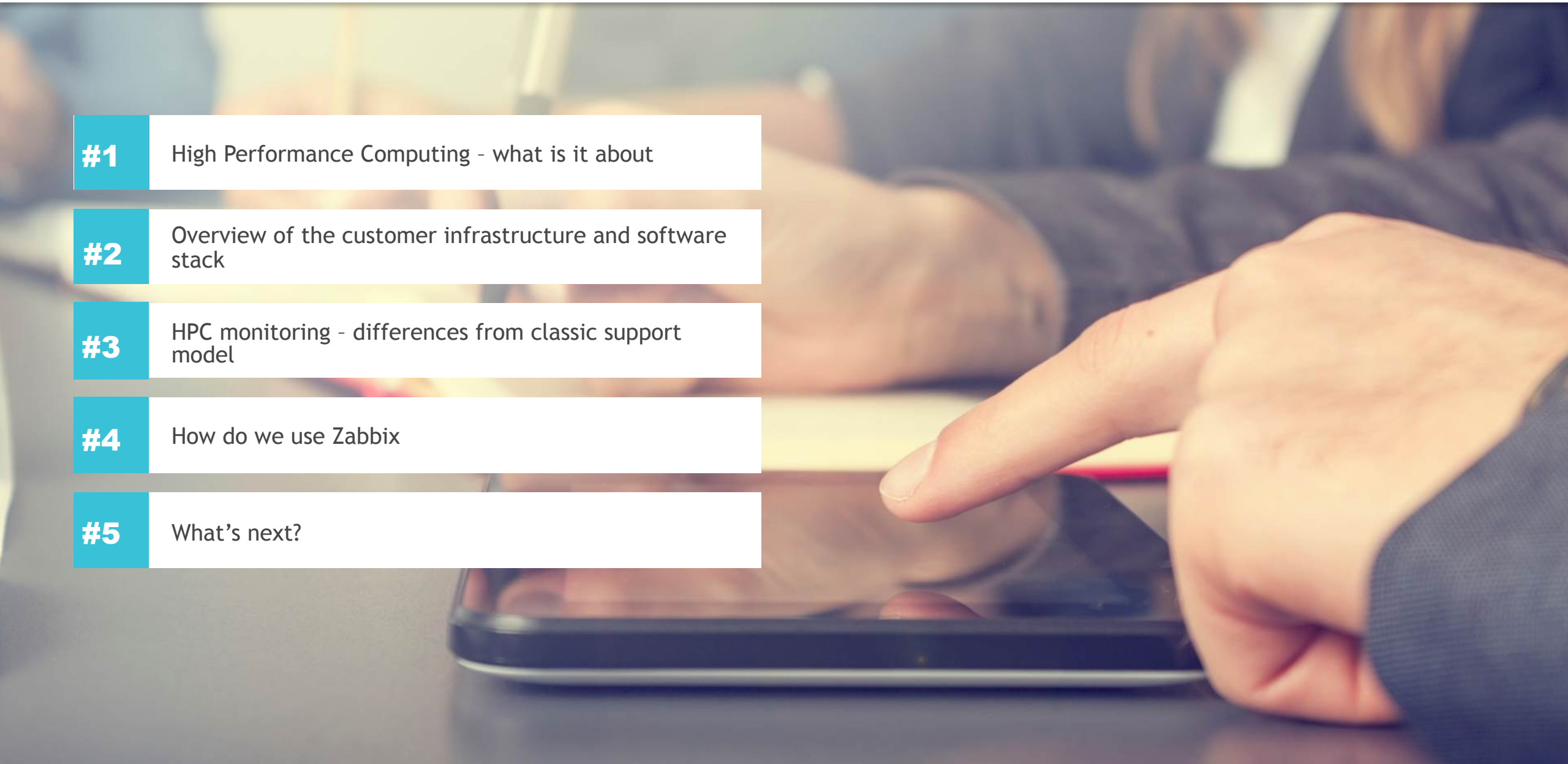
ZABBIX FOR HPC MONITORING AND SUPPORT

Mikhail Serkov

Delivery Manager/HPC Engineer

2016

AGENDA



#1	High Performance Computing – what is it about
#2	Overview of the customer infrastructure and software stack
#3	HPC monitoring – differences from classic support model
#4	How do we use Zabbix
#5	What’s next?

Novartis Institute For Biomedical Research (NIBR)

- Scientific research in Pharma area: Bioinformatics, Computational Chemistry, Drug Discovery, etc.
- About 10k CPU cores used for a scientific computation.
- Shared clusters - different workflows could run simultaneously within the same cluster.
- About 500 different scientific tools.
- Custom software (Python, Java, R)

