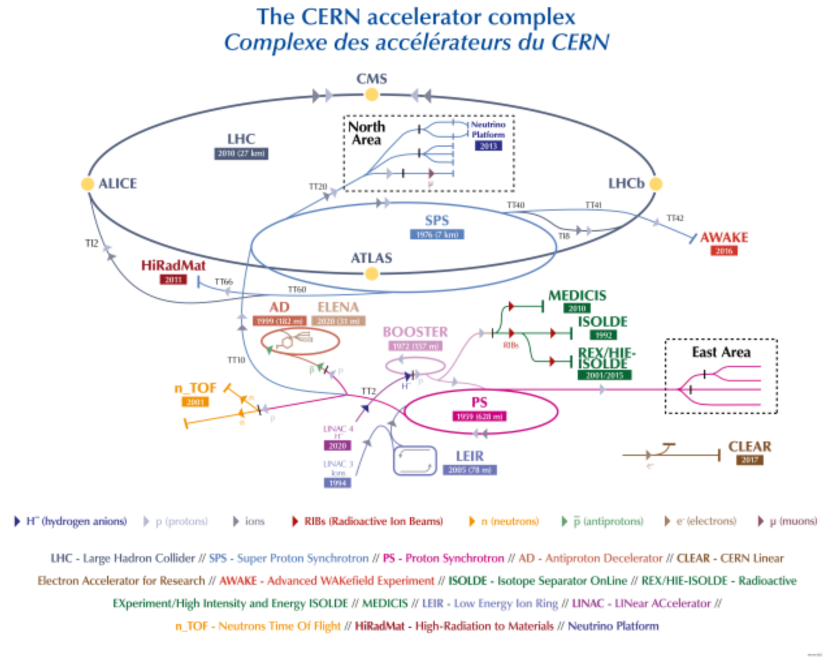
CERN

HTCondor @ CERN

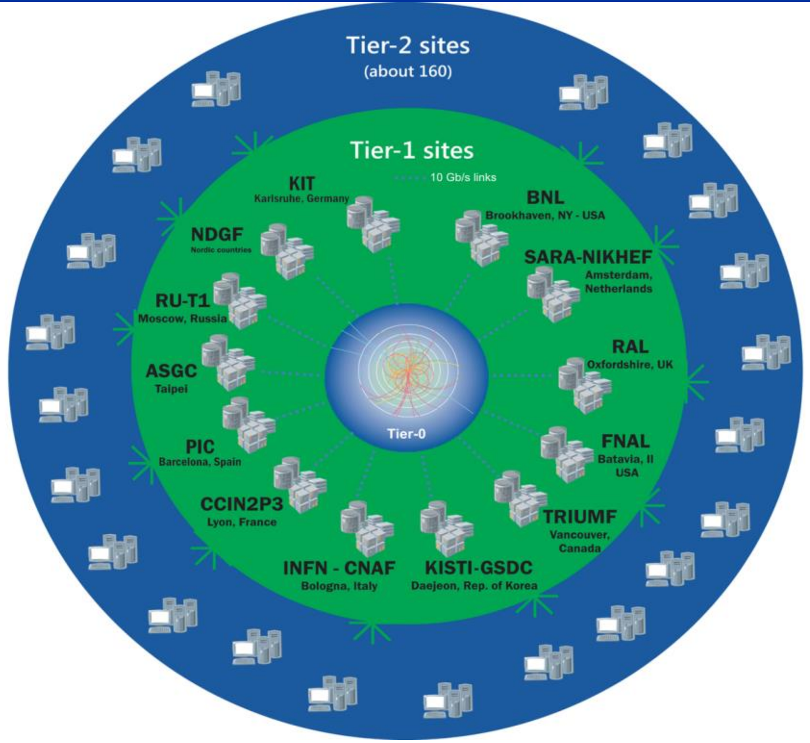
Update & Challenges

CERN Batch System


The CERN accelerator complex
Complexe des accélérateurs du CERN
CMS
LHC
2010 (27 km)
North Area
Neutrino Platform
2013
ALICE
LHCb
TT20
TT40
TT41
TT42
SPS
1976 (7 km)
TI8
TI2
AWAKE
2016
HiRadMat
2011
TT66
ATLAS
TT60
AD
1999 (182 m)
ELENA
2020 (31 m)
MEDICIS
2010
ISOLDE
1992
BOOSTER
1972 (157 m)
RIBs
REX/HIE-ISOLDE
2001/2015
East Area
TT10
n_TOF
2001
TT2
PS
1959 (628 m)
LINAC 4
2020
LINAC 3
Ions
1994
LEIR
2005 (78 m)
CLEAR
2017
H⁻ (hydrogen anions)
p (protons)
ions
RIBs (Radioactive Ion Beams)
n (neutrons)
p̄ (antiprotons)
e⁻ (electrons)
μ (muons)
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE-ISOLDE - Radioactive EXperiment/High Intensity and Energy ISOLDE // MEDICIS // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials // Neutrino Platform

