Environmental Footprint Aware Data Center Load Balancing

We are experiencing a new wave of technology from regular web search to programming, from socializing to shopping, from audio generation to artwork. There is hardly any area that has not been combined with AI/ML for a better experience. However, this rapid growth of AI/ML-based systems as well as with existing internet-based services overwhelmed the computation resources or the the data centers. They have been flooded with numerous computational tasks as a result they consume a staggering amount of energy and other resources like water, air, etc. Further, the demand for more data centers as computational resources are increasing which forces us to build more data centers. In this project, we analyze the cost related to the computations on the data centers. A data center has two types of cost one involves construction and design other cost is for operating the data center. This project focuses on the operational cost that has been presented with experimental analysis. The operational cost mainly refers to how much energy has been used to perform the computation. However, energy consumption causes carbon emissions except for cases where energy is produced from renewable sources. Besides, renewable sources are very limited and can not be fed to all data centers. Therefore, the data centers often have to rely on energy generation by burning fuels that emit carbon. Moreover, the data centers get extremely hot while performing computations. Therefore, it requires them to keep cooler. Often the natural waters and airs are used to cool them. In this project, the environmental footprint is analyzed by inspecting the data center's carbon and water consumption and proposing an equitable means of reducing the environmental footprint by distributing the workloads to the data centers optimally.

Additional Key Words and Phrases: Predictive Control, Optimization, Footprint

ACM Reference Format:
. 2023. Environmental Footprint Aware Data Center Load Balancing . 1, 1 (December 2023), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Environment-friendly data center design has gained a lot of attention after realizing that data centers are causing serious damage to scarce environmental resources like water and causing significant greenhouse gas emissions. Though the carbon emission by data centers has been one of the well-known invested research, water is an under-invested one. However, water also needs proper attention otherwise very soon significant percentage of people will be without potable water. There are lot of works addressing carbon emission such as [2],[1],[3],[3] also there are works that also address the water footprint of the data centers such as [5],[4]. Also, there is new research in [6] that addressed both of these environmental aspects. All these works pointed out the importance of environment-aware design for sustainability and long-term stability. [6] points out the staggering amount of water usage by the data centers such as Microsoft's water consumption spiked 34% from 2021 to 2022 which is roughly equal to 1.7 billion gallons or more than 2,500 Olympic-sized swimming pools. Also, Google has reported a 20% growth in water consumption. They have discovered also that for every 20-50 questions in Chat GPT hosted in IWOA, USA data center drinks 500ml water. Furthermore, it has been discovered that 2-4% of global greenhouse gas emissions are by Data Centers. And it is been speculated that the global adaptation of LLM-like models would further exacerbate the situation. It is approximated that the carbon emission can blow up to 23%. It

Author's address:

XXXX-XXXX/2023/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

is safe to say that both carbon and water consumption would rise 4-5 times with LLM-like models. The direness of the issue motivated us to look into the problem more deeply. In this project, firstly, an optimization problem considering the carbon and water has been proposed. Later, considering the online nature of the problem where the context (how much workloads are to be distributed among the data centers, what the prices of the energy, what is the carbon efficiency and water efficiency) is revealed sequentially, a different predictive approach is proposed to deal with the information gap due to sequential reveal of information. The predictive model is an ML model that infers the future context based on the current context.

The entire report has been organized as follows, in section 2 the model components will be discussed. This will be followed by section 3 where the optimization will be presented formally and lastly, section 4 will be presenting the experimental setup and the results.

2 MODEL

We consider the industry scale data centers across the globe and propose an algorithm that performs operational cost-friendly load balancing taking not only the energy into account but also ensuring fair water usage, and reducing carbon emission.

Data Centers Here, N Geographically distributed data centers are considered each having a max capacity of M_i for $i = \{1 \cdots N\}$. Each of these data centers is responsible for processing incoming requests in the form of computations from users via workload-balancing J gateways. These requests can be thought of as relating some AI inferences using huge machine learning models(Such as chatGPT). Each of these inferences incurs direct power cost and indirect environmental impacts such as water footprint and carbon footprint. In the current context, considering the rapid inclusion of AI inferences in every sector, the continuous increase in energy consumption by data centers will cause an increase in the environmental footprint. Here we define the cost associated with direct energy consumption and the cost associated with water and carbon.

