This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Berezovskaya, Yulia; Yang, Chen Wei; Vyatkin, Valeriy

Towards reinforcement learning approach to energy-efficient control of server fans in data centres

Published in:
Proceedings - 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2021

DOI:
[10.1109/ETFA45728.2021.9613639](https://doi.org/10.1109/ETFA45728.2021.9613639)

Published: 30/11/2021

Document Version
Peer reviewed version

Please cite the original version:
Berezovskaya, Y., Yang, C. W., & Vyatkin, V. (2021). Towards reinforcement learning approach to energy-efficient control of server fans in data centres. In *Proceedings - 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2021* IEEE.
<https://doi.org/10.1109/ETFA45728.2021.9613639>

© 2021 IEEE. This is the author's version of an article that has been published by IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Towards reinforcement learning approach to energy-efficient control of server fans in data centres

Yulia Berezovskaya*, Chen-Wei Yang*, Valeriy Vyatkin*[†]

* Luleå University of Technology, Luleå, Sweden, Email: (yulia.berezovskaya, chen-wei.yang)@ltu.se

[†] Aalto University, Helsinki, Finland, Email: vyatkin@ieee.org

Abstract—Modern data centres require control, which aims to improve their energy efficiency and maintain their high availability. This work considers the implementation of a server fan agent, which is intended to minimise the power consumption of the corresponding server fan or group of fans. In the paper, the reinforcement learning approach to energy-efficient control of server fans is suggested. The reinforcement learning workflow is considered. The Simulink blocks simplifying the building of the environment for the reinforcement learning agent are developed. This work provides the framework for creating and training reinforcement learning agents of different types. As the paper is only a work-in-progress, possible type of agents and their training process is described, but training and deploying the agent is a work for the future.

Index Terms—data centre, energy-efficient control, multi-agent control, reinforcement learning

I. INTRODUCTION

Any data centre is a complex cyber-physical system, which combines equipment, networking and computation. Data centres are expected to be highly available and reliable and at the same time be energy-efficient and environmentally friendly. In the majority of modern data centres, high availability and reliability are achieved with redundant energy consumption [1]. Therefore, energy-efficient control, which adapts the energy use in data centres based on resource requirements at run-time, is required. The control concentrates on the computational and cooling systems since they are the main energy consumers in any data centre [2].

As distributed control for data centres, we suggest a multi-agent system. Such a system comprises adaptable, autonomous agents with their own objectives. Besides, the agents can interact with each other as human beings do; that is, they can cooperate, coordinate and negotiate [3]. In our previous work [4], we presented the design of a multi-agent system that aims to provide energy-efficient control in data centres as distributed systems of interdependent components.

In the current work, we start developing the multi-agent system based on the architecture suggested in [4]. To avoid scattering, we concentrate on implementing the component of the multi-agent control, which is responsible for energy-efficient cooling of the CPU (Central Processing Unit). This work deals with a server fan agent, which is intended to minimise the power consumption of the corresponding server fan or group of fans. At each time point, the agent is expected to adjust the rotation speed of the server fans to the actual temperature of the corresponding CPUs. Our suggestion is

in applying reinforcement learning to the agent's decision-making process. As the agent should communicate with a dynamic environment and make its decision in constantly changing conditions [5], reinforcement learning is a promising approach.

The current work aims to investigate the reinforcement learning approach to energy-efficient control of server fans. The goal of the paper is to consider the reinforcement learning workflow, which consists of determining the environment as well as setting up and training the agent. This work considers the CPU and corresponding server fans as the environment for a reinforcement learning agent. The set of observations that fully describes the agent's environment and the value of the reward that the environment returns to the agent as a response to its action is determined. This work also considers the anatomy of the agent and different types of reinforcement learning algorithms.

Another goal of the work is in extending a modular Simulink toolbox described in our previous work [6]. This toolbox contains blocks modelling the main components of data centres and enables constructing the data centres of any configuration from the blocks. The extension includes created Simulink blocks such as reward blocks and observation blocks. These blocks allow the construction of the environment for the reinforcement learning agent.

