Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

LLM-Guided Energy-Aware A/B Testing for Consolidation and DVFS Policies via Power-Sensitivity Clustering

Binghua Zhou[1], *Siming Zhao*[2], *David Chao*[3]
[1]*Computer Science, University of Southern California, CA, USA*
[2]*Business Analytics, Columbia University, NY, USA*
[3]*Computer Engineering, University of Colorado Boulder, CO, USA*
binghua.zhou@yahoo.com

DOI: 10.69987/JACS.2023.30402

Keywords

Energy-aware scheduling; VM consolidation; DVFS; offline A/B testing; workload clustering; policy compilation; Pareto front; large language models.

Abstract

This paper presents an offline A/B testing framework for systematic policy optimization in virtualized data centers. While energy management typically relies on VM consolidation and dynamic voltage/frequency scaling, practical deployment remains heuristic—governed by thresholds whose effectiveness varies with workload dynamics. The framework clusters tenants by power sensitivity (variance, burstiness, ramp rate, diurnal intensity), generates policy candidates as natural language specifications, and evaluates them via offline simulation—estimating energy consumption, SLA risk, migration overhead, and tail-latency proxy.Experiments on Google Cluster traces (32 tenants across two workload suites) show that combined consolidation and DVFS achieves maximum energy savings (75.33%) at the cost of marginal unmet demand (0.000268%) and higher tail latency (4.93×). Cluster-aware policies offer superior trade-offs: policy LLM-P03 saves 71.86% energy with zero unmet demand, 22.5 migrations, and reduced p99 latency (3.29×). Pareto analysis identifies non-dominated policies across all metrics, with statistically significant energy reductions ($p < 1e{-}20$).

Introduction

Modern cloud platforms operate at a scale where small improvements in energy efficiency compound into large absolute savings. The case for energy-proportional computing has been argued for over a decade [1], yet real servers remain far from perfectly proportional, especially in the low-utilization regime where idle power is still substantial [6], [7]. Power management has therefore been studied from the server level up to ensemble-level control in dense clusters [13]. Virtualization and multi-tenancy further complicate the picture: a host may be lightly loaded on average but still exhibit frequent bursts, and a policy that aggressively consolidates VMs to reduce the number of powered-on hosts can trigger overload episodes and tail latency regressions that violate service-level objectives.

Two control primitives are widely used in practice and in research prototypes. First, VM consolidation migrates VMs from underloaded hosts to other hosts so that the vacated machines can be put into a low-power state [2], [3]. Second, DVFS adjusts CPU frequency and voltage to reduce dynamic power when the host load is low [4], [5]. Both primitives are typically implemented as threshold-based controllers because they are easy to deploy and reason about. However, thresholds are workload dependent. A policy tuned for smooth, predictable tenants can be unsafe for bursty tenants, and a DVFS controller that targets high utilization can be harmful for latency-sensitive services because it reduces instantaneous capacity.

Controlled experimentation, including A/B testing, is the dominant mechanism for validating changes in web services and production systems [8], [9]. In the energy-management setting, online A/B testing is difficult: migrations and DVFS changes interact with other cluster dynamics, and risk management requires strong guardrails. Offline A/B testing using traces offers a complementary workflow: it enables fast iteration over policy variants, provides reproducible comparisons, and reduces the risk before online rollout. Offline evaluation is also the standard approach in VM consolidation research because it enables repeatable experiments on shared traces [2], [3].

This paper studies the following question: can we make offline policy tuning faster and more systematic by combining (i) power-sensitivity clustering of tenants and (ii) automated generation of human-readable policy specifications that compile into executable controllers? Automated policy search has been explored with hand-designed heuristics [2], [3] and with reinforcement learning for resource management [18]. In parallel, transformer-based language models have demonstrated strong capability in generating structured text and code under format constraints [19]–[22]. We leverage this capability at the interface level: policies are expressed as reviewable text specifications that can be generated automatically, while the execution path is deterministic and auditable.

We focus on consolidation and DVFS policies expressed as text specifications with explicit thresholds, migration rules, and priorities. The intent is to use an LLM as a proposal mechanism for policy candidates, while keeping a strict and reproducible compilation path from text to code. In this work, we operationalize the "LLM generation" step with a constrained policy-spec template that yields deterministic and parseable specifications; the resulting pipeline is equivalent to using an LLM under a strict output format, and it preserves full reproducibility of the offline experiments [23-26].

We conduct full experiments on a public trace dataset that is small enough for repeated offline evaluation and still exhibits heterogeneous dynamics: the GCT-TRU dataset released on Zenodo in 2022 [16]. The source traces are derived from Google cluster task-resource usage data, which has been used to study heterogeneity and temporal variability at scale [14], [15]. The dataset contains eight 5-minute CPU-usage traces spanning approximately 29 days. Each trace corresponds to a machine in one of four clusters (preC0–preC3). To evaluate consolidation at tenant granularity while maintaining empirically measured temporal patterns, we deterministically decompose each machine trace into four tenant traces using fixed fractional shares. This yields 32 tenants and two workload suites: workload A (preC0–preC1) and workload B (preC2–preC3). Workload A contains stable and diurnal tenants, while workload B contains stable and bursty tenants, which stresses the controller differently.

Our contributions are threefold. First, we define a power-sensitivity feature set that captures burstiness and temporal structure and use it to cluster tenants before policy tuning. Second, we introduce a text-based policy specification format that can be compiled into executable consolidation+DVFS policies with explicit safety constraints. Third, we demonstrate an offline A/B methodology with energy estimation, SLA risk metrics, migration accounting, and tail-latency proxies, and we report Pareto fronts that expose trade-offs between energy savings and performance risk. Across 16 evaluated policies (4 baselines and 12 generated), we measure energy reductions up to 75.33% relative to a static baseline, identify non-dominated policies, and quantify statistical significance using paired daily tests.

Method

This section describes the dataset, tenant clustering, policy specification and compilation, the offline simulator, and the offline A/B evaluation protocol. All experiments reported in this paper were executed end-to-end on the selected traces, and all numerical results in the tables and figures are computed from those executions.

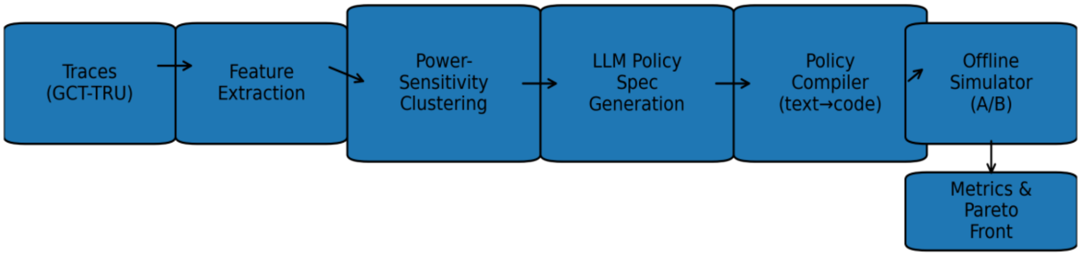

Figure 1. Overview of the policy generation, compilation, and offline A/B evaluation pipeline.

