A parameterized test bed for carbon aware job scheduling

Ein parametrisierbares Testbed für kohlenstoffbewusste Jobplanung

Vincent Opitz

Arbeit zur Erlangung des Grades “Master of Science” der
Digital-Engineering-Fakultät der Universität Potsdam

A parameterized test bed for carbon aware job scheduling

Ein parametrisierbares Testbed für kohlenstoffbewusste Jobplanung

Vincent Opitz

Arbeit zur Erlangung des Grades “Master of Science” der Digital-Engineering-Fakultät der Universität Potsdam

A parameterized test bed for carbon aware job scheduling
(Ein parametrisierbares Testbed für kohlenstoffbewusste Jobplanung)
von Vincent Opitz

Arbeit zur Erlangung des Grades "Master of Science" der Digital-Engineering-Fakultät der Universität Potsdam

Betreuer:in: Prof. Dr. rer. nat. habil. Andreas Polze
Universität Potsdam,
Digital Engineering-Fakultät,
Fachgebiet für Betriebssysteme und Middleware

Gutachter:in: Prof. Dr. Jack Alsohere
University of San Serife
Facultiy of Computer Doings

Amelia van der Beenherelong
ACME Cooperation

Datum der Einreichung: 1. September 2024

Zusammenfassung

ummarize thesis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

1 Introduction **3**

2 Background **5**

3 Related Work **9**

4 Methodology **11**

4.1 Improving the current Job model 11

4.1.1 Power Measurements on Machine Learning Jobs 11

4.1.2 Defining a new model 13

4.2 Choosing an implementation approach 13

4.2.1 Carbon-aware scheduling via a slurm plugin 14

4.2.2 Using a Simulation approach 14

4.3 building ontop of the existing gaia sim 15

4.4 Evaluating carbon-aware scheduling with the new job model 15

5 Results **17**

6 Discussion **19**

7 Future Work **21**

References **23**

Zusammenfassung

S

ummarize thesis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1 Introduction

- I could start with a fact stating that datacenters use X amount of energy which in turn emits carbon for its production
- We can reduce the amount of carbon emitted for the same amount of work by scheduling our work in smart way
- this is not a new idea; datacenters around the world are already using data provided by e.g. electricitymaps
- one thing that is not represented in literature however is the heterogenity of jobs, as well as the emphcheckpoint & resume capabilities of some jobs, particularly machine learning jobs which are projected to make UP X amount of energy in the coming years [QUOTATION NEEDED]
- I want to create a better model for jobs in datacenters
- I then want to use that new model to investigate wether we can exploit the heterogenity of jobs to further reduce carbon emissions
- One hypothesis is for example that, we can match high-power phases of jobs to the lowest carbon-intensive time frames, so we do not "waste" those rare times with low-power phases
- We can also investigate wether the previously proposed stop & resume techniques still work if the assumption that stopping and resuming a job has no overhead is broken.
- We hypothesis that a stop & resume strategy can stil lead to reduced carbon emissions but only on certain jobs.

2 Background

This chapter will be used to introduce some basic terminology aswell as some basic arguments on why carbon-aware scheduling is useful.

The composition of the public grid The public grid is made up of many different energy producers. This *energy mix* is composed of different sources: low-carbon technologies like solar, wind, or hydroelectricity and carbon-intensive sources like coal, gas, or oil.

Figure 2.1 shows one such energy example, for a specific location, in this case Germany. As it is summer, solar production makes up large amount of power at that time. Germany is further interesting in the sense, that it is a country that has no nuclear power stations anymore.

Over the day, the demand and supply for power changes. Carbon-efficient sources will generally follow the weather: Solar production follows the amount of sun shining and thus follows a *diurnal* rythm over the day. Wind and hydroelectricity will also depend on the weather. Nuclear energy is generally also considered carbon-efficient, but is generally not useful for carbon-efficient scheduling as the amount of energy produced from is generally stable throughout the day, meaning that there is little use in deffering work to a later time.