The extension also provides scripts that set up the environment, establish an inner organisation of the agent, determine the reinforcement learning algorithm utilised by the agent, and finally run the learning process. The extension deals only with types of reinforcement learning agents that are provided by the Reinforcement Learning Toolbox in Matlab/Simulink. Thus, this work provides the framework for creating and training the reinforcement learning agents of different types. As the paper is only a work-in-progress, possible type of agents and their training process is described, but training and deploying the agent is a work for the future.

The rest of this paper is organized as follows. Section II provides related works. Section III considers the reinforcement learning workflow in general. Section IV deals with the implementation of reinforcement learning workflow concerning server fan agents. Section V concludes the paper with ideas for future work.

II. RELATED WORKS

This section considers works, which concentrate on approaches to energy-efficient control of server fans in data centres.

In [7], authors propose a multi-input multi-output (MIMO) fan controller, which tunes the speeds of individual fans proactively based on the prediction of the server temperatures. In [8], authors use the digital twin to tune the Model Predictive Control (MPC) of server fans. Many research works concentrate on the development and tuning of Proportional Integral Derivative (PID) controllers, such works as [9], [10], [11]. Another approach is in applying the fuzzy logic to design server fan controllers, for example, such ideas are considered in [12] and [13]. All aforementioned approaches demonstrate good results, however, we suggest reinforcement learning as one more control method. The idea for the future is to implement a reinforcement learning agent and comparing its performance with PID controller and controller based on fuzzy logic.

Reinforcement learning assumes interaction between an agent and its environment to achieve a maximum reward. As we deal with the server fan agent, we consider the CPU and corresponding server fans as the agent's environment. On the majority of modern servers, each CPU is cooled by several server fans. Thus, we can consider two directions for the agents' development. In the first one, the single-agent controls all the server fans corresponding to the given CPU by setting the same rotation speed to them. In this case, ordinary reinforcement learning is suggested as a decision-making approach. In the second direction, each server fan is controlled with its own agent and can be set its own rotation speed. In this case, multi-agent reinforcement learning is a promising approach.

III. REINFORCEMENT LEARNING

This section describes the basic ideas of reinforcement learning that we rely on in developing the energy-efficient control of server fans.

In the reinforcement learning framework, the decision-maker is called the agent. Surroundings comprising everything outside the agent is called the environment. The agent interacts with the environment all the time:

- 1) the agent gets information about the environment state called observations (O_t);
- 2) based on observations the agent decides the action to take (A_t);
- 3) after the action is applied to the environment, the agent receives observations of its changed state (O_{t+1}) and reward value produced by the environment (R_{t+1});
- 4) based on new observations and reward value, the agent decides the next action.

The process continues throughout the lifetime of the agent-environment system [5]. The interaction can be represented as sequence:

$$O_0, A_0, O_1, R_1, A_1, O_2, R_2, A_2, \dots$$

Fig.1 demonstrates the described agent-environment interaction.

Figure 1. The agent-environment interaction

The environment generates observations that demonstrate to the agent how the inner state of the environment is changed in response to the action from the agent. The environment also generates the reward for each state. The reward is just a quantitative objectification of the agent's purpose at each time step. The immediate reward is an important descriptor of the state, but it is even more important is the total reward over time. Such an expected total reward is called a value. The value is also an important concept of reinforcement learning. In terms of the value, the agent thinks for the long term rather than a short-term benefit.

According to [5], the value can be calculated with

$$Q(O_t, A_t) = R(O_t) + \sum_{i=1}^T \gamma^i \cdot R(O_{t+i}) \quad (1)$$

In (1),

- $R(O)$ is the reward function of the environment state represented as observation O .
- γ is a discount rate ($0 \leq \gamma \leq 1$) determines the present value of future rewards: the reward received i step in the future is reduced by γ_{i-1} times. If $\gamma = 0$, the agent maximises only immediate rewards; if $\gamma = 1$, future rewards are worth as they are received immediately.
- T is a number of time steps ($T \leq \infty$). If $T = \infty$ the infinite sum in (1) has a finite value, if and only if $\gamma < 1$.

