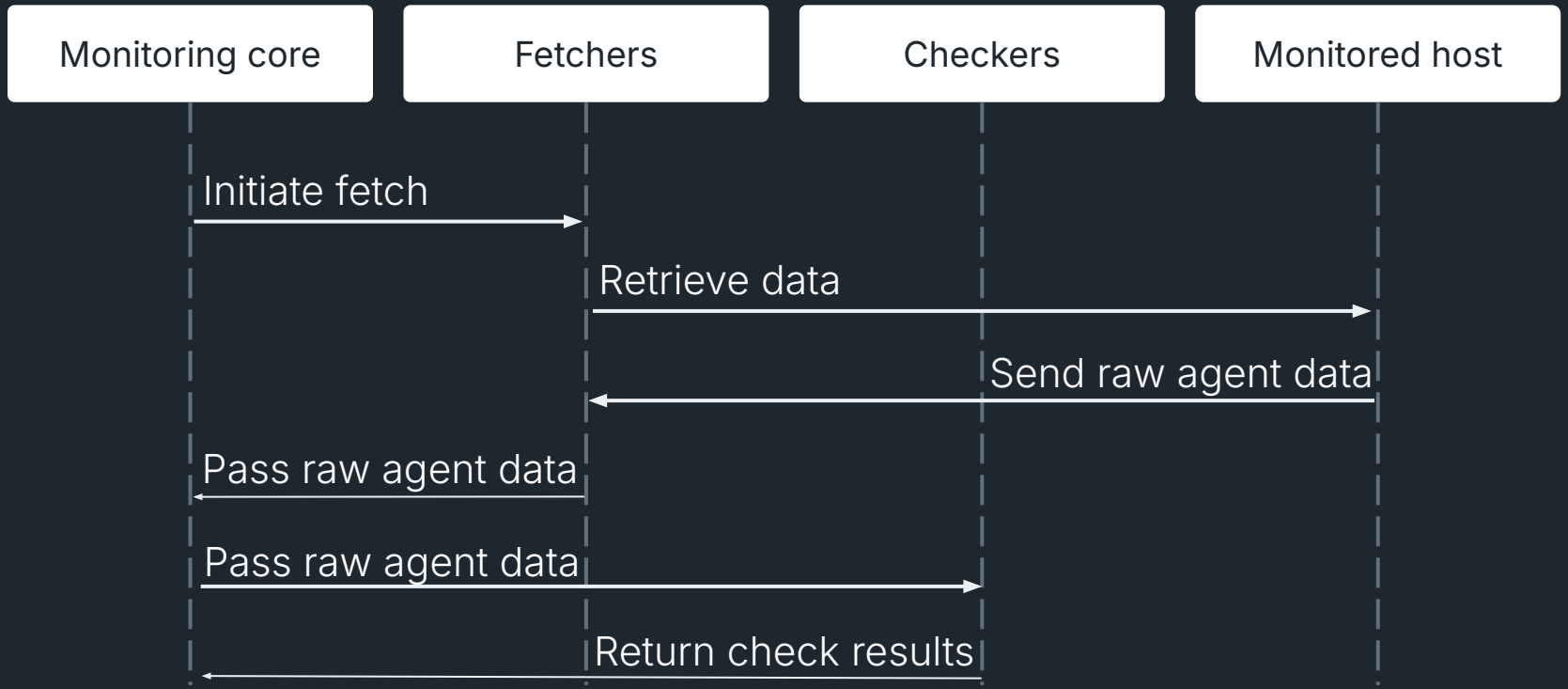
Active checks (Check_MK inventory , HTTP, TCP, ...)