QUOTATION NEEDED, maybe use look in on of the related work papers

As near-time weather predictions have high accurarcy , this enables X.

Thus, at each point in time the amount of CO_2 per unit of electric power can be determined by averaging the amount of power each source supplies to the grid and how much carbon it emits. Figure 2.2 is an example X-day timeframe of the carbon-curve.

QUOTATION NEEDED

Power grid Signals Carbon-aware scheduling commonly works two possible metrics or signals: the *average emissions* is the metric describing the amount of carbon per unit of energy, basically what has been used in the text so far. Another metric is the *marginal emissions*, answering the question of "if more energy is used at this point in time, how much carbon would that cost?". The answer to that question is usually that non-renewable power plants have to increase production as renewable sources cannot increase production to meet demand (as the sun and wind intensity are set by the weather). Going by the marginal metric, any kind of carbon aware scheduling would be pointless, as any work at any point in time would result in the same increase of carbon. Thankfully, there are secondary reasons that would render carbon-aware scheduling useful. One of them are is *curtailment*. This encompasses any methods that reduce the amount of produced renewable energy. As the power grid always needs to have a balance between demand and production, curtailment methods such as turning of wind turbines, selling power at a loss, or charging batteries may be used. Carbon aware scheduling via the average emissions signal, would lower the amount of curtailment needed, as demand for energy would increase at those times. Another argument can also be made that by increasing

find the quote

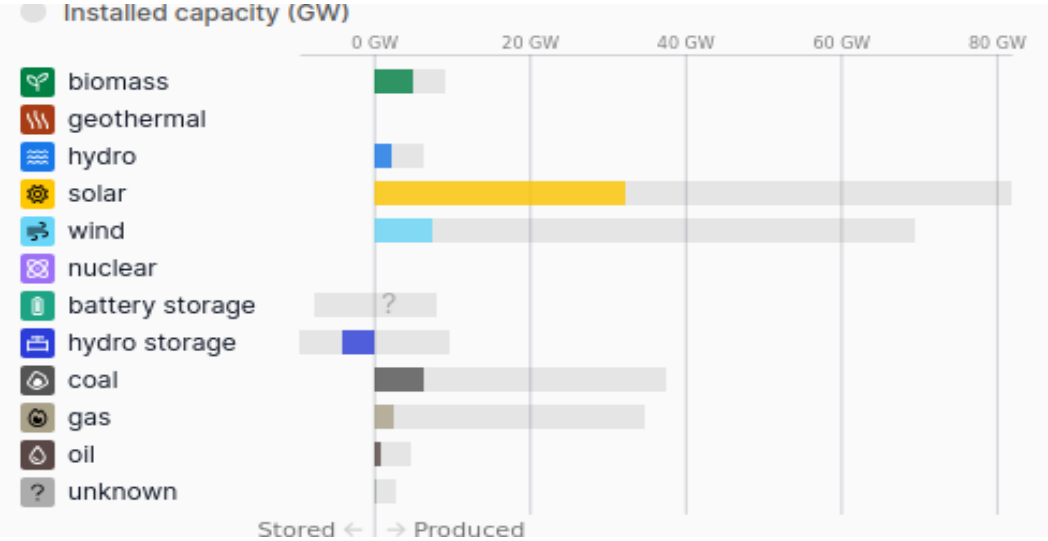
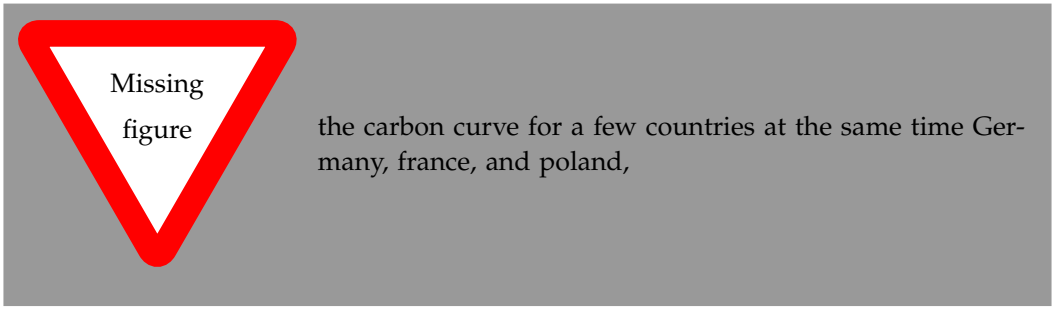


