



Article

Selective Admission and Occupancy-Targeted Placement for Carbon-Aware Kubernetes Scheduling: A Measurement-Calibrated CERN Case Study

Sebastián Andrés Uribe Ruiz ^{1,2,*} , Laura Eve Sarah Llinares ¹, Matteo Bunino ¹ and Ricardo Rocha ¹

¹ European Organization for Nuclear Research (CERN), 1211 Geneva 23, Switzerland; llinares.laura@cern.ch (L.E.S.L.); matteo.bunino@cern.ch (M.B.); ricardo.rocha@cern.ch (R.R.)
² School of Applied Sciences and Engineering, Universidad EAFIT, Medellín 050022, Colombia
* Correspondence: sebastian.andres.uribe.ruiz@cern.ch or sauriber2@eafit.edu.co

Abstract

Temporal carbon-aware scheduling can reduce electricity-related emissions, but deferring work to lower carbon-intensity intervals can concentrate demand and increase waiting. We present benefit-gated deferral (BGD), a selective admission policy for nonpreemptive batch jobs on shared clusters, evaluated using realistic high-energy physics workloads in a measurement-calibrated CERN case study. BGD requires relative and absolute estimated carbon gains and ranks eligible starts by estimated grams avoided per hour waited. Arrival patterns extracted from CERN's Next Generation Triggers platform define the primary scenario. Measurements of CMS simulation and reconstruction, ATLAS event generation, and LHCb simulation calibrate occupancy-dependent power and runtime. A 100-worker simulation compares BGD with immediate admission under simulated default Kubernetes and occupancy-targeted placement over a 90-day French carbon-intensity series. Synthetic arrivals provide a controlled sensitivity benchmark. With 24 h flexibility, the simulations yield accounted-emissions reductions of 17.40% and 7.98% under default placement at 30% and 50% offered load. Adding occupancy-targeted placement increases these reductions to 28.34% and 12.17%, while 95th-percentile waits exceed 20 h. Completion flexibility and occupancy-dependent execution affect the benefit of deferral, which must be assessed alongside waiting times and deadline compliance.

Keywords: carbon-aware scheduling; Kubernetes; high-energy physics; workload traces; energy efficiency; admission control; discrete-event simulation



Academic Editors: Efthimia Mavridou and Eleni Vrochidou

Received: 9 September 2026

Revised: 23 September 2026

Accepted: 24 September 2026

Published: 28 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

This study examines the carbon benefit of selective admission beyond placement alone for flexible high-energy physics (HEP) batch work in a fixed electricity region. We assess this benefit alongside waiting time and deadline compliance, accounting for the effects of worker occupancy on execution.

Carbon-aware scheduling exploits variation in electricity emissions by changing when or where computation runs. Capacity-shaping systems and Kubernetes extensions can delay flexible work [1,2], but individually attractive shifts may concentrate demand, increase completion time, or leave reserved capacity idle [3–5]. Admission determines when a job is released; placement determines its worker. Consolidation can reduce powered node-time while higher active power or longer execution increases active energy. Carbon accounting weights that energy by the time-varying electricity intensity.

In HEP computing, event generation, detector simulation, and reconstruction respond differently to concurrency. The LHC experiments are also assessing computing's environmental cost directly [6]. Here, measurements of CMS GEN-SIM, CMS RECO, ATLAS Sherpa, and LHCb SIM calibrate occupancy-dependent power and runtime, while CERN Next Generation Triggers (NGT) observations supply the arrival-time profile.

We present *benefit-gated deferral* (BGD), which requires both relative and absolute estimated carbon gains before deliberately delaying feasible work. For a job with estimated active emissions $\hat{g}_j(a_j)$ if started at arrival, absolute benefit is $\Delta\hat{g}_j(t) = \hat{g}_j(a_j) - \hat{g}_j(t)$ in grams of CO₂; relative reduction is $\Delta\hat{g}_j(t)/\hat{g}_j(a_j)$. BGD ranks eligible starts by estimated grams avoided per hour of start-time wait, combining established carbon-delay reasoning with benefit gates and logical capacity reservations.

We evaluate BGD with simulated default Kubernetes placement and *occupancy-targeted placement* (OTP). OTP consolidates toward a measurement-informed target of 32 allocated logical CPUs per 64-CPU worker. This target concerns allocation, not measured CPU utilization. The contributions are:

1. A selective admission design with relative and absolute benefit gates, savings-per-wait ranking, and an opt-in Kubernetes operator implementation.
2. A CERN case study combining NGT-informed arrivals with measured HEP execution responses and managed-idle accounting.
3. A component-level evaluation of admission and placement across load and completion flexibility, including waiting times, deadline compliance, and arrival-shape resampling.

Section 2 positions the study against prior work, Sections 3 and 4 describe the design and evaluation, and Sections 5 and 6 report results and limitations.

2. Related Work

2.1. Temporal Admission and Carbon-Delay Trade-Offs

Radovanović et al. shape flexible computation through virtual capacity curves [1], while Piontek et al. shift noncritical Kubernetes jobs using predicted emissions [2]. GAIA studies carbon, completion time, and purchasing cost jointly [4]. Hanafy et al. examine synchronized demand shifts and propose Carbon Savings-per-Delay (CSD), dividing savings by delay D plus an offset β [5].

BGD's positive-wait ranking is a nonpreemptive start-time adaptation of zero-offset CSD, meaning $\beta = 0$. For the same job, candidate set, carbon estimates, and start-time delay, the rankings coincide: normalizing savings by a common job-specific baseline or changing delay units only rescales the ratio. CSD's original completion-delay formulation is not interchangeable with start-time wait when runtime varies. BGD handles zero wait separately and combines this established objective with benefit gates and capacity reservations.

Hall et al. provide probabilistic performance guarantees [7]; BGD instead uses heuristic reservations and measures compliance after execution. Doshi and Shah find that simple overnight batching can recover much of a tuned policy's carbon benefit for GPU workloads while missing fewer urgent deadlines [8], motivating simple controls alongside forecast-based admission.

CARA combines temporal and geographic execution choices [9]. Beena et al. select regions and windows using intensity thresholds [10], but report decisions rather than a paired reduction against a stated baseline. Our evaluation holds the region fixed; its greedy control tests minimum-intensity selection without reproducing either complete framework.

2.2. Cluster Coordination and Workload Constraints

GREEN coordinates machine-learning cluster resources [11], and CarbonFlex combines provisioning with elastic job scheduling [12]. In our model, jobs retain their CPU allocation

and are neither migrated nor suspended after admission, although occupancy changes their power and progress. PCAPS and deadline-constrained workflow scheduling incorporate dependencies [13,14]; our independent batch units have declared completion flexibility and no dependency model.

Benhari and Trystram study adaptive carbon-aware scheduling under HPC capacity constraints [15]. Sukprasert et al. identify limits arising from duration, service requirements, and regional electricity variability [3]. Together with CSD's demand-synchronization analysis [5], these studies motivate examining shared-capacity effects alongside intensity. We additionally model the measured occupancy response of HEP execution.

2.3. Energy-Aware Placement and Simulation

Kubernetes supports configurable resource scoring and bin packing [16,17]. Trimaran's TargetLoadPacking uses measured CPU utilization to pack toward a configured target, then spreads among highly loaded nodes [18]. OTP instead uses allocated requests and retains nonempty-worker and tight-fit preferences above its target. OTP adapts this established placement principle and is not evaluated against Trimaran.

Admission changes concurrency, whose effects on power and runtime are absent from a fixed-power, fixed-runtime model. Our response curves inform the target and update both power and progress as occupancy changes, while idle accounting captures powered node-time. This supports comparing admission with and without OTP, but does not establish an optimal target.

EA-K8S models microservice CPU demand, communication, and data-center PUE [19]. Raith et al. predict placement energy using graph neural networks [20]; Gouaouri et al. combine power awareness, balancing, and packing [21]; PAX addresses heterogeneous infrastructure [22]. Yang et al. survey this broader container-scheduling design space [23]. Our study instead concerns independent HEP jobs on homogeneous workers, with deliberate deferral controlled through admission.

EcoKube models heterogeneous edge-cloud orchestration [24], and GPS evaluates predictive greedy scheduling using runtime and CPU utilization as an energy indicator [25]. Rui et al. connect measured task power to larger Kubernetes orchestration simulations [26]. We similarly use measurements to calibrate a simulated infrastructure, but track occupancy-dependent job progress and active power together with managed idle energy.

3. System and Policy Design

3.1. Scheduling Interface

The system considers independent, opt-in batch jobs. A job specifies its CPU request, estimated duration, and allowed completion flexibility. An admission policy selects its release time using the available carbon forecast and logical capacity reservations. A placement policy subsequently assigns the released job to a feasible worker. Running jobs are neither suspended nor migrated.

For job j , arrival time a_j , nominal duration d_j , and flexibility F , the experimental completion deadline is

$$D_j = a_j + d_j + F. \quad (1)$$

Here, F is the allowance beyond nominal completion at $a_j + d_j$, not permission to start F hours late. Candidate starts are spaced at 15 min intervals up to

$$\ell_j = \max(a_j, D_j - 1.65d_j). \quad (2)$$

The factor 1.65 provides a runtime margin. Deadline compliance is measured after execution, rather than guaranteed by this latest-start calculation.

