15, February, 2026

Thermal-aware Scheduling: A Review of Machine Learning and Optimization Strategies

Atharva Kharwadkar
University of California, Davis
e-mail: akharwadkar@ucdavis.edu

Abstract: With its applications assimilating into everyday life, Artificial Intelligence (AI) has and will likely continue to define this era of technological growth. Today, there are Large Language Model powered office workflows, Machine Learning (ML) autonomous agents, and now AI recursively tending to the body of hardware that enables it. A byproduct of performing AI tasks on a hardware platform is heat. Current studies relate that heat impedes hardware performance, and as such, this has become a topic of investigation. However, hardware platforms come in many forms, ranging from personal devices at the micro-level, to collective hardware systems that form a data center at the macro-level. For this reason, this review focuses on the underlying trends in technology and strategy.

Index Terms: Thermal-aware, scheduling, machine learning.

1. Introduction:

Whilst one study might lay the blueprint for managing thermals of particular hardware, the principles and processes that enable decision-making are still relevant to other uses. Many frameworks are principled on ML to enact predictive control of the system; ML has been explored in varying paradigms such as reinforcement learning (RL), reduced order learning for model predictive control (MPC), and even the novel introduction of Multi-objective Reinforcement Learning (MORL) [1, 2]. An MPC strategy by Dowling et al. has shown to train on thermal simulation and predict accordingly, but not all frameworks place emphasis on thermal modeling [1]. To that end, RL strategies have shown greater focus on trial and error approaches to learning. There is also, with good reason, a shared aspiration between researchers to attain “lightweightedness” or “cheaper computation” in framework features [1, 3]. Reviewing these studies will help cover existing frameworks, and maybe lead to more novel frameworks for Thermal-aware Scheduling (TAS).

2. Discussion:

Decision-Making:

Based on current literature, it would seem that RL has been a largely popular approach to applying decision-making. EdgeEngine, developed by Ahmadi et al., employs a lightweight RL system to make “swift” and “conservative” adjustments to Nvidia Jetson TX2 hardware running ML tasks [3]. These adjustments involve navigating between configurations of running frequencies for the Central Processing Unit (CPU), Graphical Processing Unit (GPU), and memory. EdgeEngine delivers just 3% greater energy consumption and 2% longer runtimes in comparison to ORACLE, an engine that uses perfect knowledge to simulate the most optimal process. To facilitate this, the RL agent generates a q-table storing the expected reward value for each action in any state. At each time step, the agent seeks to extract the maximal reward (exploitation), and randomly, it will attempt an action that has unknown results (exploration). When the chosen action is unsuccessful, the RL agent is penalized – RL strategies only learn based on the results of their exploration. RL is therefore a model-free approach, and thus EdgeEngine proposes a framework that is versatile between hardware platforms.

Kanani et al. develop RL deployment differently in THERMOS, arguing that the goal of traditional RL is too one-dimensional [2]. The creators of THERMOS expand on this by illustrating that traditional RL constrains all optimization preferences – such as energy consumption, latency, and thermal constraints – into just one objective. In the context of ML, an objective is constructed by the sum of weights in respect to each established preference. Here, THERMOS introduces novelty in MORL. The traditional ML objective is unraveled into all its constituent dimensions and the weights are placed into a vector which MORL will train against. Successively, MORL will generate approaches that aim to optimize all preferences in parallel. EdgeEngine enacts scheduling temporally, but THERMOS schedules tasks to chiplet clusters that should not exceed thermal constraints – essentially scheduling spatially. THERMOS achieves 89% faster execution time for AI workloads on chiplet PIM architectures, indicating that the impediment of thermal elements has been handled.

A study by Lu et al. (TAWS) also uses RL-driven decision-making, but adds a safety protocol [4]. TAWS handles scheduling spatiotemporally, the RL agent decides which GPUs are put to use, at what frequency-voltage pair, and the fraction of the task if it is split between multiple GPUs. Due to their explorative nature, RL agents can frequently cause failure. However, TAWS challenges this by developing Constrained Policy Optimization (CPO). In essence, CPO periodically projects risk-averse moderation onto the RL agent's scheduling to guarantee energy and thermal limits.

POD-TAS presents an alternative approach to applying decision-making by use of MPC [1]. Dowling et al. principle POD-TAS on reduced-order learning to perform real-time thermal modeling. While POD-TAS does make ML-accelerated thermal judgement to manage CPU cores, the decision-making is treated entirely as an optimization problem. POD-TAS aims to ensure that all cores are working continuously without suffering from suboptimal thermal impediment. To tackle this, POD-TAS designates cores between mutually exclusive states: they are either Idle or Running, and also Cool, Warm, or Hot. Typically, tasks are scheduled to Cool or Warm cores that are either Running or Idle, with Hot cores being expressly forbidden from becoming Running. POD-TAS therefore works spatially between mapped cores.

A study by Gao develops TAS for edge-side inference tasks resulting in 20-50% improvement in performance without employing ML in scheduling [5]. Instead, this framework uses programming to perform mathematical optimization before scheduling. This process includes minimizing the latency delivered by the CPU, GPU, and Neural Processing Unit, while working under the constraints of a threshold speed that might generate the maximum feasible temperature. This study proves that it is not imperative to generate decision-making through ML, but the quality of the results in this study do not assert the effectiveness of the approach as much as studies that do employ ML.

