GridPP
UK Computing for Particle Physics

UNIVERSITY of GLASGOW

UK Flat Cash

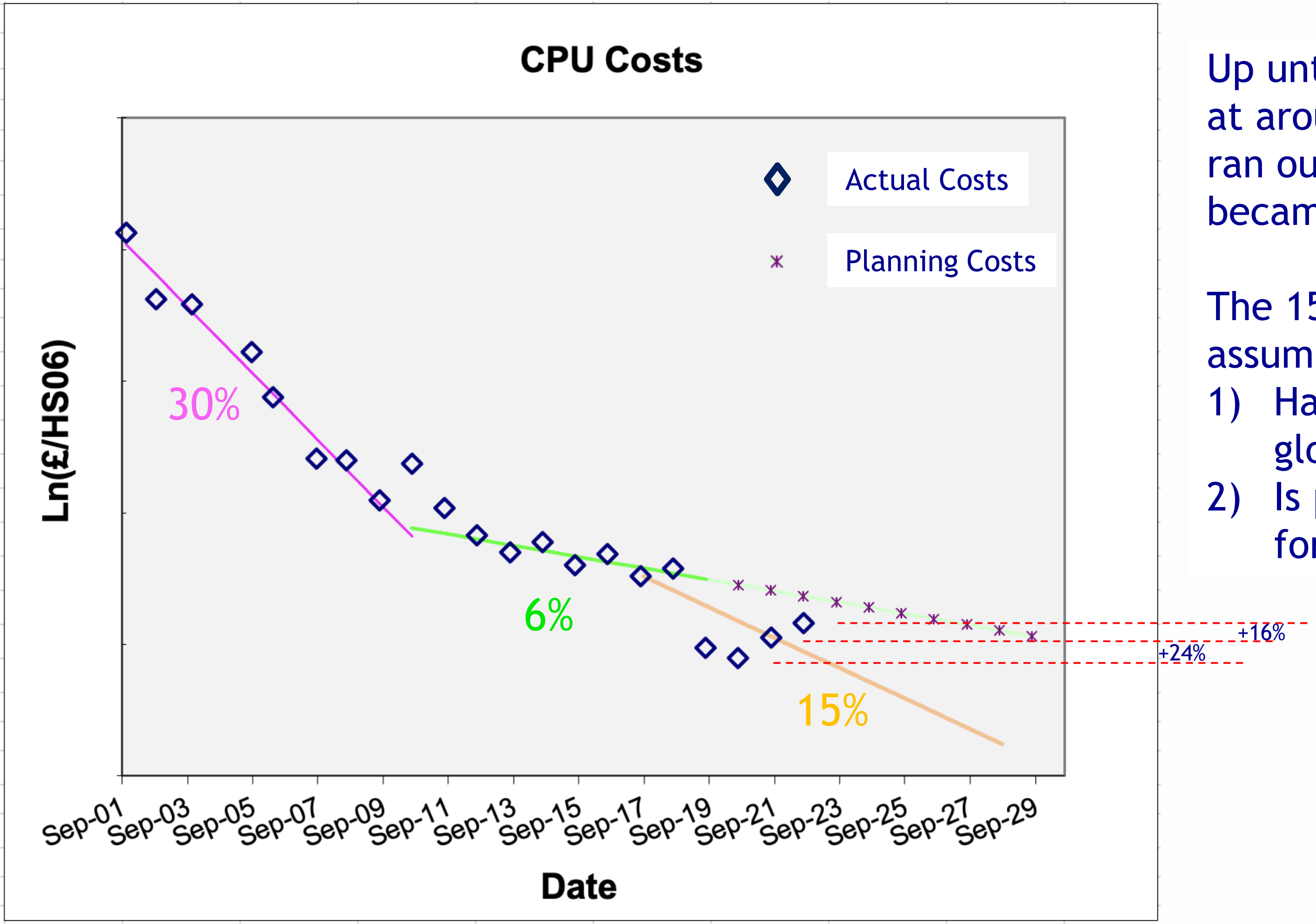
Thoughts on Power Costs

WLCG Workshop
9th Nov 2022

David Britton
GridPP Project Leader
University of Glasgow

Part-1: Hardware cost trends in the UK

20 Years of RAL CPU Costs

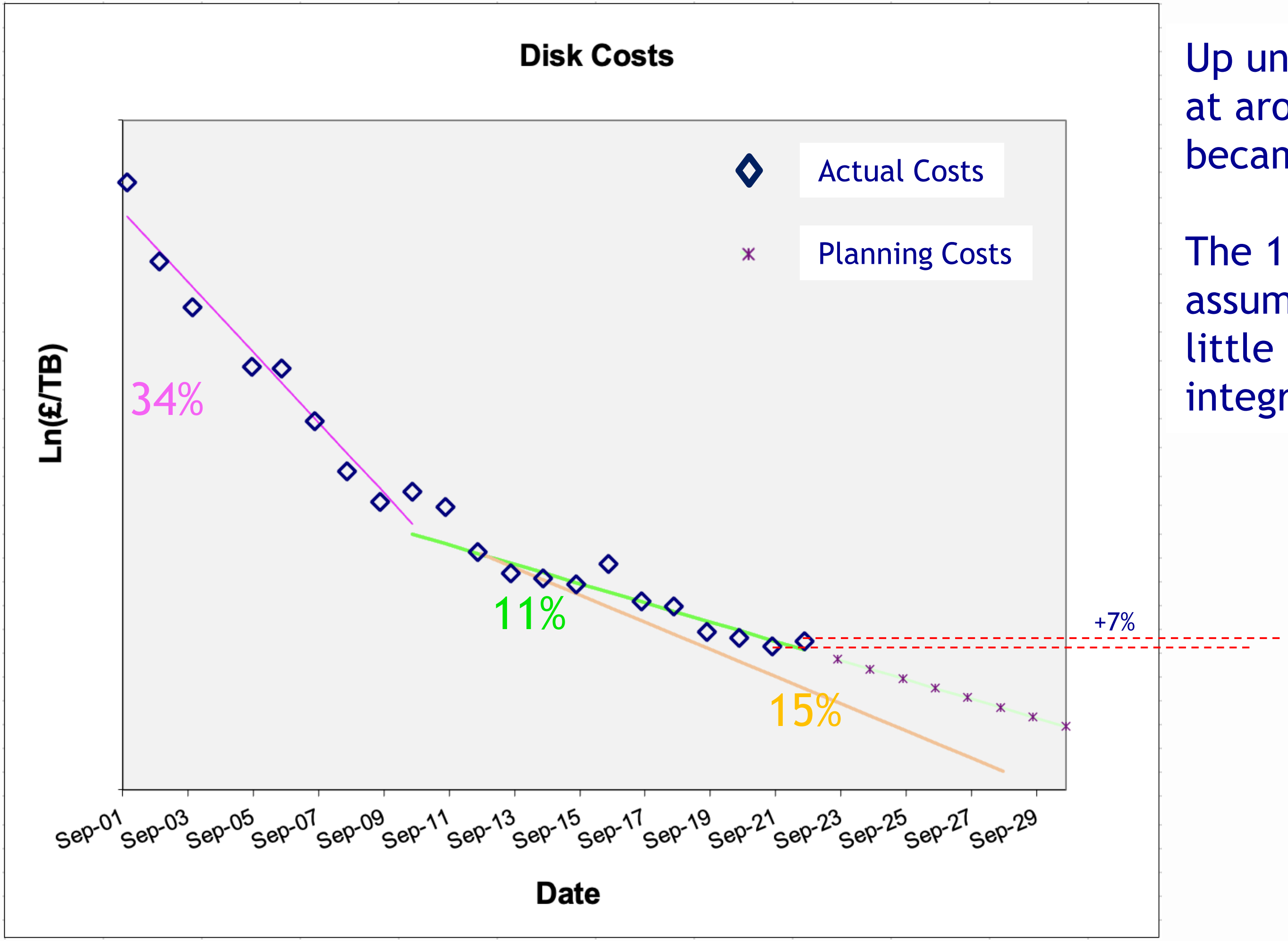


Up until 2009, CPU costs fell at around 30%/year. As we ran out of "clock speed" this became more like 6%/year.

The 15% **"flat-cash"** assumption:

1) Has proved sensitive to global events.
2) Is probably too optimistic for the UK.

20 Years of RAL Disk Costs



Up until 2009, Disk costs fell at around 34%/year and then became more like 11%/year.

The 15% “**flat-cash**” assumption is probably just a little optimistic once integrated over several years.

