News from the DESY clusters

HTC in GRID & NAF

Beyer, Christoph with slides and input from Thomas Hartmann & Yves Kemp
Paris, 16-04-2024

HELMHOLTZ

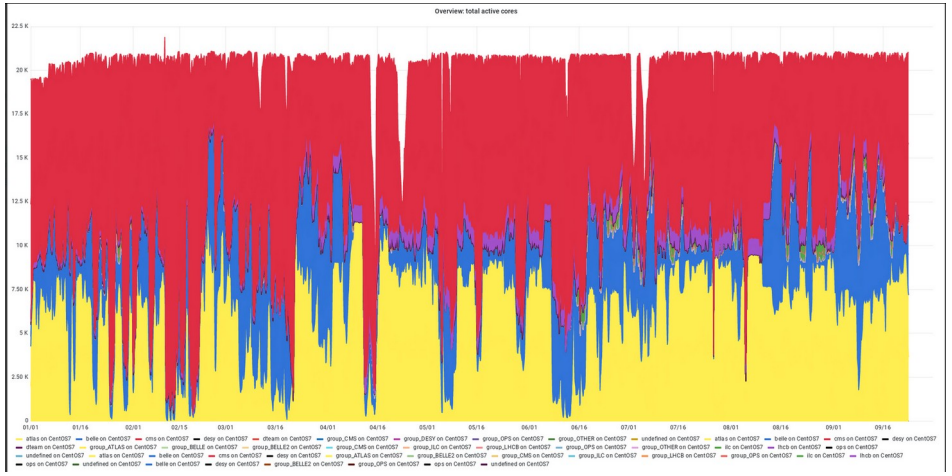
DESY.

Two HTC pools in the data centre

Computation for HEP mostly

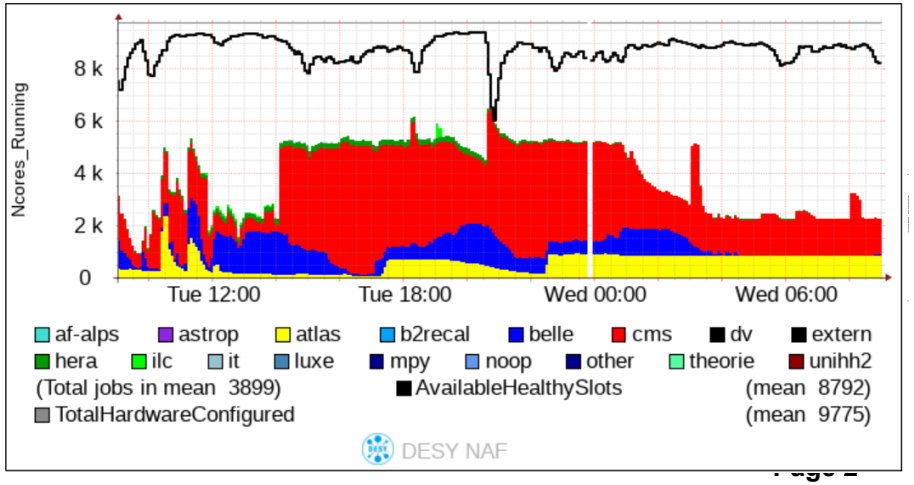
GRID HTC pool

- cluster utilized 24/7
- high utilization - more efficient/effective than the NAF user cluster
- No local/DESY accounts and dependencies
- Pilots from the usual sources



NAF = National Analysis Facility - User Cluster

- complementary to the Grid for individual users' jobs
- Causing 80% of the trouble and support work
- cluster utilization by the users fluctuating
  - day/night user behaviour + seasonable effects (aka conferences & holidays)
- Very individual job setups
- Local/DESY accounts with $HOME in AFS
- Highly depending on data access manged by mounted NFS filsystems mostly
  - CVMFS
  - DCACHE
  - DUST (GPFS)
  - AFS