Thermal Modeling

Evidence supporting the importance of thermal modeling is highly disorganized in these studies. Frameworks that use RL agents do not come with an understanding of heat transfer, or the notion that pushing hardware to run under greater load can cause excess heat. It is more likely to learn the correlation between high loads and heat after being penalized for failing to meet thermal constraints. The RL agent does not model heat transfer, but rather maps scheduling configurations to the certainty that it will meet thermal constraints.

While RL-driven TAS frameworks entirely bypass thermal modeling, other TAS frameworks show massive efforts to produce quality predictions. One such framework is POD-TAS; as stated earlier, POD-TAS is principled on reduced-order learning so as to offer real-time thermal modeling and MPC thereof. In comparison to EdgeEngine and THERMOS, the implementation of ML in POD-TAS has been transposed from the decision-making process to the thermal modeling process. POD-TAS starts by training on high-fidelity thermal simulations on a model of the hardware, and produces a model that can approximate the heat response in a range of conditions. These simulations cannot be processed in real-time. So, Dowling et al. use Proper Orthogonal Decomposition (POD) to extract the most pertinent details, and return to the Partial Differential heat transfer equations that fuel the simulation software. Because these equations are computationally expensive, Dowling et al. apply Galerkin Projection

to transform Partial Differential Equations into computationally cheaper Ordinary Differential Equations. Consequently, POD-TAS offers a high accuracy thermal understanding in real-time to its decision-making process.

Though POD-TAS engages in a complex thermal-modeling process, other non-RL frameworks have shown to develop modeling straightforwardly. For instance, Gao recognizes dependencies between hardware and thermal properties. For this framework, the temperature in a hardware system is decomposed into the environment temperature, and temperature induced by power. The latter is shown to be a product between the thermal resistance of the hardware, and power. The thermal resistance is constant, but Gao calculates power by performing a series of arithmetic using hardware operation speeds, and voltage. Both properties show to produce power. This form of modeling, however, does not offer a spatial map of temperature. Therefore, the framework sacrifices spatial inferences that can further optimize decision-making in the way that POD-TAS can. Together, these studies prove that the importance of thermal modeling is highly dependent on the scheduling system that it is subsidiary to.

3. TAS in the Future:

It is necessary to produce replication studies; many of these frameworks have been experimented on simulated hardware, and are therefore unproven in real applications. That is not to say that the concepts they explore are not useful. Through these studies, it is shown that programming optimization might be the most straightforward approach to developing TAS, but the complexities of RL and MPC prove to be most worthwhile. Despite this, developing TAS will not be as simple as choosing between RL and MPC. The variance in approaches taken by studies show choice. As such, novel approaches must be constantly pursued.

This review did not emphasize the theoretical gains of any framework because it is not the sole metric for successful TAS. The principles that support TAS can be translatable between different hardware systems, but the required outcome might differ. In the context of a datacenter, a 1% improvement in performance can become significant over the course of a year, but failing frequently can be exponentially disastrous. The same cannot be said for TAS frameworks designed for personal devices.

There will be novel strategies unexplored by this literature review, especially since hardware intended for computation is constantly evolving. Hopefully, TAS frameworks in future applications will stem from a highly methodical process of selecting features from what has been explored in contemporary studies.

Works Cited:

[1] Anthony Dowling, Lin Jiang, Ming-Cheng Cheng, Yu Liu, “Regulating CPU Temperature With Thermal-Aware Scheduling Using a Reduced Order Learning Thermal Model ,” *Future Generation Computer Systems*, vol. 166, pp. 297-311, May 2025, doi: 10.1016/j.future.2024.107687.

[2] Alish Kanani, Lukas Pfromm, Harsh Sharma, Jana Doppa, Partha Pande, Umit Ogras, “THERMOS: Thermally-Aware Multi-Objective Scheduling of AI Workloads on Heterogeneous Multi-Chiplet PIM Architectures,” *ACM Transactions on EMbedded COmputing Systems*, vol. 24, no. 5s, pp. 1-26, Sep. 2025, doi: 10.1145/3762655.

[3] Amirhossein Ahmadi, Hazem A Abdelhafez, Karthik Pattabiraman, Matei R Ripeanu, “EdgeEngine: A Thermal-Aware Optimization Framework for Edge Inference,” *SEC ‘23*, pp. 67-79, Aug. 2024, doi: 10.1145/3583740.3626616.

[4] Rui Lu, Dan Wang, “A Thermal-Aware Workload Scheduler for High-Performance LLM Inference in Cooling-Regulated Datacenters,” *ACM SIGENERGY Energy Informatics Review*, vol. 5, issue 2, pp. 98-104, Jul. 2025, doi: 10.1145/3757892.3757906.

[5] X. Gao, "TAS: A Temperature-Aware Scheduling for Heterogeneous Computing," in IEEE Access, vol. 11, pp. 54773-54781, 2023, doi: 10.1109/ACCESS.2023.3281839.