weizenbaum journal OF THE DIGITAL SOCIETY

05 ISSUE 01

WEIZENBAUM JOURNAL OF THE DIGITAL SOCIETY
Volume 5 \ Issue 1 \ w5.1.5 \ 03-10-2024
ISSN 2748-5625 \ DOI 10.34669/WI.WJDS/5.1.5

Information on this journal and its funding can be found on its website:
https://wjds.weizenbaum-institut.de

KEYWORDS

high performance computing

software

AI & climate

sustainable AI

RESEARCH PAPER

IDLEWiSE

A Conceptual Approach for AI-Assisted Energy Efficiency in HPC Clusters

Chiara Fusar Bassini [*1] \ Leonard Hackel [2] \
Thorren Kirschbaum [3] \

[1] Hertie School, Berlin
[2] Technische Universität Berlin \ Berlin Institute for the Foundations of Learning and Data (BIFOLD)
[3] Helmholtz-Zentrum Berlin \ Freie Universität Berlin
[*] Corresponding author, c.fusarbassini@hertie-school.org

ABSTRACT

The growing energy demand for high-performance computing (HPC) systems raises severe concerns about their environmental impact. Novel system paradigms and computational schemes are needed to limit energy consumption while ensuring the efficiency and availability of computing resources. In this contribution, we introduce a concept for an Intelligent Decision Tool for Lowering Energy Waste in System Efficiency (IDLEWiSE), which aims to decrease the energy consumption of HPC clusters operating below total capacity by selectively shutting down idle computational units. This paper outlines an optimization tool using efficient machine-learning algorithms like decision trees to learn optimal shutdown policies online. We further locate our approach in the context of existing energy-economizing instruments and perform a strategic analysis and stepwise validation of the proposed concept. The study also includes qualitative anonymized findings from a survey of German scientific HPC cluster administrators, corroborating the urgent need for energy-efficient tools and practices for practitioners.

1 Introduction

High-performance computing (HPC) clusters have enabled significant breakthroughs in many fields, ranging from the discovery of the Higgs Boson (Belyaev et al., 2017) to unraveling SARS-CoV-2's genomic evolution (Zvyagin et al., 2023). However, modern data and HPC centers, which are at the core of digital transformation, are increasing the global energy demand significantly (Lin et al., 2023). The unprecedented computational power made available to the broad public through large-scale HPC cluster facilities comes at the cost of further direct and indirect environmental damages. This tradeoff between scientific discoveries and sustainability concerns motivates researchers to seek more sustainable solutions in HPC computing and artificial intelligence (AI) in general. Ullrich et al. (2024) compiled an overview of such applications and highlighted the need for critical deployment of AI rather than indiscriminate "techno-solutionism." Within the broad spectrum of sustainable AI, several initiatives have been undertaken to address environmental issues (e.g., lowering power usage and improving orchestration systems; Siegmund et al., 2022) and, more recently, social issues as well (e.g., fairness mitigation in deployment; Weerts et al. 2023). In this paper, we mainly focus on demand minimization measures and only briefly touch upon other ecological issues.

In 2021, the information and communication technology (ICT) sector, which includes data and HPC centers, personal digital devices, networks, and televisions, was estimated to account for 1.8–2.8% of total global carbon emissions (Freitag et al., 2021). As embedded AI systems are rapidly rising across industries, the energy footprint of computing applications has been growing steadily and, under current market paradigms, is expected to increase exponentially (Castro, 2022; Schwartz et al., 2020). Consequently, innovative ideas and technological advances are urgently needed to transition into an era of energy-efficient HPC. This paper aims to inspire researchers and practitioners to seek innovative solutions to alleviate this problem. In particular, we conceptualize a software tool for the reduction of energy consumption of small HPC clusters, focusing specifically on architectures hosted on older hardware. Indeed, older hardware tends to benefit most from shutdown policies due to high idle energy consumption.

Today, HPC systems are mostly designed to satisfy the priorities of practitioners, which concern the amount of available computing capacity and the high availability of services. Preliminary analysis and anecdotal evidence from discussions with academic and small-scale HPC providers and administrators reveal that, especially for this subsegment, energy consumption is seen as a secondary concern or even remains unaddressed altogether. To tackle the problem of rising HPC energy demand on a larger scale, we need easy-to-implement solutions that do not sacrifice computing capacity or availability. Such solutions may come in the form of improved hardware developments that aim

to lower energy consumption, software improvements to enhance algorithm efficiency, HPC management tools that intelligently allocate computational resources, and new best practices in HPC computing that honor high-efficiency low-resource computations.

This work presents a concept for the HPC scheduling tool Intelligent Decision Tool for Lowering Energy Waste in System Efficiency (IDLEWiSE). We conceive IDLEWiSE as an HPC management software specifically designed for the needs of locally managed, small-to-medium-size HPC clusters. By deploying machine learning-supported shutdown policies, it optimizes the use of available computing resources and reduces HPC energy consumption. In particular, the tool selectively shuts down computational units that are expected to be idle for a longer period; the shutdown policy is based on the expected idle times of the units, which are computed by an efficient learnable algorithm (e.g., a decision tree). The use of learnable algorithms and the implementation of an online learning strategy ensures adaptability to the needs of individual clusters and to changes in cluster deployment over time. With our tool, we aim to address the following question:

> How can artificial intelligence help efficiently manage the energy consumption of HPC clusters, especially in the case of smaller, locally managed HPC systems with older and heterogeneous hardware?

The paper is structured as follows. Section 2 overviews related work and compares previous approaches with our proposed concept. In Section 3, we present an anonymized survey of German scientific HPC administrators that substantiates the need for projects reducing the energy consumption and improving the efficiency of scientific HPC clusters. The section also describes our approach in more detail and provides a contextual analysis. A strategy for evaluating the proposed concept is then defined in Section 4. Finally, in Sections 5 and 6, we respectively summarize the results of our analysis and discuss possible future research directions.