HIGH PERFORMANCE COMPUTING

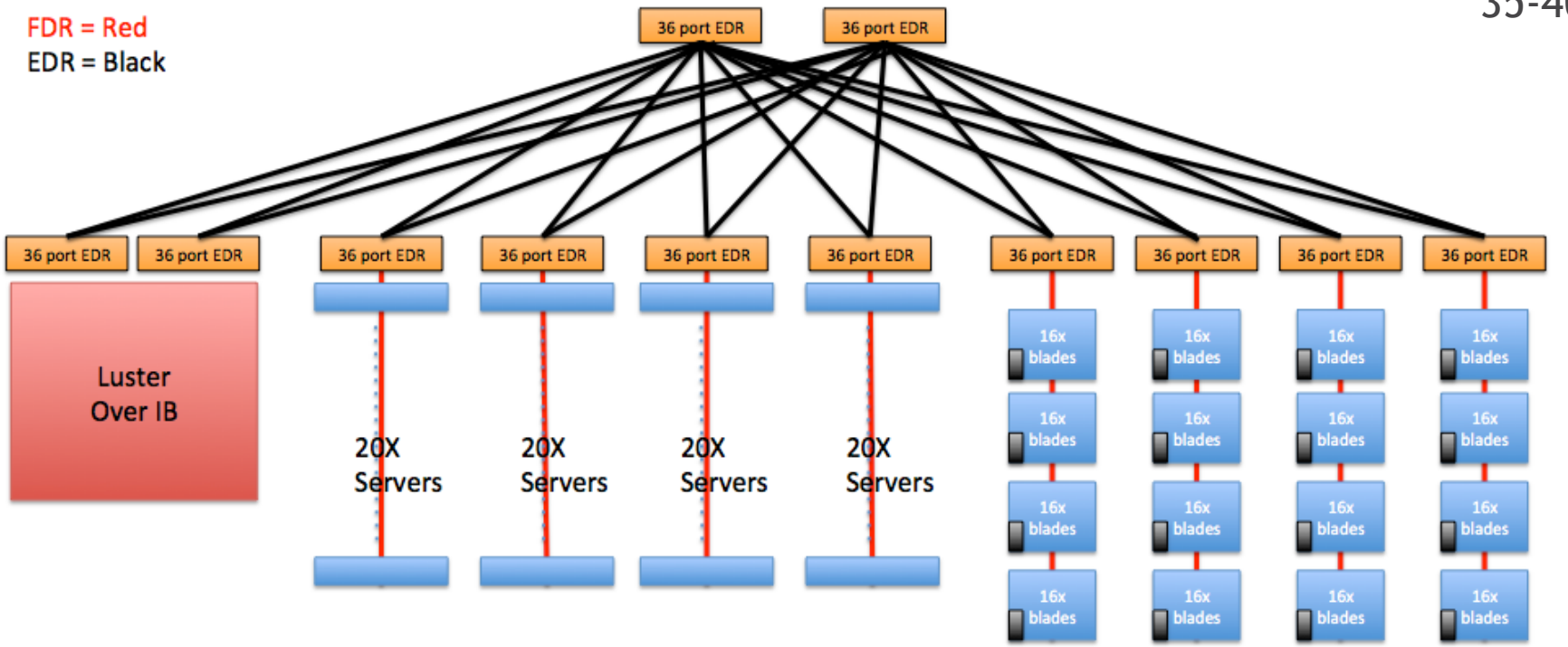
- Hundreds or even thousands of computation nodes
- Grid Computing technologies and software (SGE, UGE, SoGE, PBS, etc)
- Massive parallel computation across the nodes
- Strong requirements for all subsystems on hardware and software level (storage, network, power, OS)
- No magic. Linux boxes, shell scripts on a low level ☺️

Example of a job submission:

```
qsub -pe smp 10 -l m_mem_free=4G,h_rt=3600,lustre=1,gpu_card=2 workflow.sh
```

OVERVIEW OF THE CUSTOMER INFRASTRUCTURE

250 GPU's
70TB RAM
35-40KW/Rack



TYPICAL COMPUTATION NODE CONFIGURATION

- 28 CPU cores (2 sockets x 14 cores each)
  - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
- 200 GB RAM
- 10 GB Ethernet + InfiniBand interfaces
- 8 GPU cores (4 cards x 2 cores each)
- NFS over 10 GB Ehternet
- Lustre over InfiniBand

OVERVIEW OF SOFTWARE STACK

- More than 500 of scientific tools
- Bioinformatics, Computation Chemistry, Xtallography, Molecular Dynamics, etc
- RHEL6.5
- Univa Grid Engine
- Zabbix 2.4

HPC MONITORING DIFFERENCES

- We need information like ‘who, what, when’, not only system metrics.
- Users are allowed to run whatever they want using grid scheduler on the computation nodes.
- 100% CPU utilization and 100% RAM utilization for node is perfectly fine.
- Node crash – not such a big deal.
- Preventing global issues by using aggregated metrics.
- Metrics not only for monitoring but for a performance analysis.
- Users are having access to the monitoring system (but restricted).

WHY ZABBIX?

- Able to monitor of a huge systems with a lot of metrics
- Flexible
- **Out of the box**
- Ability to aggregate metrics
- API for a data extraction
- **GUI convenient for both support team and scientists**
- Autodiscovery
- New nodes automatic configuration

ZABBIX CONFIGURATION

Server configuration:

- 20 CPU cores (2 sockets x 10 cores each)
  - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
- 120 GB RAM

Number of hosts: 601

Number of items: ~200k

Number of triggers: ~37k

DB Size: 187GB

WHAT DO WE MONITOR

Local metrics (node level)	Global metrics (cluster level)
All default Linux checks (LA, CPU utilization, RAM, swap, etc) - agent	Meta CPU utilization – aggregation of CPU utilization of HPC nodes.
Every single GPU core (Temperature, Utilization if possible) - agent	NFS global transmit/retransmit - aggregation of nodes values
Every single CPU core (Utilization, Temperature) - agent	Grid specific – used/active slots, running jobs, pending jobs, top users - external scripts
NFS shares availability / utilization / mount details - agent	CPU/Memory oversubscription - aggregation of nodes values
Slots / RAM reserved - external scripts	Overloaded nodes - aggregation of HPC values
HPC jobs - external scripts	Pending time - external scripts
…	….

HPC specific examples

1) Expected utilization VS Real one

Every job has a resource request for number of CPUs, RAM, etc. In every moment we can compare real utilization with an expected one. If they are not close, we need to investigate if someone oversubscribing resources or overload nodes.

Solution: Zabbix not only checks current system metrics, but also keeps an expected values. If they are too different we receive warning.

2) Users on a computation node

Users are not restricted to SSH to any node (debugging, tracing job in real time, interactive jobs, etc). However we should check if user has job on the node he is logged into.

