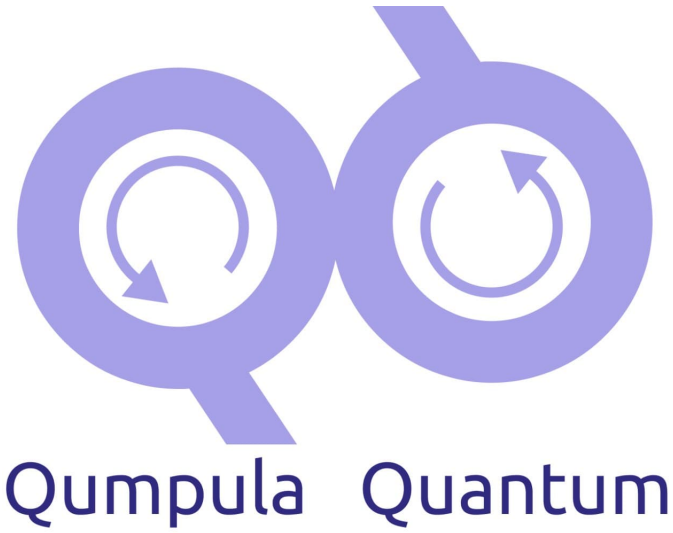
Quantum Computing -based Optimization for Sustainable Data Workflows in Cloud Infrastructures

team: Qumpula Quantum

Valter Uotila

QHack Open Hackaton during February 21–25, 2021



Team description

`Hello QHack!` My name is Valter Uotila. This April I will start as a Ph.D. student at the University of Helsinki. In the coming years, I will be researching topics related to multi-model data management, their optimization with quantum computers, and applications of category theory combining these two fields. This work is about the two first topics: data management and quantum computing.

One of my goals is also to introduce possibilities of quantum computing to the database community. I have participated in multiple database conferences during the past years and I think the field would benefit from quantum computing enhanced optimization methods. The plan for this work is to make this an interesting demonstration system that I can represent at a relevant database conference such as VLDB, Sigmod, or ICDE. Besides quantum computing, this work will raise awareness on sustainability aspects.

Although the team Qumpula Quantum has now just one person, the same team has worked on different quantum computing-related topics [1] previously.

Introduction and motivation

Quantum computing is hard. The problems quantum computing will solve should be hard too.

There are some studies [1] which indicate that data centers produce around 2 percent of the emissions in the world. See the appendix of this document to find more details about sustainability of data centers. Especially we note that Google has implemented an interesting Carbon Footprint tool which enables an individual user to track their computations' carbon footprint. Inspired by the idea that in the future such detailed data would be available from other companies as well, this work proposes an optimization algorithm to divide a workload among the data centers and machines so that the carbon footprint is minimized.

We are tackling here a problem that is currently out of reach technically. Anyway, hackathons are rarely about being very realistic. We hope that this work encourages companies to implement sustainable solutions for data management and also make customers aware that they should demand sustainable cloud computing services and critical truthful statistics about sustainability. It also demonstrates an interesting application of quantum computing to the database field although the same problem could be solved with classical computing.

Even if it is hard to prove that quantum computing performs more efficiently than classical computing, we want to point out that the efficiency of computations is just one aspect. Noting this work's sustainability theme, there should be more research on the energy consumption of quantum computers. At least theoretically [2], Landauer's principle shows that quantum computing could be

[1] https://www2.helsinki.fi/en/news/data-science-news/valter-uotila-and-sardana-ivanova-qualified-for-bmw-quantum-computing-challenges-finals

performed with very low energy consumption. Of course, in the reality is different. We believe that in the future quantum computers will become the most sustainable computers to run.

Problem definition

The problem we are facing here is informally the following. There are companies, such as Databricks, Snowflake and Aiven which are collecting different cloud databases into a single service. In theory, they could provide a service where the actual cloud service provider is hidden from the end-user. For simplicity, we assume that the user (or the customer) does not care about the price of the service. Their only requirement is to run their workloads as sustainably as possible. We could also include the price parameter into the solution and optimize over the price and sustainability.

We aim to define the problem as generally as possible. Although we are developing the model at the level of companies that are collecting various first-hand cloud computing companies together, it works also for companies that are providing the cloud computing resources.

Let us define the problem more formally. Assume that the company X is operating as an aggregator as described in the beginning. The company X is offering cloud computing services from the first-hand cloud computing providers $\{Y_1, \ldots, Y_n\}$. We call these companies *cloud partners*. For example, Databricks' cloud partners are AWS, Azure, and Google.