2 Background

2.1 Related Work

Different strategies may be used to increase the energy efficiency of HPC systems, including hardware and software improvements, HPC management and scheduling, and new best practices for HPC computing. Different heuristic rules and, more recently, machine learning approaches have been investigated to implement stable and interpretable policies.

We divide energy-efficiency measures into three types: improvements in hardware efficiency, improvements in the scheduling of clusters, and energy-aware models and computations. Whenever possible, a holistic approach should be preferred so that energy consumption can be reduced by multiple measures operating in parallel and synergy effects can be exploited. At the same time, any energy-efficiency measure should be carefully crafted to effectively curtail energy consumption while maintaining stable computational performance and on-demand server availability.

2.1.1. Improvements in Hardware Efficiency

The current evolution of hardware accelerators and accelerated systems shows significant progress in various dimensions. New manufacturing processes, such as the introduction of advanced 5-nm and 3-nm processing nodes for chip manufacturing, enable higher integration density and performance. At the same time, these processes reduce energy consumption because smaller transistors usually consume less energy while switching faster.

For specific applications, tailored hardware improvements can lead to significant gains in performance, for example, by using field-programmable gate arrays for low-locality calculations (i.e., Monte Carlo simulations [Metropolis & Ulam, 1949] for extreme weather event forecasting) or tensor cores in graphics processors for machine-learning applications (Betkaoui et al., 2010; Markidis et al., 2018). In addition, adaptive voltage, frequency controls, and advanced cooling technologies are increasingly being used to further optimize the energy consumption of modern hardware (Mishra & Khare, 2015).

Despite these successes, challenges and research gaps remain. For each application, the efficient use of existing technologies requires fine-tuning algorithms to account for the specific hardware at hand. Developing applications that can exploit the full potential of the current hardware is an ongoing challenge (Markidis et al., 2018) because both the hardware components and the software architectures must be taken into consideration to maximize performance and energy efficiency.

2.1.2. HPC Scheduling Software

Adaptive HPC resource management is instrumental to exploit the efficiency of hardware. This entails different aspects, ranging from system-level software management to the use of energy-efficient programming languages. For example, Pinheiro et al. (2001) developed an algorithm that dynamically turns cluster nodes on and off to save power consumption, relying on simple heuristics combined with simple metrics such as current cluster load. In contrast, our approach aims to use learnable algorithms that dynamically adapt to individual cluster requirements to enable more informed decisions. Furthermore, Duan et al. (2015) proposed a dynamic idle interval prediction scheme for estimating

future central processing unit (CPU) idle interval lengths and choosing the most cost-effective sleep state to minimize overall power consumption. Their approach is similar to the one we present, but it is limited to CPU resources. Still, it requires the identification, implementation, and accessibility of different sleep states for each computational unit to maximize energy efficiency, which may not be straightforward for any given HPC cluster. Raïs et al. (2016) compared different shutdown policies with respect to efficiency gains, considering shutdown and startup costs in terms of time and energy. They show that the energy saved by temporarily shutting down nodes is higher than the cost associated with shutting down and starting up the nodes. Therefore, they conclude, selective shutdown policies are beneficial for energy saving. More recently, Daid et al. (2021) introduced a machine-learning approach, deploying neural networks and linear regression to achieve efficient minimum execution and completion time scheduling while fulfilling service-level agreements in data centers. While these approaches are practical in their respective application fields, they are not suited for deployment in small and heterogeneous clusters that use different accelerators over distributed computational nodes, or they are only applicable for the scheduling of computational tasks that are already issued.

It is worth mentioning that there has been significant interest in reducing the energy consumption of energy clusters outside of academia, with major information technology companies pioneering the field. In 2016, researchers at DeepMind noticed that dynamic environments like data centers and HPC clusters seldom operate under optimal conditions because of the complex interactions between the hardware and its environment (Evans & Gao, 2016). They proposed applying machine learning to enhance the energy management of data centers. In 2018, Google reportedly switched to a human-out-of-the-loop control system, which iteratively chooses the next actions that minimize the energy consumption of data centers within a set of robust safety constraints (Gamble & Gao, 2018). Large-scale industrial applications demonstrate the stability and feasibility of AI solutions for energy efficiency. Unfortunately, there is little open-sourced information on their implementation code; it is also unclear whether the HPC centers of different companies with different hardware (e.g., smaller local HPC clusters at university and research centers) could benefit from transfer learning when deploying such pre-trained tools. With our solution, we want to address this gap in the industry and academia, by conceptualizing a low-cost and open solution that can be generically applied to different computing hardware, independent of age or specialization.

2.1.3. Software Efficiency and Paradigm Shift

The introduction of new best practices in the field of HPC computing holds great potential to reduce computing expenses and associated energy consumption in the future. In current research, the availability of more computing resources usually translates into the use of larger computational models that yield improved accuracy, albeit at the expense of huge computational efforts. This is true, for instance, for research in the fields of fluid mechanics (Kurz et al., 2022; Trebotich, 2024), meteorology (Michalakes, 2020; Rosenberg et al., 2023), quantum chemistry (Calvin et al., 2021; Chang et al., 2023), and machine learning (Dhar, 2020; Dodge et al., 2022). The high energy consumption of modern models and computer simulations is predominantly due to their exponentially growing size and complexity. To counteract this trend, a paradigm shift in the field is needed.

Instead of using increasingly larger models that exploit huge hardware infrastructures and massively parallel algorithms to squeeze out more performance, the amount of energy and resources consumed by each new model must be taken into account to avoid overshooting HPC energy demand. Thus, it becomes paramount to design the source code to make it as efficient as possible and remove unnecessary layers of complexity. One example from the field of machine learning is EfficientNet (Tan & Le, 2019) and, more generally, driver optimizations (Mittal, 2019).

In the field of machine learning, Schwartz et al. (2020) suggest substituting established performance indicators, such as accuracy, with performance measures that incorporate model efficiency into their evaluation. The authors outline the differences between Red AI (the current mainstream approach to improving the accuracy of state-of-the-art results through increases in computational power and model complexity) and Green AI (which yields state-of-the-art results or improvements without raising computational costs). Given that any increase in model size implies an increase in computational requirements, a Red AI approach is not deemed sustainable in the long term.

