Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

FSP CMS
Erforschung von
Universum und Materie

SPONSORED BY THE
Federal Ministry
of Education
and Research

CMS

Analysis Facilities Forum

German NAF: CMS user experience

Johannes Lange
johannes.lange@uni-hamburg.de

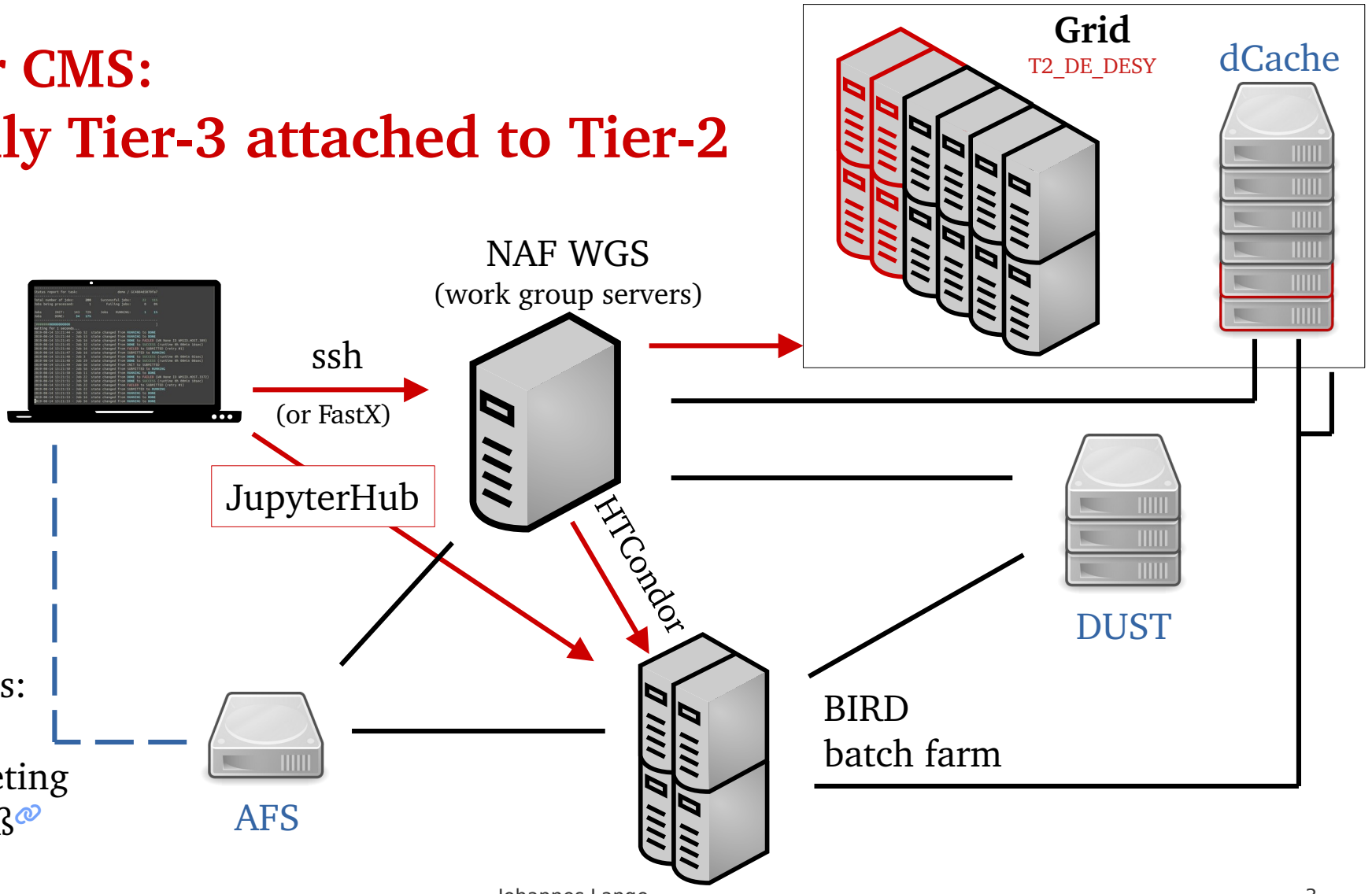
Universität Hamburg

27 October 2022

NAF: National Analysis Facility @DESY Hamburg

- Supposed to be used by the German HEP community
  - ATLAS, **CMS**, LHCb, ILC, CALICE, BELLE, HERA, smaller experiments
- CMS groups:
  - RWTH Aachen
  - DESY
  - Universität Hamburg
  - KIT

NAF for CMS:
Basically Tier-3 attached to Tier-2



Technical details:
talk at
AF Kick-off Meeting
by Christian Voß

NAF user storage

- AFS
  - `/afs/desy.de/user/<u>/<user>`
  - source code, documents etc.