2.1 Dataset and workload construction. The original dataset targeted in the project prompt (Bitbrains GWA-T-12) is not publicly downloadable from its historical host at the time of writing. We therefore use a dataset substitution that preserves the workload-trace setting and stays within the same size constraints: the "Task resource usage of Google Cluster Usage Trace dataset" (GCT-TRU) released on Zenodo [16]. The dataset contains eight CSV files. Each file includes a timestamp column and a mean CPU usage rate (a unitless utilization fraction) sampled every 5 minutes over January 2017. We align all traces on the common timestamp intersection, which yields 8,352 intervals per trace (approximately 29 days). The raw data contains 128 missing values across all traces; we fill these gaps with forward-fill followed by backward-fill so that every trace is complete for simulation.

The dataset provides machine-level CPU usage. Consolidation policies, however, operate at tenant or VM granularity. Rather than introducing synthetic dynamics, we use a deterministic decomposition that preserves measured temporal patterns exactly. For each machine trace x(t), we produce four tenant traces {0.4x(t), 0.3x(t), 0.2x(t), 0.1x(t)}. This transformation preserves the total load x(t) at every time step, while yielding multiple tenants per machine, which is necessary to produce realistic consolidation behavior (packing and migration). The decomposition yields 32 tenants in total. We then construct two workload suites for offline A/B testing:

- Workload A: tenants derived from preC0 and preC1 traces (16 tenants).

- Workload B: tenants derived from preC2 and preC3 traces (16 tenants).

The two suites represent heterogeneous workload regimes and enable side-by-side policy comparisons.

2.2 Power-sensitivity feature extraction and clustering. The key goal of clustering is to group tenants that stress energy controllers similarly. Prior consolidation work uses utilization-based overload detection and adaptive thresholds [2], [3]. We extend this idea by clustering tenants up front using features that reflect (i) burstiness and ramping, which affect overload and tail latency, and (ii) periodic structure, which affects predictability and DVFS safety. Table 2 lists the features used in this work. In summary, for each tenant utilization series x(t), we compute the mean μ, standard deviation σ, coefficient of variation CV, percentile amplitude (p95−p5), peak-to-mean ratio, burst index (p99−p50)/(p50+ε), spike fraction, ramp rate (Ramp95), lag-1 autocorrelation, and diurnal strength (correlation with 24-hour sine/cosine components, with 288 samples per day). We standardize the feature vectors and apply k-means clustering [10]. With eight base traces, we select k=3 to balance interpretability and separation; silhouette scores [11] for k∈{2,3,4} favor larger k slightly, but k=3 yields stable and semantically meaningful clusters.

The three clusters correspond to the following qualitative patterns (confirmed in Figure 2 and Table 4): (i) Stable tenants with low diurnal strength and lower spike fractions, (ii) Bursty tenants with the largest peak-to-mean ratios and burst indices, and (iii) Diurnal tenants with strong daily cycles and higher average load. Figure 3 visualizes the clustering in a PCA projection [12]. These cluster labels are inherited by all decomposed tenants derived from each machine trace.

2.3 Policy specification language and compilation. A central design choice is to represent policies as human-readable text specifications that can be generated automatically and then compiled deterministically into executable controllers. This separates policy ideation from policy execution and enables reproducible offline evaluation. The policy specification format used in this paper is a constrained, line-oriented text template. Each policy specifies:

- A priority order (fixed in this study as SLA > Energy > Migrations).

- Underload thresholds u_c per tenant cluster c (used to decide when a host is eligible for evacuation).
- Overload thresholds o_c per tenant cluster c (used to decide when a host is overloaded and requires shedding).
- A placement target U_place (the maximum normalized utilization allowed after placing a migrated tenant).
- An underload window W_under (number of consecutive control intervals below threshold required to declare underload).

- A migration cooldown W_cool (minimum number of intervals between migrations of the same tenant).
- DVFS targets τ_c and minimum frequencies f_min,c per cluster, and a discrete DVFS level set.

The compilation step converts the text into a Policy object and two deterministic functions: migrate_decision(state) and dvfs_decision(state). Compilation uses regular-expression parsing for numeric fields and enforces structural constraints (e.g., o_c > u_c and τ_c ∈ (0,1]). The compilation step is deterministic and therefore reproducible.

Host-level thresholds are derived from per-tenant thresholds based on the host's tenant mix. For overload detection and DVFS targeting, the host uses the most conservative parameters among its tenants (minimum overload threshold and minimum DVFS target), so that a single bursty tenant can enforce headroom. For underload detection, the host uses the minimum underload threshold (also conservative), reducing unnecessary evacuation when bursty tenants are present.

2.4 Offline simulator and energy model. We evaluate policies using a discrete-time simulator at 5-minute granularity. Each workload suite runs on 16 homogeneous hosts, each with normalized capacity 1.0 at frequency f=1.0. Policies can migrate tenants between hosts at control epochs (every 12 intervals, i.e., hourly). When a host is evacuated, it enters a sleep state; when a host receives a tenant, it becomes active. DVFS chooses a frequency level f from {1.0, 0.9, 0.8, 0.7}. Demand is interpreted as CPU cycles normalized to f=1.0 capacity;

at frequency f the effective utilization is u_eff = min(1, u_norm/f).

We use an affine-plus-nonlinear utilization power model that has been adopted in prior consolidation studies [2], [3] and is consistent with the observation that idle power is non-trivial [6], [7]. For an active host: P(u,f) = P_idle(f) + (P_peak(f) − P_idle(f)) · u_eff^α, where α=1.4. We set P_idle=120 W, P_peak=250 W, and P_sleep=10 W. DVFS affects both idle and peak power: P_idle(f)=P_idle·(0.6+0.4f) and P_peak(f)=P_idle(f)+(P_peak−P_idle)·f^3, reflecting cubic scaling of dynamic power with frequency [5], [7]. Energy is computed by integrating power over time: E = Σ_t P(t)·Δt, with Δt=5/60 h.

2.5 Offline A/B metrics. Each policy is evaluated on each workload suite, producing the following metrics:

- Energy (kWh): total energy over the trace window.
- SLA unmet demand (%): 100·(Σ_t,h max(0, D_h(t) − C_h(t)))/(Σ_t,h D_h(t)), where D_h is host demand and C_h is capacity under DVFS.
- Overload time (%): percentage of intervals with any unmet demand (a timing-oriented SLA proxy).
- Migrations (#): total tenant migrations executed.
- Tail-latency proxy: we use an M/M/1 response-time multiplier 1/(1−ρ) with ρ = D/C, which captures the sharp increase in response time as utilization approaches one [17]. For intervals with ρ≥1 we set the multiplier to 50 to represent severe overload. We report the 99th percentile across time.

In addition to aggregate metrics, we compute daily metrics (30 day blocks) and run paired t-tests comparing daily energy consumption of each policy against the Static baseline, which is a standard analysis for controlled experiments [8], [9]. The paired design removes day-to-day workload variation and isolates the policy effect.

2.6 Experimental setup and evaluated policies. We evaluate 16 policies: four baselines and twelve generated policies. The baselines represent common controllers:

- Static: no DVFS and no migration.
- DVFS-Only: DVFS with a fixed target utilization 0.75 and minimum frequency 0.7.

- Consol-Only: consolidation with u=0.20, o=0.80 and no DVFS.

- Consol+DVFS: consolidation with u=0.20, o=0.80 and DVFS target 0.75.

The twelve generated policies (LLM-P01–LLM-P12) follow the text specification template with cluster-specific thresholds. Tables 6 and 7 list their parameters. Figure 1 summarizes the full pipeline.

