Job Monitoring Plans @ DESY

Extending the user job monitoring on the National Analysis Facility

Thomas Hartmann
DESY IT

NAF in Numbers

BIRD HTCondor Cluster

HTCondor Cluster:

- ~8000 cores
- ~52TB total memory / >6GB memory/slot
- > 40GB disk/slot
- shared scratch space
- Ethernet networking (10G for newer generations)
- GPU batch nodes available

Workgroup Servers:

- per group
  - dedicated servers
  - VMs
- user development
- batch submission
- specialized GPU workgroup servers

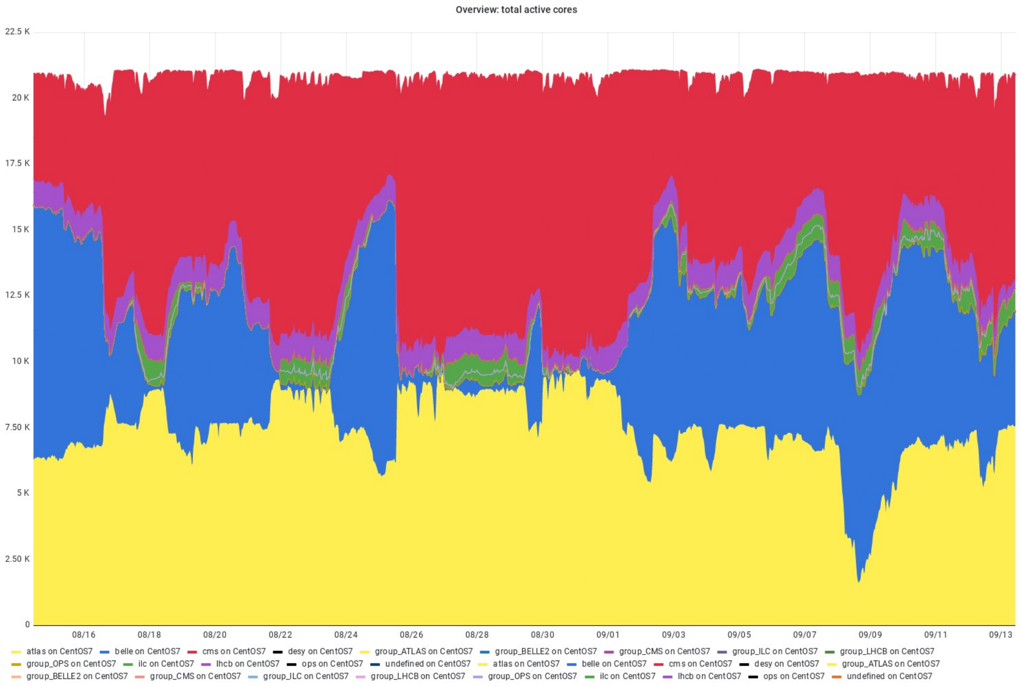
HTC and HPC workloads at DESY

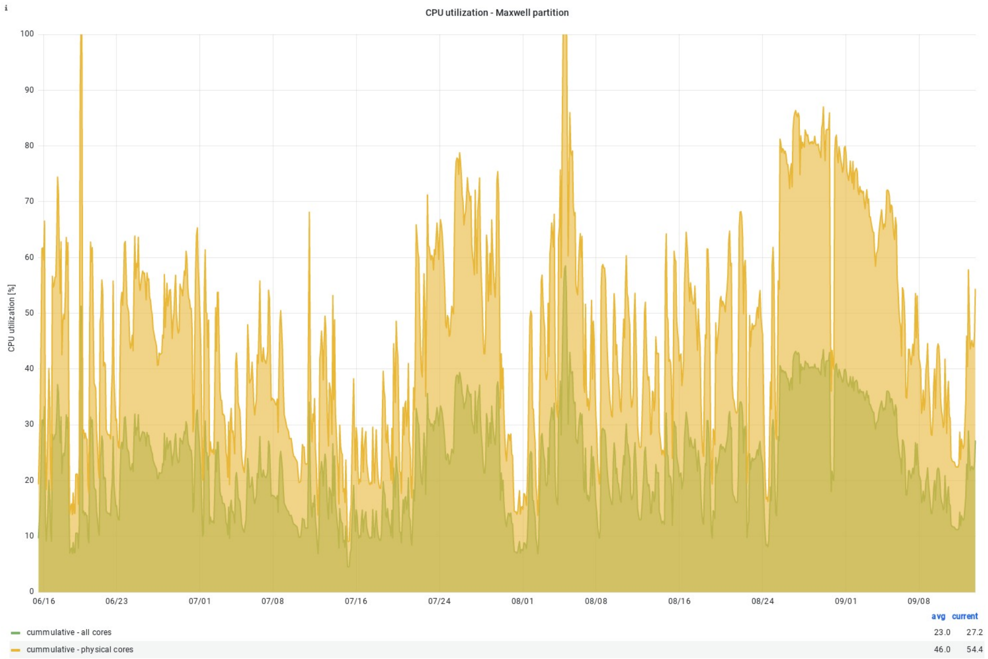
Clusters: Grid:HTC & Maxwell:HPC

High Throughput Computing

High Performance Computing

Grid (HTCondor)
Maxwell (slurm)