Figure 2.1: An example snapshot of the energy mix in Germany, at DATE



the carbon curve for a few countries at the same time Germany, france, and poland,

Figure 2.2: Carbon in per unit of energy in select countries

power demand during times when renewable production is high, energy producers would be "signalled" that there is demand for renewable energy. Lastly, as renewable energy is generally cheaper in production than non-renewable energy , scheduling work on low carbon-periods coincides with cheaper energy prices as well. While this would not reduce costs on most energy contracts, there are also some contracts that use dynamic pricing[1]

need a citation here!

Types of work in a datacenter According to [1], there are 3 environments in which scheduling may take place. *Batch systems* describe environments in which there is no user interaction. A user may submit their job and it would be executed according to the scheduler at some point in time. . On the contrary, in *interactive* settings, a user would be interacting with the system "live", and would thus expect quick responses to their inputs. The last environment would be a *real time* sytem. There deadlines and predictability would dictate how a scheduler would operate.

does this need more examples?

For this work's topic, carbon-aware scheduling, only batch systems will be looked at as they allow more freedom to the times of scheduled jobs.

I could also add more stuff about metrics and so on, but Im not using them yet in my implementation

welche metriken gibt es POI-A, etc etc.

- energy is produced by different sources, leading to a different amount of carbon/unit of energy over time.
- Welche Methoden gibt es carbon einzusparen (vllt. auch in der related work) (temporal, spatial, ressource scaling)
- Begründen warum sich das lohnen kann (marinal / relative carbon), kein wasting von erneuerbaren energien (losses among saving energy into batteries / "curtailment" !!)
- welche metriken gibt es POI-A, etc etc.
- Jobs <-> dynamic ernergie verbrauch, wie wird sowas gemessen?

Power intake of a computer As there will be power measurements in 4.1.1, some basic understanding of energy and power used for computation will be provided:

- I could mostly borrow from the EBRH slides; there is some base power needed that is correlated to the hardware (dynamic and static energy)
- this also depends on frequency (which is why later we set our cpu frequency to some hardcoded value [or dont do that in the case of my GPU lol])
- basically, use this paragraph to outline all things we take care of in my power measurements

[1] one example for dynamic pricing would be Tibber: `https://tibber.com/de`

3 Related Work

A systematic approach I used the following system for finding related work to my topic; first, my supervisors and I would brainstorm for search engine keywords. The specific words are in the attachments, but can be grouped into two groups: one for all things carbon-aware and one for keywords about servers / computing / HPC and such.

Literaturrecherche aufarbeiten und hier verlinken

Using these two groups, I could then create a google scholar queries via the cross product between them. Using the double-quotation feature would further limit the results. For each query, I would then read the abstracts of roughly the first 5 results, depending on if their titles sounded subjectively fit. Additionally, I would also further explored papers by finding new ones through *connected papers*[2] These would then be entered into a spreadsheet: for each query, 5 paper titles. I then "rated" them into three categories:

- green, meaning that they seem very connected and are good first entries into the topic
- orange, which would indicate that are are somehow connected to the paper and might be read at a later date
- red, the paper is either irrelevant or had some other flaw. These would not be touched again in the course of my work

With this approach, X abstracts were read. Figure 3.1 shows that X abstracts were deemed "good" for the purpose of this work. The complete list of papers analysed can be found in