Everything is data driven

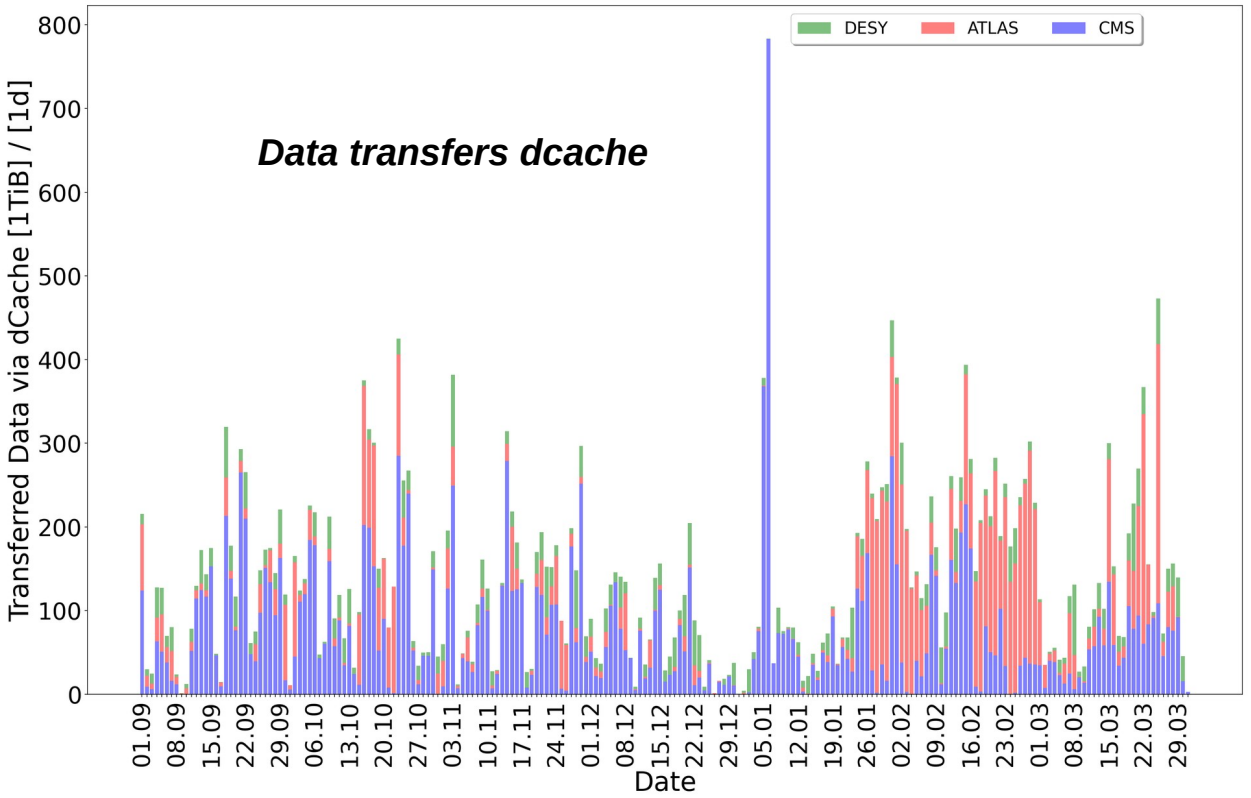
Some numbers from the NAF ...

DCACHE (PNFS)

- Transfer rates up to 1 PB per day
- ~ 20PB used
- 142 billion files (142.000.000.000)
- Hardware: 223 storage nodes

DUST (GPFS)

- 1,9 of 3,1 PiB used
- 1074 users- and 58 group folders
- 1,22 billion files (1.220.000.000)
- Hardware
  - 1x Lenovo DSS-G 240 (2. Generation)
  - 6x Server mit 2x100 GbE for NFS access