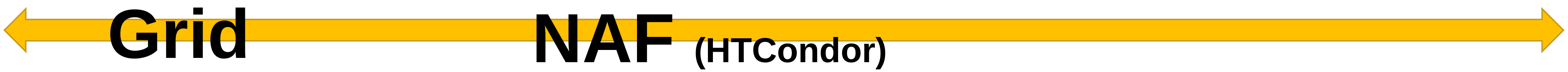


NAF

Complementing the Grid

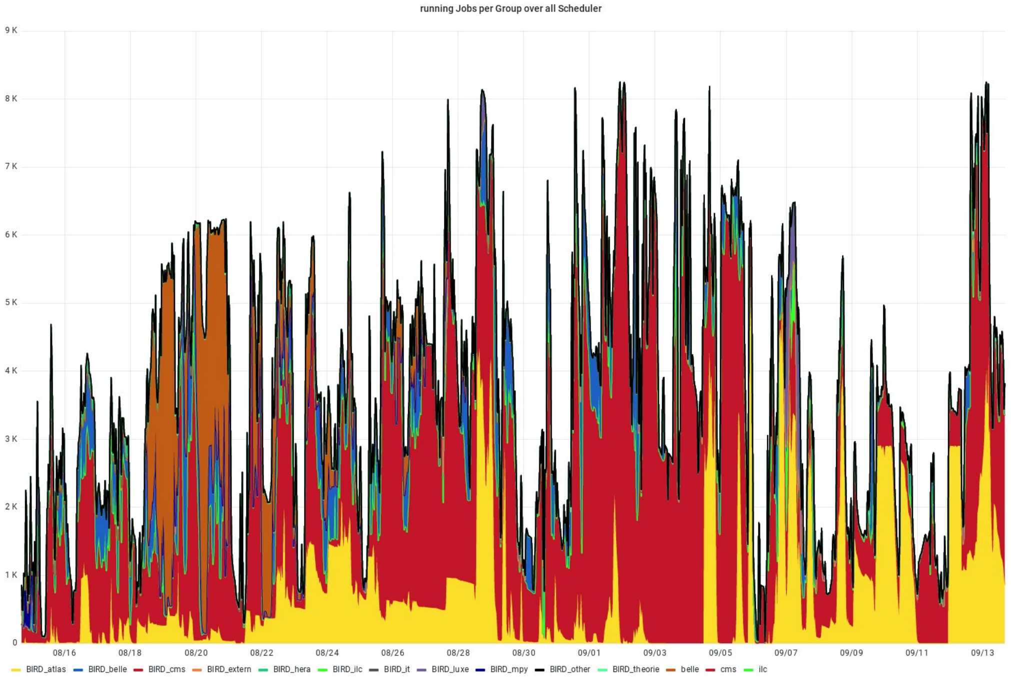
High **Throughput** Computing

High **Performance** Computing



Grid

NAF (HTCondor)



- same base set up as Grid
- NAF specific optimizations

- individual users with dynamic workloads
- start latency more critical
  - utilization >75% affects user patience

Data Storages

Storage Instances

File Systems/Storages

- NFS4 mounted

- GPFS scratch space
  /nfs/dust/{**atlas**,**belle**,**cms**,…}

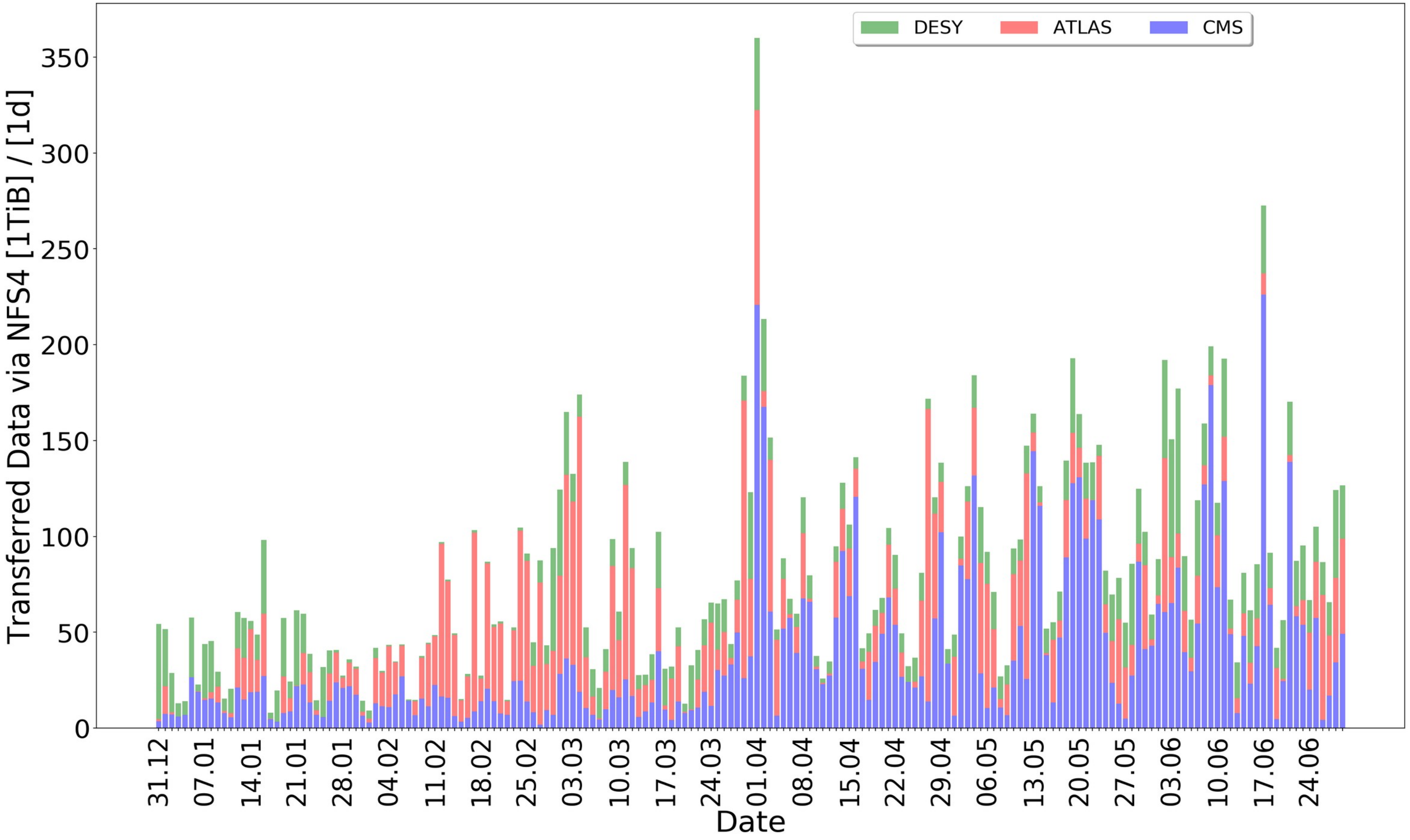
- dCache grid storages per VO
  /pnfs/desy.de/{**atlas**,**belle**,**cms**,…}
  - tape support for Belle II (only dedicated users)