In some recent studies, the authors estimate the amount of carbon dioxide produced by their computations to give an indication of the energy efficiency and environmental impact of their work (Lacoste et al., 2019; Black et al., 2022; Lang et al., 2023; Gardner et al., 2024). The widespread use of such indicators and the further development of best practices that honor the use of energy-efficient models are a crucial step in reducing the energy consumption of HPC tasks and should be taken into account in future work across all disciplines.

2.2 Requirement Analysis

To obtain an overview of the current viewpoints and needs of existing HPC systems, we surveyed seven German scientific HPC system administrators. The interviewees' input was used to extract a list of minimum requirements to consider during the conceptualization phase. Their views on the proposed concept are summarized below. An excerpt of the questions asked during the survey and a summary of the corresponding answers are presented in Table 1. The collected responses are reported in no particular order to maintain speaker confidentiality. Based on the answers to our survey, we outline five main requirements for an energy-efficient HPC management system: interpretability/explainability, transparency, cost-effectiveness, redundancy, and data security. We incorporate these paradigms into the technical specifications of our artifact.

Overall, HPC energy consumption and its efficient management were perceived as highly relevant topics. Many HPC administrators expressed interest in educating themselves on the topic but pointed out other existing priorities within HPC management. In this sense, the surveyed stakeholders unanimously agreed that although reducing energy consumption is desirable and necessary, it must not come at the expense of computational performance or hardware lifetime.

Perhaps surprisingly, we also found the topic to be highly political given that the availability of computational research infrastructure is closely tied to medium-to-large-scale financial decisions. The acquisition of hardware and the required maintenance of the HPC systems must be estimated with high precision to avoid unnecessary costs. Therefore, all stakeholders of the HPC systems are interested in maintaining a high load on the systems while reducing operational costs. A high share of idle computational units on an HPC cluster might indicate, for instance, a suboptimal endowment of hardware, insufficient maintenance, or incorrect access policies.

The idle times of different HPC clusters vary strongly, both over time and across different HPC clusters. The typical loads reported from the larger HPC systems in our sample range from 50% to 80%. Based on the collected responses, we found that larger clusters usually work at higher capacities, while small clusters tend to have a higher and more strongly fluctuating share of idle units. Very small computational clusters, which are used by only a few users, reported the highest percentage of idle times.

Regarding the hardware used in the different HPC systems, we found that, on average, old hardware would benefit more from shutdown policies due to high energy consumption in idle state. In contrast, newer hardware often goes by default into a standby state with low energy consumption when idle (e.g., around 20% as compared to full load). However, even an energy consumption of 20% compared to full load can represent a high power consumption, depending on the specific hardware. Therefore, the concept presented here is viable for most currently used hardware.

All medium-sized or large-scale HPC systems manage their computing resources using scheduling software. Modern HPC scheduling software, such as SLURM (Simple Linux Utility for Resource Management), can efficiently use the available computing power: idle times can be minimized, for instance, by backfilling (i.e., small computing jobs are started earlier to use idle units). This scheduling technique is especially useful for clusters operating at high utilization and can help prevent restarting nodes on standby or shutdown states. However, if the current load on the relevant cluster is significantly below maximum capacity, additional tools are required to intelligently shut down and restart computational units to maximize energy efficiency.

A final point mentioned by many stakeholders is the availability obligations HPC cluster centers have to comply with. These are often contractually fixed and must be met to avoid issues, which might range from simple service delays to serious disruptions, depending on the HPC application. An established strategy for ensuring an agreed level of performance is the redundancy of both the computing units and the network ports. These redundancy requirements directly affect the implementation of any shutdown policy algorithm, such as IDLEWiSE. Indeed, any approach to optimizing energy consumption and server efficiency must envisage certain security buffers, backup units, and sufficient margins for estimation errors so as to guarantee contractual compliance.

Question	Answers	Requirement
Do you have any implemented strategy for minimizing the consumption of your HPC and data clusters? If so, describe it.	- No systematic strategy in place - Some practical rules and heuristics	- Interpretability / explainability
Do you have any statistics on the relative idle times of your HPC cluster(s)?	- Roughly 20% - No statistics available / confidential	- Transparency
Are you planning to implement any further strategies to improve the current energy efficiency?	- Yes, but not a priority. - No, budgetary / personnel constraints	- Cost-effectiveness
In an ideal setting, what would be the best approach to minimizing electricity consumption for your HPC cluster(s)?	- New, modern hardware - Fixed shutdown policies (e.g., in the evenings, holidays)	- Cost-effectiveness. - Interpretability / explainability
Are there any technical constraints limiting the adoption of energy-efficient policies and practices for your HPC cluster(s)?	- Compliance with availability obligations - Heterogeneity of units between clusters	- Redundancy
Are there any regulations or internal protocols limiting the adoption of energy-efficient policies and practices for your HPC cluster(s)?	- Privacy regulation - Financial decisions on cluster allocation are often not in the hands of HPC management itself	- Redundancy. - Data security

Table 1. Interview questions

3 Conceptualization

We conceptualized a tool aimed at minimizing the power consumption of HCP systems that operate below maximal capacity, focusing on small-scale HPCs and older hardware. The tool is based on simple learnable algorithms, such as decision trees, which estimate the future idle times of computational units based on previous usage data. It is designed to continuously learn the most efficient shutdown policies. In particular, it relies on online learning algorithms that adapt while the tool is in active use, as opposed to other tools that use strict rules or offline learning without adapting to the potentially changing environment in which they are used.

The tool was conceptualized based on evidence from conversations with German HPC system administrators. For example, the usage of federated learning for the training of algorithms was motivated by the high privacy requirements of many interviewees. We also describe implementation alternatives that accommodate security constraints and embed the trade-off between efficient energy usage and on-demand service provision. In the next subsection, we outline the insights from our survey, describe the technical specifications of our project concept, and further discuss the deployment of energy-efficient measures in the context of data and HPC centers.