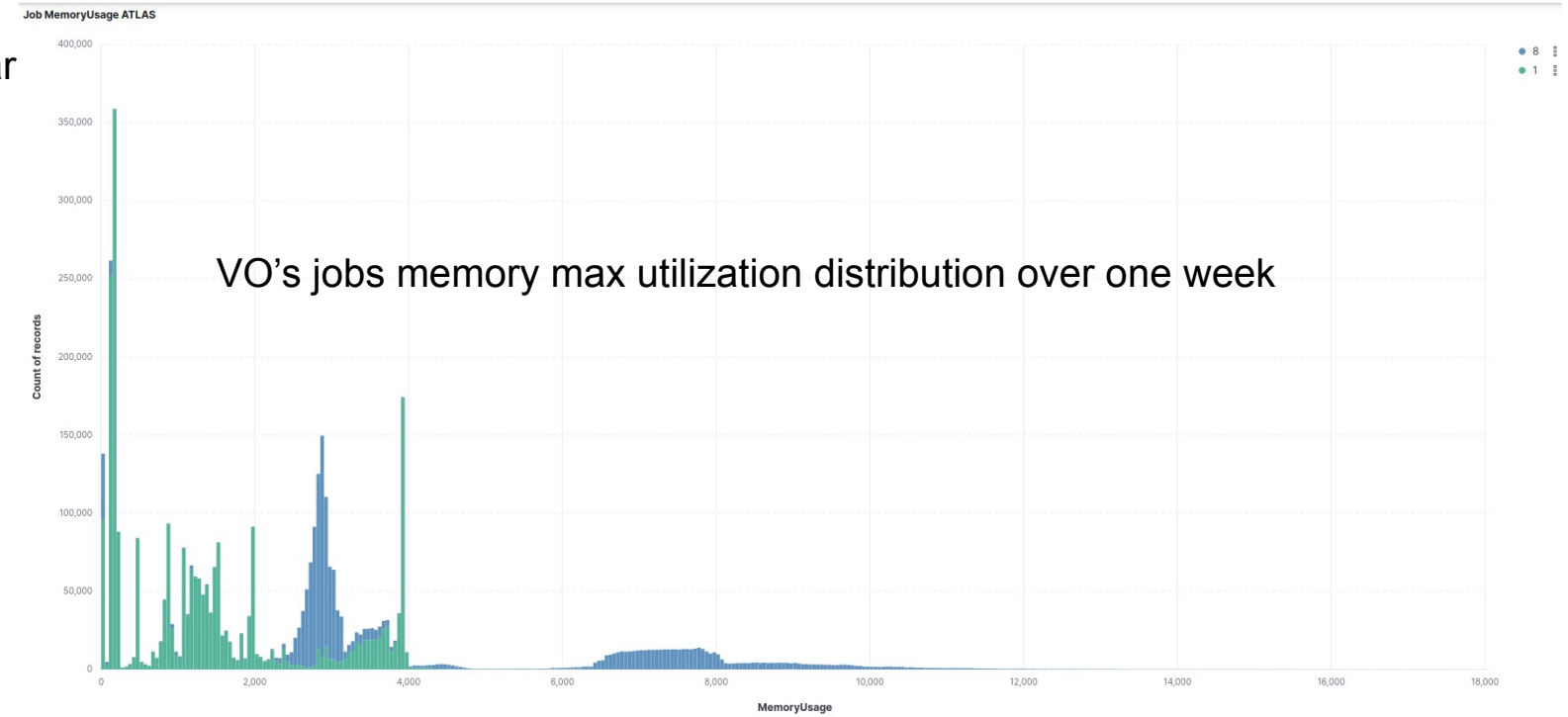
Summary

- Hitting the limits of the current hard- and software setup
- Biggest optimization potential = more effective/intelligent file access by the users
- Hard- and software upgrades in the budget for 2024
- Will see if NFS in EL9 is pushing boundaries in any way

News from the GRID

Living in interesting times

- Migration to EL9 ongoing
  - Germinal EL9 cluster deployed – migrating nodes from EL8 legacy cluster until EOL EL7
  - No accounting/middleware, i.e., running dark wrt. accounting
- New plans for higher memory jobs, 16 core pilots and +4 day runtimes question the current scheduling model
  - Decreasing entropy & no runtime estimates for pilots twart effective scheduling & badput minimization
- Status of on-site experiments/VOs unclear wrt. the Grid (WLCG? EGI?)



VO’s jobs memory max utilization distribution over one week

OS & HTCS upgrade NAF

Nothing very surprising here

- Skipping EL8 (like most sites) apart from 1 or 2 workgroupserver for CMS
- Direct upgrade to EL9
- Current (site)license agreement with RedHat to our favour
- Worker (EP) will be RHEL9
- Some ALMA9 workgroupserver optional
- HTCS LTS (23.0.8)

- ID tokens as the main tool for daemon to daemon communication (was kerberos)
- Ongoing support for KRB & AFS (token shepherding etc)
- Overall setup with few ‘big’ scheds (native GPFS), remote submit from numerous WGS and multiple shared FS (NFS) for data access stays the same
- Some (few) EL9 worker in the old pool established for testing
- New pool is built in parallel because some major config rewriting is necessary in order to tidy up everything ;)

