If you can't measure it, how do you know it works?

Matija Grabnar, NetTV Plus
matija.grabnar@nettvplus.org

Who are we?

- NeT TV Plus is a member of United Group (SBB, Telemach, etc).
- We stream live TV channels (from all ex-YU countries) over the internet to clients all over the world
- Hundreds of servers (encoders, streamers, databases, C&C, etc) in tens of data centers around the world
- Hundreds of gigabits of total traffic

Why do we measure?

- Performance monitoring
- Troubleshooting
- Maintenance prediction
- Optimization support
- Configuration management

Performance monitoring

net tv. plus

- Notice problems before users do: our complaints per user have fallen by more than 90%
- Is this thing on?
- Does the server ping?
- Does the application respond to it's port?
- What is the traffic going in/out?
- What is the CPU load?
- What is the memory usage?
- This app logs something about once a second if everything is OK: how old is the log?

Troubleshooting

- See what went wrong even if the server no longer works
- Locate the problem in the system
- It worked before, it doesn't now: what changed?
- Notify NOC and admins of problems before customers notice

Maintenance prediction / preventive maintenance

- Because admins don't like waking up to panic calls
- Looking at the weekly graph we can see this disk fills up about 10% every week (set up a file cleaning job, or buy a new disk?)
- Looking at this monthly graph we can see CPU usage is increasing about 3% every month (optimize the application or buy more servers?)
- Looking at this yearly graph we can see this leased line will start peaking over capacity in three months.

Optimization support

- Testing environments are well and good, but there is nothing like the real world
- Knowing how the application behaves in real world shows if the optimization was the right one
- When hardware isn't performing as it should, find out what the problem is

Configuration management

- If any system setting is changed from default, it has to be monitored
- This includes maxopenfiles, sysctl settings, buffer settings for interfaces, etc
- This really helps when vendors supply a “preconfigured” server at a remote location.

What do we measure? (1)

- cpu (both total and per-core),
- memory allocation,
- disk usage, utilization and latency,
- network throughput (for each network interface separately),
- the value of each sysctl variable that we changed from the default,
- number of open/close-wait sockets for each relevant port (for each network interface separately)
- CPU/memory/io usage/gc time/number of context switches/number of processes & threads for each of the relevant applications,
- response time for each DNS listed in /etc/resolv.conf

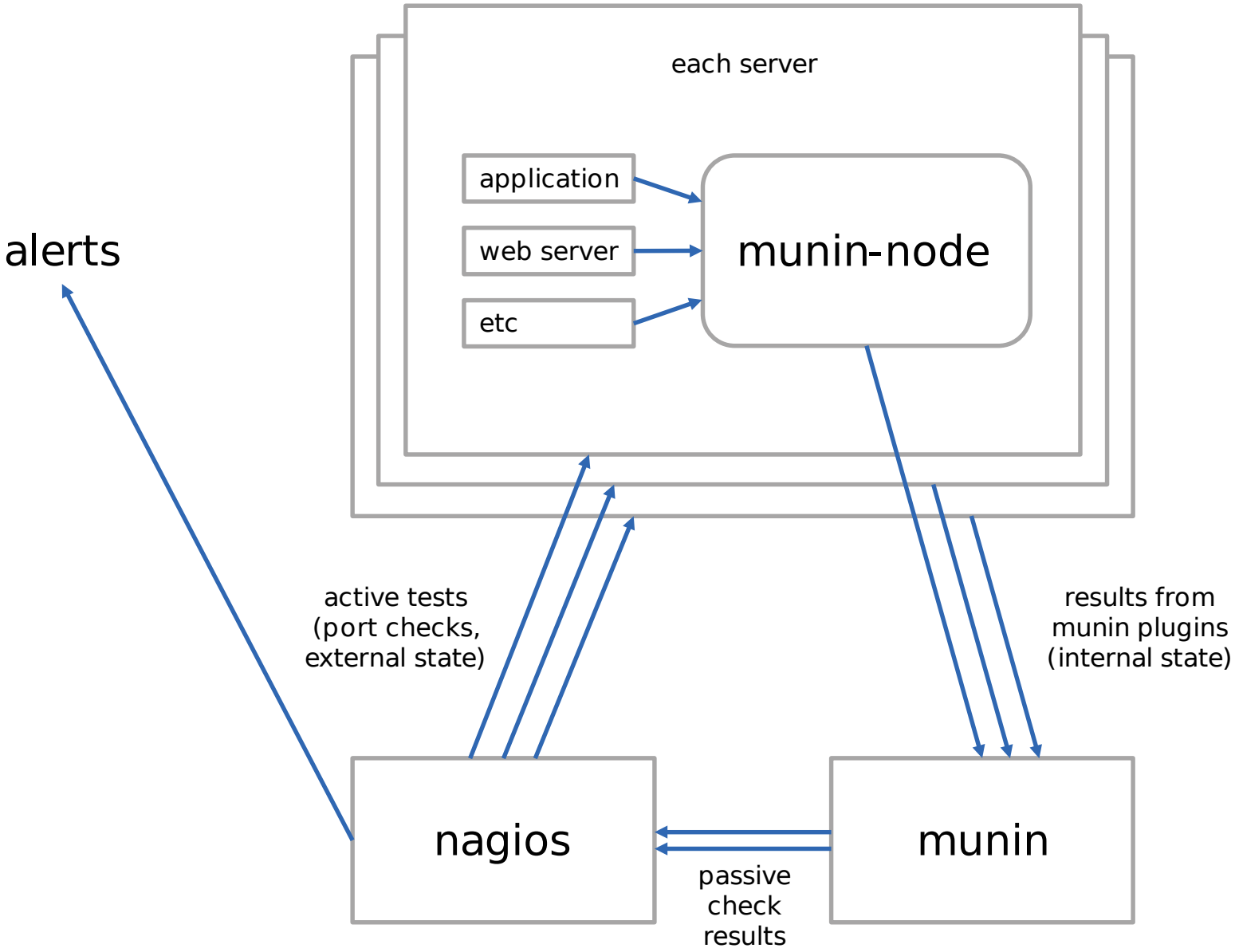
What do we measure? (2)