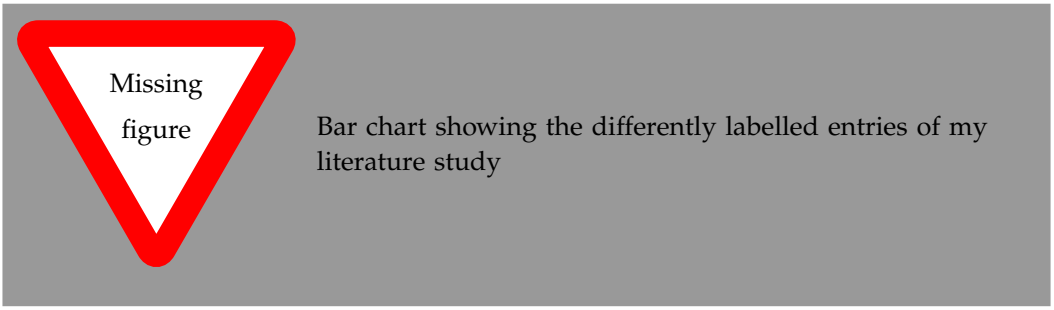
figure out results of the literature stufy

THE ATTACHMENTS

I would like to highlight some papers:

GreenSlot: Scheduling energy consumption in green datacenters[0] seems to be the first paper that deals with carbon aware scheduling by implementing it as a slurm plugin. In

[2] https://www.connectedpapers.com/



Bar chart showing the differently labelled entries of my literature study

Figure 3.1: Results of the literature study

contrast to our scenario, where we try to optimize carbon emissions via the public electricity grid, GreenSlot is about datacenters having their own renewable energy production (solar panels on the roof). Using weather data, *GreenSlot* would then predict when solar energy production is high, scheduling jobs to those time frames.

The War of the Efficiencies: the Tension between Carbon and Energy Optimization [o] outline the different ways of carbon aware computing. Among those is *temporal shifting*, the idea that jobs can be executed later when energy is more carbon efficient, is also the main idea for my work. They also use *spatial shifting*, moving jobs across the globe to areas where higher carbon efficiency is possible. *Ressource scaling* uses dynamic amounts of hardware according to carbon emissions. In the end, *rate shifting* is the idea to also scale hardware frequencies. During carbon-efficient times, CPU speeds would be increased, leading to faster processing speeds and more energy usage.

All of these techniques are then tested under various parameters. My work will only make use of the temporal shifting, and abstract away the other methods of further saving carbon.

Let's wait awhile: how temporal workload shifting can reduce carbon emissions in the cloud is to be one the paper I'll be building on the most. [**wiesner_lets_2021**] uses a simulation approach, to simulate temporal shifting. Their workload model consists of known length jobs that can use *checkpoint & restore* to be executed at different time slices. They would further use different traces and test these traces under the assumption of different job-deadlines, meaning that each job would have to be completed by a certain timeframe, and also different regions of the world as described in the background part. Their main take-aways are that increased deadlines lead to reduced carbon emissions, but that this effect also has diminishing returns. They also deduced that regions such as california, with high amounts of solar power, have higher potential for carbon-savings in comparison to nuclear-heavy regions such as france.

think about linking this to the earlier part

4 Methodology

Based on our related work, describe (using the goals outlined in the beginning) the approach I take: creating a better model to be used in carbon-aware scheduling using the model and evaluating it.

4.1 Improving the current Job model

We should probably argue why the current job model used in literature is not sufficient (WaitAWhile assumed constant power usage and no overhead from stoppig and resuming)

4.1.1 Power Measurements on Machine Learning Jobs

Options for measuring power There are multiple options for measuring the power of a given computer. One way of classifying these options would be under them being either *logical measurements* or *physical measurements*.

Logical ones would create a model on some metrics and derive the used power. One example would be using Linux' *perf* tool to read hardware performance counters.

Advantages of choosing a logical approach would be that no external hardware is needed and that the overhead of the measurement would be low, as the hardware counters are being kept track of anyway. Disadvantages on the other hand would be that such a model would have to be created or chosen and would include some form of error as all models do.

Physical measurements follow another route; measurement devices would be put between the operating hardware and the power supply. The point where a power measuring device is inserted would dictate what could and could not be measured, a wall mounted measurement device could only measure all power going into a computer and not differentiate between individual programs.