Follow-up on power modulation

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

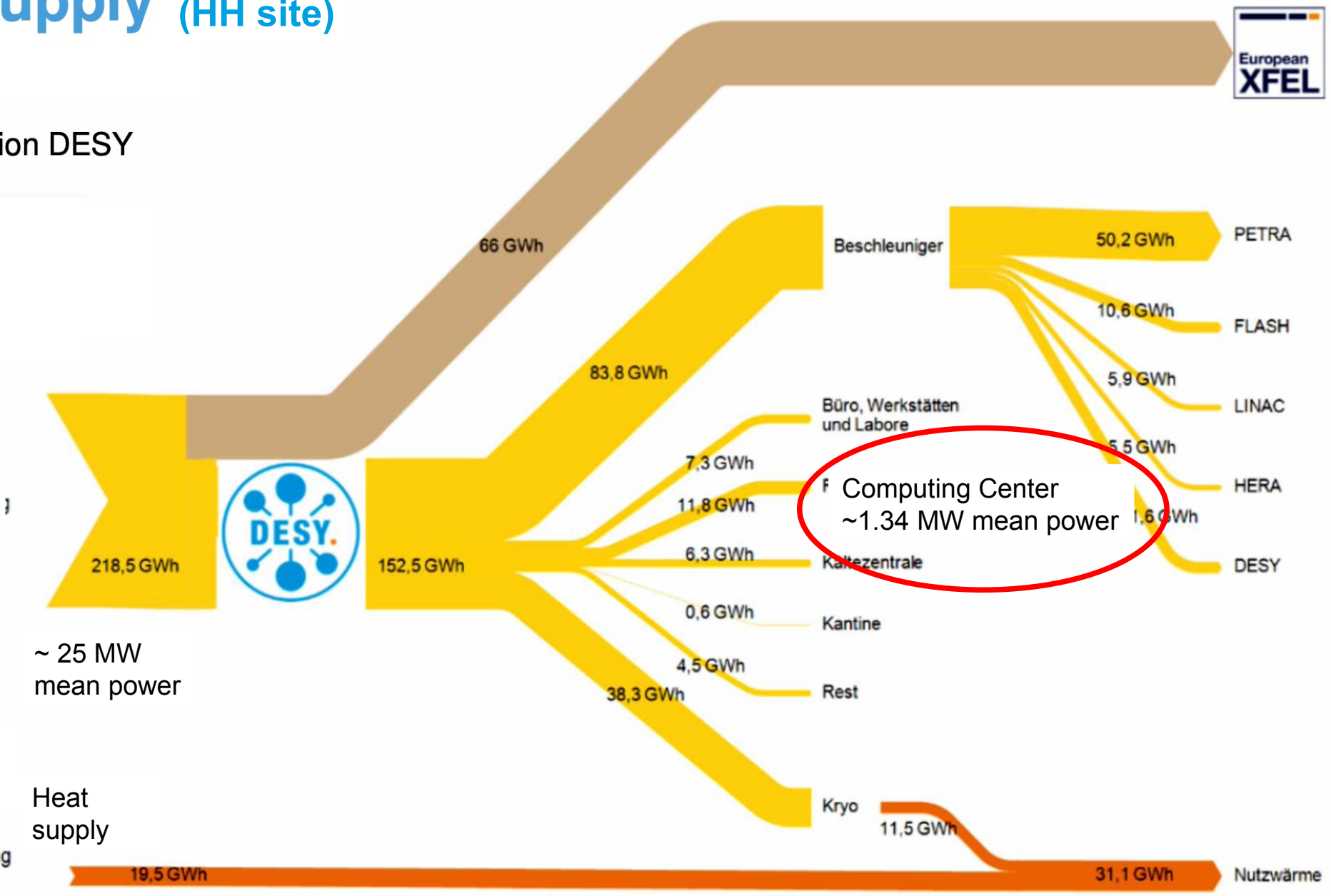
- Energy prices now more or less same as before the crisis + long running contracts at DESY
- Preparation for ‘it’ prooved to be useful anyway and did lead to a clearer sustainability concept
- Time for immediate action is over
- Time to design and build really sustainable research infrastructures

Energy supply (HH site)