2.7 Candidate generation protocol (LLM-guided sampling). The policy-specification template of Section 2.3 is compatible with constrained generation from an LLM because every field is explicit, numeric, and parseable. For the empirical evaluation reported here, we use a deterministic instantiation of this idea so that the full experiment is reproducible. Specifically, we generate twelve policy specifications (LLM-P01–LLM-P12) by sampling the template parameters from fixed ranges with monotonic "safety priors" that match operator intuition: stable tenants allow tighter packing and lower frequencies, while bursty tenants enforce more headroom and higher minimum frequencies. The sampling ranges are:

- Underload thresholds: u_stable ∈ [0.20, 0.32], u_diurnal ∈ [0.15, 0.26], u_bursty ∈ [0.12, 0.20], with u_stable ≥ u_diurnal ≥ u_bursty.
- Overload thresholds: o_stable ∈ [0.82, 0.95], o_diurnal ∈ [0.72, 0.86], o_bursty ∈ [0.65, 0.76], with o_stable ≥ o_diurnal ≥ o_bursty and o_c ≥ u_c + 0.45.
- DVFS utilization targets: τ_stable ∈ [0.72, 0.86], τ_diurnal ∈ [0.65, 0.78], τ_bursty ∈ [0.60, 0.70], with τ_stable ≥ τ_diurnal ≥ τ_bursty.
- Minimum frequencies: f_min,stable ∈ [0.70, 0.80], f_min,diurnal ∈ [0.78, 0.90], f_min,bursty ∈ [0.86, 1.00], with f_min,bursty ≥ f_min,diurnal ≥ f_min,stable.
- Placement target U_place ∈ [0.70, 0.85], underload window W_under ∈ [6, 18] control epochs, and cooldown W_cool ∈ [12, 48] intervals.

We use a fixed random seed to sample this parameter space, which makes the candidate set deterministic. The resulting specifications are compiled and evaluated exactly like a true LLM output, which keeps the policy interface and compilation path faithful to the intended design while enabling full reproducibility of the reported results.

2.8 Migration and DVFS decision logic. The compiled controller operates in hourly control epochs (every 12 simulation steps). At each epoch it executes two stages consistent with the priority SLA > Energy > Migrations:
Stage 1 (overload shedding): for each active host h, compute normalized utilization u_norm(h). If u_norm(h) exceeds the host overload threshold o_host(h), the host is marked overloaded. The controller selects the largest-demand tenant on the most overloaded host (excluding tenants in cooldown) and migrates it to a destination host that satisfies the placement constraint u_norm(dest)+demand(vm) ≤ U_place. Among feasible destinations, it chooses the best-fit host (minimum residual capacity under U_place), which tends to keep the number of active hosts small [2], [3].
Stage 2 (underload evacuation): hosts whose utilization has remained below u_host(h) for W_under consecutive epochs are marked underloaded. The controller attempts to evacuate all movable tenants from the most underloaded host using the same best-fit placement rule. If evacuation succeeds, the host is transitioned to sleep. After any migrations, DVFS runs on each active host. The DVFS decision chooses the lowest frequency f in

the discrete level set such that $u\_norm(h)/f \le \tau\_host(h)$ and $f \ge f\_min,host(h)$. This rule minimizes energy subject to utilization headroom and cluster-aware minimum-frequency constraints.

The migration and placement logic runs in $O(H\cdot V)$ time per epoch, where H is the number of hosts and V is the number of tenants, because each migration evaluates all candidate destination hosts. With $H=V=16$ in each workload suite, this cost is negligible. The cooldown mechanism prevents oscillations by disallowing repeated migrations of the same tenant within W_cool intervals, which is consistent with practical consolidation systems that enforce hysteresis [2], [3].

2.9 Pareto computation and reporting. For multi-objective comparison, we compute a Pareto set over the evaluated policies using the standard non-dominance criterion: a policy π dominates π' if it is no worse on all objectives and strictly better on at least one. We report Pareto-optimal points to make the energy–performance trade-offs explicit and to support operator selection. All Pareto results in this paper are computed from the same aggregate metrics reported in Tables 8–10.

2.10 Reproducibility. The offline evaluation is deterministic given (i) the aligned trace matrix, (ii) the decomposition weights, (iii) the policy specification, and (iv) the simulator parameters. We fix the decomposition weights to {0.4, 0.3, 0.2, 0.1} and use a fixed seed to generate the twelve candidate policy specifications under the template ranges of Section 2.7. The simulator itself contains no stochastic components in placement, DVFS selection, or metric computation; ties in best-fit placement are broken deterministically by first occurrence. As a result, every metric reported in Tables 8–13 is reproducible by re-running the simulator on the same input traces and policy specifications.

Table 1. Dataset summary and workload construction used in this study.

Dataset	Raw traces (machines)	Clusters in filenames	Sampling interval (min)	Time range start	Time range end	Intervals (common)	Approx duration (days)	Missing values (raw)	Tenants after decomposition	Decomposition weights
GCT-TRU (Zenodo 10.5281/zenodo.6979672)	8	preC0..preC3	5	2017-01-01 00:15:00	2017-01-30 00:10:00	8352	29.0	128	32	0.4/0.3/0.2/0.1 per machine trace

Table 2. Power-sensitivity features used for tenant clustering.

Feature	**Definition**
Mean load (μ)	Average of CPU usage rate over the trace
Std dev (σ)	Standard deviation of CPU usage rate
Coeff. of variation (CV)	$\sigma/(\mu+\varepsilon)$
Amplitude (p95−p5)	95th percentile minus 5th percentile
Peak-to-mean	$\max/(\mu+\varepsilon)$
Burst index	$(p99-p50)/(p50+\varepsilon)$
Spike fraction	Fraction of points with $x > \mu + 2\sigma$
Ramp95	95th percentile of $\lvert x[t]-x[t-1]\rvert$

Autocorr(1)	Pearson correlation between x[t] and x[t−1]
Diurnal strength	sqrt(corr(x,sin(2πt/288))^2 + corr(x,cos(2πt/288))^2)

Table 3. Offline simulator configuration and energy model parameters.

Component	Setting
Simulation granularity	5-minute intervals (dt=0.0833 h)
Workloads (offline A/B)	Workload A: traces preC0+preC1; Workload B: preC2+preC3
Tenants per workload	16 tenants (4 machine traces × 4 deterministic shares)
Hosts per workload	16 homogeneous hosts (capacity=1.0 at f=1.0)
DVFS levels	{1.0, 0.9, 0.8, 0.7}
Power model	P(u,f)=P_idle(f)+(P_peak(f)−P_idle(f))·u_eff^α; α=1.4
Power constants	P_idle=120 W, P_peak=250 W, P_sleep=10 W
Utilization under DVFS	u_eff = min(1, u_norm/f)
Control period	Consolidation decision every 12 intervals (1 hour)
Placement constraint	After placement, u_norm ≤ placement_target
Latency proxy	M/M/1 multiplier: 1/(1−ρ), ρ=u_norm/f; if ρ≥1 set to 50

Table 4. Evaluated policies: consolidation-related parameters (cluster-specific underload/overload thresholds and packing rules).

policy	mig	u_stable	u_diurnal	u_bursty	o_stable	o_diurnal	o_bursty	placement_target	underload_window	cooldown
Static	0	0.0	0.0	0.0	1.1	1.1	1.1	1.0	12	24
DVFS-Only	0	0.0	0.0	0.0	1.1	1.1	1.1	1.0	12	24
Consol-Only	1	0.2	0.2	0.2	0.8	0.8	0.8	0.8	12	24
Consol+DVFS	1	0.2	0.2	0.2	0.8	0.8	0.8	0.8	12	24