Workload The computational task in a data center can be divided into two categories one is delay sensitive and the other is delay insensitive. Usually, the task that requires immediate response is delay-sensitive. On the other hand tasks like ML model training do not require immediate response, these types of computation or tasks can be considered delay insensitive. In this project, the main focus is the delay-insensitive workloads. Distribute such workloads among the data centers such that the overall energy cost, water footprint, and carbon footprint are minimized. The workload is represented by $\lambda(t)$ at time t from all the J gateways.

Electricity Cost On a more granular level, each data center is equipped with servers to process the requests (Computations). The processing incurs the consumption of energy (electricity). Depending on the locations of the data centers, the varying availability of different fuels contributes differently to the overall cost. Moreover, the presence of different renewable energy affects the price of the energy for a data center. Usually, the energy required to process the incoming request can be modeled using the following equation which is used already in many previous works such as [5].

$$e_i(y_i(t)) = \gamma_i(t) \cdot M_i(t) \left[e_{i,s} + e_{i,d} \cdot \frac{y_i(t)}{M_i(t)} \right] \tag{1}$$

In Eqn.(1), $y_i(t)$ is the total resource used to process the incoming request at time t, $e_{i,s}$ is the static power required to keep the servers on, $e_{i,d}$ is the dynamic power for the data center i and $\frac{y_i(t)}{M_i(t)}$ is the server utilization. Using Eqn.(1) the computed energy per data center is used to get the overall monetary cost for the energy. If for datacenter i the electricity price is $p_i(t)$ at time t, the total cost would be following:

$$g_i(y_i(t)) = p_i(t) \cdot e_i(y_i(t)) \tag{2}$$

Water Footprint The water footprint depends on the total energy consumption per data center. The water footprint can be calculated using the following equation from [5], where the direct and indirect water usage efficiency factors are considered per location.

$$w_i(y_i(t)) = \left[\frac{\epsilon_{i,D}(t)}{\gamma_i(t)} + \epsilon_{i,I}(t)\right] \cdot e_i(y_i(t)) \tag{3}$$

In Eqn.(3) $\epsilon_{i,D}$ is the direct water usage efficiency (WUF) and $\epsilon_{i,I}$ is the electricity water intensity factor (EWIF).
Carbon Footprint To compute the carbon footprint we use the carbon usage efficiency Γ_i per location. The following equation is used to get the carbon footprint $c_i(y_i(t))$

$$c_i(y_i(t)) = \Gamma_i(t) \cdot e_i(y_i(t)) \tag{4}$$

2.1 Problem Formulation

All the entities defined above combined to formalize the environmental footprint-aware load balancing. Assume at each time, $t = 0, 1 \cdots T$ the goal is to assign the incoming workload $\lambda(t)$ at time t among N data centers such that it minimizes the overall operational cost $g(y(t))$, minimizes the maximum water footprint $w_i(y(t))$ of a datacenter i, among all data centers and minimizes the maximum carbon footprint $c_i(y(t))$ of a data center i among all data centers. Here, we are considering $J = 1$, which means all the data centers are connected via a single gateway. It is kept intentionally like this for simplicity. However, it is easily scalable for multiple gateways as well. In the following, we formally combine the operational cost term along with the water and carbon efficiency term to solve a linear optimization problem to get optimal load balancing considering delay-tolerant batch job processing.

$$\min_{y} \sum_{t=1}^{T} g(y(t)) + \kappa_w \left(\max_i \sum_{t=1}^{T} w_i(y_i(t)) \right) + \kappa_c \left(\max_i \sum_{t=1}^{T} c_i(y_i(t)) \right) \tag{5a}$$

$$s.t., \quad y(t) \leq \lambda(t) + \delta_{(t-1)}, \forall t \tag{5b}$$

$$\sum_{t=1}^{T} \sum_{i=1}^{N} y_i(t) = \sum_{t=1}^{T} \lambda(t), \tag{5c}$$

$$y_i(t) \leq M_i \quad \forall i, t \tag{5d}$$

In Eqn.(5a),κ_w and κ_c are the positive constants denoting the importance of water and carbon footprint in the minimization term which is technically transforming the water and carbon footprint term into a monetary value. Constraint in Eqn.(5c), allows the processing of incoming requests in such a way that the overall cost is minimized by distributing the requests to the most economic data centers (the data centers for which the energy costs are reasonably low) at time t. The optimization also ensures that no data center has a workload more than the capacity through constraint in Eqn.(5d). In Eqn (5b),$\delta_{(t-1)}$ denotes the remaining workload from the previous time step $t - 1$. Mainly, it says the data center can't allocate workloads more than what it has been received.

