Do you really need the answer now? Polling carbon intensity data when submitting High-Performance Computing batch jobs

Gary Polhill[1]

The James Hutton Institute, Craigiebuckler, Aberdeen, Scotland, UK
gary.polhill@hutton.ac.uk

Abstract

High-Performance Computing (HPC) clusters have high electricity demands, and in the academic sector particularly, are funded by public money. With many international, national and regional government organizations having policy to reduce greenhouse gas (GHG) emissions, beneficiaries of public funds are increasingly being asked to report on and take active steps to cut their GHG emissions. In this short article, I show how HPC users in the UK can delay job submission until the carbon intensity of electricity production is below a preferred level.

1 Introduction

The principal source of Greenhouse Gas (GHG) emissions arising from High-Performance Computing (HPC) cluster use during day-to-day utilization is the use of fossil fuels for electricity generation. This is measured by the 'carbon intensity' (g CO_2 e/kWh). More broadly, Turner's [1] training course discusses HPC carbon intensity mitigation strategies, including those associated with commissioning and administering the HPC systems.

Strategies for reducing carbon intensity offering empowerment to HPC users are those associated with *demand* shifting. Turner [1] details two approaches, one is to move the job to a system where the power is generated by lower carbon intensity methods (spatial shifting), the other is to delay the job to times when the carbon intensity of electricity generation is lower (temporal shifting). Spatial shifting is an option for administrators of multi-site HPC clusters. For users, temporal shifting is likely to be the primary means by which users of HPCs can have some control over the carbon emissions associated with running their jobs.

Companies such as Electricity Maps[1] provide an Application Programming Interface (API) to live data on carbon intensity of electricity generation. In the UK, information on carbon intensity of electricity production can be obtained without charge from the National Energy System Operator's Carbon Intensity website [2]. Critically, this service not only provides a regional breakdown of carbon intensity, but also uses machine learning to estimate carbon intensity over the coming 48 hour period – the typical timeframe over which one might expect an HPC batch job to run.

Users can have a degree of control over GHG emissions from their HPC batch jobs using a shell script that polls the Carbon Intensity website until lower expected GHG emissions are anticipated. The rest of the document shows an example of such a script, and briefly discusses potential issues and consequences of using it.

[1] https://www.electricitymaps.com/
[2] https://carbonintensity.org.uk/

2 Delaying job submission

The `delay.sh` script below uses `curl` and `jq` to parse the output of the Carbon Intensity website, waiting for the forecast carbon intensity of electricity production at the postcode `$pc` to be no more than `$max_ci`. Assuming the job submission script is `job.sh`, the user would submit the job with `nohup nice delay.sh job.sh &` before logging out.

Listing 1: `delay.sh`

```
# Parameters
pc=DD2           # Only the first part of the postcode is needed
max_ci=10        # Maximum acceptable carbon intensity
poll=60          # Polling interval

# URL and JSON query
url=https://api.carbonintensity.org.uk/regional/postcode/$pc
qu='.data[0].data[0].intensity.forecast'

# Implement delay
ci=`expr $max_ci + 1`
while [ $ci -gt $max_ci ]
do
  sleep $poll
  ci=`curl -X GET $url -s -H 'Accept: application/json' | jq $qu`
done

# Submit the job (SLURM)
sbatch "$@"
```

3 Discussion

Improvements to the above script could increment `$max_ci` the longer the delay persisted to allow for the possibility that the carbon intensity of electricity production does not drop to the prescribed level in a reasonable time.

However, real GHG savings would require cluster configuration and sufficient users to adopt delaying submission so that nodes would be suspended. Once the carbon intensity dropped, there might be more than the normal number of simultaneously submitted jobs. Further work could use simulation to explore these collective and institutional issues in more depth.

Acknowledgments

This work was supported by the Scottish Government Rural and Environment Science and Analytical Services Division (project reference JHI-C5-1)

References

[1] Andrew Turner. Green software use on HPC (course materials). https://epcced.github.io/2025-07-02_GreenHPC_UCL/, 2025.