Glasgow's ARM cluster: Experience and Findings

Prof David Britton
Dr Gordon Stewart
Dr Samuel Skipsey
Dr Emanuele Simili
Dr Bruno Borbely
Dr Dwayne Spiteri

Journées LCG-France, CC-IN2P3 - 29/11/2023

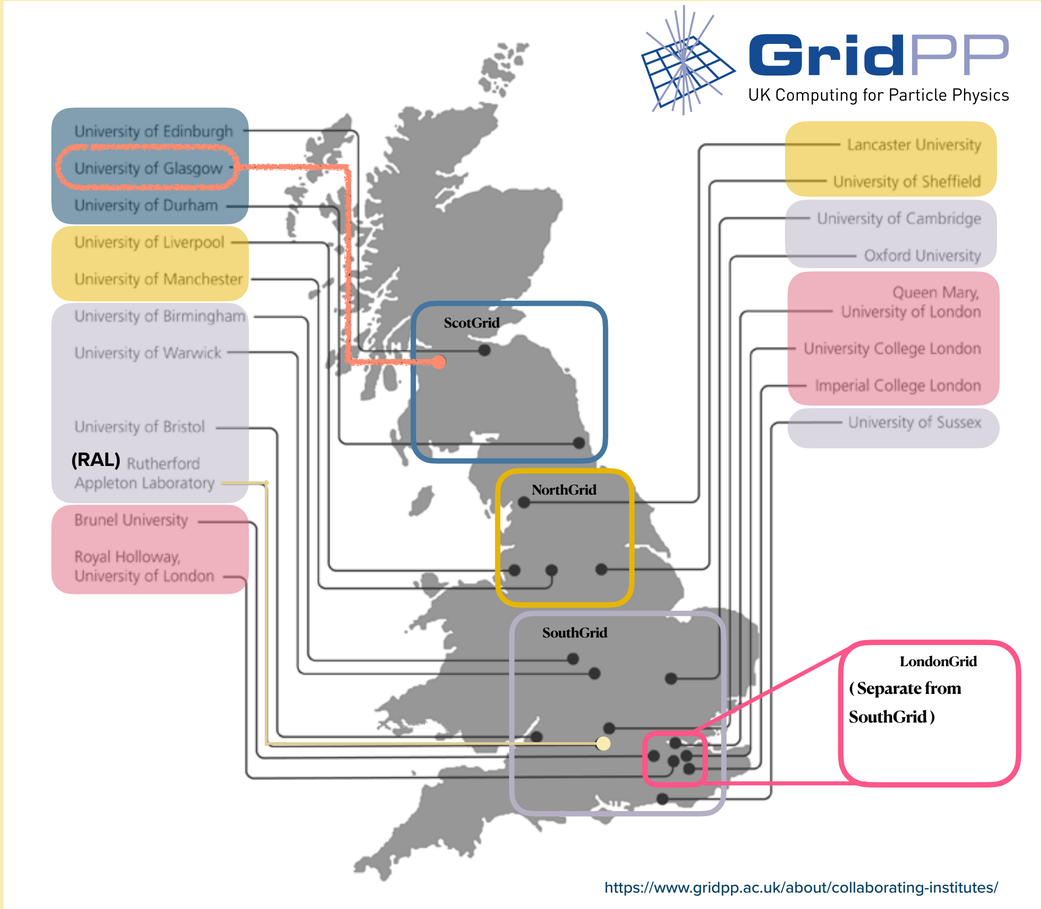
GridPP and Glasgow

- GridPP is the UK's branch of the WLCG. Its 18 Tier-2 sites and 1 Tier-1 (RAL) are split into 4 groups (ScotGrid, NorthGrid, SouthGrid and LondonGrid).

- Glasgow is the largest Tier-2 site in ScotGrid.

Glasgow Data-centre Highlights

- **Compute**
  ~15.5k x86 cores
  ~1.8k ARM cores
- **Storage**
  12PB raw storage
- **Networking**
  10 GbE to worker nodes
  10/40 GbE to storage
  40 GbE to the main campus
  2x10GbE to Janet (UK's inter-university system)



What we have

- After hearing about our work, AMPERE got in touch with us and kindly agreed to donate some ARM servers to us.

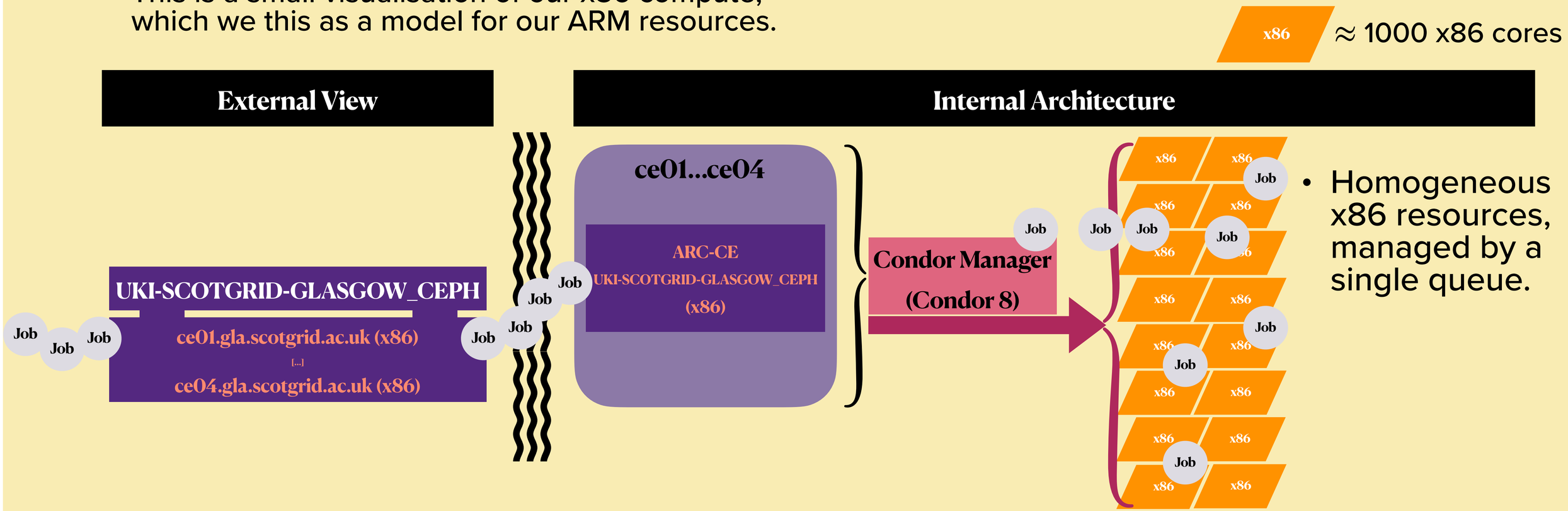
CPU	Ampere Altra Q80-30
Sockets	2
Max Frequency	3.0 GHz
Memory	512 GB

AMPERE

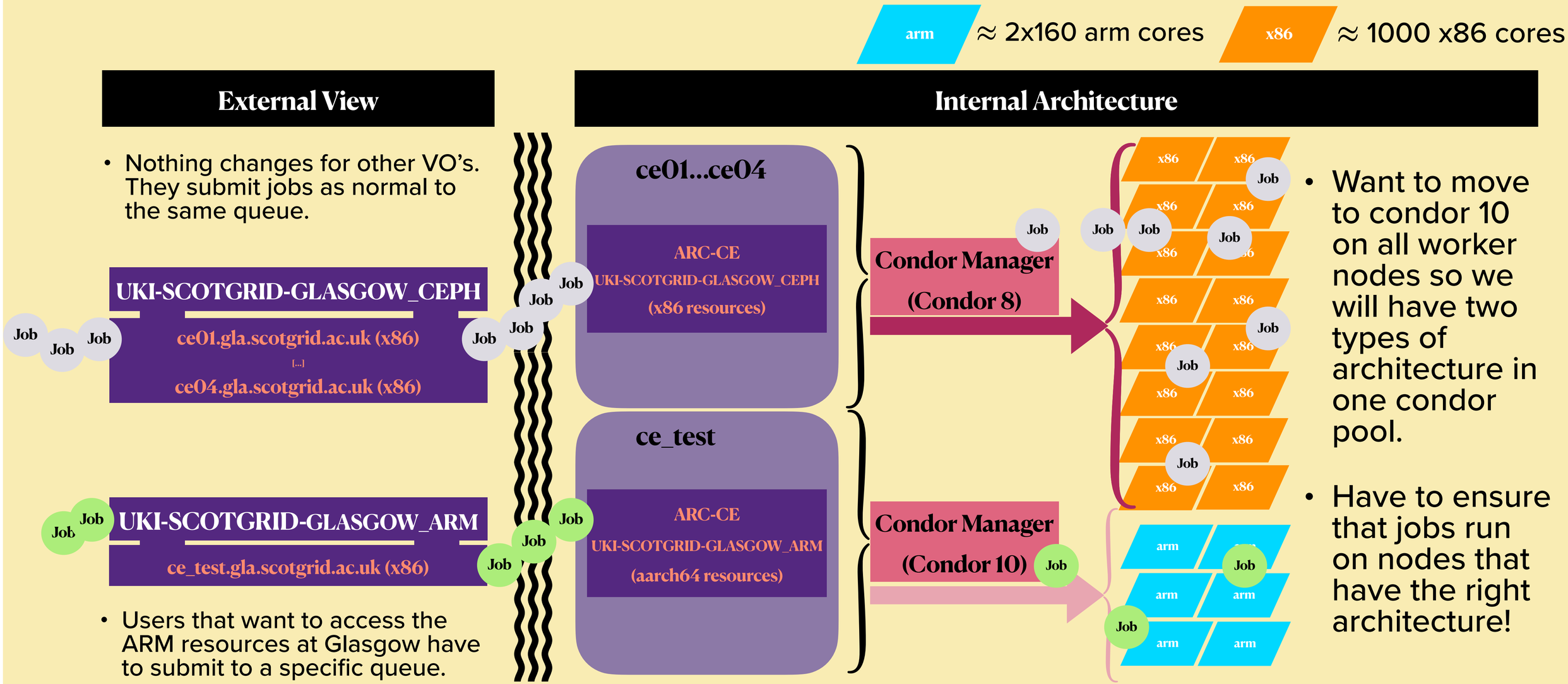
- With several 2U servers we now have 1760 active cores.

How do we get VO's to see and use our ARM resource?

- This is a small visualisation of our x86 compute, which we this as a model for our ARM resources.



How do we get VO's to see and use our ARM resource?



Technical setup differences between ARM and x86

- Our ARM nodes are on Rocky9 / HTCondor 10, attached to their own CE and Condor manager, while the rest of the site remains at Centos7 / HTC8 (this will change in the near future).
- While there are configuration changes between x86 to ARM, some changes are due the move from EL7 to EL9. By and large the ARM nodes are treated the same as x86 ones.
- For Provisioning, we currently use a bespoke PXE- / kickstart-based system (think Cobbler but targeted a little more closely at our needs) and that works across all our systems.
- Configuration management is via Ansible, and works across all hosts.
- Automatic updates are run across the site, with a couple of exceptions, and this is also true of the ARM nodes - patching normally involves checking applied updates rather than pushing them out. All possible through Ansible (SSH and Python underneath).

