
CERN

CERN IT
HTCondor Pool Monitoring

Joint HTCondor Stakeholders Meeting (12th December 2018)

Luis Fernandez Alvarez <luis.fernandez.alvarez@cern.ch>

INTRO / INFRA / NODES / SERVICE / AUTOMATION

CERN IT / Batch Service

- Currently running two production HTCondor pools:
  - Shared pool (195K cores)
  - T0 pool (30K cores)
- Control plane mainly CentOS7
  - 2 HA Central Managers per pool
  - 25 Schedds (shared), 3 schedds (T0)
- Worker nodes running SLC6 (95%) & CC7 (5%)

Monitoring targets

Service

- *Is HTCondor running smoothly?*
- Startds, schedds, central managers…

Nodes

- *Are the nodes in the pool healthy?*
- Virtual machines & physical nodes

Infrastructure

- *Is the underlying infra working fine?*
- Mainly OpenStack

Monitoring ecosystem

Provided by CERN MONIT

- Collectd
- Flume
- Kafka
- InfluxDB
- ElasticSearch
- Grafana / Kibana

Run internally by the Batch Service

- Fifemon
- Graphite
- Condor Probes
- Filebeat / Logstash
- Prototypes

INTRO / **INFRA** / NODES /
SERVICE / AUTOMATION

Infrastructure monitoring

- Worker nodes run on OpenStack, critical component in our pools
- Virtual machines scattered across **50 projects** with rather homogeneous setup
  - VM provisioning is automated via a nightly job
- Some configuration bits adding noise:
  - Neutron vs nova-network projects
  - Software bugs, mainly: kernel, qemu-kvm, libvirtd
- Cloud operations impacting VM availability:
  - Replace hard disk, motherboards, opportunistic resources, etc.

Infrastructure monitoring (II)

- Questions to answer at this stage:
  - *Are production machines going to error state?*
  - *Is our automated provisioner successfully creating new nodes?*
  - *Is there any cloud intervention affecting our resources? When and how?*
- Actions required on both sides

Infrastructure monitoring (III)

- OpenStack metrics collected by Cloud team and stored in CERN MONIT infra:
  - VMs per project and their states
  - Consumed by us via Grafana
  - Alarms defined in Grafana to spot errors
- Cloud team publishes interventions in a programmatic way via a message bus:
  - We listen and take actions automatically
  - Automation in progress

Internal / Infrastructure - OpenStack Overview

Previous week | Refresh every 30m

project_name: All | Bin: 1m

Active	ERROR	Building	Stopped	Deleted	Rescued
23581	9	8	2	N/A	N/A

VMs in Active state

Metric	Value
IT-Batch - Wigner Project 010 - Dedicated	1.05 K
IT-Batch - Wigner Project 001 - Dedicated	1.03 K
IT-Batch - Wigner Project 015 - Dedicated	990.00
IT-Batch - Wigner Project 013 - Dedicated	956.00
IT-Batch - Wigner Project 011 - Dedicated	935.00

VMs in ERROR state

Metric	Value
IT-Batch - Geneva Project 014 - Dedicated	8.00
IT-Batch - Geneva Project 032 - Dedicated	1.00

VMs in Build state

Metric	Value
IT-Batch - Geneva Project 038 - Opportunistic	6.00
IT-Batch - Geneva Project 014 - Dedicated	1.00
IT-Batch - Geneva Project 013 - Dedicated	1.00

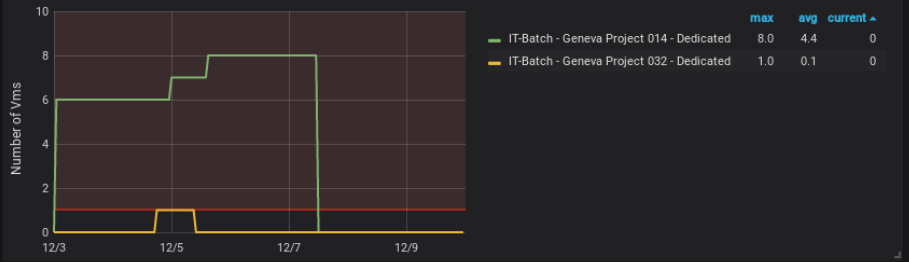
VMs in Stopped state

Metric	Value
IT-Batch - Geneva Project 013 - Dedicated	1.00
IT-Batch - Geneva Project 007 - Dedicated	1.00