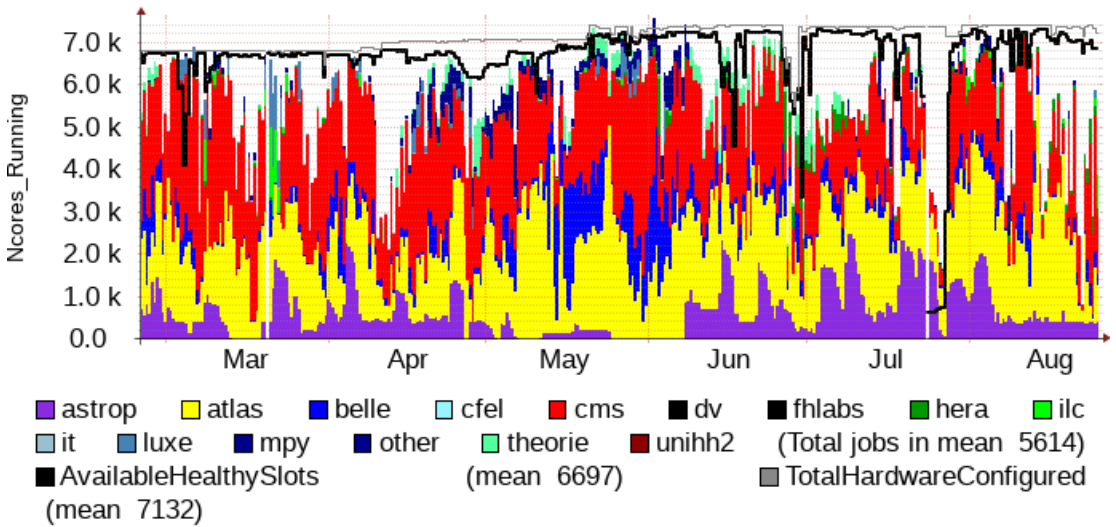
- DUST
  - `/nfs/dust/cms/user/<user>`
  - for large files (e.g. n-tuples),
    not intended for source code / many small files

- dCache
  - read-only NFS mount (WGS and worker nodes)
    `/pnfs/desy.de/cms/tier2/store/[data|mc|`**`user`**`]`
  - for large files (e.g. private productions, n-tuples)

NAF batch system: BIRD

(Batch Infrastructure Resource at DESY)

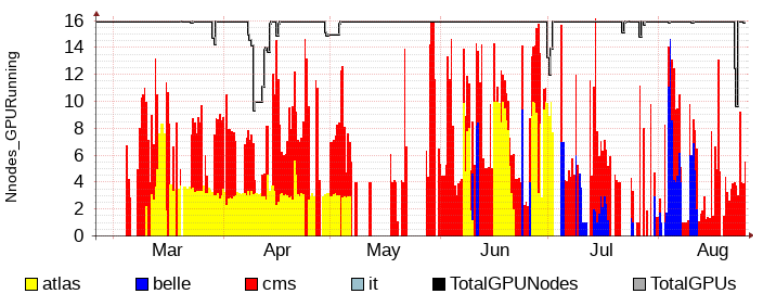
- HTCondor with fair share between experiments
  - migrated from SGE 2017 – 2018
- largest number of HEP groups in Germany: ATLAS



CMS

ATLAS

- Uni Hamburg CMS group integrates their resources in NAF
  → approx. equal share between ATLAS and CMS
- oversubscription for “lite” jobs
  (1 core, 2 GB mem, 3 hours runtime):
  more than fair share can be used, if resources are available
- GPU-nodes available:

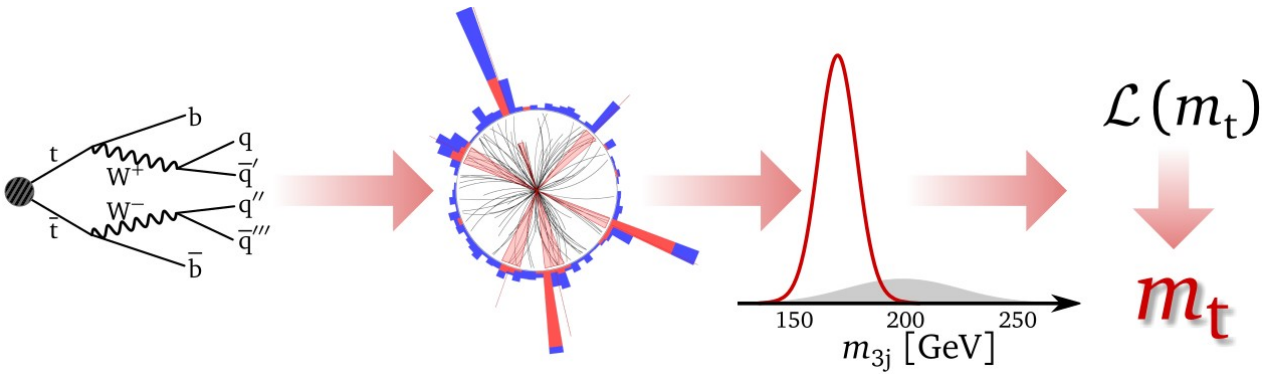


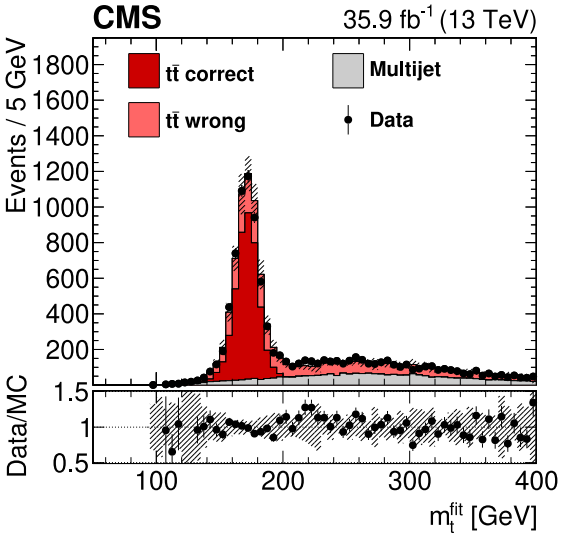
[EPJC 79 (2019) 313]

Example analysis

SM precision measurement: top quark mass, all-jets channel

- completely performed @NAF, no grid-jobs
- computationally challenging:
  - huge QCD multijet background
  - combinatorics, kinematic fitting
  - systematic uncertainties: many variations