The 15 min spacing permits sub-hourly releases with a bounded candidate set. Although carbon inputs are hourly, candidate windows can differ in capacity availability and overlap with intensity boundaries. Candidate times are anchored to arrival and include the latest allowable start; reservation slots are anchored to the replay start. Sensitivity to this resolution has not been quantified in the reported evaluation.

Reservations are constructed in arrival order over 15 min slots. Let $q_j(t)$ denote projected peak occupancy over a candidate execution interval after including the new job. The planner uses this cluster-level fraction to estimate future node occupancy. Reservation and scoring durations are independently configurable. Both equal the nominal job duration in the reported simulations.

3.2. Benefit-Gated Deferral

BGD estimates the active carbon cost of starting job j at candidate time t :

$$\hat{g}_j(t) = \frac{c_j \hat{d}_j}{3.6 \times 10^6} p_{\alpha_j}(q_j(t)) s_{\alpha_j}(q_j(t)) \hat{I}_{a_j}(t, \hat{d}_j). \quad (3)$$

Here, c_j is the logical CPU request, α_j is the workload archetype, p_{α} is active watts per allocated CPU, s_{α} is a runtime multiplier, and $\hat{I}_{a_j}$ is the arrival-time forecast averaged over the candidate duration. Durations are expressed in seconds and intensity in grams per kilowatt-hour.

Specifically, with $\hat{I}_{a_j}$ denoting the forecast available at arrival,

$$\hat{I}_{a_j}(t, \hat{d}_j) = \frac{1}{\hat{d}_j} \int_t^{t+\hat{d}_j} \hat{I}_{a_j}(\tau) d\tau. \quad (4)$$

The evaluated scoring duration is $\hat{d}_j = d_j$. The multiplier s_{α} scales estimated energy in Equation (3), but does not extend this forecast-averaging window or the nominal reservation duration. This is a planning approximation, distinct from the occupancy-dependent execution intervals used for realized accounting.

When immediate execution is projected feasible, define the eligible future starts as

$$\mathcal{E}_j = \{t > a_j : t \text{ fits reservations, } t \leq \ell_j, \Delta \hat{g}_j(t) \geq \max(\theta \hat{g}_j(a_j), g_{\min})\}, \quad (5)$$

where $\Delta \hat{g}_j(t) = \hat{g}_j(a_j) - \hat{g}_j(t)$. The selected release time is

$$t_j^* = \begin{cases} a_j, & \mathcal{E}_j = \emptyset, \\ \arg \max_{t \in \mathcal{E}_j} \frac{\Delta \hat{g}_j(t)}{(t - a_j)/3600}, & \text{otherwise.} \end{cases} \quad (6)$$

The thresholds $\theta = 0.20$ and $g_{\min} = 0.10$ g require a fractional improvement while excluding small absolute benefits; they are not established optima. The absolute gate is more restrictive when estimated immediate emissions are below $g_{\min}/\theta = 0.50$ g. For example, a 30% reduction from 0.20 g saves 0.06 g and does not meet the absolute threshold. With fixed forecasts and reservations, higher thresholds can exclude candidates; aggregate effects need not be monotonic because decisions change subsequent reservations. The comparisons do not isolate sensitivity to either threshold.

Capacity-induced postponement is handled separately. If immediate execution is not projected feasible, the policy chooses the lowest-cost feasible candidate without the run-now benefit gates. If every candidate is projected full, it selects the least-loaded candidate. Released jobs that cannot be physically placed enter a queue ordered by nominal

latest start. Thus, planned postponement includes both carbon-motivated decisions and capacity effects.

The policy ranks candidate times for an individual job, not jobs against each other. Reservations are not revised from realized execution events in the archived simulation. Equation (3) also excludes the full incremental idle cost and the effect on incumbent jobs. Those consequences are evaluated by the node-time execution model in Section 4.3.

3.3. Placement and Component Controls

The *Default Kubernetes* baseline simulates CPU-only LeastAllocated placement [16]: all workers with sufficient free CPUs are considered. The worker with the most free CPUs is selected, with ties resolved by worker index. Each worker exposes 64 allocatable logical CPUs. Memory, accelerators, affinity, taints, topology, image locality, volumes, and preemption are not modeled. Released jobs share a queue ordered by $\max(D_j - d_j, a_j)$, then arrival and job identifier. These common simulation constraints do not reproduce the complete kube-scheduler, its queue sorting, or a versioned plugin configuration.

OTP first prefers placements whose resulting occupancy is at most u^* , then prefers nonempty nodes and tighter fits. When every feasible choice exceeds the target, it can still admit above the target. The target is a preference rather than a hard capacity limit. We use $u^* = 0.50$, corresponding to 32 allocated logical CPUs.

For a fixed archetype at constant occupancy, energy per nominal CPU-hour is proportional to

$$e_\alpha(u) = s_\alpha(u) \left[p_\alpha(u) + \frac{P_0}{64u} \right], \quad (7)$$

where P_0 is the modeled package idle power. Increasing occupancy amortizes idle power while potentially increasing active power and runtime. This calculation motivates target-based placement without claiming an optimum for a time-varying mixed queue.

We evaluate immediate admission and BGD with each placement rule to distinguish admission and placement effects. Immediate admission provides a baseline without deliberate deferral; first-fit tests consolidation without a target. A 40% relative-threshold control minimizes estimated cost without an absolute floor or savings-per-wait ranking, so it is a policy comparison, not a component ablation. Forecast-only greedy admission selects starts by intensity without benefit gates. The perfect-foresight control also changes the decision rule and cannot isolate forecast error. Comparisons retain identical job streams and execution assumptions.

3.4. Kubernetes Operator and Experimental Deployment

The opt-in operator is intended for experimental deployment at CERN. A publisher writes a versioned *CarbonForecast*; the controller stores the selected start time and logical reservation in a *CarbonDecision*. It releases Jobs through `spec.suspend` and standalone Pods through scheduling-readiness gates, after which kube-scheduler handles placement. Each workload receives one persistent decision; running containers are not paused. Default shadow mode records decisions but releases workloads immediately.

A namespaced *CarbonPolicy* configures mode, benefit gates, maximum delay, slot spacing, forecast age, reservation-duration factor, logical CPU capacity, and watts per CPU. Durations come from workload metadata. Unlike the simulation, the operator uses fixed watts per CPU, clock-aligned slots, forecast coverage of the full scoring interval, and no least-cost postponement when immediate execution is infeasible. Logical reservations cover managed workloads without reserving physical capacity. Invalid or stale forecasts can fail open; outages require monitoring and recovery.

The operator retains BGD's gates and savings-per-wait ranking despite these implementation differences. OTP and worker power-state control require separate integration; reported savings come from simulation, not deployment.

4. Materials and Methods

4.1. Primary Scenario: NGT-Informed HEP Batch Work

The primary evaluation combines arrival patterns extracted from CERN NGT with measurement-calibrated models of HEP batch workloads. The sampled platform population comprises user-profile namespaces and continuous-integration runners, rather than all CERN computing activity. Observed submission times determine the demand profile; the workload catalog separately defines CPU batch jobs. The trace covers 77 days, exactly 11 weeks, from 20 June 2026 at 22:00 UTC to 5 September 2026 at 22:00 UTC. It contains 15,286 pods across 368 namespaces. Of these, 14,715 were created inside the window and supply the arrival shape, 567 were created before the window, and four lack a creation timestamp. The latter two groups are excluded from arrival estimation. The CERN MONIT Prometheus long-term store was queried at five-minute resolution, merging six metrics per pod by unique identifier. Arrivals are taken from each pod's own creation timestamp.

The derived artifact is a per-day, per-hour arrival histogram without names, pod identifiers, or account identifiers. Every weekday-hour cell is observed exactly 11 times. Days with no arrivals are represented explicitly and remain eligible for resampling. The resulting activity profile specifies submission timing, while HEP archetypes, CPU requests, and duration assumptions specify the batch work. Table 1 identifies the role of each input.

Table 1. Inputs to the NGT-informed HEP scheduling scenario.

Input	Source	Evaluation Role
Arrival profile	Operational NGT pod observations	Weekday-hour pattern and day-resampling sensitivity
Execution response	Containerized HEP workload measurements	Archetype-specific active power and runtime curves
Workload composition	Declared HEP mixture and duration distribution	Controlled CPU demand and completion flexibility
Electricity signal	French historical intensity and forecast archive	Admission estimates and realized carbon accounting
Worker lifecycle	Declared idle and wake-energy parameters	Powered node-time accounting

The estimator groups creation timestamps in a 7×24 local-clock grid, adds a 0.5 pseudocount per cell, divides by the observed exposure of each cell, and normalizes the resulting rates to mean one. Summer observations use CEST and winter simulated arrivals use CET. Job arrivals then follow a nonhomogeneous Poisson process,

$$\lambda(t) = \bar{\lambda}h(t), \quad \bar{\lambda} = \frac{\rho 64N}{\mathbb{E}[c_j d_j]}, \quad (8)$$

where N is node count, ρ is nominal offered load, and $h(t)$ is the normalized weekly profile.