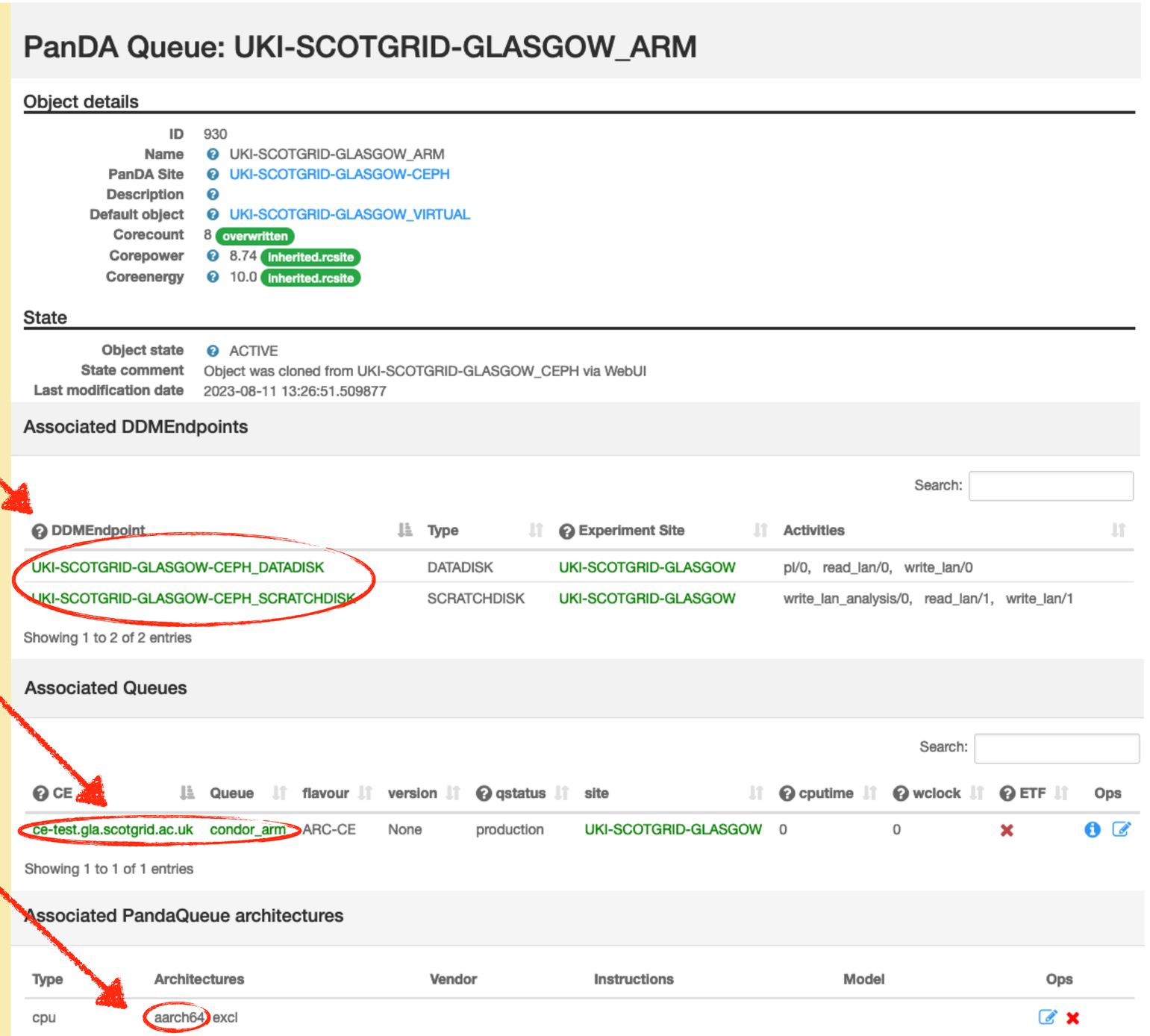
Setup: ATLAS-side

PanDA setup stores data locally (now)

Points to our temporary CE and our condor_arm queue in the ARC

aarch architecture only

- Now we just have to run some proper jobs on them



PanDA Queue: UKI-SCOTGRID-GLASGOW_ARM

Object details

ID	930
Name	UKI-SCOTGRID-GLASGOW_ARM
PanDA Site	UKI-SCOTGRID-GLASGOW-CEPH
Description	
Default object	UKI-SCOTGRID-GLASGOW_VIRTUAL
Corecount	8 overwritten
Corepower	8.74 inherited.rcsite
Coreenergy	10.0 inherited.rcsite

State

Object state	ACTIVE
State comment	Object was cloned from UKI-SCOTGRID-GLASGOW_CEPH via WebUI
Last modification date	2023-08-11 13:26:51.509877

Associated DDMEndpoints

Search:

DDMEndpoint	Type	Experiment Site	Activities
UKI-SCOTGRID-GLASGOW-CEPH_DATADISK	DATADISK	UKI-SCOTGRID-GLASGOW	pl/0, read_lan/0, write_lan/0
UKI-SCOTGRID-GLASGOW-CEPH_SCRATCHDISK	SCRATCHDISK	UKI-SCOTGRID-GLASGOW	write_lan_analysis/0, read_lan/1, write_lan/1

Showing 1 to 2 of 2 entries

Associated Queues

Search:

CE	Queue	flavour	version	qstatus	site	cputime	wclock	ETF	Ops
ce-test.gla.scotgrid.ac.uk	condor_arm	ARC-CE	None	production	UKI-SCOTGRID-GLASGOW	0	0		

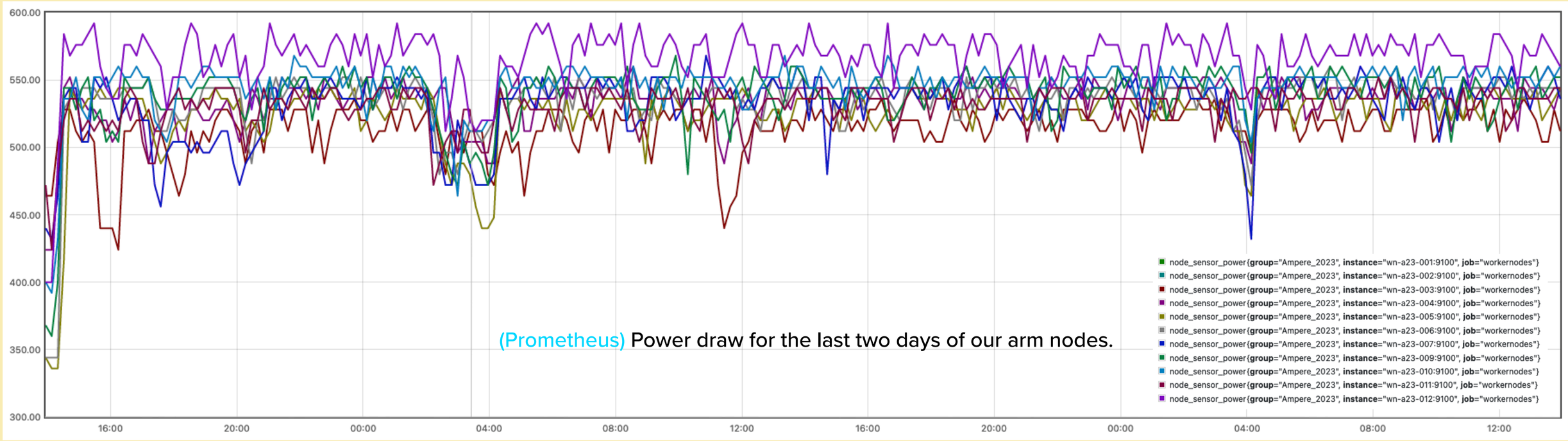
Showing 1 to 1 of 1 entries

Associated PandaQueue architectures

Type	Architectures	Vendor	Instructions	Model	Ops
cpu	aarch64 excl				

Site Monitoring

- Monitoring based around Prometheus. Managed to find version of it compiles for aarch64.
- Data relayed from ARM nodes into the same Grafana dashboard as the rest of the site with minimal changes.
- Some metrics are IPMI based - here some changes were required, but technical issues are similar to the ones whenever new hardware is installed.



ATLAS Testing - ARM Work

- The ARM nodes ran jobs starting with 50M events and with simulation, reconstruction and derivation tasks.
- Meant to run for about ~21 days, ended up taking 58 days.

34427583 task

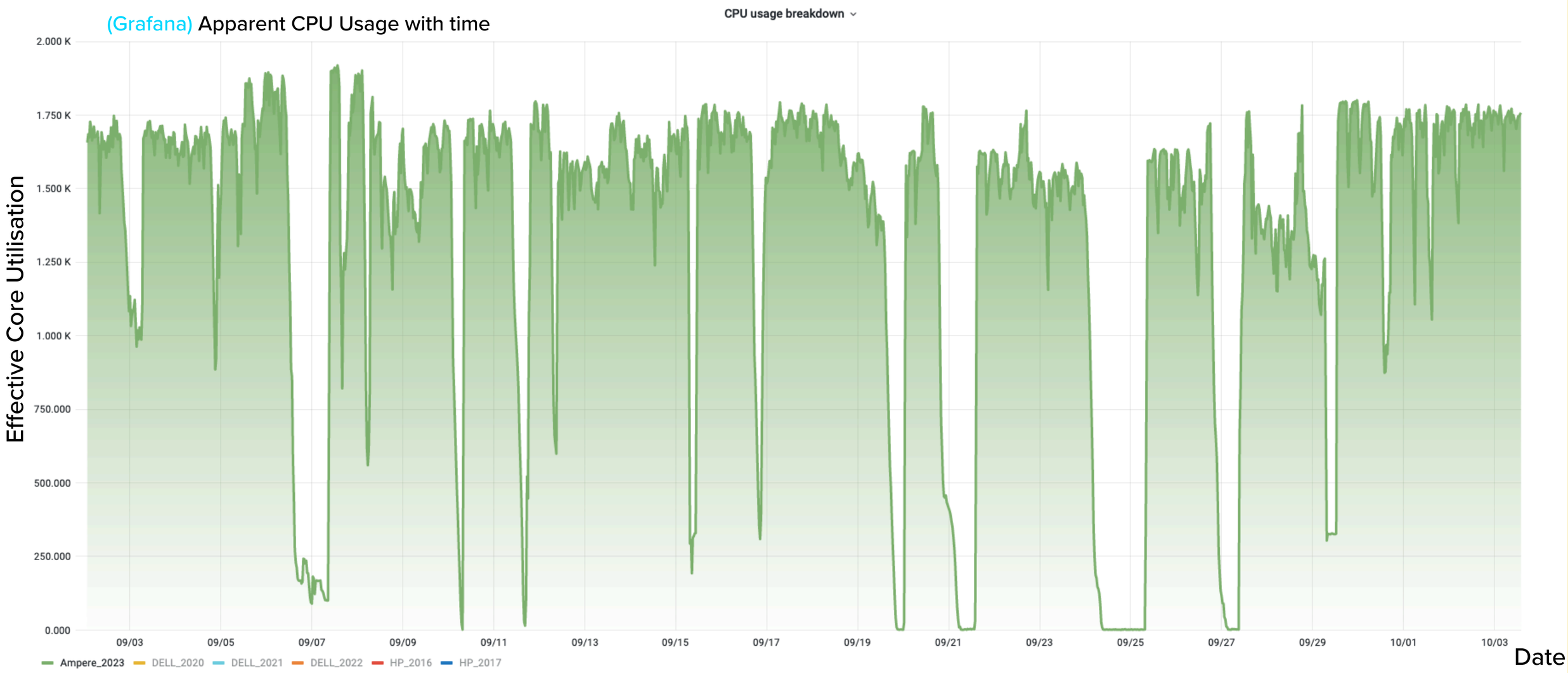
Estimated CO_2 emission in grams, based on the weighted average of regional electricity sources for ATLAS Grid computing

Task ID	Campaign	Request	Type	Processing type	Working Group	User	Nucleus	Status	N input files finished	N input events used	N output events	HS23s Expected Total done failed	Time stamps: created last modified	Cores	Priority: original current	Attempt	Parent	Tracker	CO_2 total done failed
34427583	MC23c	50572	prod	merge	AP_MCGN	janders	RAL-LCG2	finished	24,989 (100%) 24,989	49,978,000 (100%) 49,978,000	49,978,000	187.0M 179.3M 166.6M 12.6M	2023-08-21 07:10:11 2023-10-07 15:49:31	1	885 900	0	34412357	JIRA	15kg 14kg 1kg

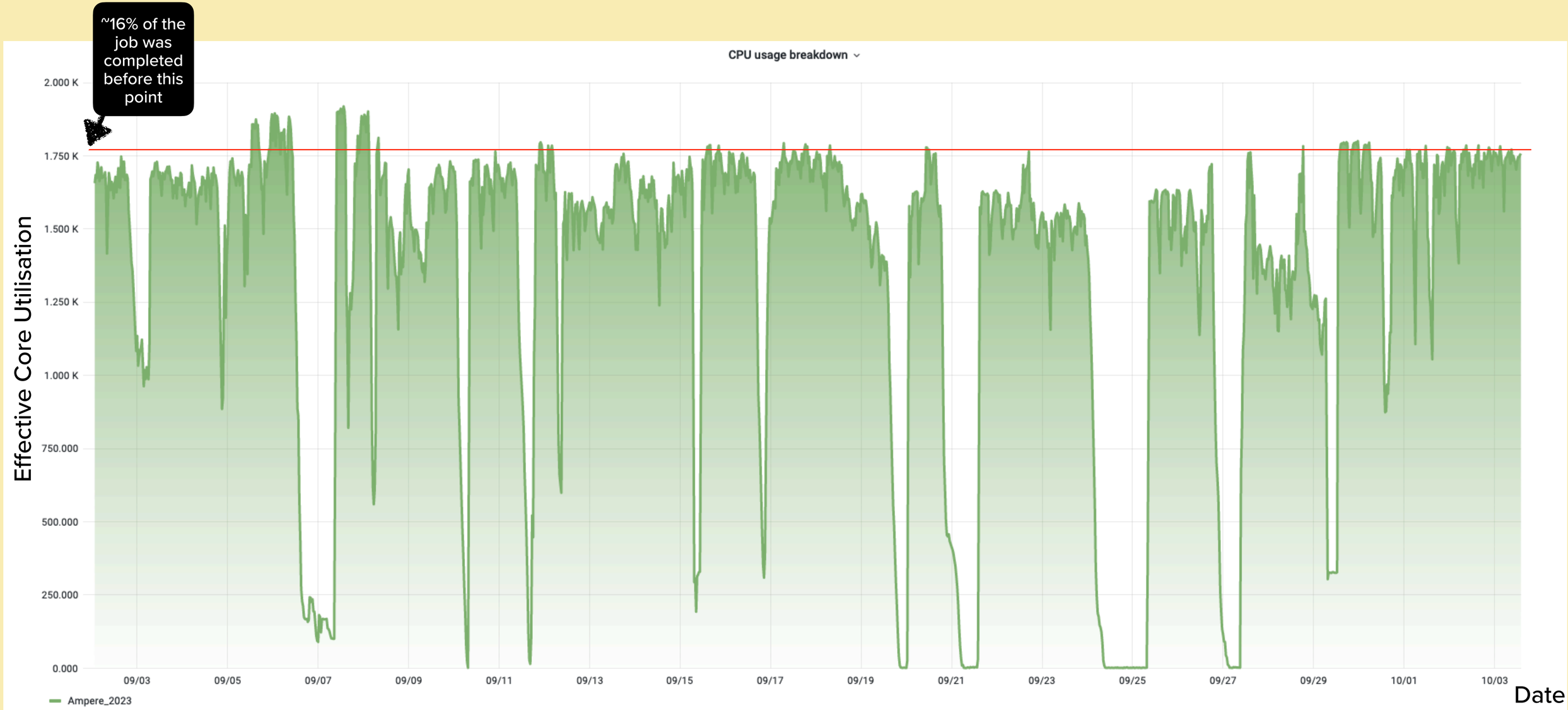
ATLAS Testing - ARM Work

- Running since Mid August, and still Finished in Early October (58 days)