3.1 Technical Specifications

Most of the time, the energy consumption of all types of computing resources – from CPUs to graphic processing units (GPUs) or other hardware accelerators – can be reduced with respect to their baseline. However, different approaches may be needed depending on whether server Peripheral Component Interconnect (PCI) Express Modules (SXM) graphics cards or regular PCI express (PCIe) versions of the same hardware are considered. For our solution concept, we chose to focus exclusively on HPC systems. Our motivation was twofold. On the one hand, scientific and academic institutions often own and locally manage HPC clusters that are composed of old and heterogeneous hardware, making it an interesting application for our conceptual approach. On the other hand, HPC clusters are one of the driving forces behind current AI advancements, so embedding energy efficiency in their management paradigms could yield great benefits, especially as their number and size scale up.

IDLEWiSE works by selectively switching off idle hardware to reduce energy consumption during periods of non-use. We define five states in which any resource can be and that differ based on energy consumption, as follows:

1) In-use: The device is currently active and executing instructions with a high-performance profile, similar to power state C0 in a high-performance state according to Unified Extensible Firmware Interface (UEFI) Process Power States (UEFI Forum, Inc., 2021). One example of this is the Nvidia A100 80GB SXM graphics card, which consumes approximately 400 W when active (NVIDIA, 2020).
2) Idle: The device is currently active but not in use, similar to power state C0 with performance state P0 (lowest). One example of this is the Nvidia A100 80GB SXM graphics card, which consumes around 50 W when idle.
3) Stand-by: The device is in lower power mode than idle (e.g., C1). This operational state is not supported by all devices.

4) Sleeping: Also sometimes called deep sleep, suspended, or hibernated. The device state is saved to lower-power devices – for instance, suspended to random access memory (RAM) or drives, such as hard disc drives (HDDs) or solid-state drives (SSDs). One example of this is the NVidia Jetson Nano SC7 state, the lowest power state of a device without hardware shut-off. However, this operational state is not supported by all devices.

5) Shutdown: The device is not used and turned off. Power can be unplugged without any problem and without having to switch states before. This operational state causes no power draw.

To decide which state each resource should be put in for any given time, it is necessary to develop a predictor that can reliably predict the medium-term resource demand, as shown in Figure 1. This medium-to-long-term demand estimation is used to correctly transition sources from idle or stand-by to sleeping state and vice versa while preventing unnecessary switching between states and loss of available HPC computing at the same time. Because state transitions from shutdown to in-use may require considerable time, the predictor must take into account these constraints in its policy implementation and extrapolate future resource requirements based on historical patterns and current system behavior. Estimated transitions to the In-Use state are as follows:

\ From idle: almost instantaneous (few milliseconds), like starting a program from a desktop

\ From stand-by: quick (hundreds of milliseconds to a few seconds), like starting a program from a lock screen on a laptop

\ From sleeping: slow (seconds), like starting a program from a sleeping laptop

\ From shutdown: very slow (minutes)

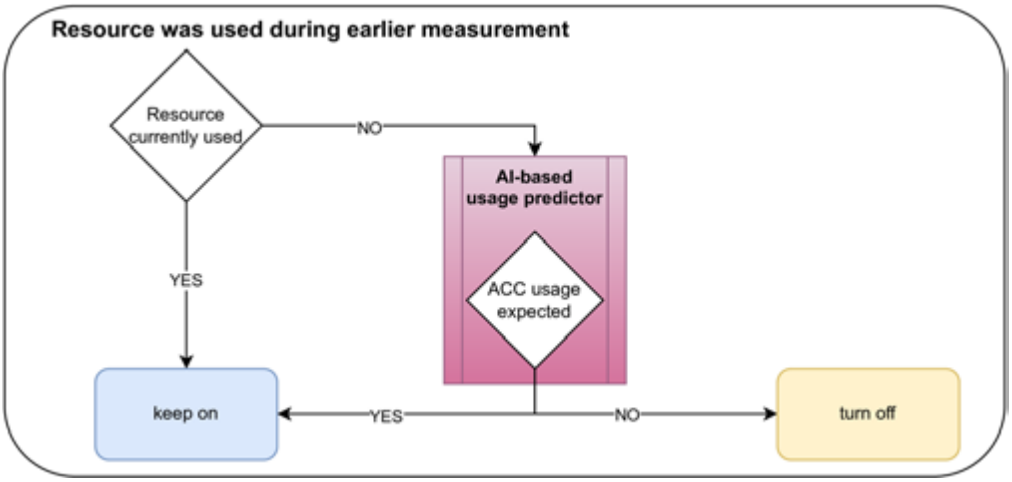
We would like to highlight that the difference between the transition from sleeping to in-use and from shutdown to in-use is very large and, therefore, shutdown cannot be used as a substitute for sleeping for most existing hardware. However, if the prediction can be made far in advance (i.e., many minutes) or the penalty for a false prediction is low, using the shutdown state in place of the sleeping state is a viable alternative. Transitions from and to the shutdown state should be used carefully because if the prediction is wrong, a significant amount of power is needed to start an unnecessary node; similarly, if a node was not started and, consequently, a job has a significantly longer waiting time, costs may be incurred. This is especially relevant for clusters that are also used for interactive development, such as teaching clusters.

In addition, the energy cost to start a resource from the shutdown state is usually much higher than for other state transitions. Therefore, the prediction has to be associated with high confidence to avoid incorrect decisions as much as possible. For every prediction, priority should be given to ensuring that the requested resources are always available. In an ideal implementation, these are set to the idle state immediately before the associated request is sent, allowing unused resources to remain in the sleeping state for as long as possible.