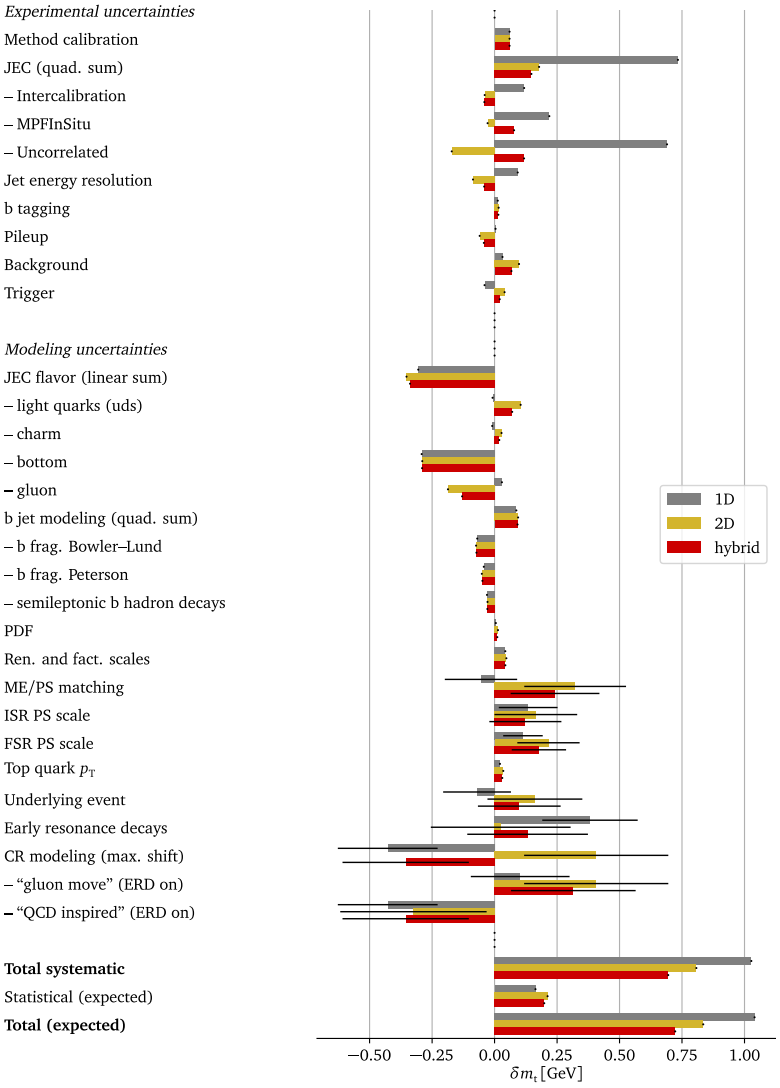




Example analysis

Overview of tasks

- Starting from MINIAOD data format (signal MC, background MC, data, background estimation from data)
- writing very specialized n-tuples (skimmed and slimmed) used for
  - plotting
  - statistical framework
- O(200) variations to process
- O(1000) pseudo-experiments per variation
- background estimation studies: different n-tuple format (slimmed)



Example analysis

job submission: grid-control

- we advertise grid-control as standard job submission tool
- take care of job submission, checking and possibly resubmission
- great parameterization mechanism
- a number of batch systems are supported as backend

[arXiv:1707.03198]