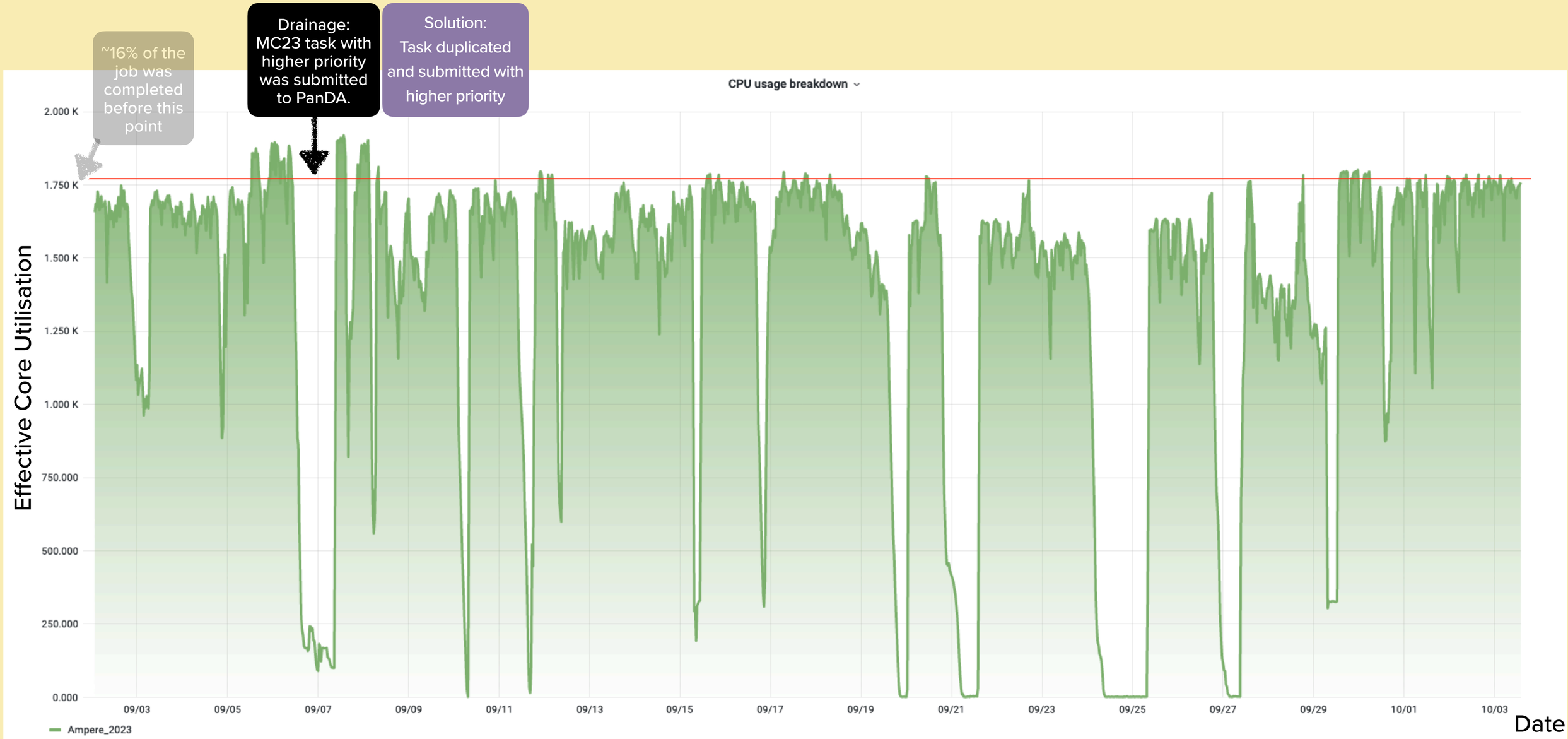
(Grafana) Apparent CPU Usage with time



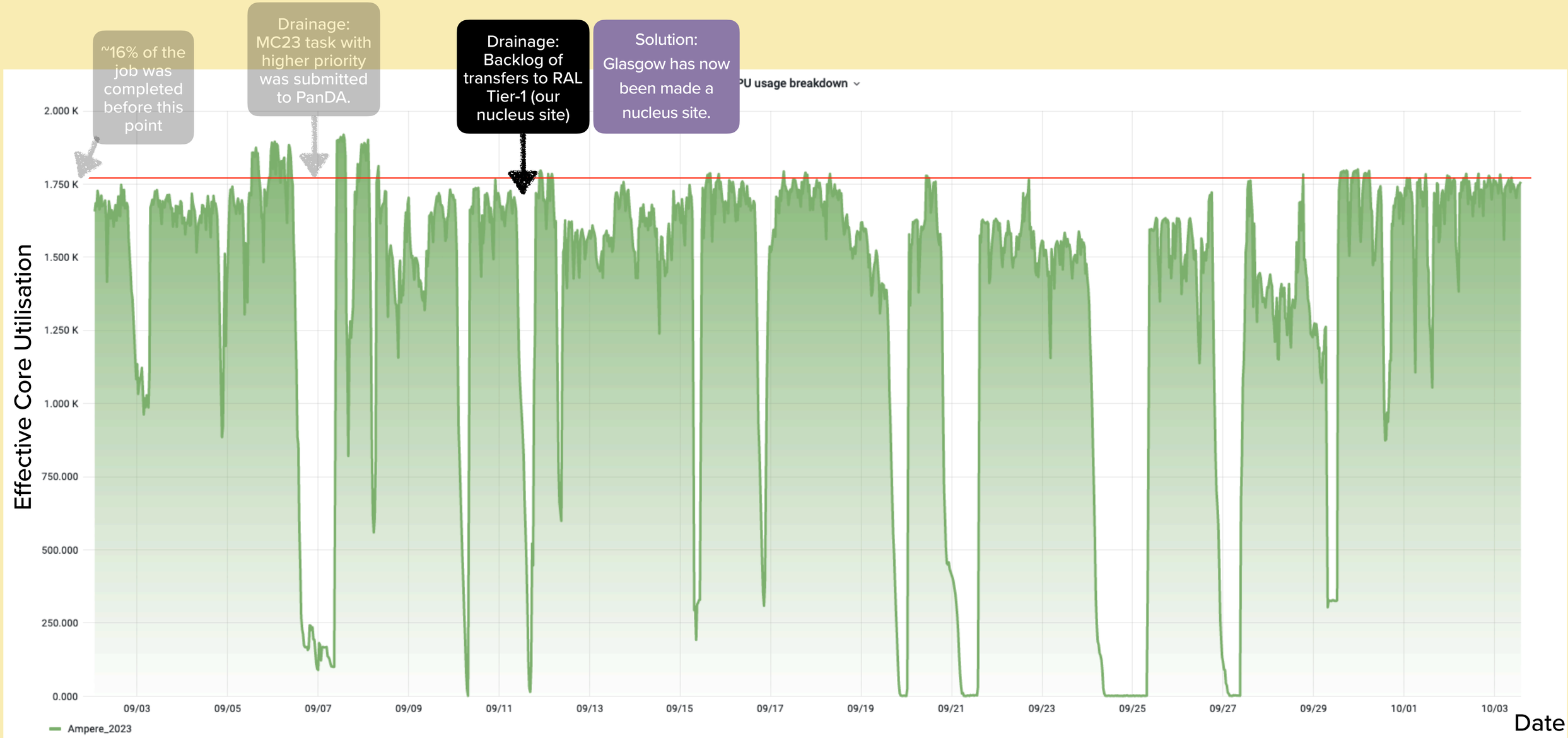
ATLAS ARM Tests



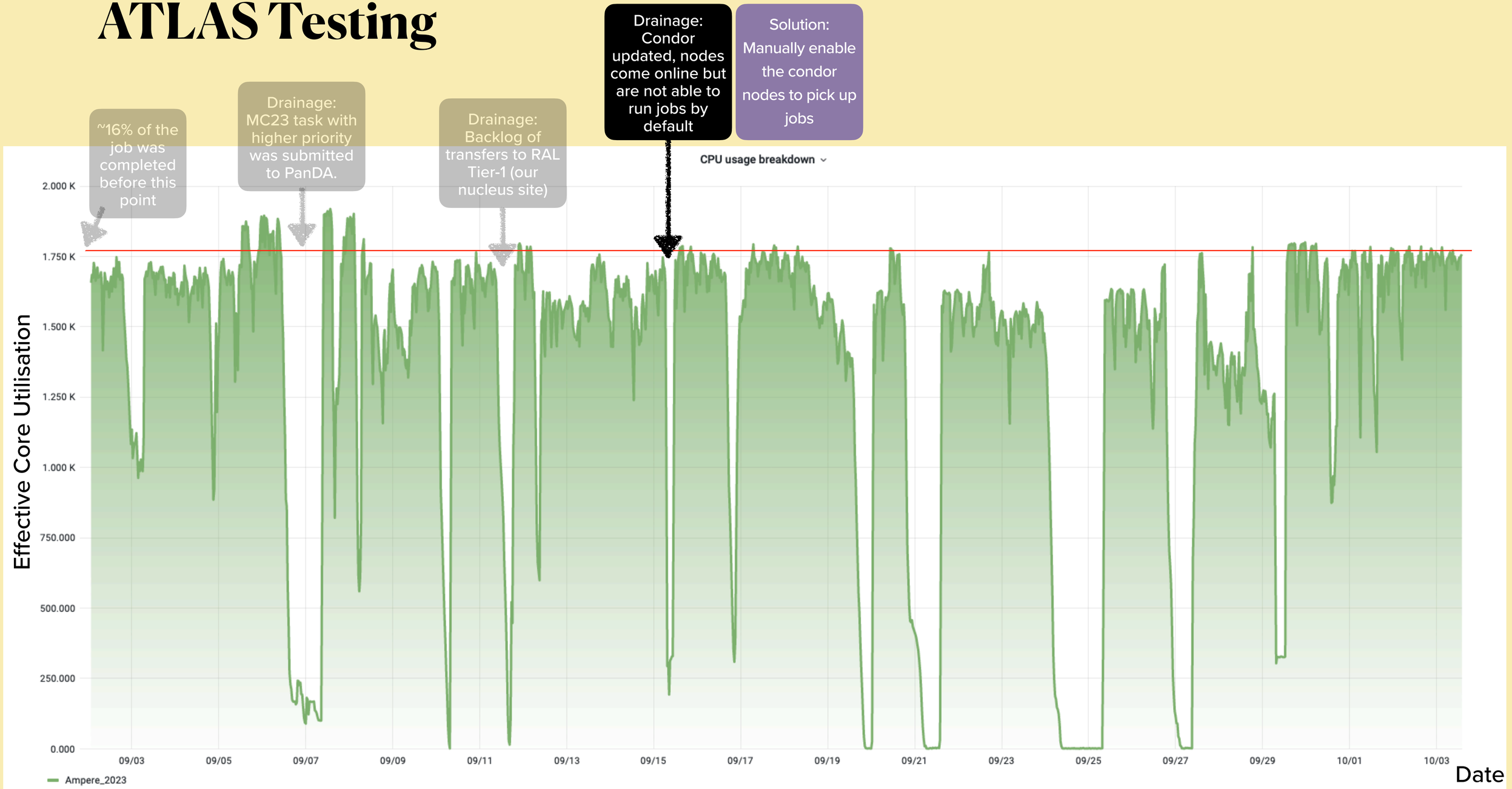
ATLAS ARM Tests



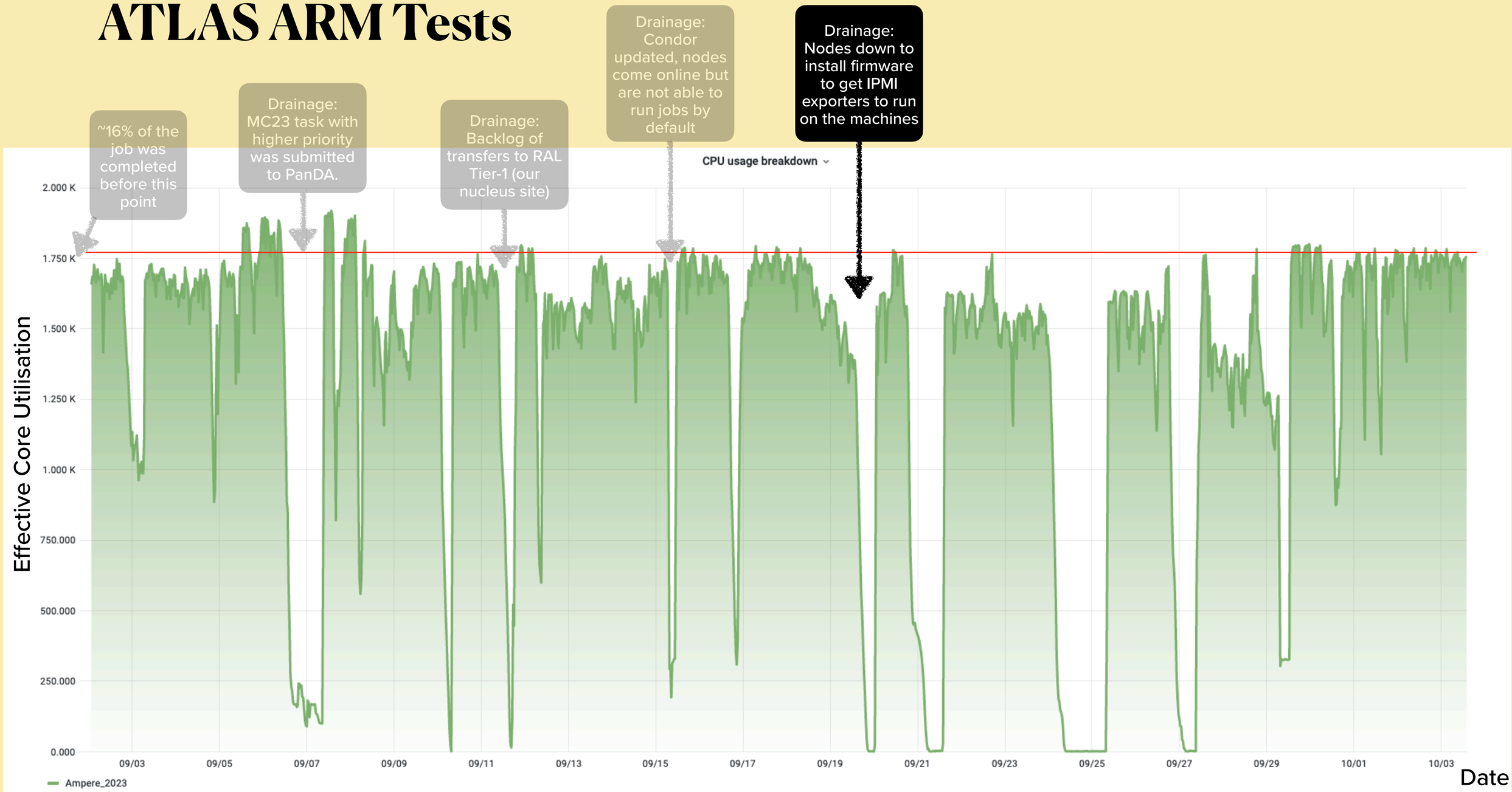
ATLAS ARM Tests



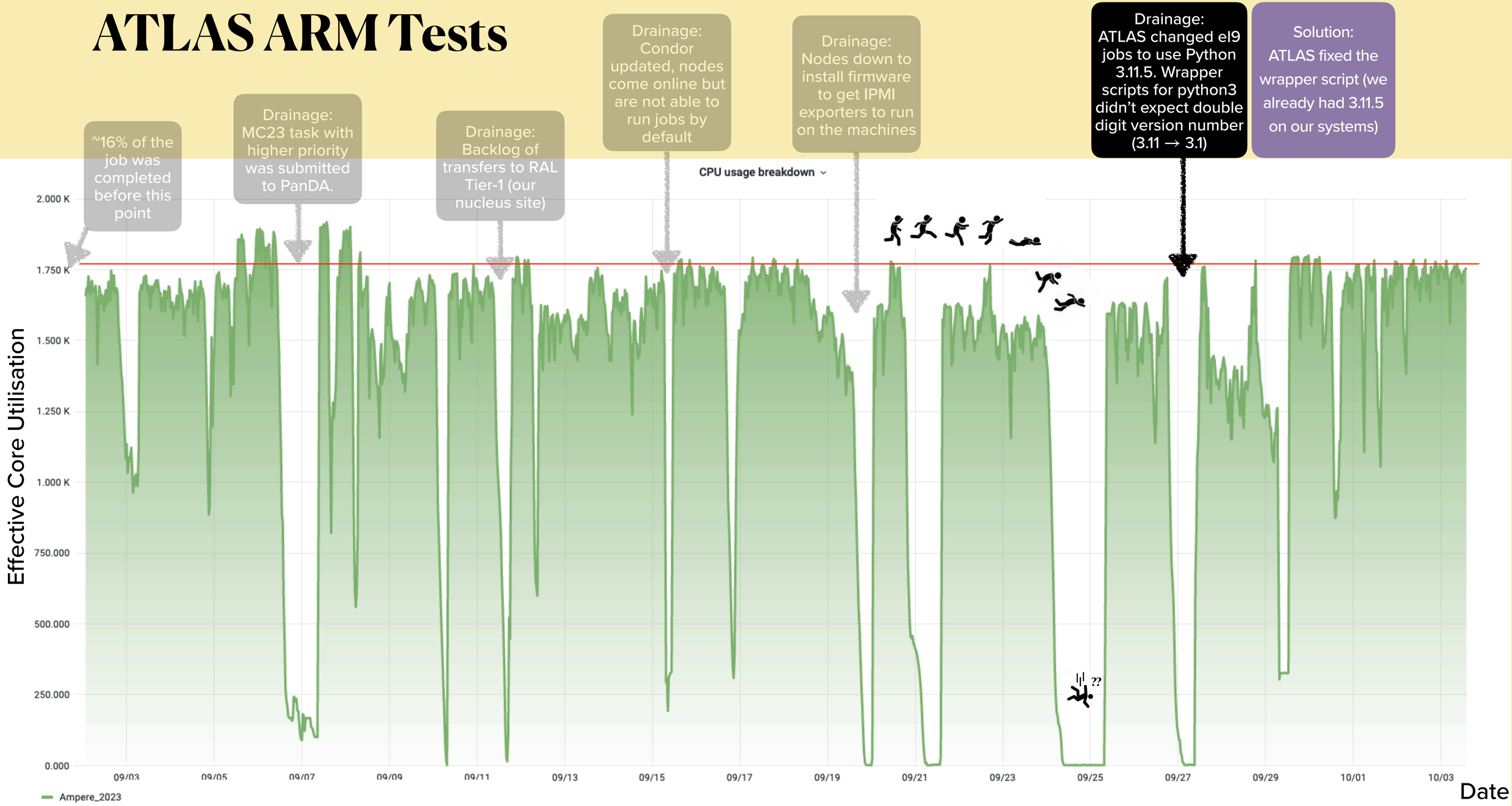
ATLAS Testing



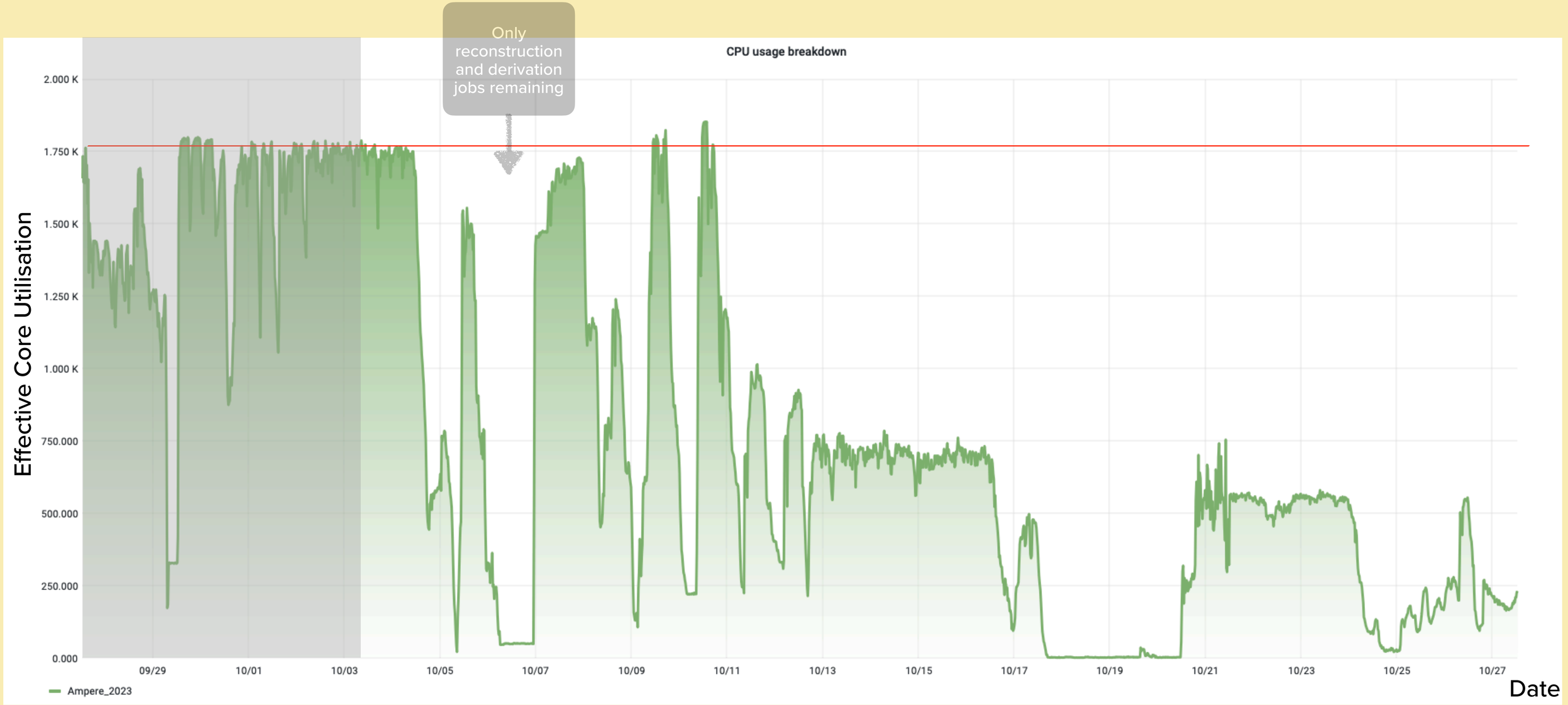