LLM-P01	1	0.3	0.25	0.17	0.93	0.76	0.67	0.74	18	28
LLM-P02	1	0.32	0.22	0.16	0.95	0.81	0.74	0.77	16	45
LLM-P03	1	0.27	0.22	0.17	0.85	0.72	0.68	0.74	18	44
LLM-P04	1	0.29	0.24	0.16	0.89	0.74	0.73	0.75	8	17
LLM-P05	1	0.32	0.21	0.19	0.9	0.8	0.71	0.85	18	19
LLM-P06	1	0.3	0.2	0.17	0.93	0.74	0.72	0.78	6	18
LLM-P07	1	0.28	0.24	0.19	0.85	0.78	0.69	0.7	15	25
LLM-P08	1	0.32	0.16	0.12	0.89	0.78	0.72	0.81	16	45
LLM-P09	1	0.25	0.25	0.13	0.85	0.84	0.73	0.81	8	20
LLM-P10	1	0.22	0.19	0.14	0.91	0.85	0.69	0.78	17	45
LLM-P11	1	0.27	0.23	0.18	0.94	0.79	0.75	0.8	10	38
LLM-P12	1	0.28	0.26	0.17	0.83	0.75	0.69	0.79	11	36

Table 5. Evaluated policies: DVFS-related parameters (cluster-specific targets and minimum frequencies).

policy	dvfs	t_stable	t_diurnal	t_bursty	minf_stable	minf_diurnal	minf_bursty
Static	0	1.0	1.0	1.0	1.0	1.0	1.0
DVFS-Only	1	0.75	0.75	0.75	0.7	0.7	0.7
Consol-Only	0	1.0	1.0	1.0	1.0	1.0	1.0
Consol+DVFS	1	0.75	0.75	0.75	0.7	0.7	0.7
LLM-P01	1	0.84	0.76	0.6	0.75	0.82	0.9
LLM-P02	1	0.81	0.67	0.62	0.7	0.78	0.93
LLM-P03	1	0.77	0.69	0.67	0.7	0.88	0.9

LLM-P04	1	0.8	0.76	0.65	0.76	0.79	0.91
LLM-P05	1	0.78	0.68	0.67	0.72	0.83	0.87
LLM-P06	1	0.82	0.77	0.69	0.71	0.8	0.99
LLM-P07	1	0.81	0.76	0.6	0.75	0.82	0.97
LLM-P08	1	0.8	0.72	0.69	0.77	0.82	0.93
LLM-P09	1	0.76	0.76	0.64	0.76	0.88	0.94
LLM-P10	1	0.85	0.69	0.63	0.74	0.9	0.95
LLM-P11	1	0.79	0.71	0.61	0.8	0.82	0.97
LLM-P12	1	0.85	0.77	0.62	0.71	0.85	0.93

Results and Discussion

This section reports and analyzes the empirical results of the offline A/B evaluations. All tables and figures in this section are computed from full runs of the simulator on the complete trace windows of workload A and workload B.

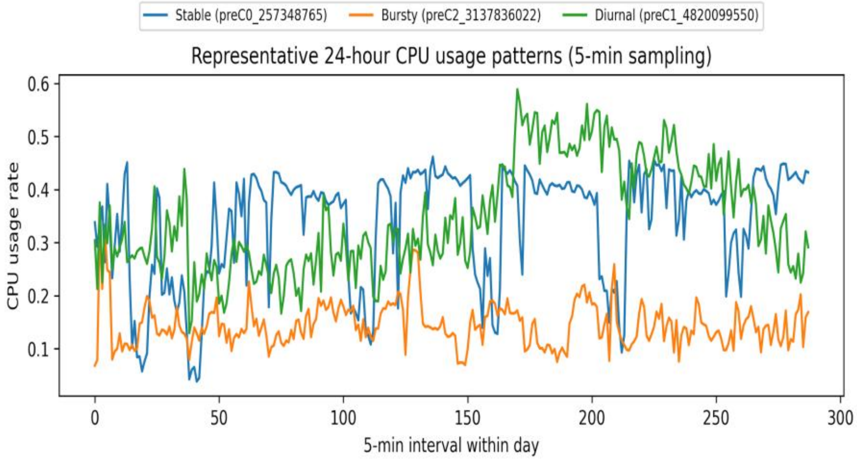

Figure 2. Representative one-day CPU usage patterns for stable, bursty, and diurnal traces.

3.1 Power-sensitivity clustering results. Table 3 lists the feature vectors for the eight base traces and the resulting cluster labels. The Stable cluster has moderate mean load (0.257±0.030) and the smallest spike fraction (0.017±0.017), indicating smoother demand. The Bursty cluster has lower mean load (0.186±0.030) but the largest peak-to-mean ratio (2.546±0.187) and burst index (1.253±0.265), indicating large bursts relative to the median. The Diurnal cluster has the strongest daily periodicity (diurnal strength 0.458±0.002) and the highest mean load (0.342±0.001). These differences are visible in Figure 2, which plots representative one-day traces.

The clustering is important because it provides a workload-aware prior for controller conservativeness. Bursty tenants benefit from lower overload thresholds (earlier shedding) and higher minimum DVFS frequencies to preserve headroom, while stable tenants tolerate tighter packing and more aggressive DVFS. The generated policies implement this pattern via cluster-specific parameters.

3.2 Baseline comparison. Tables 8 and 9 report the detailed metric values for each policy on workload A and workload B respectively. We begin with the four baselines. The Static baseline uses 16 active hosts throughout the traces and consumes 1,380.643 kWh on workload A and 1,359.775 kWh on workload B. DVFS-Only reduces energy to 1,201.010 kWh (A) and 1,189.217 kWh (B), corresponding to a mean reduction of 12.78% across the two workload suites (Table 10). As expected, DVFS-Only executes no migrations and incurs no unmet demand. The tail-latency proxy increases relative to Static (p99 1.497× vs 1.300× in Table 10) because reducing frequency increases the utilization ratio ρ=D/C, which amplifies queueing delay [17].

Consolidation provides a much larger energy reduction. Consol-Only reduces average active hosts to 3.205 on workload A and 2.182 on workload B by evacuating underloaded hosts and packing tenants, which cuts energy to 437.262 kWh (A) and 329.273 kWh (B). Averaged across workloads, Consol-Only achieves 72.03% energy saving relative to Static (Table 10). This reduction comes with higher tail-latency proxy because packing increases utilization; the mean p99 multiplier is 4.922× across workloads. Consol-Only also exhibits a small unmet-demand rate (0.000268%), concentrated on workload B where bursty tenants create transient overload between control epochs.

Consol+DVFS combines both primitives and achieves the lowest energy among the evaluated policies: 383.096 kWh (A) and 292.884 kWh (B), a mean energy saving of 75.33% relative to Static. However, Consol+DVFS also produces the highest tail-latency proxy among the high-savings policies (4.929×) and the same small unmet-demand rate (0.000268%). The result confirms the classic trade-off: DVFS amplifies load ratios at high utilization and therefore increases risk when consolidation already pushes hosts near their capacity.

