Containerized Batch System Monitoring.

Monitoring batch system jobs with container-oriented tools enhanced by LRMS job data and distributed over CVMFS

Monitoring HEP-jobs like containers

Since microservices as containers, have become en-vogue, various monitoring tools have been emerged. While such tools are primarily focused on Docker as the framework with the largest followership, these tools can be put to use also for a more generic approach. Since containers are based on standard kernel features to encapsulate processes with their own environments in dedicated namespaces and cgroups, these container monitoring tools can be reused more generically for any kernel resources in cgroups.

While containers and container orchestration frameworks are becoming more established as LRMSes alongside traditional batch systems, batch systems as HTCondor [2] or Slurm are prevalent in the HEP world as the workloads of the Grid computing communities have evolved along these.

Since both, batch systems and container frameworks, use the same kernel features for resource management and control on hosts, it suggests itself to refit established off-the-shelf tools for monitoring.

We chose Google's cAdvisor [1] as lightweight tool without further depoendencies and a rich REST API.

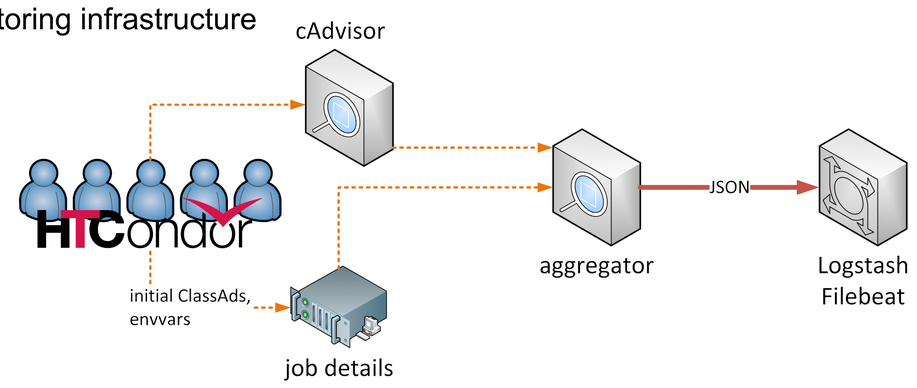
We deploy cAdvisor and our Logstash-based extension as Singularity containers via CVMFS

Expanding HTCondor Job Details

We enrich the standard cAdvisor statistics for each HTCondor job by details from the jobs' environments. While HTCondor provides tools to poll job and slot ClassAds, we decided to extract ClassAd values, which are static over a job's lifetime, from the process itself.
Likewise, we evaluate the initial job environment to extract further job details.

Using only the file system allows to minimize the load on HTCondor daemons as well as opens up an exercise path for a more generic approach. For example, to collect information from other batch systems or for initial environment variables set by users.

After polling locally the base stats as JSON blobs from cAdvisors REST API and extending these without our own stats, we forward the per job JSON data to our ELK [3] monitoring infrastructure