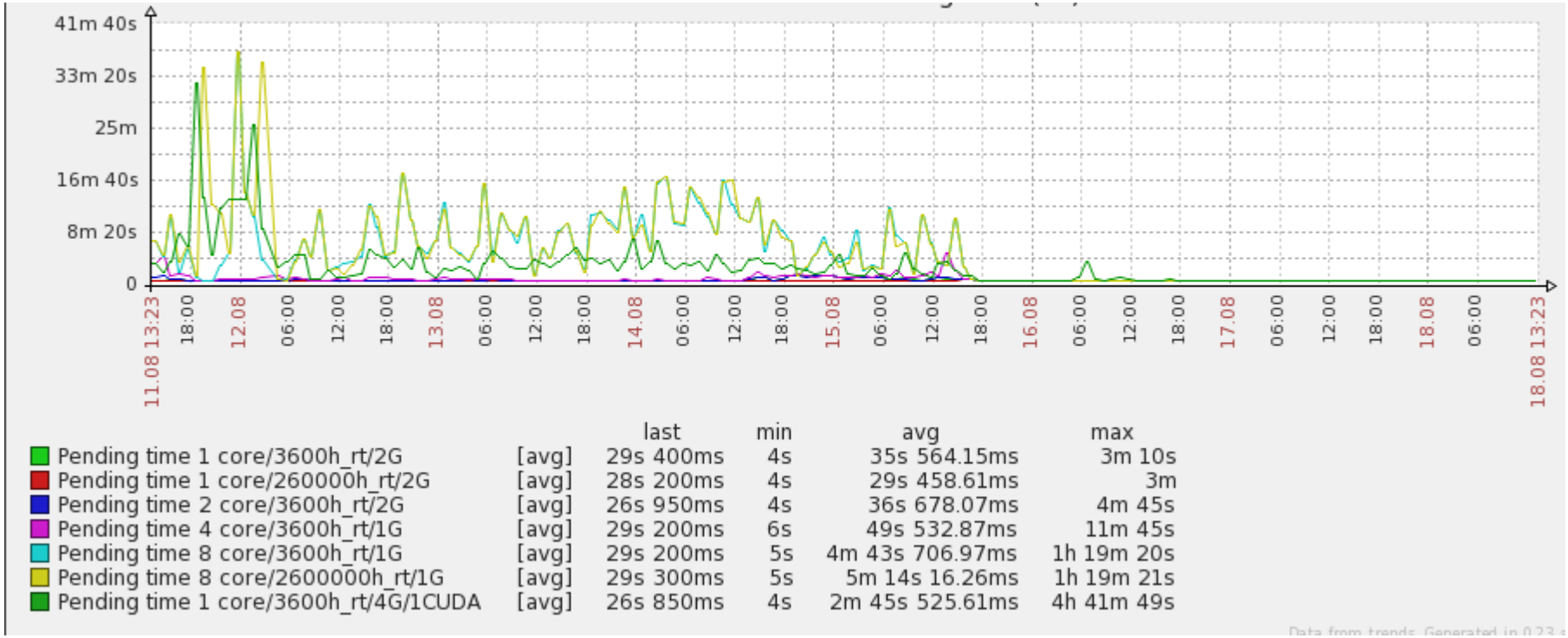
Solution: We have a trigger that notify us if we have anyone logged on the node with no job running. Additionally we store a list of logged in users for any single moment.

HPC specific examples

Pending time probes

It is really hard to predict the pending time for any particular job in the pending list, as they all have different resource requests, and runtimes. It is not a FIFO and the pending time is always related to resources user wants to have.

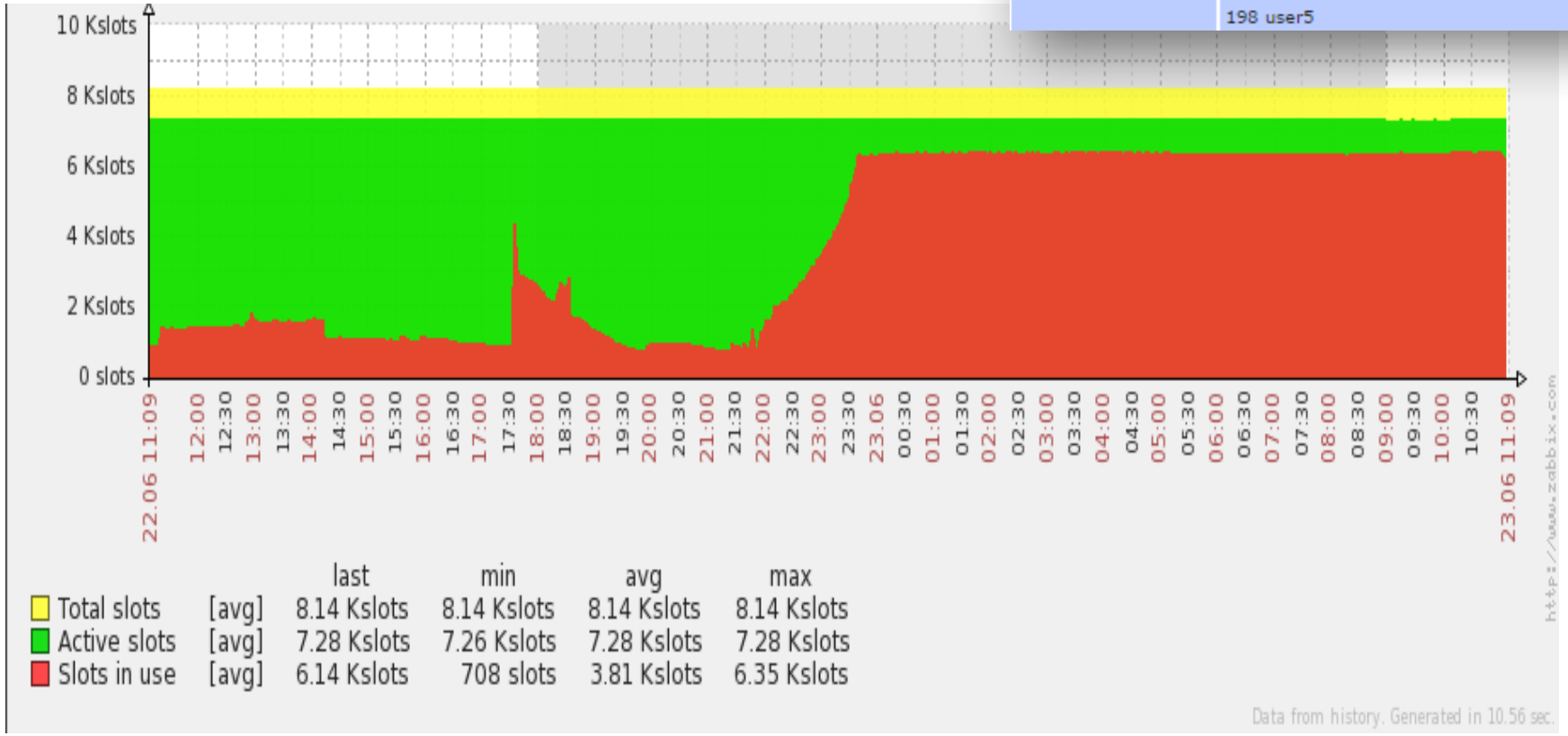
Solution: Zabbix runs ‘pending probes’ (empty jobs) and checks how long does it take. This is a good indicator for queue state at the moment.