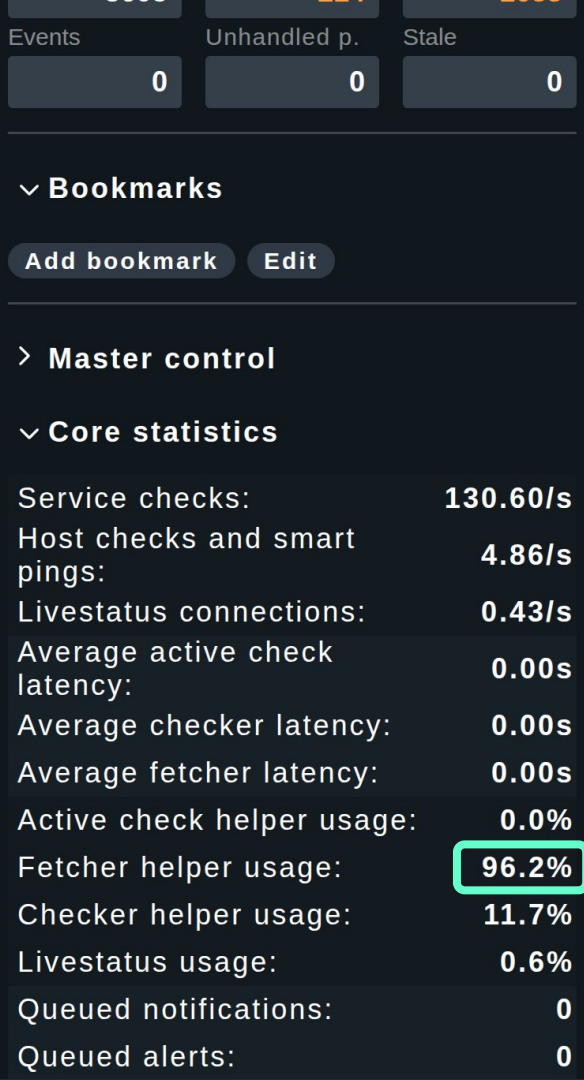
- Highly utilized active check helpers

Other services, like CPU load or Memory

- Typically related to a failing Check_MK service

An execution of a Check_MK service





Reasons for staleness

Check_MK and Check_MK discovery services

- Highly utilized fetcher helpers
- Highly utilized checker helpers

Active checks (Check_MK inventory , HTTP, TCP, ...)

- Highly utilized active check helpers

Other services, like CPU load or Memory

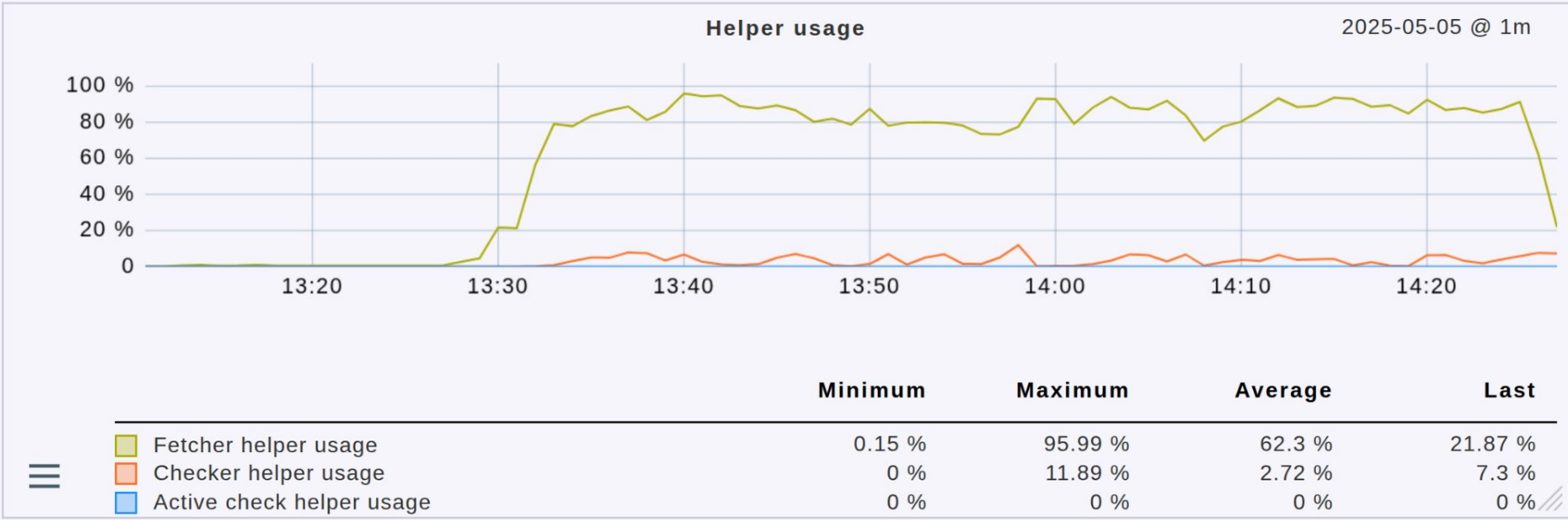
- Typically related to a failing Check_MK service

