Energy Savings

By power modulation of a HTC pool

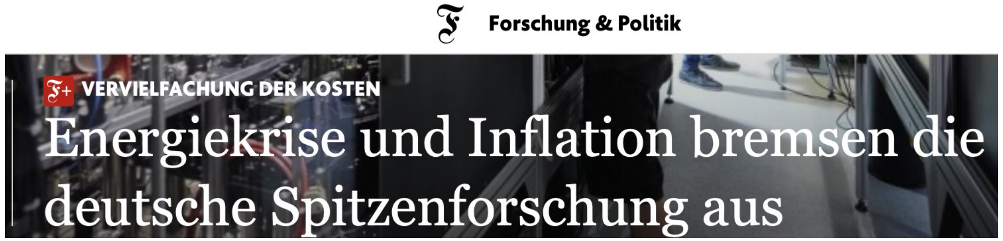
Beyer, Christoph with slides and input from Thomas Hartmann & Yves Kemp
Orsay, 21-09-2023

Recent history and upcoming future

Winter 2021/22 expected to be critcal – spoiler it was not

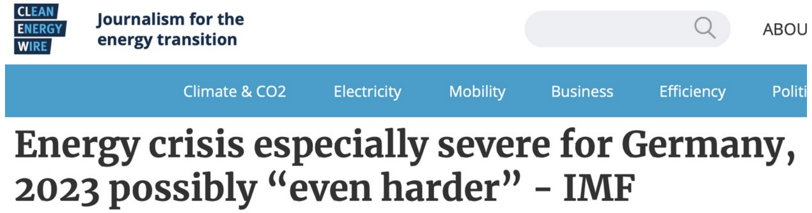
- Assumption: There will be (frequent and) short-term interruptions in power provisioning
- Reality: Did not happen. At least not on short-term.
- Power consumption profile rather well known. Power production profile (RE) known up to 2 days in advance (TransnetBW “StromGedacht”)
- Assumption: Energy prices will kill us. Reality: Prices in 2022 not that exceptionnally high

- The time for immediate action is over
- ~~Time to relax and get back to business as usual~~
- Time to design and build really sustainable research infrastructures

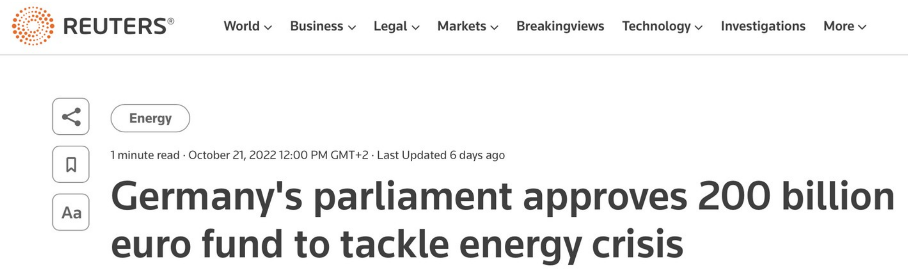

Forschung & Politik

VERVIELFACHUNG DER KOSTEN

Energiekrise und Inflation bremsen die deutsche Spitzenforschung aus



Energy crisis especially severe for Germany, 2023 possibly “even harder” – IMF



1 minute read · October 21, 2022 12:00 PM GMT+2 · Last Updated 6 days ago

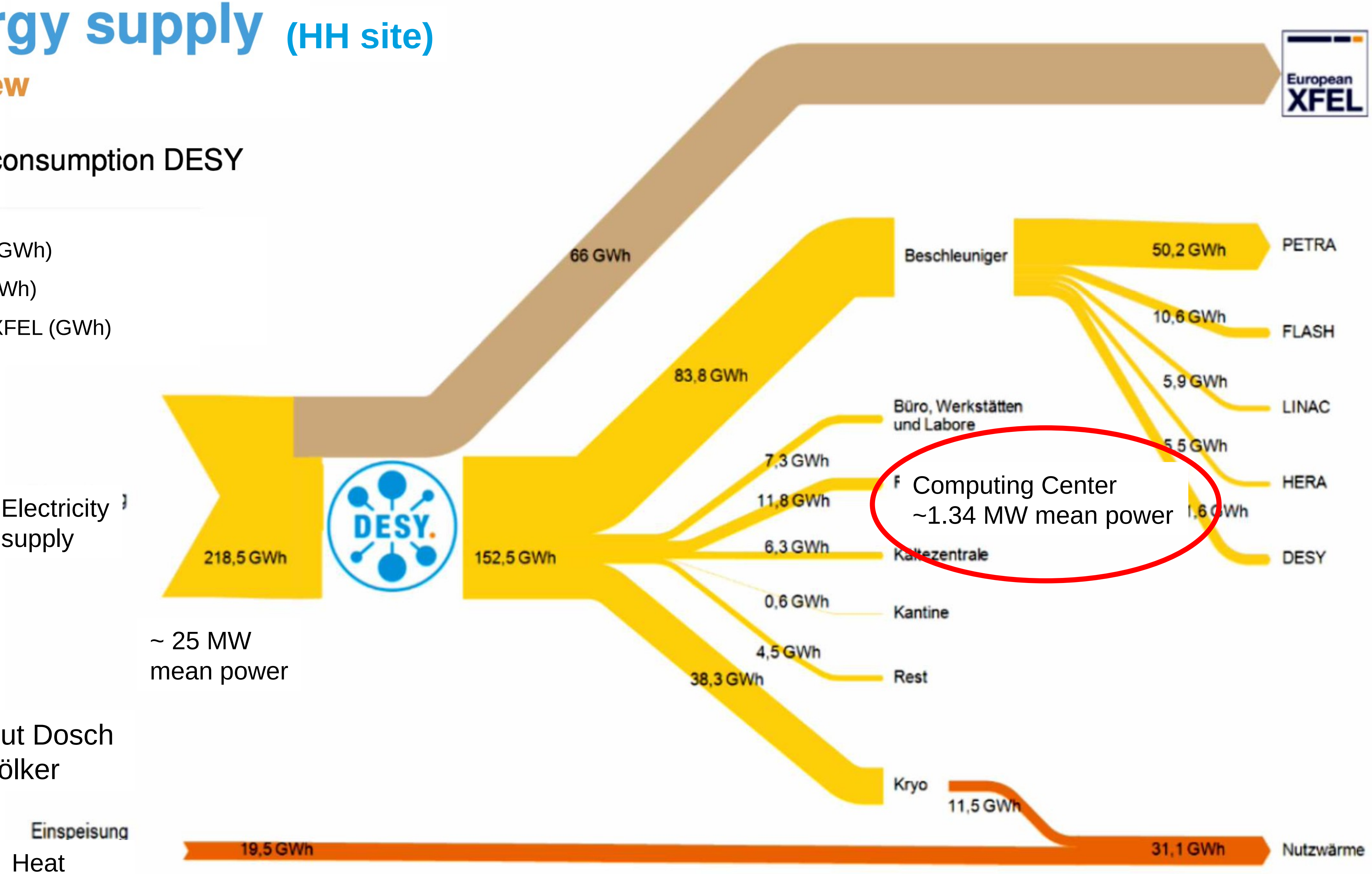
Germany's parliament approves 200 billion euro fund to tackle energy crisis

