Simulating Carbon Opportunity Cost at Grid Sites

Dr Dwayne Spiteri et al

HEPiX Spring Workshop - 16/04/2024

Contents

- ***I created a simulation of the computing site at Glasgow***
  with the aim to test the carbon consumption of various running methods.

- Why do this?
- Brief overview of the simulation
  - Validation
- Experiments Conducted
- Results
- Carbon Running Cost of new Servers
- Embedded Carbon
- Conclusions

Motivation

- There are a lot of good options for sites to choose from for future kit, especially as HEP workloads are running well on ARM architectures (ARM Compute Testing and Provision at Glasgow's Tier2)

- Can't always try before you buy - but you often can either get performance markers from your own tests or from the community

- Wanted to see if we can take the information that can be found about machines:
  frequency options - F | ***power - p(F)*** | ***HEPScore - s(F)***
  and simulate grid performance - especially with perceived benefits of running clocked down - and run weeks worth of work in minutes without affecting delivery of Tier2 service provision

- A dataset (from the UK National Grid ESO) is fed in to get an idea of real-time and forecasted carbon intensity. Estimates for power and Carbon Use are calculated per time-step, and produce metrics at the end of the simulation.

Simple Simulation Schematic

Simulation.py

1 ***Specify variable parameters of the simulation*** mainly:

- The number and type of nodes your cluster is made from (ampere, dell, grace)
- The amount of starting jobs and how many jobs are submitted per hour
- Maximum length of the simulation

DataLogger.py

7 ***Formats output statistics***

- Total (and average): CPU used, time elapsed, jobs started/ completed, (peaktime) power used and estimated C02e emissions.

WorkerNode.py

2 ***Create different kinds of worker nodes***

- Different types of worker node Attributes like hostnames, cores, memory, max power consumed, frequency
- Formulas for scaling power consumption
- Methods for automatically clocking up and down nodes
- Updates with whether the job is finished per timestep

Cluster.py

5 ***Spins up a cluster to run specified workloads***

- Defines things like amount of memory, cores available to outside sources from input worker nodes
- Define how you run the cluster in the event you want to try and run it differently - clock down nodes at certain times of day for example

6 ***Run Simulation***

- Calculates the total power used and CO2e emitted per timestep (10 minutes)
- Takes Jobs from the scheduler if able
- Passes data from the worker nodes to the DataLogger
- Ends when you run out of work, or out of time

JobFactory.py

3 ***Create different kinds of jobs from different VO's***

- Assume jobs run for samples amount of time drawn from previously measured distributions (for testing all jobs are set to be 5hrs long)
- Require amounts of memory and cores to be used

JobScheduler.py

4 ***Create a programme of work to be run on a cluster***

- Initialises jobs from ones requested from types of ones available
- Updates with jobs to be submitted to the cluster per time-step

Current Output

DataLogger.py

7 ***Formats output statistics***

- Total (and average): CPU used, time elapsed, jobs started/ completed, (peaktime) power used and estimated CO2e emissions.

```
========
Summmary
========

Total Simulated-time Duration       :  5.4 days
Total Real-time Duration            : 10.2 minutes

Jobs Started                        :  50000
Jobs Finished                       :  50000

Total CPU duration                  : 2000000.0 hours
Average CPU duration                : 5.00 hours

Total energy consumed by compute    : 1428.75 kWh
Peaktime (5-9pm) energy consumption: 256.65 kWh
Average energy consumption per job : 28.57 Wh

Estimated CO2e emmissions           : 112.188 kg
Estimated Peaktime CO2e emmissions : 21.009 kg
Average CO2e emmissions per job     : 2.244 g
Peaktime CO2e emmissions percentage: 18.726 %
```

- Each time the simulation is called, a file gets produced with the following information

Simulated and Real-time duration of the simulation

Job information

Total and Average CPU duration

Estimated energy used in total, during peak times and job-average

Estimated CO_2 (e)quivalent emissions for said work

Validation

- Validate the simulation running with one Dell reference node running at its highest frequency setting (maximally fill one node)

