RESOURCE ISOLATION IS NOT PERFORMANCE ISOLATION: VIRTUAL FREENESS FOR DISAGGREGATED LLM SERVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Phase-disaggregated LLM serving isolates prefill and decode on independently provisioned GPU pools. But does resource isolation provide performance isolation? Production-grounded H100 analyses reveal four findings. First, prefill imbalance can delay decode-ready work and starve separately provisioned decode GPUs, propagating overload across the pipeline. Second, phase-local metrics can miss each phase's limiting resource, creating false headroom that amplifies imbalance. Third, even bottleneck-matched signals omit state needed to decide whether workers can accept additional work within the SLO. Finally, accurate load information alone is insufficient: recovery requires an intervention whose scope matches the imbalance without unnecessary execution cost. We introduce Virtual Freeness, which captures each phase's decision-critical state and expresses remaining worker capacity on a common SLO-feasibility scale. We also propose the minimum-sufficient-intervention principle, which selects the least disruptive action sufficient to restore capacity and escalates only when necessary. PHASOR implements these ideas through three coordinated controls: Micro routes requests by deadline-aware prefill risk; Meso restores worker-local KV allocatability through targeted migration; Macro converts donor GPUs under persistent pool-wide imbalance. We evaluate PHASOR against five serving baselines on an $8\times$H100 platform, using dense and MoE models from 8B to 110B. Across 12 model-workload combinations, PHASOR achieves the highest SLO-goodput and lowest P99 TTFT. Relative to the strongest prior system in each setting, it improves SLO-goodput by 6.7–29.0% and reduces P99 TTFT by 22.0–48.6%. It also achieves the lowest E2E P99 and TPOT in 11 of 12 settings. Holding signals and actuators fixed, its minimum-sufficient intervention policy improves SLO-goodput by 9.0–13.5% and reduces KV movement, GPU drain time, and role conversions by 21.3–52.3% relative to global-first and independent policies. Total runtime overhead is 2.8%. We will make the code and data open source upon acceptance.

1 INTRODUCTION

Large language model serving must accommodate heterogeneous requests and rapidly changing traffic while satisfying stringent time-to-first-token (TTFT) and time-per-output-token (TPOT) objectives (Agrawal et al., 2024; Zhong et al., 2024). Prefill processes input prompts through compute-intensive parallel execution, whereas decode repeatedly accesses model weights and an expanding KV cache during autoregressive generation. Phase-disaggregated systems therefore place prefill and decode on dedicated GPU pools, enabling phase-specific batching, scheduling, and provisioning (Zhong et al., 2024; Patel et al., 2024; Qin et al., 2025). Separating these phases removes their direct contention for the same device, but leaves a fundamental question unresolved: *does resource isolation provide performance isolation?*

Our empirical analysis in Section 2 explains why resource isolation does not ensure performance isolation and why recovery requires more than accurate load estimation. This analysis reduces effective cross-phase control to two coupled decisions: how much additional SLO-compliant work each P/D worker can accept, and which intervention can restore SLO-feasible capacity. We address the first decision by introducing *Virtual Freeness* (Φ), a control signal that models remaining

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

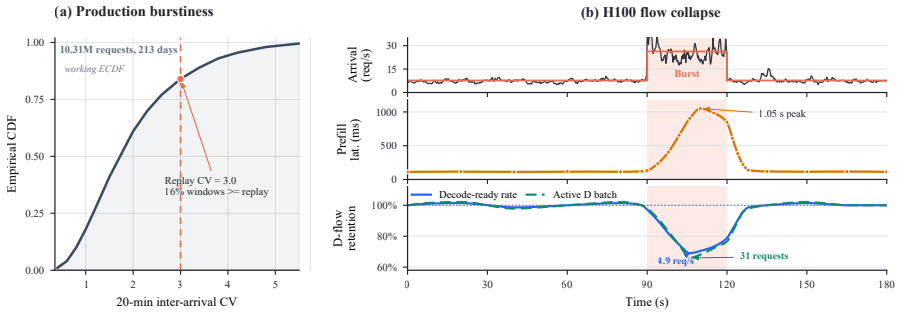


Figure 1: Production bursts and pipeline starvation. (a) Inter-arrival CV in 20-minute BurstGPT windows. (b) A production-derived shock raises prefill latency and reduces decode-ready work under fixed 4P:4D.

capacity through each phase's governing constraint. Prefill feasibility is determined by whether queued computation can finish before request deadlines, whereas decode feasibility is determined by whether usable KV capacity can admit projected demand. Φ maps these distinct constraints onto a common residual SLO-feasibility scale.

To address the second decision, we introduce the *minimum-sufficient-intervention principle*: select the least disruptive action sufficient to restore SLO-feasible capacity, and escalate only when a narrower action cannot resolve the imbalance. PHASOR integrates this principle with Virtual Freeness through three coordinated control scales. Micro redirects incoming requests to correct movable work skew without disturbing existing state. Meso relocates only the KV state needed to restore worker-local allocatability while preserving the P/D allocation. Macro converts a donor GPU only when a persistent pool-wide deficit cannot be repaired within the pressured pool. Together, these controls form a cost-ordered escalation path that matches intervention cost to the scope of capacity loss, closing the loop from accurate diagnosis to efficient recovery.

Our contributions are as follows:

- We present a production-grounded H100 analysis showing that resource isolation does not prevent pipeline-wide performance propagation. We identify the Utilization Phantom caused by mismatched and incomplete phase-local state, and show that correct diagnosis alone cannot determine an efficient recovery action.
- We introduce Virtual Freeness, which preserves the decision-critical constraints of prefill and decode while expressing their remaining SLO-feasible capacity on a common, directly comparable scale.
- We formulate the minimum-sufficient-intervention principle and instantiate it in PHASOR. Its Micro, Meso, and Macro controls restore capacity at the narrowest effective scope through deadline-aware routing, allocatability-aware KV repair, and directional GPU role conversion.
- We evaluate PHASOR across four model configurations, three workload regimes, and five serving baselines on H100 GPUs. Relative to the strongest baseline in each setting, Phasor improves SLO-goodput by 6.7–29.0% and reduces P99 TTFT by 22.0–48.6%. Controlled comparisons separately validate both design principles, and total runtime overhead is 2.8%.

2 MOTIVATION AND PROBLEM ANALYSIS

Our production-grounded analysis examines performance propagation despite physical isolation (RQ1), the missing decision state in phase-local signals (RQ2), and the recovery scope required by different feasibility losses (RQ3).

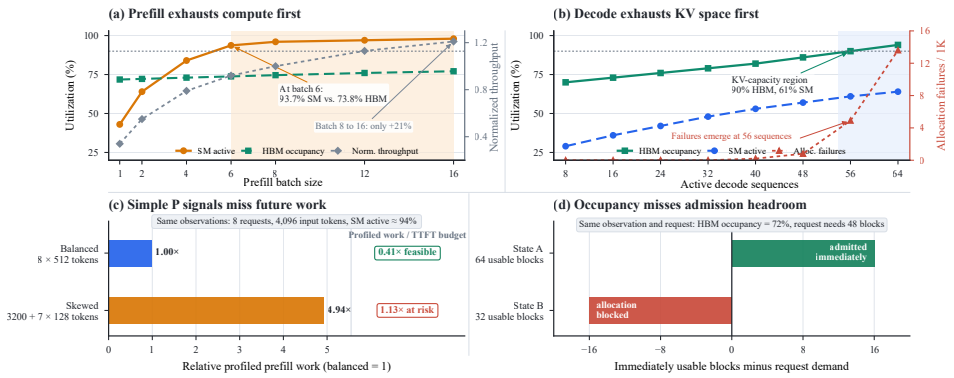


Figure 2: Phase-local signal failures. (a,b) Different limiting resources in prefill and decode. (c,d) Similar observations conceal different queued work or admission outcomes.

2.1 From Traffic Burst to Pipeline Starvation

BurstGPT (Wang et al., 2025b) contains 10.31 million requests collected from Azure OpenAI services over 213 days. Across 20-minute windows, 16% have an inter-arrival-time coefficient of variation (CV)$\geq$ 3.0 (Figure 1(a)). This recurring burstiness motivates our controlled replay with CV $\approx$ 3.0 for RQ1. We replay an MGP shock-and-recovery workload on 8$\times$H100 SXM 80GB GPUs using Qwen3.5-27B (Qwen Team, 2026) in FP16 with tensor parallelism (TP)=1 and a fixed 4P:4D allocation. Load rises from 7.7 to 26.3 requests/s during t = 90–120 s, then returns to 7.7. Keeping allocation and all scheduling policies fixed isolates the arrival shock from resource reconfiguration or policy adaptation. Figure 1(b) shows that a burst-induced performance disturbance is not confined to prefill. When the offered load increases by 3.4$\times$, prefill latency rises from approximately 116 ms to 1.05 s, while the decode-ready rate falls from 7.1 to 4.9 requests/s and the active decode batch contracts from 47 to 31 requests. Since the P/D allocation remains fixed and requests cannot enter decode before completing prefill, the decode contraction is caused by delayed prefill completions rather than direct resource contention. Decode consequently waits for ready work despite owning dedicated GPUs, producing **pipeline starvation**.

Insight 1. Physical separation removes resource contention, not performance propagation: phase-local overload can still become pipeline-wide starvation.

2.2 Phase-Local Signals Are Incomparable and Incomplete

Preventing pipeline starvation requires balancing P/D pressure before local overload propagates. RQ2 examines the first obstacle: mismatched and incomplete phase-local signals. In Figure 2(a), prefill reaches 93.7% SM utilization at batch size 6 while HBM occupancy remains at 73.8%; beyond batch size 8, doubling the batch yields only 21% additional throughput. Conversely, Figure 2(b) shows decode at 90% HBM occupancy and only 61% SM utilization at 56 active sequences, where allocation failures have already risen to 4.9 per 1,000 admissions. Each phase can therefore appear underloaded when viewed through a metric that misses its limiting resource, while the resulting P/D signals remain unsuitable for cross-phase scheduling.

Insight 2. Cross-phase scheduling cannot rely on separate phase-local metrics because they fail to capture how the limiting resource changes across the P/D pipeline, creating a Utilization Phantom. The resulting false headroom directs work toward overloaded workers and amplifies load imbalance.

Matching each phase with an appropriate local signal does not resolve the problem. In Figure 2(c), two prefill queues have the same eight requests, 4,096 input tokens, and approximately 94% SM activity, yet their request-length compositions produce a 4.94$\times$ difference in quadratic queued work. Their profiled demands consume 0.41$\times$ and 1.13$\times$ the same remaining TTFT budget. Figure 2(d) exposes the corresponding decode-side gap: under the same 72% aggregate HBM occupancy, a 48-block request is admitted with 64 immediately usable blocks but blocked with only 32. Thus, phase-appropriate signals still omit the queued work, deadline slack, and allocatability that determine actual worker load.

Insight 3. Combining phase-appropriate P/D signals is insufficient: they omit queued nonlinear work and TTFT slack in prefill, and allocatable KV capacity in decode, creating false headroom that reinforces imbalance.

These findings yield three requirements for an effective cross-phase control signal. First, it must be *phase-faithful*, preserving the nonlinear compute demand of prefill and the allocatable-space constraint of decode. Second, it must be *decision-complete*, incorporating the queued work, SLO slack, and usable capacity omitted by phase-local signals. Third, it must be *phase-comparable*, expressing P/D headroom with the same operational meaning so that the control plane can determine which phase faces greater SLO pressure. These properties enable cross-phase pressure diagnosis; RQ3 asks which intervention can turn that diagnosis into effective recovery.

2.3 Recovery Scope Must Match the Feasibility Loss

Three controlled states with similar SLO loss—movable work skew, KV-placement loss, and pool-wide shortage—differ in their narrowest effective repair. Appendix A reports the oracle comparison of routing, KV migration, and role conversion.