Energy supply (HH site)

Overview

Power consumption DESY 2021

- ▮ Power (GWh)
- ▮ Heat (GWh)
- ▮ Power XFEL (GWh)



Electricity supply

~ 25 MW mean power

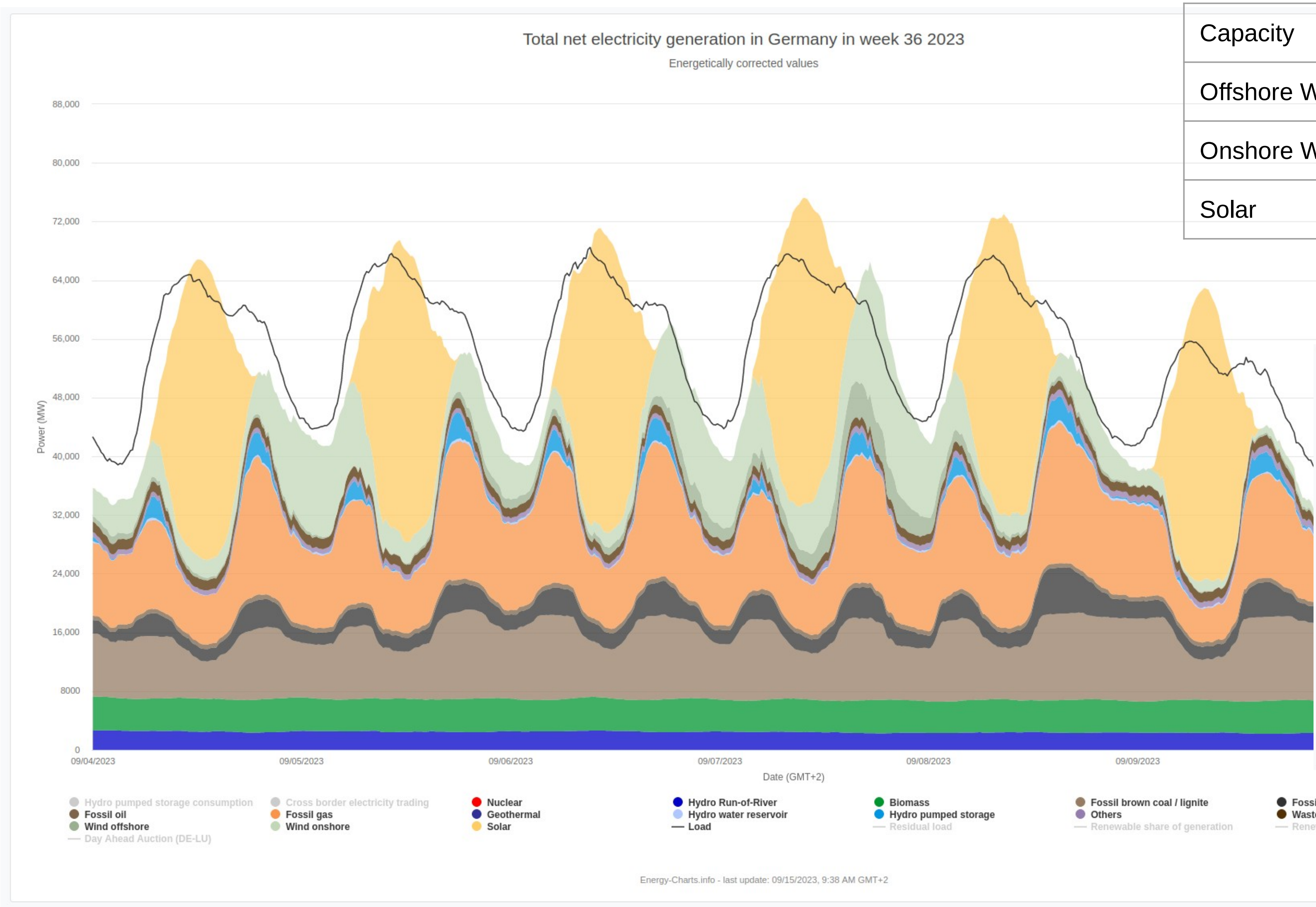
Computing Center
~1.34 MW mean power

Slide: Helmut Dosch & Denise Völker

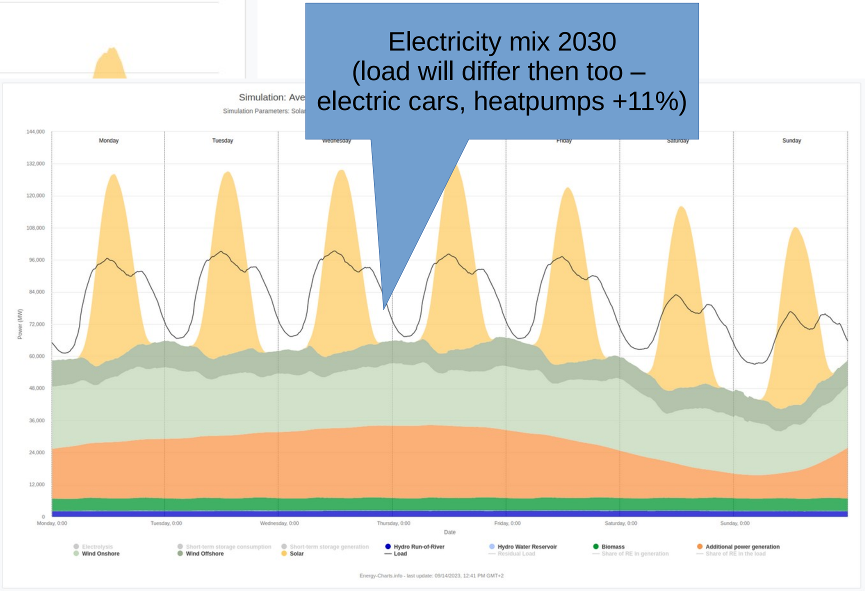
Heat supply

Public net electricity generation in Germany Last week & 2030

https://www.energy-charts.info/index.html



Capacity	2022(GW)	2030 (GW)	Factor
Offshore Wind	7.8	30	4
Onshore Wind	56	115	2
Solar	66	215	3



Electricity mix 2030
(load will differ then too –
electric cars, heatpumps +11%)

https://www.energy-charts.info/index.html

Two HTC pools in the data centre

A lot more to optimize of cause but …

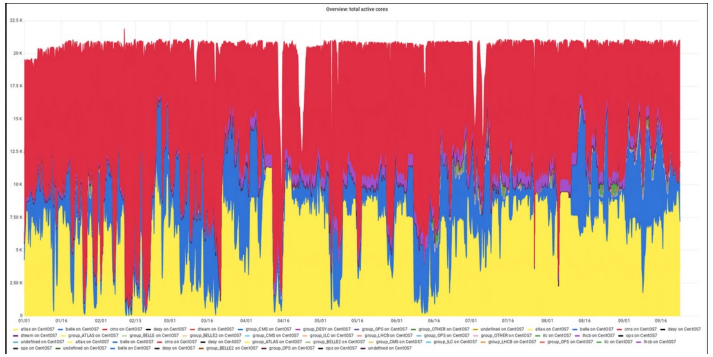
GRID HTC pool

- cluster utilized 24/7