Overview

Power consumption DESY 2021

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



Electricity supply

~ 25 MW mean power

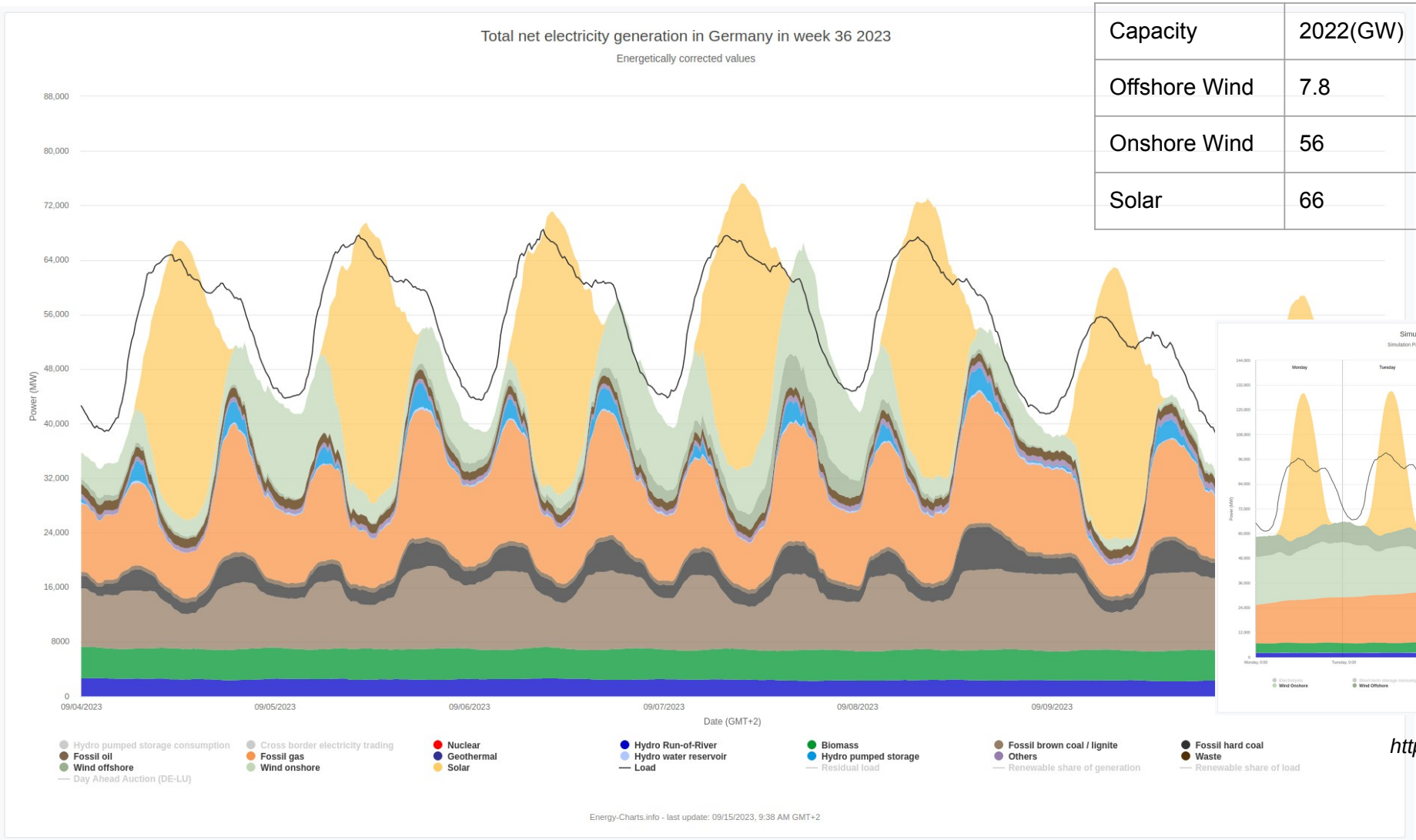
Computing Center ~1.34 MW mean power

Heat supply

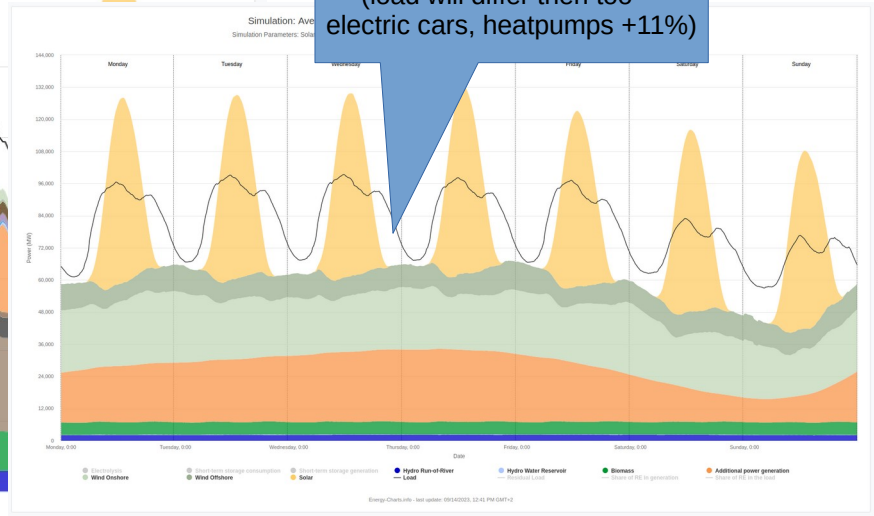
Slide: Helmut Dosch & Denise Völker

Public net electricity generation in Germany week 36 2023 & 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

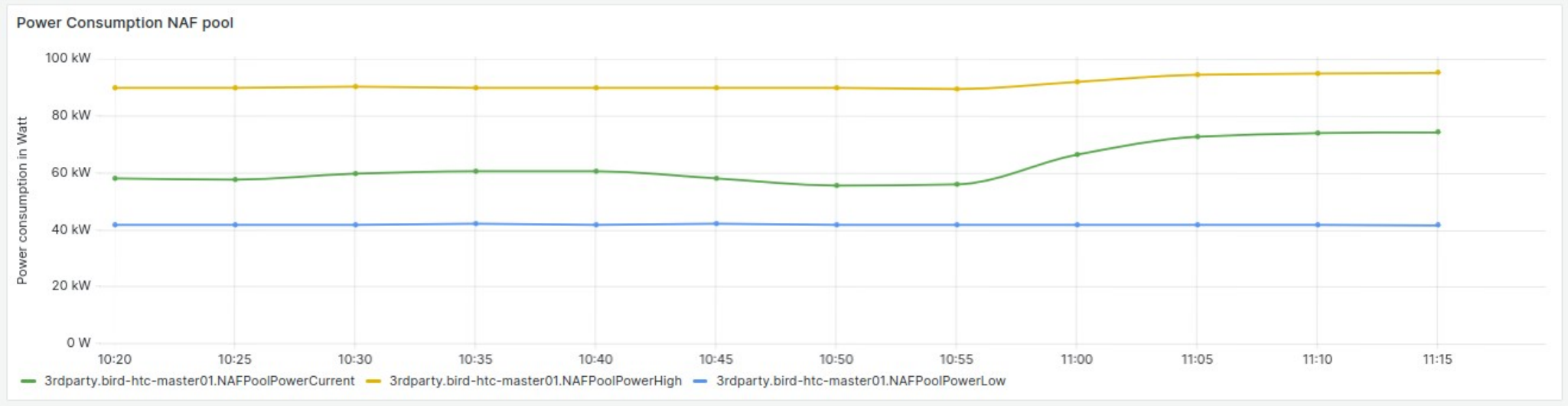


https://www.energy-charts.info/index.html

Summary for NAF power modulation

Tools and monitoring are there

- The current contract for DESY electricity consumption mainly focuses on peak usage which is extremely costly no other monetary impact
- This and the absence of variable power sources like green vs. fossile energy sets hard limits to what we can do
- All tools to steer and monitor the power consumption are there though !
- Power-down of idle nodes as effort to reduce C02 footprint
- User awareness is raised by workshops on sustainable computing with good feedback



Managing a small entity inside a bigger one

GPUs in the NAF as an example

- 14 GPUs are part of the NAF pool which consists of ~ 10k cores
  - Makes 14 slots with a special property amongst potentially 10.000 slots
- Access to GPUS limited to a fraction of users using a registry resource but still > 600 accounts
- Still negotiating the GPU slots without a separated quota scheme is random
  - For the negotiator the GPU slot is not different to any other slot
  - One option: ‘concurrency_limits’ (meant for limited licensed software initially)
    - Would result in a lot of bad-put as the usage of the GPUS is very spiky
    - One user would not be able to claim all GPU slots even if they are unused
  - Another option: manage a separate, second quota scheme for GPUS
    - A lot of work initially and ongoing, actually doubling the effort around quotas
    - Smaller user group entities tend to appear and cause recalculation and editing of the quotas
    - Condor is not very good in managing quotas on a very small scale (14 slots vs. 1k users)
- Biggest problem here – fairshare is calculated over 10.014 slots, hence no fairshare on the GPU slots
  - Absence of fairshare very visible for users