Local Production

The WLCG



User analysis

Batch stats

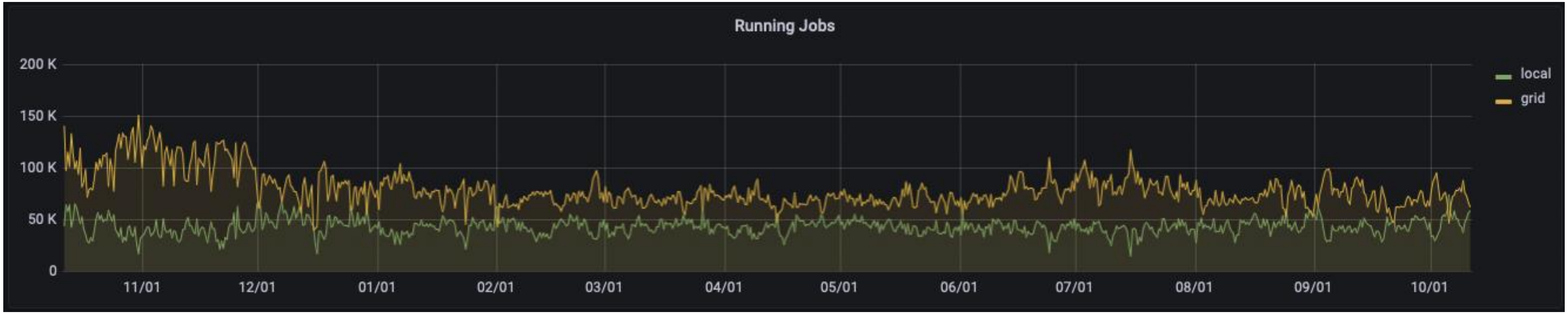
- 265k cores (3.4MHS06)
- ~2 Condor pools
- 5k worker nodes (EPs?)
- 20 Local schedds / 22 Grid CEs
- Infra (+/-) 9.0.17 / workers -> 9.0.17
- 300-350 unique daily submitters
- Compute capacity stable, number of workers quartered (v2p)

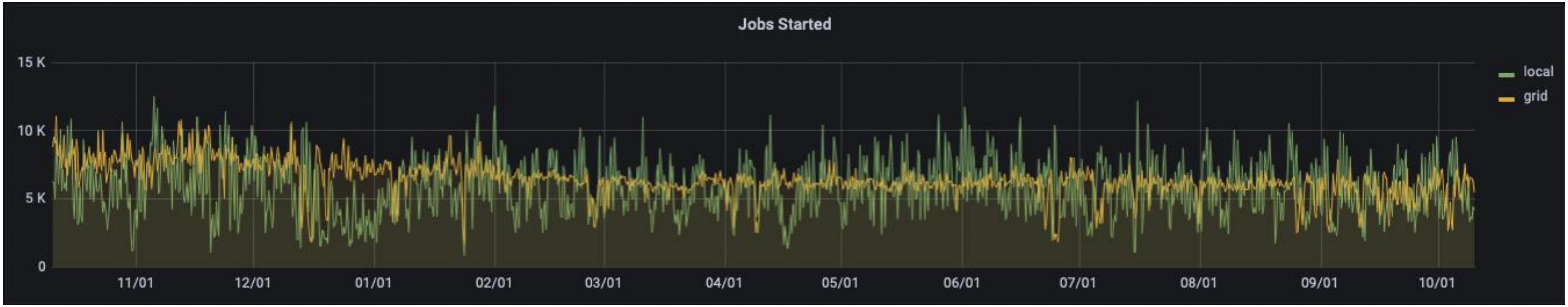
CERN Data Centre in Prévessin
Turning data into knowledge
Centre de données du CERN à Prévessin
Des données à la connaissance

Upcoming activity

- EL9 for next platform
- ARM (some, maybe)
- Move pool auth from GSI (probably to kerberos)
- CEs to token submission only
- HTCSS 10