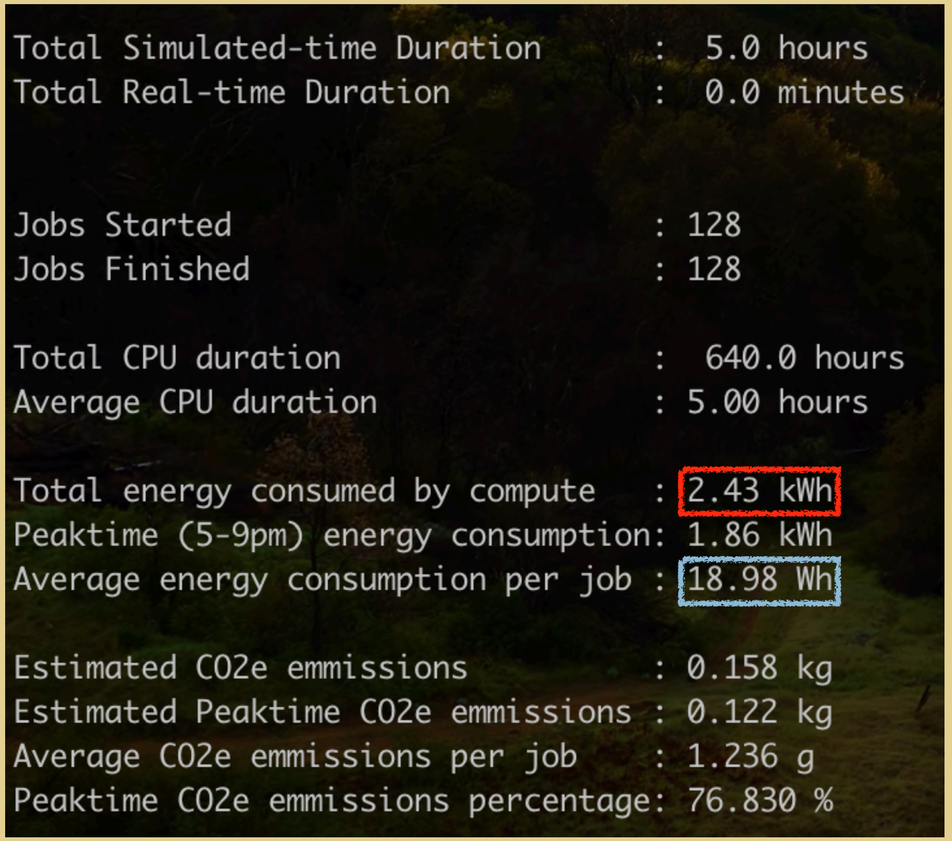
Expect

- The power displaced by the dell reference node at highest frequency = **486W**.
  This node has **128** threads.
  => Fill it with 128 single-core ‘GridPP’ jobs.
  => These jobs run for 5 hours,

- Total Energy Used:
  486W x 5h = 2430Wh = 2.43 kWh

- Avg Energy used per job
  2430Wh/128 jobs = 18.98 Wh

Observe



Simulated Experiments

Running a fixed workload

- Run 50k jobs of fixed length running on x86 nodes with the same specification and number that we have on the Glasgow grid. Compare this to when we run the same work but
  - run continuously with the frequency step of every node running on the cluster, reduced by one (and two) “frequency step(s)”.
  - reduce the frequency step of every node running on the cluster, by one (and two) “frequency step(s)” between the hours of 5pm and 9pm every evening.
  - reduce the frequency step of every node running on the cluster, by one “frequency step” when the forecasted use will be high in the next half hour segment.
- These jobs will finish within a week of simulation time and end the simulation - A test of savings for fixed amounts of work.
- Relative machine performance estimated using $P \times \frac{(\text{HEPScore})_{\text{machine@freq}_1}}{(\text{HEPScore})_{\text{machine@freq}_2}}$

Simulated Experiments

Running on different nodes types

- Run 50k jobs of fixed length running on some reference x86 nodes. Compare this to when we run the same work but
  - on an equivalent number of ARM Q80-30 and ARM Grace cores.
- Since the arm nodes had 160 cores, the grace 144, and the x86 reference node 128, I decided to make the target number of cores a fixed multiple of the least common multiplier → 5760.
- Our site has roughly 16k cores, so multiply 5760 by 3 to get 17280 cores. Hence the work will be run on
  - 135 x86 reference machines
  - 108 ARM Q80-30 machines
  - 120 Nvidia Grace machines