WHAT DO WE MONITOR: GLOBAL METRICS

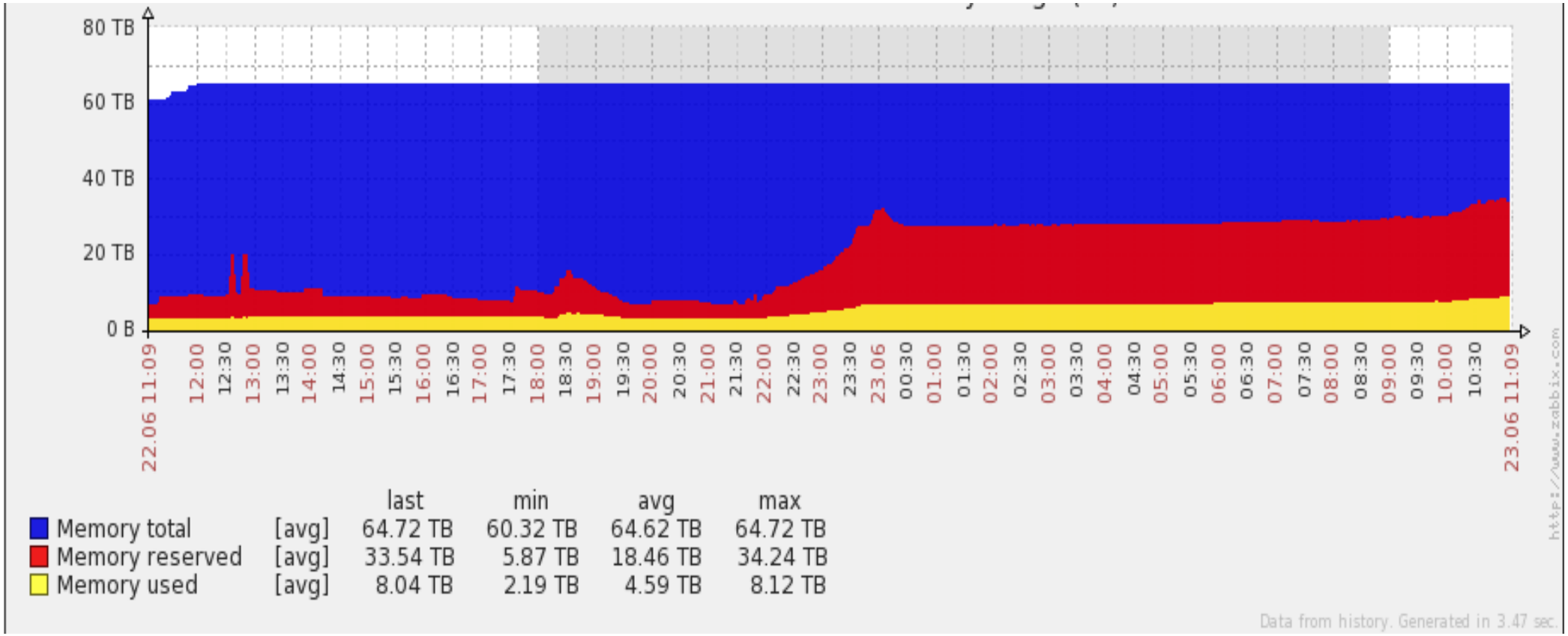
Timestamp	cluster.name: Top 5 users (slots)
2016-06-23 11:17:04	3556 user1 1153 user2 632 user3 512 user4 198 user5

