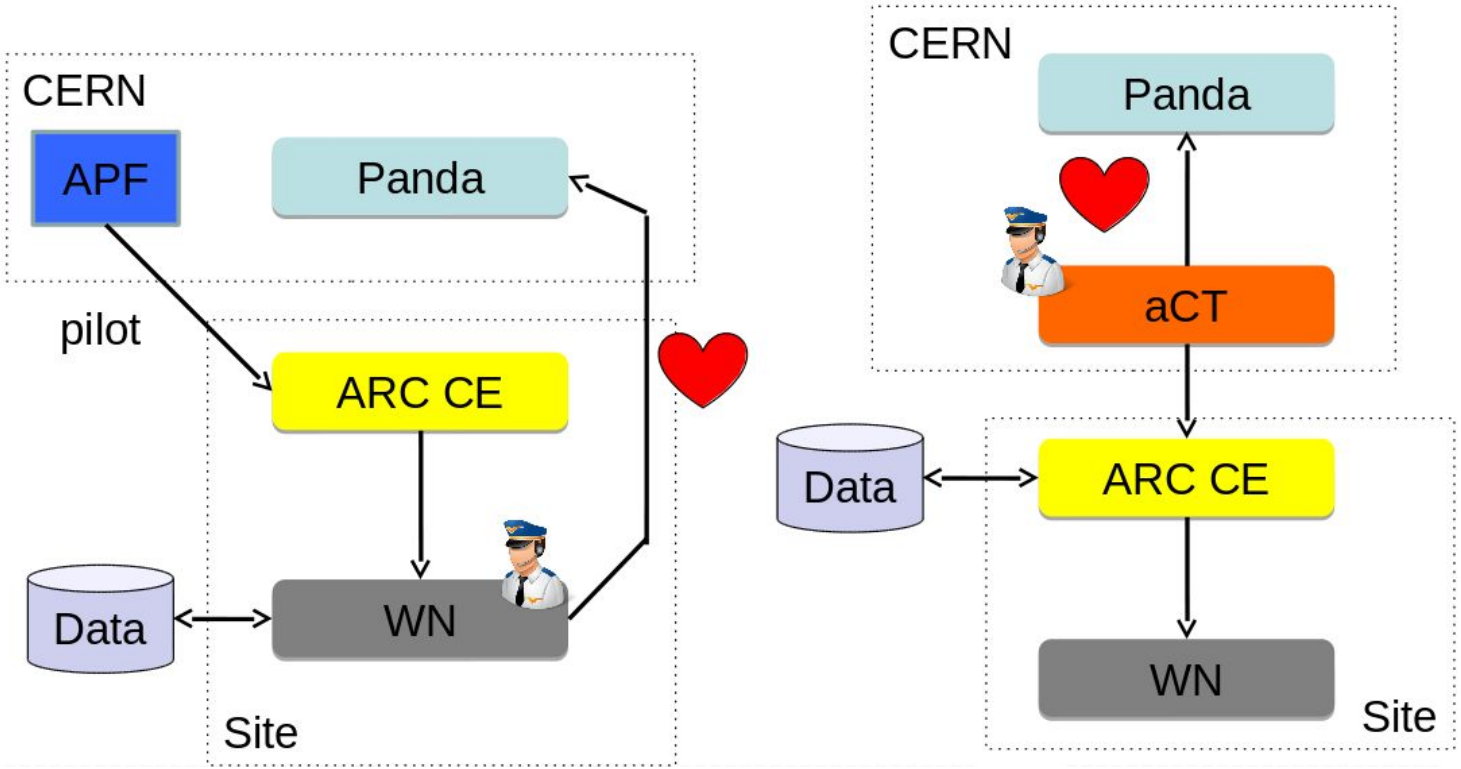
aCT: an introduction

History

- NorduGrid model was built on philosophy of ARC-CE and distributed storage
  - No local storage - data staging on WN is too inefficient
  - No middleware or network connectivity on WN
  - Everything grid-related was delegated to ARC-CE
- Panda and pilot model did not fit easily
  - An intermediate service was needed to fake the pilot and submit to ARC behind the scenes
  - ~2008 ARC Control Tower (aCT) was written by Andrej and ran in Ljubljana
  - 2013-14 aCT2 was written and the service moved to CERN
    - Multi-process instead of multi-thread, MySQL instead of sqlite
  - 2015-17 Many sites moved from CREAM CE to ARC CE
    - Creation of truepilot mode