Insight 4. Similar SLO degradation can conceal different recoverability structures. Effective recovery depends on where usable capacity remains and whether the affected load is movable, bound to resident state, or exceeds the pressured pool's aggregate capacity.

Because the action sets are nested, a broader intervention can also repair a narrower failure. It nevertheless introduces state movement, draining, or role-conversion disruption that the narrower state does not require. Conversely, repeatedly applying an inexpensive local action cannot resolve a pool-wide capacity deficit. An effective controller must therefore avoid both forms of action-scope mismatch: over-intervening when a local repair is sufficient and under-intervening when the required capacity lies beyond the current scope.

This observation motivates the *minimum-sufficient-intervention principle*: select the least disruptive action that is sufficient to restore SLO-feasible capacity, and escalate only when a narrower action cannot do so. Phasor couples Virtual Freeness with this intervention principle in the next section.

3 Phasor

Phasor couples *Virtual Freeness*, which expresses heterogeneous P/D pressure on a common residual-feasibility scale, with *minimum-sufficient intervention*, which selects the narrowest effective recovery scope. We first define the signal, then describe its coordinated Micro, Meso, and Macro controls.

3.1 Virtual Freeness

Virtual Freeness maps each phase's limiting constraint and decision-critical state to a feasibility ratio. For time-bound prefill (T) and space-bound decode (S), it is $\Phi_x(w; r, t) = 1 - \rho_x(w; r, t)$. Here, ρ_x is predicted demand relative to the corresponding phase-specific feasibility boundary. Thus, $\Phi_x > 0$ denotes residual SLO-feasible headroom, $\Phi_x = 0$ the estimated feasibility boundary, and $\Phi_x < 0$ projected overload. The underlying resources remain different, but the residual-to-boundary meaning is shared across P and D. We write $\Phi(w; r, t)$ for candidate-aware evaluation and $\Phi(w; t) \equiv \Phi(w; \varnothing, t)$ for periodic worker-state evaluation.

Deadline-Aware Prefill Freeness For request q, its arrival time a_q and TTFT deadline $d_q = a_q + T_{\mathrm{SLO}}^{\mathrm{TTFT}}$ are known. Its remaining slack at time t is $s_q(t) = \max(d_q - t, \epsilon)$, where ϵ is a small positive constant that keeps the ratio well-defined after a deadline violation. When considering assigning request r to P worker w, Phasor evaluates the affected set $A_w(r, t) = Q_w(t) \cup \{r\}$, where $Q_w(t)$ is w's current prefill workload. For periodic evaluation, $A_w(\varnothing, t) = Q_w(t)$. For an empty affected set, we define the maximum feasibility ratio as zero, so $\Phi_T = 1$. Phasor estimates the request-level prefill time as $\widehat{\tau}_P(q) = \alpha l_{\mathrm{in}}(q)^2 + \beta P_m l_{\mathrm{in}}(q)$, where $l_{\mathrm{in}}(q)$ is the input-token count and P_m the model parameter count. The quadratic term captures attention computation, while the linear term captures parameter-dependent computation. Phasor fits α and β directly in time units for each deployed model, hardware, precision, kernel, and chunking configuration. They are model-

stack calibration constants rather than workload-specific tuning parameters. To obtain $\widehat{T}_{\mathrm{rem}}(q \mid w, r, t)$, Phasor evaluates the execution order induced by placing r on w under the engine's batching policy. It accumulates predicted batch or chunk durations from the current execution progress until q completes prefill, using the model-stack calibration (Appendix F.4). Comparing this completion time with $s_q(t)$ quantifies the request's deadline risk.

Allocatability-Aware Decode Freeness Let $B_{\mathrm{cap}}(w)$ denote the KV-block budget of D worker w, and let $F_{\mathrm{alloc}}(w, t)$ denote the blocks that are immediately allocatable under the current KV placement. Phasor further estimates $\widehat{B}_{\mathrm{req}}(w; r, t)$, the additional KV-block demand over a fixed look-ahead horizon H. This estimate uses observed sequence progress and a conservative output-length quantile. Observed decode progress determines projected token growth within H, bounded by the estimated remaining output; block rounding then gives additional demand beyond already allocated KV (Appendix F.4). When a candidate request is present, its projected demand is included; periodic evaluation uses $r = \varnothing$. The term $B_{\mathrm{cap}}(w) - F_{\mathrm{alloc}}(w, t)$ captures both resident KV allocation and capacity that is currently unusable under the worker-level placement. Adding $\widehat{B}_{\mathrm{req}}$ exposes an upcoming allocation deficit that aggregate memory occupancy cannot identify. Instantiating this definition with the two phase-specific feasibility ratios gives

$$\Phi(w; r, t) = \begin{cases} 1 - \max\limits_{q \in A_w(r,t)} \dfrac{\widehat{T}_{\mathrm{rem}}(q \mid w, r, t)}{s_q(t)}, & w \in \mathcal{P} \textbf{ (Time-Bound)}, \\ 1 - \dfrac{B_{\mathrm{cap}}(w) - F_{\mathrm{alloc}}(w, t) + \widehat{B}_{\mathrm{req}}(w; r, t)}{B_{\mathrm{cap}}(w)}, & w \in \mathcal{D} \textbf{ (Space-Bound)}. \end{cases} \tag{1}$$

The Time-Bound branch measures the worst normalized TTFT risk among requests affected by a prospective assignment. The Space-Bound branch measures projected KV demand against immediately usable block capacity. All arrival times, deadlines, prompt lengths, execution progress, and KV-block states are directly observable online. Phasor predicts only remaining prefill time and near-term KV growth; neither branch requires future arrivals or oracle service times. These signals are predicted margins for the modeled constraints: Φ_T tracks prefill deadline risk, while Φ_S tracks KV admission feasibility rather than directly predicting TPOT. Positive freeness alone is not a deterministic guarantee of joint TTFT/TPOT attainment. The branches retain phase-specific constraints and decision-critical state while sharing a normalized residual-feasibility scale with a common zero and direction. Equal values indicate equal normalized margins, not equal SLO-violation probabilities. Macro compares pool means on this scale to direct capacity toward the more pressured phase; RQ5 evaluates its benefit over equally informed, separately calibrated phase-local signals.

3.2 Minimum-Sufficient Multi-Scale Control

Phasor combines Virtual Freeness with scope-specific evidence to select the narrowest sufficient recovery action. Micro, Meso, and Macro are not sequential probes; each responds to evidence at its corresponding scope and time scale.

Micro: Deadline-Aware Risk Routing For each arrival r, Micro evaluates candidate-aware freeness on every P worker (Equation 1) and routes r to $w^\star = \arg\max_{w \in \mathcal{P}} \Phi(w; r, t)$. This minimizes the worst normalized TTFT risk over the affected requests; if all candidates have negative Φ, it selects the least infeasible worker. Predicted queue schedules are updated incrementally. Micro moves no existing computation or state (Algorithm 1).

Meso: Allocatability-Aware Space Repair Routing cannot recover decode capacity stranded by resident KV placement. For D worker d, the projected allocation deficit is $\Delta_B(d, t) = \max\{0, \widehat{B}_{\mathrm{req}}(d, t) - F_{\mathrm{alloc}}(d, t)\}$. Meso triggers when $\Delta_B(d, t) > 0$: projected demand exceeds immediately allocatable capacity, anticipating allocation failure. The positive θ_S is a post-repair headroom target, not a trigger threshold. Meso selects eligible KV state only until projected source $\Phi_S \geq \theta_S$, subject to destination feasibility and migration limits. This restores worker-local allocatability without changing the P/D allocation or compacting physical pages. TryMeso refreshes completed repairs and returns $g_D = \mathrm{true}$ only when a targeted relocation is expected to restore $\Phi_S \geq \theta_S$ or such a repair remains in flight. Migration keeps source decoding online during bulk

Algorithm 1 Phasor Multi-Scale Decisions

Require: Pools $\mathcal{P}, \mathcal{D}$; transient $\mathcal{M} \leftarrow \emptyset$
Require: Repair target $0 < \theta_S < 1$; $\theta_{\text{flow}}, K, \Delta t, T_{\text{cool}}$
$k_P, k_D \leftarrow 0$; $t_{\text{conv}} \leftarrow -\infty$
On each arrival r**:**
DISPATCH($r, \arg\max_{w \in \mathcal{P}} \Phi(w; r, t)$)
Every Δt**:**
$g_D \leftarrow$ TRYMESO($\mathcal{D}, \theta_S$)
$\Delta_\Phi \leftarrow$ MEANPHI($\mathcal{D}$) − MEANPHI($\mathcal{P}$)
if $\Delta_\Phi < -\theta_{\text{flow}}$ and g_D **then**
    $k_P, k_D \leftarrow 0$
else
    $(k_P, k_D) \leftarrow$ PERSIST($\Delta_\Phi, \theta_{\text{flow}}$)
    if $\max(k_P, k_D) \geq K$ and $t - t_{\text{conv}} \geq T_{\text{cool}}$ **then**
        $(\mathcal{X}, \mathcal{Y}) \leftarrow (\mathcal{D}, \mathcal{P})$ if $k_P \geq K$, else $(\mathcal{P}, \mathcal{D})$
        if $\mathcal{X}_{\text{eligible}} \neq \emptyset$ **then**
            $v \leftarrow \arg\max_{u \in \mathcal{X}_{\text{eligible}}} \Phi(u, t)$
            ASYNCCONVERT($v, \mathcal{X} \to \mathcal{M} \to \mathcal{Y}$)
            $k_P, k_D \leftarrow 0$; $t_{\text{conv}} \leftarrow t$
        end if
    end if
end if

transfer, with an atomic final handoff; Appendix E.2 details the snapshot, delta-copy, ownership, and failure-recovery protocol.

Macro: Directional Donor-Based Conversion Within-pool actions cannot repair a persistent shortage of aggregate phase capacity. Macro compares pool means through $\Delta_\Phi = \bar{\Phi}_D - \bar{\Phi}_P$: positive values indicate greater P pressure, and negative values greater D pressure. It defers D-pool expansion while TRYMESO reports sufficient or pending repair. Otherwise, a gap $|\Delta_\Phi| > \theta_{\text{flow}}$ for K consecutive control epochs permits one highest-Φ eligible donor to change role, subject to cooldown. Here, θ_{flow} is the pool-gap threshold, K its persistence requirement, and T_{cool} the minimum interval between conversions. Donors exclude workers undergoing KV repair, draining, conversion, or cooldown. The selected donor drains before conversion through transient $\mathcal{M}$, which holds no reserved capacity; Appendix E.3 gives the execution protocol.

Closed-Loop Coordination Algorithm 1 combines arrival-driven routing with periodic control every Δt. $\mathcal{P}$ and $\mathcal{D}$ are the active P/D pools; k_P, k_D count consecutive epochs of pressure in each pool, and t_{conv} records the last conversion start. DISPATCH routes the arriving request, and MEANPHI computes the pool mean. PERSIST increments the pressured pool's counter and resets the other, or resets both inside the threshold band. ASYNCCONVERT drains and converts donor v from $\mathcal{X}$ to recipient pool $\mathcal{Y}$, then activates it there. Worker eligibility and atomic handoff prevent conflicting actions; refreshed state closes the loop. Appendix E.1 provides the architecture and expanded algorithm.

4 EVALUATION

We evaluate end-to-end effectiveness (RQ4), the value of Virtual Freeness (RQ5), and control attribution, intervention efficiency, overhead, and robustness (RQ6–RQ7).