- high utilization - more *efficient/effective* than the NAF user cluster
  - w/o respect to job start latency
  - much higher inertia...
  - dynamic adaption to power provisioning only on longer time scales

- some sensitivity on payload efficiency (wall vs cpu time)

- investigated transparent job/CPU throttling as stop gap

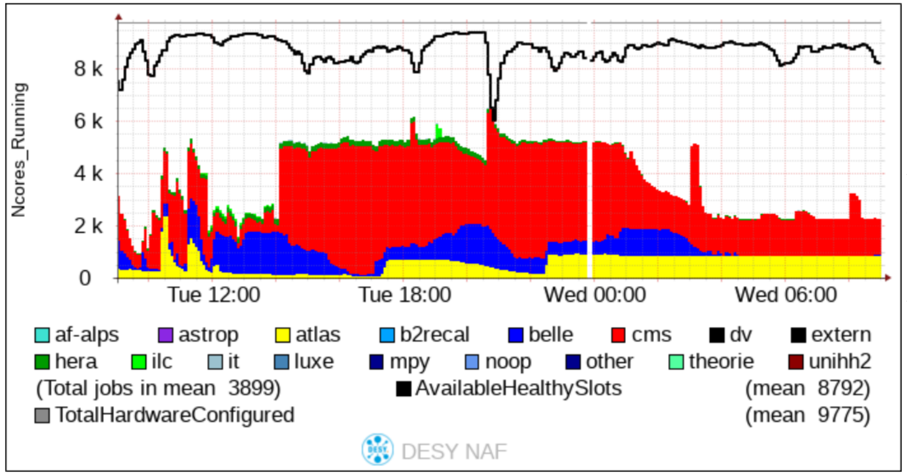


NAF = National Analysis Facility - User Cluster

- complementary to the Grid for individual users’ jobs

- cluster utilization by the users fluctuating
  - day/night user behaviour + seasonable effects (aka conferences & holidays)
  - power consumption closely coupled

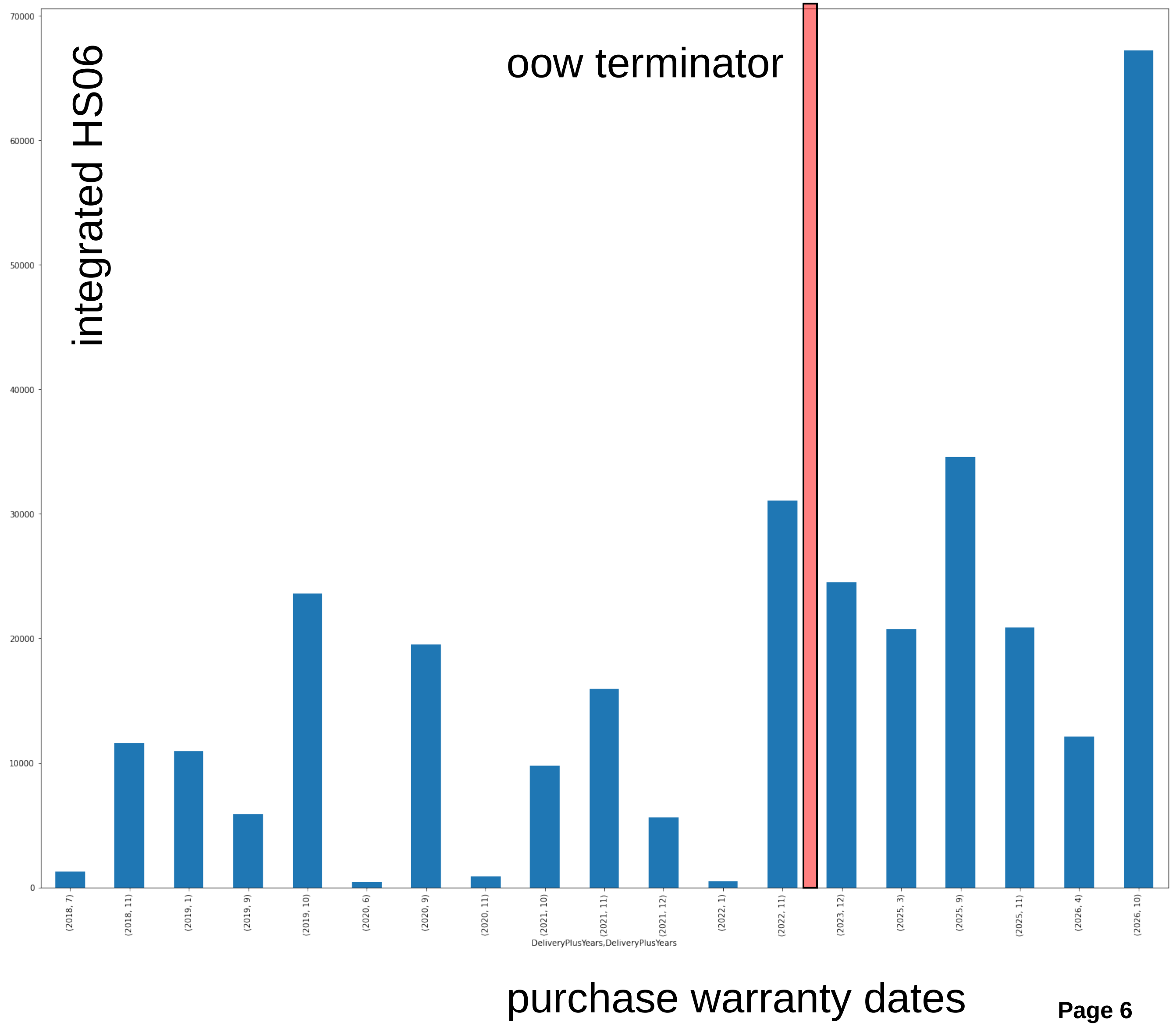
- had been keeping resources available 24/7
  - low job start latency pleases/placates users
  - now might become a noticeable cost