APF vs aCT (NorduGrid mode)

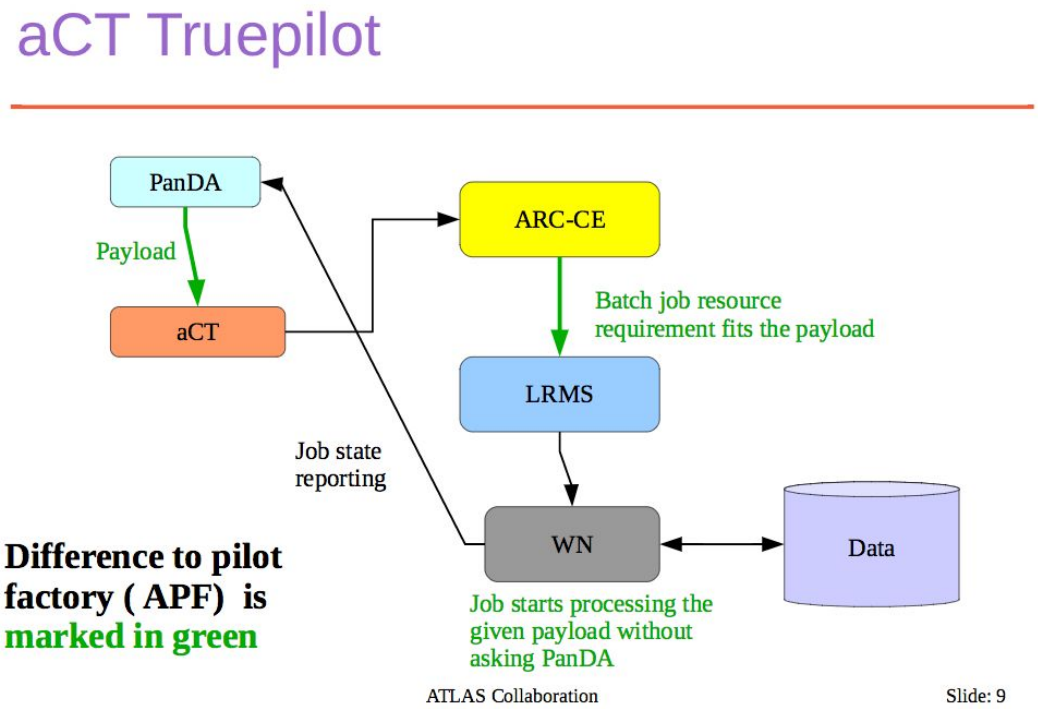


NorduGrid mode

- AGIS: pilotmanager = aCT, copytool = mv
- aCT has to emulate certain parts of the pilot
  - getJob(), updateJob()
- Post-processing
  - Pilot creates pickle of job info and metadata xml of output files
  - ARC wrapper creates tarball of these files along with pilot log
  - aCT downloads this tarball after job has completed
  - Log is copied to web area for access via bigpanda stdout links
  - Pickle info is altered to set schedulerid and log url
  - Output files are validated (check size and checksum on storage vs pilot xml)
  - Job info and xml are sent to panda with final heartbeat
- If job fails badly (pilot crash or batch system kill) and no pilot info is available
  - aCT sends what it can to Panda
  - Error info and timings from the CE

True pilot

- AGIS: pilotmanager = aCT, copytool != mv
- For sites running ARC CE who do not need the full “native” mode with staging etc
- aCT fetches the payload and submits it to the ARC-CE
- ARC-CE submits the batch job with predefined payload and requirements
- Pilot on the worker node does the same as on the conventional pilot sites, but skips the fetching of payload from PanDA
- aCT sends heartbeats to Panda up until job is running, then leaves it to pilot