The primary scenario uses the pooled profile to represent aggregate platform activity. Population sensitivity uses two separate partitions: user-profile versus continuous-integration (CI) runner namespaces, and a batch-controller group versus the remaining pods. The batch-controller group contains pods associated with Job, MPIJob, PyTorchJob, Workflow, and EphemeralRunner owners. The remaining group includes other controller

types and is not exclusively interactive. These partitions provide alternative timing profiles, not measured HEP workload mixtures. Their effects are reported in Section 5.4.

4.2. HEP Workloads and Measurement Calibration

The hardware model is based on dual-socket AMD EPYC 7302 (Advanced Micro Devices, Santa Clara, CA, USA) workers with 32 physical cores and 64 logical CPUs. Calibration measurements were collected on the reference bare-metal Kubernetes 1.35.3 worker at CERN. Table 2 records the physical measurement scope separately from the simulated cluster size.

Table 2. Calibration configuration and primary simulation scale.

Property	Configuration
Processor architecture	Dual-socket AMD EPYC 7302
CPU capacity per modeled worker	32 physical cores, 64 logical CPUs
Measured hosts	1
Calibration design	26 archetype-occupancy cells, 2 to 4 repetitions each
Finalized calibration reports	68, including workload and quality metadata
Energy telemetry	Kepler 0.11.4, Linux powercap CPU-package counters
Telemetry storage	Prometheus 3.13.1
Primary simulated cluster	100 homogeneous workers

The workloads cover detector simulation (CMS GEN-SIM and LHCb SIM), reconstruction (CMS RECO), and event generation (ATLAS Sherpa), with distinct measured responses. These applications are commonly used in CERN computing and in application-based HEP benchmarking, as documented by HEPsScore and the HEPiX benchmarking resources [27,28]. Table 3 specifies the evaluation mixture and its declared scenario weights.

The separate Python 3.12 Monte Carlo control estimates π by counting uniformly sampled unit-square points inside a quarter circle. Each worker process draws 3.15×10^9 samples with seed $20260713 + i$, where i is the process index. Calibration sweeps 1, 8, 32, and 60 concurrent worker processes, with three repetitions each. Fixed random streams make this non-HEP control reproducible, although runtime varies. Its declared 25% job share corresponds to 10% of expected nominal CPU work under the common duration distribution. Sensitivity to removing it has not been measured.

Table 3. Workload catalog. Occupancy points count allocated logical CPUs out of 64. Asterisks identify CMS anchors incorporated from mixed-workload measurements.

Archetype	CPUs/Job	Catalog Occupancy Points	Job Share
CMS GEN-SIM	4	4, 8, 16, 32, 40, 56*	30%
CMS RECO	4	4, 8, 16, 32, 40, 56*	20%
ATLAS Sherpa	1	1, 2, 4, 8, 16, 32	15%
LHCb SIM	1	1, 2, 4, 8, 16, 32	10%
Monte Carlo control	1	1, 8, 32, 60	25%

Kepler supplies the node CPU active-energy counters used for calibration [29]. The pipeline handles resets and interpolates counter values at interval boundaries; calibration does not sum pod-level attribution.

For a cell with m repetitions, $c = 64u$ allocated CPUs, duration T_r , and node-context active energy E_r^{act} , the sweep anchors are

$$p_\alpha(u) = \frac{1}{mc} \sum_{r=1}^m \frac{E_r^{\text{act}}}{T_r}, \quad s_\alpha(u) = \frac{\bar{T}_\alpha(u)}{\bar{T}_\alpha(u_{\min})}. \quad (9)$$

Occupancy denotes the allocated CPU fraction, with $u_{\min}$ its lowest measured value. Curves use linear interpolation and flat endpoint extrapolation, with the CMS mixed-run anchors added below. Values $s_\alpha < 1$ indicate shorter runtimes than at the reference occupancy; the measurements do not establish their cause.

The 68 finalized reports are 26 archetype-occupancy cells measured with 2 to 4 repetitions each, all on the same host. They therefore describe available calibration material and within-cell repeatability rather than independent hardware replicates. The archetype-level power-factor parameters in Section 4.3 aggregate variability estimates from those small cells. Platform services remain present, and the loader does not exclude every co-tenant or counter-reset flag. CMS mixed-workload anchors are included in calibration. The available ATLAS and LHCb occupancy sweeps end at 32 logical CPUs, and the CMS sweeps end at 40, with the 56-CPU CMS points supplied by co-scheduled runs. Section 6.3 reports how much of each evaluated configuration’s execution falls above these measured ceilings.

Figure 1 shows why the workload distinction matters. At 32 occupied logical CPUs, active power spans approximately 3.68–4.12 W per CPU. The high-occupancy CMS runtime multipliers reach 1.41 for GEN-SIM and 1.36 for RECO. The control reaches 1.61 at 60 CPUs, while the observed ATLAS and LHCb curves remain near 1.02–1.03 at their 32-CPU end-points. These responses enter both execution-energy accounting and job progress.

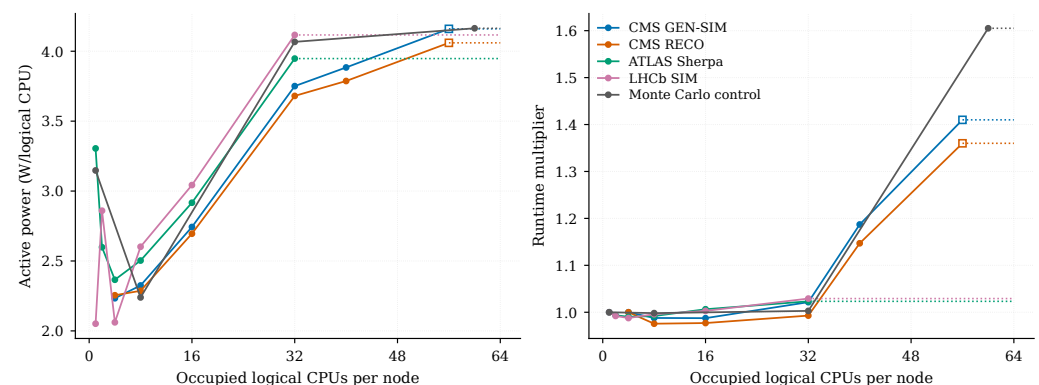


Figure 1. HEP workload power and runtime response curves supplied to the simulator. Circles denote catalog points, open squares high-occupancy CMS anchors, and dotted segments constant extrapolation. Line colors identify the workload archetypes labeled in the right panel. The Monte Carlo control provides a separate computational reference.

4.3. Node-Time Execution and Energy Accounting

For jobs $\mathcal{J}_n(t)$ running on worker n , occupancy and active power are

$$u_n(t) = \frac{1}{64} \sum_{j \in \mathcal{J}_n(t)} c_j, \quad (10)$$

$$p_n^{\text{act}}(t) = \bar{\xi}_n(t) \sum_{j \in \mathcal{J}_n(t)} c_j p_{\alpha_j}(u_n(t)), \quad (11)$$

where $\bar{\xi}_n(t)$ is the arithmetic mean of running jobs’ sampled power factors. One factor is drawn per job and held fixed throughout its execution. If v_α is the mean of the positive within-cell power coefficients of variation for archetype α , the generator uses

$\log \xi_j \sim \mathcal{N}(0, \max(v_\alpha, 10^{-6})^2)$, with a fallback $v_\alpha = 0.03$ if none is available. The resulting multiplicative factor has median one; its mean is not constrained to one.

A job begins with nominal work d_j . Between occupancy events, remaining work decreases at

$$\frac{dw_j(t)}{dt} = -\frac{1}{s_{\alpha_j}(u_n(t))}. \quad (12)$$

Arrivals and departures close an energy segment and update completion events for all jobs on the affected worker. Runtime is therefore responsive to concurrent occupancy rather than fixed at the initial placement.

The calibrated active term is Kepler's attributed active component, and powered workers additionally carry the fixed idle power P_0 . Together, these define the accounting boundary used throughout the evaluation. With realized intensity $I(t)$ in g/kWh, powered-state indicator $z_n(t)$, and time in seconds, accounted emissions are

$$C = \frac{1}{3.6 \times 10^6} \sum_n \int [P_n^{\text{act}}(t) + P_0 z_n(t)] I(t) dt + \sum_{b \in B} E_b I(t_b). \quad (13)$$

Each worker follows a three-state lifecycle. It is powered and busy while it holds at least one job. When its last job completes, it remains powered for a keep-alive interval to avoid a restart between closely spaced jobs. If no job arrives within that interval, the worker powers down and consumes no energy in the model until the next arrival triggers a wake-up. Placement therefore affects both job assignment and powered node-time, allowing consolidation to reduce managed idle energy.

Energy is calculated by omitting intensity. The idle parameter is $P_0 = 101.34$ W, taken as a low quantile of the reported idle estimates rather than their mean. Idle attribution depends on unused CPU capacity and total package draw, so estimates obtained during busy periods can exceed empty-node consumption. The low quantile supplies a lower-end estimate for the modeled idle component. Idle power is charged during execution and a 15 min keep-alive interval after a worker becomes empty. A subsequent wake-up incurs $E_b = 0.00844$ kWh, equivalent to five minutes at P_0 . Wake-up latency is not modeled.