Global cluster utilization



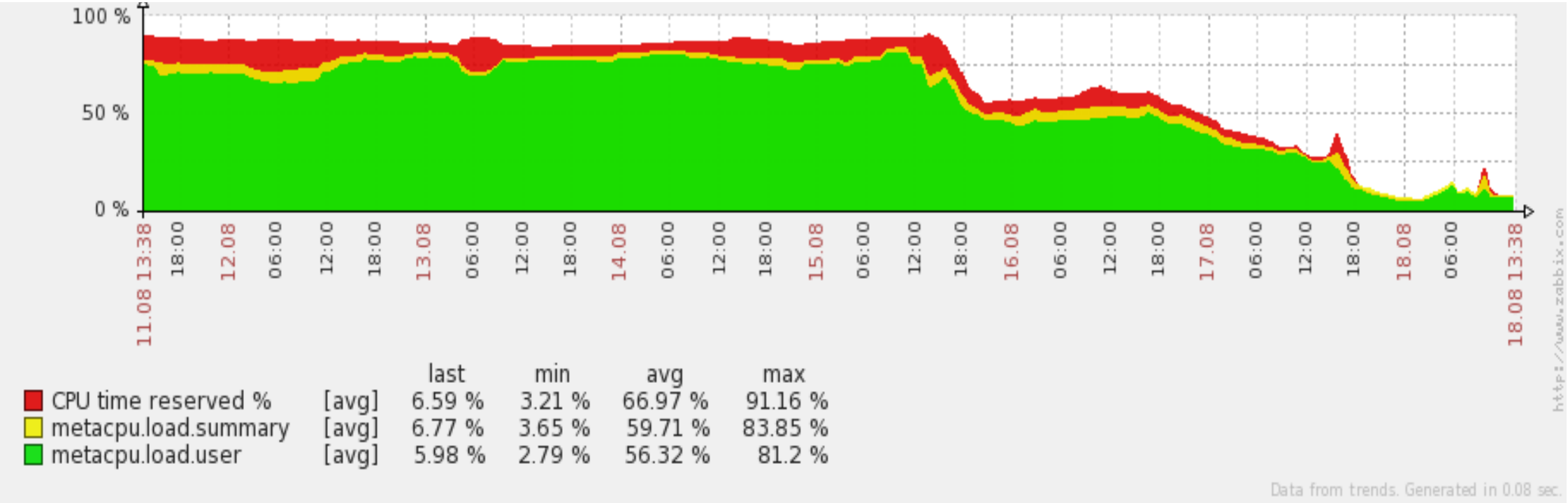
WHAT DO WE MONITOR: GLOBAL METRICS

RAM oversubscription



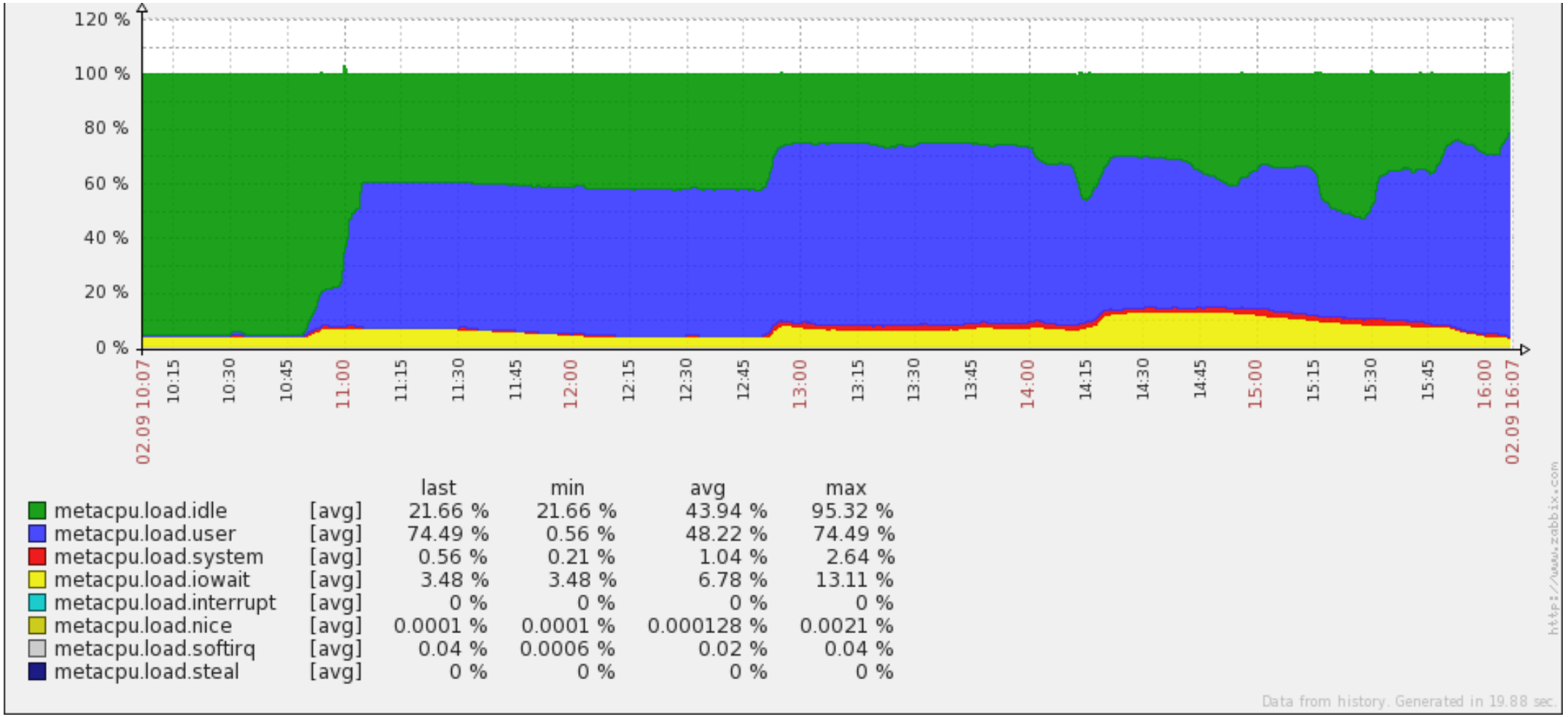
WHAT DO WE MONITOR: GLOBAL METRICS

CPU time oversubscription



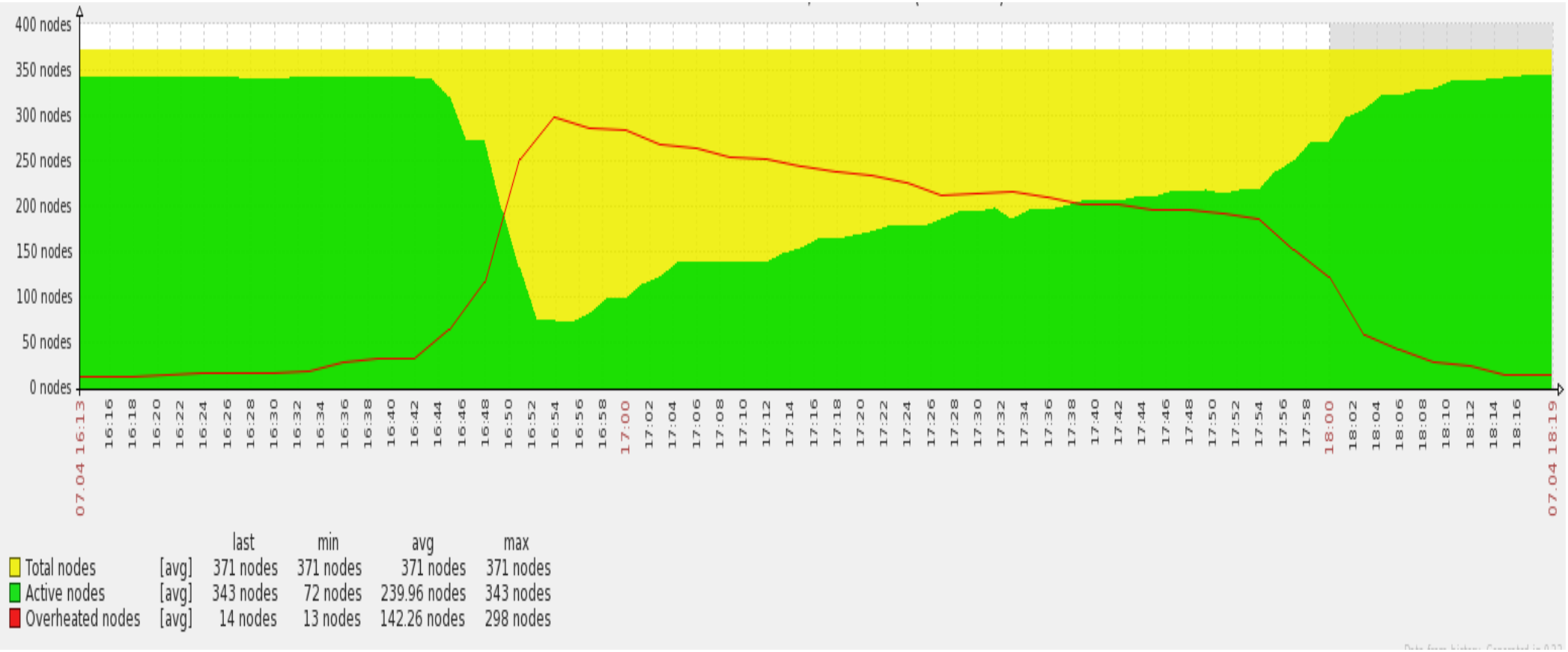
WHAT DO WE MONITOR: GLOBAL METRICS