Advantages of physical measurements are that they can give a more holistic measurement of a system as would be the case for a wall mounted measurement device. Portability is an issue however, unlike operating-system supported tools such as perf, a measurement device would need more effort to be used on another system (or be entirely not useable, for example when such devices are only rated for a certain power level).

Due to having a power-measurement tool on-site in our university and it allowing whole-system measurements directly, I chose to follow the physical measurement option.

Measurement tool The concrete tool used is the *Microchip MCP39F511N Power Monitor (henceforth called MCP)*, which can be inserted between the device to test and the wall mounted power supply . The MCP can report the current power consumption in 10 mW

Für die Durchführung von Messungen ist es zwingend erforderlich, die verwendete Testumgebung (Hardware wie auch Software) zu dokumentieren sowie Messparameter gewissenhaft zu wählen. Zu den wichtigsten Parametern gehren die Anzahl der Messwiederholungen, die Anwendung von Warmup-Läufen sowie die Identifikation und Vermeidung möglicher störender Einflüsse.

I feel like I should add a paragraph somewhere explaining what I want to measure

this needs some more here

I should add a nice picture here of the MCP

steps, each 5ms. I then used *pinpoint*, a tool for energy profiling that can use different inputs, among them being the MCP, to read out its data.

The test environment The experiments were run on my personal computer, the components of that are listed via the *hwlist* tool, with unnecessary columns and rows being redacted for brevity:

Listing 4.1: Hardware that was measured

```
   $ lshw -short -C processor -C memory -C display -C bus
Class          Description
==========================
bus            AB350 Gaming K4
memory         16GiB System Memory
processor      AMD Ryzen 5 1600X Six-Core Processor
display        GP104 [GeForce GTX 1070]
```

Information about the operating system is given via *hostnamectl*, again some parts redacted:

Listing 4.2: Used operating system information

```
    $ hostnamectl
   Operating System: Ubuntu 24.04 LTS
             Kernel: Linux 6.8.0-39-generic
```

Measured Program Machine learning (ML) was used as the main motivation for checkpoint & resume scheduling in the related works[**wiesner_lets_2021**] and thus was also chosen by me to be measured and modeled.

The concrete model and framework is secondary for our measurement. In my case, a small model would be chosen in order to have less data for processing aswell as faster iterations on the measurement script.

There is a vast amount of machine learning frameworks. For a high-level model, the featureset of the framework only needed to support checkpointing, resuming, and some basic form of logging. Glancing at the documentation of popular frameworks such as *torch*, *tensorflow*, and *huggingface* shows that these features are commonly supported.

With not much bias towards any framework, huggingface was chosen because my supervisor Felix supplied a sample "hello-world"-esque machine learning script for python *roberta.py*[3]

The huggingface trainer supports callbacks, I thus modified the code by adding timpestamped logs. These "Events" would be output into another .csv File I could later use.

later use for WHAT?!

- describe the specifics of my measurement setup (multiple runs, system at rest, measure before and after)

[3] https://github.com/Quacck/master-thesis/blob/main/power-measurements/roberta.py

- also be sure to mention that we added logging to the script execution (to be used for the phases later)

Creating repeatable measurements A script[4] was created to execute each experiment. On a high-level view, the following experiments were conducted:

1. Run the whole program start to finish
2. Run it partially, checkpointing after some step, sleeping, resuming from that step
3. Run it partially, checkpointing after some step but aborting before the next checkpoint. Then resume as above.
4. Run only the startup phase up until the ML would start

Measurement results

- Warum ist das interessant
- which jobs did I measure and why?
- Wie habe ich die Messungen ausgeführt, MCP beschreiben, sowie pinpoint als Schnittstelle
- Experimentier-parameter, also wodurch versuche ich sicherzustellen, dass meine Messungen auch sinn machen / reproduzierbar sind usw usw.
- Wie habe ich das ausgewertet, schöne graphs zeigen

4.1.2 Defining a new model

Now that we know what a high-level job looks like, we can pick it apart and reduce the real-world measurements of one program to a more generic model.