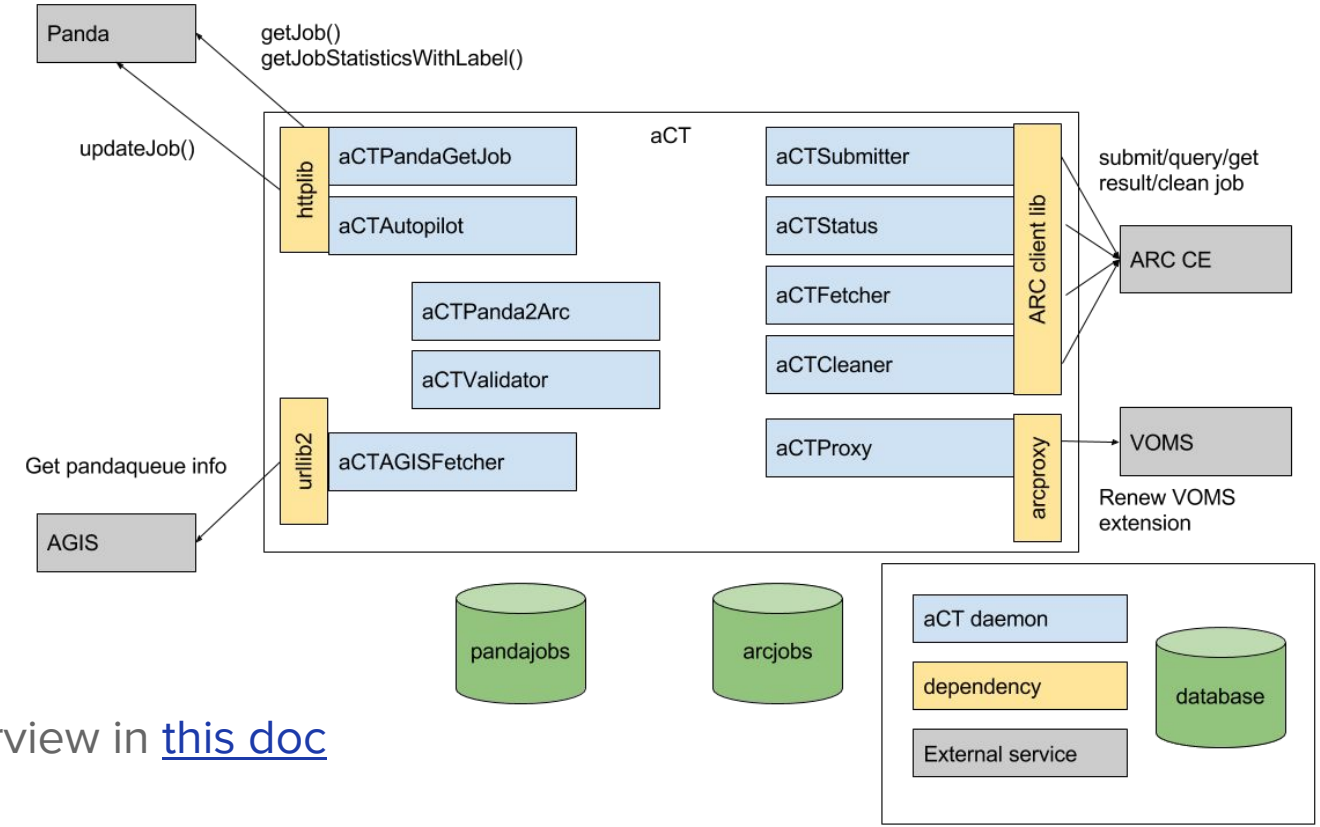


Event service (NorduGrid mode)

- For SuperMUC HPC and ES on BOINC, aCT prefetches events
  - getEventRanges() is called directly after getJob()
  - A file with the eventranges is uploaded to the CE when job is submitted
  - If pilot sees the eventranges file it uses it instead of asking Panda
- When job finishes, metadata xml is copied back to aCT to see which events were done
- aCT calls updateEventRanges() with the completed events

- For true pilot ES jobs aCT does nothing special

General Architecture



- Overview in [this doc]

aCT Daemons

ATLAS Daemons:

- aCTPandaGetJob: Queries panda for activated jobs and if there are any, gets jobs
- aCTAutopilot: Sends heartbeats every 30 mins for each job and final heartbeats
- aCTAGISFetcher: Downloads panda queue info from AGIS every 10 minutes. This info is used to know which queues to serve and the CE endpoints.

