Hardware monitoring with collectd

Luca Gardi - luca.gardi@cern.ch

Introduction

- explain the differences between Lemon and collectd
- summarize needed changes for hardware monitoring
- explain the choices made during the process
- provide a status update
- explain current issues and proposed fixes

1 - Lemon and collectd

Lemon

- **developed by CERN**
- in production since 2006 (at least)
- old monitoring infrastructure has been replaced
- retirement efforts started mid-2017

collectd

- **open source project**
- collects system and service metrics
- optimized to handle thousands of metrics
- modular and portable with community plugins
- **easy to develop new plugins** in Python/Java/C/Perl
- continuously improving and well documented

2 - Why collectd?

Pros:

- community-driven and rich ecosystem
- **alarms and plugins definitions are puppet-based**
- better reusability, documentation
- easier to set up for quick metric collection
- easier metric dispatch in plugins

Cons:

- alarms generated on transition
- **existing plugins require re-writing**
- MONIT provides a lemon-sensor wrapper but is deprecated

3 - HW monitoring in the Lemon era

Agent sensors:

- **lemon-sensor-smart:** SMART logs monitoring
- **lemon-sensor-tw:** 3ware RAID controllers
- **lemon-sensor-megaraidsas:** LSI MegaRAID controllers
- **lemon-sensor-adaptec:** Adaptec RAID controllers
- **lemon-sensor-sasarray:** JBODs monitoring
- **lemon-sensor-blockdevice-drives:** log parser for SCSI errors
- **lemon-sensor-ipmi:** IPMI monitoring

On-behalf monitoring (centralized):

- **pdu-xmas:** centralized out-of-band PDU monitoring (SNMPv2)

4 - Moving towards a collectd era

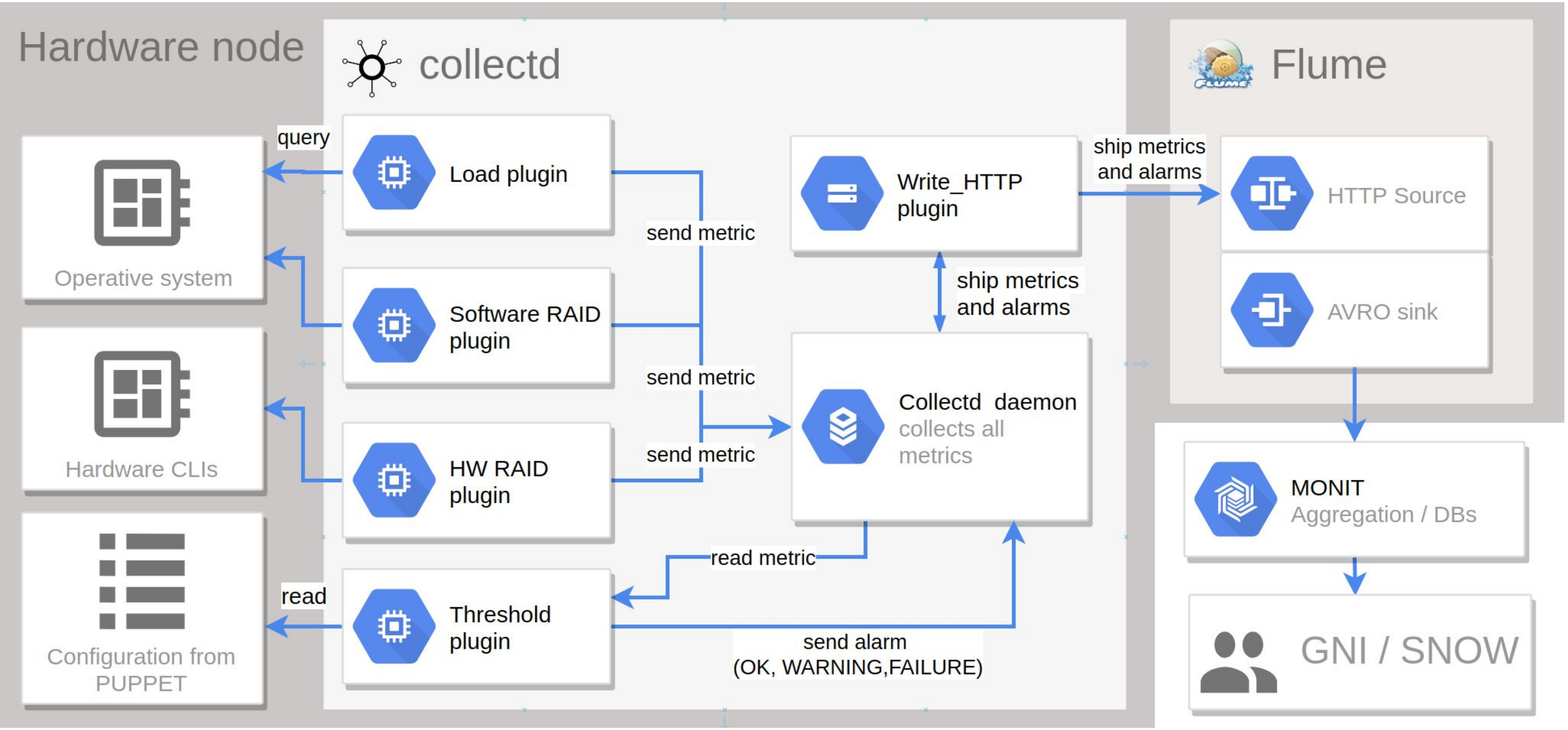
- **very specific and complex needs**
  - heterogeneity of hardware and configurations
  - hardware RAID controllers
  - intense use of IPMI