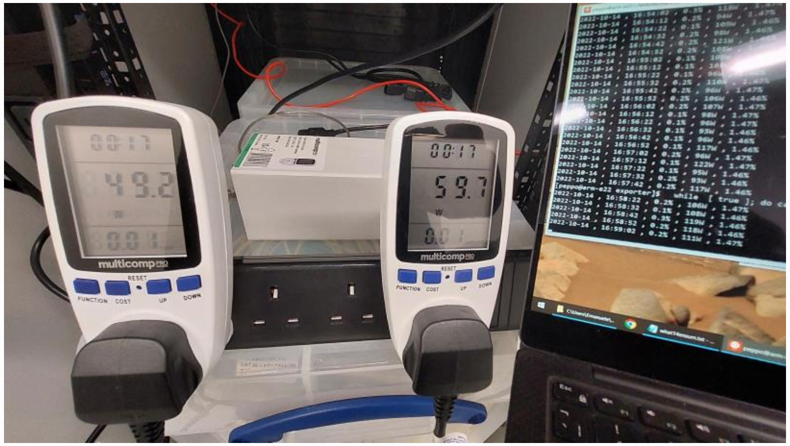
Part-2a: Power savings using ARM

GridPP
UK Computing for Particle Physics

ARM vs X86

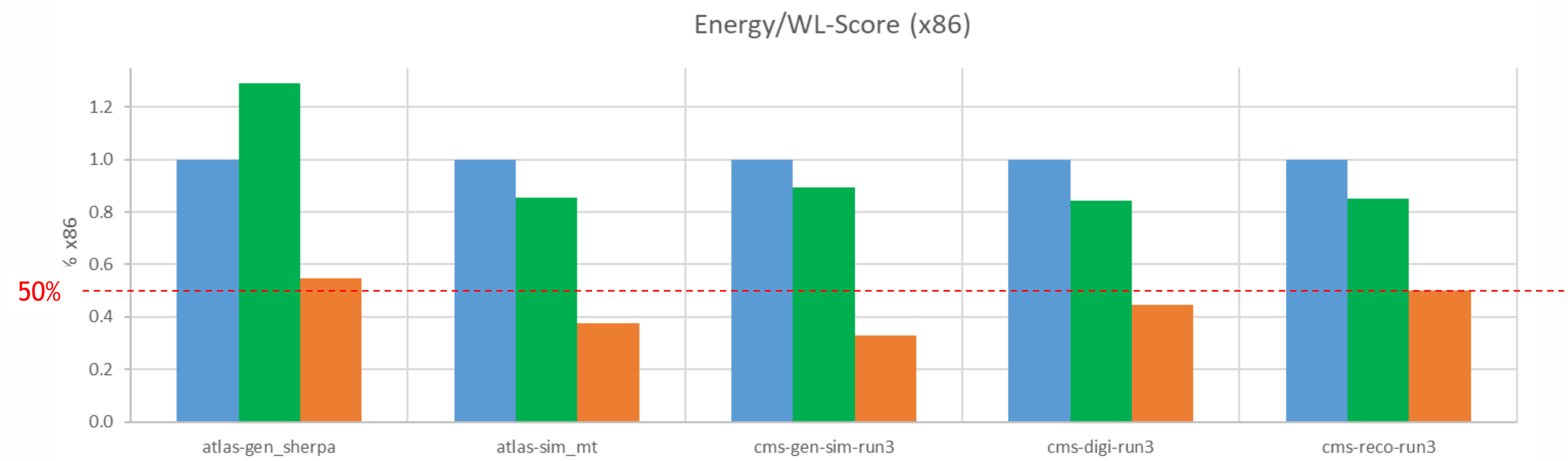
- Compared power consumption of two ~similar spec-ed and priced machines: X86_64 and ARM_64 (see back-up slide for details).
- Most relevantly, ran the available ATLAS and CMS HEPSCORE benchmarks. This means the images were compiled (presumably ~optimally) for the two architectures by the experiments.
- Power consumption monitored from IPMI interface and and validated (at 5% level for now) with metered-plugs.

See ACAT presentation by Emanuele Simili: https://indico.cern.ch/event/1106990/contributions/4991256/



Energy/WL-Score: ARM vs X86

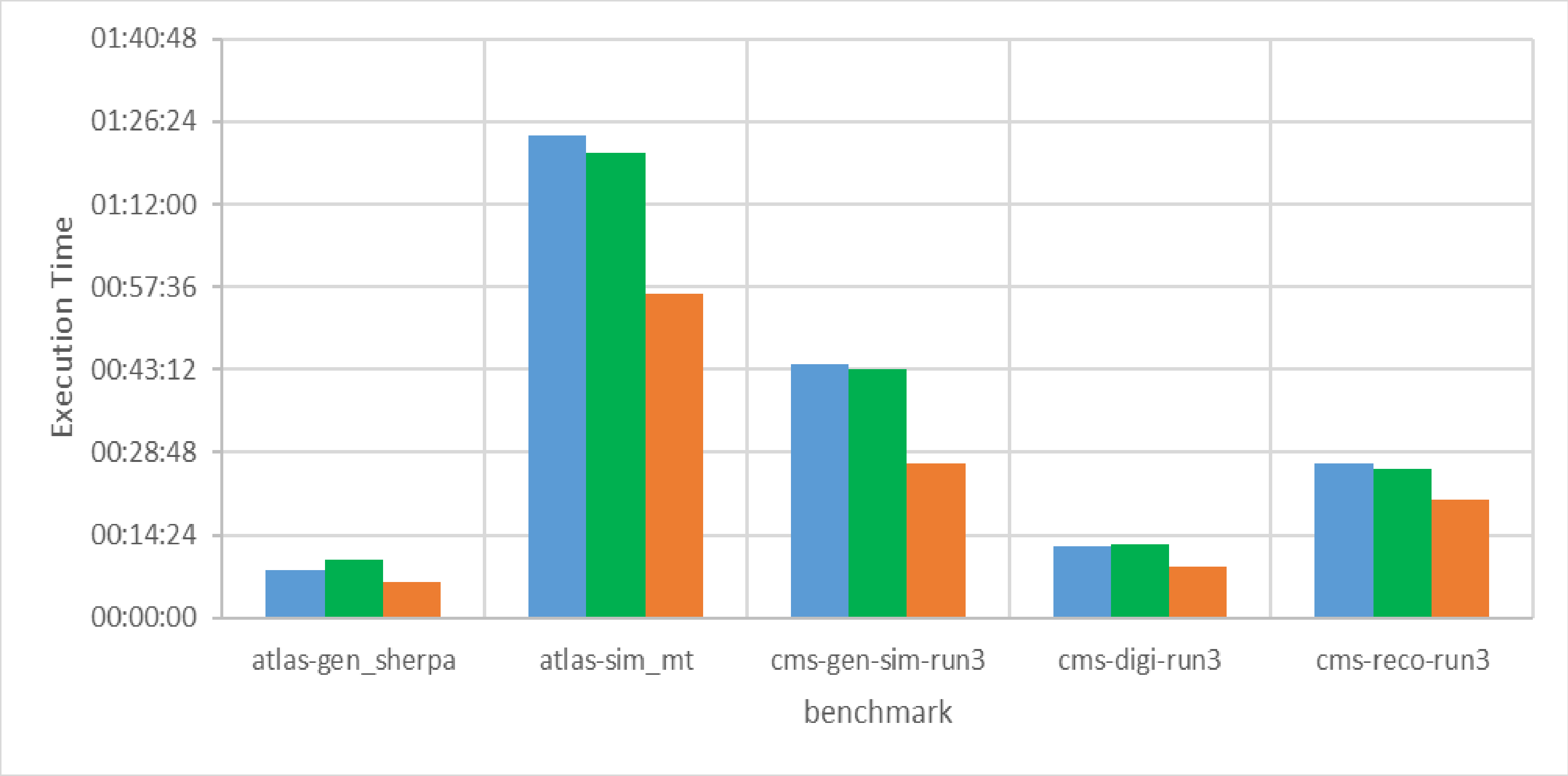
- Measured energy-used per unit WL-Score for 2 ATLAS and 3 CMS HEP-Score workloads.
- Three test machines:
  - x86_64: Single AMD EPYC 7003 series (SuperMicro)
  - 2*x86_64: Dual AMD EPYC 7513 series Processors (DELL)
  - **arm64: Single socket Ampere Altra Processor (SuperMicro)**
- Normalised to (first) x86. Lower is better!