Each cloud partner Y_i, for $i = 1, \ldots, n$, operates multiple data centers $(d_1^i, \ldots, d_t^i)$ across the world. For simplicity, we assume that each data center is just described by the computing resources they are providing. Additionally, every data center has a time-dependent offset value which describes emissions related to power supply, lights, and cooling of the data center. The appendix of this document describes these properties in more detail.

The company X sells the computing resources from cloud partners. Usually, the whole workload pipeline is implemented in a single cloud partner's service since it is difficult to move data between the partners. In this work, we assume that the company X has solved this issue. On the other hand, if we take the cloud partner's perspective, it can move the data inside its infrastructure. This means that we can drop the company X's role. The model applies and optimizes the sustainability also from cloud partner's perspective which makes it interesting for cloud partners like Amazon and Google.

Let us fix a cloud partner Y and its data centers $(d_1, \ldots, d_t)$. We assume that every data center d_k, for $k = 1, \ldots, t$, consists of machines $(m_1, \ldots, m_i)$

$$m_t = (\text{CPU}, \text{GPU}, \text{RAM}, \text{SSD}, \text{HDD}, \text{Backup}),$$

where we have made the problem a bit more complicated by dividing storage into HDD and SSD since cloud partners tend to separate these two hardware by their power consumption. The construction is relatively nested now:

$$\text{machine } m \subset \text{data center } d \subset \text{cloud partner } Y \subset \text{aggregator company } X.$$

Besides we have the offset function

$$\mathrm{e}\colon m_k \times t \to \mathbb{R}_+$$

which means that we have been accessing machine m for time t and thus we have used power supply, cooling, lights, etc. of a certain data center which adds additional cost.

Next, we assume that the customer is providing their workflow. For example, the customer wants to develop a machine learning solution, AR/VR application, develop some IoT application, Big Data, or networking. These workflows can differ dramatically and they require varying resources at varying times of the development process. Whatever the user's workflow is, we assume that the cloud partners (or the aggregator company X) are providing sufficiently internal services to run the customer's workflow. Let

$$w = (w_1, \ldots, w_n)$$

be the customer's workflow. We know that the workflow is not necessarily finite i.e. it can be continuous and running "forever" in many applications. For simplicity, we assume that the workflow is finite so that we can divide it so that the sustainability metric is minimized.

Now the problem is the following: what is the optimal way to perform the workload in the cloud partner's machines so that we can run the workload as sustainable way as possible? Making the problem more realistic, we want to add the time component into the calculations: at each time step t should we move the workload to another machine or stay in the current machine to keep the computation's carbon footprint in minimum? This optimization step does not only take into account the next step in the workflow but it optimizes over the whole workflow that is left. Thus, it might be a not-optimal single step to move the workload to another machine but in long run, it minimizes the emissions. Compare this to a reinforcement learning task where one action might be bad but it is still probable that it optimizes the total reward.

Although the problem seems to be static, that is not the case. If we take into account the availability of the machines, their characteristics, and the fact that green energy might stop flowing (in the case that there is no wind to move windmills or there is no water in rivers to create hydroelectricity), the problem of minimizing the sustainability becomes harder and non-trivial. For example, if a data center x cannot get wind power and it must switch to coal power for a longer period, then the algorithm starts suggesting data centers and available machines where the emissions get minimized.

Solution in theory

In this section, we provide the theoretical background that we use to model and solve the sustainability problem. The problem is formulated as a special kind of shortest path finding problem.

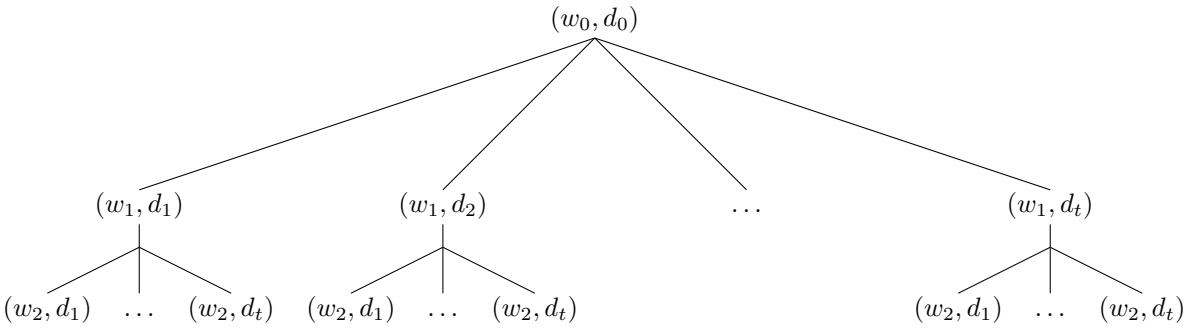
Quadratic Unconstrained Binary Optimization model