2xAMD64ht: Dual Socket AMD EPYC 7513 32-Core Processor (DELL)
CPU: 2 * x86 AMD EPYC 7513, 32C/64HT @ 2.6GHz (TDP 200W)
RAM: 512GB (16 x 32GB) DDR4 3200MT/s → 4 GB/core
HDD: 3.84TB SSD SATA Read Intensive
OS: CentOS 7.9 → Alma 9

- Relative machine performance estimated using $P = \frac{(\text{HEPScore})_{\text{machine w/ target CPU}}}{(\text{HEPScore})_{\text{machine w/ AMD-EPYC-7513 CPU}}}$

The Baseline Run: 50k at Glasgow

- Run 50,000 jobs of fixed length running on x86 nodes with the same specification and number that we have on the Glasgow grid

```
Total Simulated-time Duration       : 27.8 hours
Total Real-time Duration            :  1.0 minutes

Jobs Started                        : 50000
Jobs Finished                       : 50000

Total CPU duration                  : 250451.5 hours
Average CPU duration                : 5.01 hours

Total energy consumed by compute    : 1362.10 kWh
Peaktime (5-9pm) energy consumption: 292.48 kWh
Average energy consumption per job : 27.24 Wh

Estimated CO2e emmissions           : 94.188 kg
Estimated Peaktime CO2e emmissions : 19.462 kg
Average CO2e emmissions per job     : 1.884 g
Peaktime CO2e emmissions percentage: 20.663 %
```

- Total Energy Used: 10.36 MWh
- Avg Energy used per job = 22.55 Wh
- Carbon emissions per job = 1.50 g
- Fractional Peaktime emissions = 20.7

- Assume jobs started and not completed are half done for consumption metrics

- Will show results with various clockdown strategies, and then using different types of machine

Clockdown: 50k jobs Summary

	Baseline	One-Step Clockdown	One-Step Peaktime Clockdown	Two-Step Clockdown	Two-Step Peaktime Clockdown	Forecast Clockdown
Completion time (hours)	**27.8**	33.5	***29.2***	41.3	**30.0**	31.2
Total Energy (kWh)	**1362.1**	**1288.7**	1362.3	1434.4	1384.8	***1333.4***
Total Carbon Emmission (kg)	**94.19**	**87.61**	94.75	91.60	96.36	***91.55***
Avg Energy/ Job (Wh)	**27.24**	**25.77**	27.25	28.69	27.70	***26.67***
Avg Carbon emmisons/ job (g)	**1.89**	**1.75**	1.90	***1.83***	1.93	***1.83***
Emissions during peaktime (%)	**20.7**	22.60	***17.5***	21.2	**16.0**	21.7

- If you need to save power/carbon at specific times of the day, peaktime clockdown is the way to go as the work reduction is low.
- Running nodes permanently clocked down will save you power but at a heavy cost in work, if you want to save carbon, then you need to be smarter about how clock down
- Forecasting is a better strategy and manages to both use less energy overall.

Clockdown: 50k jobs Summary

Percentage difference relative to the baseline

	Baseline	One-Step Clockdown	One-Step Peaktime Clockdown	Two-Step Clockdown	Two-Step Peaktime Clockdown	Forecast Clockdown
Completion time (hours)	**27.8**	+20.5	**+5.04**	+48.56	**+7.91**	+12.23
Total Energy (kWh)	**1362.1**	**-5.39**	+0.01	+5.31	+1.67	**-2.11**
Total Carbon Emmission (kg)	**94.19**	**-6.99**	+0.59	-2.75	+2.30	**-2.80**
Avg Energy/ Job (Wh)	**27.24**	**-5.40**	+0.04	+5.32	+1.69	**-2.09**
Avg Carbon emmisons/ job (g)	**1.89**	**-7.41**	+0.53	**-3.17**	+2.12	**-3.17**
Emissions during peaktime (%)	**20.7**	+9.18	**-15.46**	+2.42	**-22.71**	+4.83

- Increase in completion time is roughly inversely proportional to reduction in work.
- If your aim is to reduce energy consumption during certain times. Peaktime clockdown reduces your power use at peak times, one (two) step clockdown can reduce these by 15% (23%) for a 5% (8%) loss in work. But this will come at the cost of 0.5% (2%) more carbon produced overall.
- Forecasting is a better strategy and manages to both use less energy and carbon emission overall at the cost of less work.