The accounting boundary is CPU-package energy with a modeled worker lifecycle. DRAM, platform devices, power-supply losses, and cooling are outside that boundary. Releasing a virtual node can end its allocation to a cloud tenant, but does not establish that the underlying physical server stops consuming power. On bare-metal infrastructure, the modeled transition requires a power-state actuator, node draining, and boot orchestration. The operator of Section 3.4 implements admission control only, with placement and power-state management remaining separate integration components. Physical savings from powered-off intervals require those transitions to occur. The engine continues processing jobs beyond the arrival horizon, and the bundle provides a seven-day carbon extension for completion accounting. Coverage of the final completion is discussed in Section 6.3.

4.4. Carbon Signal and Experimental Configuration

The arrival horizon covers 90 days from 1 December 2025. French production-based carbon intensity averages 24.99 g/kWh and ranges from 11.0 to 60.5 g/kWh over this interval. RTE's production indicator excludes cross-border electricity exchanges [30]. This is the regional accounting signal for the experiment, not a marginal-emissions estimate or a reconstruction of CERN electricity procurement.

Admission decisions use the archived mixture-of-experts forecast moe6h: 597 origins six hours apart, including 360 in the arrival horizon, each supplying 24 hourly predictions. The adapter selects the latest origin available at decision time, uses its first prediction for the origin hour, and holds its last value beyond the forecast horizon. This follows issuance

order; blending forecast origins was not evaluated. Both signals are hourly piecewise constant; the realized-intensity loader forward-fills gaps and backfills leading gaps after hourly reindexing.

Table 4 compares the bundle's rounded predictions with realizations for origins and targets within the 90-day window. Errors are prediction minus realization. Repeated targets make forecast pairs dependent; metrics are not weighted by job decisions. They exclude the origin-hour proxy and held-last extension.

Table 4. Archived forecast error during the evaluation window, in g/kWh. No forecast was retrained for this analysis.

Horizon (h)	Pairs	MAE	RMSE	Mean Error
1–6	2159	2.97	4.27	−1.37
7–12	2153	3.62	5.17	−1.30
13–24	4288	4.91	6.98	−1.46
1–24	8600	4.10	5.96	−1.40

Absolute errors increase across the reported lead-time groups, and mean errors are negative. Their scheduling effect depends on job windows, capacity, and gates; the perfect-foresight control changes other policy settings and cannot isolate that effect.

The primary evaluation uses 100 workers, offered loads of 30% and 50%, flexibility of 6 and 24 h, and three paired simulation seeds. Jobs request one or four logical CPUs, retain their allocation, and run to completion. Nominal durations follow a lognormal distribution with median 5400 s and log-space standard deviation 0.9, clipped to 300–21,600 s. Memory capacity, networking, GPU allocation, migration, dependencies, and gang scheduling are not represented as scheduling constraints. No power capping is applied.

Offered load ρ sets the nominal CPU-work arrival rate relative to cluster capacity through Equation (8). It is distinct from CPU utilization or allocated occupancy during execution. The reported realized-allocation statistic divides accumulated allocated CPU-seconds, including post-horizon execution, by capacity times the 90-day arrival horizon. Slower execution increases this statistic because jobs hold CPUs longer. Within each arrival arm, all policies receive exactly the same job stream. Across different arrival profiles, job attributes are paired by index but finite-horizon job counts can differ.

The 30% and 50% loads provide two demand levels under the same capacity, rather than estimates of actual NGT utilization or a measured saturation threshold. Six- and 24-hour completion allowances contrast restricted and more permissive flexibility without claiming to reproduce production service agreements. Arrivals are generated by Poisson thinning with bound $1.05\bar{\lambda} \max h(t)$. Arrival random streams use seeds 1, 2, and 3, and separate job-attribute streams use the corresponding seed plus 1,000,000,007. Archetype and nominal duration are sampled independently. The clipped mean duration used to set $\bar{\lambda}$ is estimated from 200,000 fixed-seed draws with seed zero. These generation settings are unchanged across paired policies.

4.5. Synthetic Benchmark and Resampling

An independently specified synthetic arrival profile supplies a controlled benchmark for the same HEP execution model. Its unnormalized multipliers are 1.4 during weekday 08:00–20:00 UTC, 0.7 on weekday nights, and 0.8 at weekends. This arm changes the arrival profile while retaining the workload catalog, duration distribution, policy parameters, and energy accounting.

Arrival sensitivity is also assessed using 30 resampled NGT shapes, drawing days with replacement within weekdays and evaluating three seeds per shape. We report the mean and 5th, 50th, and 95th percentiles of the seed-averaged reductions, with linear

percentile interpolation. The central 90% empirical band measures variation across the archived shape replicates, not uncertainty in the complete physical model.

4.6. Metrics and Paired Comparisons

The principal metric is active plus managed-idle accounted emissions. Energy, planned-postponement fraction, waiting time, deadline misses, realized allocation, and run-time multiplier accompany it. Waiting time is actual start minus arrival and includes capacity queueing. A deadline miss means completion after Equation (1). Reported quantiles are means of per-run quantiles, not quantiles of pooled jobs.

For treatment k , immediate Default Kubernetes baseline b , and seed r ,

$$S_{k,r} = 100 \left(1 - \frac{C_{k,r}}{C_{b,r}} \right). \quad (14)$$

We report the mean of the three paired percentages. Baselines are recomputed within each load, flexibility, and arrival arm.

To decompose the combined configuration, let C_D denote Default Kubernetes with immediate admission, C_O OTP with immediate admission, and C_{OB} OTP with BGD:

$$S_{\text{placement}} = 100(C_D - C_O)/C_D, \quad (15)$$

$$S_{\text{added admission}} = 100(C_O - C_{OB})/C_D. \quad (16)$$

These terms add on a common baseline. The relative benefit of BGD against OTP's own immediate baseline instead divides by C_O . The component decomposition is descriptive and does not assume that placement and admission effects are independent.

Seed-level intervals use a Student t multiplier with two degrees of freedom and characterize simulation variation conditional on the input models. Across the three nonbaseline configurations in both arrival arms, both loads, and both flexibility settings, the resulting 95% half-width does not exceed 0.16 percentage points, and is below 0.10 points in half of the 24 cells. These intervals describe seed variation under fixed inputs, not uncertainty in the arrival estimator, calibration, or accounting boundary. Cluster sizes are separate experimental factors, not additional replicates.

OpenAI Codex assisted with manuscript drafting, repository inspection, and the preparation of verification and analysis scripts. All reported quantities derive from the simulator and calibration measurements described above.

5. Results

All reductions use the paired baseline defined in Section 4.6 and the CPU-package boundary and modeled worker lifecycle of Section 4.3. They describe simulated package-level savings, not measured whole-server savings. Admission can be controlled independently of worker power states, but the reported energy and carbon outcomes include their modeled effects.

5.1. Selective Admission in the NGT-Informed HEP Scenario

BGD reduces accounted emissions under both evaluated loads without changing placement. With 24 h flexibility, the reductions are 17.40% at 30% offered load and 7.98% at 50% load (Table 5). In absolute terms, these correspond to approximately 183 kg of CO₂ avoided against a 1053 kg baseline, and 118 kg against a 1477 kg baseline, over 90 days on 100 workers. The smaller reduction at higher load shows that the admission benefit is sensitive to demand at fixed worker capacity.

Table 5. Accounted-emissions reduction (%) in the NGT-informed HEP simulation at 100 workers. Each value is the mean of three paired seed reductions relative to Default Kubernetes with immediate admission. Across these cells, the 95% seed half-width does not exceed 0.16 percentage points. BGD denotes benefit-gated deferral and OTP occupancy-targeted placement.

Configuration	30% Offered Load		50% Offered Load	
	6 h	24 h	6 h	24 h
Default Kubernetes + immediate	0.00	0.00	0.00	0.00
Default Kubernetes + BGD	3.88	17.40	1.03	7.98
OTP + immediate	16.56	16.56	4.70	4.70
OTP + BGD	19.31	28.34	6.04	12.17

Adding OTP produces combined reductions of 28.34% and 12.17% in the corresponding 24 h cases, or approximately 298 kg and 180 kg avoided against the same baselines. This configuration changes both the release time and the worker selected for each job. Its additional benefit depends on placement integration and the modeled worker lifecycle.

For the 24 h cases, the paired 95% seed intervals, expressed as mean reduction plus or minus a half-width in percentage points, are 17.40 ± 0.12 and 7.98 ± 0.13 for BGD with default placement, 16.56 ± 0.12 and 4.70 ± 0.06 for OTP alone, and 28.34 ± 0.05 and 12.17 ± 0.11 for the combined configuration at 30% and 50% load, respectively. These narrow intervals are conditional on fixed inputs and do not characterize uncertainty in the calibration or accounting boundary.