4.1 EXPERIMENTAL SETUP

Platform and models All experiments use one NVIDIA HGX node with eight H100 SXM 80GB GPUs connected by NVLink/NVSwitch. We evaluate Llama3.1-8B (Grattafiori et al., 2024) (FP16, TP=1), Qwen3.5-27B (Qwen Team, 2026) (FP16, TP=1), Qwen1.5-110B (Qwen Team, 2024) (FP8, TP=2), and Qwen3-30B-A3B (Yang et al., 2025) (BF16, TP=1). The 110B model forms four TP=2 workers initialized as 2P:2D; TP=1 models use 4P:4D. Phasor's Mixed Pool starts empty and contains only workers undergoing conversion.

Table 1: Performance across model scales and architectures under the unified H100 protocol.

Model	System	Prefill-heavy TTFT/TPOT ↓	Prefill-heavy E2E/GP ↓ / ↑	Decode-heavy TTFT/TPOT ↓	Decode-heavy E2E/GP ↓ / ↑	Burst TTFT/TPOT ↓	Burst E2E/GP ↓ / ↑
Llama3.1-8B	Semi-PD	4.45/21.5	8.18/40.0	0.270/40.9	73.59/17.2	2.23/28.0	10.73/29.0
	DistServe	4.04/20.1	7.85/43.3	0.284/36.9	65.83/18.2	2.42/27.8	10.15/27.3
	Splitwise	3.81/22.5	7.76/43.3	0.276/39.7	71.23/17.5	2.01/29.9	10.46/30.2
	Bullet	3.35/19.8	6.86/46.0	0.205/**33.1**	**57.69**/19.2	1.35/25.2	8.52/33.5
	Mooncake	3.85/20.8	5.59/43.2	0.225/35.2	62.88/18.5	1.60/26.0	9.55/31.2
	Phasor	**2.28/12.6**	**4.89/55.2**	**0.155**/34.5	60.58/**21.1**	**0.78/23.5**	**5.85/43.2**
Qwen3.5-27B	Semi-PD	3.05/58.0	11.65/13.0	0.185/58.0	117.19/2.7	2.00/76.0	22.56/15.8
	DistServe	3.11/58.6	11.12/12.8	0.205/58.2	118.82/2.7	2.34/83.1	24.65/13.2
	Splitwise	2.86/64.7	12.15/13.1	0.194/60.5	123.50/2.6	1.92/86.3	24.25/15.1
	Bullet	2.38/49.5	9.65/14.9	0.145/43.5	88.33/2.9	1.35/68.0	19.96/18.4
	Mooncake	2.70/53.0	10.23/13.9	0.158/46.8	94.65/2.8	1.55/69.5	20.72/17.1
	Phasor	**1.62/40.5**	**7.56/17.3**	**0.105/41.5**	**84.33/3.2**	**0.72/55.0**	**13.75/22.9**
Qwen1.5-110B	Semi-PD	4.90/78.0	17.34/5.9	0.310/114.5	236.33/1.2	2.75/136.0	36.68/7.3
	DistServe	5.40/82.2	17.25/5.5	0.360/122.0	251.65/1.1	3.44/157.2	42.37/5.6
	Splitwise	5.00/90.3	18.54/5.6	0.335/124.2	255.89/1.1	2.75/160.1	42.66/6.6
	Bullet	3.90/68.0	14.89/7.1	0.245/96.5	196.68/1.4	2.05/119.0	31.85/9.0
	Mooncake	3.60/65.0	13.68/7.4	0.220/88.0	177.56/1.5	1.85/112.0	29.68/9.2
	Phasor	**2.35/55.0**	**10.23/8.6**	**0.170/84.0**	**171.62/1.6**	**0.95/98.0**	**24.22/11.6**
Qwen3-30B-A3B (MoE)	Semi-PD	2.65/28.5	5.72/36.9	0.145/25.5	50.62/4.1	1.28/29.6	7.98/29.6
	DistServe	2.75/29.2	5.65/35.9	0.162/26.0	52.56/4.0	1.51/32.8	8.98/24.3
	Splitwise	2.52/32.1	5.32/36.7	0.153/26.9	53.33/4.0	1.24/33.9	8.75/28.0
	Bullet	2.45/27.8	4.21/37.8	0.123/23.6	46.51/4.2	1.04/27.6	7.80/30.4
	Mooncake	2.52/27.0	4.32/37.4	0.118/22.9	44.56/4.3	1.10/26.8	6.65/31.2
	Phasor	**1.82/24.0**	**3.65/42.8**	**0.092/20.7**	**41.66/4.8**	**0.61/23.5**	**5.60/37.2**

TTFT/E2E: s; TPOT: ms; GP: requests/s. Values are three-run means; bold marks the best result per model and workload.

Systems and fairness We compare Phasor with DistServe (Zhong et al., 2024) and Splitwise (Patel et al., 2024) for phase-disaggregated provisioning and scheduling, Semi-PD (Hong et al., 2025) and Bullet (Lin et al., 2026) for P/D coexecution, and Mooncake (Qin et al., 2025) for KV-centric serving. All use the common vLLM substrate and H100 testbed, sharing the model checkpoint, precision, parallelism, GPU budget, SLOs, batching limit, and 85% HBM limit within each model group. Prefix caching and speculative decoding are disabled, active requests are capped at 512, and chunked prefill is evaluated separately. Appendix B.2 specifies retained baseline mechanisms and model-stack adaptations.

Workloads We derive all requests from first-turn ShareGPT conversations and preserve their empirical length variation. Prefill-heavy, Decode-heavy, and Burst traces have mean input/output lengths of 2,048/128, 128/2,048, and 1,024/256 tokens, respectively; recorded response lengths determine generated-token counts, with EOS termination disabled. Steady traces use Poisson arrivals, while Burst uses a Markovian group process fitted to BurstGPT (Wang et al., 2025b) with CV ≈ 3 and a peak at 90–120 s. For the models ordered above, Prefill-heavy and Decode-heavy loads are $(60, 18, 9, 45)$ and $(24, 3.4, 1.65, 5)$ requests/s; Burst loads are $(20{\to}60{\to}20)$, $(7.7{\to}26.3{\to}7.7)$, $(4{\to}13.2{\to}4)$, and $(12{\to}40{\to}12)$ requests/s. Steady runs use a 60-s warm-up and 600-s measurement window; Burst runs last 180 s and report the 90–120 s peak. Full workload and software details appear in Appendix B.

SLOs and methodology For each model–workload pair, we set the TTFT and TPOT SLOs to $7.5\times$ their isolated single-worker P90 latencies before system comparison and then hold them fixed across systems. SLO-goodput counts only requests satisfying both SLOs. We report P99 TTFT, TPOT, and E2E latency together with SLO-goodput, averaged over three runs; compared systems replay identical request sequences and arrival timestamps.

Table 2: Controlled attribution on Qwen3.5-27B (mean±SD over three runs). Cells report E2E P99 (s) and SLO-goodput (req/s); the two Full rows share the same runs.

	Prefill-heavy		Decode-heavy		Burst	
Variant	P99 ↓	GP ↑	P99 ↓	GP ↑	P99 ↓	GP ↑
Decision-state comparison with identical controllers						
Token+Occ	9.60 ± 0.32	13.2 ± 0.3	111.00 ± 3.70	2.75 ± 0.08	19.00 ± 0.84	7.90 ± 0.30
DirectHW+Occ	10.40 ± 0.42	12.4 ± 0.3	117.00 ± 4.10	2.62 ± 0.08	20.50 ± 0.96	7.40 ± 0.30
RichPhase	7.68 ± 0.25	15.7 ± 0.3	101.63 ± 2.90	3.05 ± 0.06	17.26 ± 0.69	8.74 ± 0.30
Full Φ	$\mathbf{7.00 \pm 0.22}$	$\mathbf{16.7 \pm 0.2}$	$\mathbf{94.50 \pm 2.60}$	$\mathbf{3.18 \pm 0.06}$	$\mathbf{15.40 \pm 0.55}$	$\mathbf{9.70 \pm 0.20}$
Cumulative multi-scale control						
Base	12.00 ± 0.46	10.8 ± 0.3	164.00 ± 5.80	2.15 ± 0.08	24.50 ± 1.15	6.30 ± 0.40
+Micro	10.30 ± 0.36	12.5 ± 0.3	162.00 ± 5.60	2.20 ± 0.07	22.50 ± 1.02	6.90 ± 0.30
+Meso	9.70 ± 0.34	12.8 ± 0.3	123.00 ± 3.90	2.65 ± 0.07	18.50 ± 0.78	8.10 ± 0.30
+Macro (Full)	$\mathbf{7.00 \pm 0.22}$	$\mathbf{16.7 \pm 0.2}$	$\mathbf{94.50 \pm 2.60}$	$\mathbf{3.18 \pm 0.06}$	$\mathbf{15.40 \pm 0.55}$	$\mathbf{9.70 \pm 0.20}$

4.2 End-to-End Effectiveness (RQ4)

Overall performance and model scale Across four models, three workloads, and five baselines (Table 1), Phasor achieves the highest SLO-goodput and lowest P99 TTFT in all 12 settings, improving goodput by 6.7–29.0% and reducing TTFT by 22.0–48.6% over the strongest baseline in each setting. It also leads E2E P99 and TPOT in 11 settings, reducing them by 3.3–31.3% and 4.5–36.4%, respectively, when it leads, while raising joint SLO attainment by 8.0–18.0 percentage points. These gains span dense and MoE models; Phasor also outperforms DistServe and Splitwise on all four metrics in every setting. Appendix C.1 details the model-scale and workload comparisons.

Workload-dependent boundary Phasor leads every reported metric under Prefill-heavy and Burst traffic. The exception is Llama3.1-8B under steady Decode-heavy traffic: Bullet achieves 4.1% lower TPOT and 4.8% lower E2E P99, while Phasor reduces TTFT by 24.4% and improves goodput by 10.0%. This stable decode-dominated regime favors fine-grained coexecution; Phasor gains most under queued prefill work, KV allocatability loss, or changing P/D pressure. Appendix C.1 gives the detailed boundary analysis. The load sweep further shows widening goodput gains near saturation (Appendix C), extending the comparison beyond a single operating point.

4.3 Value of Virtual Freeness (RQ5)

Controlled comparison We isolate the benefits of decision-complete phase-specific state over conventional signals and of a common feasibility scale over equally rich local estimates. All variants use the same Micro, Meso, and Macro controllers, actuators, and tuning budget, differing only in decision state. Token+Occ uses token headroom and memory occupancy; DirectHW+Occ uses SM activity and occupancy. These conventional signal designs are independent baselines, not ablations of Virtual Freeness (Appendix G). RichPhase is a stronger baseline: it receives the same nonlinear queued-work estimate and remaining TTFT slack as Φ_T, together with the same projected KV demand and allocatable capacity as Φ_S. However, it retains separately calibrated P-risk and D-risk scales and determines Macro direction through normalized threshold exceedance. Full Φ maps both phase-specific estimates to normalized residual feasibility, providing a shared zero boundary and directly comparable normalized margins for Macro's pool-pressure comparison.

Value beyond rich phase-specific estimators Despite identical phase-specific information and independent P/D calibration, RichPhase trails Full Φ: the common residual-feasibility scale reduces E2E P99 by 7.0–10.8% and improves goodput by 4.3–11.0%. Together with the conventional-signal comparisons, these results support both decision-complete local state and a common control scale.