Helper performance monitoring

OK OMD prod performance

Livestatus version: 2.4.0, Host checks: 60.5/s, Service checks: 165.3/s

60.49/s / 165.31/s



	Minimum	Maximum	Average	Last
Fetcher helper usage	0.15 %	95.99 %	62.3 %	21.87 %
Checker helper usage	0 %	11.89 %	2.72 %	7.3 %
Active check helper usage	0 %	0 %	0 %	0 %

Configure via global setting: Maximum concurrent Checkmk fetchers / checkers / active checks

Guidance for configuring fetchers/checkers

Fetchers *(default: 13)*

- Network bound
- ~70MB per process
- Limiting factor: Memory
- Too many slow down startup
- Aim for less than 60% utilization

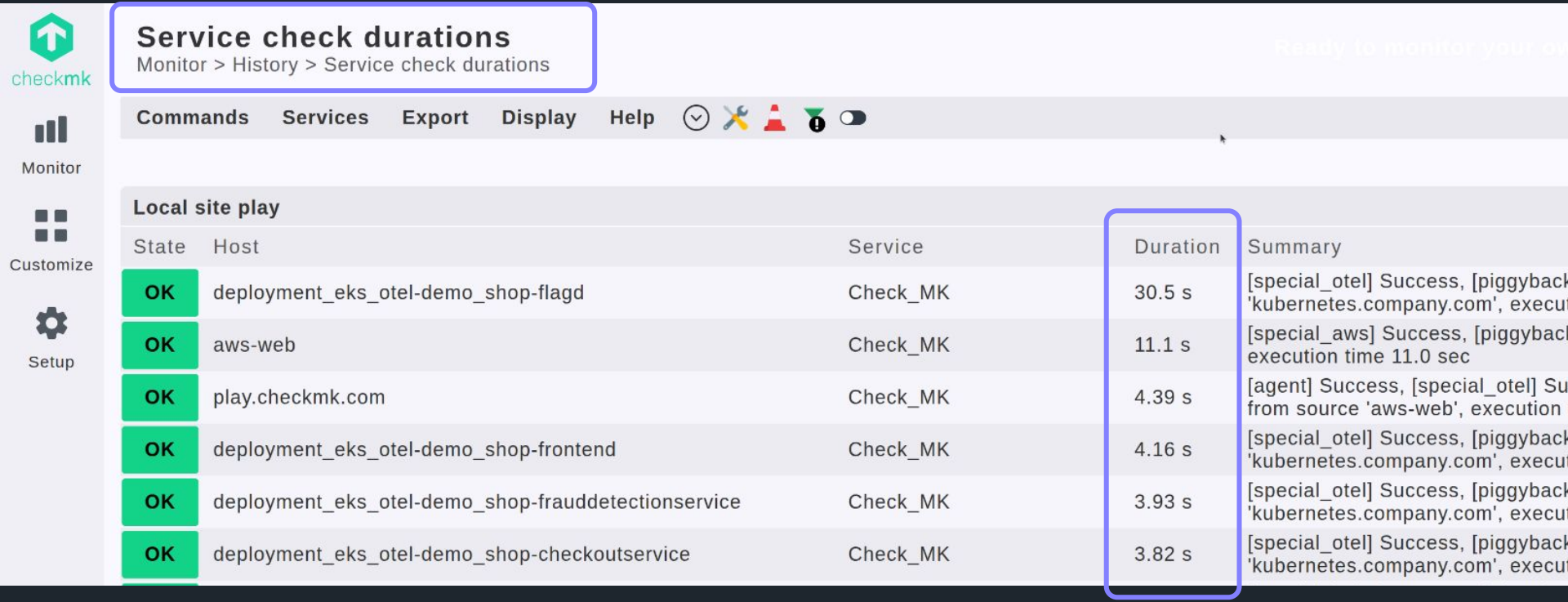
Checkers *(default: 4)*

- CPU bound
- >200MB per process
- Limiting factors: CPU, Memory
- # checkers smaller than # CPU cores
- Too many checkers ⇒ Context switches
- Aim for less than 80% utilization

Also covered by new process monitoring!

Tuning: Identify long-running checks

Checkers only apply to Check_MK & Check_MK Discovery service!



Tuning fetchers: 99% of time tuning SNMP

https://checkmk.com/blog/snmp-troubleshooting


11