Status of the DESY-HH Clusters

News and lesser news

Christoph Beyer, Martin Flemming, Krunoslav Sever, Thomas Hartmann
Luca Gebhardt, Joja Meyn, Christian Voss
DESY IT

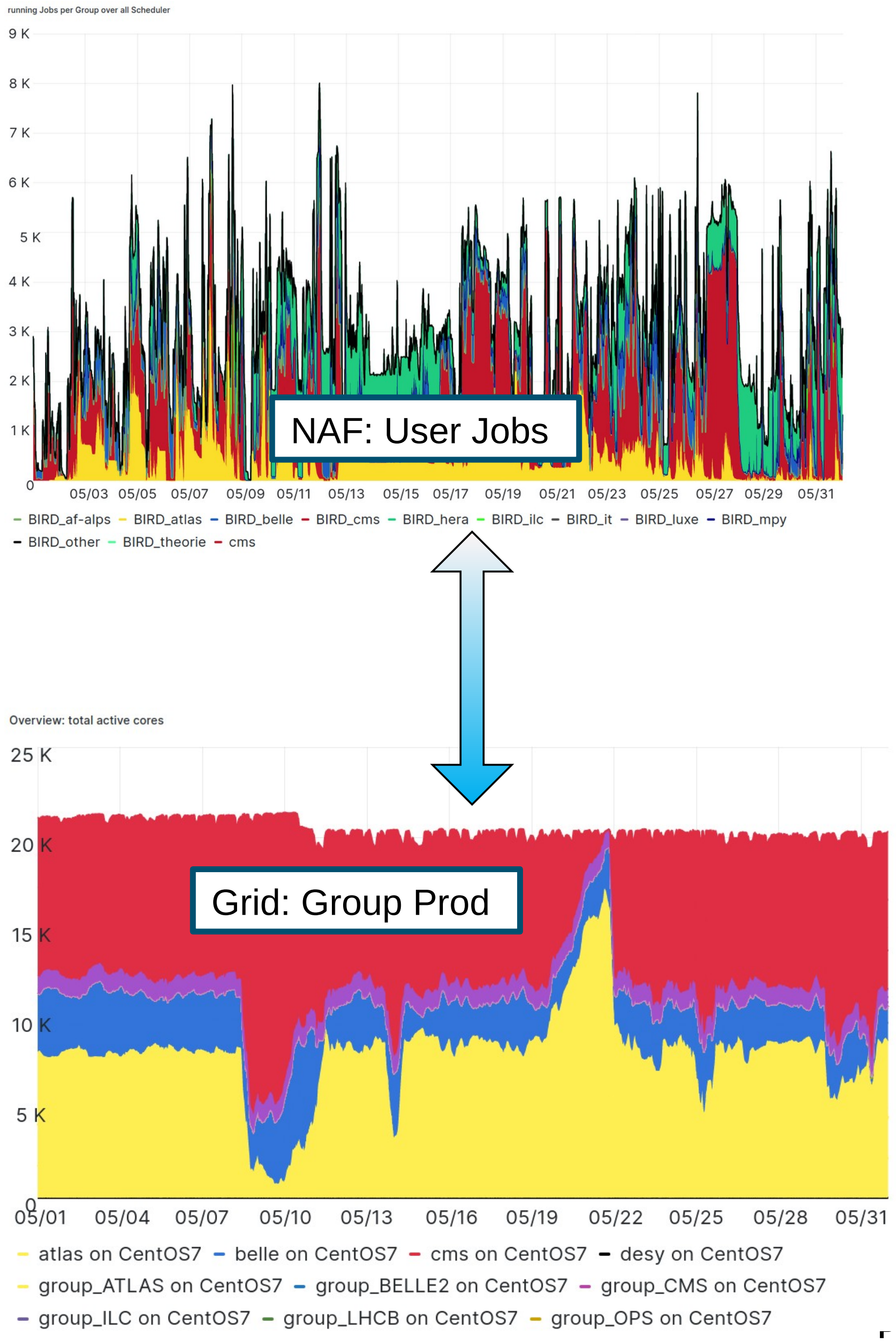
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES
DESY.

Reminder: NAF and Grid Clusters

HTC Clusters at DESY-HH

- 2 HTC clusters
  - User jobs: **N**ational **A**nalysis **F**acility
  - Group Production: Grid
  - Logical separated
  - Same code and admin base

- Differing workloads
  - Different energy saving options
  - Different ~~Problems~~ Challenges
  - Abandoned idea somewhat to unify both