Local v Grid

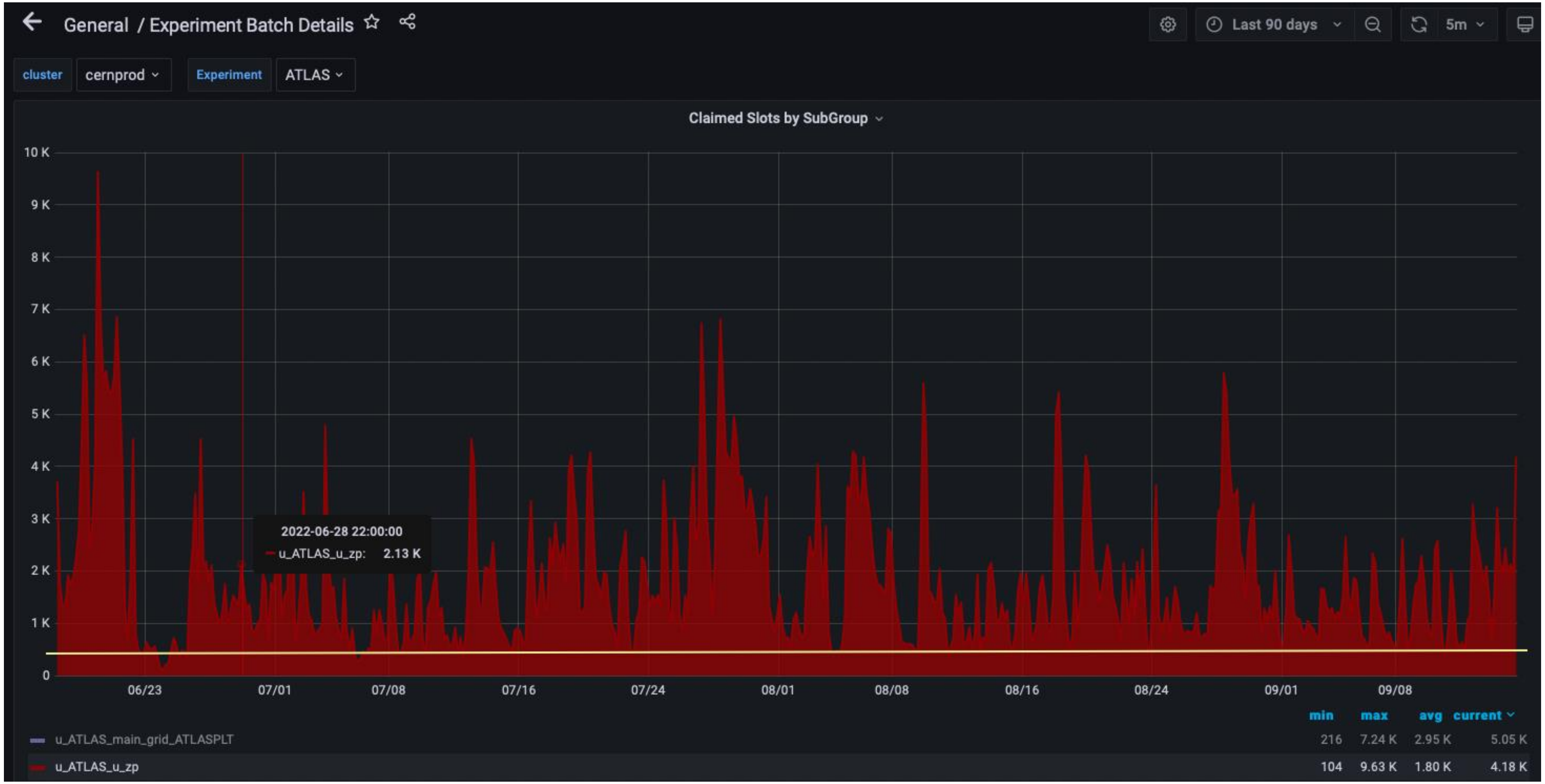

Running Jobs
200 K
150 K
100 K
50 K
0
11/01
12/01
01/01
02/01
03/01
04/01
05/01
06/01
07/01
08/01
09/01
10/01
local
grid


Jobs Started
15 K
10 K
5 K
0
11/01
12/01
01/01
02/01
03/01
04/01
05/01
06/01
07/01
08/01
09/01
10/01
local
grid

”Why do my jobs take so long to schedule? Last week I got 1000s of cores, this week none. How can we use the system more efficiently?”

User Story…

Inefficient?



General / Experiment Batch Details

cluster: cernprod
Experiment: ATLAS

Last 90 days · 5m

Claimed Slots by SubGroup

2022-06-28 22:00:00
u_ATLAS_u_zp: 2.13 K

	min	max	avg	current
u_ATLAS_main_grid_ATLASPLT	216	7.24 K	2.95 K	5.05 K
u_ATLAS_u_zp	104	9.63 K	1.80 K	4.18 K