Cluster Energy Efficiency

HepSpec by Generation – measurement & evaluation done by T. Hartmann

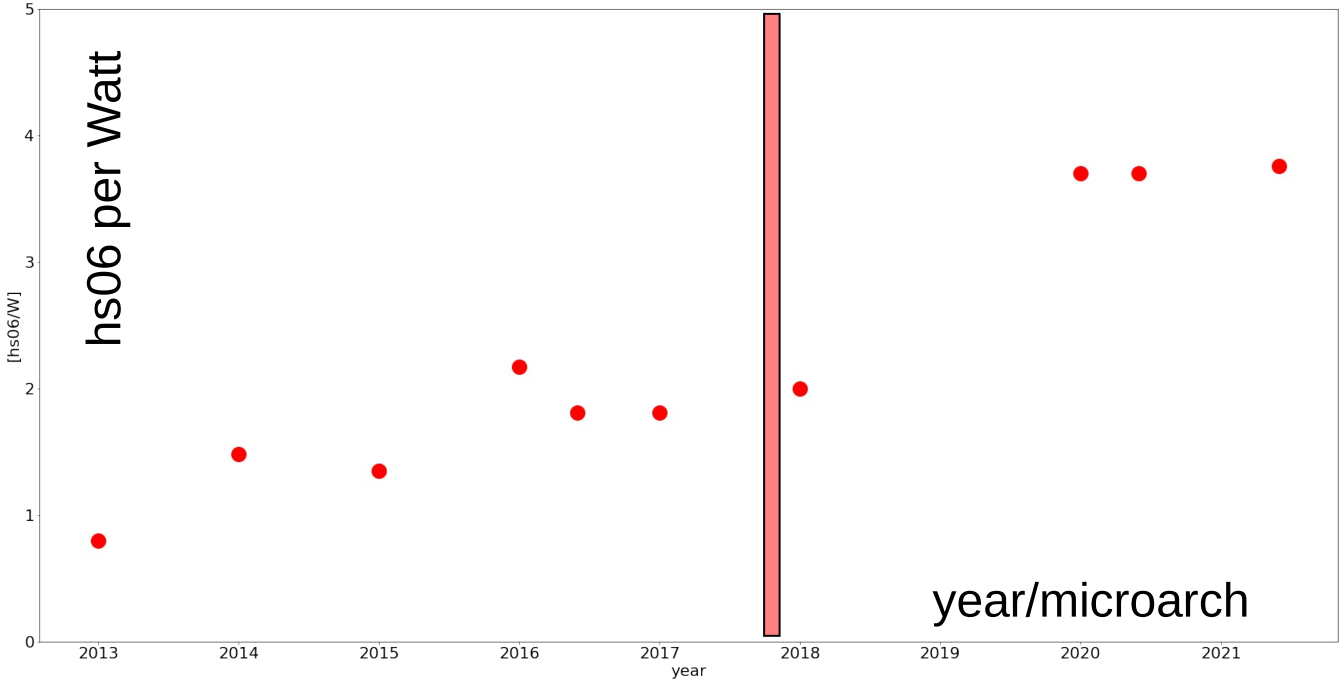
- Grid pledge policy so far
  - Pledges with under warranty workers
  - Extra HS06s from oow workers