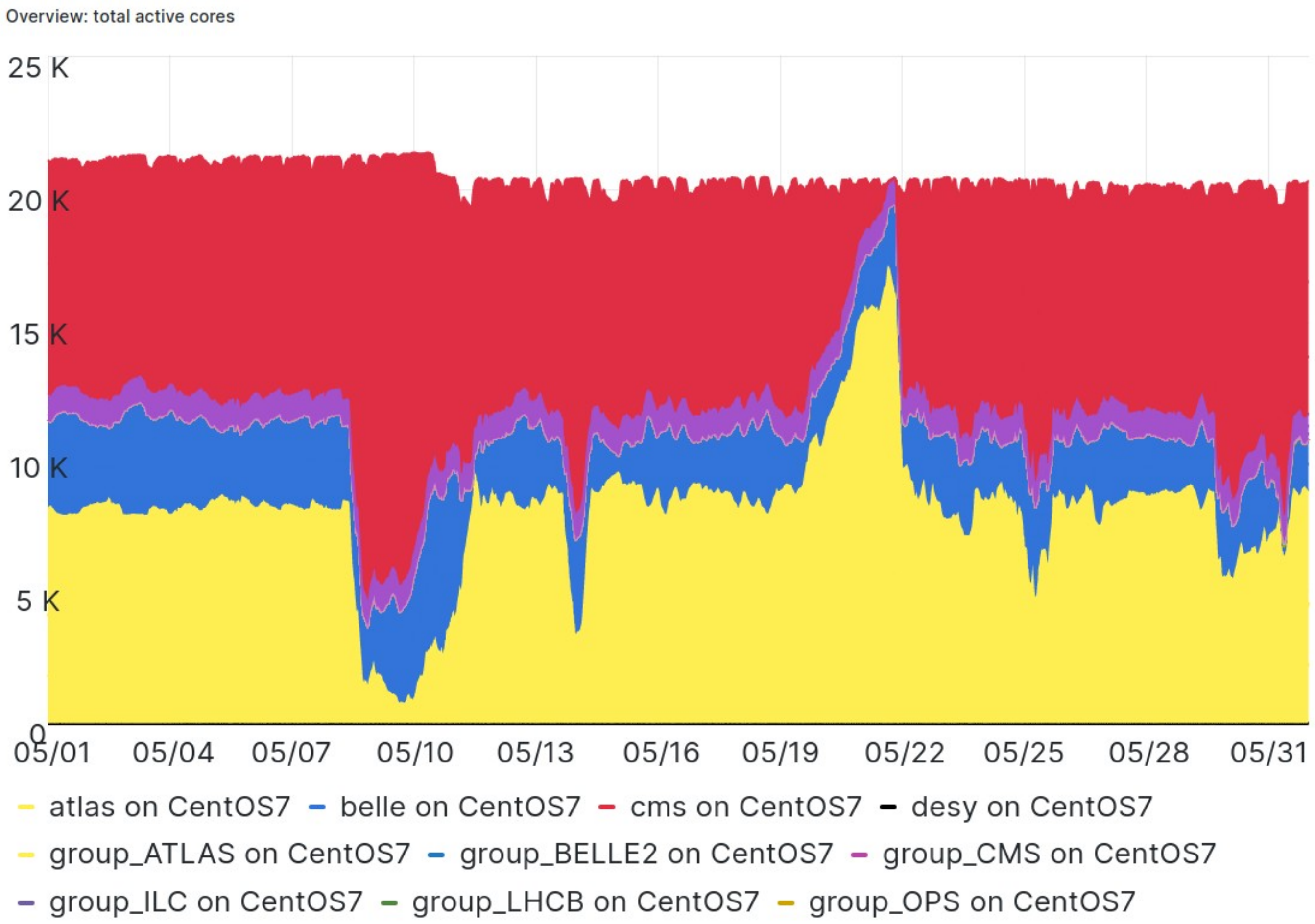
Grid Cluster at DESY-HH

HTCondor Cluster for HEP Communities

- Primarily HEP Groups
- Centralized pilot jobs
  - Group Production Payloads

- Goal: Full utilization 24/7/365
- Job start up latency not critical

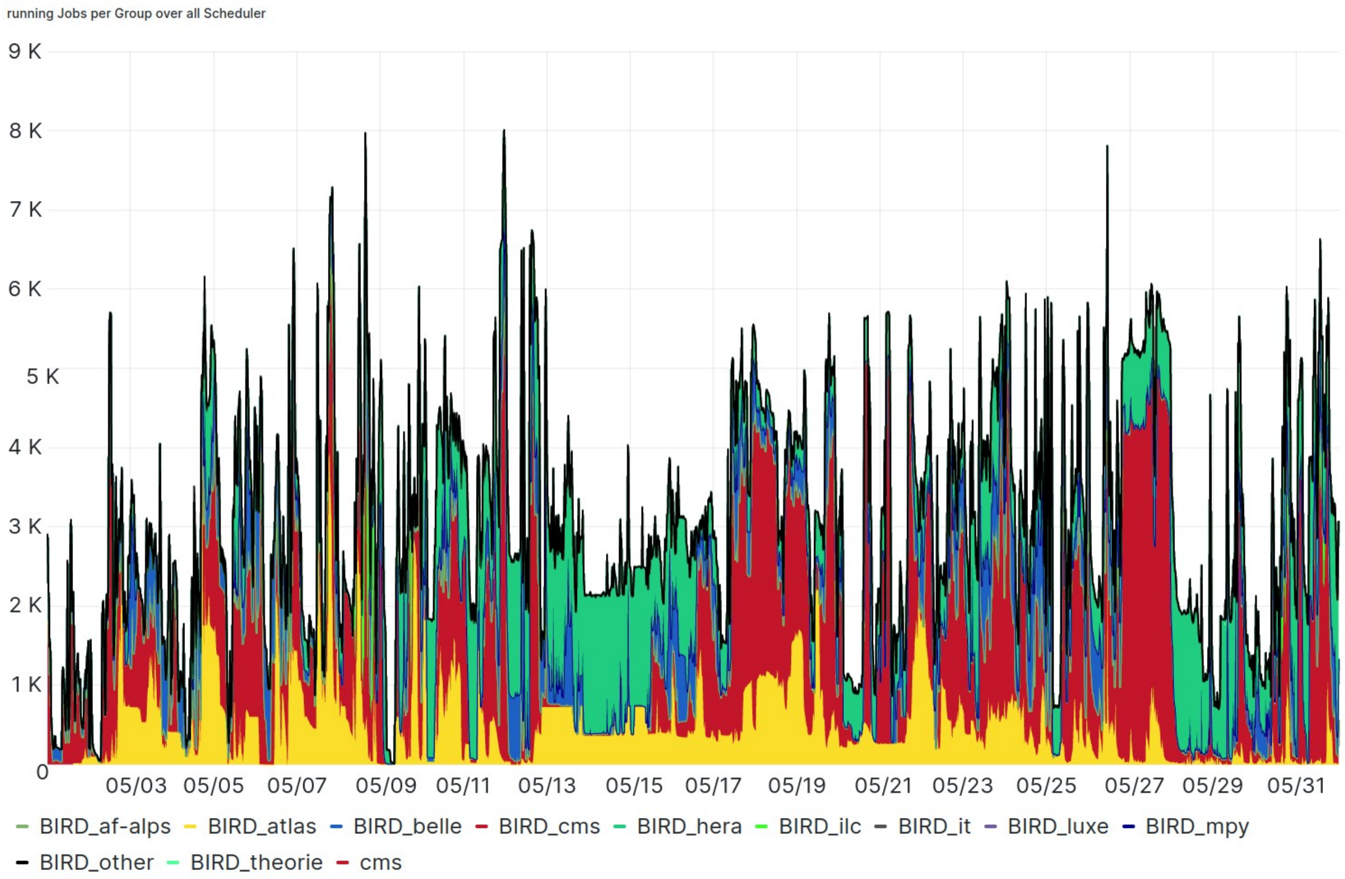
- Submission via HTCondorCE
- No shared FS



User Cluster: National Analysis Facility

HTCondor Cluster for German HEP Users

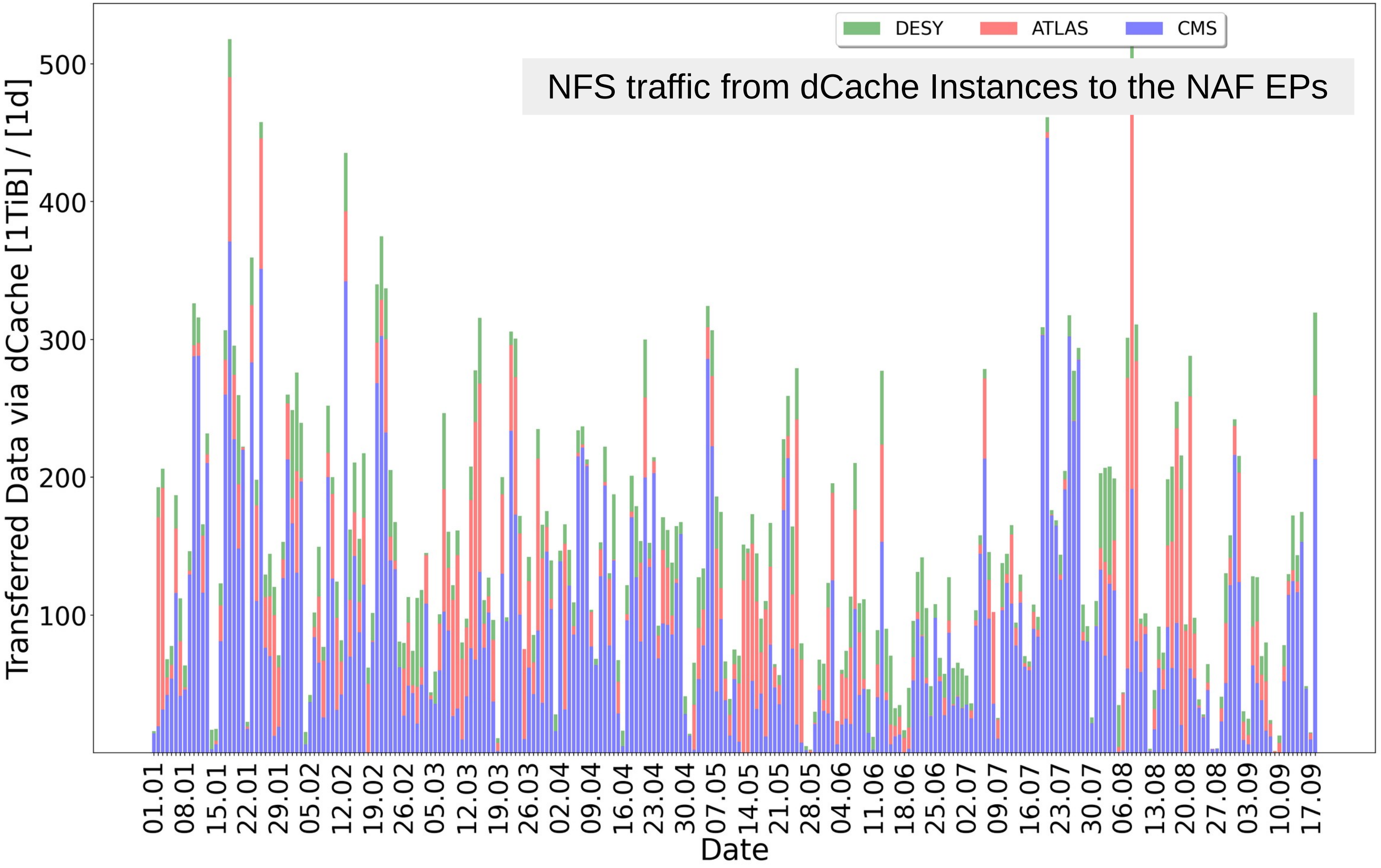
- Individual users
- Remote Submission via Workgroup Servers
  - Dedicated Scheduler & Token Renewal
- Utilization dynamic
- Overall utilization depends on work hours, holidays, ..., deadlines, conferences
- Job start up latency relevant for user satisfaction
- Shared FS's (AFS, dCache/NFS4, GPFS/NFS4)



User Cluster: National Analysis Facility

File I/O

- Users love paths/POSIX
- NFS protocol of choice
  - Access authz: POSIX user:group
  - dCache: long-term storage + Grid