4.4 Control Attribution and Practicality (RQ6–RQ7)

Does intervention selection matter Table 2 reports cumulative control gains; Figure 3(a) shows Meso's effect on allocatable KV capacity. To isolate intervention selection, we compare four policies

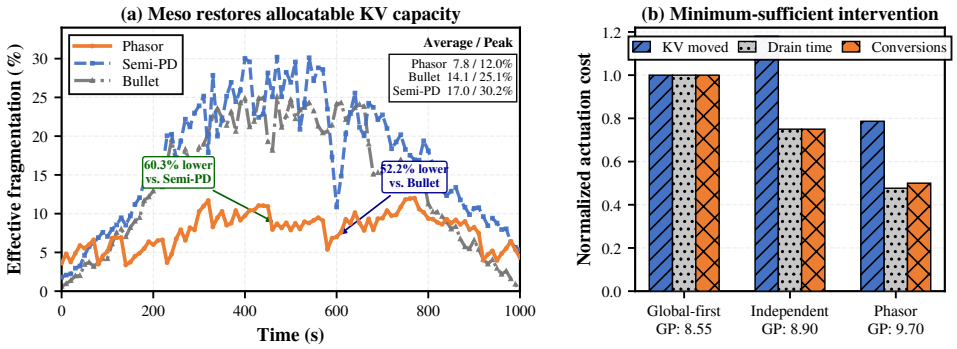


Figure 3: KV recovery and intervention cost on H100. (a) Effective KV fragmentation during a 1000-s production-derived MGP burst; percentage annotations refer to peaks. (b) Qwen3.5-27B Burst costs, each normalized to Global-first; labels show SLO-goodput (req/s).

with the same Full Φ signal and identical routing, migration, draining, and conversion implementations. Local-only disables Macro and consequently cannot repair a pool-wide shortage, reaching only 8.1 ± 0.3 requests/s and an E2E P99 of 18.50 ± 0.78 s under Burst. Among policies retaining all actions, Global-first converts a worker before attempting narrower repair, Independent permits all triggers without cross-scale precedence, and Phasor selects the narrowest sufficient action before escalating. Phasor reaches 9.70 ± 0.20 requests/s while moving 32.8 ± 2.4 GB, draining for 6.1 ± 0.5 GPU-s, and performing 4.0 ± 0.0 conversions. Relative to Global-first/Independent, respectively, it reduces KV movement by 21.3%/33.3%, drain time by 52.3%/36.5%, and conversions by 50.0%/33.3%, while reducing E2E P99 by 10.5%/7.8% and improving goodput by 13.5%/9.0%. Each scale restores a distinct form of lost SLO-feasible capacity. Explicitly selecting the narrowest sufficient recovery scope avoids ineffective local control and unnecessary global actuation while preserving useful system state.

Overhead and robustness Phasor incurs $2.8 \pm 0.2\%$ total control and actuation overhead: Micro decisions take 11.7 ± 0.5 μs and Meso's P99 handoff pause is 8.1 ± 1.2 ms. It remains effective with chunked prefill and robust across the evaluated calibration and controller settings. Table 3 and Appendix F report the full overhead breakdown, chunked-prefill and MoE results, and parameter sensitivity.

5 Related Work and Positioning

DistServe and Splitwise specialize P/D placement and provisioning (Zhong et al., 2024; Patel et al., 2024); Semi-PD and Bullet adaptively coordinate intra-GPU resources (Hong et al., 2025; Lin et al., 2026). Phasor addresses the cross-worker starvation that can persist despite efficient phase execution, coupling capacity diagnosis with recovery-scope selection. TokenScale's Token Velocity unifies prefill, network, and decode rates to guide autoscaling, with Convertible Decoders absorbing bursts (Lai et al., 2025). Phasor instead normalizes deadline-constrained prefill work and allocatable decode KV capacity into residual-feasibility margins. This distinction links the signal to whether request routing, local repair, or pool reallocation is sufficient; Sections 4.3 and 4.4 separately evaluate signal quality and intervention selection. Mooncake couples KV-centric disaggregation with SLO-aware scheduling (Qin et al., 2025); FlowKV and BanaServe combine state movement with load-aware scheduling or resource rebalancing (Li et al., 2025; He et al., 2026). Phasor's contribution is the decision rule connecting these recovery scopes: targeted KV repair precedes pool conversion when it can restore local feasibility, while persistent pool imbalance permits escalation. Appendix H expands these comparisons and their scope.

6 Conclusion

Phasor couples Virtual Freeness with minimum-sufficient intervention to restore SLO-feasible capacity across disaggregated P/D workers, improving goodput and tail latency while avoiding unnecessary state movement and role conversion in the evaluated workloads.

AI USE STATEMENT

We used generative AI tools solely to improve the grammar, wording, and readability of this manuscript. All AI-assisted revisions were reviewed by the authors to ensure that they preserved the intended meaning and technical accuracy. The authors take full responsibility for the final content of this manuscript.

REFERENCES

Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, Alexey Tumanov, and Ramachandran Ramjee. Taming Throughput-Latency tradeoff in LLM inference with Sarathi-Serve. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 117–134, Santa Clara, CA, July 2024. USENIX Association. ISBN 978-1-939133-40-3. URL https://www.usenix.org/conference/osdi24/presentation/agrawal.

Jingqi Feng, Yukai Huang, Rui Zhang, Sicheng Liang, Ming Yan, and Jie Wu. Windserve: Efficient phase-disaggregated llm serving with stream-based dynamic scheduling. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, ISCA '25, pp. 1283–1295, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712616. doi: 10.1145/3695053.3730999. URL https://doi.org/10.1145/3695053.3730999.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang

Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi (Jack) Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu (Sid) Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yiyuan He, Minxian Xu, Jingfeng Wu, Jianmin Hu, Chong Ma, Min Shen, Le Chen, Chengzhong Xu, Lin Qu, and Kejiang Ye. BanaServe: Unified KV cache and dynamic module migration for balancing disaggregated LLM serving in AI infrastructure. *Software: Practice and Experience*, 56(4):424–444, 2026. doi: 10.1002/spe.70054. URL https://doi.org/10.1002/spe.70054.

Ke Hong, Lufang Chen, Zhong Wang, Xiuhong Li, Qiuli Mao, Jianping Ma, Chao Xiong, Guanyu Wu, Buhe Han, Guohao Dai, Yun Liang, and Yu Wang. semi-pd: Towards efficient llm serving via phase-wise disaggregated computation and unified storage, 2025. URL https://arxiv.org/abs/2504.19867.

Cunchen Hu, Heyang Huang, Junhao Hu, Jiang Xu, Xusheng Chen, Tao Xie, Chenxi Wang, Sa Wang, Yungang Bao, Ninghui Sun, and Yizhou Shan. Memserve: Context caching for disaggregated llm serving with elastic memory pool, 2024. URL https://arxiv.org/abs/2406.17565.

Yibo Jin, Tao Wang, Huimin Lin, Mingyang Song, Peiyang Li, Yipeng Ma, Yicheng Shan, Zhengfan Yuan, Cailong Li, Yajing Sun, Tiandeng Wu, Xing Chu, Ruizhi Huan, Li Ma, Xiao You, Wenting Zhou, Yunpeng Ye, Wen Liu, Xiangkun Xu, Yongsheng Zhang, Tiantian Dong, Jiawei Zhu, Zhe Wang, Xijian Ju, Jianxun Song, Haoliang Cheng, Xiaojing Li, Jiandong Ding, Hefei Guo, and Zhengyong Zhang. P/d-serve: Serving disaggregated large language model at scale, 2024. URL https://arxiv.org/abs/2408.08147.

Ruiqi Lai, Hongrui Liu, Chengzhi Lu, Zonghao Liu, Siyu Cao, Siyang Shao, Yixin Zhang, Luo Mai, and Dmitrii Ustiugov. TokenScale: Timely and accurate autoscaling for disaggregated LLM serving with token velocity, 2025. URL https://arxiv.org/abs/2512.03416.

Weiqing Li, Guochao Jiang, Xiangyong Ding, Zhangcheng Tao, Chuzhan Hao, Chenfeng Xu, Yuewei Zhang, and Hao Wang. FlowKV: A disaggregated inference framework with low-latency KV cache transfer and load-aware scheduling, 2025. URL https://arxiv.org/abs/2504.03775.

Zejia Lin, Hongxin Xu, Guanyi Chen, Zhiguang Chen, Yutong Lu, and Xianwei Zhang. Bullet: Boosting gpu utilization for LLM serving via dynamic spatial-temporal orchestration. In *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '26, pp. 290–306, 2026. doi: 10.1145/3779212.3790135. URL https://doi.org/10.1145/3779212.3790135.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. Splitwise: Efficient generative llm inference using phase splitting. In *Proceedings of the 51st Annual International Symposium on Computer Architecture*, ISCA '24, pp. 118–132. IEEE Press, 2024. ISBN 9798350326581. doi: 10.1109/ISCA59077.2024.00019. URL https://doi.org/10.1109/ISCA59077.2024.00019.

Ruoyu Qin, Zheming Li, Weiran He, Jialei Cui, Feng Ren, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: Trading more storage for less computation — a KVCache-centric architecture for serving LLM chatbot. In *23rd USENIX Conference on File and Storage Technologies (FAST 25)*, pp. 155–170, Santa Clara, CA, February 2025. USENIX Association. ISBN 978-1-939133-45-8. URL https://www.usenix.org/conference/fast25/presentation/qin.

Qwen Team. Qwen1.5-110B: The first 100b+ model of the Qwen1.5 series, April 2024. URL https://qwenlm.github.io/blog/qwen1.5-110b/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Biao Sun, Ziming Huang, Hanyu Zhao, Wencong Xiao, Xinyi Zhang, Yong Li, and Wei Lin. Llumnix: Dynamic scheduling for large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 173–191. USENIX Association, 2024. URL https://www.usenix.org/conference/osdi24/presentation/sun-biao.

Chao Wang, Pengfei Zuo, Zhangyu Chen, Yunkai Liang, Zhou Yu, and Ming-Chang Yang. Prefill-decode aggregation or disaggregation? unifying both for goodput-optimized llm serving, 2025a. URL https://arxiv.org/abs/2508.01989.

Yuxin Wang, Yuhan Chen, Zeyu Li, Xueze Kang, Yuchu Fang, Yeju Zhou, Yang Zheng, Zhenheng Tang, Xin He, Rui Guo, Xin Wang, Qiang Wang, Amelie Chi Zhou, and Xiaowen Chu. Burstgpt: A real-world workload dataset to optimize llm serving systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 5831–5841, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737413. URL https://doi.org/10.1145/3711896.3737413.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 193–210, Santa Clara, CA, July 2024. USENIX Association. ISBN 978-1-939133-40-3. URL https://www.usenix.org/conference/osdi24/presentation/zhong-yinmin.

A Recovery-Scope Diagnostic

This section supports Section 2.3: the offline diagnostic separates the capacity losses that motivate the three online control scopes. The 90% recovery criterion below defines sufficiency for this diagnostic; it is distinct from the online Meso headroom target θ_S. RQ2 establishes the state required to diagnose cross-phase pressure, but correct diagnosis alone does not determine an effective recovery action. To answer RQ3, we examine whether similar SLO degradation can be repaired at the same control scope. On the same controlled H100 deployment, we construct three states that each reduce joint SLO attainment by approximately 15 percentage points, but differ in where usable capacity remains. In *work skew*, the pressured pool retains sufficient aggregate capacity and queued work can be redirected. In *KV-placement loss*, aggregate decode capacity is also sufficient, but usable capacity is stranded behind resident KV state. In *pool-level deficit*, demand exceeds the aggregate capacity of the pressured pool, while the opposite pool retains sufficient donor headroom. These states therefore present similar external symptoms but fundamentally different recovery conditions. To separate recoverability from the behavior of any particular online policy, Figure 4 evaluates three nested offline-oracle action sets. A_1 permits request assignment without moving resident state; A_2 additionally permits within-pool resident-state relocation; and A_3 further permits cross-pool role transfer. The oracle observes the complete request trace, service times, and allocation state, and does not use Virtual Freeness or Phasor's online controller. For action set A_k, we measure normalized SLO-goodput recovery as

$$R_k = \frac{G_k - G_{\text{fault}}}{G_{\text{healthy}} - G_{\text{fault}}}, \tag{2}$$

where G_{healthy}, G_{fault}, and G_k denote SLO-goodput before degradation, after degradation, and after the oracle intervention, respectively. We regard an action set as sufficient when its mean recovery reaches 90%. Figure 4(b) reveals three distinct recoverability structures. Under work skew, assignment alone restores $92 \pm 3\%$ of the lost SLO-goodput because spare capacity exists and the affected work is movable. Under KV-placement loss, assignment recovers only $18 \pm 4\%$, whereas adding resident-state relocation raises recovery to $91 \pm 3\%$. Here, capacity exists but cannot be reached without moving the state that binds requests to the pressured worker. Under pool-level deficit, assignment and within-pool relocation recover only $9 \pm 3\%$ and $16 \pm 4\%$, respectively; neither can

create capacity inside an already saturated pool. Cross-pool role transfer is necessary among the tested action sets for this constructed deficit and restores $93 \pm 3\%$. These oracle results establish recoverability under the specified states, not an optimality guarantee for the online controller.