ARC Daemons (use python ARC client library):

- aCTSubmitter: Submits jobs to ARC CE
- aCTStatus: Queries status of jobs on ARC CE
- aCTFetcher: Downloads output of finished jobs from ARC CE (pilot log file to put on web area, pickle/metadata files used in final heartbeat report to panda)
- aCTCleaner: Cleans finished jobs on ARC CE
- aCTProxy: Periodically generates a new VOMS proxy with 96h lifetime

Internal Daemons:

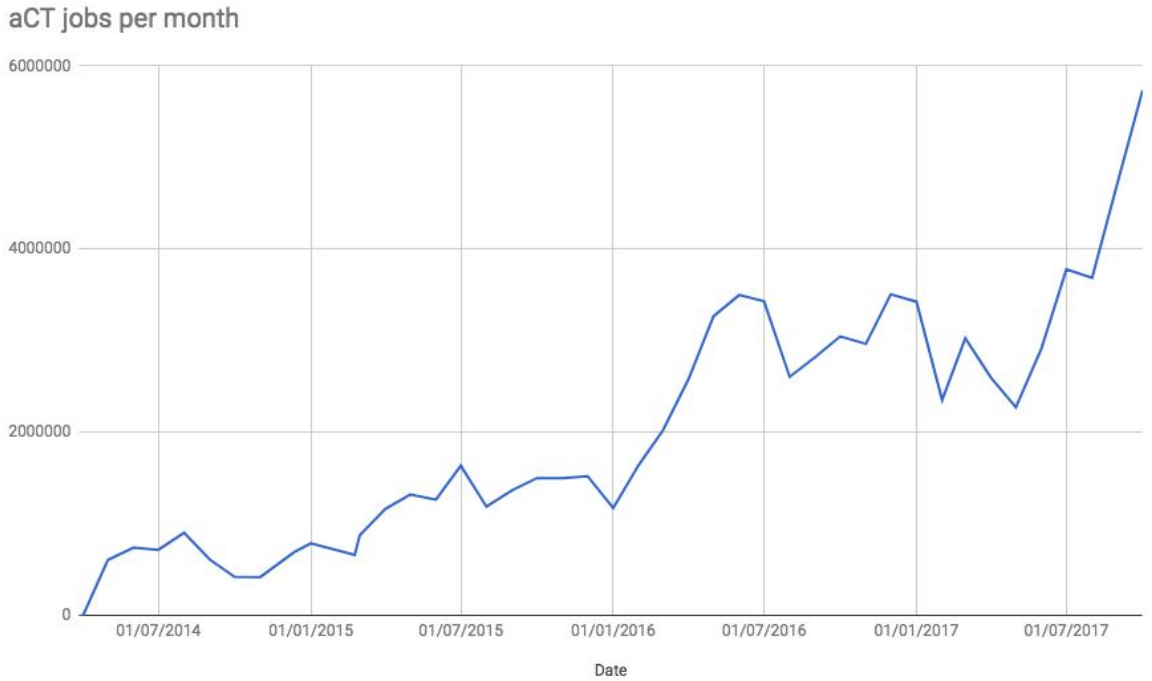
- aCTPanda2Arc: Converts panda job descriptions to ARC job descriptions and configures ARC job parameters from AGIS queue and panda job info
- aCTValidator: Validates finished jobs (checksum of output files on storage etc) and processes pilot info for final heartbeat report

Service setup and configuration

- 2 prod machines and 2 dev machines at CERN
- Prod machines use MySQL DBonDemand, dev machines have local MySQL
- Configuration is via 2 xml files, one for ARC and one for ATLAS
- One prod machine runs almost all jobs
- One prod machine is warm spare running one job per queue
  - <maxjobs>1</maxjobs> can be changed if main machine goes down
- Central services admin twiki

Current status

- ~200k jobs per day from one machine
- Peak 250k jobs per day
- Increase in last couple of months probably FZK