Energy Consumption

Optimizing the Cluster Energy Profiles

Adapting to Green Energy and becoming more dynamic

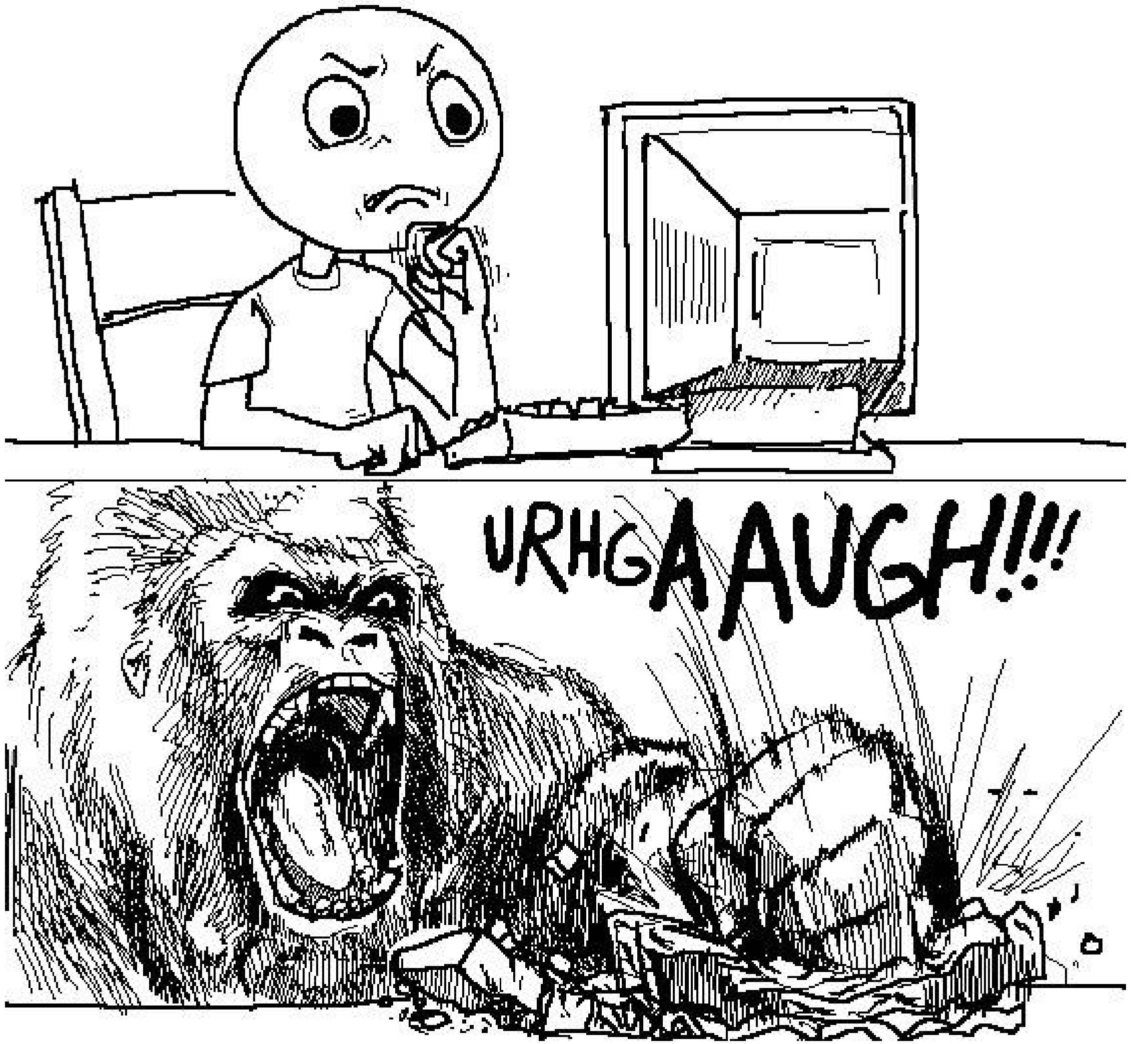
https://indico.cern.ch/event/1274213/contributions/5570403/

See Christoph’s Talk

https://indico.cern.ch/event/1274213/contributions/5570403/

Operating Systems

Changes in Linux Ecosystems



Changes in Linux Ecosystems

RHEL & Debian Flavours

- Production Clusters still on CentOS 7
- Had been preparing move to AlmaLinux 9
  - Had no trust in CentOS Stream and aimed for Alma as EL clone
  - Significant changes (again) to the RHEL flavoured niches
- Evaluating Ubuntu now as well
  - Need “Enterprise” OS for other systems - going for Ubuntu there
- Middleware/Accounting status beyond EL7 unclear

Cluster Plans

RHEL & Debian Flavours

- Initial plan was
  - Alma9, cgroups v2, Condor$_{\text{feature}}$ 10.X, CondorCE 6,...
  - Cluster sec to tokens
  - Fully embrace the new illustrious Grid/CE token world
  - Evaluating Grid Middleware/accounting alternatives
    - AUDITOR from Uni Freiburg

https://alu-schumacher.github.io/AUDITOR/

- **Lession for the long term**: separation HW OS from Middleware OS from User App OS

Monitoring

Embracing Job Events

Powerful Tool

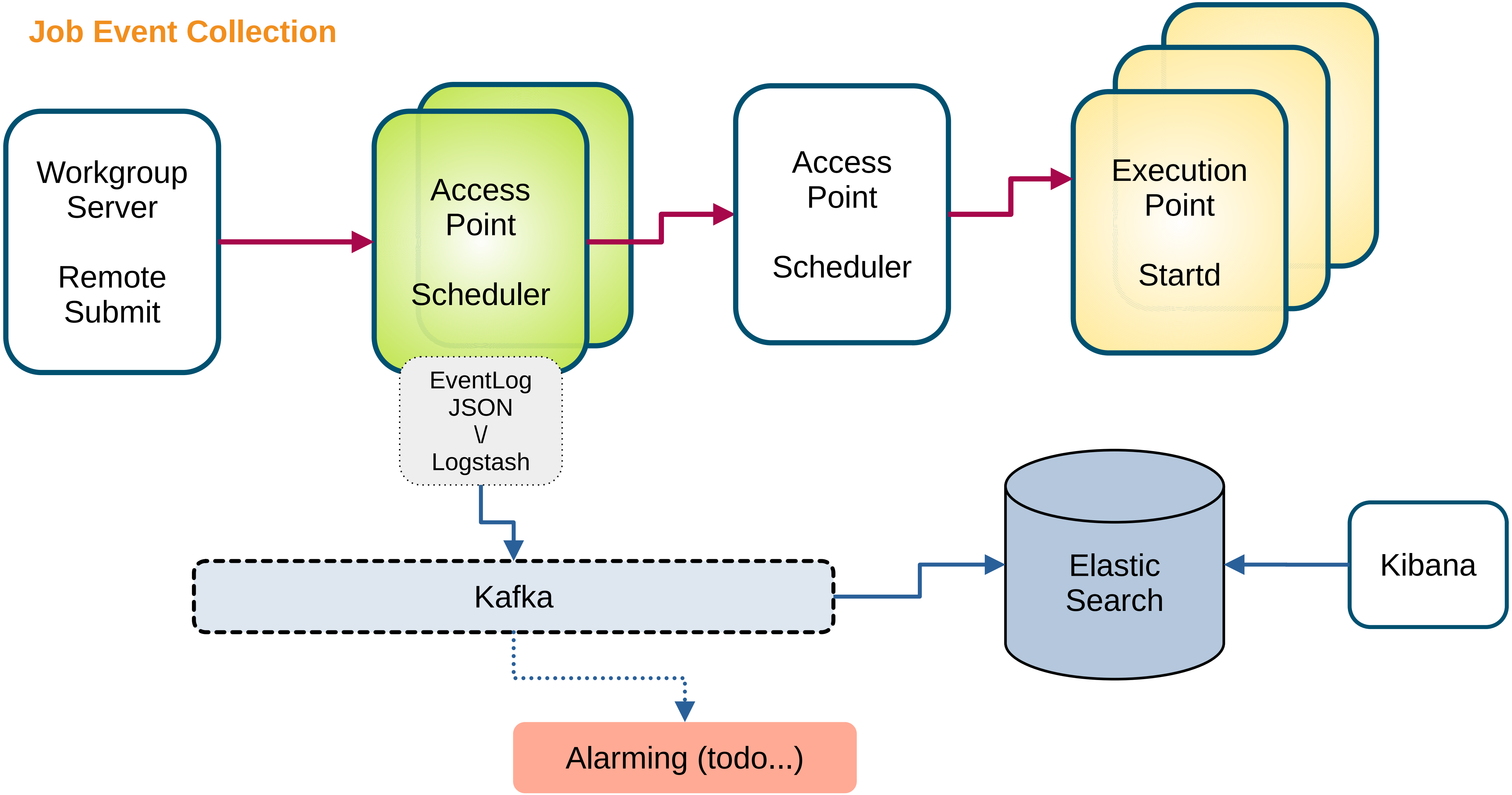
- Job events have become central to our cluster maintenance
  - (pull) time series nice – but (push) detailed job events powerful to understand the cluster
  - Who else is using job events?