- **no community sensors** we could adopt
- good news! **code can be ported from lemon sensors**
- adopt TDD (Test-Driven Development)
- compatibility with python 2.4, 2.7, 3.4
- Continuous Development (CI/CD) using GitLab

CF

5 - Plugin architecture

CERN IT Department



6 - The big migration

Collectd plugins:

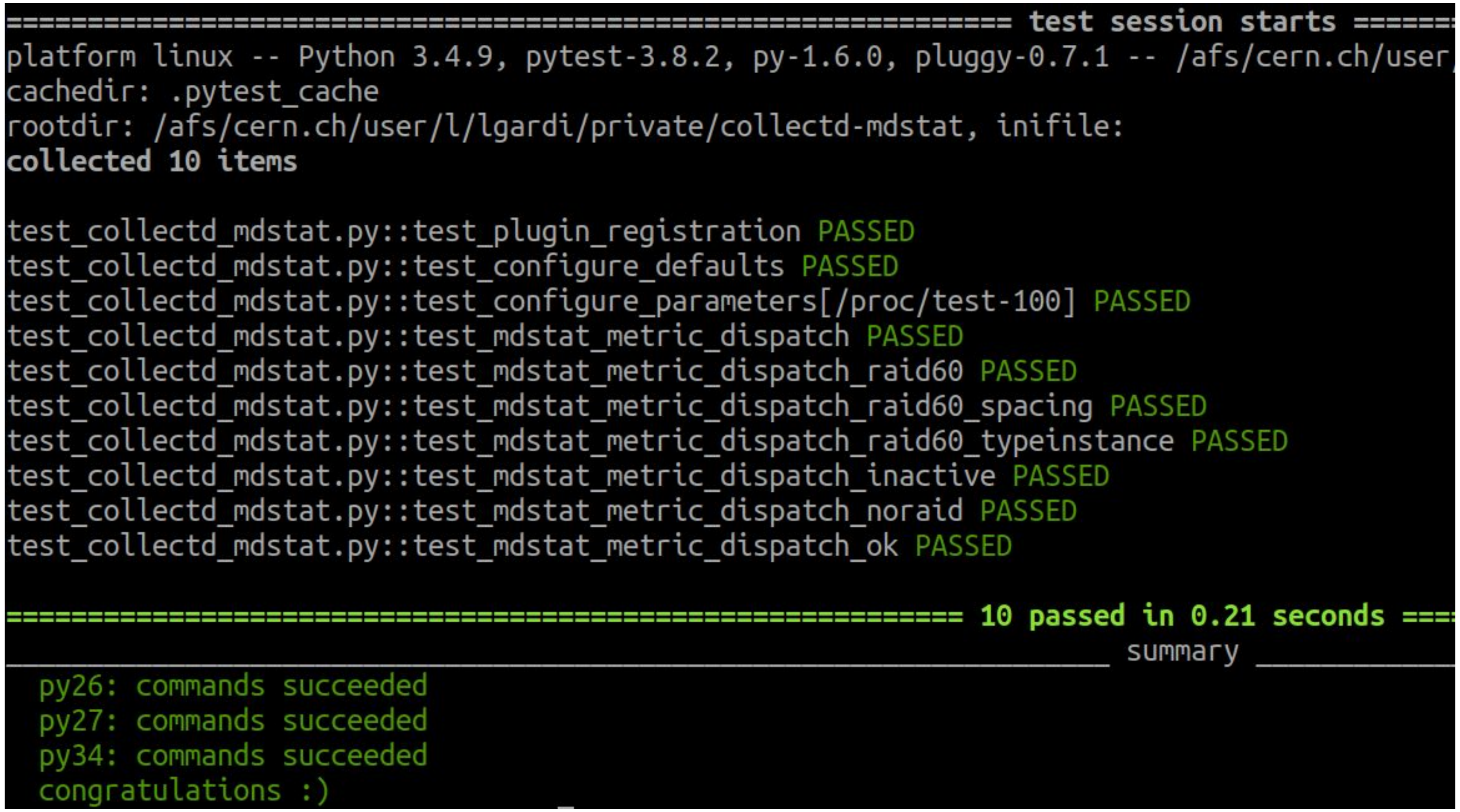
- **collectd-mdstat:** in production (new) ✔
- **collectd-smart-tests:** in production ✔
- **collectd-megaraidsas:** in QA ✔
- **collectd-sasarray:** in development
- **collectd-blockdevices:** in pipeline
- **collectd-adaptec:** in pipeline
- **mcelog:** from the community

Centralized monitoring:

- **CINNAMON:** in production ✔ (requires minor changes)
- **PODIUM:** in development

7 - Plugin development workflow

- identify output metrics and write the tests
- write the plugin
- if tests.color == **green**: plugin.puppet_deploy()



```
============================================================= test session starts ======
platform linux -- Python 3.4.9, pytest-3.8.2, py-1.6.0, pluggy-0.7.1 -- /afs/cern.ch/user
cachedir: .pytest_cache
rootdir: /afs/cern.ch/user/l/lgardi/private/collectd-mdstat, inifile:
collected 10 items

test_collectd_mdstat.py::test_plugin_registration PASSED
test_collectd_mdstat.py::test_configure_defaults PASSED
test_collectd_mdstat.py::test_configure_parameters[/proc/test-100] PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch_raid60 PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch_raid60_spacing PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch_raid60_typeinstance PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch_inactive PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch_noraid PASSED
test_collectd_mdstat.py::test_mdstat_metric_dispatch_ok PASSED

=========================================================== 10 passed in 0.21 seconds ===
____________________________________________________________________ summary __________
  py26: commands succeeded
  py27: commands succeeded
  py34: commands succeeded
  congratulations :)
```

8 - Plugin deployment workflow

- RPM packaging and repositories using Koji
- **Collectd plugin definition on Puppet**
    - it-puppet-module-cerncollectd_contrib on GitLab
- standard CERN CRM QA -> Production pipeline (1 week)
- **deployed on physical machines**
    - it-puppet-module-hardware: physical.pp