Cluster Energy Efficiency

Arch HS06 per Watt

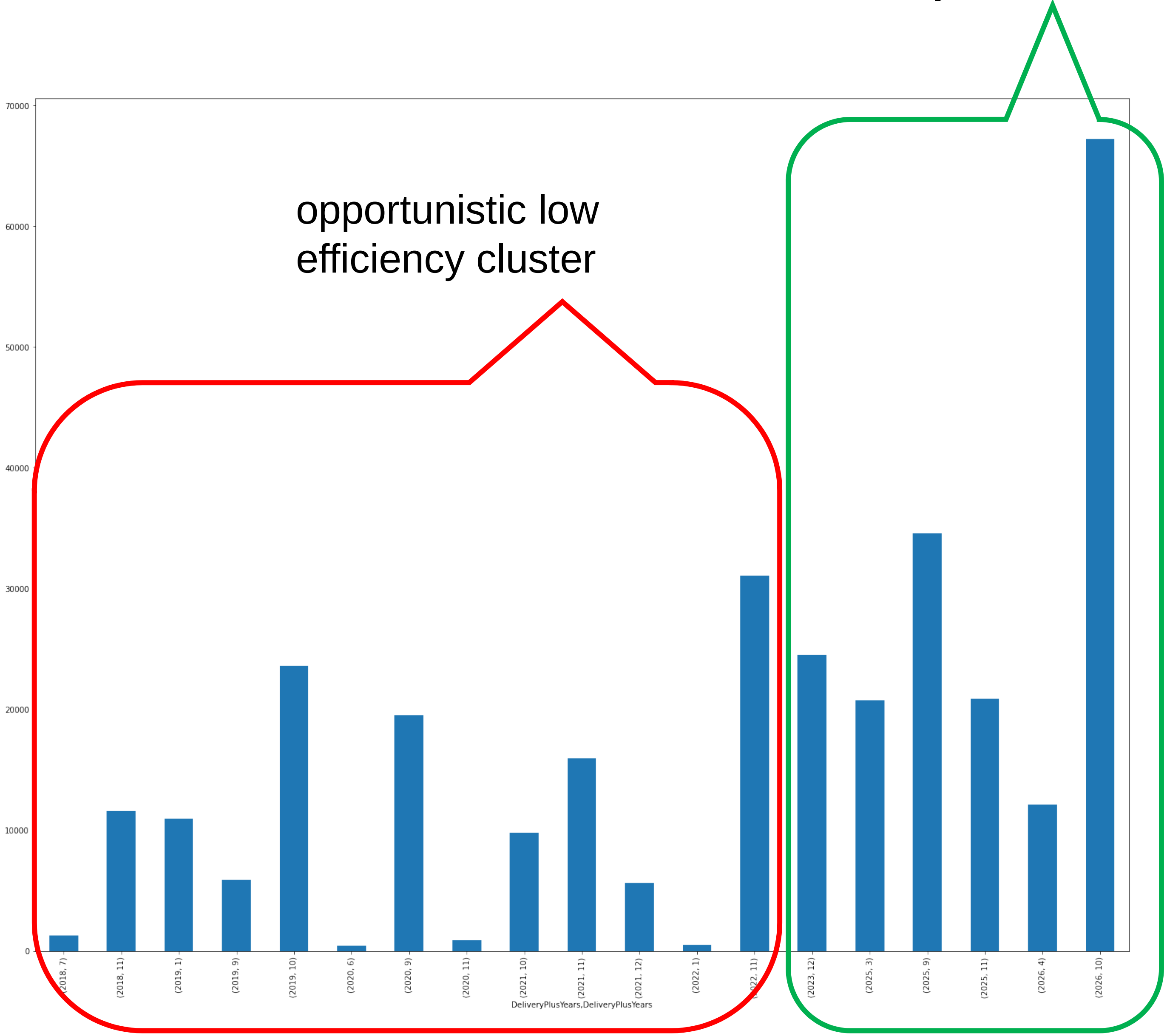
- Significant efficiency gains with recent microarchs (aka Zen)
- HS06 per Watt gain ~4x from oldest workers still in production



Cluster Energy Efficiency

Cluster sub designations

- Need to reconsider cluster operations with respect to efficiency
- Operating inefficient EPs 24/7/365 still justifiable?
- Pledged high efficiency resources always online
- Low efficiency cluster as opportunistic resource
  - Load shedding when necessary
  - Scheduling needs to be adapted



pledge high efficiency cluster

opportunistic low efficiency cluster

Job/CPU Throttling

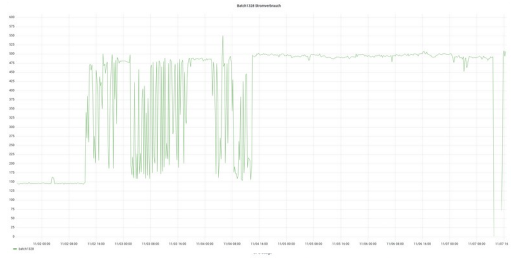
On demand throttling

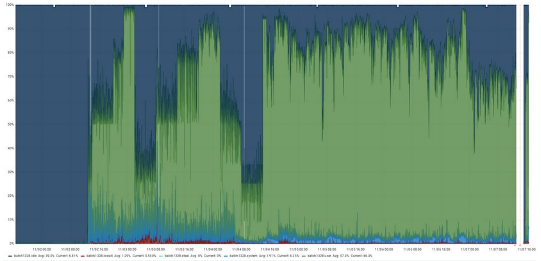
- run a few tests
  - throttling node to [100%, 75%, 50%, 25%] CPU time + [0 load, off]
  - PSU & PDU power consumption(s)
  - ~75W per 25% steps (@25% extra *savings* due to IOwait...)

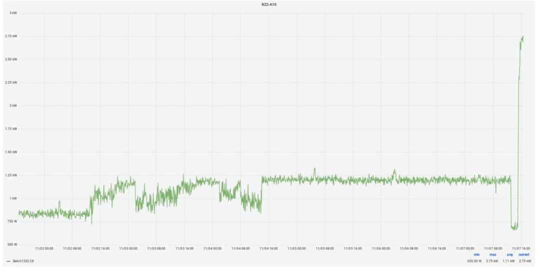
- base idle load ~150W incl. PSU ~10% inefficiency

- realistically 1/3 of the power consumption might be saved by throttling...
- ...with a ~150W base offset
  - not very efficient (effective??) for a nearly 100% utilized HTC cluster

- **conclusions** for power savings or cluster power ceiling
  - load shedding nodes for good...