- NAF users occassionally with workflows straining the systems
- Straight forward digging for users, jobs, starts, errors,…

- Currently student (Luca) working on interlooping with dCache storage events
- One lesseon: Synthetic emulation of storage killing DDOS jobs not really easy
  - HTCondor+dCache+GPFS *in principle* pretty stable
    (apparently users more *creative* then us admins)

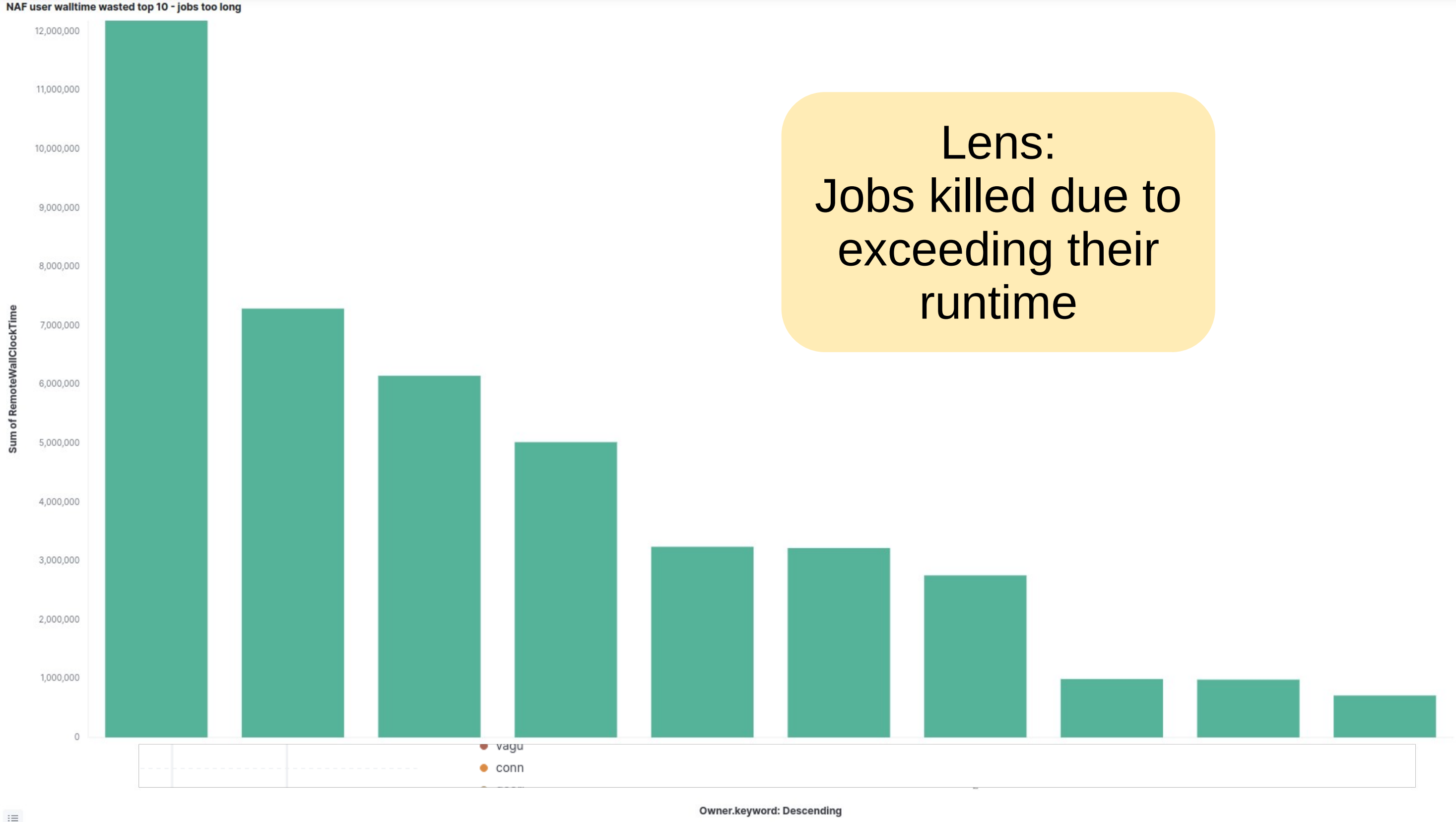
Event Flow

Job Event Collection

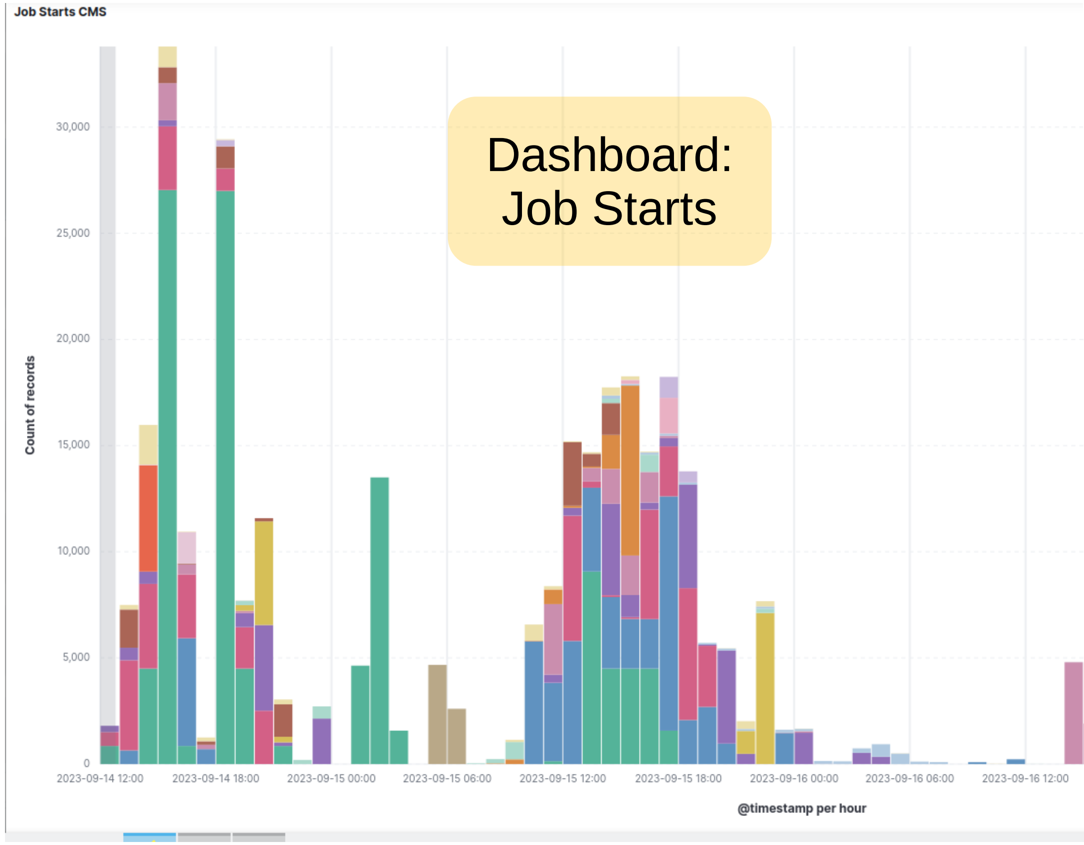


Day 2 Day Debugging

Dashboards & ad hoc lenses



Lens:
Jobs killed due to exceeding their runtime



Dashboard:
Job Starts

ToDo and Wishlist

Currently grok'ing Daemon Logs

- Machine readable daemon logs ok'ish
- JSON formatted Daemon events might be much easier to parse
- Could daemon logging **push** their "events" as JSONs?

- Aim: traingulate cluster issues to jobs events to storage events etc. pp.

