

When Faster Is Worse: Time-Based Metrics Fail Under Energy Constraints

Oren Hadri
Independent Researcher

Abstract

Execution time is the dominant optimization and evaluation metric in systems and machine learning research. This practice implicitly assumes that faster execution preserves ordering with respect to energy consumption or energy-bounded productivity. We show that this assumption does not hold in general.

For execution policies that preserve total computational work but induce different power profiles—including frequency selection, temporal scheduling, and resource allocation—completion time does not uniquely determine energy consumption. We present a minimal formal argument establishing non-injectivity of the time–energy mapping, together with a controlled empirical witness on a commodity GPU. Our results demonstrate that time-only metrics can select energetically inferior execution policies under realistic power management mechanisms.

1 Introduction

Performance evaluation in modern computing systems is overwhelmingly time-centric. Metrics such as wall-clock runtime, throughput, and Job Completion Time (JCT) dominate both optimization objectives and empirical evaluation. This convention implicitly assumes that execution time is an information-preserving proxy for efficiency.

This assumption is fragile once power consumption depends on execution policy. On contemporary hardware, dynamic voltage and frequency scaling (DVFS), utilization-dependent boost states, and parallelism choices induce heterogeneous power profiles across policies that execute identical computational work. In such settings, faster execution may increase average power sufficiently to raise total energy consumption.

This paper makes a narrow but fundamental claim: under power-dependent execution policies, time-only metrics do not, in general, preserve ordering

with respect to energy consumption. As a result, optimizing for execution time alone no longer defines a well-posed objective when energy or power is a binding constraint.

2 Background: Time and Energy

Energy consumption of a single execution is defined as:

$$E = \int_0^T P(t) dt,$$

where T is execution time and $P(t)$ is instantaneous power draw. Runtime optimization minimizes T ; energy optimization minimizes E . These objectives coincide only if $P(t)$ is independent of execution policy.

A common abstraction of dynamic power is:

$$P_{\text{dyn}} \propto C \cdot V^2 \cdot f.$$

Under DVFS, voltage scales with frequency, yielding superlinear growth of power with performance. As a result, reductions in execution time do not imply proportional reductions in energy.

3 Formal Observation: Time–Energy Ordering Can Fail

Consider a fixed computational workload executed under two policies π_1 and π_2 (e.g., schedulers, DVFS rules, or placement strategies). Assume both policies complete the same total work (i.e., they differ only in how the work is executed over time). Let the completion times satisfy

$$T(\pi_1) < T(\pi_2).$$

Define total energy as the time integral of instantaneous power:

$$E(\pi) = \int_0^{T(\pi)} P_\pi(t) dt.$$

Equivalently, using average power $\bar{P}(\pi) \triangleq \frac{1}{T(\pi)} \int_0^{T(\pi)} P_\pi(t) dt$,

$$E(\pi) = T(\pi) \cdot \bar{P}(\pi).$$

Lemma (Ordering failure). Even if π_1 completes faster, it may consume more total energy than π_2 . In particular, if

$$\frac{\bar{P}(\pi_1)}{\bar{P}(\pi_2)} > \frac{T(\pi_2)}{T(\pi_1)},$$

then $E(\pi_1) > E(\pi_2)$.

Proof. Starting from the condition,

$$\frac{\bar{P}(\pi_1)}{\bar{P}(\pi_2)} > \frac{T(\pi_2)}{T(\pi_1)} \iff T(\pi_1)\bar{P}(\pi_1) > T(\pi_2)\bar{P}(\pi_2).$$

Using $E(\pi) = T(\pi)\bar{P}(\pi)$, we obtain $E(\pi_1) > E(\pi_2)$ despite $T(\pi_1) < T(\pi_2)$. \square

Interpretation. Completion time is not a monotone proxy for energy: a policy can reduce runtime while increasing average power by a larger multiplicative factor, leading to higher total energy.

Example. Let $T(\pi_1) = 1\text{s}$, $\bar{P}(\pi_1) = 10\text{W}$, so $E(\pi_1) = 10\text{J}$. Let $T(\pi_2) = 2\text{s}$, $\bar{P}(\pi_2) = 4\text{W}$, so $E(\pi_2) = 8\text{J}$. Then π_1 is faster but consumes more energy.

3.1 Extension: Resource Allocation and Multi-GPU Policies

We define an execution policy broadly as any choice that affects the power profile while preserving total computational work, including frequency selection, temporal scheduling, and resource allocation (number of active devices).

Consider a fixed workload W executed on k identical GPUs. Total energy is given by:

$$E(\pi) = \sum_i \int_0^{T(\pi)} P_i(t) dt.$$

Assume each device exhibits a non-zero baseline power draw when active, in addition to a dynamic component dependent on utilization, frequency, and voltage. Compare two policies: π_1 executes W on a single GPU at moderate frequency; π_2 splits W across two GPUs, increasing frequency and parallelism to minimize completion time. It is possible that π_2 completes faster while consuming more total energy. This establishes that time-based metrics fail to preserve energy ordering across resource allocation policies.

4 Empirical Witness

We demonstrate this phenomenon on an NVIDIA RTX 3070 Laptop GPU. A fixed sequence of FP32 matrix multiplications is executed with all data resident on the device. Computational work and numerical precision are held constant. Execution policies differ only in temporal scheduling. Continuous execution minimizes runtime but consistently consumes more total energy than duty-cycled alternatives.

5 Discussion

Work-Preserving vs. Work-Reducing Optimizations. The ordering failure identified in this work is specific to work-preserving optimizations, which accelerate execution without reducing total computational work. In contrast, work-reducing optimizations—such as algorithmic improvements, pruning, caching, or early exit strategies—often align time and energy objectives. Our result cautions against extrapolating this intuition to work-preserving execution policies.

Time-Based Metrics as Information-Losing Projections. When power consumption depends on execution policy, time-based metrics become information-losing projections. In this setting, asking which policy minimizes execution time is no longer a well-defined research question, as time fails to induce a consistent ordering over feasible executions.

Condition for Time–Energy Equivalence. Time-based optimization preserves energy ordering only under the restrictive condition that average power is a deterministic function of execution time, independent of policy. Modern systems with DVFS, utilization-dependent boost states, and resource allocation choices violate this condition.

6 Limitations

This study does not characterize how frequently such inversions occur, nor does it evaluate specific schedulers or optimization frameworks. Results are limited to a single platform and workload. The multi-GPU argument is theoretical and intended to illustrate generality of the ordering failure, not to claim empirical prevalence.

7 Conclusion

Execution time is not an information-preserving proxy for energy consumption once power depends on execution policy. For work-preserving executions on modern hardware, time-only metrics can induce inconsistent ordering under energy or power constraints, leading to systematically incorrect comparisons between execution policies. The implication is not that execution time is irrelevant, but that optimizing for time in isolation no longer defines a well-posed objective when energy is a binding constraint. In such settings, research questions framed solely around minimizing execution time fail to align with the physical quantities that actually limit system behavior. Rather than proposing a specific alternative metric, this work argues for a minimal reformulation of evaluation practice in energy-constrained regimes: execution time should be treated as a secondary variable, conditioned on explicit energy or power budgets. How to design metrics and evaluation protocols that reflect this constraint remains an open problem.

References

No external references are cited.