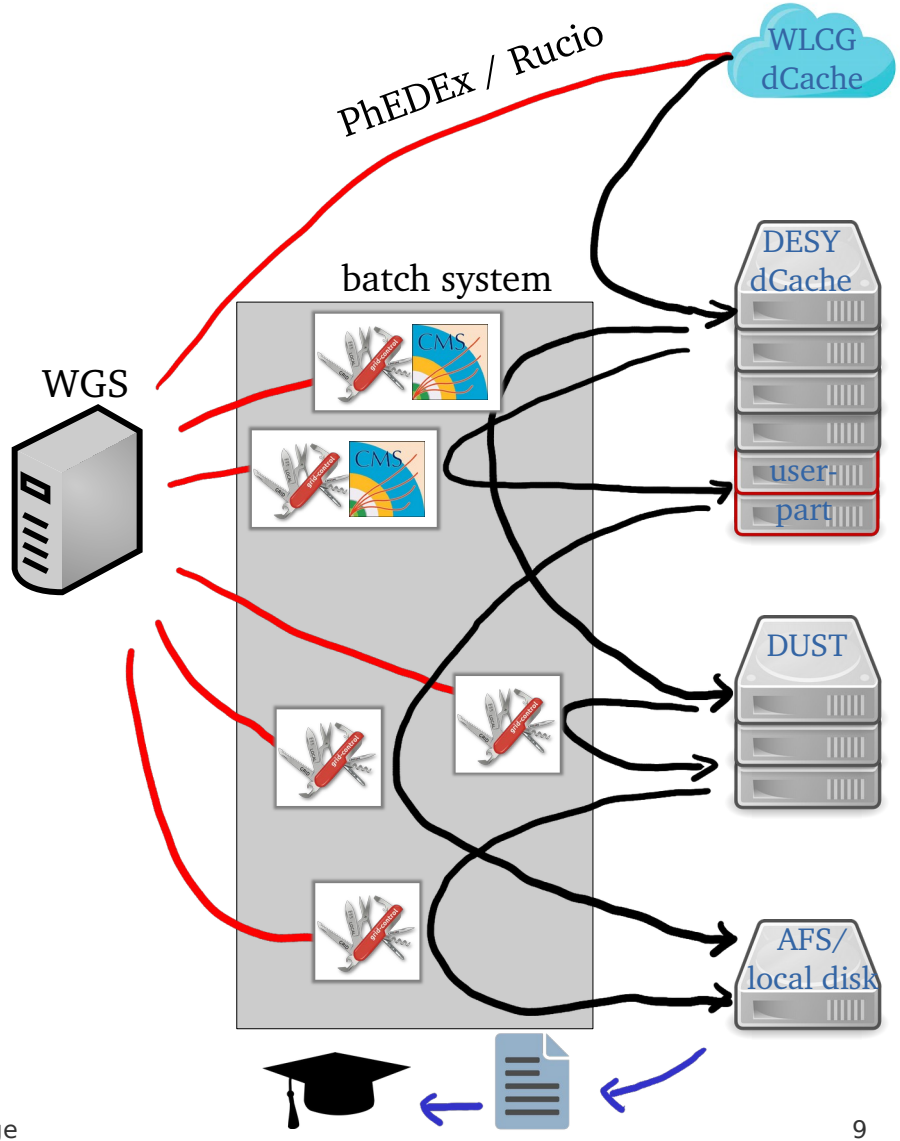
```
------------------------------------------------------------------------
Status report for task:                          demo / GC4884d5070fa7
------------------------------------------------------------------------
Total number of jobs:      200     Successful jobs:       22   11%
Jobs being processed:        1        Failing jobs:        0    0%

Jobs       INIT:     143   72%     Jobs    RUNNING:       1    1%
Jobs       DONE:      34   17%
------------------------------------------------------------------------
[#######00000000000                                                   ]
Waiting for 1 seconds...
2019-08-14 13:21:44 - Job 52  state changed from RUNNING to DONE
2019-08-14 13:21:44 - Job 53  state changed from RUNNING to DONE
2019-08-14 13:21:45 - Job 16  state changed from DONE to FAILED (WN None ID WMSID.HOST.389)
2019-08-14 13:21:45 - Job 32  state changed from DONE to SUCCESS (runtime 0h 00min 16sec)
2019-08-14 13:21:46 - Job 16  state changed from FAILED to SUBMITTED (retry #1)
2019-08-14 13:21:47 - Job 16  state changed from SUBMITTED to RUNNING
2019-08-14 13:21:48 - Job 3   state changed from DONE to SUCCESS (runtime 0h 00min 02sec)
2019-08-14 13:21:48 - Job 29  state changed from DONE to SUCCESS (runtime 0h 00min 08sec)
2019-08-14 13:21:49 - Job 56  state changed from INIT to SUBMITTED
2019-08-14 13:21:50 - Job 56  state changed from SUBMITTED to RUNNING
2019-08-14 13:21:50 - Job 11  state changed from RUNNING to DONE
2019-08-14 13:21:51 - Job 22  state changed from DONE to FAILED (WN None ID WMSID.HOST.3372)
2019-08-14 13:21:51 - Job 50  state changed from DONE to SUCCESS (runtime 0h 00min 10sec)
2019-08-14 13:21:52 - Job 22  state changed from FAILED to SUBMITTED (retry #1)
2019-08-14 13:21:53 - Job 22  state changed from SUBMITTED to RUNNING
2019-08-14 13:21:53 - Job 55  state changed from RUNNING to DONE
2019-08-14 13:21:53 - Job 16  state changed from RUNNING to DONE
2019-08-14 13:21:53 - Job 56  state changed from RUNNING to DONE
```

Example analysis

Workflow

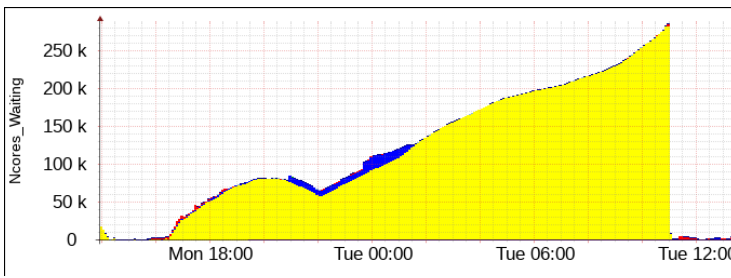
- CMS-central datasets: transferred to DESY dCache (formerly PhEDEx, now rucio)
- CMSSW-jobs (requiring full software stack)
  - write custom n-tuples to DUST
  - for background studies, write (large) n-tuples to dCache
- own C++/Python programs: do not depend on full CMSSW-stack, use n-tuples
  - pseudo-experiments, statistical procedures
  - background studies
  - plotting etc. (partially on WGS)