3 MODEL PREDICTIVE CONTROL (MPC)

The optimization presented above in Eqn(5a) assumes the complete knowledge of the system for the entire time horizon which is not feasible and practical. As the context information becomes available sequentially. Therefore, at time t it won't be able to utilize the future information which would result in a non-optimal solution. To solve the problem a different method is adopted which is known as Model predictive control. Through MPC a K length

prediction horizon is utilized to get the approximate distribution of the incoming workloads, energy price, water and carbon efficiency such that it minimizes the total energy cost, water footprint, and carbon footprint. In MPC a predictive model continuously predicts the future K contextual information using the current input at time t. Using the predicted information the system solves an equivalent optimization to return the best distribution policy for time t In the following, the objective function along with the constraints are presented.

$$\min_{y(t),y(t+1)\cdots y(t+K)} \Big[\Big(\sum_{k=0}^{K} g(y(t+k)) + \sum_{\tau=1}^{t-1} g(y(\tau))\Big) + \kappa_w \max_i \Big(\sum_{k=0}^{K} w_i(y_i(t+k)) + \sum_{\tau=1}^{t-1} w_i(y_i(\tau))\Big) + \kappa_c \max_i \Big(\sum_{k=0}^{K} c_i(y_i(t+k)) + \sum_{\tau=1}^{t-1} c_i(y_i(\tau))\Big)\Big] \tag{6a}$$

$$s.t., \quad \sum_{i=1}^{N} y_i(t) \le \lambda(t) + \delta_{(t-1)}, \quad \forall t \tag{6b}$$

$$\sum_{k=0}^{K}\sum_{i=1}^{N} y_i(t+k) = \sum_{k=0}^{K} \lambda(t+k) + \delta_{(t-1)} \tag{6c}$$

$$y_i(t) \le M_i \quad \forall i, t \tag{6d}$$

In (6a), we consider the historical cost into account such that it does not overwhelm the region that is already stressed due to a heavy load of energy, water, or carbon. The constraint in (6b) ensures that the distribution should not be more than what is available at some time $t + k$. Further, the next constraint in (6c) ensures that the total distributed workloads among the data N centers should be equal to the total workload over the current time and the time window of length K. And the constraint in (6d), is made sure that no data center gets workloads more than it can process. In (6b) and (6c), δ represents the remaining workload from the previous iteration at time t.

4 EXPERIMENT

We show the performance of the proposed method through trace-based simulation. In the following, we present the baseline setup and the trace used for the simulation.

4.1 Setup

Data Center We use $N = 10$ geographically distributed data centers with homogeneous capacity. Our model can be easily extensible for heterogeneous capacities for different data centers. Among the 10 data centers, four are from the U.S. which are located in **Virginia, Georgia, Texas, and Nevada**, another four are from Europe located in **Belgium, the Netherlands, Germany, and Denmark**, and the other two are from Asia which are in **Singapore and Japan**.

Workload For the experiment, we considered delay-tolerant workloads (such as training a large AI model) as scheduling these types of workloads can effectively result in minimizing the overall cost. We extract the workload data from the power usage of a large language model BLOOM for 19 days. This workload is going to be distributed to each of the 10 data centers through a single gateway as mentioned earlier. These 19 days of data are again augmented by the sliding window technique to generate approximately 400 days' worth of data. Where first 13 days of data were used to generate the training data and the later 6 days of data were used for generating the testing data. Additionally, some parts of the training data are used for validation.

ML Predictor : As the ML time series predictor an LSTM model is used. The input dimension is set to 1 and the output dimension is set to 24. The output dimension refers to the predicted horizon length. It is kept at 24

because in general the energy market or related companies frequently forecast day-ahead demands and prices to make optimal decisions. Similarly, we are using length 24 denoting a day ahead context information. Also, we used mean square error as the loss function and Adam optimizer for loss minimization in the prediction.

4.2 Results

Here, we present the comparative results for two schemes one is our equitable approach that addresses both the water and carbon footprint and the other is the one that only minimizes the energy cost. For these two schemes, we present the compared results for energy cost, water footprint, and carbon footprint.

Fig 1 shows that the energy cost is comparatively smaller for the method whose sole focus is to minimize the price. On a deeper level, the MPC results are deviating from the offline results. It totally depends on how well the context is predicted.