In Figure 1, we outline the decision-making procedure for a single resource, depending on whether or not the specific resource was used during an earlier measurement (cases a and b, respectively). In both cases, the resource must be in a turned-on state (idle or in use) if the resource is requested or is currently in use. If the resource is not requested or is no longer in use, the predictor will extrapolate from the available data whether usage is expected in the short term, which corresponds to the time it takes to transition the resource from sleeping to ready/idle. If usage is expected, the resource will stay on (case a) or will be turned on (case b). Otherwise, it will be turned off (case a) or stay turned off (case b). Importantly, the extrapolation is highly specific to the cluster used and based on cycles in the usage of the cluster. The tool assumes that clusters follow seasonal patterns, which can be learned during training. For instance, for clusters deployed in academia, there might exist cyclic patterns associated with semester obligations: for companies they could be related to regular events such as quarterly reports. The length of these cycles should be tunable via a configurable time horizon. Outliers, such as project deadlines, would be the main contributing factor to extended wait times or higher idle periods due to distorted learning.

To avoid large additional consumption of computational resources and energy, the predictor should require few computational resources for online load prediction and run lightly and efficiently during deployment. It should also be easily adaptable to include cluster-dependent parameters and change the prediction behavior, such as the aforementioned load cycles.

(a)



(b)

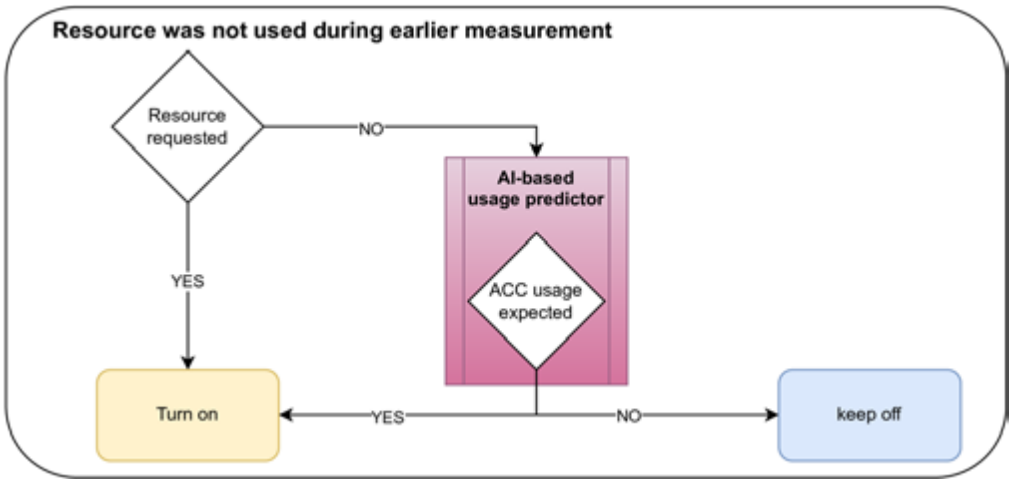


Figure 1: Schematic decision-making behavior for the predictor in two exemplary cases

As another example, the tradeoff between energy savings on the one hand and hardware lifetime and computational availability on the other hand can be adapted to the special requirements of the relevant HPC system by tuning the expected idle time penalty of the predictor, as shown in Figure 2. One possibility for achieving such predictor efficienctly is a decision tree with an additional simple site-specific heuristic. The algorithms may be pre-trained on general HPC data (acquired in advance and trained via federated learning or with anonymized data) and then adapted to the specific requirements of each HPC system using recent local data for fine-tuning and online training.

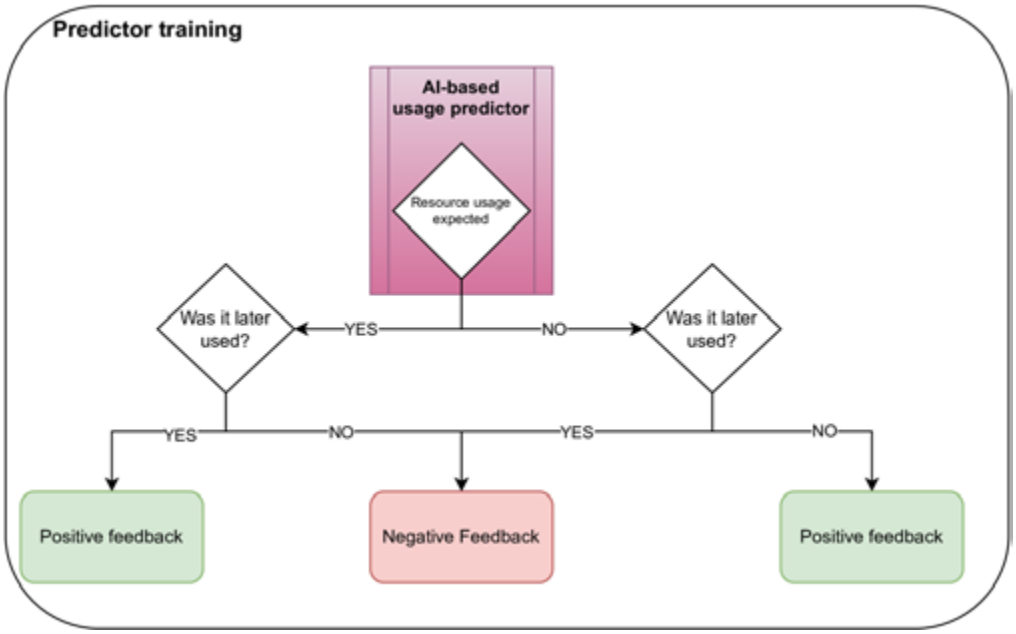


Figure 2: Training process for the AI-based predictor

Note: Training can be based on any cost metric, such as a reward for reinforcement learning, loss calculation for backpropagation, or new decision samples for a deeper decision tree.

4 Validation

4.1 Comparison of Stakeholder Requirements and IDLEWiSE Design Specifications

To ensure the conceptual alignment of IDLEWiSE with real-world needs, stakeholder requirements derived from interviews were compared with the proposed system's design specifications. This comparison highlights how the concept addresses critical concerns such as energy efficiency, cost-effectiveness, and usability in the context of HPC clusters.