Some takeaways, pitfalls

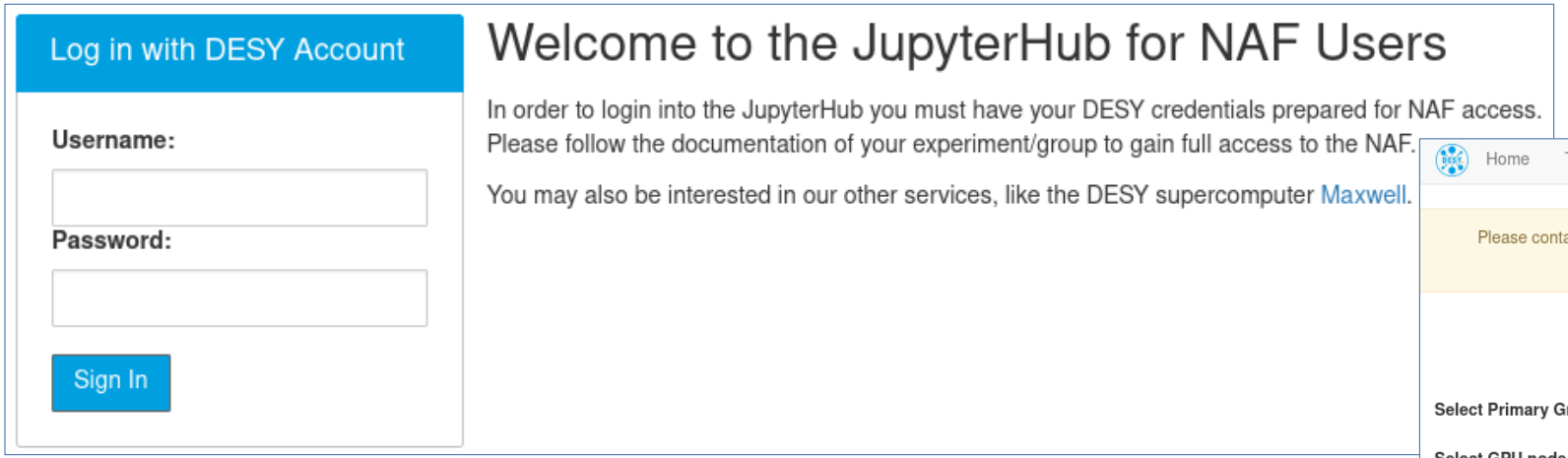
- having data locally, mounted via NFS can be very convenient
  - (bachelor) students do not need CERN account, grid certificate
- for long-running CMSSW-jobs one should consider submitting to the grid (crab)
  - they “hurt” your priority in the batch system
    → more lightweight jobs down the line take longer to start
- sometimes users kill the HTCondor schedd by submitting too many jobs at the same time (becomes unresponsive with O(100k))
  - in grid-control easily controlled:

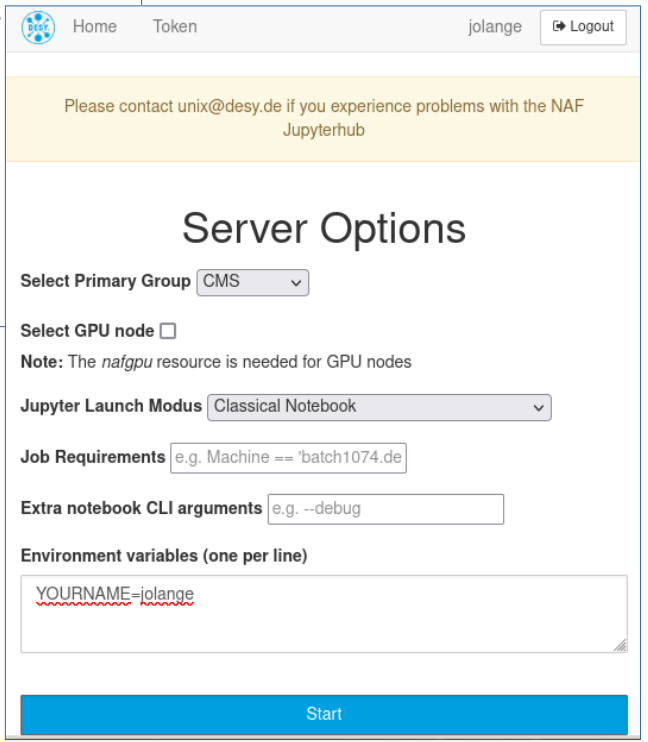
```
[jobs]
in queue = 1000
```



- we have different schedd nodes (1 ATLAS, 1 CMS, …)
  → not everybody is directly affected

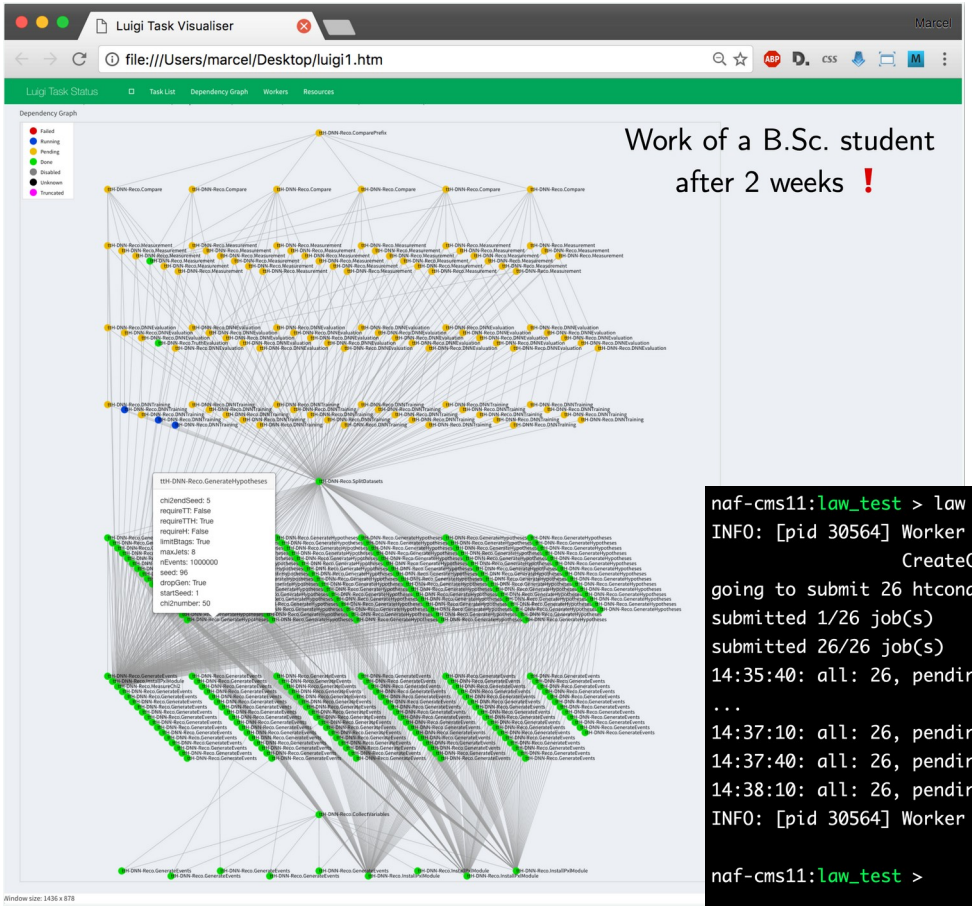
JupyterHub @ NAF





- notebook instances spawned as HTCondor jobs
- useful for onboarding new students
  - esp. true during the pandemic

Going more complex: law

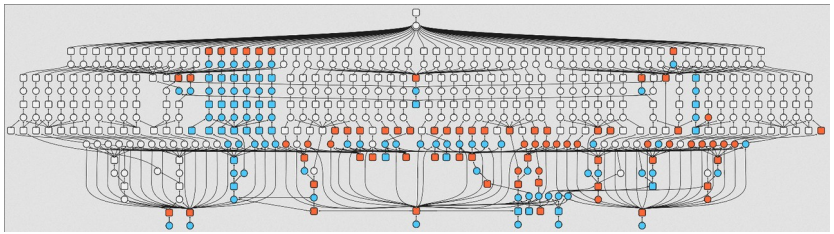


- based on *luigi* pipelining framework
- used by an increasing number of groups
- using HTCondor backend @NAF

