

Reporting Carbon Footprint

Adam Calleja & Projeet Bhaumik

Objective and Assumptions

Objectives

To develop a framework to calculate and report a user's carbon footprint.

Assumptions

- The jobs are running exclusively on nodes.
- The jobs have completed.
- We know the number of nodes and the number of CPUs per node for each partition.
- A job's energy consumption consists only of the energy consumed by its nodes.
- The maximum runtime of a job is 36 hours.

Preparing The Data

→ First we obtained a .csv file containing SLURM *sacct* data for all jobs within a given time frame.

→ We also obtained a .csv file containing the number of nodes and CPUs per partition using the *sinfo* command.

→ Removing all the jobs that did not align with our assumptions allowed us to simplify the problem.

→ Sorting the DataFrame by the start time ensured that we could easily check for overlapping jobs later.

→ We ensured that all columns have the correct data type so that we could use their associated functionalities.

Finding Exclusive Jobs

There are two categories of exclusive jobs:

01

Jobs which have been allocated all of the CPUs on the node's they're using.

02

Jobs which are allocated only a part of the node, but do not overlap with any other jobs running on that node.

We maximised the efficiency of our code by:

- Only comparing job that run within 36 hours of each other.
- Parallelising our code using the *joblib* library. To parallelise our code we separated the DataFrame into a list of DataFrames each one containing the jobs that run on a particular node.

- 100% of exclusive jobs were from the first category.
- 41.5% of shared jobs overlap with jobs ran by the same user only.
- Before checking whether or not a job overlaps with other jobs, we needed to separate out the jobs that run on multiple nodes to get a separate row for each node it runs on.

Calculating The Carbon Footprint

In order to calculate the carbon footprint of a job, we need to query:

- the Victoria Metrics database to obtain the relevant power data.
- the carbon intensity API to obtain the corresponding carbon intensities.

We had two approaches when querying the APIs:

1. Query all of the necessary data initially and store it in a DataFrame.
2. Query the individual data only when we need it.

We ensured all times were in UTC so that they aligned with the APIs.

We have decided to compare the carbon footprint to the equivalent distance travelled in a medium sized diesel car to help illustrate the quantities.

Next Steps

- 01 Create an email template.
- 02 Write a python script to send consistent email reports to users.
- 03 Create a graphical dashboard that visualizes users' job data.
- 04 Take all jobs into consideration, starting with jobs which only overlap with other jobs submitted by the same user.
- 05 Take into account other causes of energy consumption (e.g., cooling, data transmission, etc.)

**Thank
you !**