The agent in the reinforcement learning paradigm always aims to select the action that produces the most reward in the long term. The agent involves two main components: a policy and a reinforcement learning algorithm. The policy is a function that maps a set of observations to a set of actions. It can be represented as a function with tunable parameters. The goal of the learning algorithm is to adjust the policy parameters so that the most optimal action is taken. The algorithm takes into account the action, the observations from the environment, and the amount of reward collected.

There are three groups of reinforcement learning algorithms: policy-based algorithms, value-based methods and actor-critic algorithms [14].

From all aforementioned, reinforcement learning workflow addresses two main components: the environment and the agent. Therefore, to develop a controller based on reinforcement learning, the environment is set up and the agent is implemented. The workflow consists of follows:

- setting up the environment that includes defining which observations the environment can provide as well as how the reward can be calculated for each environment state;
- establishing the agent that means defining the policy and selecting the reinforcement learning algorithm;
- training the agent in interaction with the environment;
- deploying the agent into the system: agent-environment.

The next section considers the implementation of the reinforcement learning workflow to energy-efficient control of server fans. Two first steps namely setting up the environment and establishing the agent are implemented.

IV. IMPLEMENTATION

To energy-efficient control of server fans, this work suggests developing the server fan agent. The agent takes current values of CPU temperature and server fans power consumption and decides fans rotation speed using the reinforcement learning approach. The agent also considers CPU utilisation and temperature at the server inlet.

Since the agent learns through interaction with the environment, it is necessary to provide the environment or its substitute. This work uses a simulation of the environment as its substitute. The simulation allows avoiding the usage of real hardware and the risk of damage it. At the same time, the simulation can run faster than in real-time or be parallelised, speeding up a slow learning process. Some situations that need to be tested are difficult, dangerous or expensive to perform on real equipment; thus, it is easier to simulate them. The simulation should demonstrate behaviour close to the real one.

Figure 2. CPU-Server Fans simulation

Fig.2 shows the model of the CPU and corresponding server fans that substitutes the environment. The model is built using the Simulink toolbox presented in [6]. Fig.3 demonstrates that the modelled behaviour, which is an evolution of CPU temperature, is quite realistic.

Figure 3. Comparing the modelling results and real data: CPU temperature

To complete setting up the environment, observations the agent receives and the reward for each possible state are required.

Observations determines the state of the environment at each time point:

- T_{CPU} is CPU temperature ($^{\circ}C$);
- $e = T_{ref} - T_{CPU}$ is difference between current CPU temperature and reference temperature ($^{\circ}C$);
- P_{SF} is power consumption of the server fans (W);
- $Util$ is CPU utilisation (%);
- T_{in} is temperature at CPU inlet ($^{\circ}C$).

The selection of the reward function is a challenging issue. This work suggests calculating the reward by (2).

$$R = c_1 \cdot (e \leq 0) + c_2 \cdot (e_{cr} - |e|) + c_3 \cdot P_{SF} \quad (2)$$

In (2),

- c_1 penalizes for exceeding reference temperature, $c_1 < 0$;
- c_2 rewards for closeness to reference temperature and penalizes for moving away from it, $c_2 > 0$;
- c_3 penalizes for increase of server fans power consumption, $c_3 < 0$.

The agent makes a decision on which server fans rotation speed should be set. As a first step, we consider the simplest set of possible actions:

- **-1**: reduces current rotation speed by a tenth of the maximum permissible rotation speed of the server fan, that is, $RPM = RPM - \frac{RPM_{max}}{10}$;
- **0**: keeps the same value of the rotation speed, that is, $RPM = RPM$;
- **+1**: increases current rotation speed by a tenth of the maximum permissible rotation speed of the server fan, that is, $RPM = RPM + \frac{RPM_{max}}{10}$.

To implement the agent, we use RL Agent block from Reinforcement Learning Toolbox in Matlab/Simulink [15]. The block is configured as Q-value agent that uses value-based algorithm and connected to environment, as shown on Fig.4.

Figure 4. Agent-environment connection

After all preparations, it is possible to train the agent. The training process is shown on Fig.5.