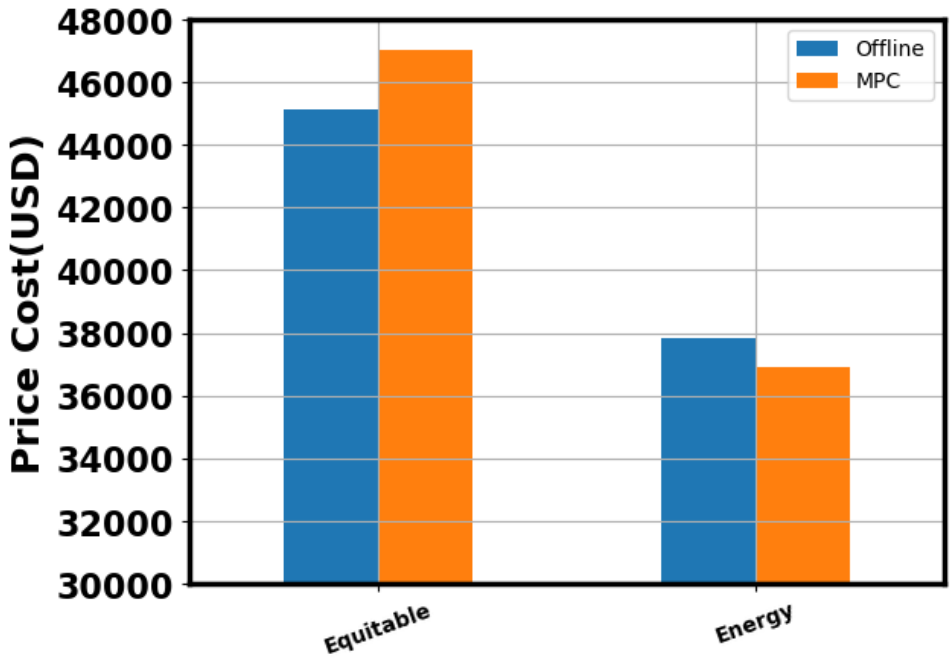

Fig. 1. Energy Cost ;

Similarly, in Fig 2, it is found that the equitable method is more water-friendly as the costs are lesser than the method that only minimizes energy. It demonstrates the importance of an equitable method of load balancing. Again, the mpc results are showing that they are not very far from the offline method.

Finally, in Fig 3, it is observed that the equitable method is more carbon efficient compared to the method that minimizes only energy.

5 CONCLUSION

In this project, we proposed an environmental footprint-aware load balancing method that minimizes the water and carbon footprint together. From the results, it is proved the successful reduction in water and carbon footprint.

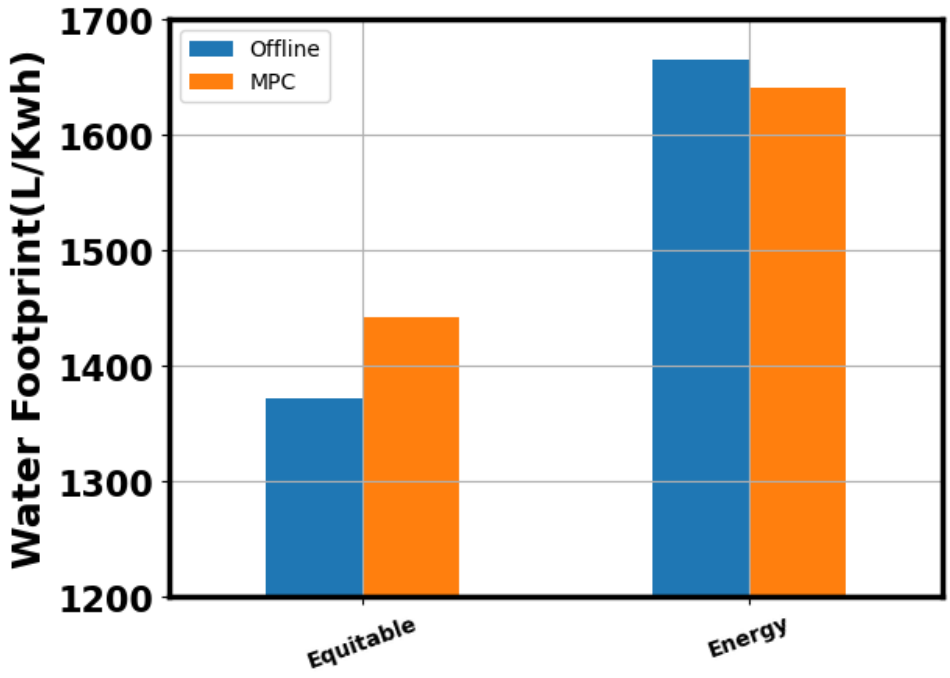

Fig. 2. Water Footprint;

However, there is still some to do the task. For instance, the accuracy of the LSTM model in predicting the context is under-analyzed. More experiments are necessary to get satisfactory results.