Execution time: ARM vs X86

- x86_64: Single AMD EPYC 7003 series (SuperMicro)
- 2*x86_64: Dual AMD EPYC 7513 series Processors (DELL)
- **arm64: Single socket Ampere Altra Processor (SuperMicro)**

Lower is better!



Comments on ARM

- Still need some physics validation of ARM (e.g. ATLAS Reco)
  - 1-3 million-core run (under discussion)
- Power-use balance may shift again with each generation
  - Need to maintain agility.
- Competition is healthy!
- HEPScore should also measure power-consumption
  - Two-dimension optimisation between computational power and power usage (discussed with Domenico).

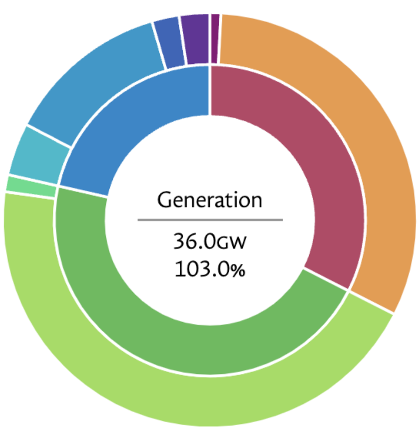
Part-2b: Load Shaping

- Responding to supply shortages
- Reducing carbon footprint by using power when there is a bigger fraction of renewable energy.

Load Shaping

- If we reduced computational workloads during periods of high fossil-fuel power generation, how much does it reduce our carbon footprint?

Generation



Note: percentages are relative to demand, so will exceed 100% if power is being exported

33.4% fossil fuels

Coal	0.31GW	0.9%
Oil	0.00GW	0.0%
Gas (open cycle)	0.00GW	0.0%
Gas (combined cycle)	11.41GW	32.6%

47.4% renewable energy

Solar photovoltaic	0.00GW	0.0%
Wind	16.13GW	46.0%
Hydroelectric	0.48GW	1.4%

22.1% other energy

Pumped storage	1.48GW	4.2%
Nuclear	4.63GW	13.2%
Biomass	0.77GW	2.2%
Other	0.86GW	2.5%

-2.9% imports and exports

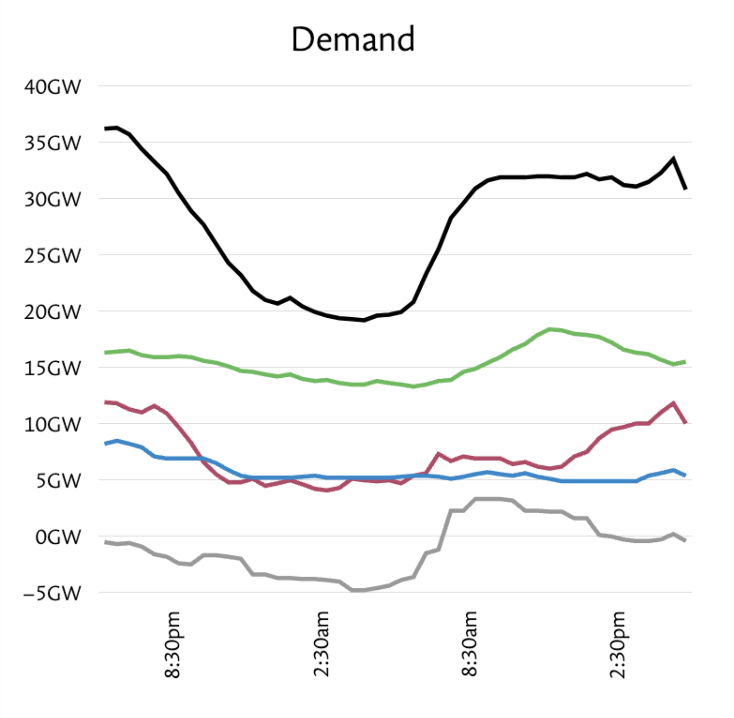
Belgium	0.62GW	1.8%
France	-2.53GW	-7.2%
Ireland	0.90GW	2.6%
Netherlands	0.20GW	0.6%
Norway	-0.23GW	-0.6%