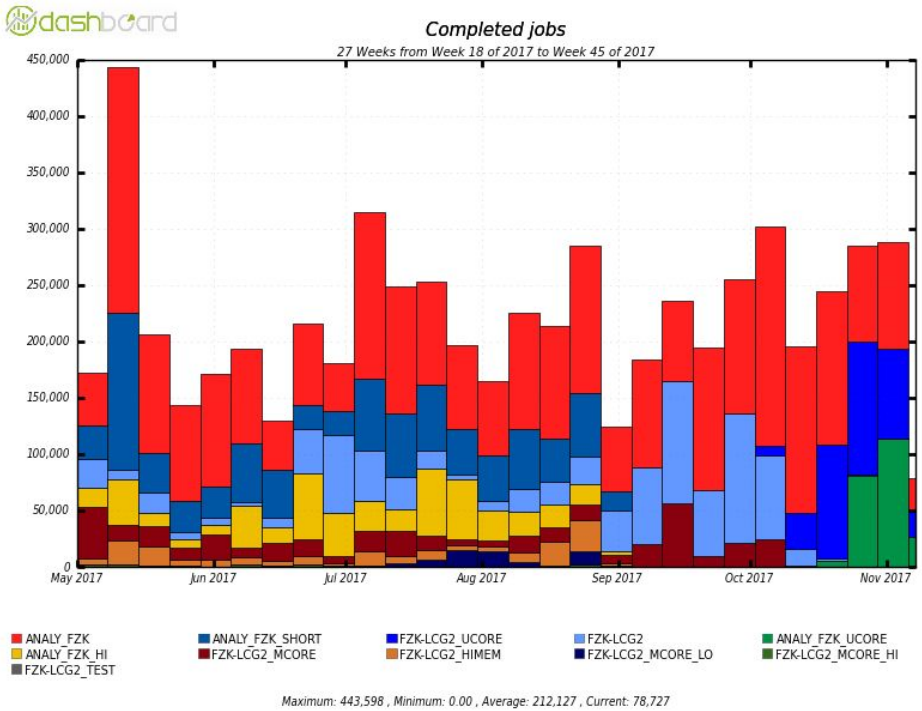
Sites served

NorduGrid
Truepilot

- T1: FZK, RAL, NDGF (4 sites), TAIWAN
- T2: CSCS, DESY, LRZ, TOKYO, MPPMU, BERN, WUPPERTAL, SiGNET, LUNARC
- T3: ARNES, SiGNET-NSC, UNIGE
- HPC: CSCS (PizDaint), LRZ (SuperMUC), IN2P3-CC (IDRIS, in testing), MPPMU (Draco + Hydra), SCEAPI (4 CN HPCs)
- Clouds: UNIBE Switch cloud
- Others: BOINC
- Full list at https://aipanda404.cern.ch/data/aCTReport.html

Unified queues

- Only change required in aCT was to take corecount from job instead of queue
- FZK went from 7 to 3 (soon 2) queues