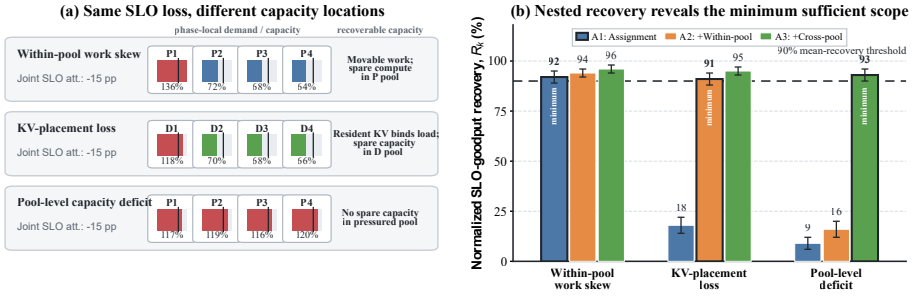


Figure 4: Recovery scope required by matched SLO losses. (a) Similar degradation arises from worker skew, stranded KV capacity, or pool-wide shortage. (b) Offline oracles progressively add request assignment (A1), within-pool relocation (A2), and cross-pool role transfer (A3).

B Detailed Experimental Setup

This section supports Section 4.1. The software stack comprises Ubuntu 22.04.5, NVIDIA driver 570.86.15, CUDA 12.8, PyTorch 2.8.0, NCCL 2.27.3, FlashAttention 2.8.3, and vLLM 0.12.0. For Qwen1.5-110B, all systems apply the same dynamic FP8 W8A8 path to a shared BF16 checkpoint: linear weights and activations use E4M3 with per-tensor static and dynamic scaling, respectively, while the LM head, nonlinear operators, and KV cache remain FP16.

B.1 Workloads and control settings

The main end-to-end protocol, including model order, request-length distributions, arrival rates, and measurement windows, is specified in Section 4.1. Load sweeps use 0.6, 0.8, 0.9, 1.0, and $1.1\times$ the reference load. Phasor uses a 50-ms control epoch, 200-ms persistence ($K = 4$ at the default epoch), and 1-s conversion cooldown. The chunked-prefill and MoE experiments use the separate workloads specified in Appendices F.2 and F.3.

B.2 Additional Reproducibility Controls

All experiments run on one NVIDIA HGX node with eight H100 SXM 80GB GPUs connected through NVLink/NVSwitch. A worker denotes either one GPU or one tensor-parallel group. All compared systems use the same checkpoint, numerical precision, parallelism, eight-GPU budget, tokenized request trace, arrival timestamps, and SLOs within each model group. We use CUDA 12.8, fix GPU power and clock settings across systems, and disable prefix caching and speculative decoding. The KV-cache memory fraction and maximum sequence length are also held constant.

Baseline implementation and adaptations. DistServe retains SLO-aware disaggregated provisioning, and Splitwise retains phase-specific scheduling and resource management. Semi-PD retains unified P/D KV storage and MPS isolation; Bullet retains spatial-temporal coexecution, `libsmctrl` SM partitioning, and its online model; Mooncake retains separate P/D workers, Transfer Engine 0.3.12, and load-aware placement. Phasor uses the same kernels, batching engine, and KV allocator, adding only Virtual Freeness and its control policies. Extensions for the three Qwen models affect only the common model-loader and kernel interfaces, leaving each baseline's scheduling, resource partitioning, KV placement, and reconfiguration policies unchanged. Phasor begins with the P/D allocation listed in Section 4.1. The Mixed state contains no reserved GPU; it is initially empty and is used only while a drained donor changes role. For each of three seeds, every system replays exactly the same request contents and arrival timestamps. Different seeds generate independent traces. We report the mean and sample standard deviation across the three runs.

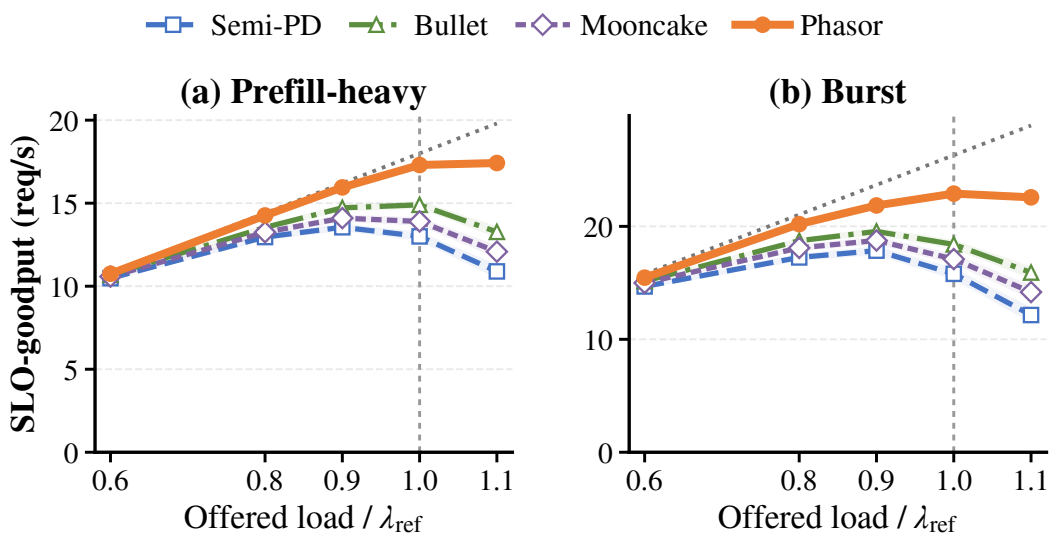


Figure 5: Capacity boundaries on Qwen3.5-27B (mean±SD over three runs). Dotted diagonals denote offered load; dashed verticals mark λ_{ref}.

B.3 Metric Definitions

TTFT is measured from request arrival to the first generated token. TPOT is the average inter-token latency of each request, and E2E latency spans request arrival to its final token. We report P99 over requests arriving within the measurement window. A request is SLO-compliant only if it satisfies both its TTFT and TPOT thresholds. SLO attainment is the fraction of compliant requests. Let $\mathcal{R}_{\text{measure}}$ denote the requests in the measurement cohort, S_T and S_D the TTFT and TPOT thresholds, and T_{measure} the measurement duration. SLO-goodput is the number of compliant requests per second:

$$\text{Goodput} = \frac{|\{r \in \mathcal{R}_{\text{measure}} : \text{TTFT}_r \le S_T \wedge \text{TPOT}_r \le S_D\}|}{T_{\text{measure}}}. \tag{3}$$

C Load Sweeps and Capacity Boundaries

This section supports Section 4.2.

C.1 Model-Scale and Workload Comparisons

The gains extend to both model scale and MoE inference: on Qwen1.5-110B, Phasor reduces E2E P99 by 3.3–25.2% and improves goodput by 6.7–26.1% over Mooncake, the strongest baseline; on Qwen3-30B-A3B, the corresponding gains are 6.5–15.8% and 11.6–19.2%.

The largest gains occur under Prefill-heavy and Burst traffic, where nonlinear queue imbalance and changing P/D pressure exercise Phasor's cross-worker and cross-pool controls. The sole metric-level exception is Llama3.1-8B under steady Decode-heavy traffic. Its short prompts, long outputs, and ample HBM headroom produce a stable decode-dominated regime with infrequent allocatability loss. Bullet's fine-grained P/D coexecution consequently achieves 4.1% lower TPOT and 4.8% lower E2E P99. Phasor nevertheless reduces TTFT by 24.4%, raises joint SLO attainment by 8.0 percentage points, and improves goodput by 10.0% through deadline-aware routing and allocatability-aware KV control. Thus, Bullet is favored when sustained decode execution determines the tail, whereas Phasor gains most when serving capacity is constrained by queued prefill work, KV allocatability, or changing P/D pressure.

Table 3: Calibration and controller sensitivity on Qwen3.5-27B, normalized to the default.

Calibration and sensitivity

Parameter	Tested values	Norm. P99	Norm. capacity
Profile fit	Lengths 128–8,192	$R^2 > 0.99$	
α scale	$2.0\times$ /$0.5\times$	0.99/1.14	0.94/0.92
θ_{S}	0.10/0.35	1.09/1.00	0.96/0.98
Control epoch	20/200 ms	0.99/1.06	1.01/0.97
θ_{flow}	0.50/0.80	1.02/1.04	0.99/0.95

Across dense and MoE models from 8B to 110B, Phasor leads every reported metric under Prefill-heavy and Burst traffic, reducing E2E P99 by 12.5–31.3% and improving SLO-goodput by 13.1–29.0%. The load sweep shows widening goodput gains near saturation, indicating increased effective serving capacity beyond a single operating point.

C.2 Behavior near the capacity boundary

Figure 5 shows that all systems convert nearly all offered load into SLO-compliant completions at $0.6 \times \lambda_{\mathrm{ref}}$, but the baselines saturate at or before λ_{ref}. At $1.1 \times \lambda_{\mathrm{ref}}$, Phasor exceeds the strongest baseline's SLO-goodput by 31.3% under Prefill-heavy traffic and 41.9% under Burst. The widening gap near saturation shows that Phasor expands effective serving capacity rather than providing a constant improvement at one operating point.

D Calibration and Control Sensitivity Summary

This section supports Section 4.4. Table 3 summarizes calibration and controller sensitivity; Appendix F.4 explains prediction construction. Runtime overhead and chunked-prefill measurements appear in Tables 4 and 5.

E KV Migration and Role-Conversion Details

This section supports Section 3.2.

E.1 Architecture and Expanded Control Flow

Figure 6 shows the observation–decision–actuation loop. Algorithm 2 expands the event-driven decisions in Algorithm 1; the following subsections specify KV migration and role conversion.

E.2 Asynchronous Pipelined KV Migration

Phasor uses request-level KV migration to restore worker-local allocatable capacity when decode-side pressure arises from an unfavorable distribution of resident KV states. This mechanism does not compact physical pages, whose non-contiguity is already handled by PagedAttention-style block allocation. Instead, Phasor relocates the complete live KV state of selected requests between decode workers. The source continues decoding during bulk transfer; final synchronization and ownership handoff require a brief pause. Phasor contributes the feasibility-driven selection and scope coordination described below.

Trigger and repair target. A new Meso repair is triggered by $\Delta_B(d,t) > 0$ (Section 3.2). Because $\Phi_S = (F_{\mathrm{alloc}} - \widehat{B}_{\mathrm{req}})/B_{\mathrm{cap}}$, this condition already implies $\Phi_S < 0$; a positive θ_S does not advance the trigger. Instead, θ_S specifies the source's projected post-repair headroom. Clearing the deficit only restores $\Phi_S \geq 0$, whereas a sufficient repair reaches $\Phi_S \geq \theta_S$. TryMeso uses this target when judging whether a local repair can defer Macro (Algorithm 1).