Stakeholders emphasized the need for transparent and comprehensible energy-saving strategies given the reliance on practical rules and heuristics in current HPC operations. To address this, IDLEWiSE incorporates a decision tree–based prediction model, which offers high interpretability and logical decision-making pathways. This design choice enables users to trace and understand resource state transitions, fostering trust in the system. Although this approach aligns with the explainability requirement, its effectiveness in specific HPC scenarios will necessitate empirical confirmation. Similarly, transparency was identified as a major gap, with stakeholders highlighting a

lack of systematic reporting on idle times and energy usage. IDLEWiSE aims to fill this gap by integrating logging and reporting mechanisms that provide visibility into system states, predicted idle periods, and energy consumption.

Budgetary constraints and limited personnel resources were mentioned as key barriers to implementing energy-efficient strategies. IDLEWiSE addresses this by offering a cost-effective solution based on software-driven optimizations rather than costly hardware upgrades. Its lightweight prediction model ensures minimal computational overhead while promising significant energy savings. However, the extent of these savings and the overall cost-benefit ratio must be validated through online validation. Another critical consideration was the need for redundancy because technical constraints such as hardware heterogeneity and availability obligations demand robust handling of resource allocation. IDLEWiSE accommodates this by adapting its state-transition logic to cluster-specific configurations and usage cycles via online learning, ensuring reliability while maintaining energy efficiency. In addition to operational concerns, stakeholders raised issues related to compliance with privacy regulations and internal protocols. Although IDLEWiSE does not explicitly incorporate data security measures, the localized training of prediction models inherently reduces the risk of sensitive data exposure.

Therefore, as a conceptual framework, IDLEWiSE demonstrates strong alignment with stakeholder priorities, addressing key needs for interpretability, transparency, cost-effectiveness, and reliability. By directly targeting idle resource management, IDLEWiSE offers a promising pathway for improving energy efficiency in HPC clusters, but empirical testing and iterative refinement will be essential to achieve its intended goals in practical settings.

4.2 Validation of the IDLEWiSE Concept

To conceptualize a step-wise validation for IDLEWiSE, we use a software quality model. Software quality models are conceptual frameworks used to assess software quality at any stage of its development lifecycle; they ensure that the architectural characteristics of software match user requirements (as highlighted in the previous subsection). We base our validation concept on the FURPS software quality model, which considers the following aspects: functionality, usability, reliability, performance, and supportability (Singh and Kassie, 2018). Our validation concept consists of: *i*) software testing in a simulation and test environment; *ii*) online validation on live HPC clusters; *iii*) user feedback; *iv*) scalability tests; and *v*) comparison to other approaches.

Phase *i* involves the creation of a simulated and controlled test environment that mimics the operating conditions of HPC clusters. In this environment, the IDLEWiSE system is implemented to verify its functionality and performance

in various load and operating conditions based on real-world examples. By using a simulated environment, the system can be tested in controlled conditions, and potential problems may be identified before deployment. This enables a systematic evaluation of the software and targeted optimization before live implementation. Moreover, this environment can be used to train the learned algorithms offline based on data collected from real-world systems. Here, the goal is to analyze the performance of IDLEWiSE in various scenarios and ensure its adaptability to dynamic HPC cluster conditions. This step is crucial for fine-tuning the system's algorithms, making them more responsive and effective in optimizing resource allocation and energy efficiency. Ideally, this step is performed for each of the targeted clusters individually.

In Phase *ii*, online validation of IDLEWiSE is performed. The trained algorithm is deployed in a live HPC cluster to test its real-time responsiveness and adaptability to dynamic changes in workload and resource demands. During this step, the tool is constantly monitored to ensure the reliability of the system in practical, dynamic settings.

Phase *iii* requires gathering feedback from end-users and administrators on the interface, potential issues, and possible additional features. Phases *i* to *iii* can be iterated multiple times so that each feedback loop improves the usability of the tool. This is critical to provide a low barrier to entry for its usage.

During Phase *iv*, scalability tests are conducted to confirm IDLEWiSE's ability to handle increasing workloads and larger HPC clusters without significant performance decreases. Although a small overhead is expected, this step is essential to ensure that the software can effectively scale up to meet the demands of growing computational environments and adapt to heterogeneous scenarios.

Finally, Phase *v* compares IDLEWiSE to other approaches, benchmarking its performance, efficiency, and functionalities against existing solutions. This comparative analysis helps validate the uniqueness and superiority of IDLEWiSE in the context of performance and energy savings, providing insights into its strengths and areas for improvement.

Validation is concluded by documenting the validation process, including by producing reports on each step. These reports include the main findings and identify challenges as well as potential improvements. Comprehensive documentation can serve as a valuable resource for future reference, ensuring transparency, reproducibility, and the continuous improvement of IDLEWiSE.

5 Discussion

In this section, we discuss aspects of the environmental footprint of HPC clusters that would not be directly tackled by the proposed tool. Although IDLEWiSE could improve the utilization of existing hardware, its overall impact on energy consumption and the environment will be determined by additional external factors. In the following, we highlight four: the environmental cost of hardware production, carbon intensity, cooling, and energy-efficient software.

Extensive literature has examined the environmental cost of HPC hardware production. The production chain of hardware heavily depends on rare-earth elements such as neodymium and dysprosium, which are primarily mined in large-scale extraction sites, with heavy consequences on the surrounding land and population. Moreover, the disposal of HPC hardware can have serious adverse health and environmental consequences as electronic waste is often exported to developing countries, where it is dismantled in unsafe conditions. By boosting the efficiency of existing old architecture systems, IDLEWiSE can improve the usage efficiency of existing hardware and extend products' lifespans. This can be done by setting devices into a lower-power state and reducing wear and tear.