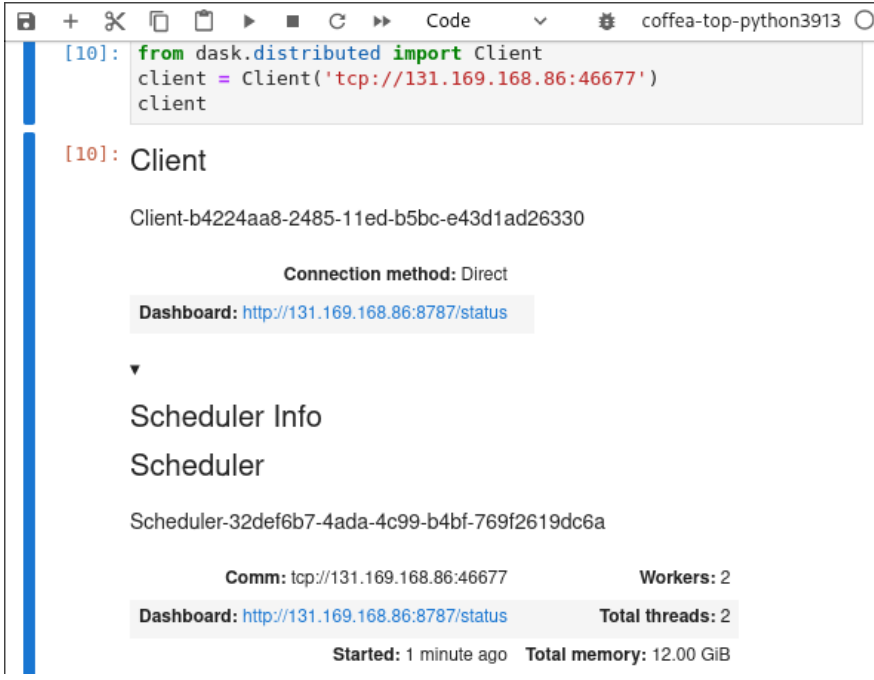
```
naf-cms11:law_test > law run CreateChars --workflow htcondor
INFO: [pid 30564] Worker Worker(host=naf-cms11.desy.de, username=riegerma) running
                  CreateChars(branch=-1, start_branch=0, end_branch=26, version=v1)
going to submit 26 htcondor job(s)
submitted 1/26 job(s)
submitted 26/26 job(s)
14:35:40: all: 26, pending: 26 (+26), running: 0 (+0),   finished: 0 (+0),   retry: 0 (+0), failed: 0 (+0)
...
14:37:10: all: 26, pending: 0 (+0),   running: 26 (+26), finished: 0 (+0),   retry: 0 (+0), failed: 0 (+0)
14:37:40: all: 26, pending: 0 (+0),   running: 10 (-16), finished: 16 (+16), retry: 0 (+0), failed: 0 (+0)
14:38:10: all: 26, pending: 0 (+0),   running: 0  (+0),  finished: 26 (+10), retry: 0 (+0), failed: 0 (+0)
INFO: [pid 30564] Worker Worker(host=naf-cms11.desy.de, username=riegerma) done!

naf-cms11:law_test >
```

dask-jobqueue



- attempt to make dask-jobqueue conveniently usable @NAF
- more interactive support of columnar analysis workflows
- spawn workers in **existing HTCondor infrastructure**
- very first steps
  - works on WGS with venv, conda, mamba, …
  - can connect to started client from JupyterHub@NAF
- to be done
  - make it usable direcly from JupyterHub@NAF (not configured for job-submission, yet)
  - monitor batch system usage to see if a special treatment (priority) for these jobs is needed



Summary

- NAF is vital for German CMS analyzers
  - for many, grid jobs are not even necessary
- very different workflows possible
  - different analyses with different needs
  - different tools
- batch system is the work horse
  - backend to most tools

Thanks for your attention!