REFERENCES

[1] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Aditya Sundarrajan, Kiwan Maeng, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon Dependencies in Datacenter Design and Management. *ACM SIGENERGY Energy Informatics Review* 3, 3 (2023), 21–26.

[2] Mariam Elgamal, Doug Carmean, Elnaz Ansari, Okay Zed, Ramesh Peri, Srilatha Manne, Udit Gupta, Gu-Yeon Wei, David Brooks, Gage Hills, and Carole-Jean Wu. 2023. Carbon-Efficient Design Optimization for Computing Systems. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems* (Boston, MA, USA) *(HotCarbon '23)*. Association for Computing Machinery, New York, NY, USA, Article 16, 7 pages. https://doi.org/10.1145/3604930.3605712

[3] Mariam Elgamal, Doug Carmean, Elnaz Ansari, Okay Zed, Ramesh Peri, Srilatha Manne, Udit Gupta, Gu-Yeon Wei, David Brooks, Gage Hills, and Carole-Jean Wu. 2023. Design Space Exploration and Optimization for Carbon-Efficient Extended Reality Systems. arXiv:2305.01831 [cs.AR]

[4] Mohammad A. Islam, Kishwar Ahmed, Hong Xu, Nguyen H. Tran, Gang Quan, and Shaolei Ren. 2018. Exploiting Spatio-Temporal Diversity for Water Saving in Geo-Distributed Data Centers. *IEEE Transactions on Cloud Computing* 6, 3 (2018), 734–746. https://doi.org/10.1109/TCC.2016.2535201

[5] Mohammad A. Islam, Shaolei Ren, Gang Quan, Muhammad Z. Shakir, and Athanasios V. Vasilakos. 2017. Water-Constrained Geographic Load Balancing in Data Centers. *IEEE Transactions on Cloud Computing* 5, 2 (2017), 208–220. https://doi.org/10.1109/TCC.2015.2453982

[6] Pengfei Li, Jianyi Yang, Mohammad A. Islam, and Shaolei Ren. 2023. Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models. arXiv:2304.03271 [cs.LG]

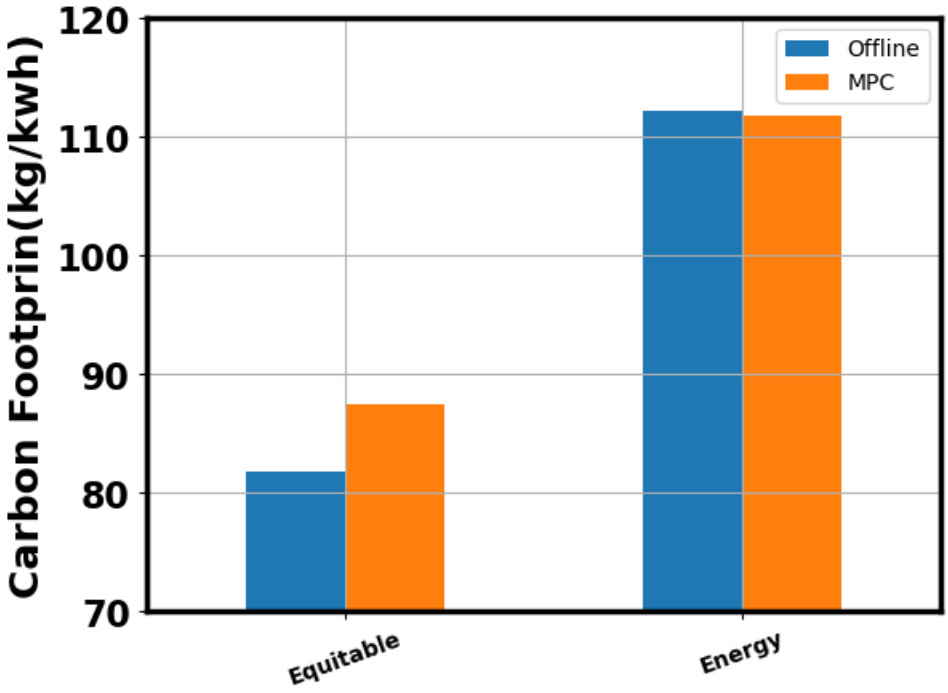

Fig. 3. Carbon Footprint;