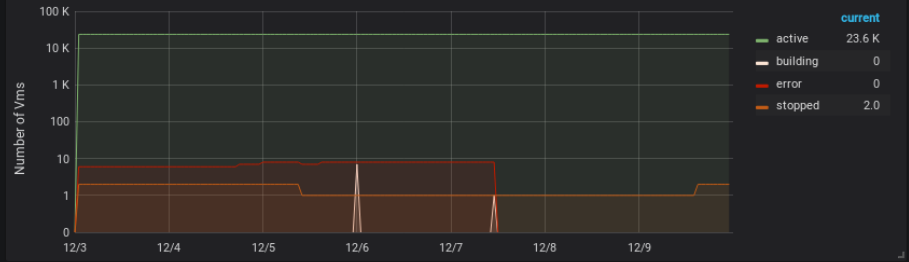
VMs in Rescued/Deleted state

No data to show

Batch VMs in ERROR state



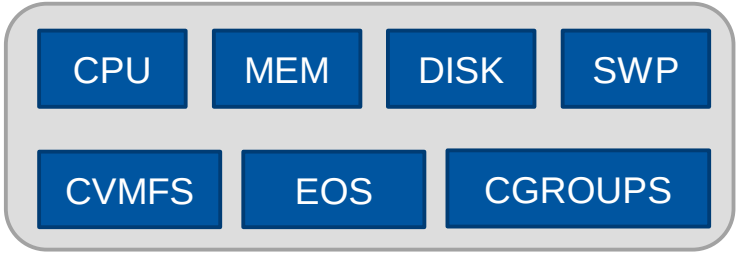
Batch VMs



INTRO / INFRA / **NODES** /
SERVICE / AUTOMATION

Node monitoring

- This layer covers the internal resources of every node in the infrastructure:



- What we want to spot here:
  - Storage not accessible, nodes running out of space, resource usage, known bugs.
  - Impact of new config changes, software releases, etc.

Node monitoring

- Base monitoring provided by CERN IT:
  - Collectd + Apache Flume for common node resources: cpu, load, disk, heartbeat, etc. (InfluxDB)
  - Rsyslog + Apache Flume. Shipped to ElasticSearch
- Extended by us:
  - CVMFS plugin for collectd: https://github.com/cvmfs/collectd-cvmfs
  - Log based metrics (collectd tail) for EOS logs (WIP)
  - Process monitoring: nscd, mounts,…
  - Not collectd based (yet): cgroup monitoring

Node monitoring: cgroups

- Prototype: CGroups Simple (CGS)
- Fine-grained cgroup metrics to be exposed to users in the future
- Photographers, transformers & writers
- Standalone, evaluating transforming it into collectd plugin

Node monitoring: heartbeat

- Simple but interesting metric: collectd plugin + external processor
- Given its design and our scale it's a good indicator of how smoothly things are running

Node monitoring: alarms

- Alarms:
  - Collectd: threshold plugin (local to the machine)
  - Grafana alerts for aggregated metrics
- Actuators for collectd alarms:
  - Based on MONIT collectd plugin.
  - Space cleanup, fix known issues, etc.


Internal / HTCondor - Worker Details
Dec 11, 2018 09:32:28 to a few seconds ago
Refresh every 30m
Node
Hostgroup
Environment
Datacentre
AppState
Links
b64707d887.cern.ch
bi/condor/gridworker/shareshort
batchtest
wigner
production
Foreman, LanDB
Server Metrics
Puppet events
Load
CPU usage
Disk Usage
Alarm Status
df_pool_percent_bytes_used_value:
OK
filecount_yumtransactions_files_value:
OK
no_contact:
OK
processcount_nscd_gauge_value:
OK
swap_swap_io_in_value:
OK
tail_base_count_vm_kill_value:
OK
Memory operations
Memory usage
Processes state
Network usage
Worker Blocks (AFS, EOS, CVMFS,...)
Aggregated Memory Usage
Mount Time (percentile 95)
Aggregated Used File Descriptors
CVMFS Loaded Catalogs
CVMFS Cache Cleanups (24h)
Memory Usage (VMS)
Memory Usage (RSS)
CVMFS Mount Time (~ls time)
CVMFS Used File Descriptors
CVMFS I/O Errors

INTRO / INFRA / NODES /
SERVICE / AUTOMATION

HTCondor monitoring: global

- At this level, we want to make sure that:
  - Condor is healthy, users can submit jobs, we're not hitting scaling issues,…
- Based on: http://fifemon.github.io/
- Graphite backend
  - Metrics namespaced per pool (shared and T0)
  - 90Gb of data after 2 years
- Fifemon probes running every 4 minutes
  - Extended with CERN pool specific metrics

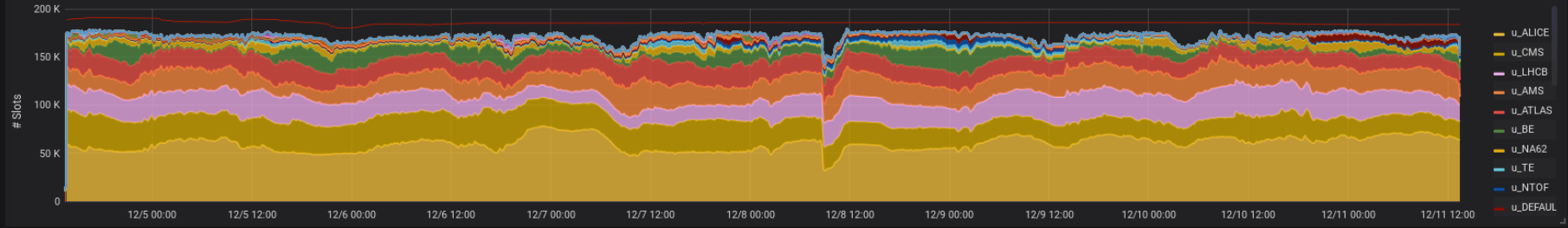
C5 Summary

Last 7 days

cluster cernprod

Average Number of Jobs Running Concurrently	Total Job Runtime (Last Week)	Average Efficiency (Last week)
132669	1155 years	74%