- ToDo at us: integrate adstash job ads with the job events
  - Currently separate corners in ES

Opportunistic Resources

Overlay Batch System

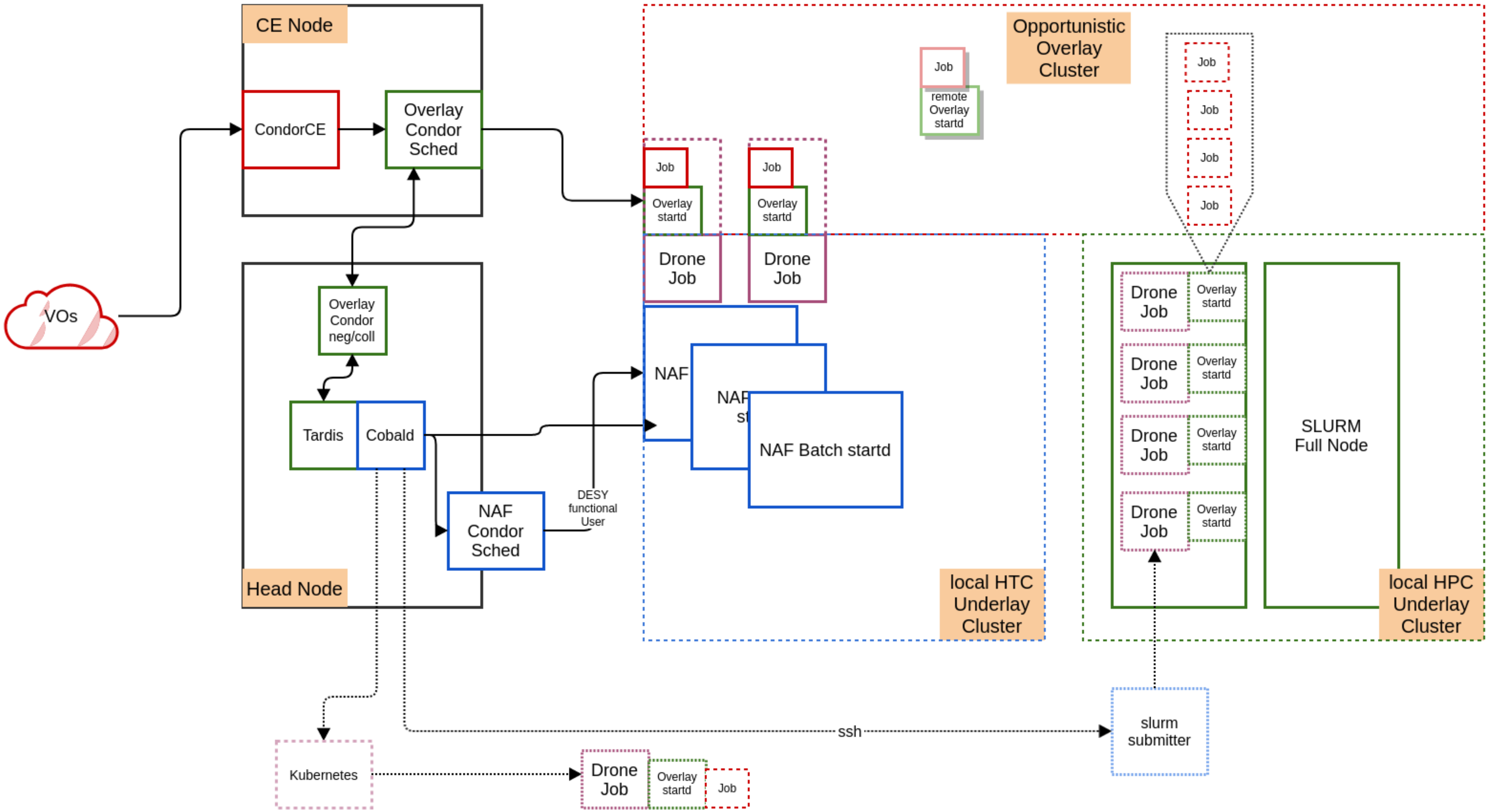
Backfilling resources

https://cobald-tardis.readthedocs.io/en/latest/

- Participating in PUNCH project
- Cobald/Tardis Overlay HTCondor Cluster
  - startd *drones* in local HTCondor slots (or SLURM, K8s,...)
  - DESY-HH contributing resources
- Long term idea/*proliferation*
  opportunistic utilization of all the untapped resources nodes in an overlay cluster

Opportunistic Resource Utilization

Dynamic Overlay Cluster ~ Breathing Scale up/down



~~Issues~~ Challenges

Jupyter Notebooks

Users becoming more memory hungry

- Jupter hub on the NAF
  - Notebook jobs via dedicated scheduler/negotiator
    - running on dedicated slots
  - Our idea: lightweigt notebooks for day to day work
    - Worked in the past pretty well
  - Users’ idea: load everything(tm) in memory for interactive stuff
    - It’s easy and compexity is hidden
  - Had been lenient enforcing mem limits
  - Killing (randomly from the user perspective) notebook jobs not well received...

Jupyter Notebooks

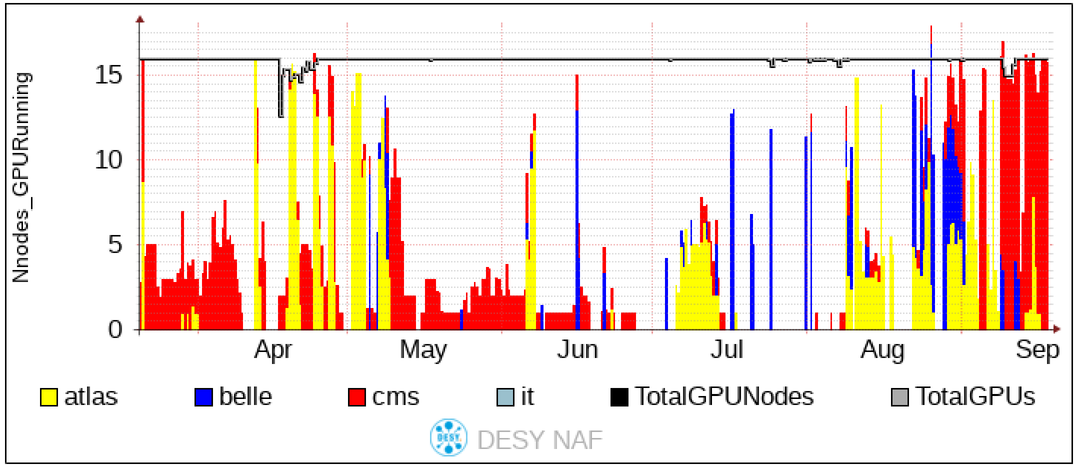
Scale out from Notebooks

- Going for scale out
- Apprentice (Joja) working on **htmap** and Apache **Dask**
  - Helper Python lib for users to import
    - Easy™ scale out functions/data ingress/egress (hopefully)
    - Spawn jobs onto the cluster from a notebook/job onto the cluster
  - Hiding NAF details
    - Token/ticket renewal
    - EP with notebook jobs becoming also remote submitters
    - Dedicated Scheduler (+ Negotiator?)
- To be seen how operations look like in the end: myriads of short jobs, I/O hammering,…?

https://htmap.readthedocs.io/en/latest/

GPU Resources

Brokering *special* Resources



- Demand for GPU nodes quite variable
  - User complaints mounted up
  - GPU nodes no special resource in the pool
    - No concurrency limit - just nodes with GPU resource to be requested
    - User quota in one bag as with all non-GPU nodes

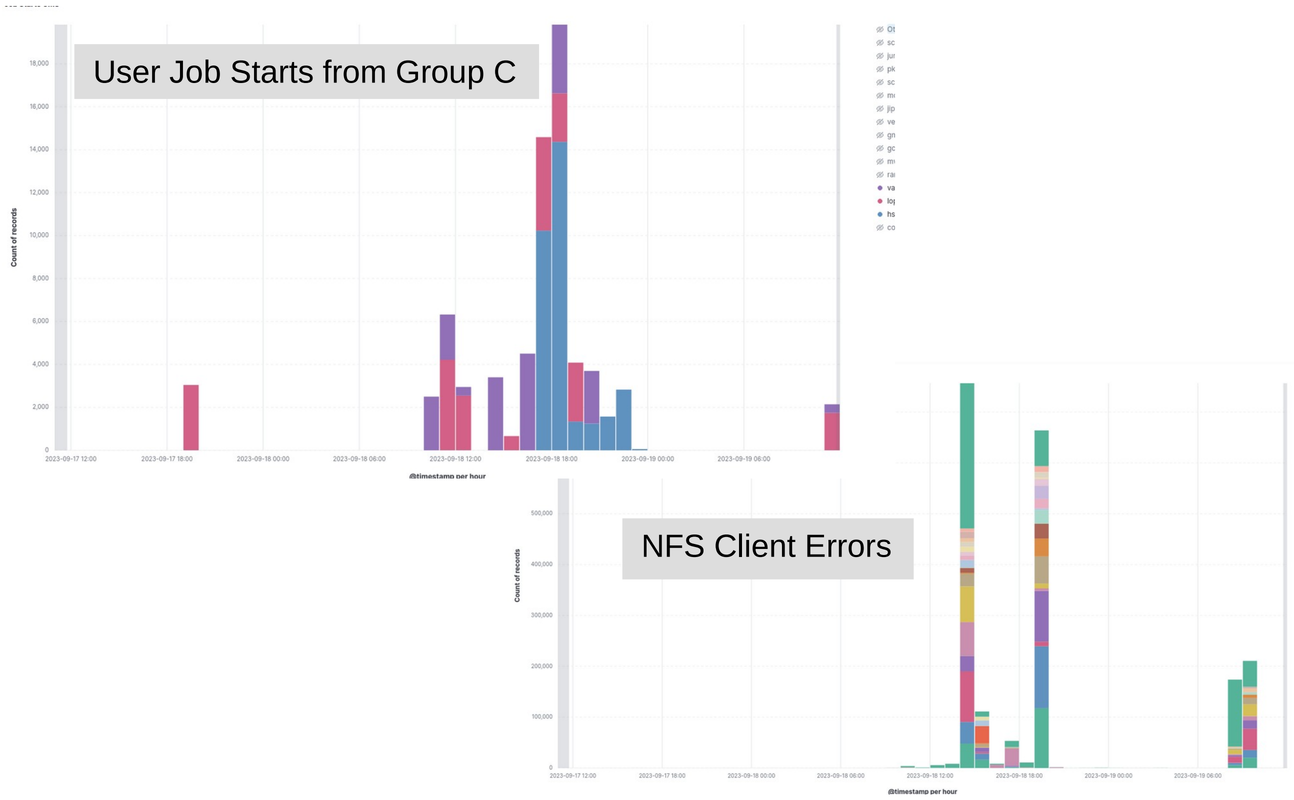
- Going for separate negotiator for GPU resources in parallel to the general urpose negotiator
  - User quota/history constraint to GPU nodes only
    - Fair sharing the GPU nodes
  - Many thanks to Todd for the suggestion!