Research increasingly shows that reducing energy consumption in absolute terms may not be the most suitable measure for minimizing negative externalities. Recent work by Dodge et al. (2022) has directed attention to the carbon intensity of data and HPC clusters as a better proxy for the environmental impact of cloud computing in terms of mere energy consumption. The carbon intensity of a grid depends on the electricity source fueling it. For example, grids powered by coal generate higher carbon emissions per kWh of electricity than those fueled by hydroelectricity or solar power. A recent study by Dodge et al. (2022) demonstrates that large language models trained on electricity from a highly carbon-intense grid, such as GPT-3, emit 20 times as many tons of CO2 as an environmentally friendlier model even though they require only three times more electricity approximately.

Traditional HPC systems contribute significantly to data centers' overall energy consumption. Exploring and implementing technologies such as liquid cooling or advanced heat exchangers could mitigate the energy demands of cooling, leading to a more sustainable HPC infrastructure. Efficient scheduling practices lead to lower power usage by the system components, which directly translates to lower related heat production and reduced cooling needs of the system. Given that cooling systems are always active to maintain constant environmental temperature and need additional power to run, further synergistic gains in the overall energy consumption of a company can be expected from this solution. Finally, although this is not within the scope of the present work, investigating the role of energy-efficient software execution will be essential.

6 Concluding Remarks

This concept paper presented IDLEWiSE, a software tool that aims to reduce the energy consumption of HPC clusters. Our approach focuses on the AI-assisted shutdown of idle computational units based on projected idle times to mitigate the environmental impact and operational costs associated with HPC clusters. The actual implementation of the tool remains a task for future endeavors, but our concept paper lays a robust foundation for its realization by delineating key components, methodologies, and further contextual considerations.

The first and main goal of this paper is to raise awareness of a gap in the currently available tools for reducing the energy consumption of HPC servers and advocate for a bottom-up paradigm shift towards more resource-aware computational systems. A survey conducted among German scientific HPC cluster administrators underscores the necessity of such projects, the pressing need for energy-efficient solutions, and the relevance and timeliness of the IDLEWiSE tool.

The IDLEWiSE project is complementary to prior academic research on the topic. It seeks to enrich the toolbox of energy-minimizing algorithms with a focus on small, local HPC clusters, such as those managed by universities, research centers, and research groups. Implementing an AI-assisted system manager such as IDLEWiSE could lead to significant efficiency gains in the consumption of existing HPC clusters so that research institutes may reduce their energy and carbon footprints at an accessible cost and without having to purchase new hardware.

Nonetheless, we acknowledge the possible limitations of the concept, whose implementation is outside the scope of our preliminary assessment. First, the efficiency gains produced by an AI solution may not be significant in comparison to carefully crafted heuristic algorithms. A second limitation of an AI-based approach is its lack of interpretability, which could deter decision-makers from deploying it in operational scenarios. In this regard, future research may target the deployment of easy-to-use, scalable, and interpretable software to meet the needs of different groups of practitioners.

References

Belyaev, N., Klimentov, A., Krasnopevtsev, D., Velikhov, V., Konoplich, R., & Prokofiev, K. (2017). *High performance computing system in the framework of the Higgs boson studies*. CERN. https://cds.cern.ch/record/2293317

Betkaoui, B., Thomas, D. B., & Luk, W. (2010). Comparing performance and energy efficiency of FPGAs and GPUs for high productivity computing. *2010 International Conference on Field-Programmable Technology*, 94–101. https://doi.org/10.1109/FPT.2010.5681761

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U. S., Purohit, S., Reynolds, L., Tow, J., Wang, B., & Weinbach, S. (2022). GPT-NeoX-20B: *An Open-Source Autoregressive Language Model* (arXiv:2204.06745). arXiv. http://arxiv.org/abs/2204.06745

Calvin, J. A., Peng, C., Rishi, V., Kumar, A., & Valeev, E. F. (2021). Many-Body Quantum Chemistry on Massively Parallel Computers. *Chemical Reviews, 121*(3), 1203–1231. https://doi.org/10.1021/acs.chemrev.0c00006

Castro, P. de O. (2022). *High Performance Computing code optimizations: Tuning performance and accuracy*. Université Paris-Saclay.

Chang, C., Deringer, V. L., Katti, K. S., Van Speybroeck, V., & Wolverton, C. M. (2023). Simulations in the era of exascale computing. *Nature Reviews Materials, 8*(5), Article 5. https://doi.org/10.1038/s41578-023-00540-6

Daid, R., Kumar, Y., Hu, Y.-C., & Chen, W.-L. (2021). An effective scheduling in data centres for efficient CPU usage and service level agreement fulfilment using machine learning. *Connection Science, 33*(4), 954–974. https://doi.org/10.1080/09540091.2021.1926929

Dhar, P. (2020). The carbon impact of artificial intelligence. *Nature Machine Intelligence, 2*(8), Article 8. https://doi.org/10.1038/s42256-020-0219-9

Dodge, J., Prewitt, T., Tachet des Combes, R., Odmark, E., Schwartz, R., Strubell, E., Luccioni, A. S., Smith, N. A., DeCario, N., & Buchanan, W. (2022). Measuring the Carbon Intensity of AI in Cloud Instances. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 1877–1894. https://doi.org/10.1145/3531146.3533234

Duan, L., Zhan, D., & Hohnerlein, J. (2015). Optimizing Cloud Data Center Energy Efficiency via Dynamic Prediction of CPU Idle Intervals. *2015 IEEE*

8th International Conference on Cloud Computing, 985–988. https://doi.org/10.1109/CLOUD.2015.133

Evans, R., & Gao, J. (2016, Juli 20). *DeepMind AI Reduces Google Data Centre Cooling Bill by 40%*. Google DeepMind. https://deepmind.google/discover/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40/

Freitag, C., Berners-Lee, M., Widdicks, K., Knowles, B., Blair, G. S., & Friday, A. (2021). The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns, 2*(9), Article 100340. https://doi.org/10.1016/j.patter.2021.100340

Gamble, C., & Gao, J. (2018, August 17). *Safety-first AI for autonomous data centre cooling and industrial control*. Google DeepMind. https://deepmind.google/discover/blog/safety-first-ai-for-autonomous-data-centre-cooling-and-industrial-control/