The road to a more sustainable pool

Summary

Short term (mostly finished)

- Monitoring the powerusage of the pool
  - Using internal sensor readings
- Automatically shutdown EPs that are idle
  - condor_rooster & foreman
- Classify EPs by there power-efficiency
  - Benchmarking
- Tweak pool to more vertical than horizontal overall behavior (prefer more effective nodes)
- Make users aware of power consumption/CO2 emission
  - Send e-mail with summarys
  - User education ‘sustainable programing’

Mid term (started)

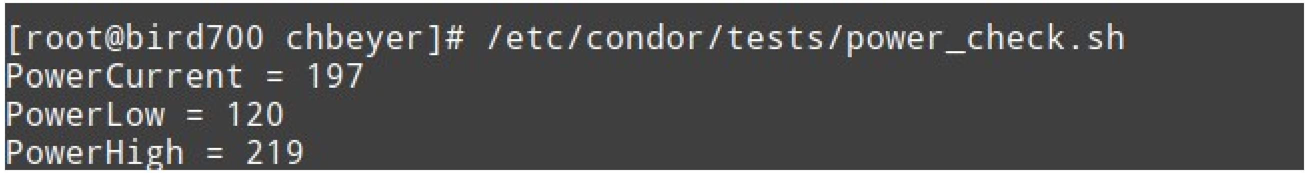
- max total cluster power consumption tunable
  - Be able to steer power consumption along a given timeline (e.g. availability of green energy)
  - cluster power ceiling
- None of the above currently coupled to monetary advantages (fixed electricity price deal)

THE MONEY TREE REDUX ©WINSTON SMITH 2017(1983)

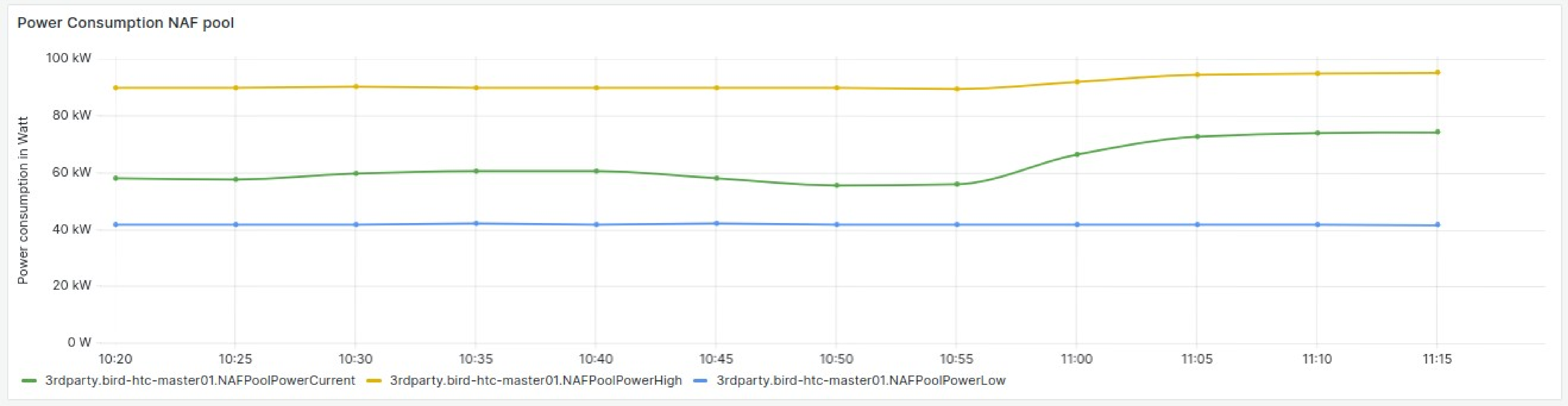
Powerusage monitoring

Import sensor readings into host classadds

- Internal power sensor readings turned out to be more exact than we thought
  - Only few racks equipped with external power measuring equipment
  - Measurement by rack difficult anyway because mixed setup per rack
- IPMITOOL & startdcron ->
- Grafana does the rest



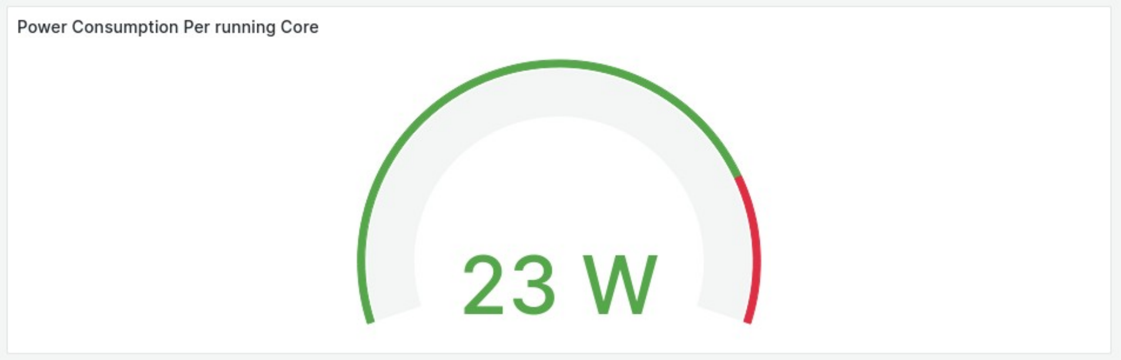
```
[root@bird700 chbeyer]# /etc/condor/tests/power_check.sh
PowerCurrent = 197
PowerLow = 120
PowerHigh = 219
```