Possible solution – a separate/2nd negotioator

Not solving all the problems but pretty good approach for us

- A 2nd negotiator can be established (even on the same machine) explicitly managing slots and jobs with a special property
  - Based on the overall quotas (no separate editing)
  - Ignoring the majority (non-GPU) jobs
  - Keeping the fair share on the small entity !

```
GPU_NEGOTIATOR = $(NEGOTIATOR)
GPU_NEGOTIATOR_ARGS = -local-name gpu_negotiator
DAEMON_LIST = $(DAEMON_LIST) GPU_NEGOTIATOR
DC_DAEMON_LIST = + GPU_NEGOTIATOR
NEGOTIATOR.GPU_NEGOTIATOR_LOG = $(LOG)/NegotiatorLog.gpu
GPU_NEGOTIATOR_MATCH_LOG = $(LOG)/MatchLog.gpu
GPU_NEGOTIATOR.NEGOTIATOR_LOG = $(GPU_NEGOTIATOR_LOG)
GPU_NEGOTIATOR.NEGOTIATOR_MATCH_LOG = $(GPU_NEGOTIATOR_MATCH_LOG)
GPU_NEGOTIATOR.NEGOTIATOR_JOB_CONSTRAINT = $(IS_JUPYTER_JOB) =?= false &&  (RequestGpus =!= UNDEFINED && RequestGpus >= 1)
GPU_NEGOTIATOR.NEGOTIATOR_SLOT_CONSTRAINT = (GPUs =!= UNDEFINED && GPUs >= 1)
```

JUPYTER notebooks -

Not longer the new kid on the block

How we envisioned – and implemented it

- Jupyterhub bridging the NAF into the WAN
- Small reserved slots for notebooks sufficient on the NAF workers
  - 1 core 1,5 GB memory
  - Soft policy, notebook stopped if mem-usage > 4,5 GB
- Fast start of notebooks due to separate negotiator/collector (<10 secs)
- Users use htmap and python bindings to outsource workload into the pool
- ‘Older’ VO’s like ATLAS and CMS will adapt to jupyter notebooks and it will become a default mean of access to the NAF
- BELLE will heavily rely on notebooks as they are widely accepted in their community
- Debugging issues beyond the notbook start itself will be time consuming and python knowledge will be necessary

JUPYTER notebooks – insights after 2 years of usage

A mixed bag

Reality strikes again

- Jupyterhub bridging the NAF into the WAN ✓
- Small reserved slots for notebooks sufficient on the NAF workers ✗
  - 1 core 1,5 GB memory
  - Soft policy, notebook stopped if mem-usage > 4,5 GB
- Fast start of notebooks due to separate negotiator/collector (<10 secs) ✓
- Users use htmap and python bindings to outsource workload into the pool ✗
- ‘Established’ VO’s like ATLAS and CMS will adapt to jupyter notebooks and it will become a default mean of access to the NAF ✗
- BELLE will heavily rely on notebooks as they are widely accepted in their community ✓
- Debugging issues beyond the notebook start itself will be time consuming and python knowledge will be necessary ✓

JUPYTER notebooks

Summary & outlook

- User want to scroll through bigger amounts of data and in general are not prepared to outsource any workloads
- Bigger notebooks in terms of memory only partly a solution (similar to fixing a memory leak) ;)
- Will provide 3 classes of notebooks in the future (similar to SWAN @ CERN)
  - 12/14/16 GB Memory 1/2/2 cores
  - Dropdown chooser in jhub
- Jupyter-resource-usage (pip install) provides a nice memory reminder in the upper right corner of the notebook
- A complete new class of notebooks with freely configurable specs running in the regular pool might be an option but start-up and availability would be tied to the general quota and priority of the user.
  - Maybe e-mail notification when notebook ready
  - Should not be too comfortable otherwise people ignore the pre-configured lightweight ones