Gardner, J. L. A., Baker, K. T., & Deringer, V. L. (2024). Synthetic pre-training for neural-network interatomic potentials. *Machine Learning: Science and Technology, 5*(1), Article 015003. https://doi.org/10.1088/2632-2153/ad1626

Kurz, M., Offenhäuser, P., Viola, D., Shcherbakov, O., Resch, M., & Beck, A. (2022). Deep reinforcement learning for computational fluid dynamics on HPC systems. *Journal of Computational Science, 65*, Article 101884. https://doi.org/10.1016/j.jocs.2022.101884

Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). *Quantifying the Carbon Emissions of Machine Learning* (arXiv:1910.09700). arXiv. https://doi.org/10.48550/arXiv.1910.09700

Lang, N., Jetz, W., Schindler, K., & Wegner, J. D. (2023). A high-resolution canopy height model of the Earth. *Nature Ecology & Evolution, 7*(11), Article 11. https://doi.org/10.1038/s41559-023-02206-6

Lin, B., Basu Roy, R., Wang, D., Samsi, S. S., Gadepally, V., & Tiwari, D. (2023). Toward Sustainable HPC: Carbon Footprint Estimation and Environmental Implications of HPC Systems. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 19*, 1–15. https://dl.acm.org/doi/abs/10.1145/3581784.3607035

Markidis, S., Chien, S. W. D., Laure, E., Peng, I. B., & Vetter, J. S. (2018). NVIDIA Tensor Core Programmability, Performance & Precision. *2018 IEEE International Parallel and Distributed Processing Symposium Workshops* (IPDPSW), 522–531. https://doi.org/10.1109/IPDPSW.2018.00091

Metropolis, N., & Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association, 44*(247), 335–341. https://doi.org/10.1080/01621459.1949.10483310

Michalakes, J. (2020). HPC for Weather Forecasting. In A. Grama & A. H. Sameh (Eds.), *Parallel Algorithms in Computational Science and Engineering* (pp. 297–323). Springer International Publishing. https://doi.org/10.1007/978-3-030-43736-7_10

Mishra, A., & Khare, N. (2015). Analysis of DVFS Techniques for Improving the GPU Energy Efficiency. *Open Journal of Energy Efficiency,* 4(4), Article 4. https://doi.org/10.4236/ojee.2015.44009

Mittal, S. (2019). A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform. *Journal of Systems Architecture, 97*, 428–442. https://doi.org/10.1016/j.sysarc.2019.01.011

NVIDIA (2020). *NVIDIA A100 Tensor Core GPU Datasheet*. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet.pdf.

Pinheiro, E., Bianchini, R., Carrera, E. V., & Heath, T. (2001). *Load balancing and unbalancing for power and performance in cluster-based systems*. Rutgers University. https://doi.org/10.7282/t3-agfw-yt73

Raïs, I., Orgerie, A.-C., & Quinson, M. (2016). Impact of Shutdown Techniques for Energy-Efficient Cloud Data Centers. In J. Carretero, J. Garcia-Blas, R. K. L. Ko, P. Mueller, & K. Nakano (Eds.), *Algorithms and Architectures for Parallel Processing* (Vol. 10048, pp. 203–210). Springer International Publishing. https://doi.org/10.1007/978-3-319-49583-5_15

Rosenberg, D., Flynt, B., Govett, M., & Jankov, I. (2023). Geofluid Object Workbench (GeoFLOW) for Atmospheric Dynamics in the Approach to Exascale: Spectral Element Formulation and CPU Performance. *Monthly Weather Review, 151*(9), 2521–2540. https://doi.org/10.1175/MWR-D-22-0250.1

Siegmund, N., Dorn, J., Weber, M., Kaltenecker, C., & Apel, S. (2022). *Green Configuration: Can Artificial Intelligence Help Reduce Energy Consumption of Configurable Software Systems?* IEEE Xplore.

Singh, J., & Kassie, N. B. (2018). User's Perspective of Software Quality. *Second International Conference on Electronics, Communication and Aerospace Technology* (ICECA). https://doi.org/10.1109/ICECA.2018.8474755

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM, 63*(12), 54–63. https://doi.org/10.1145/3381831

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, 6105–6114. https://proceedings.mlr.press/v97/tan19a.html

Trebotich, D. (2024). Exascale CFD in Heterogeneous Systems. *Journal of Fluids Engineering*, 1–19. https://doi.org/10.1115/1.4064534

UEFI Forum, Inc. (2021). 8.1. Processor Power States – ACPI Specification 6.4 documentation. ACPI Specification, Processor Power States. https://uefi.org/htmlspecs/ACPI_Spec_6_4_html/08_Processor_Configuration_and_Control/processor-power-states.html

Ullrich, A., Rehak, R., Hamm, A., & Mühlhoff, R. (2024). Sustainable Artificial Intelligence – Critical and Constructive Reflections on Promises and Solutions, Amplifications and Contradictions. *Weizenbaum Journal of the Digital Society, 4*(1), 1–7. https://doi.org/10.34669/wi.wjds/4.1.1

Weerts, H., Dudík, M., Edgar, R., Jalali, A., Lutz, R., & Madaio, M. (2023). Fairlearn: Assessing and Improving Fairness of AI Systems. *Journal of Machine Learning Research, 24*(257), 1–8.

Zvyagin, M., Brace, A., Hippe, K., Deng, Y., Zhang, B., Bohorquez, C. O., Clyde, A., Kale, B., Perez-Rivera, D., Ma, H., Mann, C. M., Irvin, M., Ozgulbas, D. G., Vassilieva, N., Pauloski, J. G., Ward, L., Hayot-Sasson, V., Emani, M., Foreman, S., … Ramanathan, A. (2023). GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *The International Journal of High Performance Computing Applications, 37*(6), 683–705. https://doi.org/10.1177/10943420231201154s

---

Date received: February 2024
Date accepted: November 2024