Meta CPU utilization



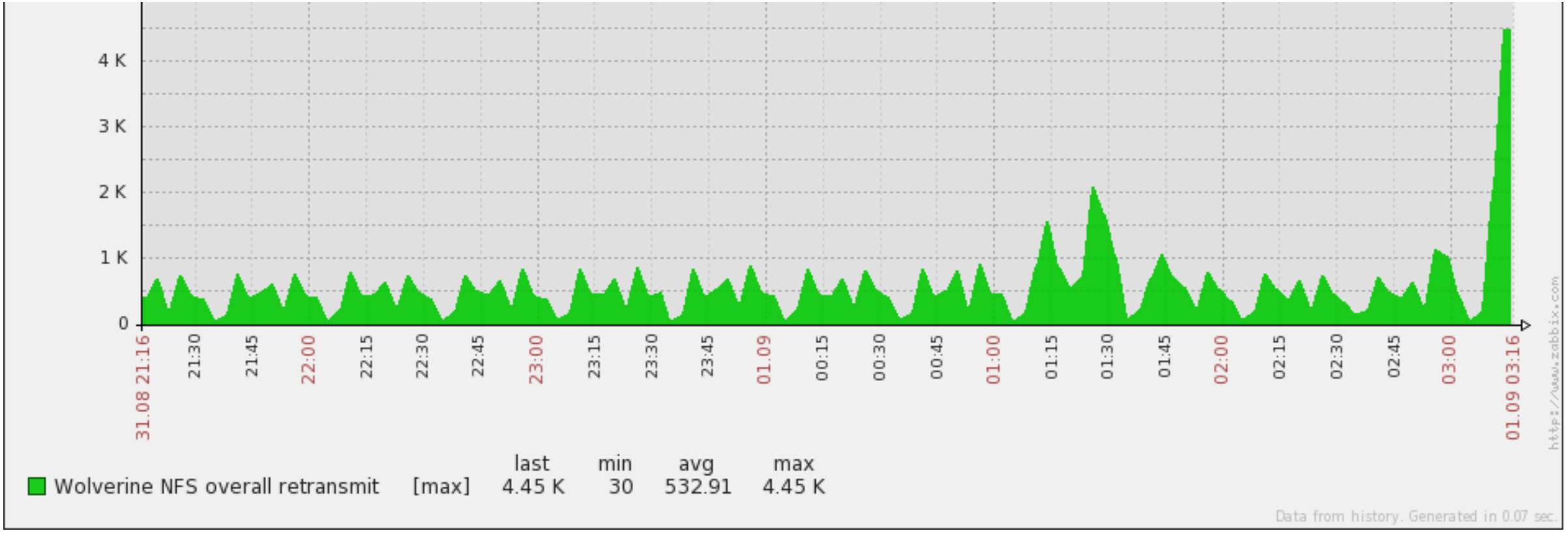
WHAT DO WE MONITOR: GLOBAL METRICS

Aggregated cluster status

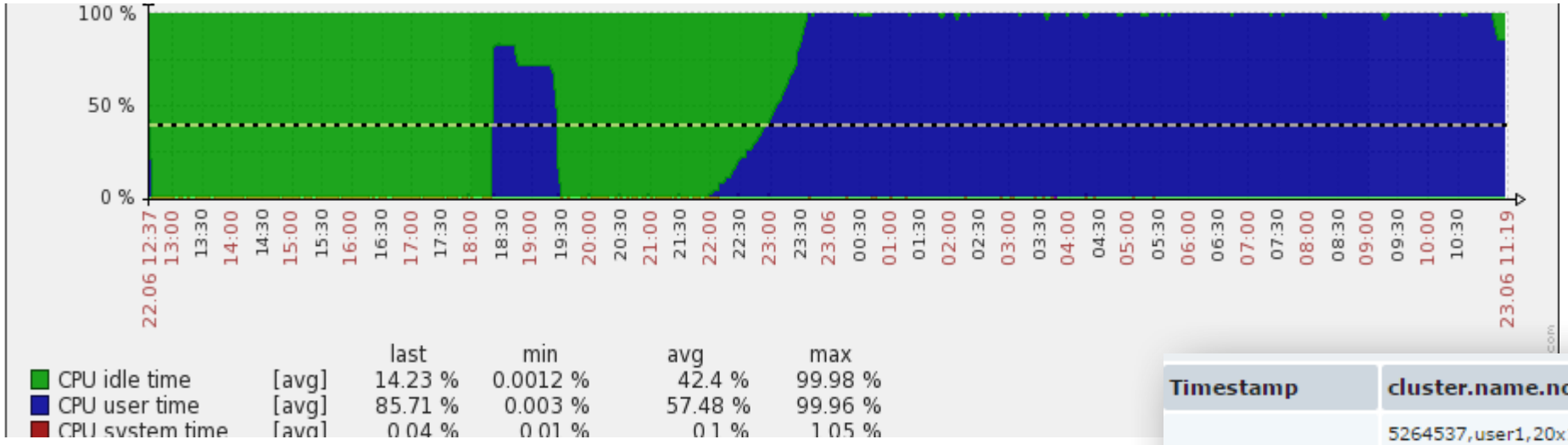


WHAT DO WE MONITOR: GLOBAL METRICS

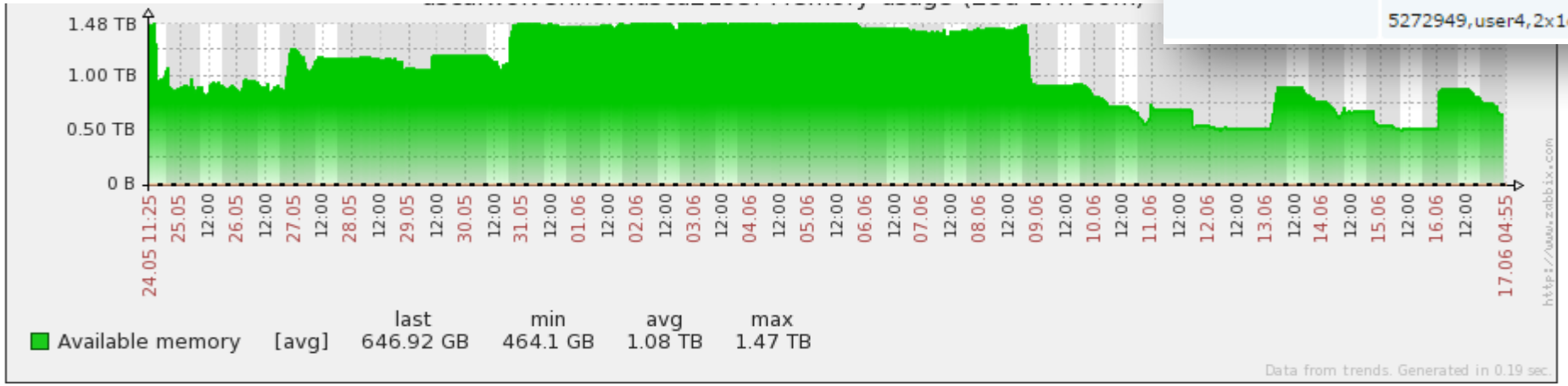
Storage operational metrics