ATLAS ARM Tests



ATLAS ARM Tests



ATLAS ARM Tests

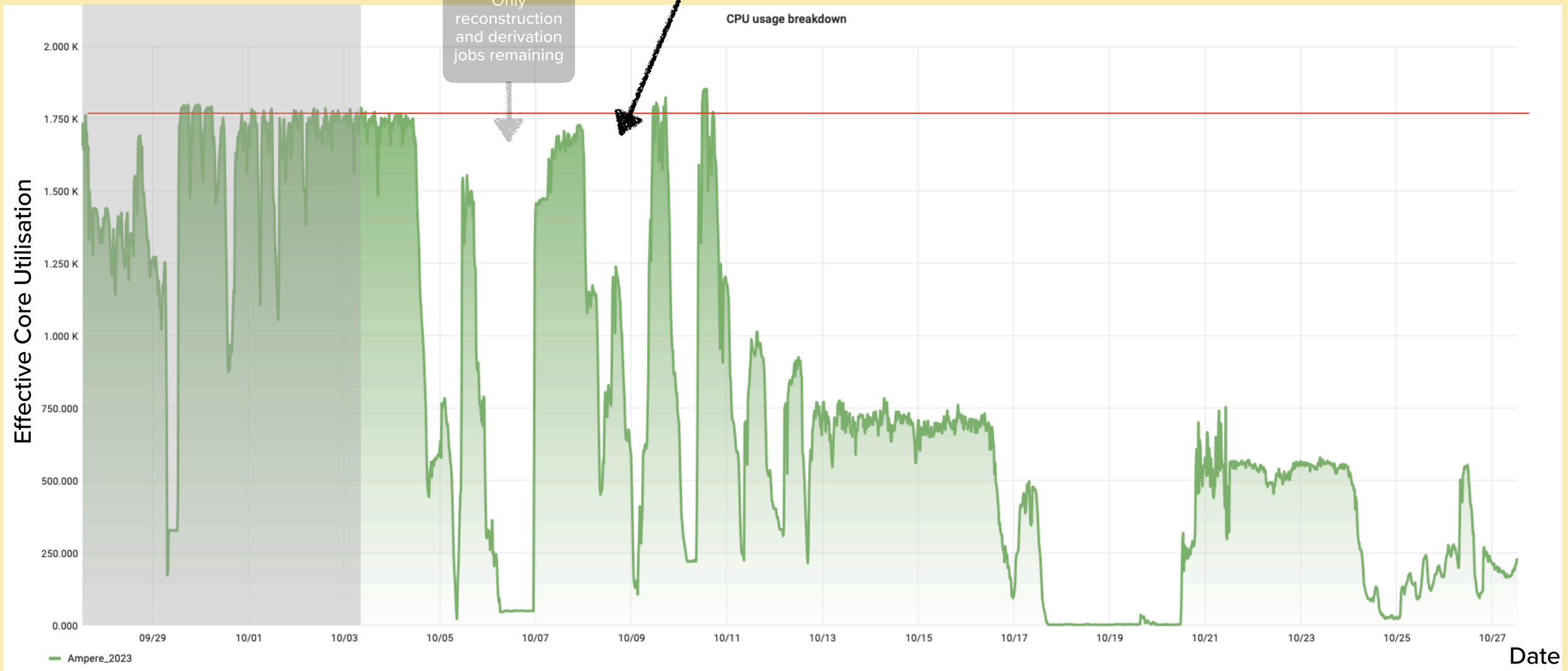


ATLAS ARM Tests

Inconsistent running : Due to the fact that each node has about 1.7TB of scratch space = ~10.5GB per core. For a multicore (8) job there should be 85GB of scratch space to play with, but reco ATLAS jobs apparently require 100GB.