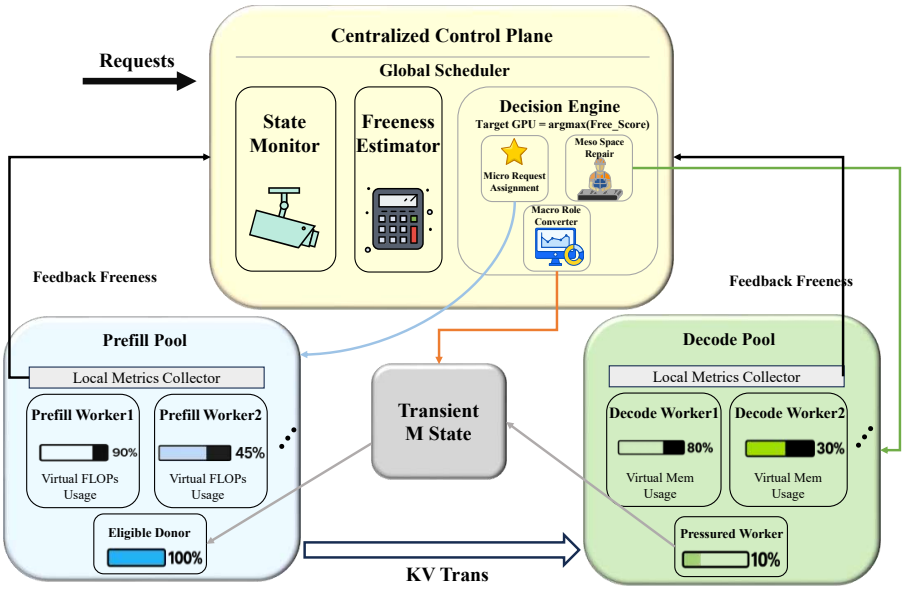


Figure 6: Phasor's closed loop. Virtual Freeness makes phase-specific pressure comparable, while scope-specific evidence selects Micro, Meso, or Macro under minimum-sufficient intervention.

Algorithm 2 Expanded Phasor Multi-Scale Control

Require: Pools $\mathcal{P}, \mathcal{D}$; transient $\mathcal{M} \leftarrow \emptyset$
Require: Repair target $0 < \theta_S < 1$; $\theta_{\text{flow}}, K, \Delta t, T_{\text{cool}}$
$k_P, k_D \leftarrow 0$; $t_{\text{conv}} \leftarrow -\infty$
while running **do**
  if request r arrives **then**
    DISPATCH$(r, \arg\max_{w \in \mathcal{P}} \Phi(w; r, t))$
  end if
  if control epoch Δt expires **then**
    $g_D \leftarrow$ TRYMESO$(\mathcal{D}, \theta_S)$
    $\Delta_\Phi \leftarrow$ MEANPHI$(\mathcal{D})$ − MEANPHI$(\mathcal{P})$
    if $\Delta_\Phi < -\theta_{\text{flow}}$ and g_D **then**
      $k_P, k_D \leftarrow 0$
    else
      $(k_P, k_D) \leftarrow$ PERSIST$(\Delta_\Phi, \theta_{\text{flow}})$
      $g_M \leftarrow [\max(k_P, k_D) \geq K] \wedge [t - t_{\text{conv}} \geq T_{\text{cool}}]$
      if g_M **then**
        $(\mathcal{X}, \mathcal{Y}) \leftarrow (\mathcal{D}, \mathcal{P})$ if $k_P \geq K$, else $(\mathcal{P}, \mathcal{D})$
        if $\mathcal{X}_{\text{eligible}} \neq \emptyset$ **then**
          $v \leftarrow \arg\max_{u \in \mathcal{X}_{\text{eligible}}} \Phi(u, t)$
          ASYNCCONVERT$(v, \mathcal{X} \to \mathcal{M} \to \mathcal{Y})$
          $k_P, k_D \leftarrow 0$; $t_{\text{conv}} \leftarrow t$
        end if
      end if
    end if
  end if
end while

Request selection. The migration unit is one active request together with its complete live KV state and minimal execution metadata. Let $\mathcal{R}_d(t)$ denote the eligible requests on source worker d, and let $B_i(t)$ be the current KV footprint of request r_i. Requests already undergoing migration or temporarily pinned by the execution engine are excluded. Selection seeks the least KV movement sufficient to reach the source headroom target, subject to destination feasibility and migration limits.

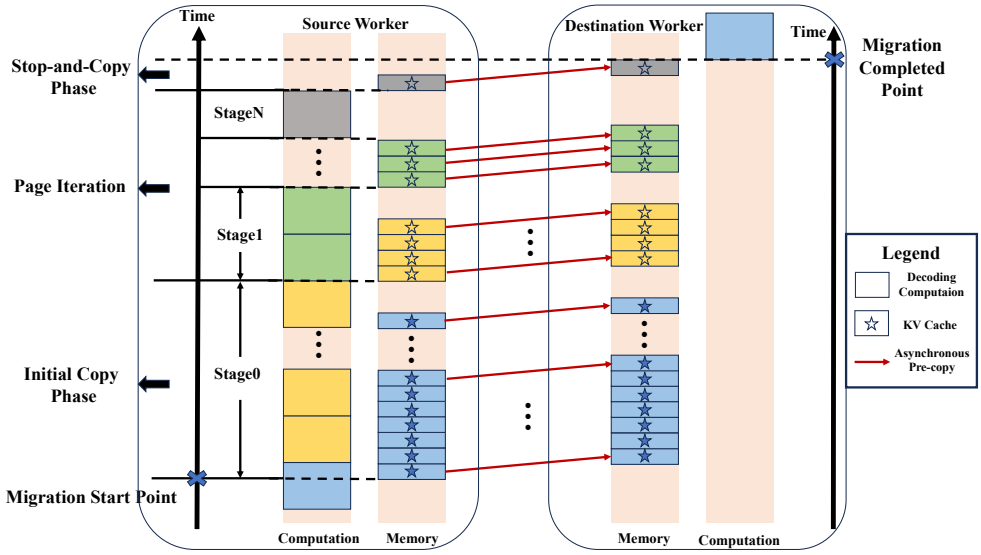


Figure 7: Asynchronous pipelined KV migration. Stage 0 copies the existing KV snapshot while the source continues decoding. Stages $1, \ldots, N$ transfer the incremental KV state generated since the preceding round. During Stop-and-Copy, Phasor briefly pauses the source, synchronizes the final delta, atomically transfers request ownership, and resumes decoding at the destination. Thus, only the final delta and handoff lie on the blocking path.

Projected source freeness accounts for both the KV blocks released and the future demand removed with the selected requests. Selection stops at the target; a budget-limited or infeasible repair is not classified as sufficient merely because it reduces the deficit.

Destination selection. A destination must remain feasible after accepting both the current KV state and its projected growth. For candidate request r_i and destination d', Phasor estimates

$$\Phi_S^+(d'; r_i, t) = 1 - \frac{\begin{array}{c} B_{\mathrm{cap}}(d') - F_{\mathrm{alloc}}(d', t) + \widehat{B}_{\mathrm{req}}(d', t) \\ + B_i(t) + \widehat{B}_{\mathrm{grow}}(r_i, t) \end{array}}{B_{\mathrm{cap}}(d')}. \tag{4}$$

where $\widehat{B}_{\mathrm{grow}}(r_i, t)$ is the request's projected additional KV demand. Phasor considers only destinations with positive post-migration freeness and selects

$$d_i^\star = \arg \max_{d' \in \mathcal{D} \setminus \{d\}} \Phi_S^+(d'; r_i, t). \tag{5}$$

This prevents Meso repair from merely moving the hotspot to another worker. If no destination retains positive Φ_S^+, no migration is launched. The unresolved pressure remains visible to subsequent control epochs and may eventually require Macro conversion.

Initial Copy Phase, Stage 0. Figure 7 illustrates the transfer process. At the migration start point, the source remains the unique execution owner of the request. Phasor records the current sequence version and asynchronously copies the existing KV snapshot to the destination over the node-local GPU interconnect. Decode continues on the source during this bulk copy, so migration does not suspend request progress in proportion to the full context size. The destination stores the replica but does not execute the request.

Delta Iteration, Stages $1, \ldots, N$. Decode appends KV entries for new tokens after the initial snapshot. Sequence versions identify the incremental state produced between copy rounds. Previously computed token entries remain unchanged; an allocator block may still receive new entries until it is full. The delta therefore includes newly valid entries, including those extending a partially filled block, rather than treating every existing physical block as immutable.

Let S_{Δ_j} be the remaining delta size after round j, BW_{eff} the measured effective migration bandwidth, and T_{sync} the synchronization cost. Phasor estimates the final blocking time as

$$\widehat{T}_{\text{final}}^{(j)} = \frac{S_{\Delta_j}}{BW_{\text{eff}}} + T_{\text{sync}}. \tag{6}$$

Delta iteration continues until $\widehat{T}_{\text{final}}^{(j)}$ falls below the configured handoff-pause budget T_{pause}, or until the maximum number of rounds $N_{\max}$ is reached.

Stop-and-Copy and atomic handoff. Once the remaining delta is sufficiently small, Phasor suspends the request at a token boundary. The source transfers the final KV blocks together with the current sequence length, generated-token state, and execution metadata. The destination verifies that the received version matches the final source version and then acknowledges readiness. Only after this acknowledgement does the centralized scheduler atomically change request ownership from the source to the destination. Decode subsequently resumes at the destination, and the source releases its KV blocks only after the ownership update succeeds.

This ordering enforces three invariants. First, exactly one worker is permitted to execute the request at any time. Second, the source remains authoritative until the destination holds a complete and validated state. Third, source-side KV blocks are never reclaimed before the destination acknowledges the final version. These invariants prevent duplicate token generation, missing KV blocks, and premature memory reclamation.

Non-convergence and failure recovery. Before ownership commits, Phasor aborts if the pause budget is not reached within $N_{\max}$ rounds, destination feasibility is lost, or the readiness acknowledgement times out. The source remains authoritative and resumes decoding if paused; the destination discards its uncommitted replica. A later Meso epoch may retry with another request or destination. This abort path assumes a live source and an available ownership coordinator; it does not provide crash recovery for either component. After ownership commits, the destination is authoritative, so the pre-commit rollback rule no longer applies. Copy traffic can still compete with serving even when the source remains online; the short handoff pause alone does not measure total migration interference.

Critical-path cost. The complete migration time still grows with the resident KV-state size because the initial snapshot must be copied. However, this bulk work overlaps source-side decode. The service-visible pause depends on the final delta and atomic ownership transfer rather than the complete context. For the reported Qwen3.5-27B configuration, the P99 handoff pause is 8.1 ± 1.2 ms (Table 4); it is not the full migration duration.

E.3 Directional Donor-Based Role Conversion

Micro routing and Meso repair redistribute work within the current P/D allocation, but neither can correct a persistent shortage of workers in one phase. Phasor therefore converts an existing worker from the less pressured pool into the opposite role. The Mixed state M represents only this transition and contains no reserved standby capacity.

The conversion trigger and Meso precedence follow Algorithm 1.

Donor eligibility and selection. The donor is selected from the less pressured pool. Workers already draining, migrating KV state, undergoing conversion, or protected by cooldown are ineligible. For tensor-parallel deployments, a logical worker is the complete tensor-parallel group, and all ranks are converted together. Among eligible workers, Phasor selects

$$v^\star = \begin{cases} \arg\max\limits_{d \in \mathcal{D}_{\text{eligible}}} \Phi_S(d,t), & \Delta_\Phi > \theta_{\text{flow}}, \\ \arg\max\limits_{p \in \mathcal{P}_{\text{eligible}}} \Phi_T(p,t), & \Delta_\Phi < -\theta_{\text{flow}}. \end{cases} \tag{7}$$

Drain before conversion. Once selected, the donor stops receiving new requests but continues serving its existing work. Prefill donors complete their queued requests, while decode donors finish

their active sequences. During draining, the worker remains part of its original pool for execution but is excluded from new routing and subsequent donor selection. No request state is discarded or transferred solely to accelerate Macro conversion.