Blind to *POSIX I/O*

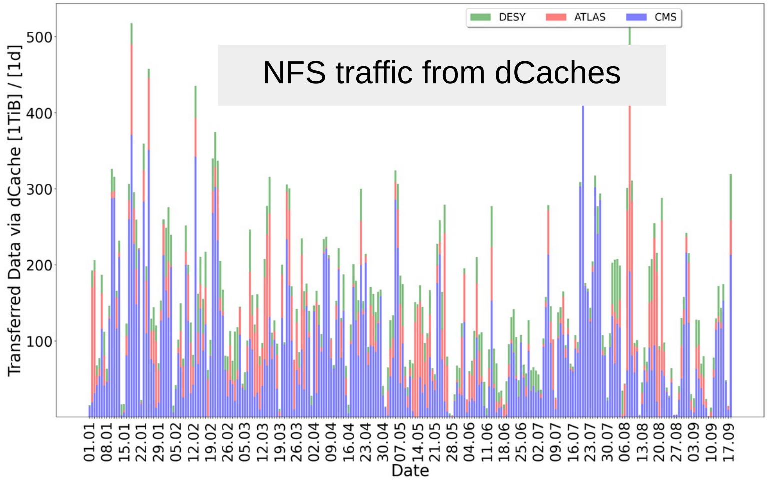
Not much progess since last year

- Users can grind storage pools to a hold
  - Users love POSIX/paths – NFS/GPFS/...
  - Better monitoring on the storage sides...
  - ...but still blind on the EP
  - Path/POSIX I/O *invisible* to Condor
  - Triangulating between storage, pools, EP nodes, individual jobs/users

- Everything will get better with more current kernels
  - More easier to tap into the kernel (hopefully)
  - Mid term aim: inject own job events with file handle statistics/details


User Job Starts from Group C

NFS Client Errors


NFS traffic from dCaches

Summary

Summary

DESY-HH Clusters

- Adapting to Energy Challenges
- Adapting to OS Challenges

- Improved our monitoring
  - eBPF could become quite useful
    (when finally being on a current kernel with all nodes)
    - Job I/O still black whole with hardly any insight
    - Side ~~car~~ motorcycle monitor job
      ideallly outside user context within the job - startd cron? precmd?

- Ongoing task to making all users equal(ly unhappy)

Appendix

{ "_index": "batch-eventlogs-2023.09.17", "_type": "_doc", "_id": "12PIoooB0tHQslGDWmKT", "_version": 1, "_score": 1, "_source":
{ "ResidentSetSize": 14971620, "CumulativeRemoteSysCpu": 134, "RemoteUserCpu": 309, "type": "json", "@version": "1", "CpusUsage":
0.999287530459436, "path": "/var/log/condor/EventLog.json", "CumulativeRemoteUserCpu": 309, "JobCurrentStartDate": 1694947239,
"BlockWriteKbytes": 700944, "SysProject": "af-belle2", "Size": 28744640, "beat": { "timestamp": "2023-09-17T10:55:53.811Z" },
"DESYAcctGroup": "BIRD_belle", "ClusterId": 40941363, "host": "bird-htc-sched14.desy.de", "NumJobStarts": 1, "Cluster": 40941363, "User":
"huwhaigh@desy.de", "RequestCpus": 1, "GlobalJobId": "bird-htc-sched14.desy.de#40941363.150#1694945033", "DiskUsage": 1, "ProcId":
150, "BlockReadKbytes": 285540, "TriggerEventTypeName": "ULOG_IMAGE_SIZE", "Proc": 150, "CpusProvisioned": 1, "Project": "af-belle2",
"RemoteSysCpu": 134, "Owner": "huwhaigh", "RemoteWallClockTime": 0, "ExitStatus": 0, "MemoryUsage": 14621, "EventTime": "2023-09-
17T12:55:53.518", "@timestamp": "2023-09-17T10:55:53.811Z", "tags": ["multiline", "bird-htc-sched14.desy.de",
"/var/log/condor/EventLog.json", "batch-eventlogs", "naf", "naf-lrms", "condor-scheduler", "condor-master", "kafka"],
"TriggerEventTypeNumber": 6, "MyType": "JobAdInformationEvent", "Subproc": 0, "EventTypeNumber": 28, "NumShadowStarts": 1 },
"fields": { "Owner": ["huwhaigh"], "NumJobStarts": [1], "TriggerEventTypeNumber": [6], "RemoteUserCpu": [309], "Size": [28744640],
"DiskUsage": [1], "type": ["json"], "MyType": ["JobAdInformationEvent"], "ExitStatus": [0], "path": ["/var/log/condor/EventLog.json"],
"Subproc": [0], "type.keyword": ["json"], "host": ["bird-htc-sched14.desy.de"], "TriggerEventTypeName.keyword": ["ULOG_IMAGE_SIZE"
], "host.keyword": ["bird-htc-sched14.desy.de"], "GlobalJobId": ["bird-htc-sched14.desy.de#40941363.150#1694945033"],
"CpusProvisioned": [1], "@version.keyword": ["1"], "Owner.keyword": ["huwhaigh"], "RemoteWallClockTime": [0], "DESYAcctGroup":
["BIRD_belle"], "tags": ["multiline", "bird-htc-sched14.desy.de", "/var/log/condor/EventLog.json", "batch-eventlogs", "naf", "naf-lrms",
"condor-scheduler", "condor-master", "kafka"], "CpusUsage": [0.99928755], "Project": ["af-belle2"], "MyType.keyword":
["JobAdInformationEvent"], "EventTypeNumber": [28], "RequestCpus": [1], "SysProject": ["af-belle2"], "EventTime": ["2023-09-
17T12:55:53.518Z"], "Project.keyword": ["af-belle2"], "CumulativeRemoteSysCpu": [134], "GlobalJobId.keyword": ["bird-htc-
sched14.desy.de#40941363.150#1694945033"], "ResidentSetSize": [14971620], "CumulativeRemoteUserCpu": [309], "User":
["huwhaigh@desy.de"], "BlockWriteKbytes": [700944], "tags.keyword": ["multiline", "bird-htc-sched14.desy.de",
"/var/log/condor/EventLog.json", "batch-eventlogs", "naf", "naf-lrms", "condor-scheduler", "condor-master", "kafka"],
"JobCurrentStartDate": [1694947239], "ProcId": [150], "Proc": [150], "RemoteSysCpu": [134], "TriggerEventTypeName":
["ULOG_IMAGE_SIZE"], "@version": ["1"], "beat.timestamp": ["2023-09-17T10:55:53.811Z"], "DESYAcctGroup.keyword": ["BIRD_belle"],
"ClusterId": [40941363], "Cluster": [40941363], "MemoryUsage": [14621], "User.keyword": ["huwhaigh@desy.de"], "@timestamp":
["2023-09-17T10:55:53.811Z"], "SysProject.keyword": ["af-belle2"], "NumShadowStarts": [1], "path.keyword": [
"/var/log/condor/EventLog.json"], "BlockReadKbytes": [285540] } }

Full Containerized Deployment in the Future?

Thinking out loud

- Has someone already experiences with Podman-based deployents?