Note:
When we can we might have to add more storage to our nodes

Only reconstruction and derivation jobs remaining

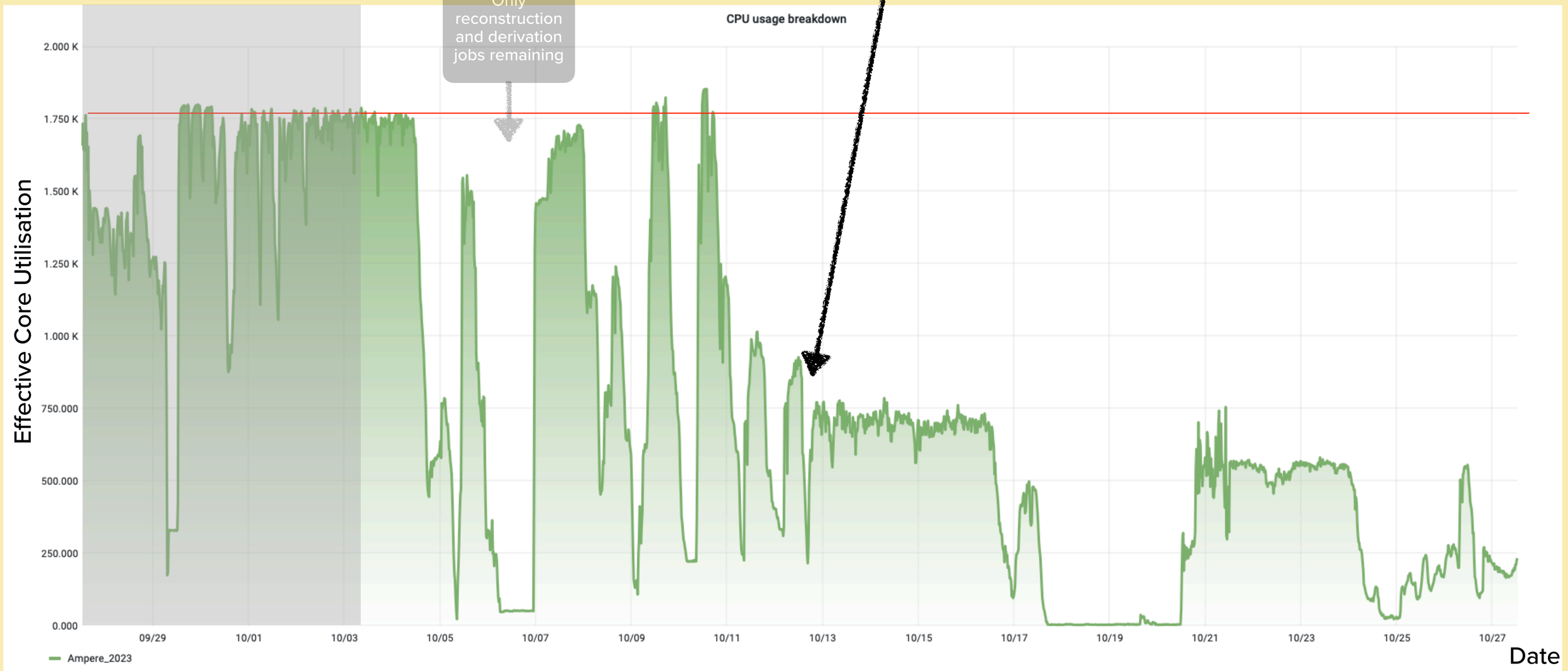


ATLAS ARM Tests

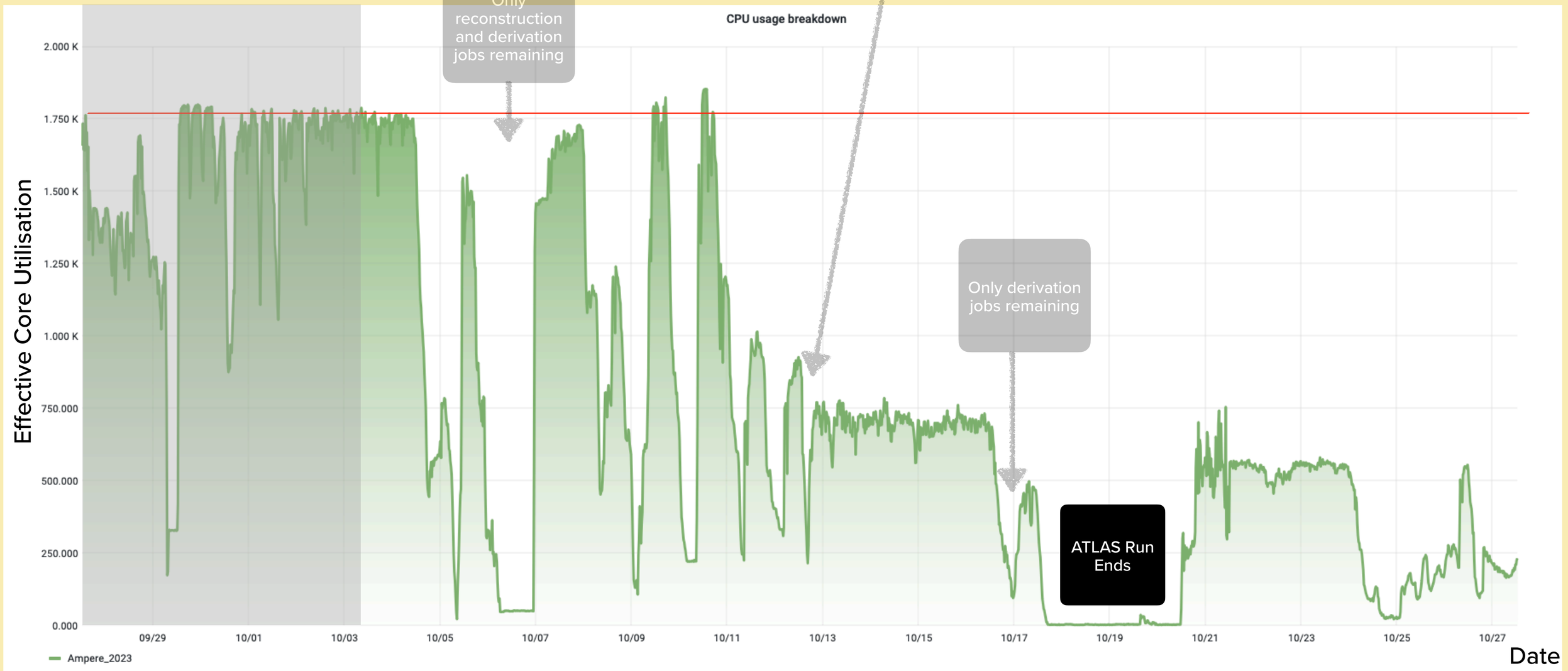
Inconsistent running : Due to the fact that each node has about 1.7TB of scratch space = ~10.5GB per core. For a multicore (8) job there should be 85GB of scratch space to play with, but reco ATLAS jobs apparently require 100GB.

Solution:
1) Setting added on ATLAS-side to delete intermediate files when running multi-step Reco transforms.
2) Nodes were half-filled by using a cap on the number of max workers in Harvester

Only reconstruction and derivation jobs remaining

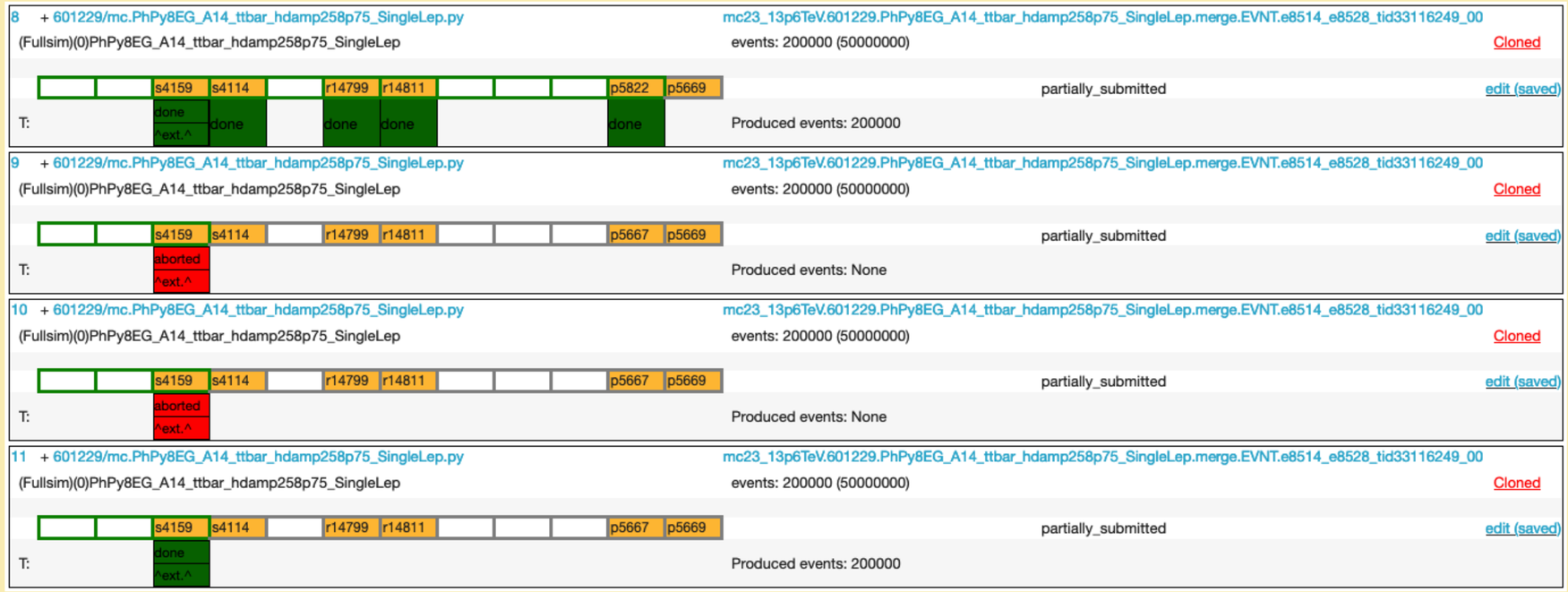


ATLAS ARM Tests



ATLAS Testing - x86 Work

- Using the same s-, r- and p-tags as the ARM work, jobs were also sent our x86 nodes to get some metrics for direct comparison.
- Only 200k events were sent in this way, site still live and taking our usual mix of work from other VO's
- Took ~26 days to complete the full chain.