- & AFS & CVMFS

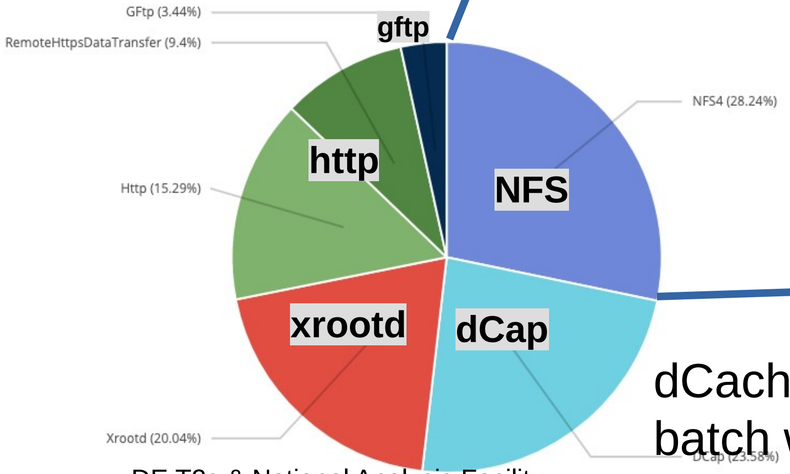
- users like paths, i.e., POSIX
  - Paths do not communicate different storage types

Data I/O over NFS on the NAF

Users like NFS



dCache NFS I/O from NAF users



dCache I/O per protocols from batch workers (NAF & grid)

NAF Wroker Health

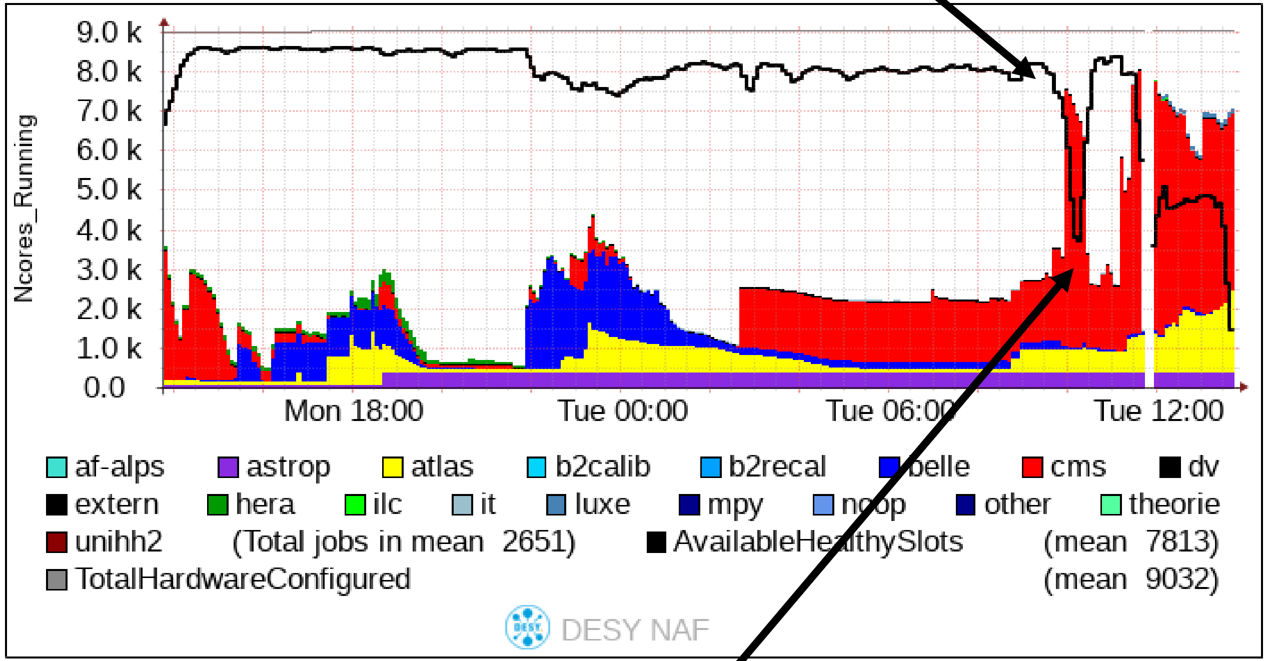
User I/O affecting Storage Clusters

Storage health cross-feeding into the NAF cluster (and vice versa)

- occasionally *awkward* user I/O patterns
  - storage DDOS: parallel reads of single files
- NFS client issues
  - 4.1 client implementations buggy (EL7/3.10)
- storage server bottlenecks

Nodes with mount issues

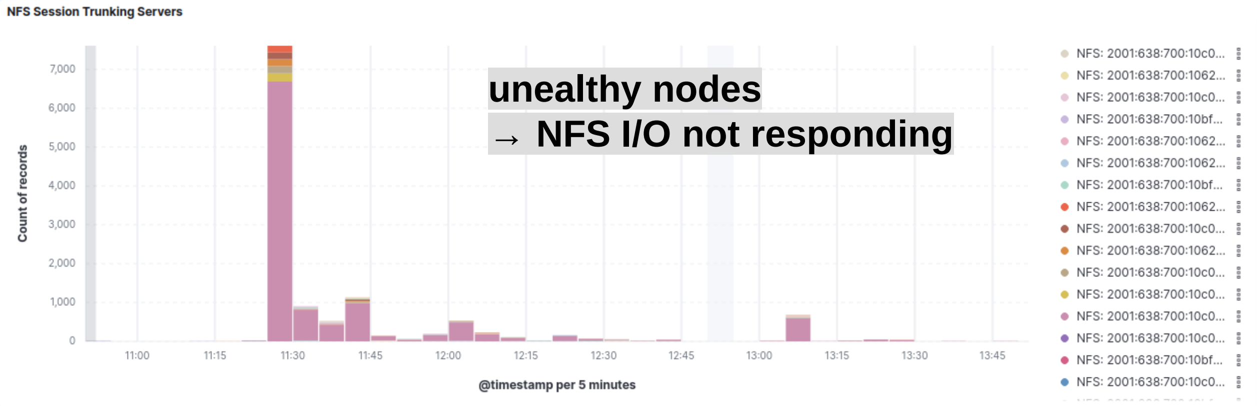
- setting themselves unhealthy
- rejecting new jobs



> userfoo:cms !?