- we can deduce phases
- each phase has a constant power draw
- give an example of how to represent the real-world measurements into a model
- now we should proof that the model actually represents the reality to a certain degree (error analysis)
- have a cute graph showing the measurements and the model-"measurements" next to each other
- also show that the stop-resuming functionality can be represented with our model

4.2 Choosing an implementation approach

We first need to explain why we chose our approach (building upon exisiting work inside GAIA). The other option that is not using a simulation would be to schedule real jobs, for example by creating a slurm plugin.

[4] https://github.com/Quacck/master-thesis/blob/main/power-measurements/measure_roberta.sh

We can then evaluate how well a slurm plugin would work for our given Forschungsragen. End that section by deeming the plugin idea as unfit, we can then shift to arguing for the simulation approach as that is also something that just came out in related work (perhaps we should see wether we list GAIA as related work or introduce it just then)

4.2.1 Carbon-aware scheduling via a slurm plugin

thank god I made notes

- why do we choose slurm specifically, and not other software like kubernetes etc. => because its also being used in scorelab at the same location as I am in
- was ist slurm? => "open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters"
- how does the slurm plugin system work? what do we need to do to get there?
- what problems occured?
- ...thus I ultimatly choose not to pursue implementing a plugin

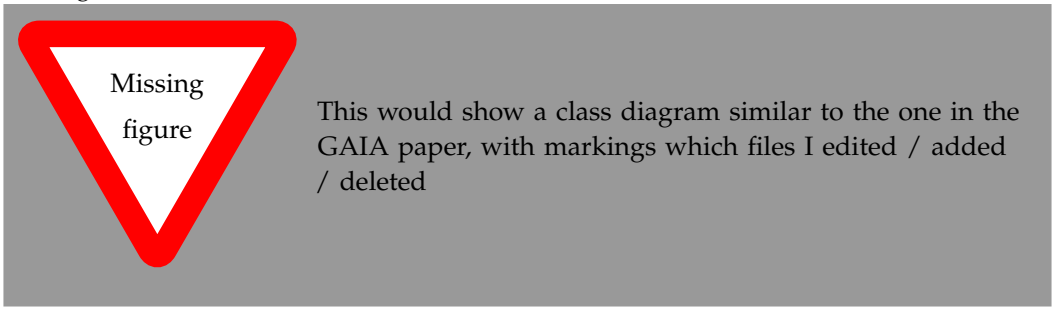
4.2.2 Using a Simulation approach

Thankfully, just at that time a new paper [o] was released. They made a prototype testbed for simulating job scheduling on cloud providers. These Jobs could be executed on spot instances (cheap VMs that seek to increase cloud utilization), on-demand instances (short-notices VMs that are thus more expensive) or pre bought VMs (medium cost, but may be wasted), the paper then discussed balancing carbon- and dollar costs.

improve this

The good part is that within that testbed, many scheduling approaches outlined in the related work section were implemented - a WaitAWhile Implementation for example . I could now extend this testbed and would ontop get something to compare against!

The way to do this section would be to a) describe what was aleady there, and b) what I changed



Describe the figure, explain why I am for example removing the part about the dollar-costs and the slurm-scheduler adapter. I should also describe which parts of the program I am modifiying to tackle my Forschungsfragen

Stuff I should describe about the simulation before I added anything:

Assumptions of the simulation

- Joblängen sind bekannt

- Jobs können zeitlich verschoben werden (begründet daraus, dass sie als Batch Jobs submitted werden, andere Jobs werden hierbei nicht betrachtet)
- User geben dabei an, wie lange der Job verschoben werden darf
- Die carbon curve auf dem electrical grid ist für kurze Zeiträume in der Zukunft bekannt
- Die Hardware ist zZ nicht begrenzt. Das war in der related work auch nicht so. Eigentlich wäre es spannend sich das anzuschauen, allerdings sind die bisherigen Scheduler halt darauf garnicht gemünzt, da werden alle Jobs unabh. voneinander gescheduled. Man könnte das via publicCloud argumentieren, allerdings wäre das questionable, in wie fern der scorelab trace benutzt werden kann (da das ja auf in einem lokalem datacenter läuft)
- TODO: Joblängen sollten dem Scheduler nicht bekannt sein. Die Workloads aus GAIA werden allerdings so gescheduled als ob man perfekte Knowledge hat. Das reicht zwar für ein upper bound an carbon savings, ist aber nicht sehr realistisch.