- K8s user namespace support still beta (CRIO not full userspace w. mapping possible AFAIS)
- Full userspace/uid mapping should be reasonable with Grid

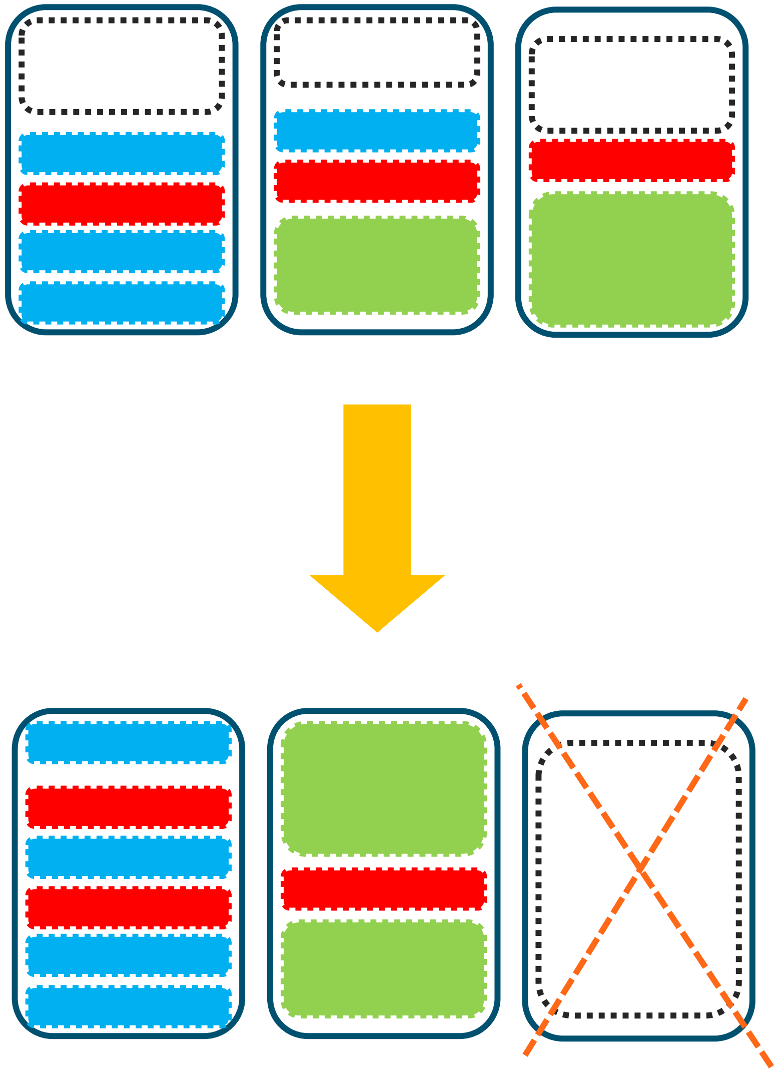
- NAF has POSIX mounted shared FS'ses (NFS kernel client...)
  - Have to run on root user namespace :(
  - No good idea, how to realize or with what runtime
    - Fully unpriviledged User App Containers in K8s ??

Cluster-wide Power Shaping

Workload dependent Power Saving: Users

- User Clusters with more dynamic utilization
  - Potentially higher job entropy
  - Cluster intrinsic power shaping

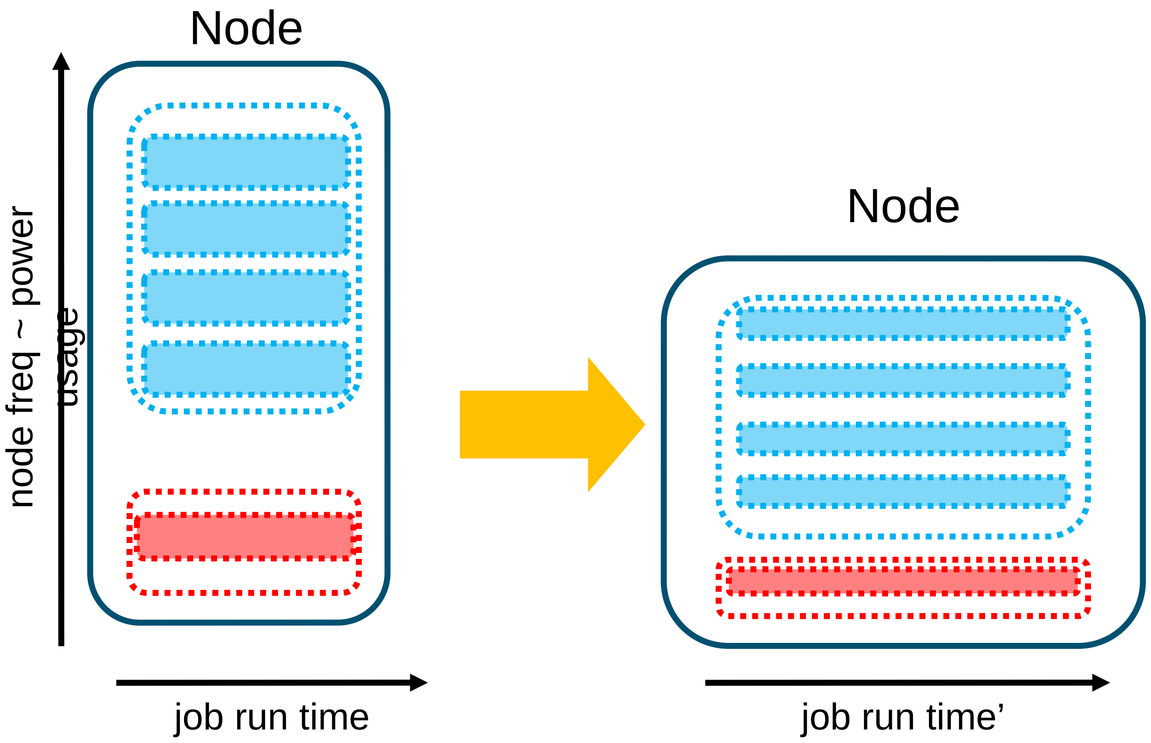
  - Horizontal vs. vertical scheduling
    - Cluster *compression*
    - price: higher job upstart latency/entropy
    - More aggressive node shedding
    - Opportunistic node ramp up with backfill workloads on standby



Power Shaping per Node

Workload dependent Power Saving: Grid

- Power consumption optimization depending on usage patterns
- Production Workload/Cluster
  - Job-Life-Time dependent scheduling difficult (payload run time potentially unknown to pilots)
  - Cluster external power shaping
  - Node/kernel power shaping transparent to payloads
- CPU Governor stepping driven by Green Energy availability



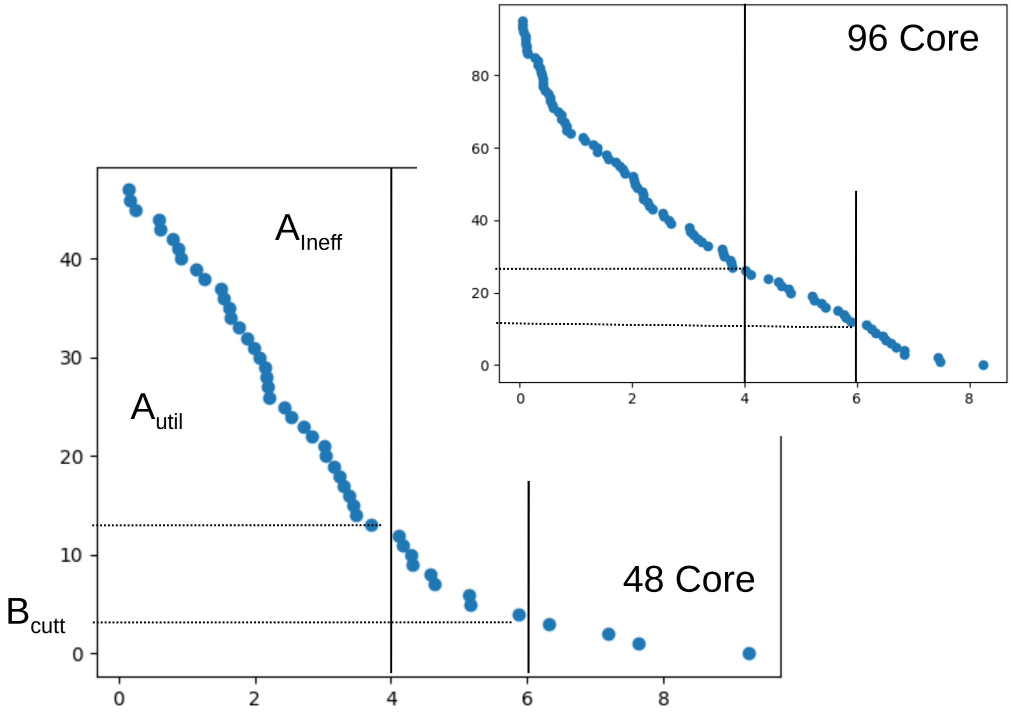
Preemption: Job Shedding minimizing Cycle Waste

User Side Implementation Necessary

- Draining Cluster/Nodes
  - Wasting idle CPU cycles
- Hard Node shedding
  - Wastes all CPU cycles so far of active jobs

- Ideally: Pre-emptable Jobs
  - Grace Period SIGTERM ⟶ SIGKILL
  - Snapshot/Stage results so far

  - Requires: User Side Implementation...