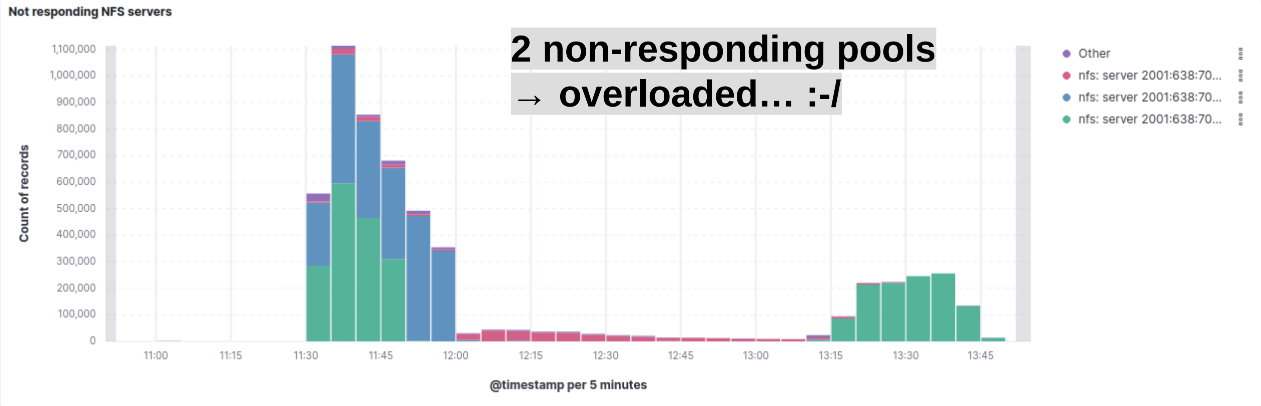
Misbehaving jobs vs. pool Issues

ddos-like jobs



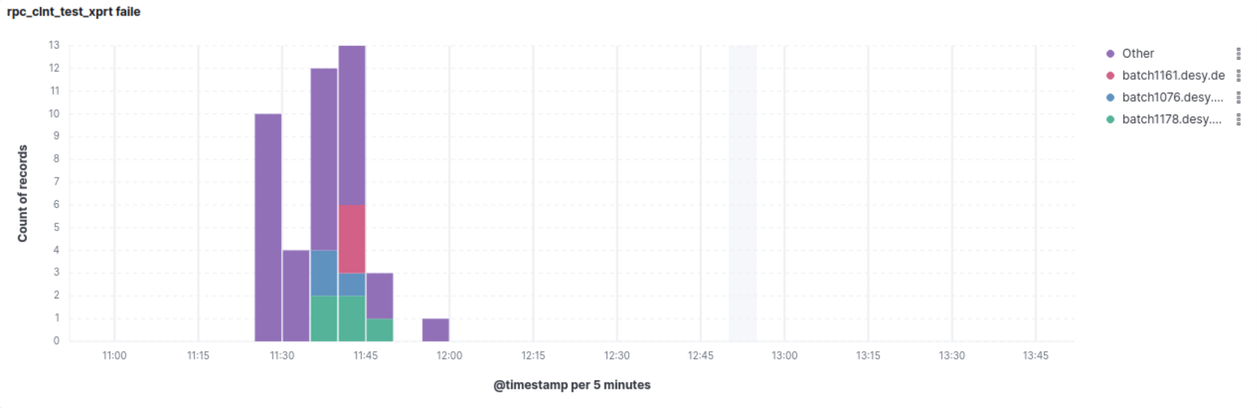
NFS Session Trunking Servers

unealthy nodes
→ NFS I/O not responding

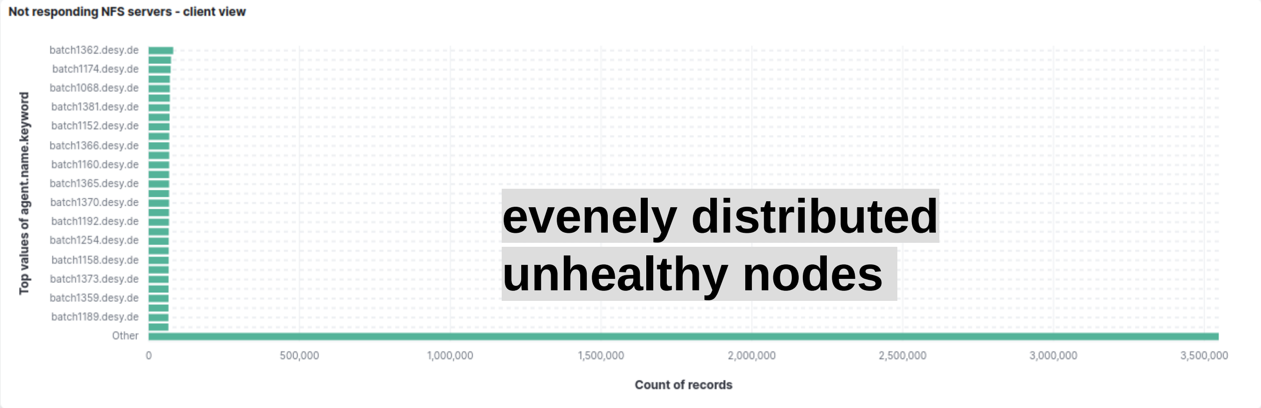


Not responding NFS servers

2 non-responding pools
→ overloaded... :-/



rpc_clnt_test_xprt faile

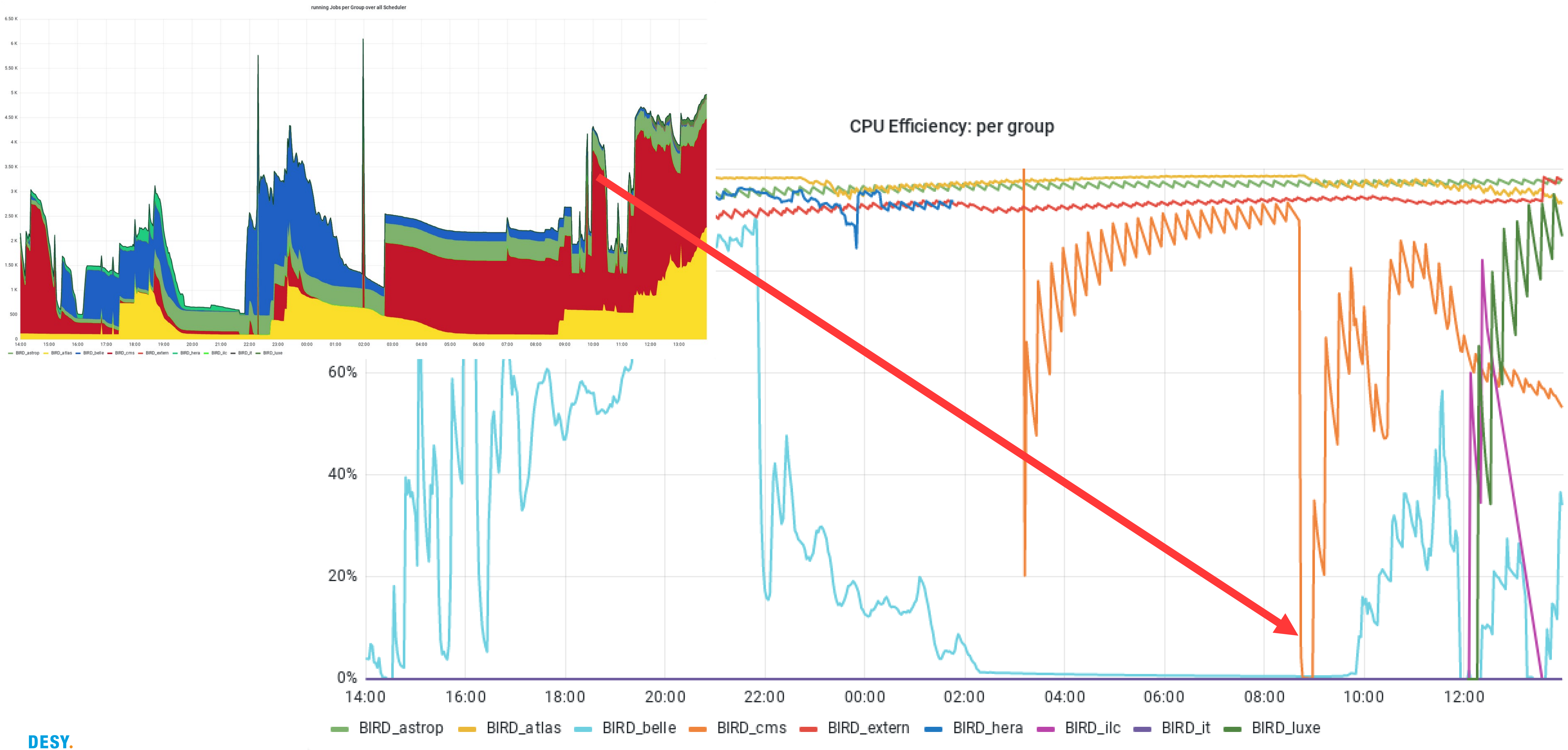


Not responding NFS servers - client view