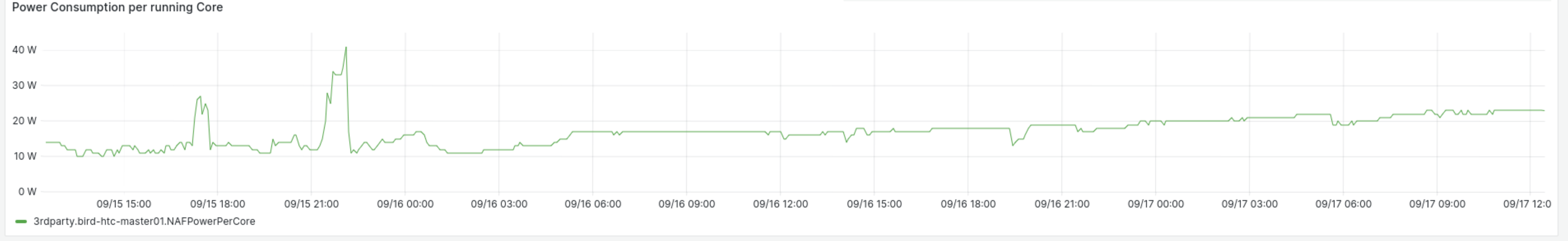


Powerusage monitoring

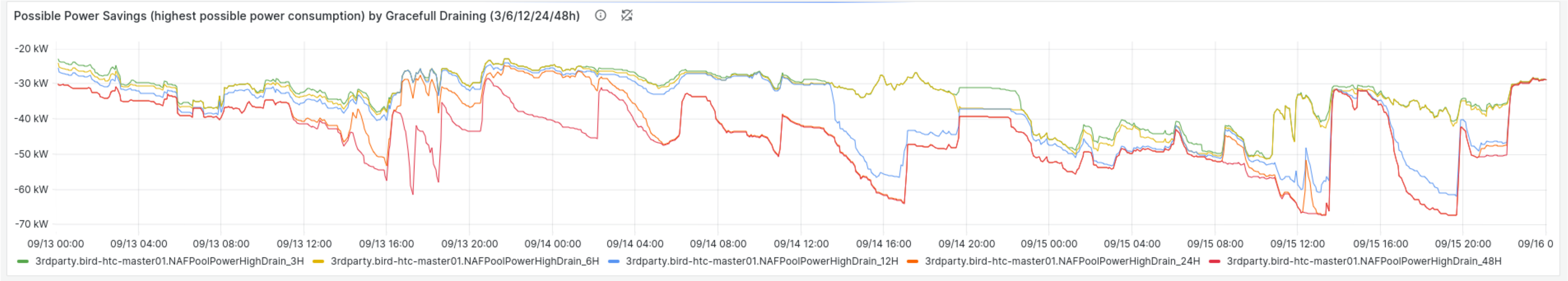
Some more possible graphs

- Power consumption per running core





- Possible power savings by graceful draining



Power modulation

How to do it in condor

- Checking the idle time of the EP was more complex than estimated, best done on the EP itself (fixed in future release I think ?)
  - Startdcron script checks the number of running slots and adds up the time
- Using the built-in ‘hibernate’ mechanism to actual turn the EP off
  - `HIBERNATE = ifThenElse((SecondsMachineIdle > 1800 && CanPowerDown =?= true && remote_administered =?= false),"S5","NONE")`
    - Takes in account seconds of idleness, ability of workernode to be powered up again, state of node (if remote administred for some reason leave it alone)

- Replaced the built-in plugin for powermanagement (easy todo and well documented, runs on the EP)
  - `HIBERNATION_PLUGIN = /usr/libexec/condor/desy_power_state.sh`
    - Announce a 12h downtime in global monitoring/alarming (Icinga)
      - `curl --silent --output /dev/null -k -u $ICINGA_AUTH -H 'Accept: application/json' -X POST ' https://icinga.desy.de:5665/v1/actions/schedule-downtime' <snip> ....`
    - Send some information to KAFKA in order to track node behavior later
    - Turn node off `sudo /sbin/poweroff`
    - Magic sysrequest could be used but would be harder on filesystems

- Problem: Condor sends a last classadd update without the necessary ‘offline’ flag when powering down the node
  - Changed `KillSignal=SIGKILL` in `/etc/systemd/system/condor.service` (report HTCONDOR-1806)

Power down idle ressources

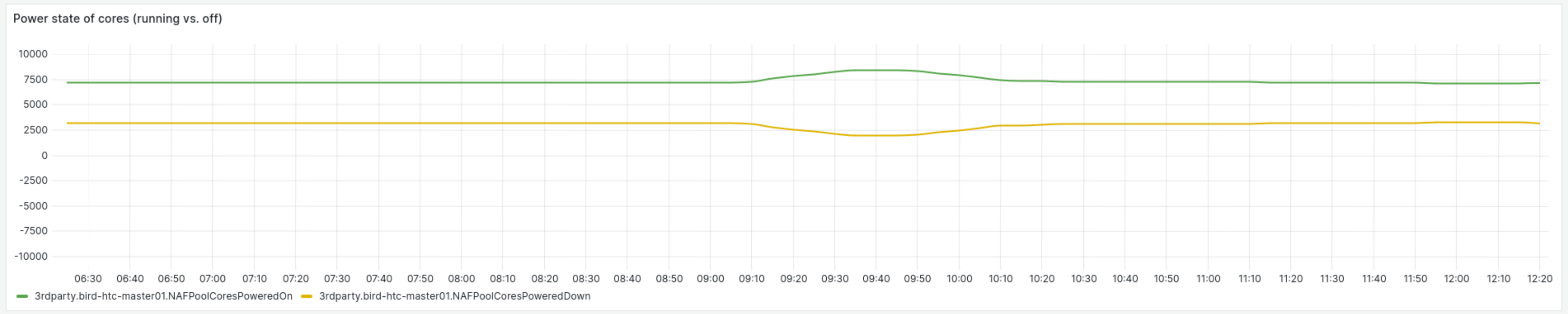
On the Collector

- We want to keep the Offline classadds for a long time
  - OFFLINE_EXPIRE_ADS_AFTER should be default = 30 days or longer (?)
- Collector Updates classadd if job matches
  - MY.MachineLastMatchTime
- Rooster checks condition of matched EP
  - ROOSTER_UNHIBERNATE = (Offline && Unhibernate) || (Offline && TARGET.LastHeardFrom > (time() + 43000))
  - Unhibernate is part of the EP classadd set during hibernation Unhibernate = MY.MachineLastMatchTime =!= undefined
  - Wake up machine if ~12h down (matching the downtime we set in ICINGA)
- Condor_rooster
  - Monitors EP classadd (MY.MachineLastMatchTime)
  - Calls plugin to wake up node if conditions met
    - ROOSTER_WAKEUP_CMD = "/var/lib/condor/util/desy_wake.sh"
  - Writes EP classadd to <STDIN> of plugin