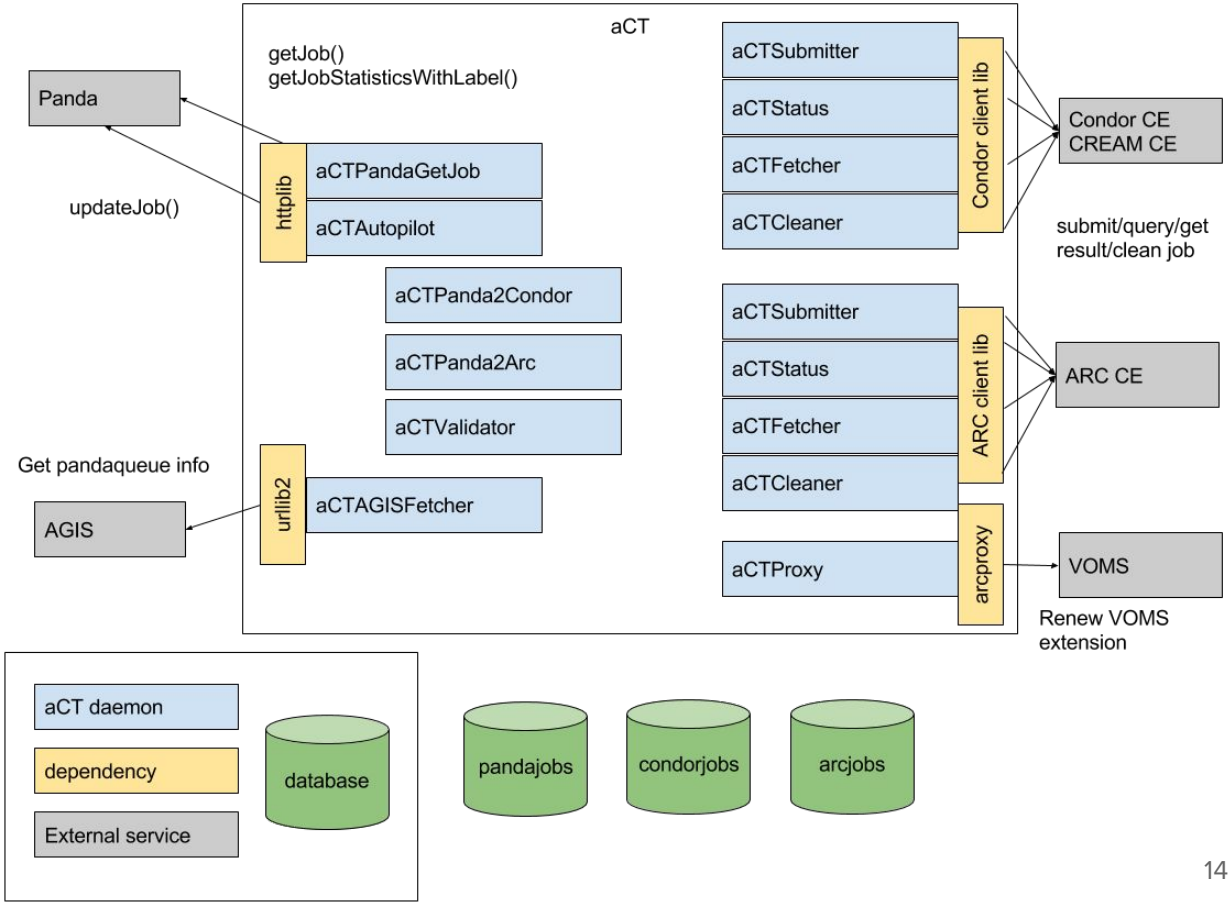


Panda communication

- getJob, updateJob, getJobStatisticsWithLabel (to check for activated jobs)
- getEventRanges, updateEventRanges
- Heartbeats sent every 30 mins or after status change
  - ~70k heartbeats/hour = 20Hz
  - A single process handles all jobs
  - A slight problem in communication with panda server can lead to large backlog
  - Solutions:
    - Heartbeat-less jobs while job is under aCT's control?
      - Only send heartbeat when status changes
      - When truepilot job is running it sends normal heartbeats
    - Bulk updateJob in panda server

Condor submission

- Separate DB table for condor jobs
- Submitter/Status/Fetcher/Cleaner for Condor
- Panda2Condor to convert pandajob to ClassAd
- Truepilot only



Condor submission

- Prototype has been implemented using condor python bindings (>= 8.5.8 needed)
- Using standard EU pilot wrapper
  - with one modification to copy the job description to working dir
- Submitter adds
  - 'GridResource': 'condor ce506.cern.ch ce506.cern.ch:9619'
- One example job
  - https://bigpanda.cern.ch/job?pandaid=3696722817

```
import htcondor
sub = htcondor.Submit(dict(jobdesc))
with schedd.transaction() as txn:
    jobid = sub.queue(txn)
return jobid
```

```
{'Arguments': '-h IN2P3-LAPP-TEST -s IN2P3-LAPP-TEST -f false -p 25443 -w
https://pandaserver.cern.ch',
 'Cmd': 'runpilot3-wrapper.sh',
 'Environment':
'PANDA_JSID=aCT-atlact1-2;GTAG=http://pcoslo5.cern.ch/jobs/IN2P3-LAPP-TEST/2017-11-07/$(Clu
ster).$(Process).out;APFCID=$(Cluster).$(Process);APFFID=aCT-atlact1-2;APFMON=http://apfmon
.lancs.ac.uk/api;FACTORYQUEUE=IN2P3-LAPP-TEST',
 'Error': '/var/www/html/jobs/IN2P3-LAPP-TEST/2017-11-07/$(Cluster).$(Process).err',
 'JobPrio': '100', <--- taken from job description
 'MaxRuntime': '172800', <--- taken from job description or queue
 'Output': '/var/www/html/jobs/IN2P3-LAPP-TEST/2017-11-07/$(Cluster).$(Process).out',
 'RequestCpus': '1', <--- taken from job description
 'RequestMemory': '2000', <--- taken from job description or queue
 'TransferInputFiles': '/home/dcameron/dev/aCT/tmp/inputfiles/3697087936/pandaJobData.out',
 'Universe': '9',
 'UserLog': '/var/www/html/jobs/IN2P3-LAPP-TEST/2017-11-07/$(Cluster).$(Process).log',
 'X509UserProxy': '/home/dcameron/dev/aCT/proxies/proxiesid5'}
```

Future plans

- Move code from gitlab to github
- Rename to ATLAS Control Tower (since it’s not ARC-specific any more)
- Better monitoring through APFmon, then harvester monitoring in bigpanda