evenely distributed
unhealthy nodes

Misbehaving jobs vs. pool Issues

Job efficiency

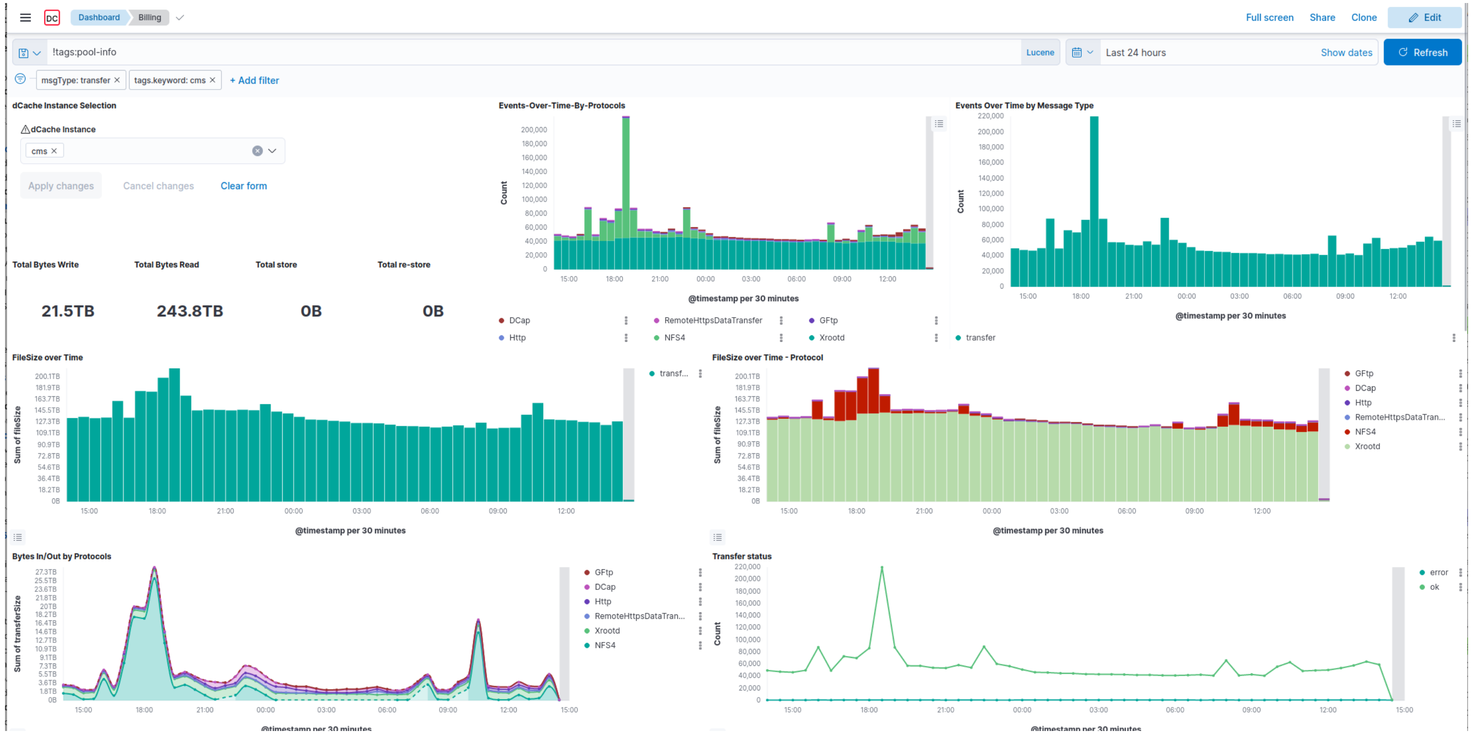


Storage View ↔ Worker/Job I/O View

dCache Instances Monitoring → complementing compute view

- good overview on the storage site
  - which worker is accessing what
  - individual storage pool view

- complementary view from compute
  - which job/user r/w what?
  - where all over the cluster?