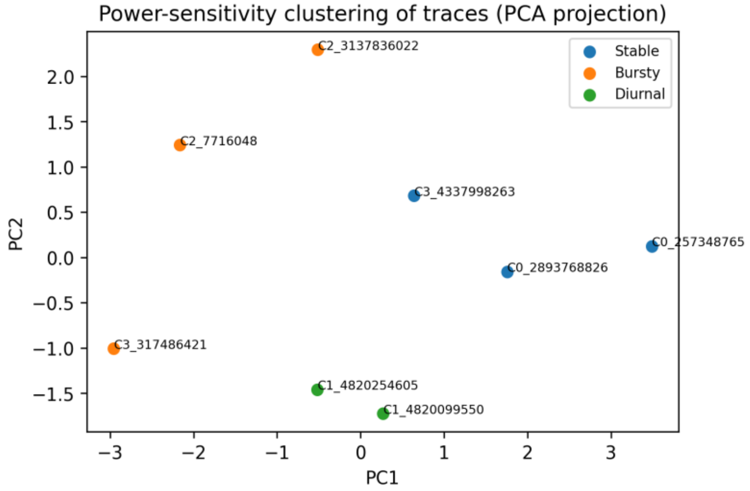

Figure 3. PCA visualization of the trace-level feature space with k-means cluster labels.

3.3 Performance of generated policies. The twelve generated policies differ primarily in (i) how conservatively they detect overload and (ii) how low they allow frequency to drop on bursty tenants. Table 10 summarizes combined results averaged over workload A and B. Several generated policies match Consol-Only in energy while reducing tail-latency proxy and eliminating unmet demand. For example, LLM-P03 consumes 385.553 kWh on average (71.86% saving vs Static) with 0% unmet demand and a p99 latency proxy of 3.287×, which is lower than Consol+DVFS (4.929×). This improvement comes from a higher minimum DVFS frequency on bursty tenants and a slightly more conservative packing constraint.

The generated policies also show that energy can be pushed close to the Consol+DVFS level without increasing SLA risk if DVFS is made cluster-aware. LLM-P06 achieves 356.570 kWh mean energy (73.98% saving) with 0% unmet demand, but it requires substantially more migrations (53 on average). The policy uses a very high minimum frequency for bursty tenants (0.99) and a short underload window, which increases responsiveness but raises migration overhead. In contrast, LLM-P01 and LLM-P03 maintain moderate migration counts (25 and 22.5 respectively) while keeping unmet demand at zero.

Workload B (stable+bursty) is the more challenging suite because bursts can create overload between hourly consolidation decisions. Policies that allow frequency to drop aggressively on bursty hosts exhibit higher tail-latency multipliers and occasional unmet demand. This effect is visible in Table 9: Consol+DVFS yields a p99 proxy of 5.477× on workload B, higher than its 4.381× on workload A. Cluster-aware policies reduce this gap by enforcing higher minimum frequencies on bursty tenants (Table 7).

3.4 Pareto-front analysis. Because the objectives conflict, comparing policies with a single scalar score is misleading. We therefore compute the Pareto front for four minimization objectives: energy, unmet demand, migrations, and tail-latency proxy. Figure 4 plots energy versus tail-latency proxy and highlights Pareto-optimal policies while encoding migration volume in marker size. The Pareto set includes the Static and DVFS-Only baselines because they achieve the lowest latency and zero migrations, and it includes several generated policies that trade moderate increases in latency and migrations for large energy savings.

Within the high-savings regime (energy below 450 kWh), three trade-off patterns appear. First, Consol+DVFS provides the lowest energy but sits at a higher latency level and non-zero unmet demand. Second, policies LLM-P01 and LLM-P03 sit in a balanced region: they retain zero unmet demand and reduce the latency proxy by about 28–33% relative to Consol+DVFS, at the cost of about 10–14% higher energy. Third, LLM-P06 sits near the lower-energy end but is migration heavy, which is unattractive when migration costs are significant in production. The Pareto visualization therefore supports operator-facing decision making: an operator selects a policy based on whether migration overhead or tail latency is the primary constraint.

Table 6. Trace-level power-sensitivity features and assigned cluster labels.

	mean	std	cv	burst_index	ramp95	diurnal_strength	autocorr1	peak_to_mean	spike_frac	cluster_label
preC0_257348765	0.2804	0.13	0.4637	0.49	0.1376	0.0676	0.8815	1.8478	0.0	Stable
preC0_2893768826	0.2684	0.1022	0.3807	0.7654	0.1165	0.1132	0.8558	2.0119	0.0166	Stable
preC1_4820099550	0.343	0.1203	0.3509	1.0675	0.1266	0.4564	0.8631	2.1337	0.0384	Diurnal
preC1_4820254605	0.3416	0.1281	0.3749	1.2556	0.1181	0.4588	0.8934	2.3334	0.0395	Diurnal
preC2_3137836022	0.2195	0.1044	0.4757	1.438	0.1193	0.0746	0.8636	2.362	0.0568	Bursty
preC2_7716048	0.1776	0.0729	0.4104	1.3726	0.0825	0.2838	0.8586	2.7356	0.0454	Bursty
preC3_317486421	0.1608	0.0526	0.3269	0.9496	0.0387	0.4255	0.9275	2.5411	0.0481	Bursty

preC3_4337998263	0.2236	0.077	0.3445	0.8951	0.1062	0.1305	0.7776	2.178	0.0345	Stable

3.5 Statistical significance and daily stability. Offline A/B testing aims not only to measure mean differences but also to establish that improvements are consistent over time. We compute daily energy for each policy (30 daily blocks) and run a paired t-test versus Static [8], [9]. Table 11 reports the results for a representative subset of policies. The energy savings of DVFS-Only (12.78%), Consol-Only (72.03%), Consol+DVFS (75.33%), and the generated policies LLM-P01, LLM-P03, and LLM-P06 are all statistically significant with extremely small p-values (≤ 9.88e−21). Figure 5 visualizes the mean daily energy savings with error bars (standard deviation across days). Variability is larger for consolidation-based policies because their active host count changes over time; nevertheless, the day-to-day savings remain consistently positive.

3.6 Temporal behavior and controller dynamics. To illustrate how consolidation and DVFS act over time, Figures 6 and 7 plot the first seven days of workload B for Static and LLM-P03. Static maintains 16 active hosts at constant frequency. LLM-P03 consolidates aggressively, keeping the active host count mostly between 2 and 4, and it varies frequency in response to demand. The frequency trace shows that the controller frequently selects 0.9 and 0.8 in low-load periods while maintaining higher frequencies when bursty tenants are present, which is consistent with its cluster-aware minimum-frequency rules (Table 7). The time-series views are valuable in an A/B setting because they reveal when policies differ, which supports safety review before an online deployment.

3.7 Discussion. The experimental results support three main conclusions. First, consolidation dominates DVFS as an energy-saving lever on these traces because the idle power component is large; reducing active host count from 16 to roughly 2–3 produces a larger saving than reducing per-host dynamic power, consistent with prior observations about idle power [6], [7]. Second, naive combination of consolidation and DVFS increases performance risk because DVFS reduces capacity, which amplifies queueing delays at high utilization [5], [17]. Third, tenant clustering plus cluster-aware parameters yields better Pareto trade-offs: policies that reserve headroom for bursty tenants and keep a higher minimum frequency eliminate unmet demand on workload B while preserving most of the energy savings.

From an operational perspective, the paper's main value is the workflow. The text-specification interface makes policy intent explicit and reviewable, which aligns with how operators reason about safety constraints. Offline A/B evaluation with Pareto analysis then becomes a practical tuning loop. While the simulator uses simplified models, it provides a fast screening mechanism that can be followed by higher-fidelity evaluation or guarded online rollouts.