X86 vs ARM Summary

	Reference x86 50k	ARM-Q80 50k		ARM Grace 50k	
	Baseline	Raw	Relative to Baseline	Raw	Relative to Baseline
Completion time (hours)	15	11.5	-23.3%	7.0	-53.3%
Total Energy (kWh)	960.6	666.3	-30.6%	689.7	-28.2%
Avg Energy/Job (Wh)	19.21	13.33	-30.6%	13.79	-28.2%
Avg Carbon emmisons/job (g)	1.28	0.87	-32.0%	0.90	-29.6%
Operational Carbon Emmissions (kg)	63.7	43.3	-32.0%	45.1	-29.2%

- Now we run the same work, but on different clusters of the same size (17280 cores)
- Akin to testing different clusters in a vacuum
- In this simple example, if you had the money right now to spend on a full cluster, then we could save more energy and carbon by having ARM over current x86 nodes.
- Running at maximum frequency only.

Some caveats of differing architecture

- Ignores effects (which are largely architecture-specific) where the machine will run work at less than the maximum frequency and therefore use less power.

- The simulation doesn’t discriminate between physical and hyperthreaded nodes, but this efficiency difference shouldn’t come into play when all nodes are fully loaded all the time.

- I can edit back in different types of jobs from different VO’s with variable lengths, but that variance will not improve these figures - don’t have a way of codifying types of experimental work that different architectures could perform differently (floating point calculations etc.)

What do different procurements look like?

- So say if we can find a new shiny toy on the market, can we investigate the carbon savings replacing out our old kit different new ones.

- Try a case study. We phase out our old kit (~2136 cores) and replace it with either:
  17 Single-Socket AMD Sienna boxes (2176 cores) or
  17 Single-Socket Altra Max M128-30 boxes (2176 cores)
  machines for which we either have measurements of (Sienna), or can make reasonable extrapolations to the required values (Altra Max - extrapolated from from M128-28 boxes).

- We have gone with the latter based on measurements, can we quantify how much better this choice was?

- What does the carbon opportunity cost of running look like?

What do different procurements look like?

No Changes (2022 Running)

```
Total Simulated-time Duration      : 27.8 hours
Total Real-time Duration           :  1.0 minutes

Jobs Started                       : 50000
Jobs Finished                      : 50000

Total CPU duration                 : 250451.5 hours
Average CPU duration               : 5.01 hours

Total energy consumed by compute   : 1362.10 kWh
Peaktime (5-9pm) energy consumption: 292.48 kWh
Average energy consumption per job : 27.24 Wh

Estimated CO2e emmissions          : 94.188 kg
Estimated Peaktime CO2e emmissions : 19.462 kg
Average CO2e emmissions per job    : 1.884 g
Peaktime CO2e emmissions percentage: 20.663 %
```

Replacing Older Nodes w/Sienna

```
Total Simulated-time Duration      : 20.0 hours
Total Real-time Duration           :  0.6 minutes

Jobs Started                       : 50000
Jobs Finished                      : 50000

Total CPU duration                 : 259273.7 hours
Average CPU duration               : 5.19 hours

Total energy consumed by compute   : 969.80 kWh
Peaktime (5-9pm) energy consumption: 211.61 kWh
Average energy consumption per job : 19.40 Wh

Estimated CO2e emmissions          : 66.048 kg
Estimated Peaktime CO2e emmissions : 13.810 kg
Average CO2e emmissions per job    : 1.321 g
Peaktime CO2e emmissions percentage: 20.909 %
```

Replacing older nodes w/AltraMax

```
Total Simulated-time Duration      : 18.0 hours
Total Real-time Duration           :  0.5 minutes

Jobs Started                       : 50000
Jobs Finished                      : 50000

Total CPU duration                 : 252801.8 hours
Average CPU duration               : 5.06 hours

Total energy consumed by compute   : 939.53 kWh
Peaktime (5-9pm) energy consumption: 217.55 kWh
Average energy consumption per job : 18.79 Wh

Estimated CO2e emmissions          : 63.599 kg
Estimated Peaktime CO2e emmissions : 14.197 kg
Average CO2e emmissions per job    : 1.272 g
Peaktime CO2e emmissions percentage: 22.323 %
```