Dashboard / Billing

!tags:pool-info

msgType: transfer

tags.keyword: cms

dCache Instance Selection

dCache Instance: cms

Last 24 hours

Total Bytes Write	Total Bytes Read	Total store	Total re-store
21.5TB	243.8TB	0B	0B

Monitoring Consolidation

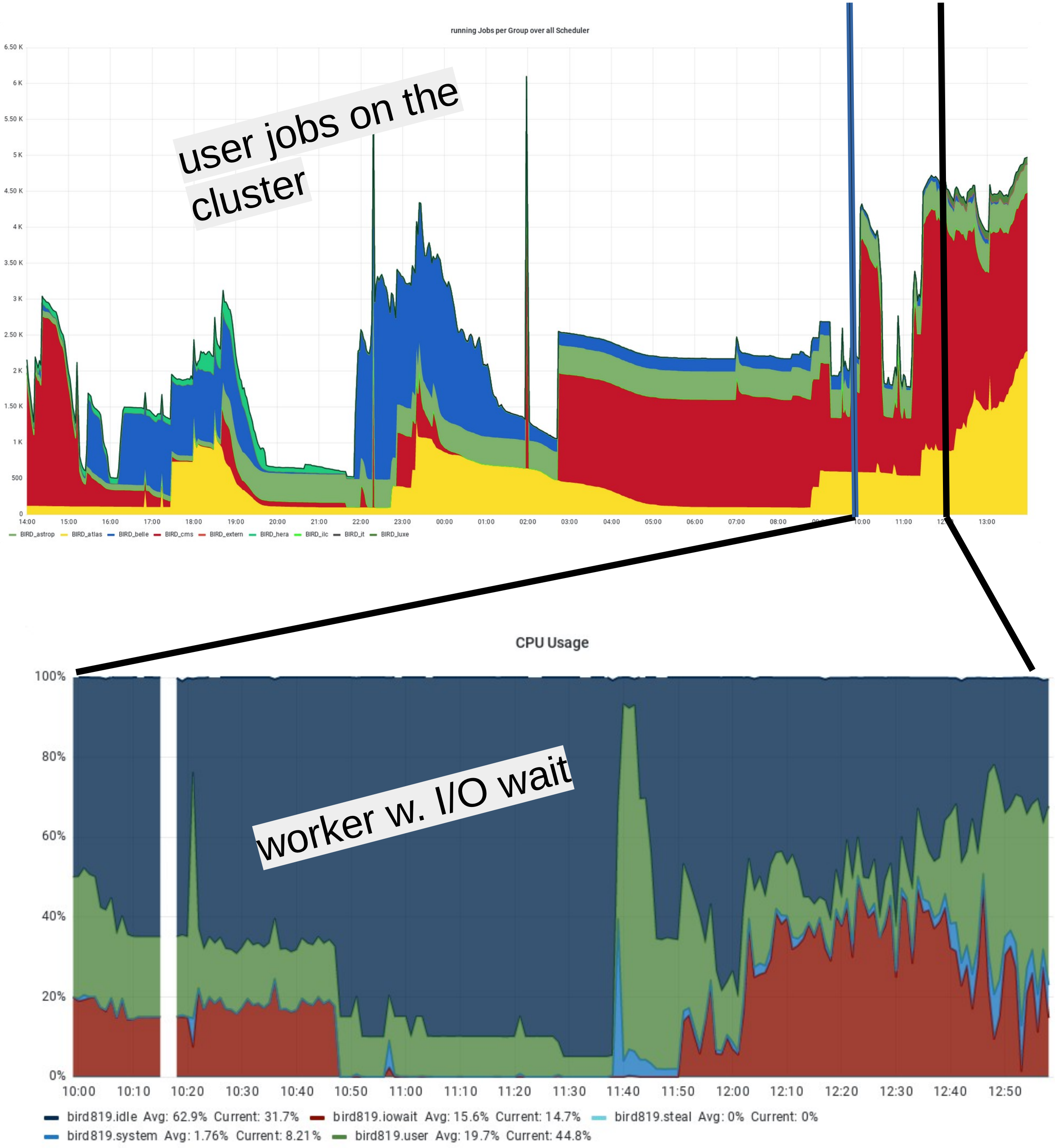
Extending & bringing together worker & job info

various information views

- individual workers know, when mounts are responsive
- cluster knows, how which users are running

storage instances know I/O towards workers

- how to distinguish between between broken user I/O and intrinsic storage instance issues?
  - workers just see I/O wait, unresponsive requests,…
  - need job view