9 - Alarms

- **based on standard collectd Threshold plugin**
- checks local metrics against defined thresholds
- states: OK, WARNING, FAILURE
- **puppet defined** (metricmgr is already read-only)
- Service Managers can override thresholds and SNOW targets

10 - What’s left and current issues

- finish porting of the sensors to collectd
- start retirement of old lemon sensors
- too many tickets:
  - fine tuning of the alarms is necessary
  - waiting for better SNOW tickets deduplication
- tickets are not very descriptive:
  - a pull request has been sent to the upstream community
- no lemon-host-check
  - do we need it?
  - is collectdctl enough?

11 - Conclusions

- collectd provides a **mature environment** for HW monitoring
- using **puppet for alarms definition** is definitely a plus for versioning and maintenance, compared to metricmgr
- after an initial series of delays, mainly due to our early adoption, we are now **more than half-way** there and progressing steadily
- targeting end of the year for finishing the migration
- a good occasion for **collaboration with IT-CM-MM**

CF

Hardware monitoring with collectd

```
                       _   _               ___
                      | | (_)             |__ \
  __ _ _   _  ___  ___| |_ _  ___  _ __  ___ ) |
 / _` | | | |/ _ \/ __| __| |/ _ \| '_ \/ __|/ /
| (_| | |_| |  __/\__ \ |_| | (_) | | | \__ \_|
 \__, |\__,_|\___||___/\__|_|\___/|_| |_|___(_)
    | |
    |_|
```

Backup slides - Plugin definition

```
class cerncollectd_contrib::plugin::mdstat (
  Integer       $interval,
  String        $mdstat_path,
) {

  require ::cerncollectd_contrib

  package { 'collectd-mdstat':
    ensure => present,
  }

  collectd::plugin::python::module { 'collectd_mdstat':
    ensure  => present,
    config  => [{
      'INTERVAL'    => $interval,
      'MDSTAT_PATH' => $mdstat_path,
    }],
    require => Package['collectd-mdstat'],
  }
}
```

Backup slides - Alarm definition

```
class cerncollectd_contrib::alarm::mdstat_wrong (
  Integer $failure_max,
  Integer $hits,
  Boolean $persist,
  Boolean $interesting,
  Optional[Hash] $custom_targets,
  Optional[String] $actuator,
) {
  ::cerncollectd::alarms::threshold::plugin {'mdstat_wrong':
    plugin      => 'mdstat',
    type        => 'disk_error',
    failure_max => $failure_max,
    hits        => $hits,
    persist     => $persist,
    interesting => $interesting,
  }

  ::cerncollectd::alarms::extra {'mdstat_wrong':
    ctd_namespace => 'mdstat',
    targets       => $custom_targets,
    actuator      => $actuator,
  }
}
```

Backup slides - Plugin deployment

```
if (versioncmp($::operatingsystemmajrelease,'6') >= 0) or (versioncmp($::operatingsystemmajrelease,'7') >= 0){

  # Software RAID failures (see target in YAML file data)
  include ::cerncollectd_contrib::alarm::mdstat_wrong
  include ::cerncollectd_contrib::plugin::mdstat

  # SMART attributes failures
  include ::cerncollectd_contrib::alarm::smart_wrong
  include ::cerncollectd_contrib::plugin::smart_tests

  # MegaRAID failures
  include ::cerncollectd_contrib::alarm::megaraidsas::bbu_status_wrong
  include ::cerncollectd_contrib::alarm::megaraidsas::controller_status_wrong
  include ::cerncollectd_contrib::alarm::megaraidsas::controller_correctable_errors
  include ::cerncollectd_contrib::alarm::megaraidsas::controller_uncorrectable_errors
  include ::cerncollectd_contrib::alarm::megaraidsas::cache_policy_on_faulty_bbu_wrong
  include ::cerncollectd_contrib::alarm::megaraidsas::cache_policy_on_raid_array_wrong
  include ::cerncollectd_contrib::alarm::megaraidsas::raid_array_status_wrong
  include ::cerncollectd_contrib::alarm::megaraidsas::missing_drives
  include ::cerncollectd_contrib::alarm::megaraidsas::unconfigured_good_drives
  include ::cerncollectd_contrib::alarm::megaraidsas::unconfigured_bad_drives
  include ::cerncollectd_contrib::alarm::megaraidsas::offline_drives

  if (versioncmp($::operatingsystemmajrelease,'6') >= 0){
    class {'::cerncollectd_contrib::plugin::megaraidsas':
      lsmod_path => '/sbin/lsmod',
    }
  } else {
    include ::cerncollectd_contrib::plugin::megaraidsas
  }
}
```