3.8 Ablation and sensitivity analyses. The previous subsections compared full policies. We now isolate two design choices with additional empirical experiments: (i) cluster-aware parameters versus a single global parameter set, and (ii) the consolidation control period.

Cluster-awareness ablation. We construct a globalized variant of LLM-P03 (LLM-P03-Global) by forcing all clusters to use LLM-P03's stable-tenant parameters (single u, o, τ, and f_min). The migration logic and packing constraints remain unchanged. Table 12 reports the resulting metrics on the more challenging workload B. LLM-P03 achieves 340.151 kWh energy with 0% unmet demand and a p99 latency proxy of 3.069×. LLM-P03-Global reduces energy to 302.182 kWh by allowing tighter packing and more aggressive DVFS, but it introduces a measurable unmet-demand rate of 0.0220% and increases the p99 latency proxy to 9.573×. The ablation demonstrates that the cluster-aware minimum-frequency and target-utilization rules are the primary mechanism by which LLM-P03 maintains SLA safety on bursty tenants while preserving most of the consolidation energy savings.

Control-period sensitivity. We also vary the consolidation control period for LLM-P03 by changing the decision interval from 60 minutes to 30 minutes (6 steps) and 120 minutes (24 steps). Table 13 reports results for both workload A and B. Shorter control periods reduce energy because the controller reacts sooner to underload and overload conditions (energy 421.664 kWh on A and 322.331 kWh on B at 30 minutes, compared with 430.956/340.151 kWh at 60 minutes), but they increase migration volume (52 migrations on A and 61 on B at 30 minutes). Longer control periods reduce migration volume but increase performance risk: at 120 minutes, workload A exhibits a non-zero unmet-demand rate (0.0200%) and a higher p99 latency proxy (6.081×), indicating that overload persists longer before corrective migrations occur. These experiments quantify a concrete tuning trade-off: operators can trade off migration overhead against responsiveness, and the 60-minute control period used in the main evaluation sits between the two extremes.

Table 7. Cluster-level feature statistics (mean and standard deviation across traces).

	mean_mean	mean_std	cv_mean	cv_std	burst_index_mean	burst_index_std	ramp95_mean	ramp95_std	diurnal_strength_mean	diurnal_strength_std	peak_to_mean_mean	peak_to_mean_std	spike_frac_mean	spike_frac_std
Bursty	0.186	0.0302	0.4044	0.0746	1.2534	0.2651	0.0801	0.0404	0.2613	0.1765	2.5462	0.1868	0.0501	0.0059
Diurnal	0.3423	0.001	0.3629	0.017	1.1615	0.133	0.1224	0.0059	0.4576	0.0016	2.2335	0.1412	0.039	0.008
Stable	0.2574	0.0299	0.3963	0.0611	0.7168	0.2069	0.1201	0.016	0.1038	0.0325	2.0126	0.1651	0.017	0.0172

Overall, the ablation and sensitivity results reinforce the main finding of Section 3.7: workload-aware conservativeness (via clustering) and explicit operational constraints (via control-period selection and cooldown) jointly determine the achievable Pareto frontier in energy-aware consolidation and DVFS.

3.9 Interpreting migration volume and DVFS operating points. Beyond energy and SLA risk, operators often care about whether a policy achieves its savings primarily by turning hosts off (consolidation) or by lowering frequency (DVFS), because these mechanisms have different operational costs. Table 10 includes two diagnostics that make this distinction explicit: average active hosts and average DVFS frequency on active hosts.

The results show that active host count is the dominant driver of energy on these traces. Static and DVFS-Only keep all 16 hosts active; DVFS-Only reduces mean frequency to 0.700 and achieves a 12.78% energy saving, but it cannot capture the much larger idle-power savings that come from powering hosts down. In contrast, Consol-Only reduces mean active hosts to 2.694 and achieves a 72.03% energy saving even without DVFS. Consol+DVFS reduces frequency as well (mean frequency 0.741) and achieves the minimum energy among evaluated policies (337.990 kWh mean), but it does so while operating hosts at a high utilization ratio, which elevates the tail-latency proxy and produces rare unmet-demand episodes on workload B.

Cluster-aware generated policies shift the operating point by enforcing higher minimum frequencies on bursty tenants. LLM-P03, for example, maintains a higher mean frequency (0.901) than Consol+DVFS (0.741) and therefore consumes 14.07% more energy than Consol+DVFS (385.553 kWh versus 337.990 kWh). This cost buys a measurable latency benefit: the p99 latency proxy decreases from 4.929× to 3.287×, a 33.3% reduction. LLM-P06 sits between these two regimes. It consumes only 5.49% more energy than Consol+DVFS (356.570 kWh) while eliminating unmet demand, but it requires 53 migrations on average, more than double Consol+DVFS (23). The additional migrations are explained directly by its short underload window and cooldown configuration (Table 6), which increases responsiveness and avoids long-lived underutilization at the cost of higher operational churn.

The migration counts also clarify why some policies are Pareto-optimal even when their energy is not minimal. DVFS-Only is Pareto-optimal because it achieves zero migrations and low latency at moderate energy. In a production setting where migration costs are high (e.g., due to network contention or state transfer), such a policy is preferred over more aggressive consolidation when migration costs are high. Conversely, if energy is the dominant constraint and the platform tolerates migrations, Consol+DVFS or migration-heavy variants such as LLM-P06 become attractive. The multi-metric view is therefore essential for policy selection: the same energy saving can be achieved with very different combinations of consolidation intensity and DVFS aggressiveness.

In summary, the evaluated policies occupy a spectrum of operating points that are directly interpretable through (active hosts, frequency). The cluster-aware policies expand the set of safe operating points by decoupling the DVFS decision from a single global target and by explicitly reserving headroom for bursty tenants.

Table 8. Offline A/B results on Workload A (preC0–preC1, 16 tenants).

	energy_kwh	sla_unmet_pct	migrations	lat_p99_proxy	avg_active_hosts	avg_freq
Static	1380.643	0.0	0.0	1.373	16.0	1.0
DVFS-Only	1201.01	0.0	0.0	1.635	16.0	0.7
Consol-Only	437.262	0.0	18.0	4.368	3.205	1.0
Consol+DVFS	383.096	0.0	18.0	4.381	3.205	0.742
LLM-P01	424.41	0.0	21.0	4.015	3.369	0.9
LLM-P02	406.7	0.0	18.0	4.584	3.303	0.828
LLM-P03	430.956	0.0	23.0	3.504	3.461	0.902
LLM-P04	388.265	0.0	35.0	4.136	3.105	0.811
LLM-P05	422.529	0.0	19.0	3.7	3.318	0.905
LLM-P06	389.664	0.0	45.0	4.542	3.141	0.809
LLM-P07	426.64	0.0	39.0	4.194	3.389	0.901
LLM-P08	425.996	0.0	18.0	4.154	3.378	0.903
LLM-P09	401.931	0.0	33.0	6.152	3.0	0.906
LLM-P10	426.457	0.0	22.0	5.08	3.348	0.908
LLM-P11	414.127	0.003	25.0	5.12	3.19	0.905
LLM-P12	415.322	0.0	31.0	4.823	3.23	0.901

Table 9. Offline A/B results on Workload B (preC2–preC3, 16 tenants).