We choose that the binary variables of the QUBO model are the pairs

$$x_{i,j} = (w_i, d_j),$$

where w_i is a single work in the user's workload and d_j is a data center. If $x_{i,j} = 1$, then the workload w_i should be executed on the data center d_j. We could also refine this at the machine level. This means that we change data center variables to describe machines m_j. We believe that currently, it is more demonstrative to express the algorithm on data center level than machine level.

The idea of the solution is simple and could be solved by classical computer easily by finding the shortest path in a certain graph. Let $w = (w_1, \ldots, w_n)$ be the workload that the user has provided to the aggregator company X. The problem can be visualized as a tree



The solution corresponds to the shortest path of the form

$$(w_0, d_0) \to (w_1, d_i) \to \ldots \to (w_n, d_k)$$

from the root to one of the leaves where the weight of the edges is defined by the sustainability metric. At every level in the tree, we have all the possible data centers. The number of nodes in this tree is simply $n \times t$ where n is the length of the workload and t is the number of data centers. Next, we describe two simple constraints that encode the problem as a QUBO.

Constraint 1

For every work w_i we have exactly one variable $x_{i,j} = (w_i, d_j) = 1$. The QUBO expression for this is simply

$$\sum_{j=1}^{t} x_{i,j} = 1$$

for every $i = 1, \ldots, n$.

Constraint 2

For every pair of variables $x_{i,j} = (w_i, d_j)$ and $x_{i+1,k} = (w_{i+1}, d_k)$ we associate the (estimated emission) coefficient $e(x_{i,j}, x_{i+1,k})$ consists of two components

$$\mathrm{e}(x_{i,j}, x_{i+1,k}) = \mathrm{q}(w_{i+1}, d_k) + \mathrm{p}(x_{i,j}, x_{i+1,k}),$$

where $q(w_{i+1}, d_k)$ describes the cost of executing the work w_{i+1} in the data center d_k and $p(x_{i,j}, x_{i+1,k})$ encodes the cost of moving the work w_i to the data center d_k.

The QUBO expression for this is

$$\sum_{i=0}^{n-1} \sum_{j,k=1}^{t} \mathrm{e}(x_{i,j}, x_{i+1,k}) x_{i,j} x_{i+1,k}.$$

Solution in practice

We implemented the solution with multiple different quantum computing software and hardware:

- Ocean software program running on D-wave's and Amazon Braket's quantum annealers,
- Qiskit program running on Braket QPUs,
- Qiskit program running on Braket Simulators.

We targeted implementations to these machines and these quantum computing frameworks because it would enable our participation in Amazon Braket Challenge, IBM Qiskit Challenge, and Hybrid Algorithms Challenge.

Please see the attached Jupyter notebooks for different implementations.

Demonstrative data set

The data set consists of two types of data:

1. Cloud partner data consist of list of cloud partners and their data centers. Each data center handles workloads varying way with different emission rates simulating the fact that different machines consume energy differently.
2. User workload data consist of list of works which match with the works that data centers implement. Each work has also emission factor which describes how intense the work is for environment.

Based on the data above, we try to simulate how the emissions would develop while the user's workload is processed. Of course, this simulation is far from truth. One option to make the simulation more accurate would be to collect data from Google's Carbon Footprint tool but this takes weeks or months of active cloud computing before the tool starts to produce data.

Results

The quantum computing algorithms are compared to the classical algorithm (Dijkstra's algorithm on weighted graph). The chapters relate to the chapters in the Jupyter notebooks.

Chapter 1: Quantum annealers

TODO collect results for the final submission. This part of the code is working now and it just requires some computations.

Cahpter 2: Universal gate-model QPUs

TODO collect results for the final submission. This part of the code is under development.

Problems

- The current model does not take into account all the characteristics of the machines.
- Efficient migration between the data centers and providers might be very difficult. This migration process also produces emissions that should be part of the model.
- Cloud partners are not providing very detailed information from their data centers' emissions. It requires quite a lot of investment in monitoring and other additional equipment.