- age of logs for several applications that write frequent logs (old log means something went wrong with the application, and triggers an alert)
- temperatures for the CPU/motherboard and the disks
- number of dmesg entries
- In a separate hierarchy we are monitoring each channel we stream, everything from the number of errors on the input, to width and height, cpu usage for that stream, etc. (each channel presented as one hostname)
- We have a special munin value which changes between warning and critical every 30 minutes, and a custom nagios plugin that checks state on Nagios to see if the value has changed recently. If it has not, that means the communication between munin and nagios has broken down, and an alert is triggered.
- We do NOT have an automated process that would detect if nagios process is down, but we do have a NOC which checks Nagios regularly.

How do we measure it?

- Currently, a mixture of munin and nagios
- Nagios does notifications, but is also useful for centralized testing of ports
- Munin runs on each monitored server / very easy to write plugins in any language

How do we measure it



Where do we measure it?

- On every host we run. Every encoder, streamer, catchup-server, database, web server or just random virtual-host-of-the-week has munin installed on it, and is sending data to be graphed on central server.
- We have a script which parses munin's list of hosts, and uses a template to automatically add Nagios tests for each host that appears in munin.

Writing a munin plugin is very simple

Minimal Config (output from plugin):

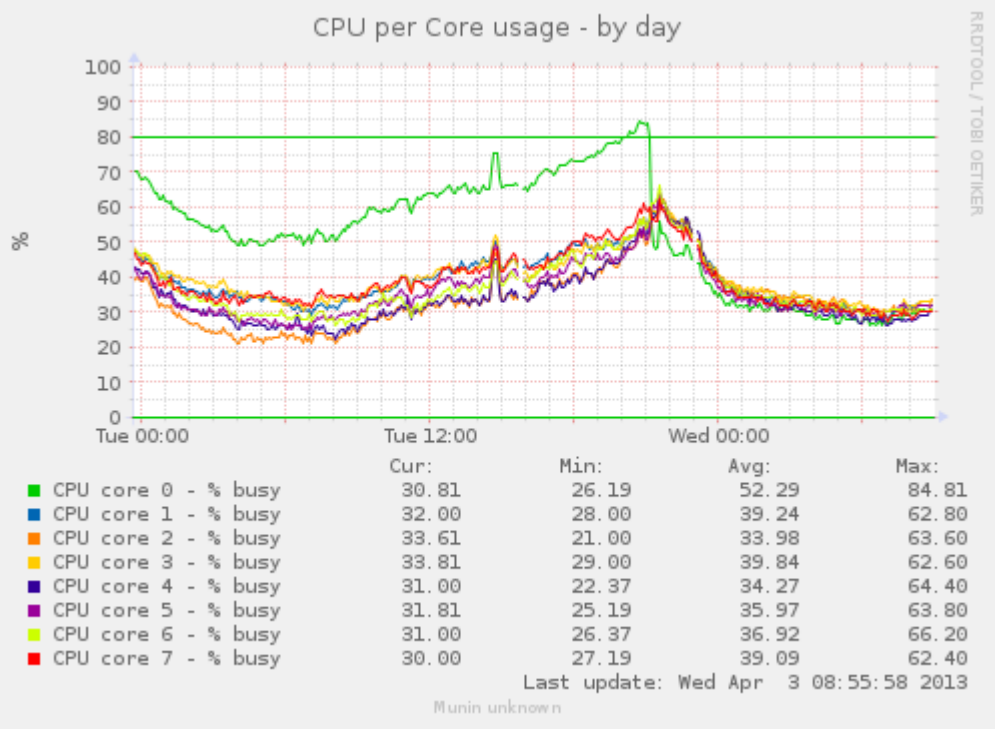
- graph_title some text
- Name.label Label of the value name

Values (output from plugin):