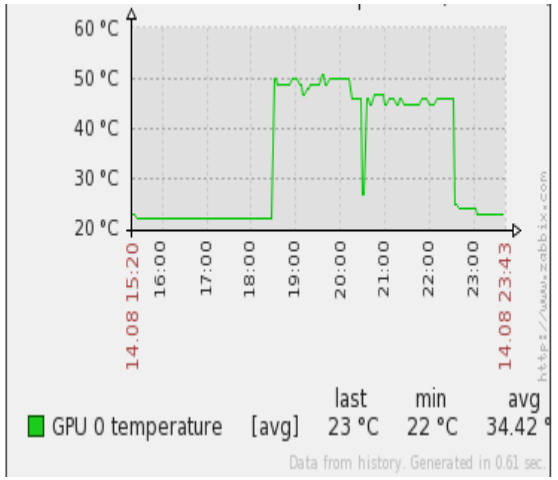
WHAT DO WE MONITOR: LOCAL METRICS

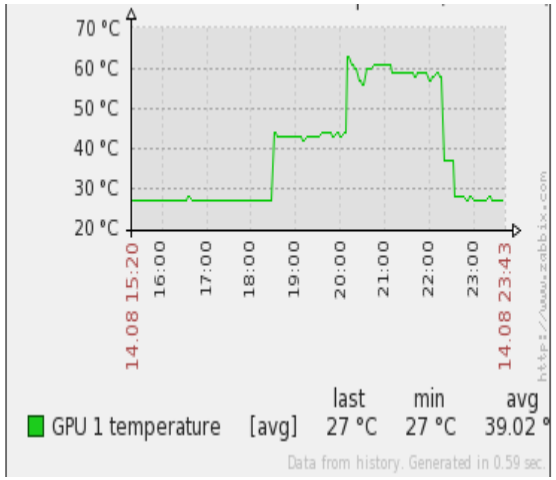


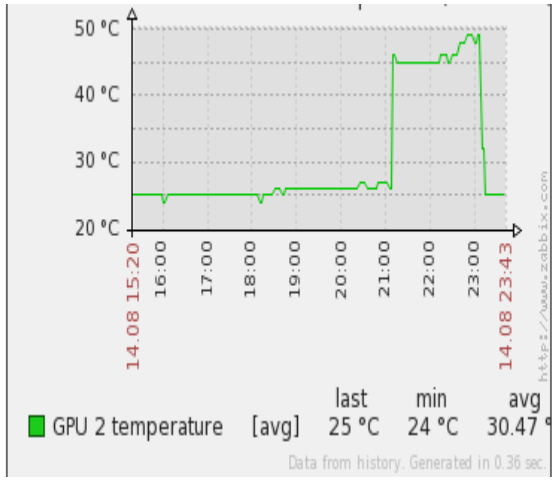
Timestamp	cluster.name.node1: active users (jobs)
2016-06-23 11:19:57	5264537,user1,20x1core 5272483,user2,1x1core 5272485,user3,1x1core 5272949,user4,2x1core