5.2. Contribution of Placement and Execution Response

At 30% offered load, OTP with immediate admission reduces accounted emissions by 16.56%. BGD then adds 11.78 percentage points on the common Default Kubernetes baseline. At 50% load, the corresponding components are 4.70 and 7.47 points. Admission accounts for a larger share of the lower-load combined reduction under NGT-informed arrivals than under the synthetic benchmark, where placement contributes 11.31 points and added admission 5.21. Figure 2 compares the two arrival profiles.

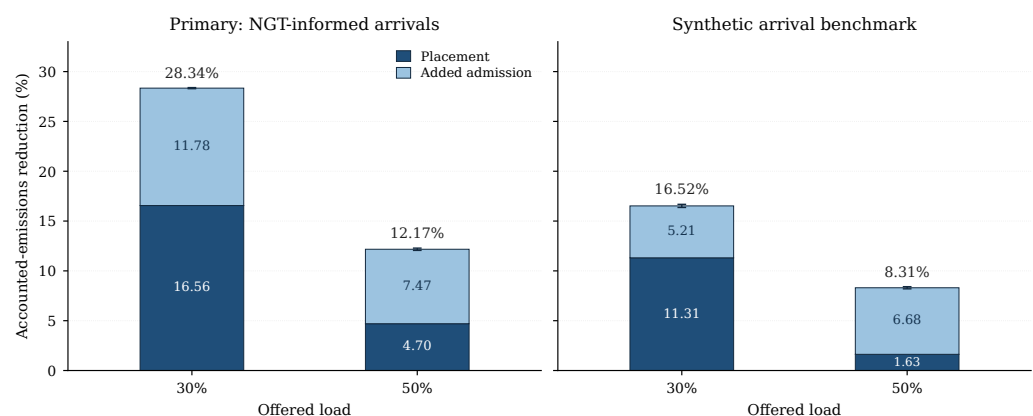


Figure 2. Contribution of OTP placement and additional BGD admission to simulated accounted-emissions reductions at 100 workers and 24 h flexibility. Both components use the immediate Default Kubernetes denominator within their own arrival arm.

At 30% offered load and 24 h flexibility under NGT-informed arrivals, Default Kubernetes with immediate admission uses approximately 18.90 MWh of active energy and 21.80 MWh of managed-idle energy. OTP alone raises active energy to 19.82 MWh while lowering managed-idle energy to 13.45 MWh. Its benefit therefore depends on reducing powered node-time despite an increase in aggregate active energy.

With Default Kubernetes unchanged, BGD instead lowers active energy to 14.05 MWh while managed idle remains near 21.66 MWh. The mean runtime multiplier decreases from 1.204 to 1.017. Under the calibrated model, changing admission timing can therefore improve execution conditions as well as carbon-intensity exposure.

Figure 3 illustrates the temporal structure in a paired, single-seed replay at 30% offered load and 24 h flexibility. Under immediate admission, occupancy follows the working week, approaching capacity during weekday daytime periods and falling close to zero at weekends. The arrival profile peaks at 6.15 times its mean rate, corresponding to a peak nominal CPU-work arrival rate of approximately 1.85 times cluster capacity. Over the 90-day replay, mean baseline occupancy is 5.45 times higher on weekdays than at weekends, and exceeds 90% for 16.8% of the observation period. The combined configuration reduces the weekday-to-weekend ratio to 1.97 and remains below 90% occupancy in this replay.

The Pearson correlation between occupancy and carbon intensity changes from +0.144 under immediate admission to −0.154 under the combined configuration. Occupancy-weighted carbon intensity decreases from 26.66 to 24.05 g/kWh, compared with an unweighted series mean of 24.99 g/kWh. These modest associations are consistent with a shift toward lower carbon-intensity execution periods. Occupancy weighting is a timing diagnostic, not an energy-weighted emissions metric, and this single-seed comparison does not isolate the separate temporal effects of admission and placement.

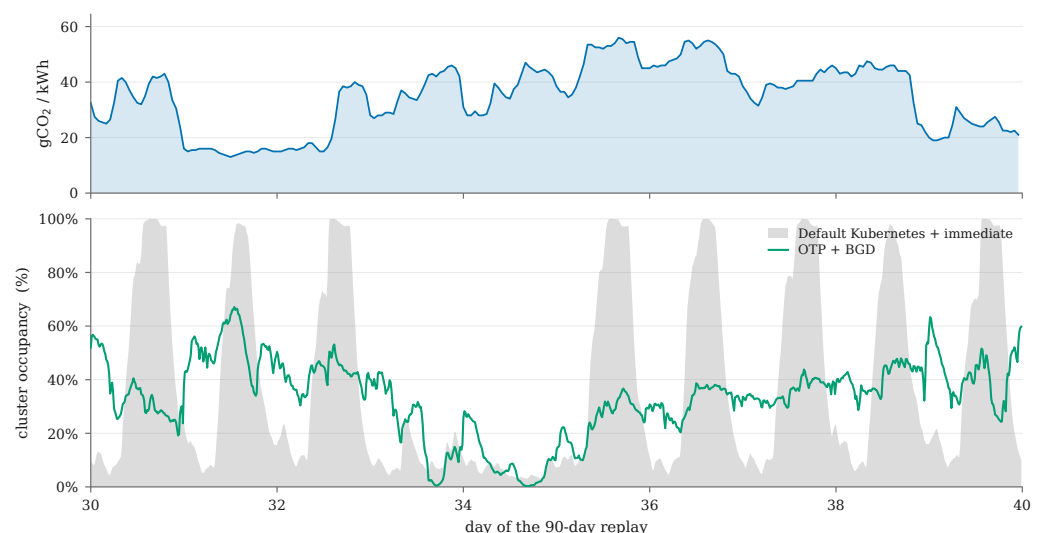


Figure 3. Ten days of the 90-day replay at 100 workers, 30% offered load, 24 h flexibility, and one paired seed. The (**upper**) panel shows realized French carbon intensity. The (**lower**) panel shows cluster occupancy for Default Kubernetes with immediate admission (shaded area) and OTP with BGD (line). The combined configuration reduces weekday peaks and increases occupancy in intervals with lower baseline demand.

5.3. Waiting Times and Deadline Compliance

At 30% offered load and 24 h flexibility, BGD schedules 43.03% of jobs for later release. The median wait remains zero, but the mean per-run 95th percentile reaches 22 h. At 50% load, the postponed fraction rises to 56.32% and the median wait is approximately six and a half hours (Table 6).

At the higher load, BGD records deadline-miss rates of 0.054% with Default Kubernetes and 0.038% with OTP. Immediate admission records no misses at the archived reporting precision in these 24 h scenarios. Additional carbon savings therefore come with both longer waits and a small observed increase in deadline misses.

Table 6. Service metrics in the NGT-informed simulation at 100 workers and 24 h flexibility. Postponed is the fraction assigned a release after arrival, including capacity-induced postponement. Waiting quantiles include actual capacity queueing. Miss is the percentage completing after the declared deadline.

Load	Configuration	Postponed (%)	Median (h)	P95 (h)	Miss (%)
30%	Default Kubernetes + immediate	0.00	0.00	1.57	0.000
30%	Default Kubernetes + BGD	43.03	0.00	22.00	0.000
30%	OTP + immediate	0.00	0.00	1.46	0.000
30%	OTP + BGD	43.03	0.00	22.00	0.000
50%	Default Kubernetes + immediate	0.00	0.00	7.87	0.000
50%	Default Kubernetes + BGD	56.32	6.50	23.18	0.054
50%	OTP + immediate	0.00	0.00	7.70	0.000
50%	OTP + BGD	56.32	6.50	23.18	0.038

Table 7 complements the median and P95 with mean and maximum waiting times from an archived diagnostic of the same 24 h configurations. Its 24 base records reproduce the original job counts and service and energy metrics within the original CSV rounding precision. Means are averaged across the three runs, whereas the maximum is the largest recorded wait across all three, not the mean of their maxima.

Table 7. Additional waiting statistics for the NGT-informed 100-worker, 24 h cases. These are empirical waits, not guaranteed upper bounds.

Load	Configuration	Mean (h)	Maximum (h)
30%	Default Kubernetes + immediate	0.20	2.80
30%	Default Kubernetes + BGD	5.48	23.83
30%	OTP + immediate	0.19	2.65
30%	OTP + BGD	5.48	23.83
50%	Default Kubernetes + immediate	2.08	9.35
50%	Default Kubernetes + BGD	8.79	23.95
50%	OTP + immediate	2.04	9.13
50%	OTP + BGD	8.77	23.95

Despite a zero median at 30% load, BGD's mean wait is 5.48 h. None of the 24 h BGD configurations meets a 12 h P95 waiting-time limit, despite their low deadline-miss rates. Completion deadlines and start-time waiting limits are different constraints, and a cap on planned release does not bound later capacity queueing. The evaluated policy does not enforce a separate waiting-time service-level objective.

Figure 4 shows the four configurations against waiting time, with horizontal segments extending from the median to the 95th percentile. It compares evaluated configurations rather than defining a complete Pareto frontier. The segments show substantial tail waits even when the median is zero.