- Name.value 123.3

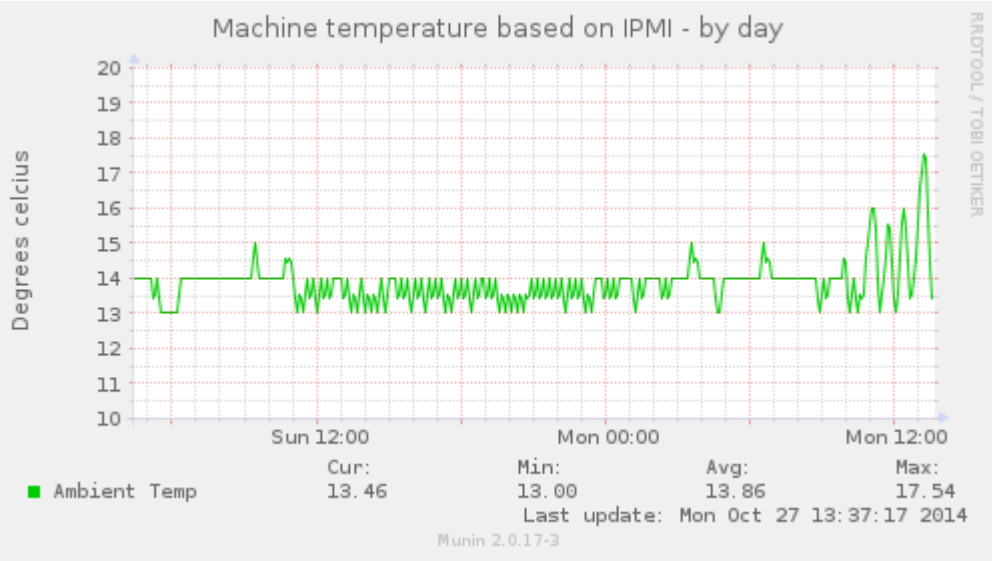
Examples of benefits 1

Why you should monitor each core separately, not just CPU as a whole.



Examples of benefits 2

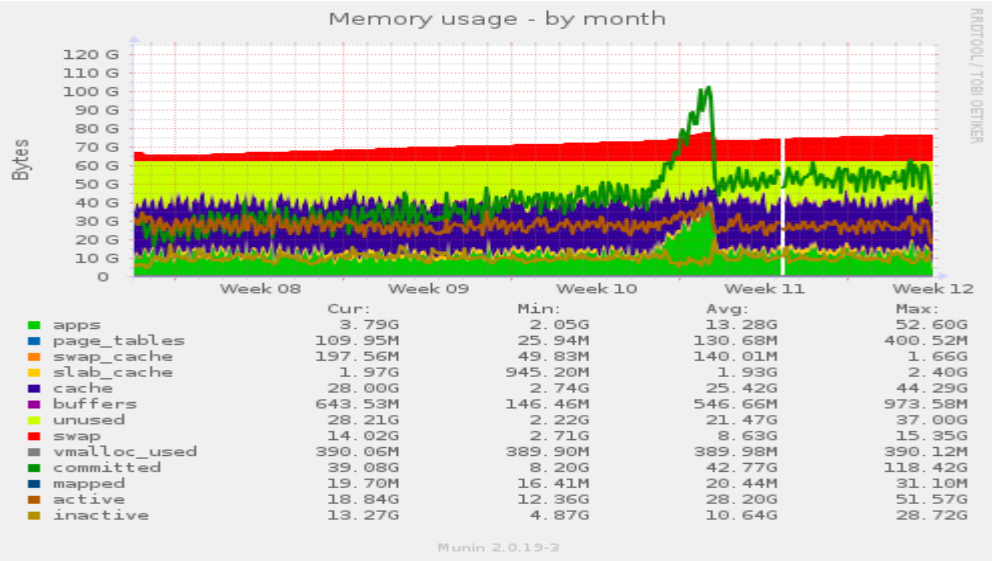
Problem with cooling in a particular DC



Examples of benefits 3

What time is it? It's a quarter to buy-more-RAM!

(or half past fix-the-memory-leak)



Optimizations (problems you run into when monitoring many hosts)

- graph_strategy cgi
- html_strategy cgi
- Max processes
- Move the png directory to ramdisk
- Munin-async
- Modify munin source to use compression on ssh
- rrdcached
- Add RAM first, CPU second, rrdcached will make a bigger difference than SSD

Communication with Nagios

- contact.nagios.command /usr/bin/send_nsca
- Adjust nagios configuration to accept all munin parameters as passive test results
- Be careful, if there are too many variables, the pipe to nagios can get overwhelmed

Conclusion

net tv. plus

- Monitor everything, make sure everybody involved in production and management has access to measurements
- Clear out warnings and errors early and often
- Escalate warnings and errors after a set time, even to upper management

Future

- We've outgrown munin/nagios and are now developing an in-house solution
- No, it's not open source, at least not yet

Questions?

matija.grabnar@nettvplus.org