Procurement - Carbon Cost

	“2022” Site	- Old + Sienna		- Old + M128-30	
	Baseline	Raw	Relative to Baseline	Raw	Relative to Baseline
Completion time (hours)	27.8	20.0	-28.0%	18.0	-35.2%
Total Energy (kWh)	1362.1	969.8	-28.8%	939.5	-31.0%
Avg Energy/Job (Wh)	27.24	19.40	-28.8%	18.79	-31.0%
Avg Carbon emmisons/job (g)	1.88	1.32	-29.8%	1.27	-32.5%
Operational Carbon Emmissions (kg)	94.18	66.05	-29.8%	63.60	-32.5%

- In this example, replacing older kit with Altra Max instead of Sienna reduces your energy consumption per job by ~3% which could mean you operationally save ~2.4kg of carbon for every 50,000 jobs run.

- The older nodes are less efficient, replacing them no matter what will give you some sort of saving but **when and how** you do it is important because of...

Embedded Carbon

- The improvements listed are only on the carbon opportunity cost of **RUNNING** work. Assume an total operational carbon cost of Y.
- A significant component of carbon in a servers lifetime is in the embedded carbon. How we account for it will change the significance conclusions we have.
- Estimates of embedded carbon range from from 50-50 to 20-80 with operation costs.
- If a machine we purchase has an embedded carbon cost of X. Do we
  - Attribute it all to purchase and treat operational carbon as independent?
    Total Carbon 2025 = Y(2025) -> Run in a way that reduces carbon
  - Assume a set lifetime of operation (5 years) and split the cost for each year - X/5?
    Total Carbon 2025 = X/5 + Y(2025) -> Optimisations of Y(2025) less impactful
  - Split the embedded carbon cost over every job you run?
    Total Carbon 2025 = X(2025) + Y(2025) -> Reduction in Jobs wastes embedded carbon

Conclusions and Future Work

- A simulation has been created to try and test different kinds of operation of Tier2 sites. It's modular, so different types and amounts of machines can be span up and run
- **Results here are preliminary, it's the first version of this simulation.**
- Sites could try to reduce impact of their loads on the grid by clocking down nodes during peak times but this doesn't reduce the overall carbon produced. Tuning clockdowns to forecast data can though, so should be potentially investigated in the future.
    - Easy enough to create an Ansible script that roles this instruction out to all nodes for example?
- We have simulated the carbon opportunity cost of replacing old kit, and have reinforced our decision to purchase more ARM machines - **Our M128-30's should be arriving any moment now**
- At the moment the simulation uses overall grid CO2 numbers, the carbon energy data is available split by region. The simulation also runs on an entire year of example data, as the simulation can start at any given time rather than using the time the simulation started, target specific times of year can be targeted for further study.
- Improvements will be tempered by how we treat embedded carbon in the future.