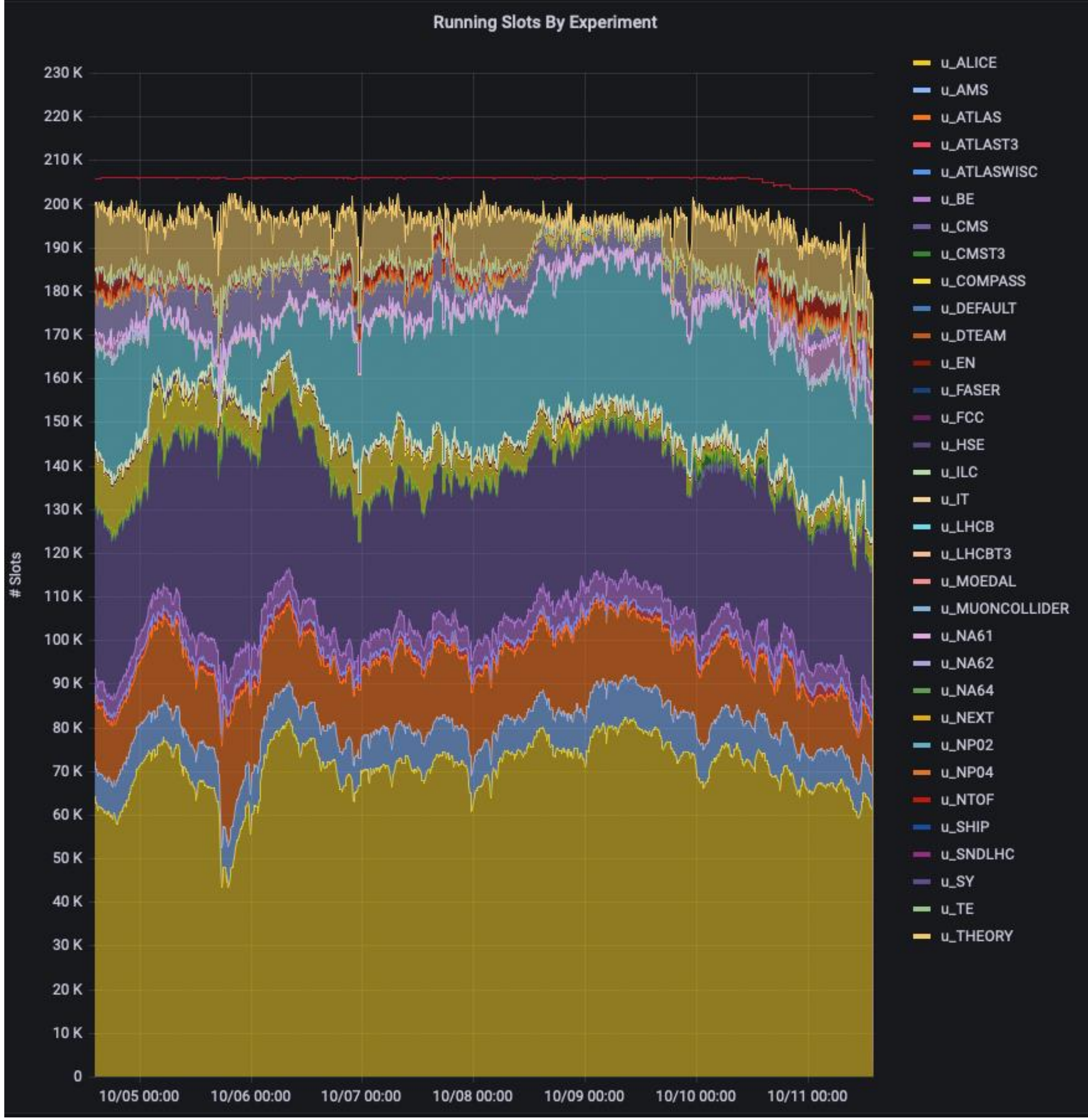
Effective Quota

User Challenges

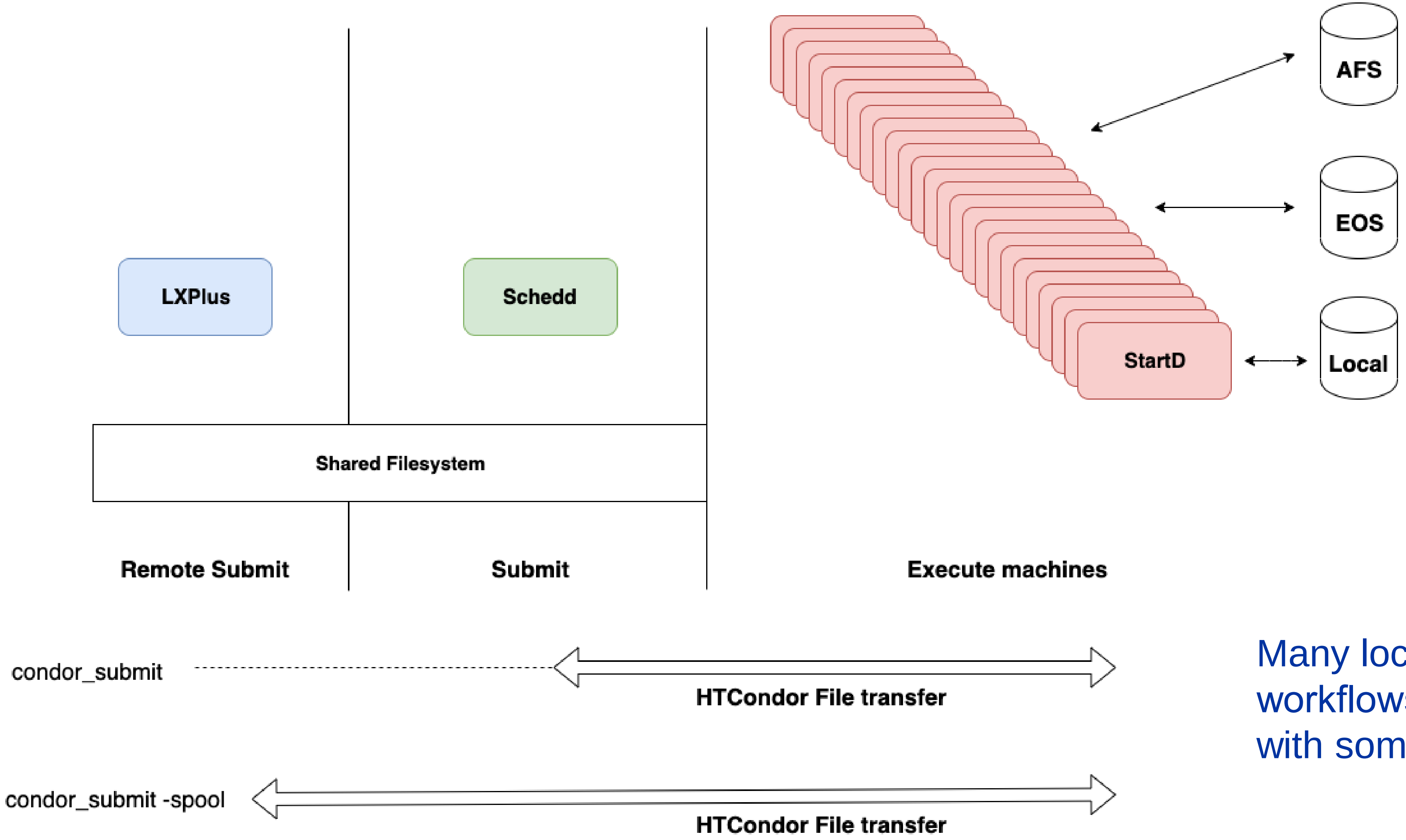
Question does illustrate some of the challenges in running our batch system