user jobs on the cluster

worker w. I/O wait

Job Profiling

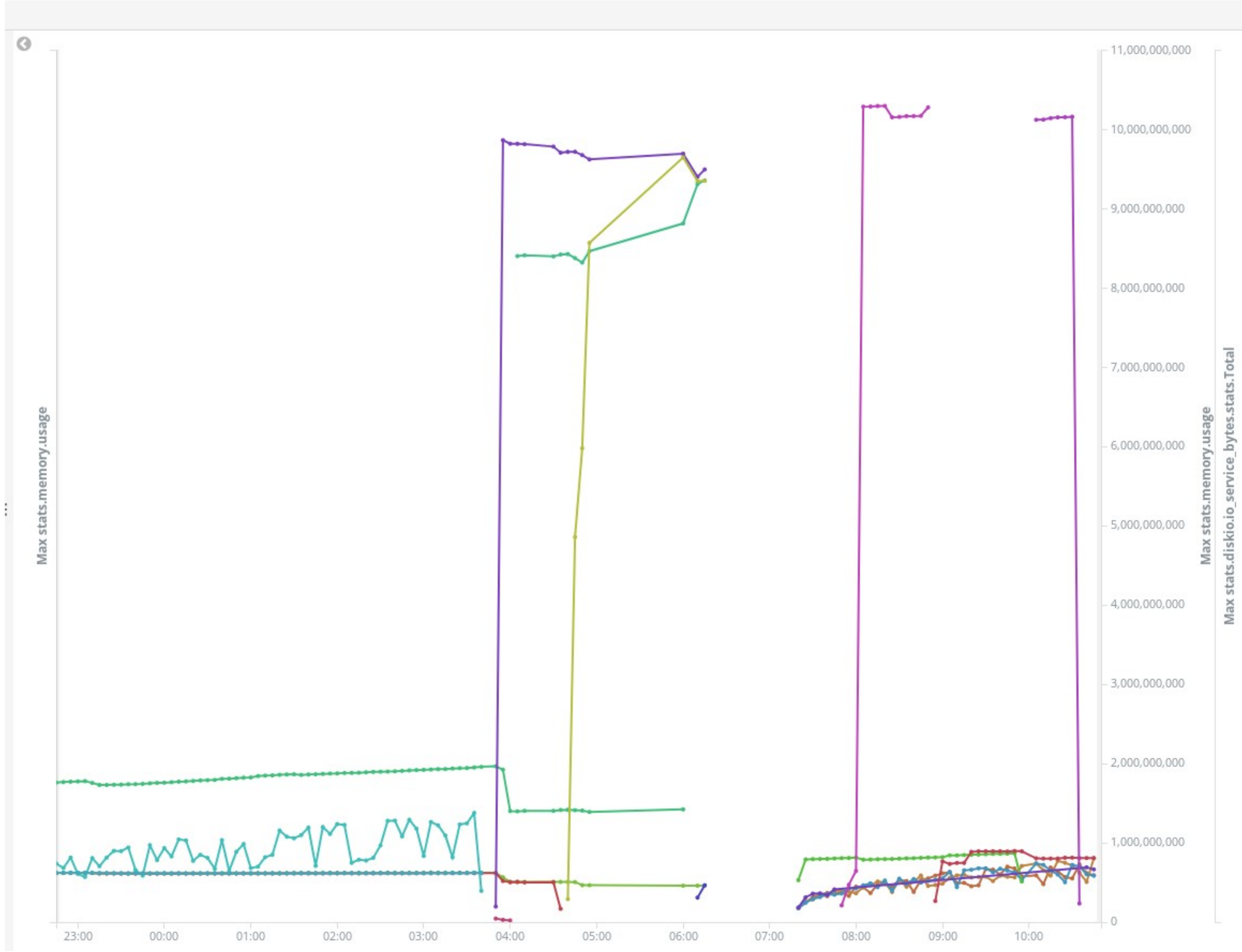
Consolidating job metrics

- job profiling with cAdvisor
  - Singularity container via CVMFS
  - worker view on jobs
    - what cgroups (aka jobs) active?
    - HTCondor *external*
    - somewhat lackluster use

⟿ while on aggregating path/file I/O information overhaul job profiling

drop cAdvisor

- moving to job (ad) view

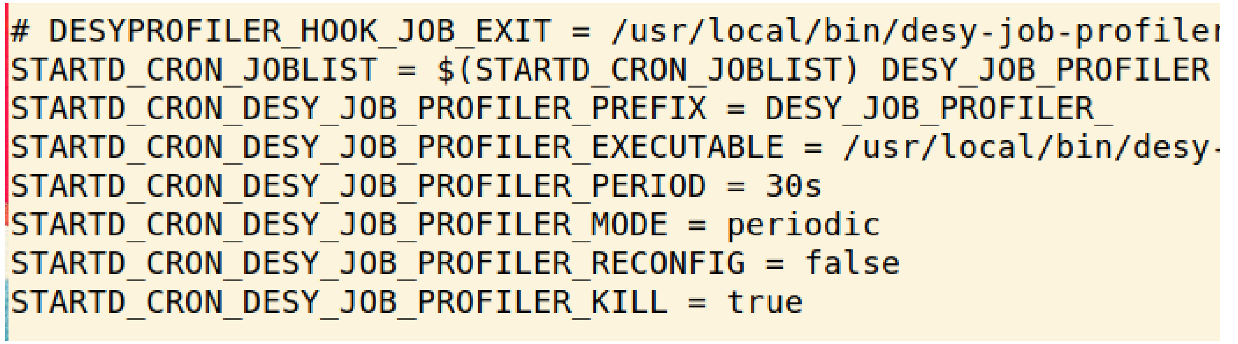


ELK: job memory usage on a worker

Job Metrics

Where/How collect and forward job metrics/ads?

- startd cron?
  - easy to deploy
    & connect to Kafka → ElasticSearch
  - worker view
    - need to gather jobs & job procs
    - subsequent mapping to jobs necessary



```
# DESYPROFILER_HOOK_JOB_EXIT = /usr/local/bin/desy-job-profiler
STARTD_CRON_JOBLIST = $(STARTD_CRON_JOBLIST) DESY_JOB_PROFILER
STARTD_CRON_DESY_JOB_PROFILER_PREFIX = DESY_JOB_PROFILER_
STARTD_CRON_DESY_JOB_PROFILER_EXECUTABLE = /usr/local/bin/desy-
STARTD_CRON_DESY_JOB_PROFILER_PERIOD = 30s
STARTD_CRON_DESY_JOB_PROFILER_MODE = periodic
STARTD_CRON_DESY_JOB_PROFILER_RECONFIG = false
STARTD_CRON_DESY_JOB_PROFILER_KILL = true
```