Six-hour flexibility identifies a more constrained operating regime. At 50% load, Default Kubernetes with immediate admission records a deadline-miss rate of 17.92%, and OTP with immediate admission records 17.17%. These misses occur without deliberate deferral, showing that execution and capacity queueing already limit deadline compliance. Adding BGD to OTP increases the carbon reduction from 4.70% to 6.04%, but raises the miss rate to 21.06%. The additional carbon reduction is therefore accompanied by poorer deadline compliance. The larger 24 h reductions depend on allowing correspondingly greater completion flexibility.

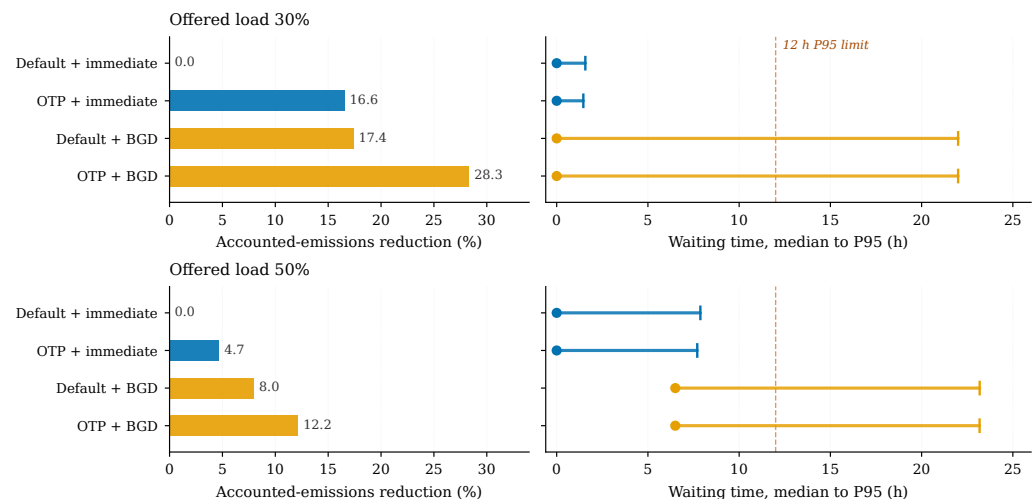


Figure 4. Carbon and waiting-time trade-offs in the NGT-informed simulation with 24 h flexibility. Circles mark median waits, and horizontal segments extend to 95th-percentile waits. Color identifies admission, and labels identify the complete configuration. Reported values average per-run statistics across three seeds.

5.4. Sensitivity to the Arrival Profile

Table 8 applies the same four configurations to the synthetic arrival benchmark. With 24 h flexibility, BGD under Default Kubernetes reduces accounted emissions by 3.43% at 30% load and 7.50% at 50%. The combined OTP configuration reaches 16.52% and 8.31%, respectively.

Table 8. Synthetic arrival benchmark using the same HEP execution catalog and 100-worker model. Entries are reductions (%) against this arm’s immediate Default Kubernetes baseline. Seed half-widths are bounded as in Table 5.

Configuration	30% Offered Load		50% Offered Load	
	6 h	24 h	6 h	24 h
Default Kubernetes + immediate	0.00	0.00	0.00	0.00
Default Kubernetes + BGD	1.76	3.43	2.54	7.50
OTP + immediate	11.31	11.31	1.63	1.63
OTP + BGD	12.75	16.52	3.66	8.31

Arrival profile has the largest effect on the lower-load admission-only comparison. At 30% offered load and 24 h flexibility, the admission-only reduction is 17.40% under NGT-informed arrivals against 3.43% in the synthetic benchmark, a factor of 5.1. At 50% load, the difference is smaller (7.98% against 7.50%). The corresponding lower-load baseline runtime multiplier is approximately 0.999 in the synthetic benchmark and 1.204 under NGT-informed arrivals. The normalized weekly arrival profiles have peak multipliers of 1.43 and 6.15, respectively, making the NGT peak approximately 4.3 times larger. Equal nominal load therefore does not imply equivalent execution conditions. Each reduction is interpreted against its own arrival-profile baseline.

Population-specific timing profiles also affect the results. At 30% offered load and 24 h flexibility, replacing synthetic arrivals with the pooled, batch-controller, user-profile, and CI-runner profiles increases BGD’s reduction relative to immediate OTP by 8.25, 8.59, 8.09, and 8.10 percentage points, respectively. This admission increment varies little across those profiles, but the combined reduction against immediate Default Kubernetes is more sensitive. With 24 h flexibility, the pooled and batch-controller profiles yield combined reductions of 28.34% and 25.37% at 30% load, and 12.17% and 11.19% at 50%

load. The pooled results thus describe aggregate platform timing, while the batch-controller profile provides a distinct sensitivity case.

Table 9 evaluates BGD relative to OTP's own immediate baseline over the resampled NGT profiles. The mean reductions are 13.50% and 7.41%, with wider shape variation in the higher-load case. At 30% load, the central 90% empirical band lies above the synthetic reference. At 50% load, the band includes that reference, indicating less separation relative to the observed shape variation, not equivalence between the profiles. These percentages have a different denominator from the combined reductions in Table 5.

Table 9. Arrival-shape sensitivity: BGD reduction relative to immediate OTP, with 100 workers and 24 h flexibility. Percentiles are over 30 resampled NGT shapes after averaging three seeds per shape.

Load	Synthetic (%)	Resampled Mean (%)	P05 (%)	P50 (%)	P95 (%)
30%	5.88	13.50	12.65	13.56	14.23
50%	6.79	7.41	5.88	7.37	8.69

5.5. Design Controls and Alternative Settings

The controls compare OTP with unrestricted consolidation and BGD with alternative admission policies under both arrival profiles.

Under NGT-informed arrivals, first-fit with immediate admission reduces accounted emissions by 15.16% at 30% offered load and 3.88% at 50%, against 16.56% and 4.70% for OTP. The paired differences favor OTP by 1.39 and 0.82 percentage points. In the synthetic benchmark, first-fit reduces emissions by 3.87% at 30% load and increases them by 4.10% at 50%, against reductions of 11.31% and 1.63% for OTP. OTP has lower accounted emissions in these immediate-admission comparisons, with a smaller advantage under the primary arrival profile.

The 40% relative-threshold variant differs from BGD in its relative threshold, absence of an absolute floor, and minimum-carbon ranking. BGD achieves a larger mean carbon reduction in seven of the eight NGT-informed combinations of placement, load, and flexibility, and an equal or shorter median wait in all eight. At 24 h flexibility, OTP with the variant reduces emissions by 26.01% at 30% load and 11.23% at 50%, against 28.34% and 12.17% for OTP with BGD. Median waits are 5.75 and 13.75 h for the variant, against 0.00 and 6.50 h for BGD.

These comparisons do not establish dominance across service metrics. At 30% load and six-hour flexibility under Default Kubernetes, the variant has a slightly larger carbon reduction (3.93% against 3.88%). At 50% load and six-hour flexibility under OTP, BGD increases the carbon reduction from 5.52% to 6.04%, but raises the 95th-percentile wait from 8.02 to 8.54 h and the deadline-miss rate from 17.12% to 21.06%. Across the eight NGT-informed combinations, BGD has higher miss rates in six and equal rates at the reported precision in two. In the synthetic benchmark's 50%, six-hour OTP case, the variant also has a larger carbon reduction (4.18% against 3.66%). The comparison therefore identifies trade-offs between complete policy configurations, not the independent benefit of either gate or the ranking rule.

A forecast-only greedy control selects the minimum-intensity interval without shared reservations. With Default Kubernetes and NGT-informed arrivals, it reduces emissions by 5.59% and 1.53% at six-hour flexibility, and by 22.53% and 7.58% at 24 h, for the lower and higher loads, respectively. In the six-hour synthetic scenarios, it instead increases emissions by 4.92% and 1.14%. Thus ungated forecast following is detrimental in the latter scenarios, but not in the corresponding NGT-informed comparisons. This profile dependence is consistent with the sensitivity to synchronized shifting identified in prior work [3,5], although the aggregate comparison does not isolate congestion as its cause.

A myopic perfect-foresight control selects each job's lowest-intensity window using realized intensity without shared reservations. It is a policy-level control rather than an optimal schedule or an upper bound on achievable reductions. Under NGT-informed arrivals with OTP and 24 h flexibility, it reduces emissions by 29.25% at 30% load and 9.11% at 50%, compared with 28.34% and 12.17% for BGD. BGD has a larger carbon reduction in four of the eight NGT-informed combinations, including three of the four using OTP. Perfect intensity information therefore does not make this myopic policy uniformly preferable to BGD. Because the comparison also changes the objective, benefit gates, and reservation mechanism, it cannot attribute the differences to reservations alone.

Repeating the four principal configurations on a 10-worker cluster at 24 h flexibility preserves the sign and ordering of their reductions. The combined configuration reaches 26.34% at 30% offered load and 11.39% at 50%, compared with 28.34% and 12.17% at 100 workers. This comparison tests sensitivity to cluster size, which remains a separate experimental factor rather than an additional replicate.