- Most of capacity is intended for “production”
- User submitted jobs with low nominal quota can acquire significant bursts of surplus
- Interactivity / responsiveness expectations different with ”users” vs ”production”
- User jobs more likely to have issues with sudden bursts of scale
- More support requirement for users v ”production” or grid.
- User analysis is 10-20% of jobs, but 80% of support overhead*

*Yes of course I made up this stat

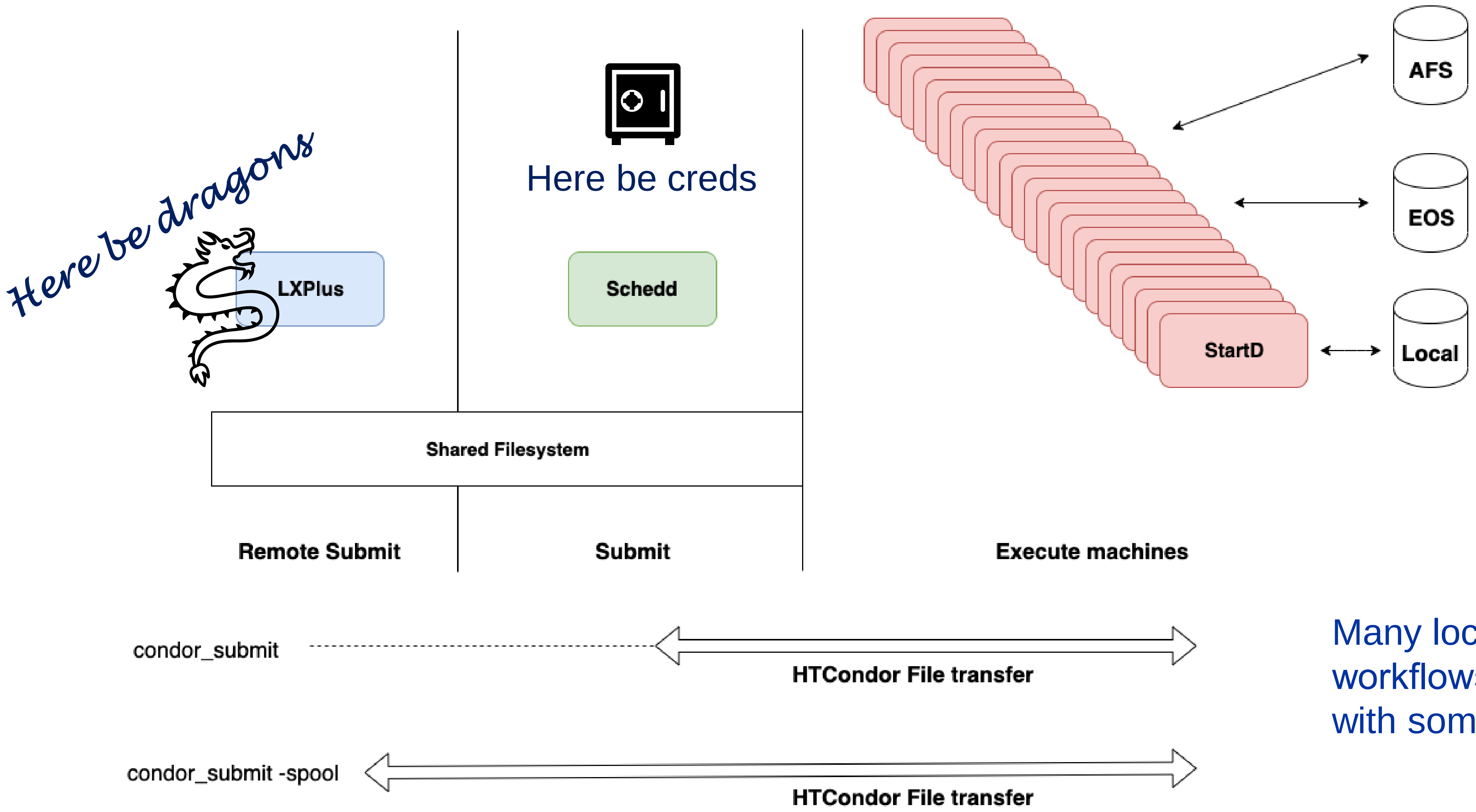


Remote submission with shared filesystem



Many local production workflows “spool”, along with some bigger users.

Remote submission with shared filesystem



Many local production workflows “spool”, along with some bigger users.