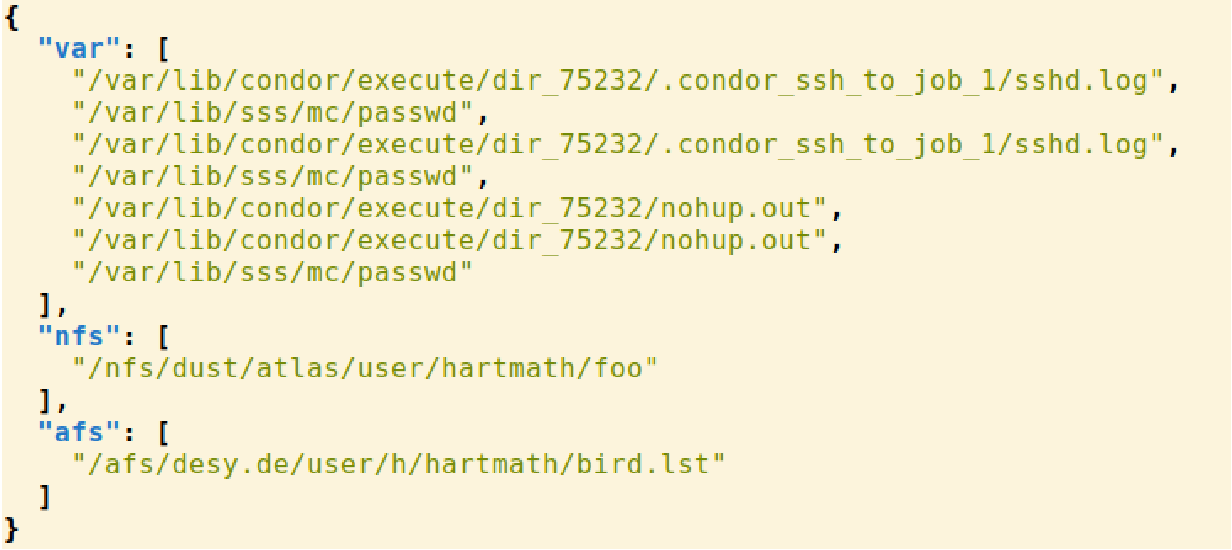
  - status condor procd?
    - option instead/in addition of tracking PIDs in cgroups tasks?

Job Metrics II

Where/How collect and forward job metrics/ads?

- job profiling shepards?
  - forking from job wrapper
  - in user context (risk of user interference…?)
  - no direct ES connection reasonable

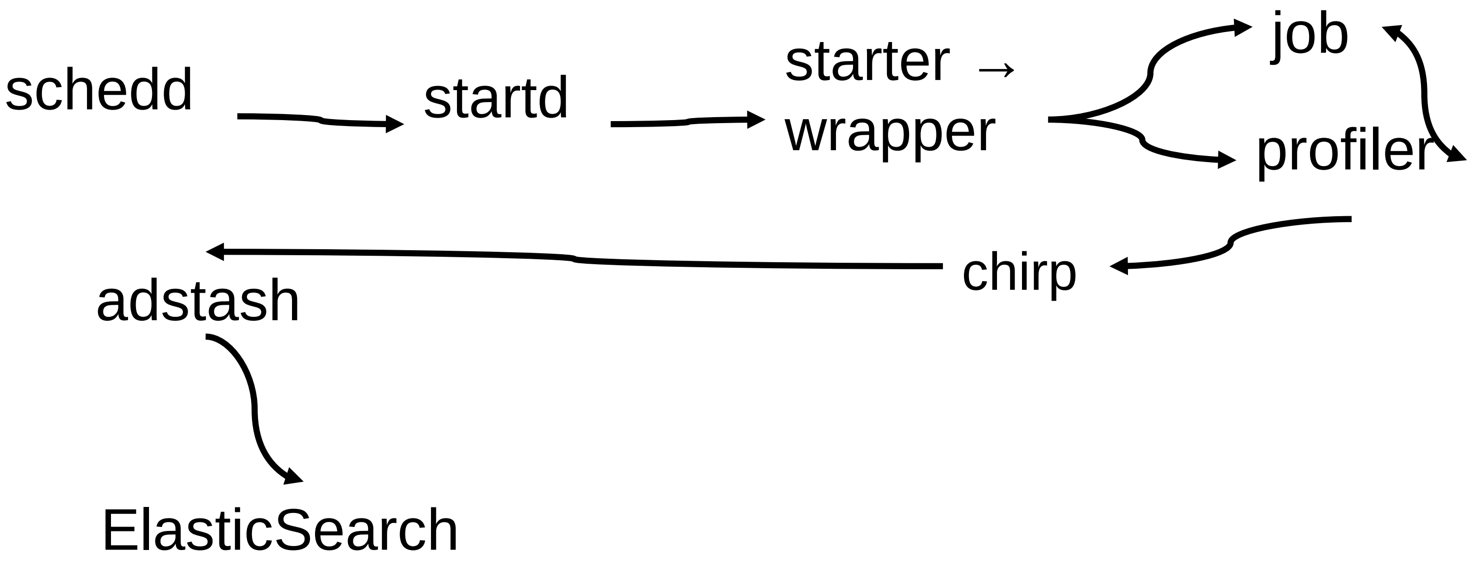
  - chirp metrics as additional class ads?
    - does the cluster communication scale?



```
{
  "var": [
    "/var/lib/condor/execute/dir_75232/.condor_ssh_to_job_1/sshd.log",
    "/var/lib/sss/mc/passwd",
    "/var/lib/condor/execute/dir_75232/.condor_ssh_to_job_1/sshd.log",
    "/var/lib/sss/mc/passwd",
    "/var/lib/condor/execute/dir_75232/nohup.out",
    "/var/lib/condor/execute/dir_75232/nohup.out",
    "/var/lib/sss/mc/passwd"
  ],
  "nfs": [
    "/nfs/dust/atlas/user/hartmath/foo"
  ],
  "afs": [
    "/afs/desy.de/user/h/hartmath/bird.lst"
  ]
}
```

Job Metrics III

Current plan



schedd

startd

starter → wrapper

job

profiler

chirp

adstash

ElasticSearch

- only a posteriori job profiles with adstash?
  - chirp and append stat arrays/lists

- near ~live updates of profiling job ads possible with adstash?

Summary

Where/How collect and forward job metrics/ads?

- overhauling our job monitoring
- zooming in on the job
  - maybe recycling metric stuff from pilots…?

- user/job file(?) I/O in the future?
  - columnar analyses event based → event *layer* ontop file
  - how to profile such access patterns?

Thank You

Questions?

Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Thomas Hartmann
DESY IT
https://naf.desy.de
https://bird.desy.de