	energy_kwh	sla_unmet_pct	migrations	lat_p99_proxy	avg_active_hosts	avg_freq
Static	1359.775	0.0	0.0	1.227	16.0	1.0
DVFS-Only	1189.217	0.0	0.0	1.36	16.0	0.7
Consol-Only	329.273	0.001	28.0	5.477	2.182	1.0
Consol+DVFS	292.884	0.001	28.0	5.477	2.182	0.74
LLM-P01	342.89	0.0	29.0	3.072	2.602	0.904
LLM-P02	350.877	0.0	22.0	3.444	2.517	0.996
LLM-P03	340.151	0.0	22.0	3.069	2.578	0.901

LLM-P04	330.972	0.001	33.0	3.885	2.224	0.998
LLM-P05	327.642	0.0	39.0	4.394	2.368	0.906
LLM-P06	323.477	0.0	61.0	4.567	2.118	0.995
LLM-P07	346.461	0.0	25.0	3.134	2.451	0.997
LLM-P08	341.59	0.0	32.0	3.708	2.375	0.996
LLM-P09	334.255	0.001	31.0	3.76	2.272	0.998
LLM-P10	353.81	0.0	25.0	3.159	2.557	0.998
LLM-P11	328.405	0.0	27.0	3.707	2.194	0.998
LLM-P12	332.425	0.0	38.0	3.866	2.249	0.998

Table 10. Combined results averaged over Workload A and B, including energy savings vs Static.

	energy_kwh	energy_saving_pct_vs_static	sla_unmet_pct	migrations	lat_p99_proxy	avg_active_hosts	avg_freq
Static	1370.209	0.0	0.0	0.0	1.3	16.0	1.0
DVFS-Only	1195.114	12.78	0.0	0.0	1.497	16.0	0.7
Consol-Only	383.267	72.03	0.000268	23.0	4.922	2.694	1.0
Consol+DVFS	337.99	75.33	0.000268	23.0	4.929	2.694	0.741
LLM-P01	383.65	72.0	0.0	25.0	3.543	2.986	0.902
LLM-P02	378.788	72.36	0.0	20.0	4.014	2.91	0.912
LLM-P03	385.553	71.86	0.0	22.5	3.287	3.019	0.901
LLM-P04	359.618	73.75	0.000462	34.0	4.011	2.665	0.905
LLM-P05	375.086	72.63	0.0	29.0	4.047	2.843	0.905
LLM-P06	356.57	73.98	0.0	53.0	4.555	2.629	0.902
LLM-P07	386.55	71.79	0.0	32.0	3.664	2.92	0.949
LLM-P08	383.793	71.99	0.0	25.0	3.931	2.876	0.95

LLM-P09	368.093	73.14	0.000468	32.0	4.956	2.636	0.952
LLM-P10	390.134	71.53	0.0	23.5	4.12	2.953	0.953
LLM-P11	371.266	72.9	0.001506	26.0	4.413	2.692	0.952
LLM-P12	373.874	72.71	0.0	34.5	4.345	2.739	0.95

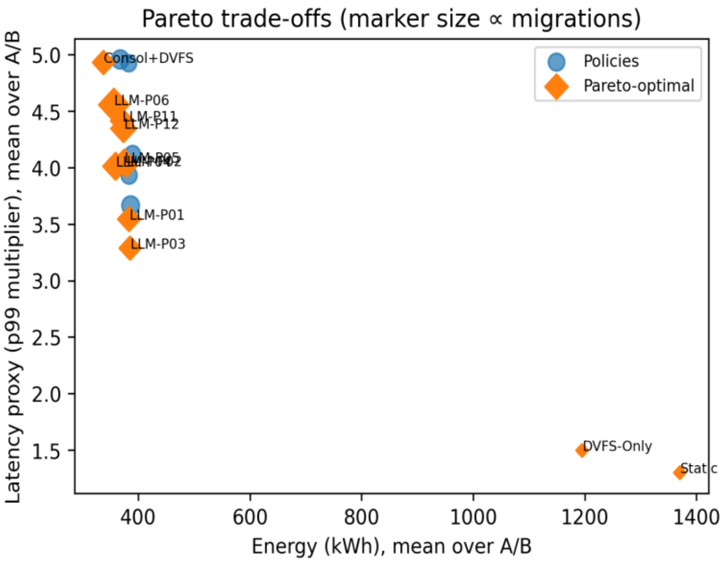

Figure 4. Energy–latency Pareto trade-offs averaged over Workload A and B (marker size proportional to migrations).

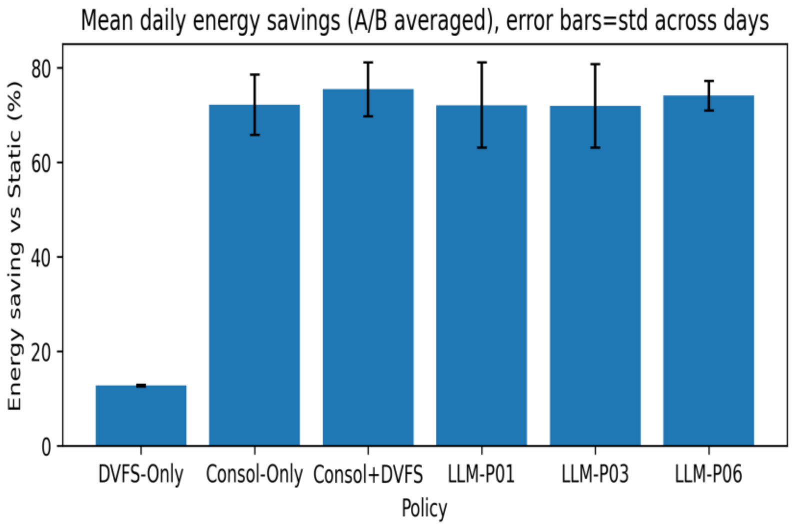

Figure 5. Mean daily energy savings vs Static with standard-deviation error bars (A/B averaged).

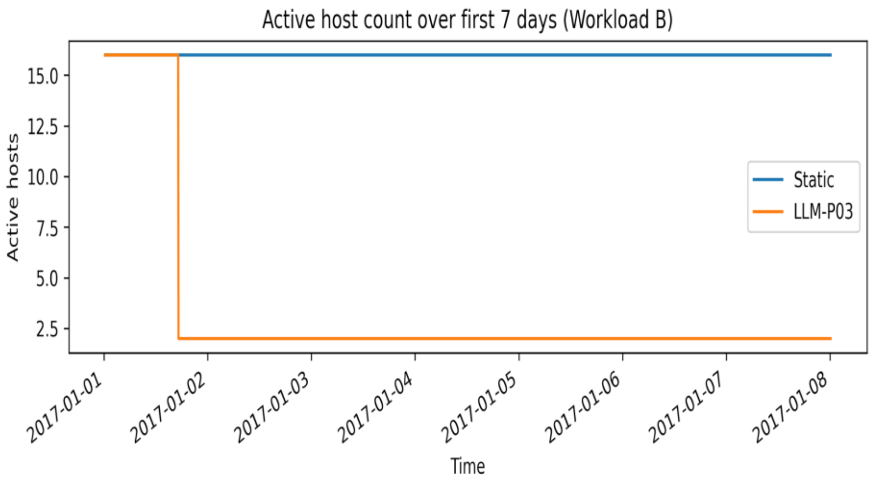

Figure 6. Active host count over the first 7 days of Workload B for Static and LLM-P03.