Conclusion and future work

- The results of performance of quantum computing were very positive. In this case the performance comparison is clear since we have the optimal classical algorithm running along with the quantum algorithms.
- The plan is to conclude the results of this project and develop a demonstration system for a relevant database conference. The database community has a lot of possibilities in quantum computing.
- We should pay more attention on the carbon footprint of cloud infrastructures. This work proposes an optimization algorithm to reduce the carbon footprint assuming that sufficiently emission data is provided.

TODO more for the final submission

Thank you for organizing QHack!

Appendix: Current state of cloud partners' sustainability plans and implementations

We wrote this part of the work for the possible paper which reviewers current approaches in the industry.

To understand better the dimensions of sustainability in data infrastructure, we studied how the biggest cloud service providers are claiming to implement their solutions to reduce carbon footprint.

Google

The company that has the most ambitious plans, at least on paper, regarding sustainability and a carbon-free future is Google. They claim they have been carbon neutral since 2007 and their goal is to compensate not only the current carbon footprint but also the carbon footprint that they have created since the company was founded. Their goal is to be carbon-free by 2030 [3, 4, 5]. Part of their plans is related to producing environmentally friendly products and creating sustainable working places.

In the perspective of this hackathon, Google offers an interesting tool that other cloud service providers are not (yet) offering. They have developed the Carbon Footprint tool [6] which enables the cloud users to track the carbon footprint of their cloud service usage. We registered to Google cloud but unfortunately, it might take up to 21 days before the tool starts producing data. Otherwise, it would have been interesting to integrate into the solution since it seems to be the only service providing real-time data (energy impact on an hourly basis) of carbon emissions related to Google's data centers. Google also points out that the source for electricity can vary over a day making differences to emissions depending on what time the workload is executed in the cloud.

Google's Carbon Footprint tool

Google's Carbon Footprint tool might be exactly what other cloud service providers, Aiven among them, would like to develop and implement as part of their services. The tool is still under development and evaluation. Google claims that the tool includes gross carbon emissions data in reports and disclosures, it can visualize carbon insights via dashboards and charts and reduce the gross emissions of cloud applications and infrastructure.

The methodology of Carbon Footprint tool is described in [7]. The calculations are open for reviewing and they are based on Greenhouse Gas Protocol [8] which provide detailed guidance for emission reports. Especially Google mentions that third-parties have evaluated that their Carbon Footprint's methodology is an appropriate way to calculate emissions from Google Cloud products [9]. Of course, one should be skeptical against this kind of "third-party evaluations" and their true independence from Google.

It is interesting to point out that the review [9] mentions that the following emissions are excluded from the data of Carbon Footprint:

- Upstream/downstream lifecycle emissions of data center equipment and buildings
- Electricity generation that is subsequently lost during transmission and distribution
- Diesel combustion for onsite backup power; fugitive emissions from refrigerants
- Emissions arising from small equipment deployments at internet service providers
- Google networking equipment deployed outside data centers

These points show that data center emissions are a sum of very versatile sources of emissions and being very precise seems to be a very hard challenge. Anyway, the previous bullets most probably do not produce a significant amount of the total emissions.

Carbon Footprint tool utilizes Greenhouse Gas Protocol's [8] location-based reporting method for electricity. The number does not take into account sources for carbon-free electricity. The method calculates from bottom-to-top and it relies on Google's own monitoring in its data center. This is one of the biggest challenges to solve the challenge of this hackathon: it is difficult to get detailed data of emissions produced by the data centers. We will discuss this more later.

Calculation methodology of Google's Carbon Footprint

In the following, we describe how Google calculates its data centers' carbon footprint.

Google separates energy usage into two categories: dynamic power and idle power. Dynamic power usage is calculated when a workload is running on the machines and idle power is calculated when the machines are idle. Google describes the allocation well in [7]:

> Each machine's hourly dynamic power is allocated to the internal services it supported that hour, based on relative internal service CPU usage. Machine idle power is allocated to each internal service based on its resource allocation (CPU, RAM, SDD, HDD) in the data center.

The other energy usage, such as lights, cooling, power systems, and energy for other ancillary needs is divided between the machines in the data center. All in all, Google claims to be able to calculate the power use allocated to each internal service in each data center on an hourly basis.

The power usage is then mapped to carbon emissions by using grid carbon emission intensity data provided by Tomorrow's electricityMap [10]. As Google describes in [7], Tomorrow's electricityMap emission factors include the conventional emissions from the production of electricity as well as the emission when the power plants are constructed and emissions coming from extraction,

production, and transportation of fuels to the power plant. Besides this data, Google uses data provided by International Energy Agency [11]. We believe the data is especially given in web-page [2]. Unfortunately, both Tomorrow's API and IEA's data are not publicly available without subscriptions.