6. Discussion

6.1. What Determines the Carbon and Service Outcomes

Under the calibrated model, OTP reduces managed idle energy through consolidation, while BGD changes both intensity exposure and occupancy-dependent execution energy. Agreement across seeds characterizes simulation variation under fixed inputs and does not by itself establish robustness to the calibration and accounting assumptions of Section 4.3.

At 50% load, six-hour flexibility produces substantial deadline misses even with immediate admission. With 24 h flexibility, most jobs finish by their deadlines, but BGD still incurs long waits. Policy selection therefore requires both waiting and completion limits. Neither the two tested loads nor the 0.10 g gate and 50% occupancy target establish a general operating threshold.

Arrival timing also changes the comparison: the NGT-informed profile produces larger peaks and greater baseline slowdown than the synthetic profile at the same offered load (Section 5.4). Its larger admission benefit is assessed against a more contended baseline; equal offered load does not make the profiles equivalent.

6.2. Relation to Prior Systems and Deployment

BGD applies established savings-delay reasoning [4,5] to HEP admission while evaluating execution and idle-energy consequences. The controls do not compare BGD with CSD under identical conditions or isolate the contribution of each gate and ranking choice.

Unlike cross-regional CARA [9] or placement-focused systems [19,20], the deployment path changes admission within existing infrastructure. Shadow evaluation followed by bounded, opt-in enforcement should measure decision overhead, release accuracy, recovery, and forecast-versus-realized accounting. It requires agreed eligibility and deadlines. Evaluating OTP and realizing savings from powered-off nodes additionally require placement and worker-lifecycle integration; an empty node does not automatically stop consuming power.

6.3. Validity and Evaluation Scope

The arrival-estimation checks cover creation-timestamp counts, weekday-hour exposure, and inclusion of zero-arrival days. All reported NGT results use the 77-day extraction and the corresponding regenerated campaign. The independently specified synthetic arm reproduced its prior outputs exactly during regeneration, providing a control for changes outside arrival estimation. The synthetic and NGT profiles use different clock phases,

and their mean job counts differ slightly across arms, although policies within each arm process identical jobs.

The trace window runs from late June to early September, whereas the replay uses a winter carbon series. Applying the observed weekly profile to winter assumes that its activity pattern remains applicable across seasons. Day resampling represents within-window variation, not this seasonal uncertainty. The traced platform includes GPU and interactive workloads. Recorded CPU requests have a median of 30 cores, compared with 1 to 4 for the modeled HEP jobs, and almost all observed pod-time belongs to pods lasting more than 24 h. The scenario therefore uses the measured arrival shape, not the trace's resource-request or lifetime distribution.

The hardware calibration covers the reference configuration rather than fleet variability. Mixed-workload power is composed from archetype rates at shared occupancy. Calibration anchors reused in the catalog are not held-out validation data, and the per-cell variability parameters rest on 2 to 4 repetitions per cell on a single host. Independent concurrent-workload measurements are needed to validate the composition assumption.

Reliance on extrapolated response curves differs between configurations. CMS GEN-SIM, CMS RECO, and the Monte Carlo control have calibration points at 56 or 60 of 64 logical CPUs. ATLAS Sherpa and LHCb SIM, together representing 25% of jobs, are measured only to 32. An execution probe measures the fraction of these two archetypes' allocated CPU-time spent above that ceiling, excluding candidate scoring and event recomputation. In a single-seed, 10-worker diagnostic with 24 h flexibility and NGT-informed arrivals, this fraction is 71.7% for immediate Default Kubernetes and 5.5% for OTP with BGD at 30% offered load. At 50% load, the respective fractions are 88.1% and 54.5%. Under synthetic arrivals, the baseline fraction at 30% load is 4.5%. The combined configuration reduces extrapolation exposure, but at the higher NGT-informed load, more than half of the two archetypes' allocated CPU-time remains outside the measured range. OTP's occupancy target is a preference, not a cap.

Exposure alone does not determine the direction of model bias. A different high-occupancy power response would change active energy, while a different runtime response would also change occupancy, queueing, and completion times. These diagnostics therefore provide no bound on the reported savings and do not measure extrapolation exposure in the 100-worker campaign. Extending the ATLAS and LHCb sweeps above 32 logical CPUs would constrain this uncertainty.

Whole-server measurements could change both absolute savings and reduction percentages. For a fixed pair of schedules, the package-level avoided energy is a lower bound only if omitted components contribute nonnegative net avoided energy. For auxiliary loads proportional to powered node-time, this condition requires energy avoided during powered-off intervals to exceed any additional wake-up energy. Excluding those loads from the model does not establish that condition. A corresponding bound on avoided emissions additionally requires nonnegative intensity-weighted savings from the omitted components. No bound follows for reduction percentages, whose baseline also changes. Benefits without idle-power removal require a consistently accounted always-on-worker comparison.

The intensity integral clips at the end of the carbon series, so energy consumed after that boundary would be assigned no emissions. The archived bundle carries 97 days of intensity against a 90-day arrival horizon. The six-hour nominal duration limit and catalog runtime multipliers constrain individual execution, but neither these bounds nor a mean runtime multiplier bounds capacity-induced waiting, and the campaign outputs record no completion times. Coverage was therefore measured by instrumenting the simulator to report the last completion of each scenario. In the most exposed configuration, 50% offered load with 24 h flexibility under the policy that defers most, the last job completes

on day 90.98; immediate admission completes on day 90.25. Deferral lengthens the tail, as expected, but both leave six days of the series unused, so the reported campaign is covered. The margin is not guaranteed at higher offered loads, where capacity-induced waiting grows and the drain lengthens.

Job durations, workload weights, and flexibility are declared. Exact nominal duration is available to the simulated policy, whereas production decisions use estimates. All simulated jobs are independent and eligible for their specified flexibility. Tenant fairness, workflow constraints, and nondeferrable services require explicit operational policies. Parameter development and evaluation share the archived winter period, so independent-period validation is needed before recommending fixed thresholds.

The admission principle can be considered for other deferrable batch environments, but neither the reported reductions nor the ranking of policies is established outside these inputs. Other CPU architectures require new power and runtime response curves and worker-lifecycle parameters. GPU or heterogeneous clusters additionally require device-specific feasibility and performance models. Different workload mixtures, dependencies, and duration-estimation errors can change both reservations and deadline compliance. The 10- and 100-worker comparison retains homogeneous workers and the same model; it is not evidence of general fleet-scale behavior.

Changing electricity region changes the timing, level, and variability of carbon intensity, as well as the meaning of the production-based accounting signal [3]. Both fractional reductions and absolute benefits can change. In particular, the 0.10 g admission floor and occupancy target should be reassessed rather than transferred as fixed recommendations. Applying this method elsewhere requires local calibration and service constraints, not just substituting another carbon series.

Three seeds characterize limited conditional simulation variation. Neither the number of simulated jobs nor the number of nodes increases the number of independent experimental replicates. The summer NGT window and winter carbon series define a scenario rather than a synchronized measurement campaign. The French production indicator supports attributed operational emissions, not causal avoided marginal generation [30].

6.4. Reproducibility

Tables and figures derive from archived outputs and a frozen input bundle. Verification checks unique comparison records, equal work between paired policies, and matching source-file hashes. Archived execution probes assess extrapolation exposure and carbon-series coverage (Section 6.3). Forecast errors and the additional waiting statistics use archives without new simulation or training. Sections 3 and 4 specify the parameters needed for reproduction. Code is not publicly released; supporting material is subject to the Data Availability Statement and NGT redistribution permissions.

7. Conclusions

We presented benefit-gated deferral for realistic HEP batch workloads and evaluated its interaction with occupancy-targeted placement in a measurement-calibrated CERN case study. The primary scenario combines NGT-informed arrival patterns with measured CMS, ATLAS, and LHCb execution responses. Comparing admission alone with the combined configuration separates the admission benefit from the additional contribution of placement.

The evaluation supports three conclusions within the stated model. First, admission and placement change different components of active and idle accounting, so their contributions are not interchangeable. Second, the admission benefit changes with offered load and arrival profile. Third, larger carbon reductions require long tail waits, while

limited completion flexibility exposes capacity constraints even before deliberate deferral. These findings favor evaluating a scheduling configuration against its service requirements, rather than selecting it by its largest reported reduction.

The Kubernetes operator implements admission control for experimental deployment at CERN without requiring worker power-state control. Savings in deployment have not been established, and the quantitative conclusions remain conditional on the modeling scope in Section 6.3. Further evaluation should compare matched carbon-delay policies, extend the load and calibration ranges, and test release behavior and energy accounting in an opt-in deployment.

Author Contributions: Conceptualization, S.A.U.R.; methodology, S.A.U.R.; software, S.A.U.R.; validation, S.A.U.R., L.E.S.L., M.B. and R.R.; formal analysis, S.A.U.R.; investigation, S.A.U.R.; resources, R.R.; data curation, S.A.U.R.; writing—original draft preparation, S.A.U.R.; writing—review and editing, L.E.S.L. and M.B.; visualization, S.A.U.R.; supervision, L.E.S.L. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted within the CERN openlab programme and supported by CERN. No external grant funding was received.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The French grid carbon intensity records are publicly available from RTE Eco2mix. The simulation code and datasets are available from the corresponding author upon reasonable request. Use of the CERN Next Generation Triggers platform data for this study was approved by the NGT team. Access to these platform data is subject to confirmation of redistribution permissions and cannot be guaranteed.