Running Slots By Experiment



CPU Delta (7 days)	Memory Delta (7 days)	Disk Delta (7 days)
-2364	-14 TiB	-98 TiB

Efficiencies by DataCentre (Last Week)

Metric	Min	Max	Avg	Current
wigner	52.15%	76.28%	67.22%	69.93%
meyrin	65.07%	88.72%	79.32%	86.99%

Efficiencies by Experiment/DataCentre

Metric	Min	Max	Avg	Current

HTCondor monitoring: pieces

- Central managers:
  - Negotiation cycles, collector stats…
- Schedd:
  - Transfers & DutyCycle
  - Basic Grafana alerts for DaemonDutyCycle
- Startd Healthchecks:
  - Startd crons: cvmfs, afs, pool space and other known issues.
  - Start expression depends on node health.
  - Considering linking healthchecks to collectd metrics / alarms.


HTCondor - Central Managers
Last 2 days
Refresh every 5m
Load
longterm
midterm
shortterm
Root FS Us...
6%
Memory usage
buffered
cached
free
slab_recl
slab_unreci
used
CPU usage
idle
interrupt
nice
softirq
steal
system
user
wait
Memory operations
in
out
Network usage
rx
tx
Collectors (CERN_Condor_Share-tweetybird03_cern_ch)
DaemonCore Duty Cycle
0.954
UpdatesLostRatio
0
UpdatesLostMax
0
Fork Queries
DEFRAG
NEGOTIATOR
SCHEDD
STARTER
SUBMIT
TOOL
ActiveQueryWorkers
Active Query Workers
Peak of Active Query Workers
GAHP
JOB_ROUTER
SCHEDD
STARTD
SUBMIT
TOOL
UNKNOWN
HTCondor - Alarms
Last 12 hours
Refresh every 1m
[Alarm] DaemonCore DutyCycle
ALERTING
2018-10-18 02:29:08
bigbird15_cern_ch=0.9098923210363301

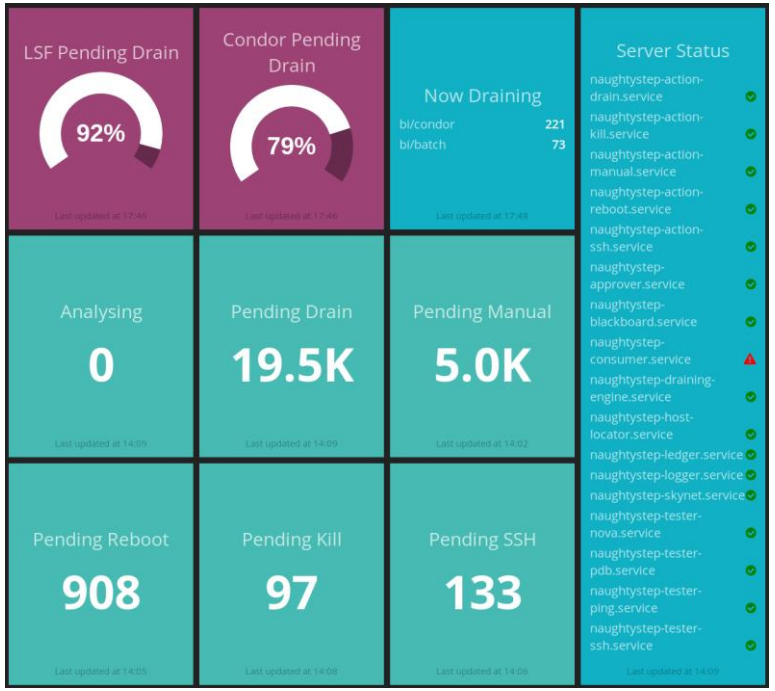
HTCondor monitoring: ongoing

- Started sending EventLog to ElasticSearch
  - Based as well on Kevin's configuration
  - Local filebeat prospector
  - Internal logstash instance for indexing running on Kubernetes
  - Shipped to MONIT infrastructure logs endpoint
- Future:
  - Consume from MONIT Kafka topic and produce new derived metrics
  - Combine with cgroups and provide a full picture to users about their jobs.

INTRO / INFRA / NODES /
SERVICE / **AUTOMATION**

Automation

- We don't want to waste our time babysitting 20K nodes…
- Working on prototypes to automate operations:
  - Draining actions (based on health checks, cloud interventions)
  - Self-healing infrastructure: take decisions based on alarms.
  - Preventive operations: continuously drain 10% to deploy new kernel in the worker nodes.
- Keeping an eye on existing tools and technologies:
  - consul, kafka, spark, etcd, kraken, vitrage, alertmanager
- Open to ideas and projects… ☺




CERN