34714849 task

Task ID	Campaign	Request	Type	Processing type	Working Group	User	Nucleus	Status	N input files finished	N input events used	N output events	HS06*sec Expected Total done failed	Time stamps: created last modified	Cores	Priority: original current	Attempt	Tracker	gCO_2 total done failed
34714849	MC23c	50572	prod	simul	AP_MCGN	janders	SWT2_CPB	done	100 (100%) 100	200,000 (100%) 200,000	200,000	223,400,000 175,231,218 174,063,834 1,167,384	2023-09-07 07:45:09 2023-09-08 11:05:42	8	220 900	0	JIRA	15,717 15,611 105

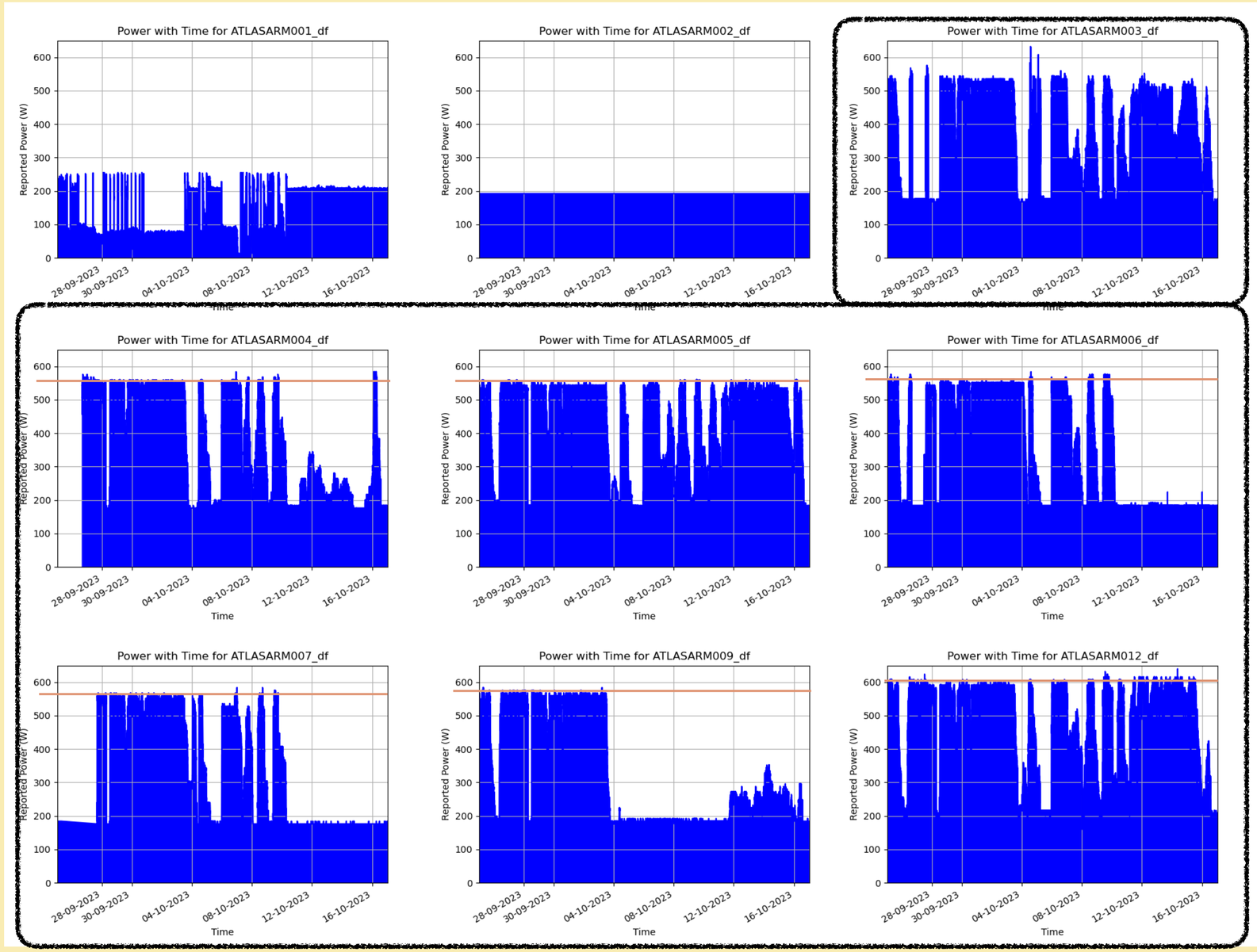
Site-side metrics

- The ARM nodes have been largely exclusively running ATLAS arm work.
- Didn’t have IPMI reporting power to Grafana when this was started.
- Had to use exported metrics from running machines every minute (timestamp, temperature, instantaneous power, frequency) to estimate the power we’ve dissipated as a site running this work.
- Many different ways of evaluating power could be used. Maximum power is reported regularly, but we feel it’s not the most representative figure.
  - To compare to figures we have for x86, I’ll estimate the peak plateau power

UNIXTIME(s)	CPU1TEMP(oC)	CPU2TEMP(oC)	POWER(W)	FREQUENCY(GHz)
1695827622	84.000	83.000	544.000	3.00
1695827682	83.000	85.000	544.000	3.00
1695827742	83.000	84.000	544.000	3.00
1695827803	82.000	84.000	528.000	3.00
1695827863	83.000	82.000	520.000	3.00
1695827923	81.000	82.000	504.000	3.00
1695827983	80.000	81.000	520.000	3.00
1695828043	81.000	80.000	512.000	3.00
1695828103	80.000	79.000	520.000	3.00
1695828163	81.000	81.000	528.000	3.00
1695828223	82.000	81.000	528.000	3.00
1695828283	79.000	81.000	512.000	3.00
1695828343	81.000	82.000	520.000	3.00
1695828403	83.000	83.000	552.000	3.00
1695828463	80.000	82.000	528.000	3.00
1695828523	80.000	83.000	536.000	3.00
1695828583	81.000	82.000	520.000	3.00
1695828643	81.000	81.000	512.000	3.00
1695828703	81.000	83.000	528.000	3.00
1695828763	82.000	81.000	536.000	3.00
1695828823	78.000	82.000	512.000	3.00
1695828883	80.000	81.000	504.000	3.00
1695828943	80.000	80.000	488.000	3.00
1695829003	80.000	79.000	480.000	3.00
1695829063	79.000	79.000	480.000	3.00

Reported power per node

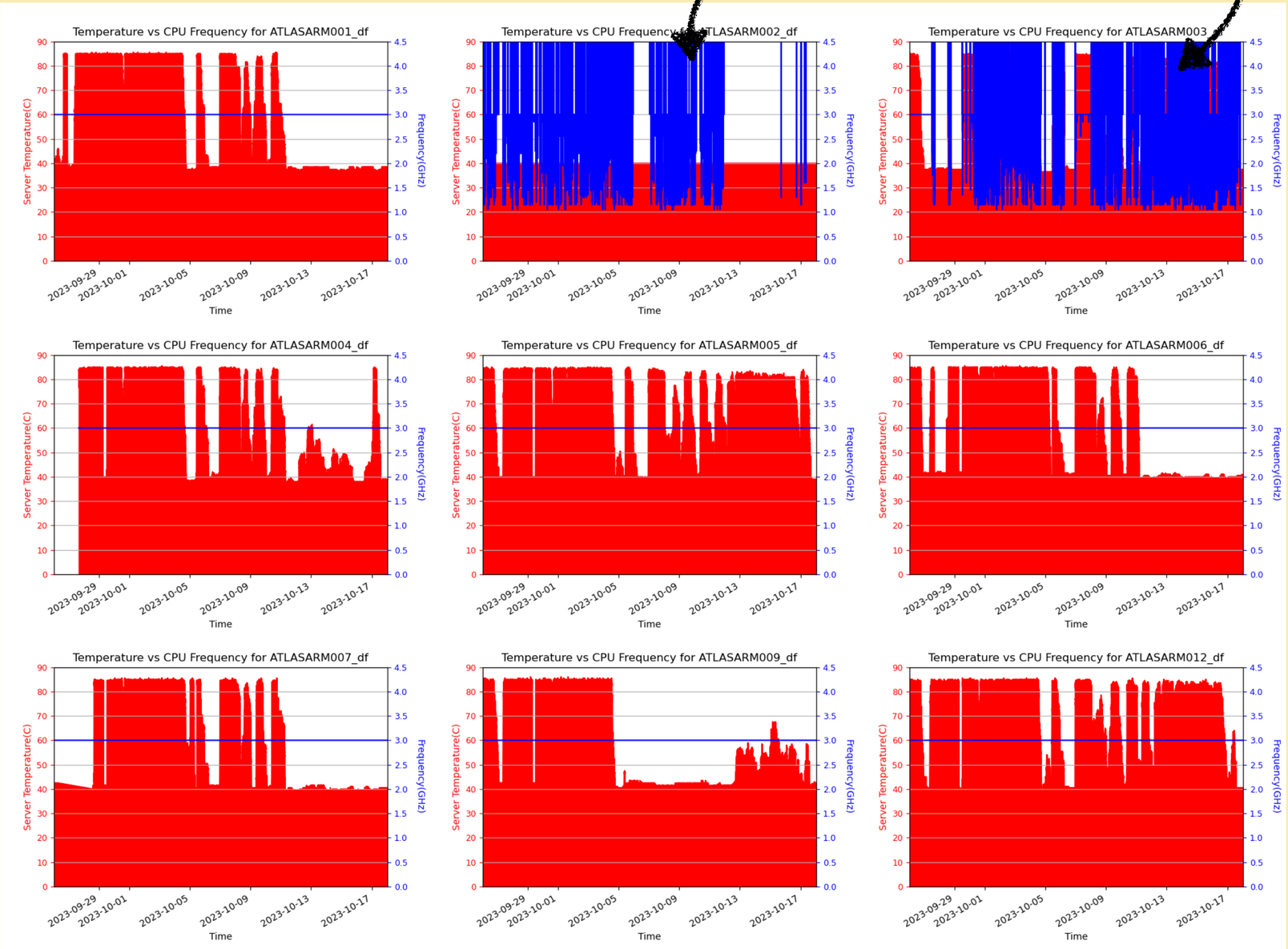
- IPMI Power reporting on some nodes was not working correctly, small hacks required.
- Not able to use this to measure power draw for ATLAS work, but can estimate.
- For nodes where the reporting works, power draw looks consistent with what we expect.
- Fixed for future runs with automatic reporting



Frequency vs Temperature

- During RAL's fixed-frequency ARM tests, the frequency was altered automatically to cope with the servers operating at high temperatures.
- This behaviour is not seen at our site. Temperature and Frequency seem uncorrelated.
- Samples were only taken once a minute, phenomena could potentially only occur at higher resolutions.

Reporting as 1000GHz so we know this is wrong



Comparison Methodology

- Use PanDA data taken on the ATLAS-side to estimate usage metrics.
- Estimate how many cores were used for how long to calculate the average CPU-hours taken for each type of ATLAS job (area of orange to the right)
- Then use a combination of reported and measured data to calculate HEPScore/Watt

States of jobs in this task [drop mode]

	pending	defined	waiting	assigned	throttled	activated	sent	starting	running	holding	transferring	merging	finished	failed	cancelled	closed
Run													100			

Job list. Sort by PandaID, time since last state change, ascending creation time, descending creation time, ascending mod time, descending mod time, priority, attemptnr, ascending duration, descending duration

PanDA ID Attempt#	Owner Group	Request Task ID	Transformation	Status	Created	Time to start d:h:m:s	Duration d:h:m:s	Mod	Cloud Site	Priority	N input events (N input files)	Max PSS/core, GB	Job info
5957143510 Attempt 2	janders AP_MCGN	50572 34714849	Sim_tf.py	finished	2023-09-08 02:36:02	0:0:04:33	0:4:46:37	2023-09-08 10:27:09	WORLD UKI-SCOTGRID-GLASGOW_CEPH online no active blacklisting rules defined	900	2000 (1)	0.34	

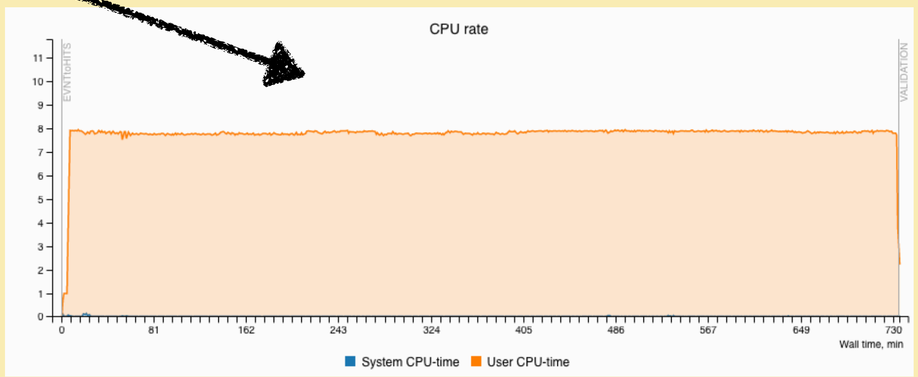
Job name: mc23_valid.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.simul.e8514_e8528_s4159.5956895702 #2

Datasets: **In:** mc23_13p6TeV:mc23_13p6TeV.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.merge.EVNT.e8514_e8528_tid33116249_00 || Rucio link
Out: mc23_valid.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.simul.log.e8514_e8528_s4159_tid34714849_00

Job name: mc23_valid.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.simul.e8514_e8528_s4159.5956895785

PanDA ID	Owner	WG	Request Task ID	Status	Type	Transformation	Created Last modified	Time to start Duration [d:h:m:s]	Site	Harvester instance Worker ID	Cores	Priority	Attempt
5956895785	janders	AP_MCGN	50572 34714849	finished	simul	Sim_tf.py	2023-09-07 20:40:37 2023-09-08 11:05:27	0:1:35:29 0:12:18:23	UKI-SCOTGRID-GLASGOW_CEPH	CERN_central_B 468057624	8	900	1
Datasets:	**In:** mc23_13p6TeV:mc23_13p6TeV.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.merge.EVNT.e8514_e8528_tid33116249_00 **Out:** mc23_valid.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.simul.log.e8514_e8528_s4159_tid34714849_00												
Files summary:	input: 1, size: 579.80 (MB); log: 1; output: 1; pseudo_input: 1												