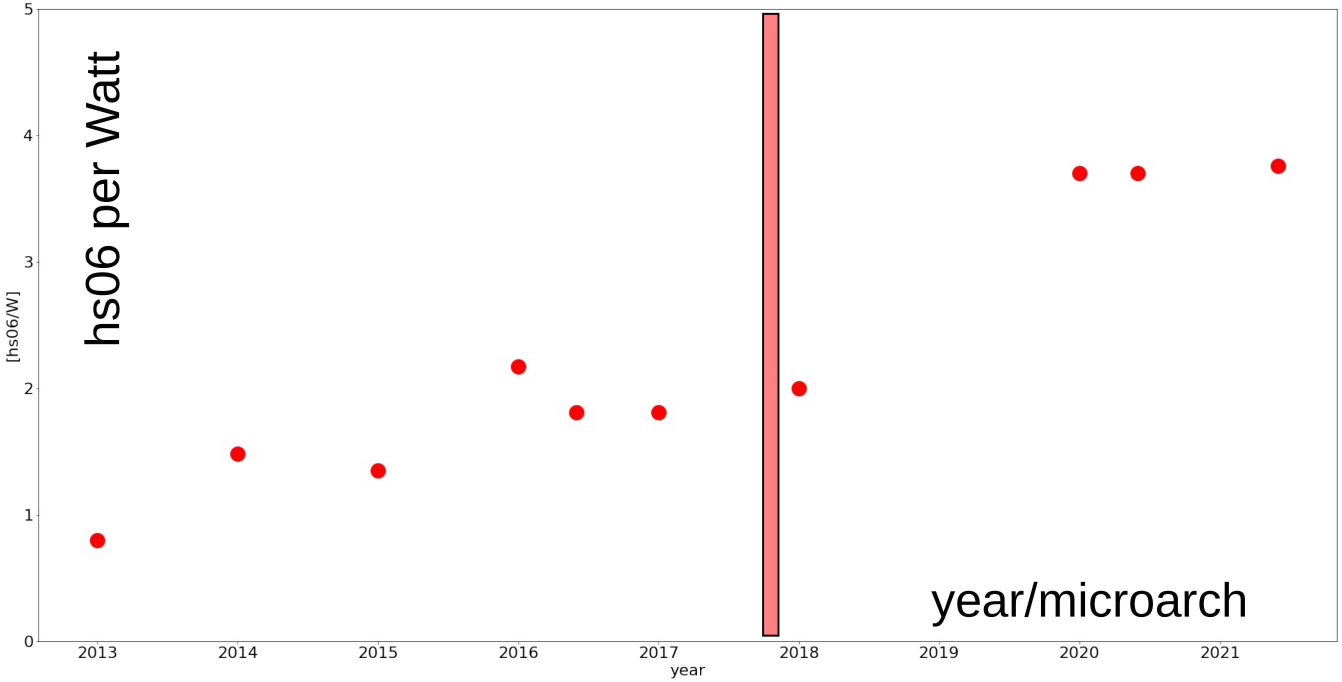
Simulation: Node Utilization while Draining

Architecture/Generation Energy Efficiency

CPU Efficiency per Electric Power Consumption

- Significant efficiency gains with recent microarchs (aka Zen)
- CPU compute power per Watt gain ~4x from oldest workers still in production

- Old, energy inefficient nodes as dynamic moderators for shedding/fan out
- Shaping Frequency depending on production job run time/draining rate



Production Cluster Energy Efficiency

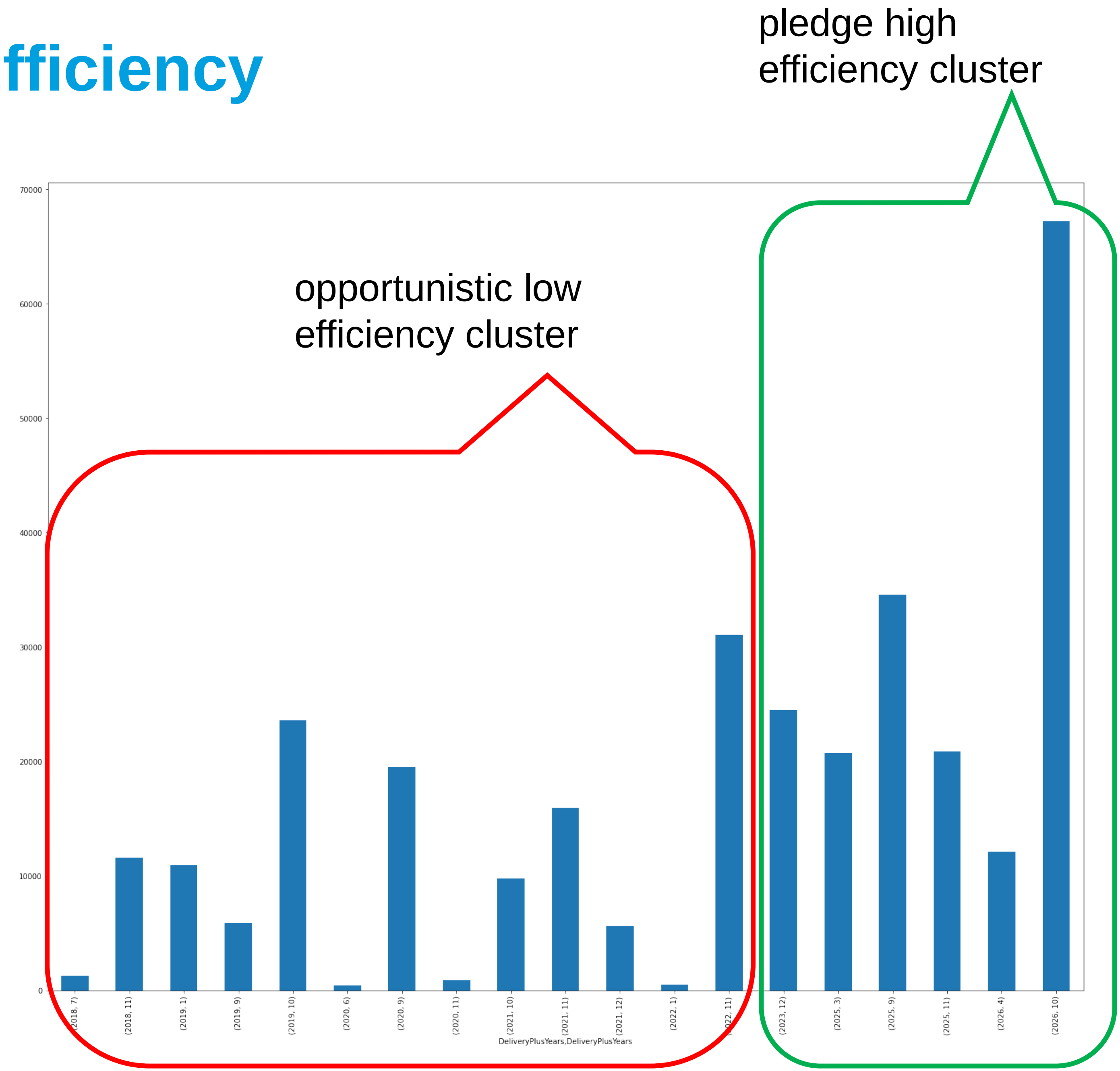
WattHours consumed for HS06 delivered

E.g.

- target deliverable: 1000 kHS06

- “combo” cluster: ~410 kWh
- “high efficiency” cluster: ~298 kWh
- “low efficiency” cluster: ~587 kWh

- Low efficiency cluster as opportunistic resource
  - Load shedding when necessary
  - Scheduling has to be adapted



pledge high efficiency cluster

opportunistic low efficiency cluster

Opportunistic Resource Utilization

Utilizing surplus green energy

- Complementary to load shedding

- Node ramp up O(~minutes)
- O(shedding)? Depends on payload runtimes and overall cluster job entropy

- Need interface to weather/green energy pricing forecasts
    - helper HTCondor Daemon with external input for cluster shaping?

- Damping wavelengths by payloads
- How to avoid significant draining idle waste cycles
    - Backfilling short jobs?
    - Assist users implementing preemption?