Automation & Monitoring in the CERN Condor pool

Luis Fernández Álvarez <luis.fernandez.alvarez@cern.ch>
The Batch Service
The Batch Service @ CERN IT

*Provides Tier-0 compute power via HTCondor to WLCG.*

- Process CPU intensive workload ensuring fairshare among various user groups.
- Maximize utilization, throughput, efficiency.
- It runs jobs from the Grid and from local CERN departments.

25/09/2019

https://indico.cern.ch/event/817927/
235K Cores

20K Virtual Machines

47 OpenStack Projects

1M Completed / Day

2 Pools

2 CMs / Pool

28 Schedds

3 DCs
Automation
Automation: why?

- Reduce the time doing tedious ops tasks.
- Avoid hidden magic scripts / procedures.
- Knowledge sharing.

![Automation Tools Diagram]
Automation: provisioning (I)

- **Goal:** keep our 47 OpenStack projects always full.
- **Before:** nightly script launched via Rundeck
  - Homegrown, difficult to maintain.
  - Navigate through Rundeck logs to guess state.
  - Not dealing with machines in ERROR state.
- **Solution:** Terraform + GitLabCI + Mattermost
Automation: provisioning (II)

- The pipeline:

<table>
<thead>
<tr>
<th>Plan</th>
<th>Authorise</th>
<th>Provision</th>
<th>Cleanup</th>
</tr>
</thead>
</table>
| Artifact | Don’t destroy!  
Don’t recreate > 10%!  
Notify Operators | 6 hours delay | Invalidate names |
Automation: provisioning (III)
Automation: draining (I)

- Different impact:
  - Low: broken hypervisors
  - High: Spectre/Meltdown, Wigner decommissioning, CentOS7 migration,…

- Simple workflow: stop accepting jobs, wait for existing jobs to finish and reboot if needed.
Automation: draining (II)

- Job limit + Target date combination for different scenarios
- Interventions: reboot, fix, upgrade, hardware...

25/09/2019
https://indico.cern.ch/event/817927/
Automation: draining (III)

- Different homegrown services involved in the implementation:
  - **roger**: state
  - **megabus**: message bus
  - **brainslug**: node agent

- **Payload:**
  ```json
  {
    "Reason": "WignerDeco",
    "Reference": "BBC-2445",
    "TargetDate": 1569218400,
    "JobDate": 1569218400,
    "Reboot": "None"
  }
  ```
Monitoring
Infrastructure, Nodes & Service
Monitoring: why?

• Have a better understanding of how our system works.
• Improve the quality of the service.
• Identify the misuse of the pool.
• …
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

25/09/2019
https://indico.cern.ch/event/817927/
Monitoring - Ecosystem

CERN Monit Team

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

Batch Service

- Fifemon
- Graphite
- Filebeat
- Logstash
- Misc. Prototypes
Monitoring: infrastructure (I)

- Worker nodes run on OpenStack, critical component in our pools
- Virtual machines scattered across 47 projects and 3 datacenters.
- Main areas 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.
Monitoring: infrastructure (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
Monitoring: infrastructure (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:
  - WIP: listen and take actions automatically
Monitoring: nodes

- What to worry about in this layer?
  - Storage not accessible, nodes running out of space, resource usage
  - Impact of: config changes or sw releases.
Monitoring: nodes

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
Monitoring: nodes (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
Monitoring: nodes (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
Monitoring: nodes (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.
Monitoring: service

- At this level, we want to make sure that:
  - Condor is healthy, users can submit jobs, we’re not hitting scalability 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
  - TODO: include the CEs into the probes.
Monitoring: service (events)

- EventLog sent to ElasticSearch
  - Based on Kevin Retzke work
  - Local filebeat prospector & Logstash instance
- Rate ~15K events / second
- Use of EVENT_LOG_JOB_AD_INFORMATION_ATTRS
- Single worker
  - Revisit the usage of “aggregate” plugin
  - Increase number of workers and instances
Average Number of Jobs Running Concurrently: 132669
Total Job Runtime (Last Week): 1155 years
Average Efficiency (Last Week): 74%

Running Slots By Experiment

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

Efficiencies by DataCentre (Last Week):

<table>
<thead>
<tr>
<th>DataCentre</th>
<th>Min</th>
<th>Max</th>
<th>Avg</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>wignera</td>
<td>52.15%</td>
<td>76.28%</td>
<td>67.22%</td>
<td>69.93%</td>
</tr>
<tr>
<td>moyrus</td>
<td>65.07%</td>
<td>88.72%</td>
<td>79.32%</td>
<td>86.99%</td>
</tr>
</tbody>
</table>

Efficiencies by Experiment/DataCentre
Monitoring: scenarios

• …when is all of this worth it?
Monitoring: scenarios (I)

- Impact of software releases:

Downgrade 8.9.2 to 8.7.10
Monitoring: scenarios (II)

- Better understanding of the pool (issues, bottlenecks):

![Graph showing various metrics and metrics with a red box highlighting a problematic rate of submissions.](https://indico.cern.ch/event/817927/)
Monitoring: scenarios (III)

• Improve the pool, dedicated subcollector for schedds:

Router queries to subcollector
Monitoring: scenarios (IV)

- How to optimize our resources?

Use spare power…
What’s next?
**Next**

- We don’t want to waste our time…
  - …babysitting 20K nodes.
  - …or looking to tens of dashboards.
- Working on prototypes to orchestrate operations:
  - **Draining actions** (based on health checks, cloud interventions)
  - **Self-healing infrastructure**: take decisions based on multiple alarms.
  - **Preventive operations**: continuously drain 10% to deploy new kernel in the worker nodes.
- Keeping an eye on: consul, kafka, spark, etcd, kraken, vitrage, alertmanager.
- Open to ideas and projects… 😊

---

<table>
<thead>
<tr>
<th>Component</th>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSF Pending Drain</td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>Condor Pending Drain</td>
<td>79%</td>
<td></td>
</tr>
<tr>
<td>Now Draining</td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>Pending Reboot</td>
<td>908</td>
<td></td>
</tr>
<tr>
<td>Pending Kill</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>Pending SSH</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>Server Status</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

25/09/2019  
https://indico.cern.ch/event/817927/