Backup

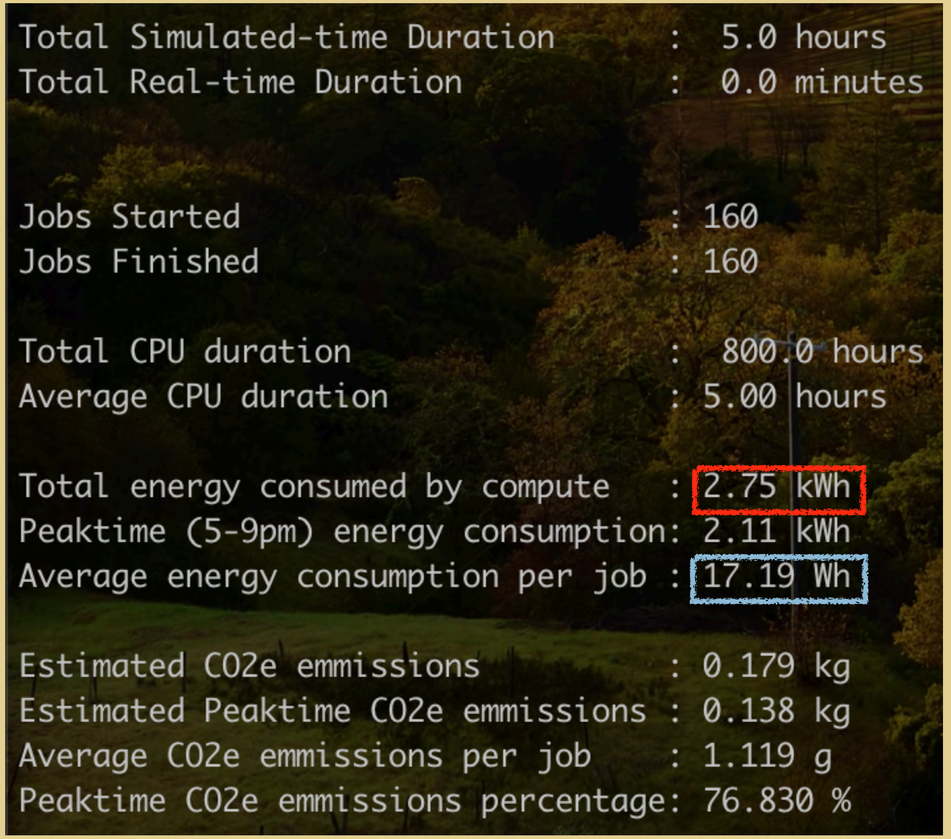
Validation of Other machines

- Validate the simulation running one **arm Q80-30** node running at its highest frequency setting (maximally fill it). No HEPScore scaling is present here.

Expect

- The power displaced by an Q80-30 node at highest frequency = **550W**.
  This node has **160** hyperthreads.
  => Fill it with 160 single-core ‘GridPP’ jobs.
  => As these jobs run for 5 hours
- Total Energy Used:
  550W x 5h = 2750Wh = 2.75 kWh
- Avg Energy used per job
  2750Wh/160 jobs = 17.19 Wh

Observe



Validation of Other machines

- Validate the simulation running one **arm-grace** node running at its highest frequency setting (maximally fill it). No HEPScore scaling is present here.