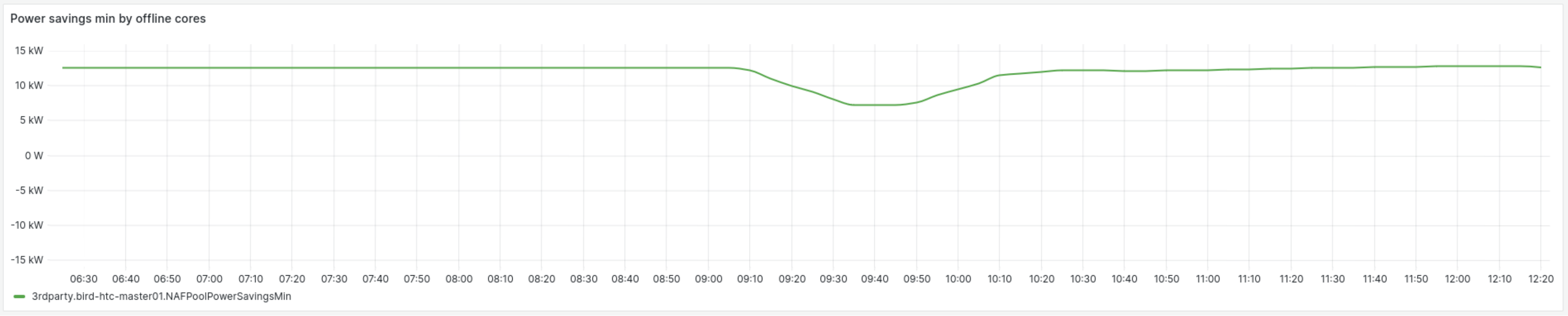
Power up idle ressources

On the Collector

- ROOSTER_WAKEUP_CMD = "/var/lib/condor/util/desy_wake.sh"
    - Ends downtime in ICINGA (curl call)
    - Uses FOREMAN to boot EP (curl call)



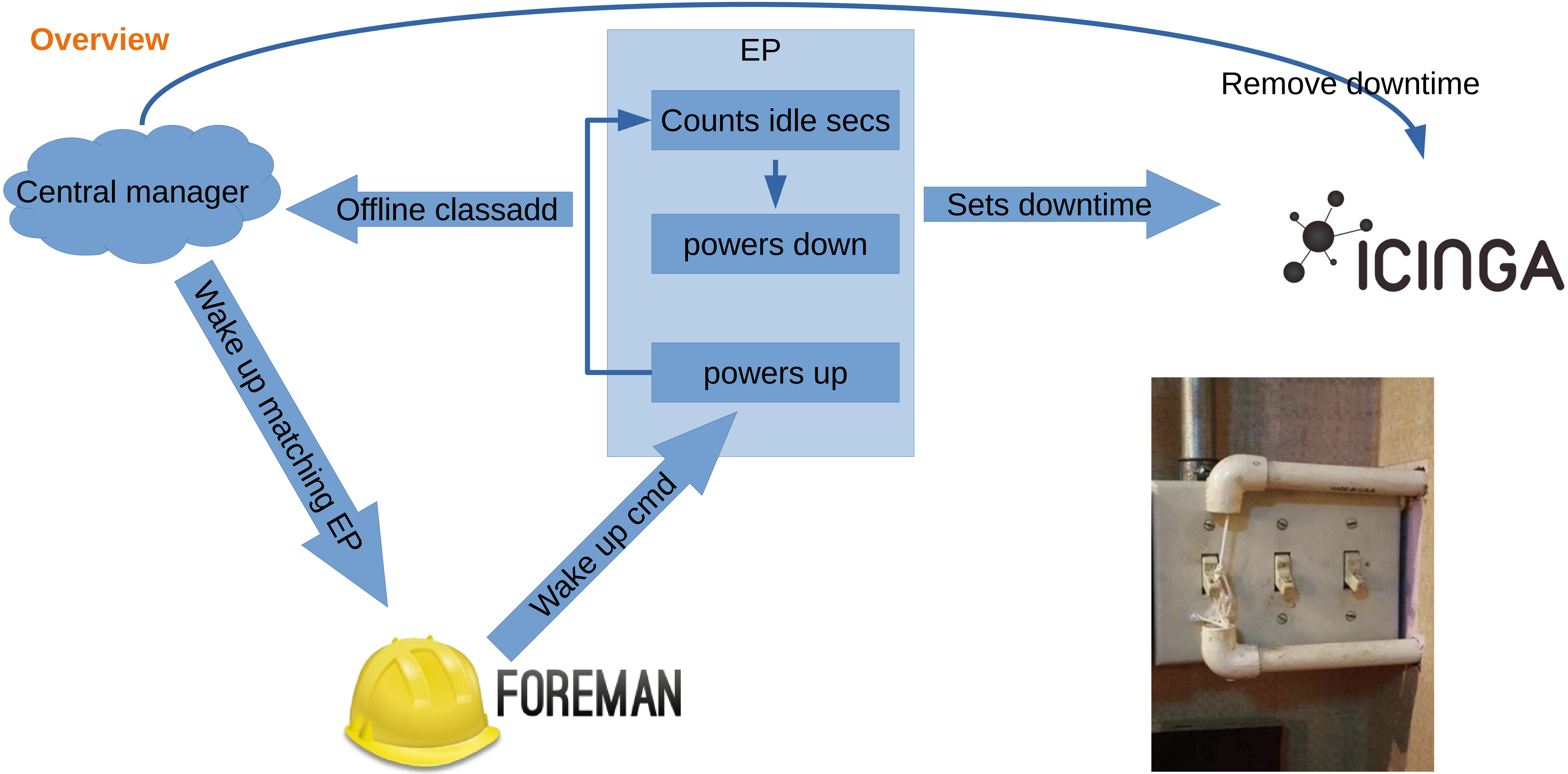
Power state of cores (running vs. off)



Power savings min by offline cores

Power down idle ressources

Overview



EP

Counts idle secs

powers down

powers up

Central manager

Offline classadd

Sets downtime

Remove downtime

Wake up matching EP

Wake up cmd

https://debeste.de/42414/Expertenl-sung-um-den-Schalter-aus-dem-Nebenzimmer

Summary and outlook

On the Collector

- Power modulation up and running
- Tagged less power efficient machines to mainly run short jobs
- Draining should be adapted to powerefficency of EPs (todo)
- Negotiation could be tweaked probably to get job density up on the EPs
- More sophisticated powermodulation should be easy to implement once
  - It is financially interesting
  - Green energy is available on the spot
- Make powerefficency a more ‘major’ point for new hardware aquisations (consider arm processors e.g.)
- Designing and building a really sustainable research infrastructure is a much bigger task with a multitude of aspects and considerations – there are quite some people working on it and hopefully it will extend the nowadays often seen green washing level !