Data being used here i could describe which data is already being used (the traces, aswell as the historical carbon data)

- Welche Traces gibt es, wodurch werden die characterisiert? (Länge, Anzahl, etc, etc) Vllt. kann man hier nen coolen vergleich erstellen, Auch könnte man ein paar Sätze darüber schreiben, wie die bisher in GAIA aufgenommen wurden.
- Wie den scorelab trace benutzen und übersetzen? Gerne auf ner halben Seite aufschlüsseln, was die einzelnen Attribute aus sacct bedeuten.
- Ansonsten kann man noch die dynamic ernergy sachen als Datenquelle auflisten, bzw. das mini experiment mit fmnist und roberta

4.3 building ontop of the existing gaia sim

Which parts of GAIA do I add on? => this should just be the schedulers and the part where the carbon is calculated, this ensures that

4.4 Evaluating carbon-aware scheduling with the new job model

Hi!

5 Results

- here we would try to show off the difference between power-and-phase-oblivious scheduling and my new implementation which can make use of that
-

6 Discussion

welche schlüsse können wir aus den ergebnissen ziehen?

7 Future Work

lol

Bibliography

[0] Walid A. Hanafy, Roozbeh Bostandoost, Noman Bashir, David Irwin, Mohammad Hajiesmaili, and Prashant Shenoy. "The War of the Efficiencies: Understanding the Tension between Carbon and Energy Optimization". en. In: *Proceedings of the 2nd Workshop on Sustainable Computer Systems*. Boston MA USA: ACM, July 2023, pages 1–7. ISBN: 9798400702426. DOI: 10.1145/3604930.3605709 (cited on page 10).

[0] Walid A. Hanafy, Qianlin Liang, Noman Bashir, Abel Souza, David Irwin, and Prashant Shenoy. "Going Green for Less Green: Optimizing the Cost of Reducing Cloud Carbon Emissions". In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*. Volume 3. ASPLOS '24. New York, NY, USA: Association for Computing Machinery, Apr. 2024, pages 479–496. ISBN: 9798400703867. DOI: 10.1145/3620666.3651374 (cited on page 14).

[1] Andrew S. Tanenbaum and Albert Woodhull. *Operating systems: design and implementation: [the MINIX book]*. en. 3. ed. The MINIX book. Upper Saddle River, NJ: Pearson Prentice Hall, 2006. ISBN: 978-0-13-142938-3 978-0-13-505376-8 978-0-13-142987-1 (cited on page 7).

[0] Íñigo Goiri, Kien Le, Md. E. Haque, Ryan Beauchea, Thu D. Nguyen, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. "GreenSlot: scheduling energy consumption in green datacenters". en. In: *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. Seattle Washington: ACM, Nov. 2011, pages 1–11. ISBN: 978-1-4503-0771-0. DOI: 10.1145/2063384.2063411 (cited on page 9).

Eidesstattliche Erklärung

Hiermit versichere ich, dass meine Arbeit zur Erlangung des Grades “Master of Science” der Digital-Engineering-Fakultät der Universität Potsdam mit dem Titel “A parameterized test bed for carbon aware job scheduling” (“Ein parametrisierbares Testbed für kohlenstoffbewusste Jobplanung”) selbständig verfasst wurde und dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt wurden. Diese Aussage trifft auch für alle Implementierungen und Dokumentationen im Rahmen dieses Projektes zu.

Potsdam, den 1. September 2024

_____

(Vincent Opitz)