Transient Mixed state. After the donor becomes empty, it leaves the original pool and enters M. Model weights remain resident because both phases execute the same checkpoint. Phasor releases phase-specific temporary state, switches the worker's scheduler and memory-management role, initializes the destination-phase execution context, and registers the worker with the pressured pool. The worker does not accept requests while it is in M.

The complete transition is therefore

$$\begin{aligned} \mathcal{P} &\to \text{Draining} \to \mathcal{M} \to \mathcal{D}, \\ \mathcal{D} &\to \text{Draining} \to \mathcal{M} \to \mathcal{P}. \end{aligned} \tag{8}$$

The number of physical GPUs remains unchanged throughout this process. Active serving capacity can temporarily decrease while the donor drains or occupies M, but Phasor does not reserve or introduce additional hardware.

Activation and cooldown. A converted worker becomes visible to routing only after its destination-phase scheduler, allocator, and execution buffers are ready. Pool membership is then updated atomically, preventing the controller from assigning work before activation completes. Phasor records the conversion initiation time and enforces the conversion cooldown in Algorithm 1. In our experiments, the control epoch is 50 ms and $T_{\text{cool}} = 1$ s.

The end-to-end response time of Macro control can be decomposed as

$$T_{\text{response}} = K\Delta t + T_{\text{drain}} + T_{\text{switch}} + T_{\text{activate}}, \tag{9}$$

where $K\Delta t$ is the persistence delay, T_{drain} depends on the donor's remaining work, and $T_{\text{switch}} + T_{\text{activate}}$ is the system actuation cost. Reporting these components separately distinguishes deliberate control stability from implementation overhead.

Coordination with Micro and Meso. The converting worker is removed from Micro routing as soon as draining begins and is excluded from Meso source and destination selection until activation completes. Hence, request routing, KV repair, and role conversion cannot issue conflicting actions on the same worker. After activation, the worker exposes its new phase-specific state and participates in the corresponding branch of Virtual Freeness.

F Practicality and Robustness

This section supports Section 4.4 with runtime measurements, two compatibility studies, and calibration details. The summary sensitivity results are in Table 3.

F.1 Runtime Overhead

We measure the overhead of each control scale separately on the same 8×H100 platform used in the main evaluation. Micro overhead is the wall-clock time required to evaluate all candidate P workers and select a destination for one arriving request. Meso overhead is reported as the P99 request pause during the final synchronization and ownership handoff; the preceding KV snapshot and delta transfers proceed asynchronously. System-level overhead measures the fraction of serving time consumed by control and actuation. Table 4 reports mean±SD over three runs.

F.2 Compatibility with Chunked Prefill

This experiment tests whether chunked prefill remains compatible with Full Phasor. Enabling 1,024-token chunks improves the reported serving metrics despite a 5.0% increase in isolated prefill P90 (Table 5). Because both configurations retain Phasor, this comparison establishes compatibility, not Phasor's incremental gain over a chunking-only controller.

Table 4: Runtime overhead of Phasor. Values are mean±SD over three runs. Micro latency measures the routing decision path; Meso pause measures the final synchronization and handoff rather than the asynchronous KV copy.

Scope	Variant or configuration	Metric	Mean±SD
Micro routing	Token+Occ	Decision latency	8.1 ± 0.4 μs
	DirectHW+Occ	Decision latency	6.4 ± 0.3 μs
	Full Φ	Decision latency	11.7 ± 0.5 μs
Meso repair	Qwen3.5-27B, Decode-heavy	P99 handoff pause	8.1 ± 1.2 ms
	Qwen3-30B-A3B, MoE	P99 handoff pause	8.9 ± 1.3 ms
System level	Semi-PD	Control/actuation overhead	$2.1 \pm 0.2\%$
	Bullet	Control/actuation overhead	$3.4 \pm 0.3\%$
	Phasor	Control/actuation overhead	$2.8 \pm 0.2\%$

Table 5: Compatibility with chunked prefill on Qwen3.5-27B. Both configurations use Full Phasor and differ only in whether dynamic prefill chunking is enabled. Values are mean±SD over three runs.

Measurement	OFF	ON	Relative change
Isolated P90 (s)	0.161 ± 0.004	0.169 ± 0.005	$+5.0\%$
Burst TTFT P99 (s)	1.82 ± 0.09	$\mathbf{1.68 \pm 0.07}$	$\mathbf{-7.7\%}$
E2E P99 (s)	2.11 ± 0.10	$\mathbf{2.00 \pm 0.08}$	$\mathbf{-5.2\%}$
Peak SLO-GP (req/s)	24.1 ± 0.6	$\mathbf{25.0 \pm 0.5}$	$\mathbf{+3.7\%}$
Decode-ready trough (req/s)	16.8 ± 0.7	$\mathbf{17.7 \pm 0.6}$	$\mathbf{+5.4\%}$

We use Qwen3.5-27B in FP16 with TP$=1$ and an initial 4P:4D allocation. The workload is a ShareGPT-derived prefill-heavy trace with average input/output lengths of approximately 2048/128 tokens. Requests follow an MGP arrival process with CV≈ 3.0 and a $10 \rightarrow 30 \rightarrow 10$ req/s burst, where the peak occupies $t = 90$–120 s of a 180 s replay. Both configurations use Full Phasor and the same fixed trace; the only difference is whether dynamic chunking with `max_num_batched_tokens=1024` is enabled.

F.3 MoE Space Management

MoE inference changes the balance between active computation and persistent memory consumption. Although only a subset of experts is activated for each token, resident expert weights reduce the HBM headroom available to the KV cache. This setting provides a stringent test of whether Meso repair recovers usable decode capacity rather than merely responding to the memory behavior of dense models.

We evaluate Qwen3-30B-A3B in BF16 with TP$=1$ on 8×H100 SXM 80GB GPUs. The model contains 128 resident experts, of which eight are activated per token. We retain the initial 4P:4D allocation and use a ShareGPT-derived decode-heavy workload with average input/output lengths of approximately 128/2048 tokens. Each configuration is evaluated through a closed-loop concurrency sweep lasting 600 s, including a 60 s warm-up, and repeated three times. Micro, Macro, Φ_T, and Φ_S remain unchanged; the only experimental variable is whether Meso KV maintenance is enabled.

Both configurations have the same 7.40 GB physical KV budget per D-GPU. The corresponding maintenance overhead is $1.6 \pm 0.2\%$.

Here, effective fragmentation denotes KV capacity that exists physically but cannot be used at the worker where upcoming admissions require it. Meso does not increase the nominal KV budget or compact contiguous physical pages. It relocates selected KV states so that stranded pool capacity becomes allocatable at pressured workers. The MoE results therefore isolate the intended mechanism: Phasor restores effective admission capacity under severe resident-weight pressure.

Table 6: Meso space management on Qwen3-30B-A3B. Both configurations retain Micro, Macro, and the same Φ formulation; the only difference is whether Meso KV maintenance is enabled. Values are mean$\pm$SD over three runs.

Measurement	w/o Meso	Full Phasor	Relative change
KV budget per D-GPU	7.40 GB	7.40 GB	0.0%
Avg. frag. (%)	21.0 ± 1.1	$\mathbf{7.0 \pm 0.5}$	$\mathbf{-66.7\%}$
Peak frag. (%)	31.0 ± 1.7	$\mathbf{10.0 \pm 0.8}$	$\mathbf{-67.7\%}$
Effective KV tokens per D-GPU	$59.5\text{K} \pm 1.4\text{K}$	$\mathbf{70.0\text{K} \pm 1.2\text{K}}$	$\mathbf{+17.6\%}$
Failures/1K requests	18.0 ± 2.0	$\mathbf{2.0 \pm 1.0}$	$\mathbf{-88.9\%}$
P99 TPOT (ms)	24.6 ± 0.8	$\mathbf{21.9 \pm 0.6}$	$\mathbf{-11.0\%}$
SLO concurrency per D-GPU	51.6 ± 1.3	$\mathbf{60.8 \pm 1.1}$	$\mathbf{+17.8\%}$

F.4 Calibration and Parameter Sensitivity

The prefill coefficients α and β model the execution cost of a fixed model stack. They are obtained offline by profiling prefill execution across input lengths and batch configurations and jointly fitting the observed execution time. The resulting regression achieves $R^2 > 0.99$. The coefficients are kept fixed across evaluated workloads on the same model stack rather than tuned per trace. A changed model, hardware, precision, kernel, or chunking configuration requires recalibration. The reported fit characterizes the profiled configurations; it does not guarantee prediction accuracy for unprofiled lengths or execution regimes.

Prefill completion-time construction. The request-level cost model in Section 3.1 is combined with the engine's batch configuration and queue order to estimate completion time. After inserting the candidate request, Phasor accounts for the unfinished portion of the current execution batch and accumulates the durations of subsequent scheduled batches until the target request completes. A shared batch duration is counted once rather than once per participating request. Completed work is excluded. With chunked prefill, the same accounting uses the remaining chunks and their processed-context state; it does not scale full-prompt latency linearly by the remaining-token fraction. The estimate uses current queued work and the candidate request, without assuming future arrivals.

KV-demand construction. For each active request, Phasor projects token growth over H from observed decode progress, limited by its conservative remaining-output estimate. It rounds the resulting token footprint to the allocator's KV-block granularity and counts only blocks beyond those already allocated, then sums these increments over the worker. For a newly assigned candidate, the estimate also includes its KV state not yet resident on the destination. Remaining-output estimates account for the request's ongoing progress; exceeding an output-length quantile alone is not treated as completion of a still-active request. The prediction uses information available at the decision time, not the request's eventual recorded output length. The same block accounting applies to Meso: destination feasibility includes incoming resident state and projected growth, while source feasibility is re-evaluated after removing the selected requests and their projected demand.

The calibration and controller sensitivity results are reported in Table 3.

G Detailed Attribution Analysis

This section supports Sections 4.3 and 4.4.

Controlled signal variants and matched settings All variants use the same Micro, Meso, and Macro state machines, control periods, actuators, donor selection, migration limits, persistence requirement, cooldown, and tuning budget. They differ only in the state used to drive these decisions.

Token+Occ uses linear remaining-token headroom for P workers and aggregate memory occupancy for D workers. DirectHW+Occ replaces the P signal with 50-ms EWMA SM activity while retaining occupancy for D. These are independent conventional signal designs rather than components removed from Virtual Freeness.

Decision-complete local state The upper block of Table 2 first establishes the cost of incomplete observations. Relative to Token+Occ, Full Φ reduces E2E P99 by 14.9–27.1% and improves SLO-goodput by 15.6–26.5%. Relative to DirectHW+Occ, the corresponding improvements are 19.2–32.7% and 21.4–34.7%. Linear token volume misses nonlinear prefill computation; SM activity reports executing kernels but not queued work or remaining TTFT slack; memory occupancy reports allocated bytes but not whether the remaining KV blocks can admit projected requests. Because every actuator remains available to all variants, these improvements arise from better decision state rather than additional control opportunities.

The mechanism changes with the dominant constraint. Under Prefill-heavy, Full Φ reduces predicted-work CV from 0.162 ± 0.011 with Token+Occ and 0.196 ± 0.014 with DirectHW+Occ to 0.041 ± 0.004. Under Decode-heavy, it reduces average/peak effective fragmentation from $12.6 \pm 0.7\%$/$19.1 \pm 1.2\%$ with Token+Occ to $7.1 \pm 0.5\%$/$10.4 \pm 0.7\%$. Under Burst, pressure-shift detection falls from 0.58 ± 0.05 s and 0.72 ± 0.06 s to 0.26 ± 0.03 s. These measurements connect the end-to-end gains to queue-risk estimation, allocatable KV capacity, and cross-phase pressure detection.