Backup slides - collectdctl output

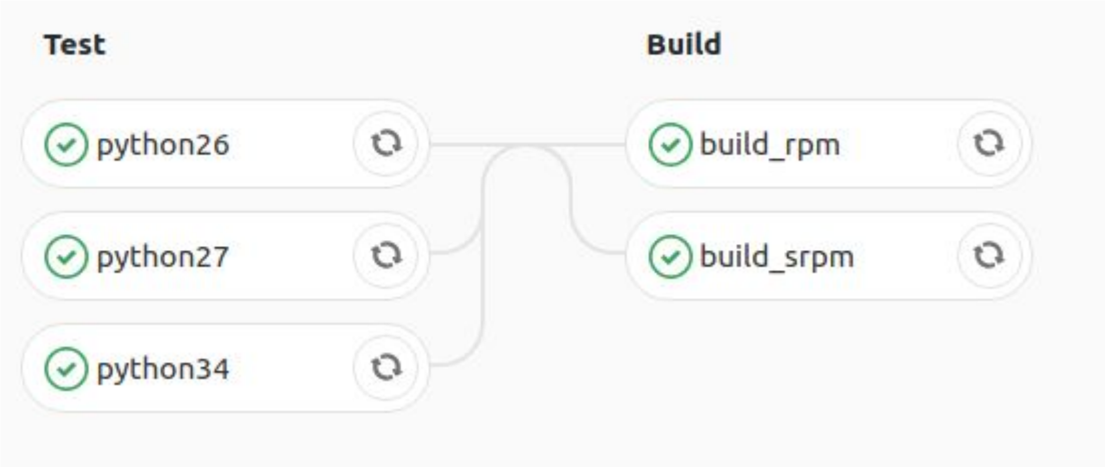
- **collectd namespace:**
  - <hostname>/<plugin>-<plugin_instance>/<type>-<type_instance>
- **listing values:**

  -
    ```
    [root@lxfsrd08c04 ~]# collectdctl listval
    lxfsrd08c04.cern.ch/megaraidsas-bbu_status/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-controller_cache_policy_on_faulty_bbu/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-controller_cache_policy_wrong_on_raid_array/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-controller_memory_correctable_errors/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-controller_memory_uncorrectable_errors/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-controller_status/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-missing_drives/count
    lxfsrd08c04.cern.ch/megaraidsas-offline_drives/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-raid_array_status/count-c0_vd0
    lxfsrd08c04.cern.ch/megaraidsas-raid_array_status/count-c0_vd1
    lxfsrd08c04.cern.ch/megaraidsas-unconfigured_bad_drives/count-c0
    lxfsrd08c04.cern.ch/megaraidsas-unconfigured_good_drives/count-c0
    ```
- **getting values:**

  -
    ```
    [root@lxfsrd08c04 ~]# collectdctl getval lxfsrd08c04.cern.ch/megaraidsas-bbu_status/count-c0

    value=0.000000e+00
    ```

Backup slides - GitLab CI/CD


Test
python26
python27
python34
Build
build_rpm
build_srpm