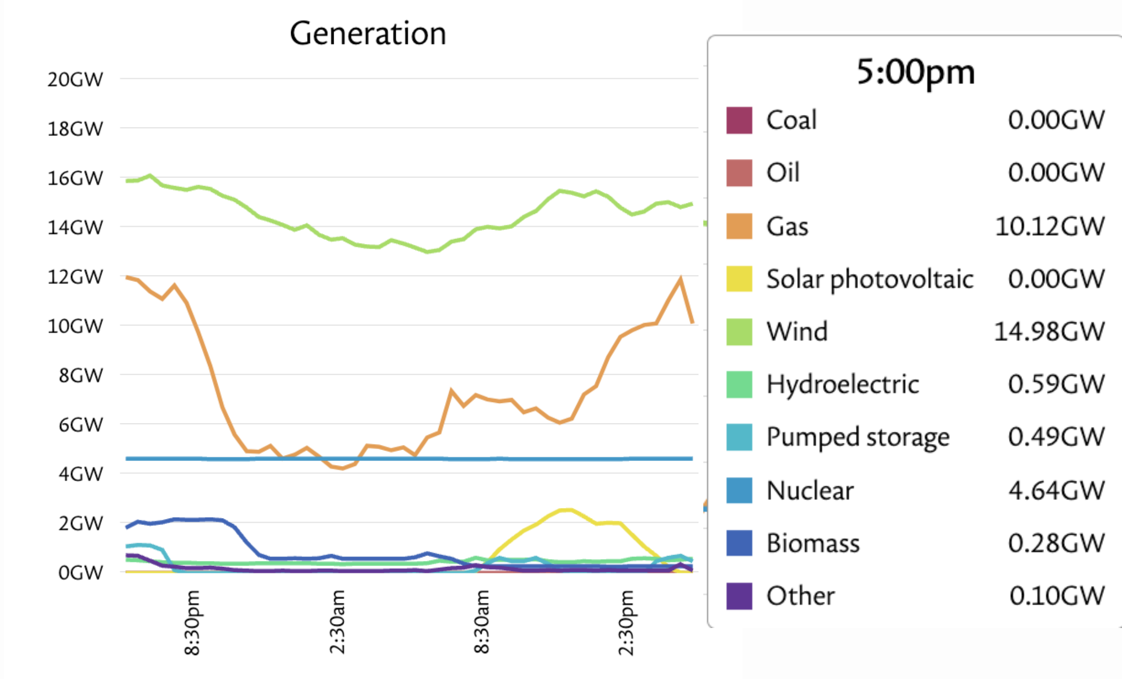
UK Power Generation (3pm, 2/Nov/22)

Source: https://grid.iamkate.com/

UK Power Demand

Source: https://grid.iamkate.com/





Proof of Principle

- Looked at 1 year of UK power generation data.
- Assume we turned off compute every time fossil-fuel generation exceeded a certain threshold (e.g 20 GW).
- Calculate our down time / loss of work.
- Calculate reduction in our carbon footprint by two methods:
    1. Using average fossil-fuel fraction of power saved.
    2. Assuming that all power saved would have been dirty-power (Brownwashing™).

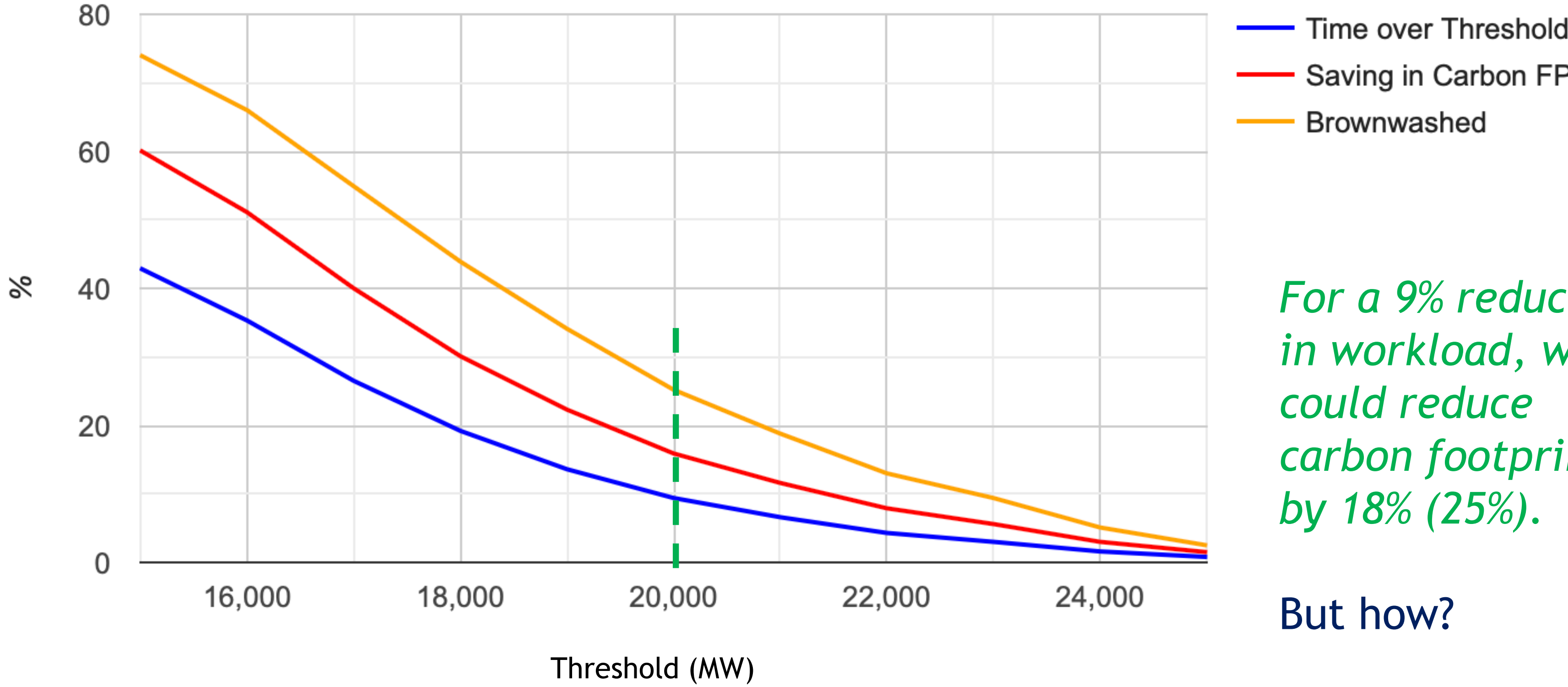
Example:

a) Run all the time and use 100 units of power, 50 units from fossil-fuels.

b) Turn off 10% of the time when fossil-fuel fraction is highest; lets say fossil-fuel generation accounted for 75% of power generated when we were off:
    1. 7.5/50 = 15% (7.5 is 75% of 10 units of power we avoided using while down)
    2. 10/50 = 20% (10 is 100% of 10 units of power we avoided using while down)

Carbon Footprint Reduction

Fractional Reduction in Carbon Footprint (Red or Orange) as a fraction for range of down-times (blue) calculated by applying a threshold cut to UK power data (Oct 2021 – Sep 2022)

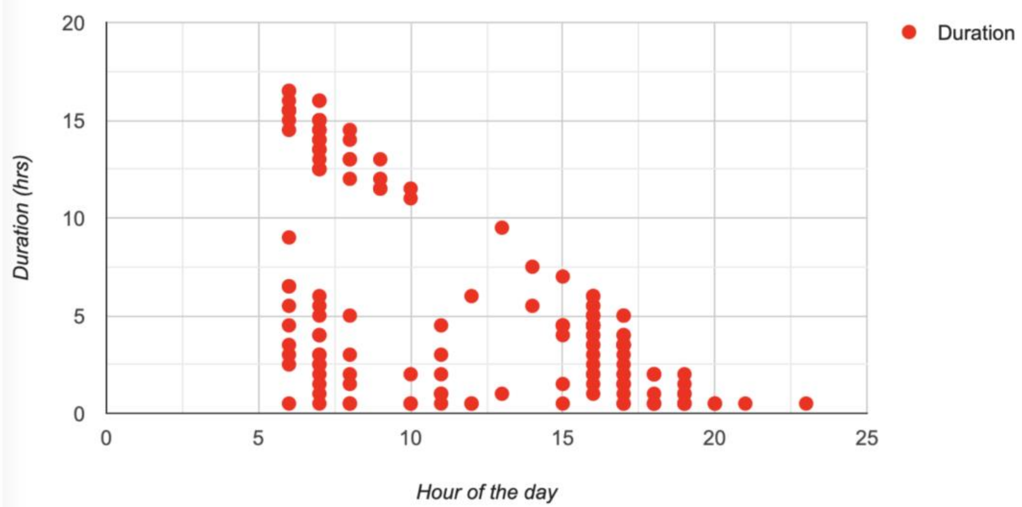


For a 9% reduction in workload, we could reduce carbon footprint by 18% (25%).

But how?