Acknowledgments: During the preparation of this manuscript, the authors used OpenAI Codex for manuscript drafting, repository and dataset inspection, and verification of reported values against archived outputs. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

BGD	Benefit-gated deferral
CI	Carbon intensity, or continuous integration for runner namespaces
CSD	Carbon Savings-per-Delay
HEP	High-energy physics
NGT	Next Generation Triggers
OTP	Occupancy-targeted placement
PUE	Power usage effectiveness

References

1. Radovanović, A.; Koningstein, R.; Schneider, I.; Chen, B.; Duarte, A.; Roy, B.; Xiao, D.; Haridasan, M.; Hung, P.; Care, N.; et al. Carbon-Aware Computing for Datacenters. *IEEE Trans. Power Syst.* **2023**, *38*, 1270–1280. [\[CrossRef\]](#)
2. Piontek, T.; Haghshenas, K.; Aiello, M. Carbon Emission-Aware Job Scheduling for Kubernetes Deployments. *J. Supercomput.* **2024**, *80*, 549–569. [\[CrossRef\]](#)
3. Sukprasert, T.; Souza, A.; Bashir, N.; Irwin, D.; Shenoy, P. On the Limitations of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud. In *Proceedings of the Nineteenth European Conference on Computer Systems*; ACM: New York, NY, USA, 2024; pp. 924–941. [\[CrossRef\]](#)
4. Hanafy, W.A.; Liang, Q.; Bashir, N.; Souza, A.; Irwin, D.; Shenoy, P. Going Green for Less Green: Optimizing the Cost of Reducing Cloud Carbon Emissions. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*; ACM: New York, NY, USA, 2024; pp. 479–496. [\[CrossRef\]](#)

5. Hanafy, W.A.; Sukprasert, T.; Souza, A.; Irwin, D.; Shenoy, P. Untangling the Carbon-Cost Tradeoffs and Stampede Effect Challenges in Cloud Computing. *IEEE Trans. Comput.* **2026**, *75*, 2715–2729. [CrossRef]
6. ATLAS Collaboration. The Environmental Impact, Carbon Emissions and Sustainability of Computing in the ATLAS Experiment. *Eur. Phys. J. C* **2025**, *85*, 1397, Erratum in *Eur. Phys. J. C* **2026**, *86*, 304. [CrossRef]
7. Hall, S.; Micheli, F.; Belgioioso, G.; Radovanović, A.; Dörfler, F. Carbon-Aware Computing for Data Centers with Probabilistic Performance Guarantees. *arXiv* **2025**, arXiv:2410.21510. [CrossRef]
8. Doshi, N.; Shah, S. Night-Window Batching versus Carbon-Aware Scheduling for Clinical AI GPU Workloads. *arXiv* **2026**, arXiv:2606.01766. [CrossRef]
9. Liow, K.W.; Wang, R.; Zheng, R.; Akter, H.; Huang, B.; Subramaniam, H. CARA: Adaptive Scheduling for Carbon-Efficient Workload Orchestration in Distributed Cloud Systems. *Sustain. Comput. Inform. Syst.* **2026**, *51*, 101407. [CrossRef]
10. Beena, B.M.; Ranga, P.C.; Manideep, T.S.S.; Saragadam, S.; Karthik, G. A Green Cloud-Based Framework for Energy-Efficient Task Scheduling Using Carbon Intensity Data for Heterogeneous Cloud Servers. *IEEE Access* **2025**, *13*, 73916–73938. [CrossRef]
11. Xu, K.; Sun, D.; Tian, H.; Zhang, J.; Chen, K. GREEN: Carbon-efficient Resource Scheduling for Machine Learning Clusters. In *Proceedings of the 22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*; USENIX Association: Berkeley, CA, USA, 2025; pp. 999–1014.
12. Hanafy, W.A.; Wu, L.; Irwin, D.; Shenoy, P. CarbonFlex: Enabling Carbon-aware Provisioning and Scheduling for Cloud Clusters. *arXiv* **2025**, arXiv:2505.18357. [CrossRef]
13. Lechowicz, A.; Shenoy, R.; Bashir, N.; Hajiesmaili, M.; Wierman, A.; Delimitrou, C. Carbon- and Precedence-Aware Scheduling for Data Processing Clusters. *arXiv* **2025**, arXiv:2502.09717. [CrossRef]
14. Schweisgut, D.; Benoit, A.; Robert, Y.; Meyerhenke, H. Carbon-Aware Mapping and Scheduling for Deadline-Constrained Workflows. *arXiv* **2026**, arXiv:2605.27652. [CrossRef]
15. Benhari, A.; Trystram, D. Adaptive Carbon-Aware Scheduling Policies for HPC Systems. In *Proceedings of the Job Scheduling Strategies for Parallel Processing (JSSPP 2025)*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2026; Volume 16210, pp. 63–81. [CrossRef]
16. Kubernetes Authors. Scheduler Configuration. Available online: <https://kubernetes.io/docs/reference/scheduling/config/> (accessed on 5 September 2026).
17. Kubernetes Authors. Resource Bin Packing. Available online: <https://kubernetes.io/docs/concepts/scheduling-eviction/resource-bin-packing/> (accessed on 5 September 2026).
18. Kubernetes SIG Scheduling. Trimaran: Real Load Aware Scheduling. Available online: <https://scheduler-plugins.sigs.k8s.io/docs/plugins/trimaran/> (accessed on 5 September 2026).
19. Rao, W.; Li, H. Energy-aware Scheduling Algorithm for Microservices in Kubernetes Clouds. *J. Grid Comput.* **2025**, *23*, 2. [CrossRef]
20. Raith, P.; Rattihalli, G.; Dhakal, A.; Chalamalasetti, S.R.; Milojevic, D.; Frachtenberg, E.; Nastic, S.; Dustdar, S. Opportunistic Energy-Aware Scheduling for Container Orchestration Platforms Using Graph Neural Networks. In *Proceedings of the 2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*; IEEE: Piscataway, NJ, USA, 2024; pp. 299–306. [CrossRef]
21. Gouaouri, M.D.E.; Bagaa, M.; Ouahouah, S.; Ouameur, M.A.; Ksentini, A. Enabling Power-Awareness for Kubernetes Scheduling Through Multi-Criteria Optimization. In *Proceedings of the ICC 2025: IEEE International Conference on Communications*; IEEE: Piscataway, NJ, USA, 2025; pp. 1500–1505. [CrossRef]
22. Dong, H.; Singh, P.; Awad, Y.; George, F.; Narayanam, K.; Arora, S.; Appavoo, J. Towards Performance and Energy Aware Kubernetes Scheduler. *ACM Sigenergy Energy Inform. Rev.* **2025**, *5*, 69–75. [CrossRef]
23. Yang, J.; Saad, Z.; Wu, J.; Niu, X.; Leung, H.; Drew, S. A Survey on Task Scheduling in Carbon-Aware Container Orchestration. *arXiv* **2025**, arXiv:2508.05949. [CrossRef]
24. Ferreira, G.; Ilager, S. EcoKube: Simulating Carbon-Aware Scheduling Policies in Heterogeneous Edge-Cloud Environments. In *Proceedings of the 4th International Workshop on Testing Distributed Internet of Things Systems*; ACM: New York, NY, USA, 2026; pp. 7–12. [CrossRef]
25. Yacoub, M.F.; Abdel Raouf, A.E.; Gad, W.; Badr, N.L. GPS: A Lightweight Greedy-Predictive Scheduling Approach for Energy-Aware Multi-Region Cloud Computing. *Future Internet* **2026**, *18*, 399. [CrossRef]
26. Ruiiu, P.; Lagorio, A.; Rubattu, C.; Anedda, M.; Sanna, M.; Fadda, M. Edge-to-Cloud Continuum Orchestrator Based on Heterogeneous Nodes for Urban Traffic Monitoring. *Future Internet* **2025**, *17*, 574. [CrossRef]
27. Giordano, D.; Barbet, J.M.; Boccali, T.; Menéndez Borge, G.; Hollowell, C.; Innocente, V.; Lampl, W.; Michelotto, M.; Meinhard, H.; Ondris, L.; et al. HEPscore: A New CPU Benchmark for the WLCG. *EPJ Web Conf.* **2024**, *295*, 07024. [CrossRef]
28. HEPiX Benchmarking Working Group. Benchmarking Working Group: HEPscore23. Available online: <https://w3.hepik.org/benchmarking.html> (accessed on 5 September 2026).

29. Kepler Contributors. Power Attribution Guide. Available online: <https://github.com/sustainable-computing-io/kepler/blob/main/docs/developer/power-attribution-guide.md> (accessed on 5 September 2026).
30. Réseau de Transport d'Électricité. éCO₂mix: CO₂ Emissions per kWh of Electricity Generated in France. Available online: <https://www.rte-france.com/en/data-publications/eco2mix/eco2-emissions> (accessed on 5 September 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.