Logs | Go to | Show | Jump to | Memory and IO plots



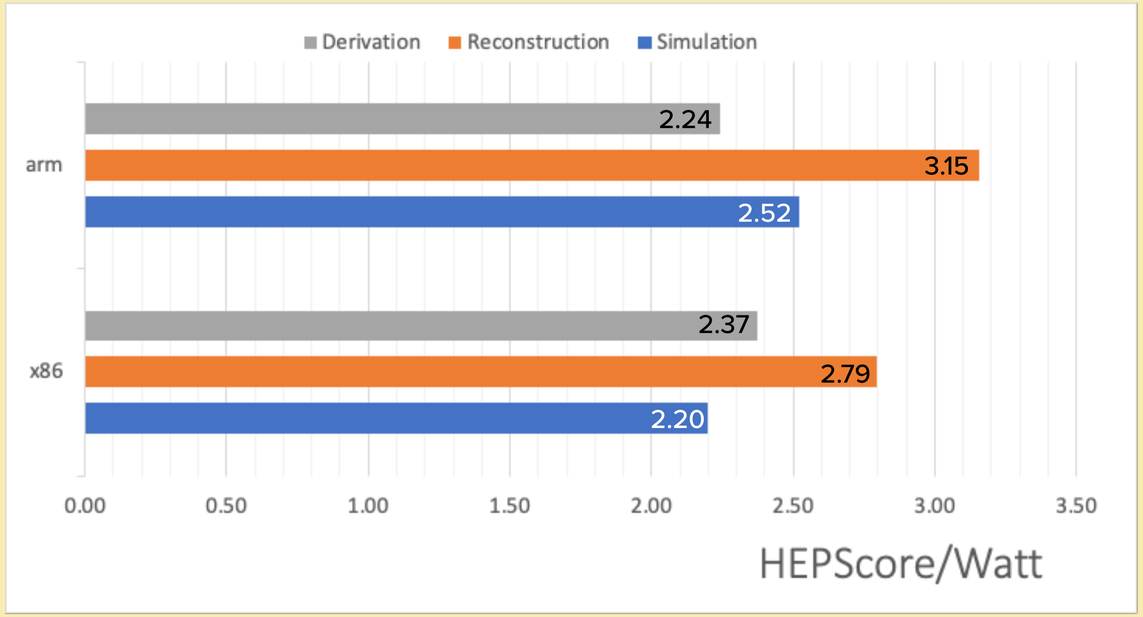
Estimates of HEPScore/Watt

$$\frac{\text{HEPScore}}{\text{Watt}} = \frac{\text{HEPScore}}{\text{Job}} \div \left(\frac{\text{Cores/Hour}}{\text{Job}} \times \frac{\text{Watt*Hours}}{\text{Core}} \right)$$

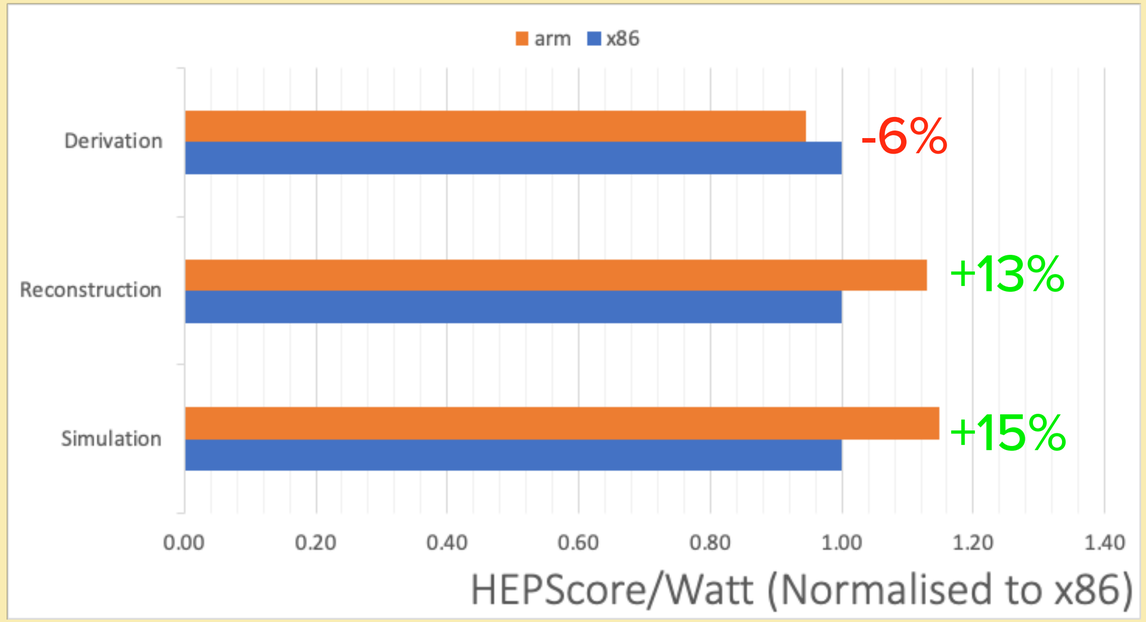
Obtained from ATLAS PanDA

Obtained using methodology from slide 25

Measured on site

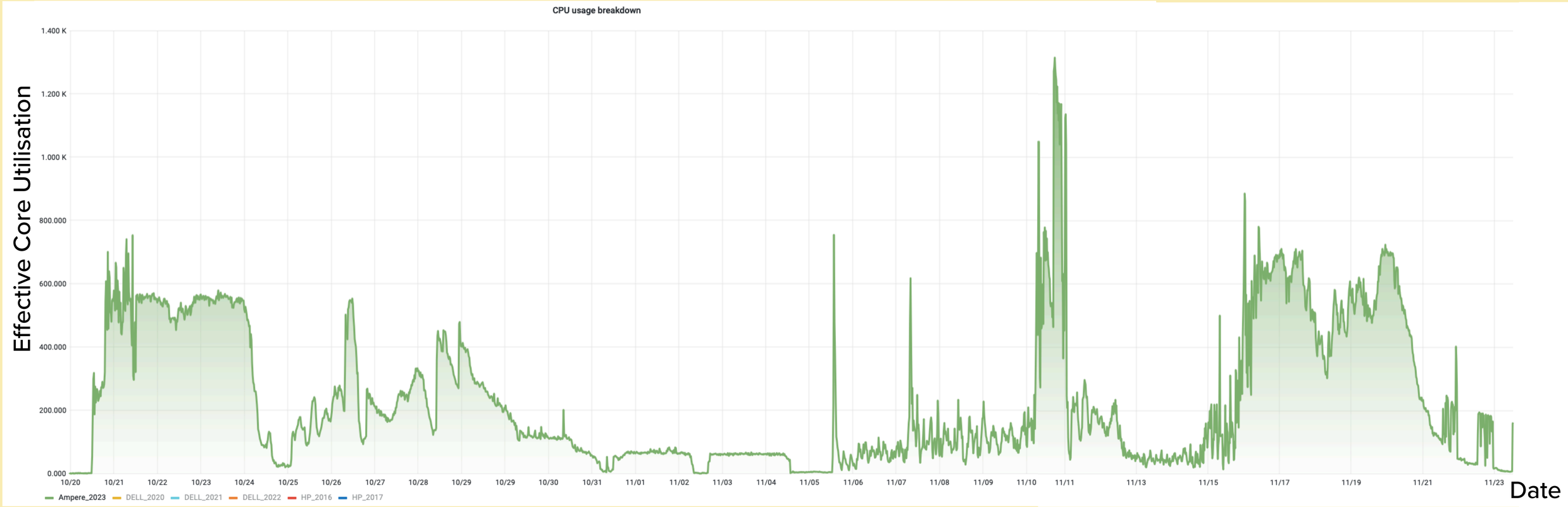


- The x86 bars here are weighted averages from the different x86 machines we have at our cluster.
- HEPScore/Watt better for ARM for reconstruction and simulation work
- ARM load order of magnitude larger, expect merge steps (I/O bound processes) of jobs to more CPU intensive relative to the jobs for x86.



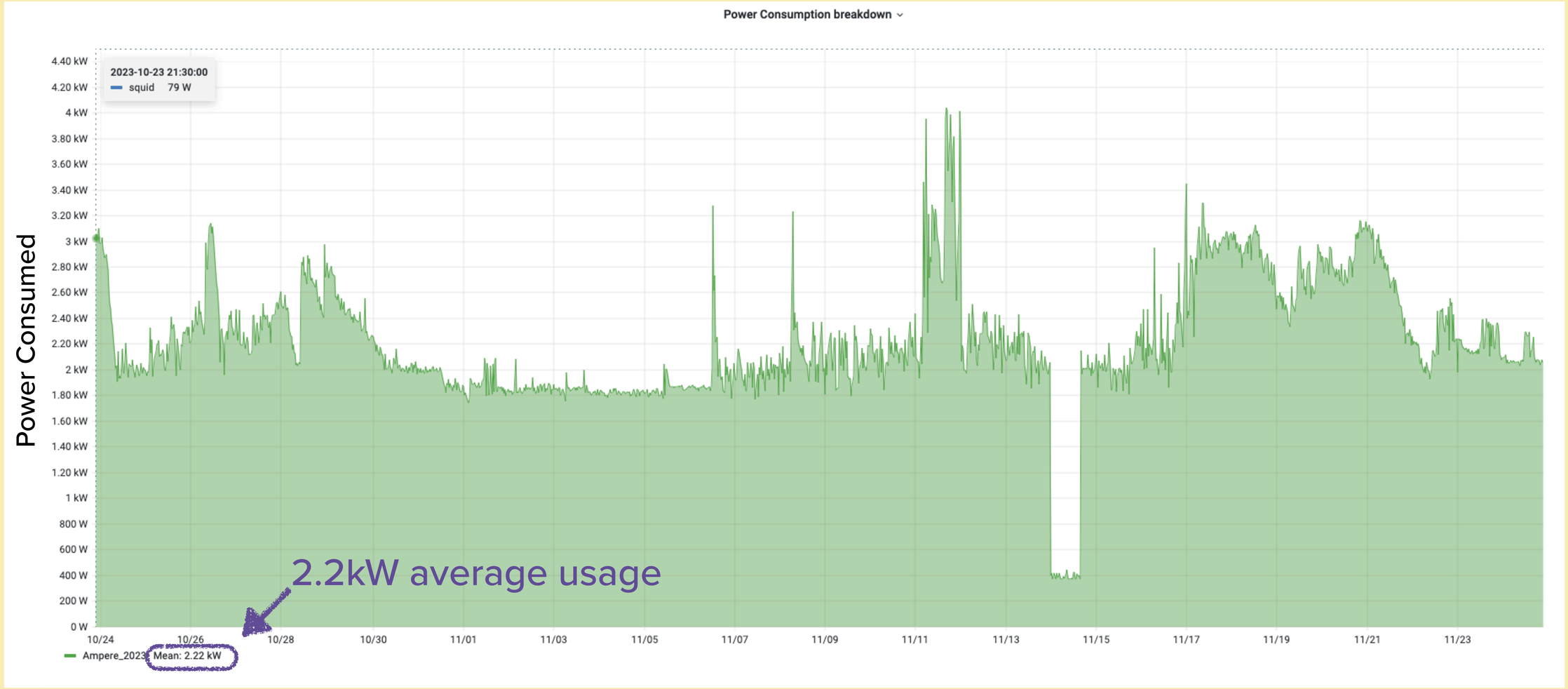
CMS Testing

- We've been running CMS work since ~20th October. It finished on Friday (~38 days).
- Inefficient core utilisation. Hypothesise two reasons for this:
  - with 3.3Gb/core some workloads could be running out of memory (slight evidence of swapping)
  - not being a CMS tier-2 site means a reliance on AAA to obtain data (bandwidth problems).
- Jobs often waiting for 4 hours, so reliable metrics weren't gathered.



CMS Testing

- Power Reporting in Grafana working at this point - Expect average usage ~6.2kW for efficient usage
- CMS currently running validation tasks on output datasets.