Basic user workflow

- ssh lxplus.cern.ch
- vim supervisors_file.sub
- condor_submit supervisors_file.sub
- [… wait …]
- Results appears as if by magic in pwd!

To make that “I type a command here , the data comes back here” interface work, we use shared filesystems

- HTCondor devs hate shared filesystems
- Batch service managers aren’t massively fond of shared filesystems
- People who run the shared filesystesm aren’t overjoyed about batch systems using their filesystems either
- It’s reallly really easy for users to understand*

* until it scales a bit more than they’d anticipated

SHARED FILESYSTEMS SUCK
SO YOU'RE NOT GOING TO USE THEM
YOU'RE NOT GOING TO USE THEM, RIGHT?
imgflip.com

File transfer plugins…

- **xrdcp plugin for EOS**
- **+/- replicate the simple submission interface**

```
executable = script.sh
log = xfer.$(ClusterId).log
error = yf.$(ClusterId).$(ProcId).err
output = yf.$(ClusterId).$(ProcId).out
output_destination = root://eosuser.cern.ch//eos/user/b/bejones/condor/xfer/$(ClusterId)/
transfer_input_files = root://eosuser.cern.ch//eos/user/b/bejones/condor/file.txt
MY.XRDCP_CREATE_DIR = True
queue
```

Some plugin hacks

- HTCondor makes it very easy to work around road bumps
- Plugin originally transferred contents of sandbox, but Out/Err named _condor_std*
- Now plugin just inspects .job.ad to name files correctly
- When using –spool condor_transfer_data has issues unless we fiddle with the attribute
- With remote submission UserLog still a problem

```
JOB_TRANSFORM_OutputDest @=end
    NAME OutputDest
    REQUIREMENTS (jobUniverse =?= 5 &&
!isUndefined(OutputDestination))
    COPY Out SubmittedOut
    COPY Err SubmittedErr
    COPY OutputDestination
SubmittedOutputDestination
    SET OutputDestination ifThenElse(JobStatus ==
4, undefined, SubmittedOutputDestination)
    SET Out ifThenElse(JobStatus == 4,
"/dev/null", SubmittedOut)
    SET Err ifThenElse(JobStatus == 4,
"/dev/null", SubmittedErr)
@end
```