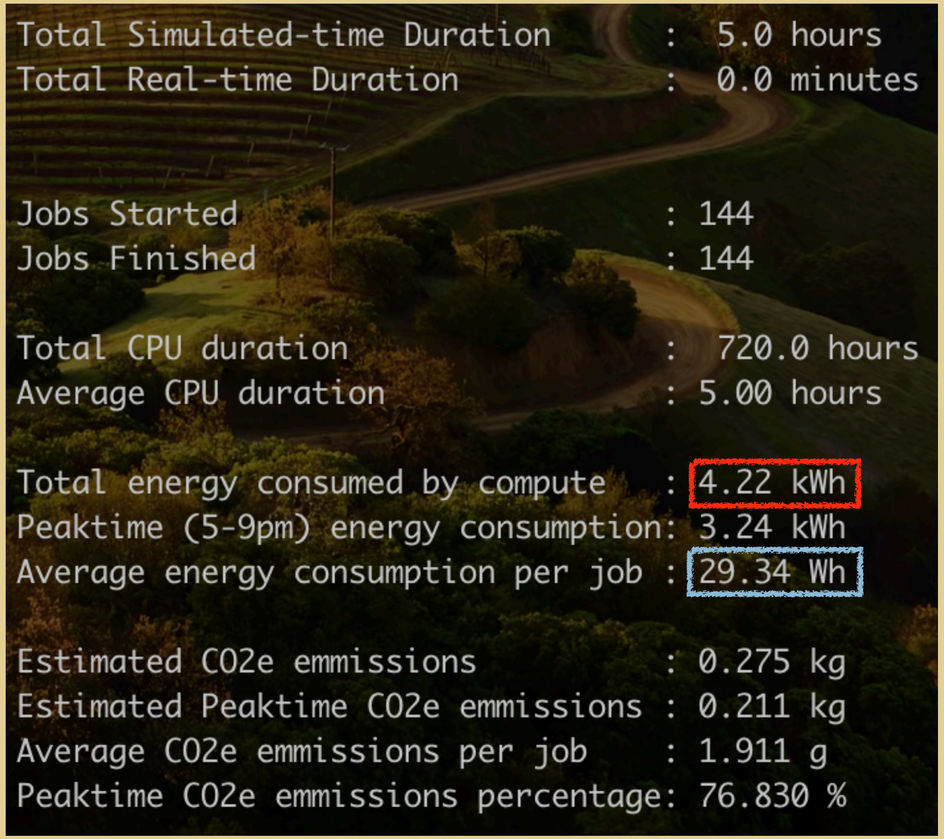
Expect

- The power displaced by a Grace node at highest frequency = **850W**.
  This node has **144** hyperthreads.
  => Fill it with 144 single-core ‘GridPP’ jobs.
  => As these jobs run for 5 hours

- Total Energy Used:
  845W x 5h = 4225Wh = 4.23kWh

- Avg Energy used per job
  4225Wh/144 jobs = 29.34Wh

Observe



Simulated Experiments

Running Continously

- Run 1M jobs of fixed length running on x86 nodes with the same specification and number that we have on the Glasgow grid and stop the simulation after one week of running. Compare this to when we run the same work but
  - run continuously with the frequency step of every node running on the cluster, reduced by one (and two) “frequency step(s)”.
  - reduce the frequency step of every node running on the cluster, by one (and two) “frequency step(s)” between the hours of 5pm and 9pm every evening.
  - reduce the frequency step of every node running on the cluster, by one “frequency step” when the forecasted use will be high in the next half hour segment.
- These jobs will finish within a week of simulation time and end the simulation - A test of savings for fixed amounts of work.
- Relative machine performance estimated using $P \times \frac{(\text{HEPScore})_{\text{machine@freq}_1}}{(\text{HEPScore})_{\text{machine@freq}_2}}$

Clockdown: 7 Day Run

Simulation output for different running strategies at Glasgow

	Baseline	One-Step Clockdown	One-Step Peaktime Clockdown	Two-Step Clockdown	Two-Step Peaktime Clockdown	Forecast Clockdown
Jobs Completed	**450576**	305592	**422512**	236056	**408688**	376432
Total Energy (MWh)	10.34	**7.11**	9.80	**6.29**	9.67	**8.61**
Total Carbon Emmission (kg)	688.7	**473.4**	650.1	**419.0**	640.3	**560.2**
Avg Energy/ Job (Wh)	**22.55**	**22.67**	22.77	25.78	23.20	**22.41**
Avg Carbon emmisons/ job (g)	**1.50**	**1.51**	1.51	1.71	1.53	**1.45**
Emissions during peaktime (%)	**17.2**	17.2	**12.5**	17.1	**11.2**	15.8

- If you need to save power/ carbon at specific times of the day, peaktime clockdown is the way to go as the work reduction is low.
- Running nodes permanently clocked down will save you power but at a heavy cost in work, if you want to save carbon, then you need to be smarter about how clock down
- Forecasting is a better strategy and manages to both use less energy overall. Lower Energy/Job => Higher HEPScore/Watt

Clockdown: 7 Day Run Summary

Percentage difference relative to the baseline

	Baseline	One-Step Clockdown	One-Step Peaktime Clockdown	Two-Step Clockdown	Two-Step Peaktime Clockdown	Forecast Clockdown
Jobs Completed	450576	-32.18	-6.23	-47.61	-9.30	-16.46
Total Energy (MWh)	10.34	-31.24	-5.22	-39.17	-6.48	-16.73
Total Carbon Emmission (kg)	688.7	-31.26	-5.60	-39.16	-7.03	-18.66
Avg Energy/ Job (Wh)	22.55	+0.53	+0.98	+14.32	+2.88	-0.62
Avg Carbon emmisons/ job (g)	1.50	+0.67	+0.67	+14.00	+2.00	-3.33
Emissions during peaktime (%)	17.2	+0.00	-27.33	-0.58	-34.88	-8.14

- Peaktime clockdown can give you a 27% (35%) reduction in emission during peak times for 6% (9%) reduction in overall work.
- For forecasting a ~16% jobs can give you a ~18% reduction in runtime carbon with respect to the baseline
-