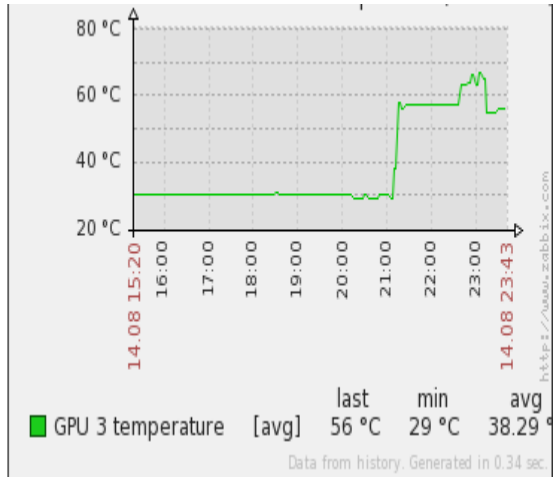


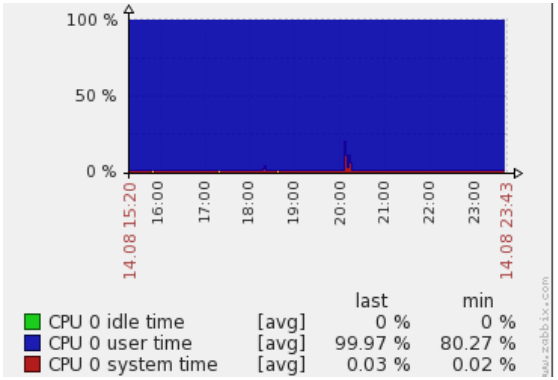
WHAT DO WE MONITOR: LOCAL METRICS

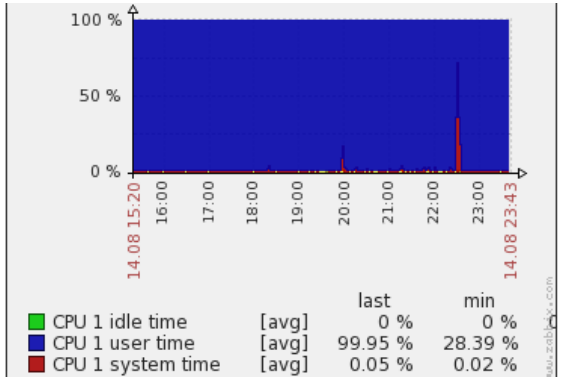


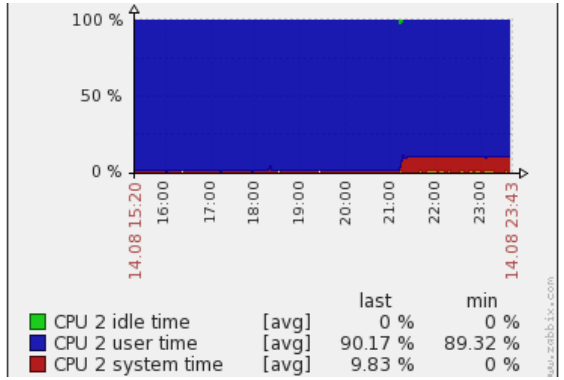


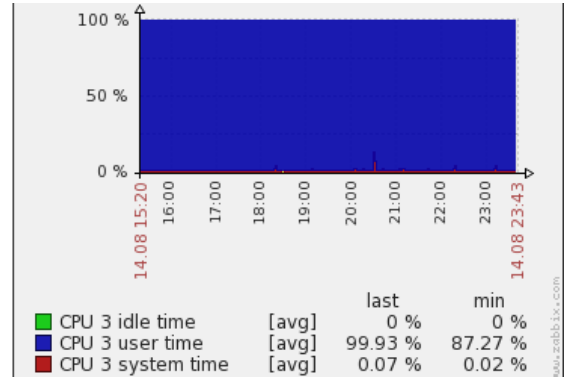








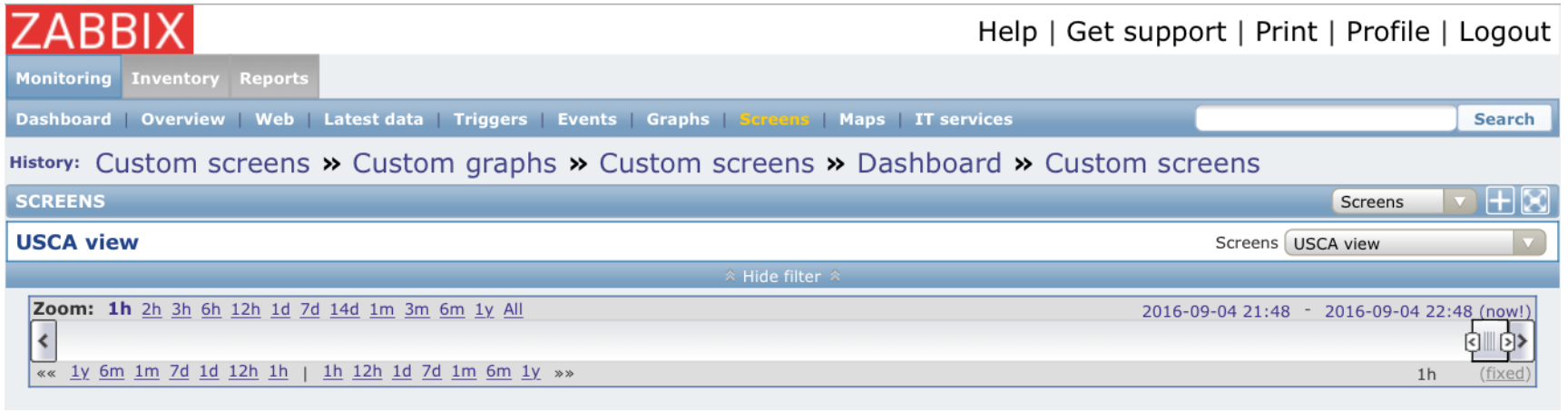




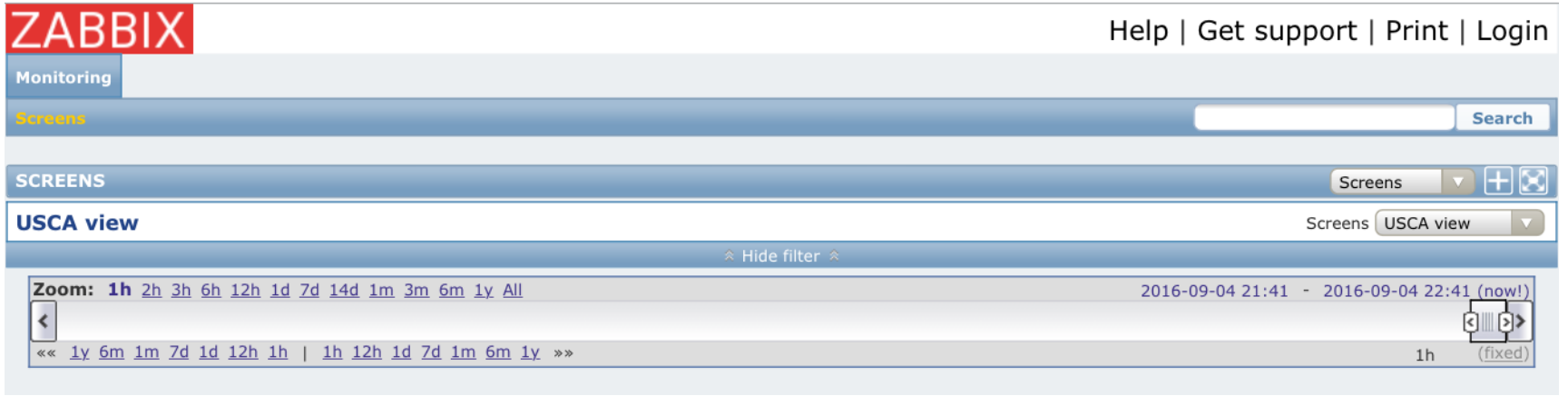
USER ACCESS

We want to provide a limited amount of information to users. They don't need any info about triggers and issues, but only metrics. We have patched Zabbix to remove all unnecessary data for guest access.

Before



After



Benefits

- Better understanding of a global issues on the cluster an reasons of why have they happened.
- Great performance indicators for other infrastructure teams (especially Storage team)
- Performance tuning of a scientific workflows. Jobs profiling. In some cases information we cat get from Zabbix is helping us to significantly improve performance of jobs.
- Proactive monitoring. With Zabbix it’s easier to understand if something is not right on the cluster or with some job. In most cases we are able to prevent global cluster issues, or at least minimize an impact.
- One monitoring system for clusters and HPC infrastructure.
- “All in one”. Lower efforts on support/maintain monitoring system(s).

WHAT'S NEXT?

- Tight integration with Grid HPC software.
- Data analysis using external tools, but with Zabbix data source.
- Create a set of CLI utilities for getting Zabbix statistics in 'human-readable' format.
- Automation of jobs profiling using Zabbix API.

Questions?