The Full Φ–RichPhase gap is smallest under Decode-heavy, consistent with its short-input, long-output requests maintaining a stable decode-dominated bottleneck. It becomes largest under Burst, where P and D pressure rise at different times and in different resource domains. Full Φ compares P/D pressure as normalized residual margins to phase-specific feasibility boundaries, giving both phases the same control interpretation. In the evaluated pressure shifts, this common scale supports more effective cross-phase control than RichPhase's independently calibrated threshold exceedance. It reduces pressure-shift decision latency from 0.44 ± 0.04 s to 0.26 ± 0.03 s, shortens post-shock recovery from 6.2 ± 0.5 s to 5.0 ± 0.4 s, and reduces role conversions from 6.0 ± 0.8 to 4.0 ± 0.0 over the 180-s trace.

Contribution of each control scale The lower block of Table 2 adds the three control scales cumulatively. Base uses linear-token routing, no Meso repair, and a fixed 4P:4D allocation; Micro introduces deadline-aware routing, Meso adds allocatability-aware KV repair, and Macro adds directional donor conversion. All other settings remain fixed. These increments are conditional on the cumulative order, not standalone or additive effects of individual modules. The gains align with their targeted scopes: Micro reduces Prefill-heavy E2E P99 by 14.2% and raises goodput by 15.7%, lowering predicted-work CV from 0.252 ± 0.018 to 0.071 ± 0.007; Meso further reduces Decode-heavy E2E P99 by 24.1% and raises goodput by 20.5%, lowering average/peak effective fragmentation from $16.9 \pm 1.0\%$/$25.1 \pm 1.5\%$ to $7.9 \pm 0.6\%$/$11.8 \pm 0.8\%$; Macro further reduces Burst E2E P99 by 16.8% and raises goodput by 19.8%. Together, the three scales reduce E2E P99 by 37.1–42.4% and improve SLO-goodput by 47.9–54.6% over Base.

Meso restores allocatable KV capacity Effective fragmentation is the fraction of nominally free KV capacity that cannot satisfy projected admissions under the current worker-level placement. Figure 3(a) traces this state during a 1000-s production-derived MGP burst on the same H100 platform. Paged allocation handles physical non-contiguity within a worker but cannot repair unfavorable KV placement across workers. Semi-PD and Bullet therefore reach average/peak fragmentation of 17.0%/30.2% and 14.1%/25.1%, respectively. By migrating only the KV states required to restore admission headroom, Phasor limits these values to 7.8%/12.0%, reducing peak fragmentation by 52.2% over Bullet and 60.3% over Semi-PD. The delayed post-burst peaks are consistent with accumulated placement and repair dynamics rather than an instantaneous response to arrival rate.

H Extended Related Work and Scope

This section expands Section 5. Phasor couples two decisions that must agree: whether a worker can accept additional work and which intervention can restore feasibility. Its contribution is a common residual-feasibility signal connected to a recovery policy that distinguishes worker-local placement loss from pool-wide capacity shortage.

Phase-disaggregated execution and provisioning DistServe and Splitwise specialize P/D resources through placement, provisioning, and phase-aware scheduling (Zhong et al., 2024; Patel et al., 2024). P/D-Serve adapts allocation and forwarding (Jin et al., 2024), WindServe schedules

streams to reduce pipeline bubbles (Feng et al., 2025), and TaiChi selects execution modes according to the SLO regime (Wang et al., 2025a). Semi-PD and Bullet improve concurrent P/D execution through resource partitioning and sharing (Hong et al., 2025; Lin et al., 2026). TokenScale matches prefill, network, and decode rates through Token Velocity and absorbs bursts with SLO-aware chunked prefill on Convertible Decoders (Lai et al., 2025). Phasor addresses the residual control problem within a fixed GPU budget: rate balance or an execution-mode choice does not by itself determine whether a particular queue can meet its deadlines or whether local KV relocation can avoid a pool conversion. Phasor makes those decisions explicit and couples them through minimum-sufficient intervention; Macro transfers a drained worker's role only when the prescribed local repair cannot resolve persistent cross-pool pressure.

Feasibility signals for coordinated scheduling Llumnix's published policy unifies dispatching, migration, and autoscaling through virtual usage and memory-based freeness, accounting for waiting demand, priority headroom, and draining (Sun et al., 2024). Phasor advances the decision semantics from virtual load to phase-specific admission feasibility: Φ_T evaluates nonlinear queued computation against each affected request's remaining TTFT slack, while Φ_S evaluates projected KV demand against worker-local allocatable blocks. Equal memory freeness can therefore conceal different prefill deadline risks, just as matched aggregate processing rates can conceal different local KV deficits. Expressing these constraints as normalized residual margins gives routing and cross-pool control a common feasibility interpretation. Crucially, the controller also checks whether a sufficient within-pool repair exists before expanding the decode pool. The distinction from virtual-memory scheduling and rate matching lies in this coupling of constraint-specific state with recovery-scope selection.

KV-cache and state orchestration Mooncake combines KV-centric disaggregation with SLO-aware scheduling and overload control, while MemServe organizes state through elastic memory pools (Qin et al., 2025; Hu et al., 2024). FlowKV couples low-latency transfers with load-aware scheduling and flexible P/D allocation (Li et al., 2025); BanaServe coordinates shared KV storage, migration, and routing (He et al., 2026). These mechanisms enable state movement. Phasor determines when movement is sufficient: Meso checks both the source's post-repair headroom and the destination's projected feasibility, and its outcome gates Macro escalation. This connection prevents local placement loss from automatically triggering a pool reallocation and avoids conflicting interventions on the same worker. Appendix E.2 gives the asynchronous execution protocol.

Contribution and supporting evidence Phasor's contribution is the combination of decision-relevant feasibility and coordinated recovery, rather than the availability of routing, migration, or conversion in isolation. Table 2 tests the signal against conventional observations and RichPhase, which retains the same phase information with a different calibration and comparison rule. Figure 3(b) holds signals and actuators fixed and shows the benefit of selecting the sufficient recovery scope: higher goodput with less KV movement, draining, and conversion than Global-first and Independent. Table 1 establishes end-to-end gains against five baselines under the matched implementation protocol in Appendix B.2. The TokenScale and Llumnix comparisons above concern their published designs; they are not direct performance measurements, and the controlled variants do not stand in for those systems.

Scope The evaluated scope is single-node dense and MoE serving with a fixed GPU budget. Positive Φ denotes modeled feasibility, not a joint TTFT/TPOT guarantee; minimum-sufficient intervention orders available recovery scopes, not globally optimal schedules. Meso requires destination headroom, and Macro requires an eligible donor and time to drain it. Network-bound operation and multi-node scaling require additional modeling and evaluation. The steady Decode-heavy exception in Section 4.2 further delineates where fine-grained coexecution is preferable. Within the tested regimes, Phasor's advantage is recovering usable capacity with an intervention matched to the actual constraint.

I Extended Controlled-Experiment Statistics

This section supports Sections 4.3 and 4.4 and extends Table 2 with TTFT, TPOT, attainment, and run-to-run variability. Repeated E2E and goodput entries are the same measurements, not indepen-

Table 7: Extended metrics for the conventional-signal comparisons on Qwen3.5-27B (mean±SD over three runs). All variants retain identical actuators and differ only in decision state; RichPhase results are in Table 2.

Signal	TTFT P99 (s)	TPOT P99 (ms)	E2E P99 (s)	Attainment (%)	Goodput (req/s)
Prefill-heavy					
Token+Occ	3.10 ± 0.11	58 ± 2	9.60 ± 0.32	73.3 ± 1.6	13.2 ± 0.3
DirectHW+Occ	3.55 ± 0.14	61 ± 2	10.40 ± 0.42	68.9 ± 1.7	12.4 ± 0.3
Full Φ	$\mathbf{1.78 \pm 0.06}$	$\mathbf{43 \pm 1}$	$\mathbf{7.00 \pm 0.22}$	$\mathbf{92.8 \pm 1.1}$	$\mathbf{16.7 \pm 0.2}$
Decode-heavy					
Token+Occ	0.158 ± 0.005	54 ± 2	111.00 ± 3.70	80.9 ± 2.4	2.75 ± 0.08
DirectHW+Occ	0.171 ± 0.006	57 ± 2	117.00 ± 4.10	77.1 ± 2.4	2.62 ± 0.08
Full Φ	$\mathbf{0.112 \pm 0.003}$	$\mathbf{46 \pm 1}$	$\mathbf{94.50 \pm 2.60}$	$\mathbf{93.5 \pm 1.8}$	$\mathbf{3.18 \pm 0.06}$
Burst					
Token+Occ	1.15 ± 0.06	72 ± 3	19.00 ± 0.84	73.1 ± 2.8	7.9 ± 0.3
DirectHW+Occ	1.32 ± 0.07	76 ± 3	20.50 ± 0.96	68.5 ± 2.8	7.4 ± 0.3
Full Φ	$\mathbf{0.72 \pm 0.04}$	$\mathbf{58 \pm 2}$	$\mathbf{15.40 \pm 0.55}$	$\mathbf{89.8 \pm 1.9}$	$\mathbf{9.7 \pm 0.2}$

Table 8: Complete cumulative-control ablation on Qwen3.5-27B. Each row adds one control scale while preserving all previously enabled components.

Configuration	TTFT P99 (s)	TPOT P99 (ms)	E2E P99 (s)	Attainment (%)	Goodput (req/s)
Prefill-heavy					
Base	4.20 ± 0.17	65 ± 2	12.00 ± 0.46	60.0 ± 1.7	10.8 ± 0.3
+Micro	3.20 ± 0.11	61 ± 2	10.30 ± 0.36	69.4 ± 1.7	12.5 ± 0.3
+Meso	3.15 ± 0.11	56 ± 2	9.70 ± 0.34	71.1 ± 1.7	12.8 ± 0.3
+Macro (Full)	$\mathbf{1.78 \pm 0.06}$	$\mathbf{43 \pm 1}$	$\mathbf{7.00 \pm 0.22}$	$\mathbf{92.8 \pm 1.1}$	$\mathbf{16.7 \pm 0.2}$
Decode-heavy					
Base	0.160 ± 0.005	80 ± 3	164.00 ± 5.80	63.2 ± 2.4	2.15 ± 0.08
+Micro	0.120 ± 0.004	79 ± 3	162.00 ± 5.60	64.7 ± 2.1	2.20 ± 0.07
+Meso	0.116 ± 0.004	60 ± 2	123.00 ± 3.90	77.9 ± 2.1	2.65 ± 0.07
+Macro (Full)	$\mathbf{0.112 \pm 0.003}$	$\mathbf{46 \pm 1}$	$\mathbf{94.50 \pm 2.60}$	$\mathbf{93.5 \pm 1.8}$	$\mathbf{3.18 \pm 0.06}$
Burst					
Base	1.55 ± 0.08	92 ± 4	24.50 ± 1.15	58.3 ± 3.7	6.3 ± 0.4
+Micro	1.08 ± 0.06	88 ± 4	22.50 ± 1.02	63.9 ± 2.8	6.9 ± 0.3
+Meso	0.96 ± 0.05	70 ± 3	18.50 ± 0.78	75.0 ± 2.8	8.1 ± 0.3
+Macro (Full)	$\mathbf{0.72 \pm 0.04}$	$\mathbf{58 \pm 2}$	$\mathbf{15.40 \pm 0.55}$	$\mathbf{89.8 \pm 1.9}$	$\mathbf{9.7 \pm 0.2}$

dent evidence; values are mean±sample SD over three runs. RichPhase's reported E2E and goodput remain in Table 2; unreported per-metric statistics are not inferred here.