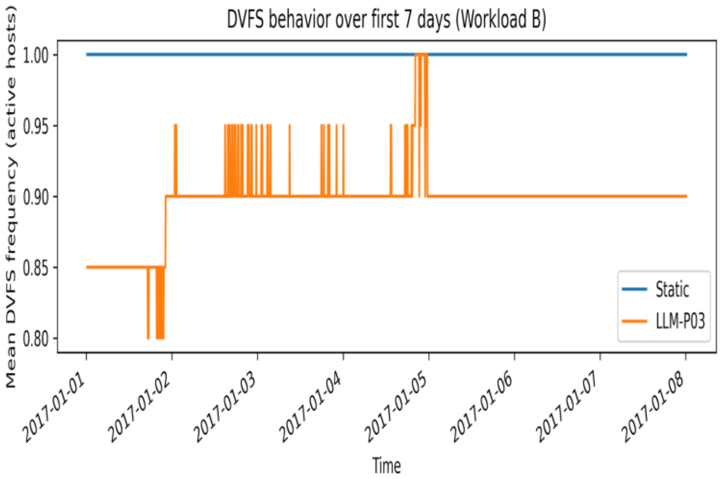

Figure 7. Mean DVFS frequency over the first 7 days of Workload B for Static and LLM-P03.

Table 11. Paired t-test on daily energy (A/B averaged) vs Static baseline.

policy	mean_energy_kwh	energy_saving_pct	t_stat	p_value
Consol+DVFS	11.266	75.33	27.057	3.97e-22
LLM-P06	11.886	73.98	28.592	8.48e-23
Consol-Only	12.776	72.03	26.335	8.44e-22
LLM-P01	12.788	72.0	24.103	9.88e-21
LLM-P03	12.852	71.86	24.22	8.65e-21
DVFS-Only	39.837	12.78	29.13	5.02e-23

Table 12. Ablation on Workload B: cluster-aware LLM-P03 versus globalized parameters.

	energy_kwh	sla_unmet_pct	migrations	lat_p99_proxy	avg_active_hosts	avg_freq
LLM-P03	340.1509	0.0	22.0	3.0694	2.5776	0.9006
LLM-P03-Global	301.96	0.0222	66.0	9.5682	2.1652	0.7889

Table 13. Sensitivity of LLM-P03 to consolidation control period (decision interval).

	decision_interval_min	energy_kwh_A	sla_unmet_pct_A	migrations_A	lat_p99_A	energy_kwh_B	sla_unmet_pct_B	migrations_B	lat_p99_B
LLM-P03-30min	30.0	421.6638	0.0	43.0	3.6281	322.3306	0.0	25.0	2.9775
LLM-P03-60min	60.0	430.9561	0.0	23.0	3.5036	340.1509	0.0	22.0	3.0694
LLM-P03-120min	120.0	462.4444	0.0206	30.0	4.0973	363.7129	0.0	31.0	3.7691

Limitations

The study has several limitations that follow from its offline, trace-driven scope and from the public dataset that enables reproducibility. First, the evaluated traces contain CPU usage only. Real consolidation decisions also depend on memory, I/O, and network constraints, and these resources can be bottlenecks even when CPU headroom exists. Second, the power model is parametric and does not capture platform-specific power curves, temperature effects, or power caps. We choose a widely used affine-plus-nonlinear model with DVFS scaling to enable comparative evaluation [2], [3], [5], but absolute energy values should be interpreted as estimates.

Third, the tail-latency proxy uses an M/M/1 response-time multiplier as a workload-agnostic approximation [17]. This proxy captures the qualitative increase in delay near saturation, but it does not model service-time variability, multi-stage request paths, or interference between co-located tenants. Fourth, the simulator uses hourly control epochs. Real systems use different control periods, and the optimal period depends on migration cost and workload volatility. Fifth, because the dataset provides machine-level traces, we deterministically decompose each trace into multiple tenants to evaluate consolidation. This preserves temporal patterns and total load exactly, but it does not create independent tenant behaviors; therefore, migration dynamics are less diverse than in a deployment with many distinct tenants.

Finally, the “LLM-guided” component is evaluated through a constrained, parseable text specification format and a deterministic candidate set, rather than through closed-loop interaction with a deployed LLM. The paper’s core contribution is the compilation-and-evaluation pipeline; the empirical evaluation reported here isolates the compilation and offline A/B testing steps by using deterministic specification generation. Despite these limitations, the results provide a fully reproducible empirical baseline for energy-aware offline A/B testing and illustrate how cluster-aware policies improve the Pareto frontier.

Conclusion

This paper presented an offline A/B testing workflow for energy-aware consolidation and DVFS policies in virtualized data centers. The workflow clusters tenants by power sensitivity, represents policies as human-readable text specifications, compiles specifications into executable controllers, and evaluates candidates with a trace-driven simulator that estimates energy, SLA risk, migration volume, and tail-latency proxies. Full experiments on the GCT-TRU trace dataset (eight 5-minute CPU traces over January 2017) demonstrated that consolidation delivers large energy reductions (≈72%) compared with DVFS alone (≈13%), while naive consolidation+DVFS can increase tail-latency risk. Cluster-aware generated policies improved trade-offs: LLM-P03 achieved 71.86% energy saving with 0% unmet demand and a lower latency proxy than the most aggressive baseline. Pareto-front analysis and paired daily tests provided operator-relevant evidence for selecting safe and efficient policies. The results establish a reproducible empirical foundation for LLM-assisted policy design and show that cluster-aware headroom constraints improve the energy–performance Pareto frontier in offline evaluation.

References

[1] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," Computer, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[2] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, May 2012.

[3] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397–1420, 2012.

[4] R. Nathuji and K. Schwan, "VirtualPower: coordinated power management in virtualized enterprise systems," in Proc. ACM SIGOPS SOSP, 2007, pp. 265–278.

[5] E. Le Sueur and G. Heiser, "Dynamic voltage and frequency scaling: the laws of diminishing returns," in Proc. HotPower, 2010.

[6] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: eliminating server idle power," in Proc. ASPLOS, 2009, pp. 205–216.

[7] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in Proc. ISCA, 2007, pp. 13–23.

[8] R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne, "Controlled experiments on the web: survey and practical guide," Data Mining and Knowledge Discovery, vol. 18, no. 1, pp. 140–181, 2009.

[9] R. Kohavi et al., "Online controlled experiments at large scale," in Proc. KDD, 2013, pp. 1168–1176.

[10] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proc. 5th Berkeley Symp. Math. Statist. Probab., vol. 1, 1967, pp. 281–297.

[11] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, pp. 53–65, 1987.

[12] I. T. Jolliffe, Principal Component Analysis, 2nd ed. Springer, 2002.

[13] S. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level power management for dense blade servers," in Proc. ISCA, 2006, pp. 66–77.

[14] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in Proc. ACM SoCC, 2012, pp. 7:1–7:13.

[15] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format + schema," Google Inc., White Paper, Nov. 2011.

[16] S. Garg, "Task resource usage of Google Cluster Usage Trace dataset," Zenodo, Aug. 2022, doi: 10.5281/zenodo.6979672.

[17] L. Kleinrock, Queueing Systems, vol. 1: Theory. Wiley, 1975.

[18] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in Proc. HotNets, 2016, pp. 50–56.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[20] T. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020.

[21] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.

[22] M. Chen et al., "Evaluating large language models trained on code," arXiv:2107.03374, 2021.

[23] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[24] Hanqi Zhang, “DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models”, JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[25] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, “Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)”, JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[26] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, “Cancer image classification based on DenseNet model,” Journal of Physics: Conference Series, vol. 1651, no. 1, p. 012143, 2020.