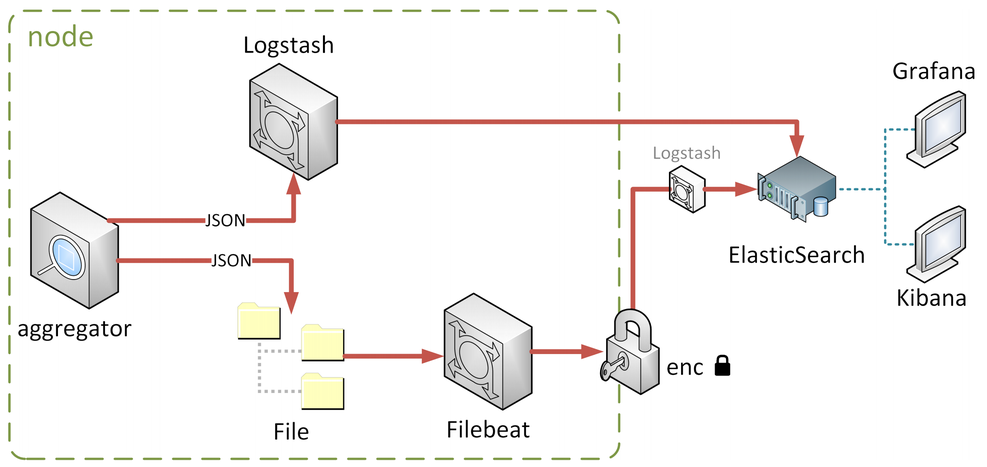


Aggregation Paths

Logstash

Initially, we envisaged Logstash as primary tool to aggregate data and forward them to our Elastic Search instance

- self-contained Singularity container via official Logstash builds on Dockerhub
- entry points: either plain Singularity run or as Singularity instance
- dependency on the cAdvisor instance
- aggregator scripts triggered by Logstash
- site-specific parameters, job environment variabls and machine ClassAds dynamically settable via SINGULARITY_ENV variables in the environment



Filebeat

When the Filebeat setup for generic log output aggregation got secured communication channels to comply with the GDPR for sensitive data, we added optionally a simple file output. As advantage, one can use existing aggregation paths.
As no service is necessary within the container, a simple timer and service unit can periodically call the aggregation scripts.

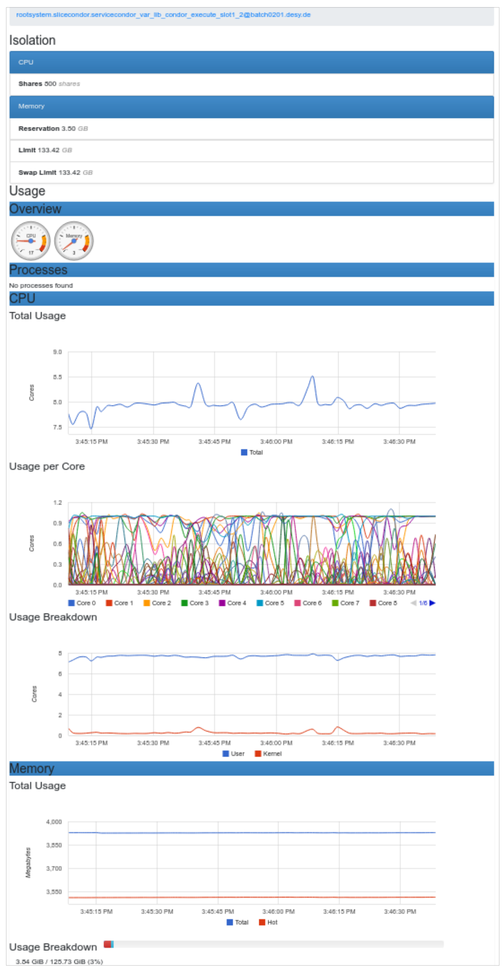
Job Visualization

Historic Job Statistics

The aggregated statistics per job slot can be sliced and selected for example per job, user or group level. E.g., a job's memory and CPU evolution selected by it's ARC CE ID or it's HTCondor job ID.

Node Utilization View

Additionally, cAdvisor brings also a build-in web server to visualize a host's current utilization. Thus, if a system administrator has to debug a node, he or she can gain quickly a graphical overview of the current user loads with one port forwarding.



View on a node on the current status of a CMS multi-core job

Evolution of a CMS multi-core job with internal scheduling

Next Steps

During the setup of the monitoring service, valuable experiences could be gained on the inner workings of systemd and CVMFS

- code clean-up and refactoring
- more injection friendly for other additions
- from other LRMSes as SLURM
- investigate how per node polling current Condor internal state information/chirp scales over LAN
- include details from jobs on the National Analysis Facility
- friendly interface for end user, e.g., in Grafana
- Evaluate, if/how cgroup network controllers as network can be added to gain also per job network statistics (excluding NFS4)

1 2 3