New users – different problems

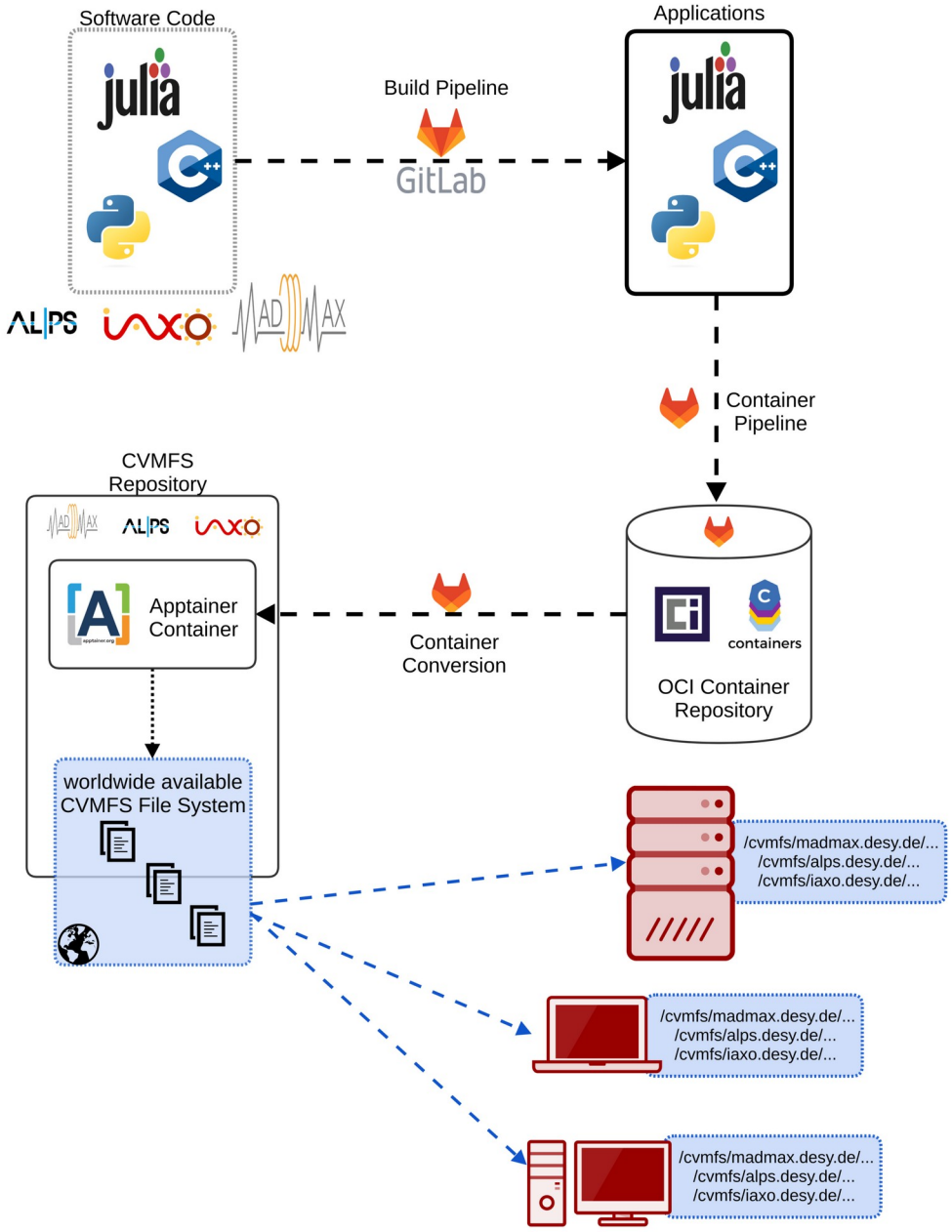
Smaller, new experiments do not have the IT related infrastrucutre we are used to

- New, smaller experiments are run by groups of 10 to 20 people
- No computing history or background as such and no computing coordinator
    - Well that is not completely correct – there is one of them who wears the hat, but …
- They do not get the ‘automatic’ introduction and best practice like the users joining the bigger Vos like CMS or ATLAS
- We were doing kind of NAF-school type meetings in the past but that was uniquely with scientists who had at least basic experiences in distributed computing
- New groups live in windows and python but basic ideas and best practice in distributed computing cannot be assumed
- It took us a while to adapt and offer individual low-threshold training
- Once the penny dropped very good feedback and another (group of) satisfied customer
- Will offer this low-threshold training on a regular base combined wit questions and answers

New users – new chances

Smaller, new experiments do not have the IT related infr

- The absence of previous distributed computing knowledge of course also comes with the chance to teach people to do it right from the beginning :)



This is a very nice meeting here !

BUT – have you been to Amsterdam ???

The European HTCondor workshop Autumn 2024

Sep 24 – 27, 2024 – Nikhef, Amsterdam