Figure made in collaboration with Steve Lloyd

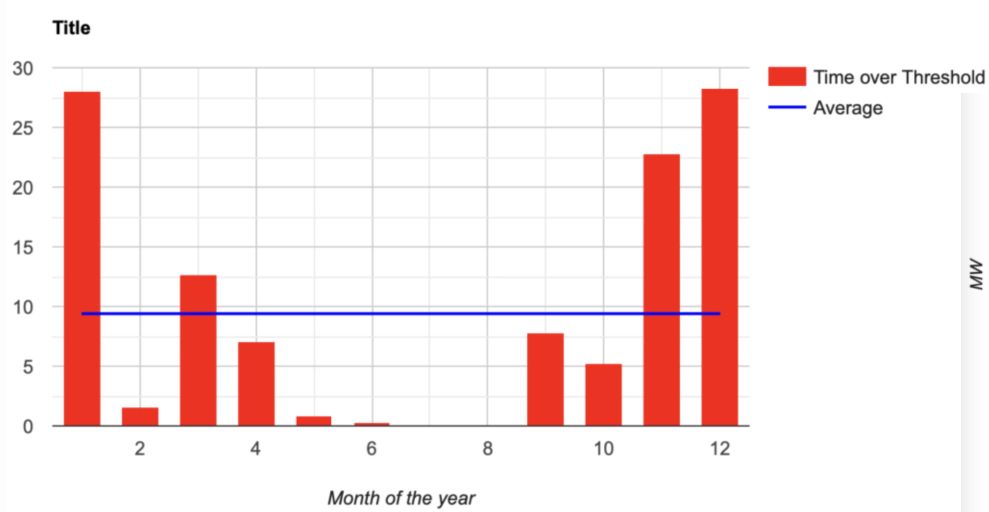
Distributions

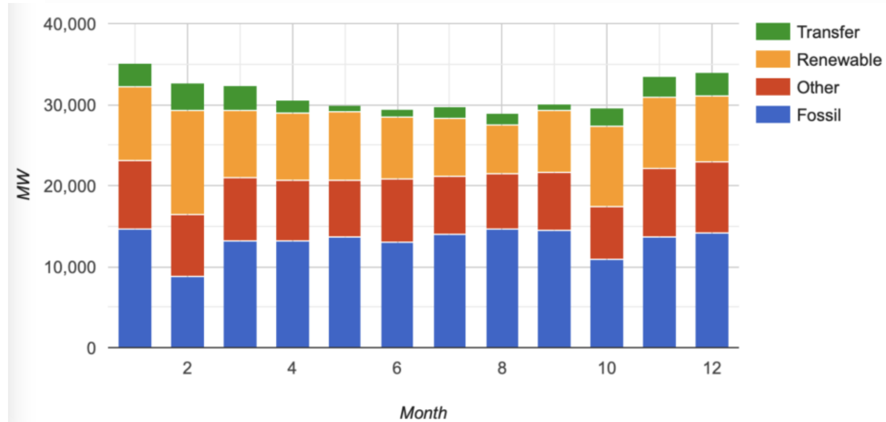


Down-time is bi-modal (~4h and ~14hrs) but ~never extends overnight (low demand). So jobs unlikely to get suspended for days.

Down-time is zero in the summer and peaks in December/January.

Figures made in collaboration with Steve Lloyd





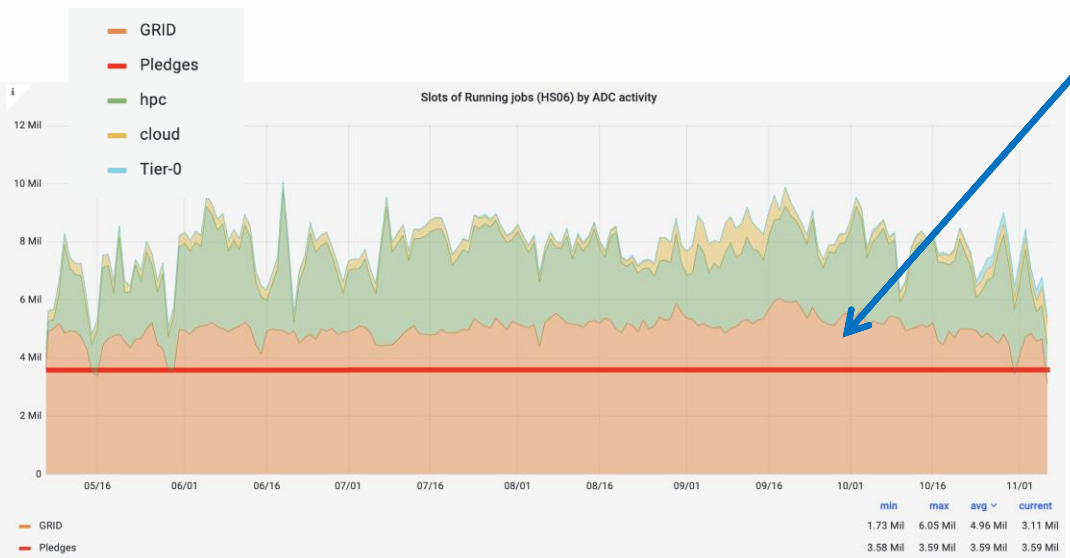
But how?

- Of course, you can’t just turn off the compute and switch it back on later.

- Could sites run at reduced clock-speed? (Rod’s talk?)
- Could experiments control jobs via pilot framework?
- Could experiments checkpoint/snapshot jobs? *(ATLAS: “Not impossible”)*
- Could sites predict periods of high fossil-fuel generation and restrict queues?
- Could sites kill jobs that had only just started?
- Other ideas?
- Combination of ideas?

But what about the lost work?

- In principle, one could ask funding agencies for 10% additional hardware to allow an overall reduction in Carbon Footprint. Unlikely!
- In practice, the LHC experiments depend very much on opportunistic resources:



The UK delivers 2-3x Tier-2 pledged CPU partly using old hardware on the Grid.

Therefore, in some ways, we already HAVE the additional hardware.

Of course, old hardware is less energy efficient, so it is not as good a solution.

But this gives motivation for not simply “turning off hardware” as soon as it gets to 5-years (which in the UK case would mean the loss of a lot more than 10%) because it can be used to compensate for load-shaping. (Cunning-Plan™)