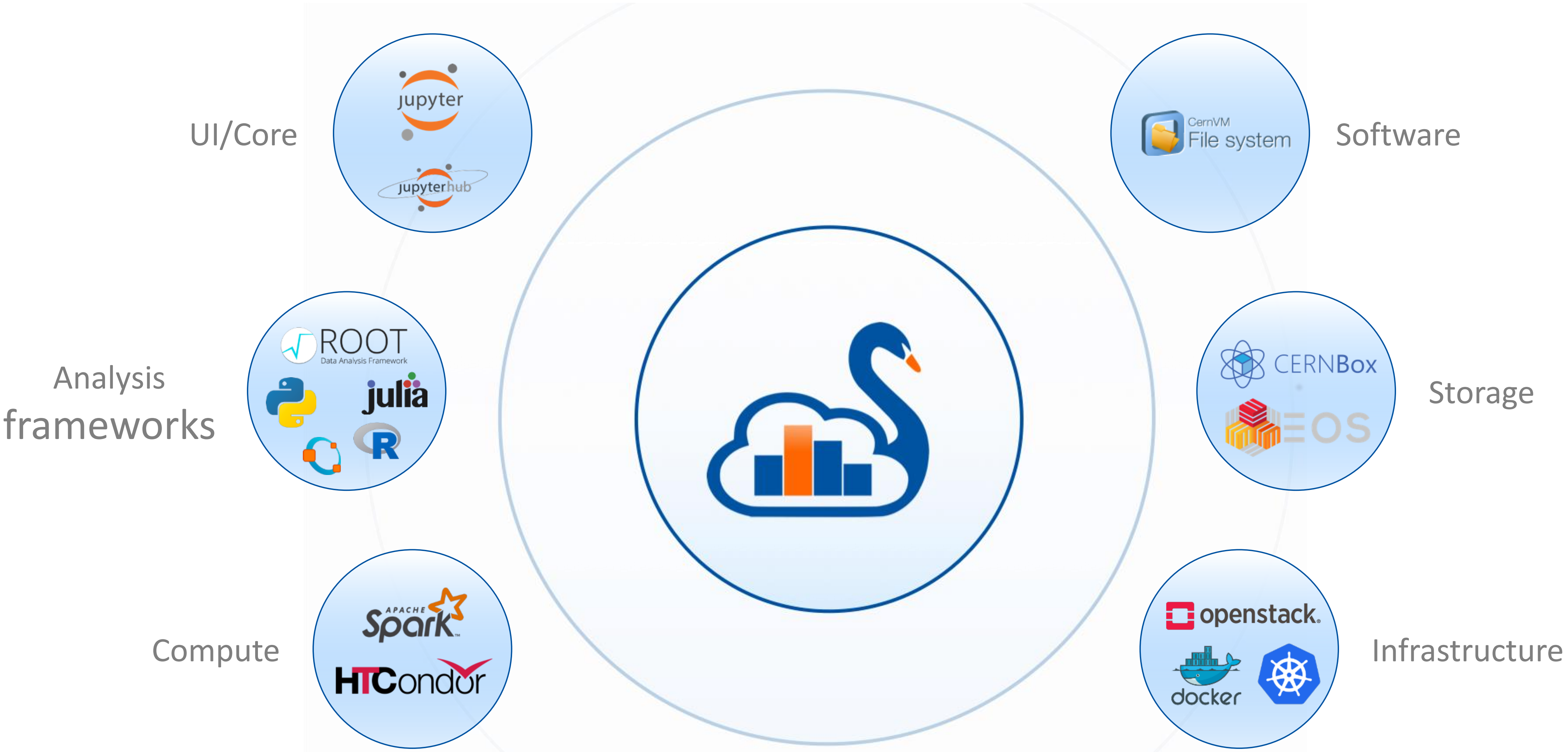
Less traditional entry points…

- Increasing interest around "analysis facilities" and metaschedulers
- Most interest around Dask, but users often pick / develop other projects
- With Dask we found that we (also!) needed to wrap/subclass it to avoid some assumptions + add some policy
- May help break some of the dependencies on filesystems, but makes some interactivity questions more difficult
- Dask (+ coffea) used from CLI via plus, but trying to improve with notebooks

SWAN: the interface

- **SWAN: Service for Web-based Analysis**
- **CERN’s Jupyter notebook service**
  - Created in 2016
  - Managed jointly by EP and IT
  - Used by 200-250 people daily
- **Jupyter interface + federation of CERN services → added value!**
  - Software (CVMFS)
  - Storage (EOS, CERNBox)
  - Computing resources (GPU, Spark, HTCondor)
- **Platform for physics analysis: supports both *single-node* and *distributed* analysis**

SWAN’s building blocks



UI/Core

Analysis frameworks

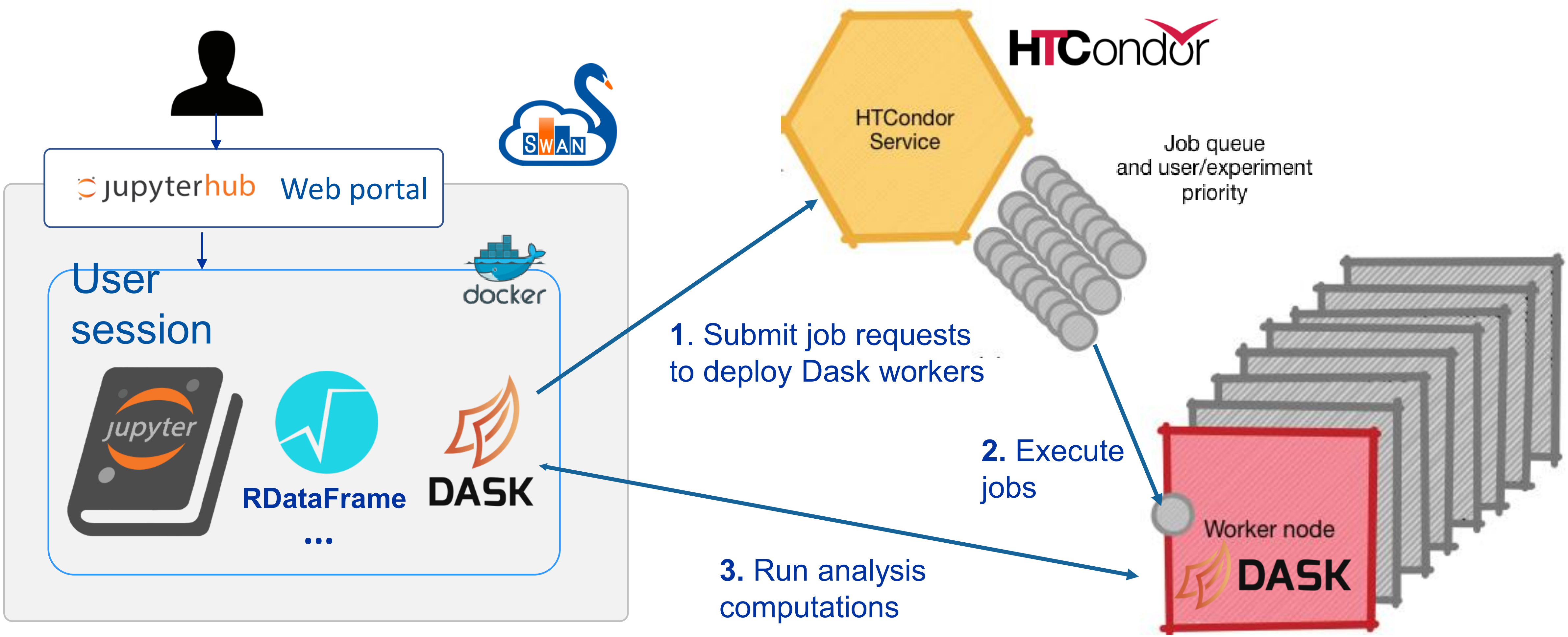
Compute

Software

Storage

Infrastructure

SWAN + HTCondor for interactive analysis



1. Submit job requests to deploy Dask workers

2. Execute jobs

3. Run analysis computations

SWAN / Dask integration

- **One issue early on resolved: needed Dask to set a contact_address for the workers to call back to a scheduler running on a k8s cluster**
  - Again: HTCondor transforms gave us flexibility to work around the problem till we had an upstream fix
- **CVMFS helps us to ensure that the Dask scheduler and the workers are running the same software**
  - Simplicity / cache layer
  - Dask / python env in /cvmfs/sft.cern.ch/lcg/… with setup for PYTHONPATH, LD_LIBRARY_PATH etc
  - One of the only times we've found a use for `getenv = True`
- **Still issues around interactivity. SWAN / Jupyter implies more interactivity, where many resources available to users are opportunistic**