Secondary Testing

- We've done some provisional benchmarking with the ARM Farm and compared it to other arm and x86 machines.
- Found that the dual socket machine wasn't twice as performant as the single socket (orange to red).
- Cache coherency overheads on the ARM dual-CPU were at times very high compared to other Intel/AMD CPUs.
- The ARM CPU's have two different protocols for cache coherency: in-chip and between-chip. Transferring between these makes it less efficient. (for further reading see this article)
- Bergamo outperforms all other systems in terms of HEPScore but…

(hyperlink)

David Britton's Talk: Carbon and Sustainability in ALTAS S&C Week - 2nd October

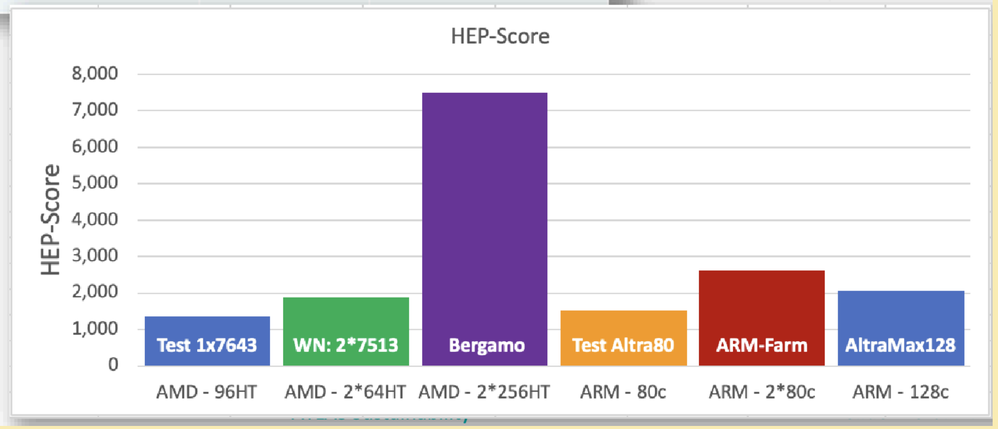
GridPP
UK Computing for Particle Physics

HEPScore Measurements

Machine	CPU	Threads	HEPScore
Test AMD	1x EPYC 7643 HT	96	1359
Std AMD WN	2x EPYC 7513 HT	128	1887
AMD Bergamo	2x EPYC 9754 HT	512	7497
Test ARM	1x ALTRA 80	80	1513
Farm ARM	2x ALTRA 80	160	2619 ??
Ampere Max	1x ALTRA Max	128	2065

x4

Not x2 ??



David Britton, University of Glasgow

Future Glasgow Farm Plans

What will we fill our racks with?

- … as it consumes a lot of power, it's less efficient at providing work per unit of power.

- Bergamo vs Altra Max!
  4 AltraMax CPUs and 2 Bergamo CPUs have comparable numbers of cores (512) and should draw roughly the same amount of power. ARM would still be more energy efficient.

- In this configuration Bergamo would take up less rack space. So you can get more threads per rack with the Bergamo (in theory).

(hyperlink)

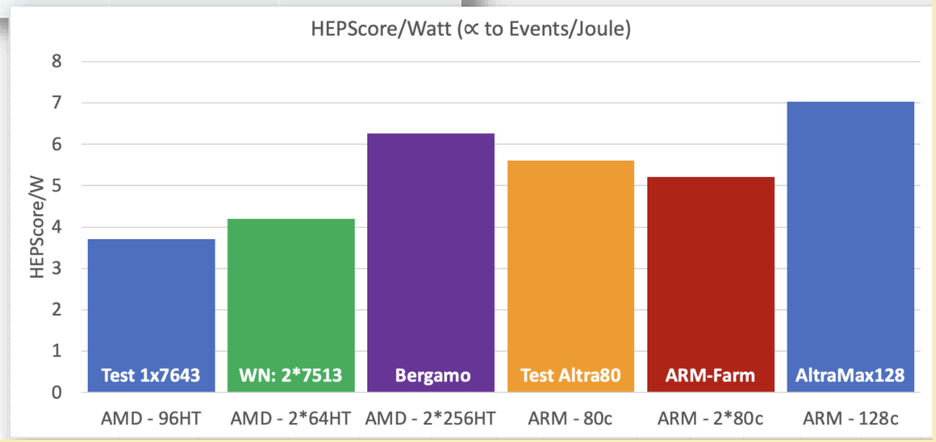
David Britton's Talk: Carbon and Sustainability in ALTAS S&C Week - 2nd October

GridPP
UK Computing for Particle Physics

HEPScore/Watt

Pow max (W)	Avg. Pow (W)	Machine	CPU	Threads	HS/Watt	
451	367	Test AMD	1x EPYC 7643 HT	96	3.7	
512	449	Std AMD WN	2x EPYC 7513 HT	128	4.2	
1347	1198	AMD Bergamo	2x EPYC 9754 HT	512	6.3	50% improvement from 7513 WN
348	270	Test ARM	1x ALTRA 80	80	5.6	
381	294	Farm ARM	2x ALTRA 80	160	5.2 ??	
664	502	Ampere Max	1x ALTRA Max	128	7.0	25% improvement from 80c

Note added: We use Average Watts to calculate HS/Watt. Other people also use Max Watts, so care needs to be taken comparing numbers.



Future Glasgow Farm Plans

What will we fill our racks with?

1) https://www.datacenterdynamics.com/en/opinions/dude-wheres-my-high-density-rack/

2) https://www.upsite.com/blog/whats-driving-higher-rack-densities-in-the-data-center

- Data-centres are usually limited not by space, but by power[1,2]. To fill a rack with Bergamo would tax our power budget before we filled the space. ARM servers are better in this regard.
- High core density could mean that cooling may become an issue. Our racks are designed to be low density (<10kW per rack), and a Bergamo machine may leave more gaps.
- What you choose depends on your data-centre design and what metrics you value. For energy efficiency ARM seems to be better.

(hyperlink)

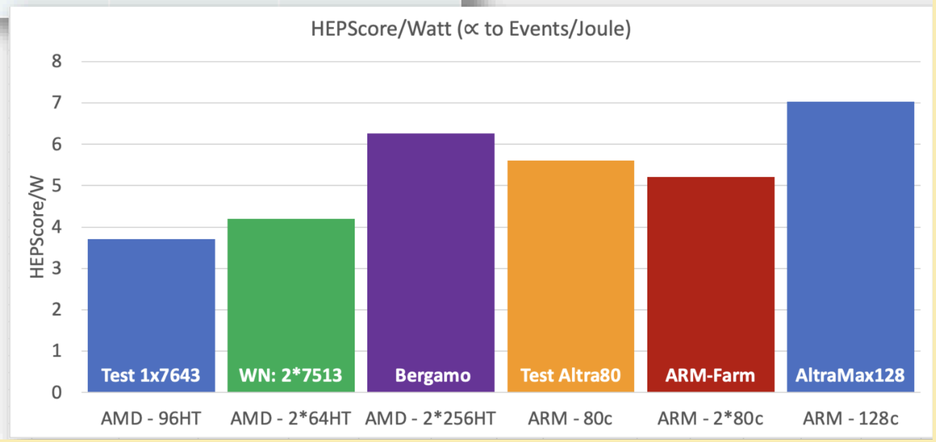
David Britton's Talk: Carbon and Sustainability in ALTAS S&C Week - 2nd October

GridPP
UK Computing for Particle Physics

HEPScore/Watt

Pow max (W)	Avg. Pow (W)	Machine	CPU	Threads	HS/Watt	
451	367	Test AMD	1x EPYC 7643 HT	96	3.7	
512	449	Std AMD WN	2x EPYC 7513 HT	128	4.2	
1347	1198	AMD Bergamo	2x EPYC 9754 HT	512	6.3	50% improvement from 7513 WN
348	270	Test ARM	1x ALTRA 80	80	5.6	
381	294	Farm ARM	2x ALTRA 80	160	5.2 ??	
664	502	Ampere Max	1x ALTRA Max	128	7.0	25% improvement from 80c

Note added: We use Average Watts to calculate HS/Watt. Other people also use Max Watts, so care needs to be taken comparing numbers.



Takeaways and Conclusions

- Our experience running these machines is still not as mature as x86. Still hitting minor silly bugs on occasion and errors in provisioning, but even with these we seem to be running well.
- Moving forward experiments may to be smart about how they advertise these resources and submit jobs to them since we expect most sites to become heterogeneous in the coming years.
- Preliminary comparisons show that the HEPScore/Watt performance increases from ARM compared to x86 are still present in a “real” environment, but smaller in magnitude.
- As a result of our tests; we decided to purchase more ARM machines and we have an ongoing procurement for them (want > 1000 cores AltraMax M128-30).
- We’ve leaned a lot running both ATLAS and CMS work. Hopefully we’ll be running ALICE work soon!

Backup

Additional Tips/Comments
Technical setup differences between ARM and x86

- As we'd never really considered the situation where we'd have architecture other than x86 in the site, PXE and Ansible required a one-off process where we went through and added in restrictions based on architecture (in the same way our Ansible configuration has already branched based on OS). This wasn't a lot of effort, though.

- We've encountered several bugs in the ARM version of packages (e.g. Grub2) which aren't in the x86 versions, but nothing we couldn't work around. Again, it's sometimes not clear whether it's actually an ARM issue or whether it's really an EL9 issue.

Best way of advertising resources in the future?

At some point we will want to get rid of the test CE

Option 1

- Modify exiting Queue

UKI-SCOTGRID-GLASGOW_CEPH

ce01.gla.scotgrid.ac.uk (x86)
ce02.gla.scotgrid.ac.uk (x86)
ce03.gla.scotgrid.ac.uk (x86)
ce04.gla.scotgrid.ac.uk (x86)
ce05.gla.scotgrid.ac.uk (aarch64)

- Dangerous, will impact the workflow of many VO's. Will have all the ARM traffic on one CE - not scalable in the future?
- All the CE's are on x86, they just front different architectures

Option 2

- Add a queue pointing to the same CE's

UKI-SCOTGRID-GLASGOW_CEPH

ce01.gla.scotgrid.ac.uk (x86)
ce02.gla.scotgrid.ac.uk (x86)
ce03.gla.scotgrid.ac.uk (x86)
ce04.gla.scotgrid.ac.uk (x86)

UKI-SCOTGRID-GLASGOW_ARM

ce01.gla.scotgrid.ac.uk (aarch64)
ce02.gla.scotgrid.ac.uk (aarch64)
ce03.gla.scotgrid.ac.uk (aarch64)
ce04.gla.scotgrid.ac.uk (aarch64)

- If Job requirements can be successfully injected this seems the safest option

Option 3

- Read architecture from the jobs themselves.

UKI-SCOTGRID-GLASGOW_CEPH

ce01.gla.scotgrid.ac.uk (x86, aarch64)
ce02.gla.scotgrid.ac.uk (x86, aarch64)
ce03.gla.scotgrid.ac.uk (x86, aarch64)
ce04.gla.scotgrid.ac.uk (x86, aarch64)

- Condor_submit has architecture flags. Could try to pulling the architecture flag from condor_submit into the ARCsub, maybe modify in job description language (JDL)?
- Would potentially need to set up/inject default architecture so that "standard" x86 jobs don't get sent to ARM cores.

Option 4

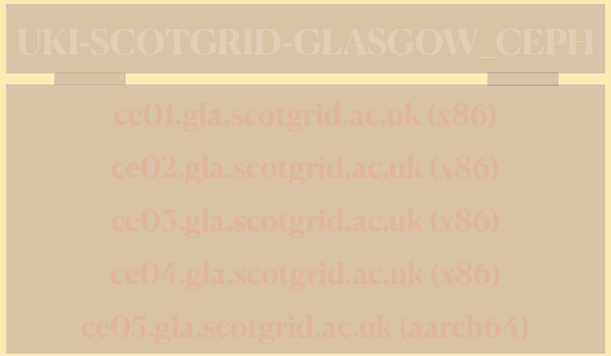
- Pilots report to VO what architecture it's running on.
- That VO sends jobs of that type. Would require every VO to add this functionality to their pilots, not really default-able.
- Potentially wasteful if a site gets a pilot running on an ARM server and has no ARM work, long term solution?

Best way of advertising resources in the future?

At some point we will want to get rid of the test CE

Option 1

- Modify exiting Queue



- Dangerous, will impact the workflow of many VO's. Will have all the ARM traffic on one CE - not scalable in the future?

- **This option preferred by ATLAS, certainly the simplest to implement**

Option 2

- Add a queue pointing to the same CE's

- If Job requirements can be successfully injected this seems the safest option

Option 3

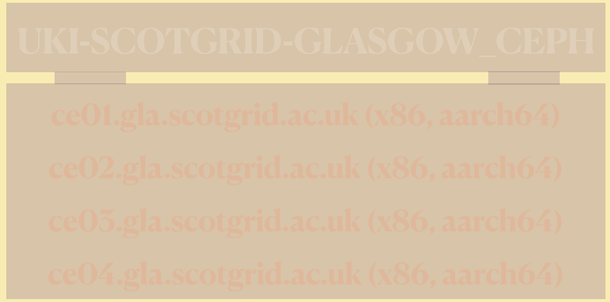
- Read architecture from the jobs themselves.

- Condor_submit has architecture flags. Could try to pulling the architecture flag from condor_submit into the ARCsub, maybe modify in job description language (JDL)?

- Would potentially need to set up/inject default architecture so that "standard" x86 jobs don't get sent to ARM cores.

Option 4

- Pilots report to VO what architecture it's running on.



- That VO sends jobs of that type. Would require every VO to add this functionality to their pilots, not really default-able.

- Potentially wasteful if a site gets a pilot running on an ARM server and has no ARM work, long term solution?