The actual calculation is simple: hourly energy usage for each internal service is multiplied by the appropriate carbon emission intensity factor for that hour. Here we should also take the location into account. This determines the internal service's gross carbon footprint per hour and location as Google describes in [7]. The calculation is described in detail in https://cloud.google.com/carbon-footprint/docs/methodologydetails-ghg-emissions.

After describing the calculation of Carbon Footprint, Google goes through how the footprint is associated with the customers. Since different companies manage customers and products in their ways, we did not think this part would be relevant anymore. The description is related to the problem of how to divide the carbon footprint along with the works that have been allocated to the internal services. We think this problem is not relevant since we are focusing on the problem in the case we have a single customer at a time. Managing a larger customer group is a different problem than calculating the carbon footprint for a special workflow of a fixed customer.

Amazon

Amazon is not offering a similar Carbon Footprint tool like Google. Anyway, they describe their actions against climate change on their websites and in a white paper [12]. On the other hand, it seems that the user does not have very many opportunities to track their carbon footprint although Amazon provides a series of blog posts discussing sustainability matters of their cloud infrastructure [13]. They propose that users use Amazon CloudWatch. Amazon dedicated a special magazine issue for sustainability [14] which also contains a link to a video on how to estimate the carbon footprint of cloud computing [15]. The video is highly relevant to the project. Amazon measures the carbon footprint of their wholly complex business infrastructure not concentrating only on their cloud computing infrastructure and data centers. Thus they use a different kind of metrics than Google. So far, we believe that the white paper [12] is the most reliable source to estimate emission.

An interesting detail is that Amazon has Amazon Wind Farm Finland - Puskakorpi in Ostrobothnia which produces 53 MW of energy.

TODO

Microsoft, Digital Ocean, Upcloud, Oracle, and MongoDB.

[2] https://www.iea.org/data-and-statistics/data-product/greenhouse-gas-emissions-from-energy#allocation-of-emissions-from-electricity-and-heat

References

[1] Md Abu Bakar Siddik, Arman Shehabi, and Landon Marston. The environmental footprint of data centers in the united states. *Environmental Research Letters*, 16(6):064017, may 2021.

[2] Charles H. Bennett. Notes on landauer's principle, reversible computation and maxwell's demon, 2003.

[3] Google. Sustainability. https://sustainability.google/, November 2021.

[4] Google. Environmental report. https://www.gstatic.com/gumdrop/sustainability/google-2020-environmental-report.pdf, 2020.

[5] Google. Reduce your google cloud carbon footprint. https://cloud.google.com/architecture/reduce-carbon-footprint, 2021.

[6] Google. Carbon footprint. https://cloud.google.com/carbon-footprint, November 2021.

[7] Google. Carbon footprint reporting methodology. https://cloud.google.com/carbon-footprint/docs/methodology, 2021.

[8] Greenhouse Gas Protocol. https://ghgprotocol.org/, 2021.

[9] 3Degrees Group Inc. Carbon footprint's methodology review statement. https://services.google.com/fh/files/misc/3degrees_cloud_services_review_statement_final.pdf, 2021.

[10] Tomorrow. electricitymap. https://app.electricitymap.org/map, 2021. https://www.tmrow.com/.

[11] International Energy Agency. Data and statistics. https://www.iea.org/data-and-statistics/data-products, 2021.

[12] Daniel Bizo. The carbon reduction opportunity of moving to amazon web services. https://sustainability.aboutamazon.com/carbon_reduction_aws.pdf, October 2019. Amazon.

[13] Katja Philipp, Aleena Yunus, Otis Antoniou, and Ceren Tahtasiz. Optimizing your aws infrastructure for sustainability, parts i, ii and iii. https://aws.amazon.com/blogs/architecture/optimizing-your-aws-infrastructure-for-sustainability-part-i-compute/, August 2021. https://aws.amazon.com/blogs/architecture/optimizing-your-aws-infrastructure-for-sustainability-part-ii-storage/, https://aws.amazon.com/blogs/architecture/optimizing-your-aws-infrastructure-for-sustainability-part-iii-networking/.

[14] Amazon. Aws architecture monthly. https://d1.awsstatic.com/whitepapers/architecture-monthly/AWS-Architecture-Monthly-August-2021.pdf, August 2021.

[15] Amazon. Aws re:invent 2020: Measuring for change: Carbon footprinting at amazon scale. https://www.youtube.com/watch?v=LS_klTsM22g, February 2021.