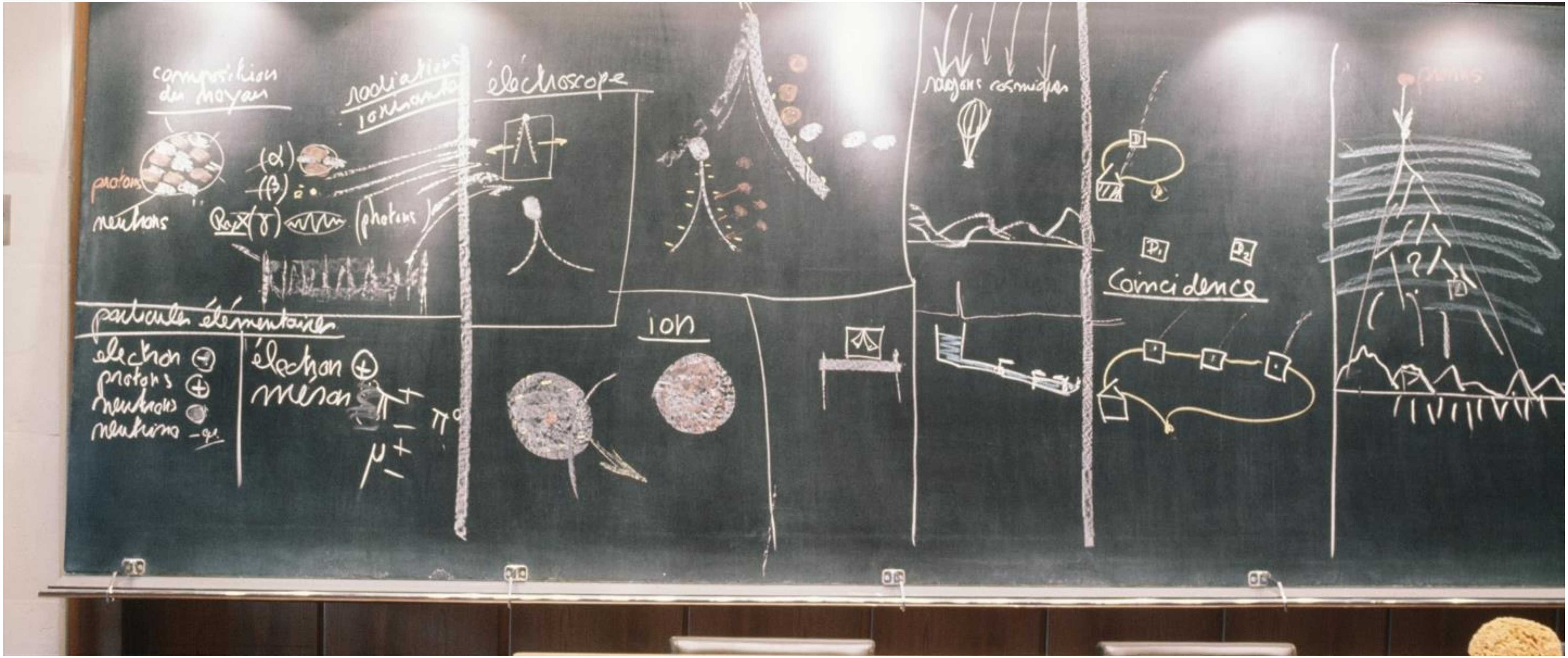
Interactivity / low latency

- **Current strategy is to reserve some resources for shorter jobs**
  - ie we use MaxRuntime (as now implemented as allowed_{job,execute}_duration)
  - Some machines will only accept jobs < $time
  - Ordinary users encouraged to submit shorter jobs
  - We have sometimes used separate negotiator for very small amounts of resources
- **Other ideas**
  - Dual Startds to allow for a small amount of slots that will take more interactive jobs?
  - Buffer partition with separate netgotiator with low ceiling per user?
  - ?
  - …
  - No, nobody wants to have jobs preempted

…just one thing about the schedds

- **Random wish list for users & schedds**
  - condor_now but for different schedd
  - condor_now but AccountingGroup based super-users for now-job / vacate-job
  - condor_move_job_to_other_schedd
    - I didn’t workshop the name
  - I/O on behalf of user to those nasty shared filesystems can still break schedd (condor_sos is very good though)

Questions?


composition du noyau
radiations ionisantes
électroscope
rayons cosmiques
protons
neutrons
(α)
(β)
RayX(γ)
(photons)
D
P1
P2
Coïncidence
particules élémentaires
électron ⊖
protons ⊕
neutrons
neutrino
électron ⊕
méson
π+
π0
μ+
ion