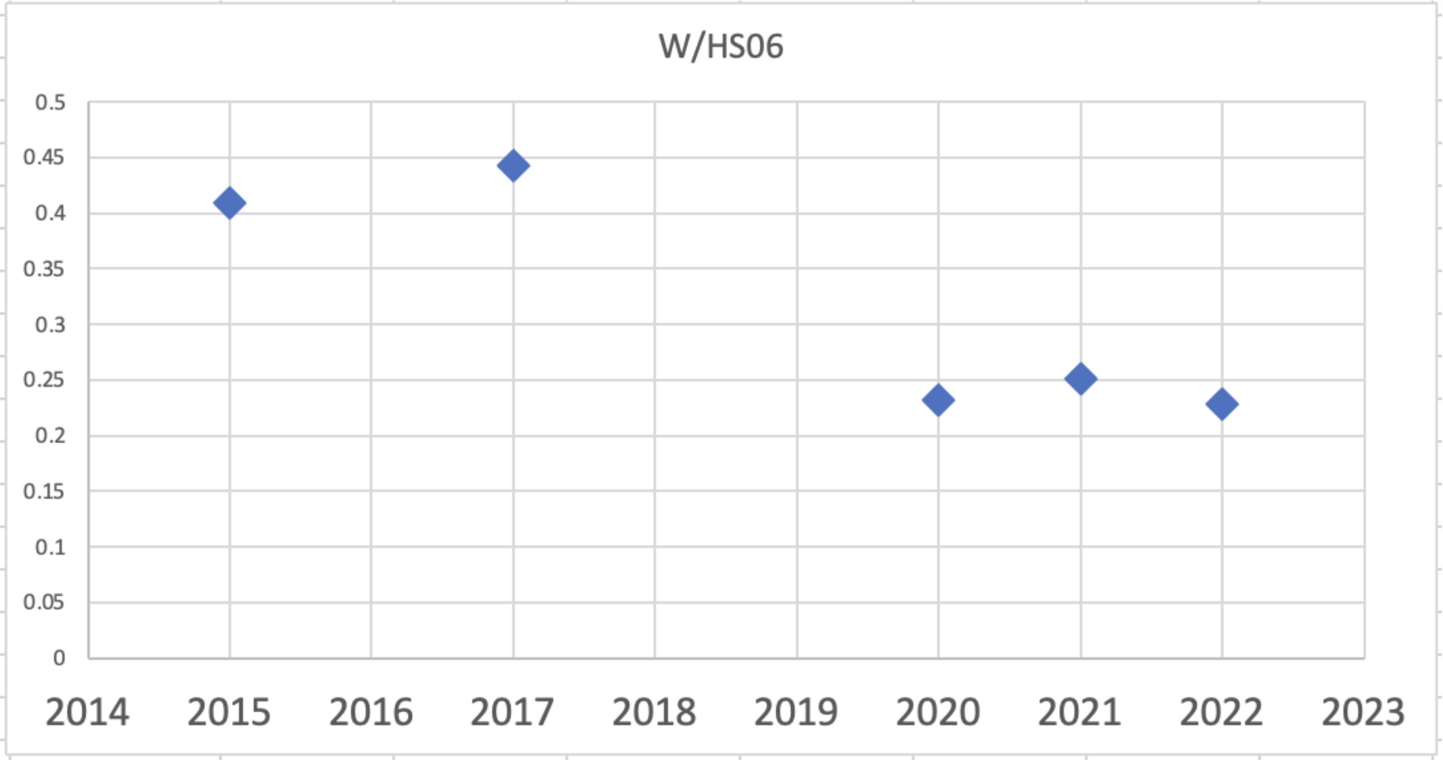
- Thanks to Emanuele Simili (Glasgow) for the ARM/x_86 work.

- Thanks to Steve Lloyd (GridPP) for collaborating on Load Shaping and making the plots.

- Thanks to Dwayne Spiteri (Glasgow) for taking an independent look at the UK power data and the issue of reporting Carbon savings, and Sam Skipsey (Glasgow) for helpful insights.

Back-up Slides

Power



Quick look at power consumption of different generations of Glasgow CPU (needs to be investigated more fully).

GridPP
UK Computing for Particle Physics

Hardware

We have recently purchased two almost identical machines of comparable price, one with an AMD x86_64 CPU (with HT enabled), the other with an Ampere arm64 CPU:

x86_64: Single AMD EPYC 7003 series (SuperMicro)
CPU: AMD EPYC 7643 48C/96T @ 2.3GHz (TDP 300W)
RAM: 256GB (16 x 16GB) DDR4 3200MHz
HDD: 3.84TB Samsung PM9A3 M.2 (2280)

arm64: Single socket Ampere Altra Processor (SuperMicro)
CPU: ARM Q80-30 80 core 210W TDP processor
RAM: 256GB (16 x 16GB) DDR4 3200MHz
HDD: 3.84TB Samsung PM9A3 M.2 (2280)

And we included in the comparison a standard workernode of our Grid cluster, with 2 AMD x86_64 CPUs (with HT enabled), which is also comparable in price* to the machines above:

2*x86_64: Dual AMD EPYC 7513 series Processors (DELL)
CPU: 2 * AMD EPYC 7513, 32C/64T @2.6GHz (TDP 200W)
RAM: 512GB (16 x 32GB) DDR4 3200MHz
HDD: 3.84TB SSD SATA Read Intensive

*** this machine is part of a 2 unit / 4 node chassis.**

ARM vs X86

HEP-Score Containers

HEP-Score containers can run on **Singularity** (or **Docker**, which we do not use):

- **(x86)** *Singularity-CE 3.9.9-1.el7* (previous version *3.8.7-1.el7* disappeared from EPEL and replaced with *AppTainer 1.1.0-1.el7*)
- **(arm)** *singularity version 3.8.4-1.el7* (still available in EPEL)

Example execution of a containerised HEP-Score job:

```
$ mkdir –p /tmp/test/results
$ chmod a+rw /tmp/test/
$ singularity run -B /tmp/test:/results oras://registry.cern.ch/hep-workloads/cms-gen-sim-run3-bmk:latest
```

We used 5 HEP-Score containers from the container_registry (prev. slide):

- gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif/atlas-sim_mt-ma-bmk:v2.0
- gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif/atlas-gen_sherpa-ma-bmk:ci-v1.0

} ATLAS (2x)

- gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif/cms-reco-run3-ma-bmk:v1.1
- gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif/cms-digi-run3-ma-bmk:v1.0
- gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif/cms-gen-sim-run3-ma-bmk:v1.0

} CMS (3x)