Figure 5. Agent training process

In Fig.5, the Episode Manager plot shows the reward for each episode (blue line) and a running average reward value (red line). Also, because the agent has the critic, the plot shows the critic's estimate of the discounted long-term reward at the start of each episode (yellow line). In a normal case, the blue line converges to the yellow one, but in our case, the training process is failed. This is the future work, the agent is trained to provide energy-efficient control of server fans.

V. CONCLUSION

In the paper, we proposed the idea of applying the reinforcement learning approach to energy-efficient control of server fans. We considered main components such as the environment and the agent as well as their interaction. We extended our toolbox [6] with new blocks: the Observation block that forms the vector of observations from the separate values; the Reward block that implements the reward function. We also added scripts that construct using those blocks the environment that is appropriate to use with the RL Agent block from Reinforcement Learning Toolbox. As the first experiments, we tried to train the Q-value agent and failed. As the main result for the current time, we consider the development of the framework that allows us to construct different reward functions and train different types of agents.

For future works, we are going to investigate reward functions, which means to select a type of function, important components that should be rewarded or penalised, as well as parameters. Another way to improve our work is to investigate

different types of the agent such as policy-based agents, value-based agents and actor-critic agents and select an appropriate one. The selection should base on the most successful training.

ACKNOWLEDGMENT

This study was supported by the Swedish Energy Agency (Energimyndigheten) through the Cloudberry Datacentres project under the grant 43090-2

REFERENCES

- [1] P. Bertoldi, "A market transformation programme for improving energy efficiency in data centres," in *Proceedings of the MELS: Taming the Beast, ACEEE Summer Study on Energy Efficiency in Buildings*, 2014.
- [2] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2015.
- [3] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & sons, 2009.
- [4] Y. Berezovskaya, C.-W. Yang, and V. Vyatkin, "Towards multi-agent control in energy-efficient data centres," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2020, pp. 3574–3579.
- [5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [6] Y. Berezovskaya, C.-W. Yang, A. Mousavi, V. Vyatkin, and T. B. Minde, "Modular model of a data centre as a tool for improving its energy efficiency," *IEEE Access*, vol. 8, pp. 46 559–46 573, 2020.
- [7] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhu, and P. Ranganathan, "Optimal fan speed control for thermal management of servers," in *International Electronic Packaging Technical Conference and Exhibition*, vol. 43604, 2009, pp. 709–719.
- [8] R. Brännvall, J. Sarkinen, J. Svartholm, J. Gustafsson, and J. Summers, "Digital twin for tuning of server fan controllers," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 1425–1428.
- [9] C. Lee and R. Chen, "Optimal self-tuning pid controller based on low power consumption for a server fan cooling system," *Sensors*, vol. 15, no. 5, pp. 11 685–11 700, 2015.
- [10] Y. Fu, N. Kottenstette, Y. Chen, C. Lu, X. D. Koutsoukos, and H. Wang, "Feedback thermal control for real-time systems," in *2010 16th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 2010, pp. 111–120.
- [11] J. Kim, M. M. Sabry, D. Atienza, K. Vaidyanathan, and K. Gross, "Global fan speed control considering non-ideal temperature measurements in enterprise servers," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2014, pp. 1–6.
- [12] C. Lee, Y. Chan, and R. Chen, "Decoupling fan speed control design of server based on fuzzy logic," *2013 IEEE International Symposium on Industrial Electronics*, pp. 1–5, 2013.
- [13] J.-S. Ko, J.-H. Huh, and J.-C. Kim, "Improvement of energy efficiency and control performance of cooling system fan applied to industry 4.0 data center," *Electronics*, vol. 8, no. 5, p. 582, 2019.
- [14] Mathworks, "Reinforcement learning with matlab and simulink," 2020. [Online]. Available: <https://www.mathworks.com/content/dam/mathworks/ebook/gated/reinforcement-learning-ebook-all-chapters.pdf>
- [15] —, "Reinforcement learning toolbox," 2021. [Online]. Available: <https://www.mathworks.com/help/reinforcement-learning/index.html>