Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Orchestrating Large Language Models in Time-Constrained Marketplace Systems: A Cost and Latency Optimization Framework

Sujoy Datta Choudhury
Independent Researcher, USA

ARTICLE INFO

Received: 03 Nov 2025

Revised: 14 Dec 2025

Accepted: 23 Dec 2025

ABSTRACT

Digital marketplaces facilitating advertising transactions and financial product exchanges operate within microsecond-level temporal boundaries where processing determinations require rapid completion. Advanced neural language architectures provide sophisticated analytical capabilities for contextual interpretation, behavioral intent extraction, and risk pattern detection, yet impose considerable computational burdens and exhibit variable processing intervals. This article addresses integration strategies for neural language systems within temporally restricted operational cycles while preserving economic sustainability and system responsiveness. The coordination challenge spans multiple technical domains: selecting appropriate architectural scales, managing pre-computed contextual states, distributing processing time across pipeline components, and optimizing accelerator utilization between immediate-response and background operations. Drawing from recent progress in inference acceleration and production marketplace deployments, this article presents a governance-oriented coordination framework that positions neural language capabilities as regulated computational resources within managed decision architectures. The article combines system engineering principles for optimization and state management, financial models analyzing deployment economics, and domain expertise in auction theory and risk quantification.

Keywords: Large Language Models, Real-time Marketplaces, Latency Optimization, Cost Orchestration, Inference Serving

1. Background and Motivation

1.1 Temporal Constraints in Digital Marketplaces

Modern transaction platforms supporting programmatic advertising and algorithmic trading function under exceptionally rigid time boundaries where complete processing cycles must finish within narrow temporal windows, even as the variety of input signals and strategic approaches undergoes continuous growth [1]. These operational domains constitute some of the most computationally demanding scenarios in contemporary digital commerce, where incremental increases in processing duration translate directly into measurable reductions in transaction value or degraded participant satisfaction. Traditional system designs utilize lightweight prediction algorithms and manually constructed logical rules engineered for execution within minimal time budgets, though such implementations inherently forfeit the nuanced analytical depth that sophisticated computational approaches could provide.

1.2 Capabilities and Limitations of Neural Language Systems

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Modern transformer-based language architectures demonstrate substantially enhanced performance in contextual interpretation, semantic intent extraction, and risk signature identification when compared against conventional algorithmic decision frameworks [2]. These systems possess inherent capacity for natural language comprehension, semantic relationship mapping, and contextually adapted response generation, creating pathways for enriched marketplace intelligence infrastructure. However, these analytical advantages arrive packaged with significant computational resource consumption and unpredictable processing duration distributions that appear fundamentally misaligned with millisecond-scale operational requirements of immediate-response decision frameworks. The primary technical challenge centers on bridging this capability-constraint divide.

1.3 Scope and Research Contributions

This investigation addresses the central technical problem of incorporating neural language architectures within time-bounded decision processes while maintaining both cost efficiency and response performance standards. The analysis traverses three interconnected technical domains: architectural design of inference acceleration systems, financial frameworks governing infrastructure deployment decisions, and operational specifications characterizing marketplace platforms [3][4]. Rather than isolating individual optimization techniques, the objective centers on establishing a comprehensive governance framework for managing neural language invocations as controlled computational resources within a broader decision-making infrastructure.

Resource Allocation	Establish final selection and pricing	Optimization algorithms, auction logic	Strategic bid formulation, dynamic pricing

Table 1: Marketplace Decision Pipeline Stages and Processing Characteristics [3]

2. Technical Foundations and Literature Review

2.1 Architecture of Neural Inference Systems

Recent comprehensive technical surveys document the architectural design space for neural language model inference implementations, examining request batching methodologies, scheduling algorithm variants, memory hierarchy management, and distributed computation patterns in substantial detail [1][2]. These analytical works describe serving infrastructure implementations exhibiting extensive configurability across three primary optimization axes: aggregate request throughput, individual request latency, and computational hardware efficiency. The technical literature reveals evolutionary progression from static single-architecture deployments toward adaptive policy-controlled infrastructures that execute resource allocation determinations dynamically based on real-time system state observations and service quality targets. This technical foundation proves essential for marketplace implementations where inference infrastructure operates as a single element within more extensive time-critical processing sequences.

Optimization Axis	Measurement Metrics	Configuration Approaches	Marketplace Implications
Aggregate Throughput	Requests per second, token generation rate	Batch size tuning, continuous batching	Enables high-volume traffic handling
Individual Latency	Time-to-first-token, end-to-end response time	Request prioritization, speculative execution	Critical for auction deadline compliance

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Hardware Efficiency	GPU utilization, memory bandwidth saturation	Kernel optimization, quantization	Reduces operational cost per decision
Service Quality	Percentile latency guarantees, availability	Multi-tier scheduling, load shedding	Maintains participant experience standards

Table 2: Inference System Optimization Dimensions and Tradeoffs [1][2]

2.2 Processing Pipeline Architecture for Marketplaces

Immediate-response marketplace decision frameworks typically comprise four sequential computational stages: opportunity candidate identification, signal enrichment processing, economic valuation computation, and resource allocation determination [3]. Established implementations predominantly employ computationally efficient prediction models and manually engineered decision logic, selected primarily for their predictable timing characteristics and minimal computational footprint. The candidate identification phase extracts viable opportunities from extensive possibility spaces, enrichment processing augments these candidates with supplementary analytical signals and contextual metadata, valuation computation assigns estimated economic worth to each alternative, and allocation logic establishes final selection and pricing terms. Individual stage processing duration directly contributes to aggregate system latency, establishing tight coupling between component-level performance and overall system outcomes.

2.3 Neural Language Applications in Advertisement Systems

Investigations examining neural language model deployment in digital advertising contexts demonstrate potential modifications to auction mechanisms, bidding strategy formulation, and creative content generation when sophisticated computational intelligence supplants fixed algorithmic logic [4]. Advertisement platforms present distinctive coordination challenges because operational decisions must simultaneously satisfy multiple competing objectives: semantic alignment with user intent, advertiser capital efficiency, platform revenue targets, and participant experience quality. Neural language architectures suggest potential improvements across these evaluation dimensions through enhanced query semantic comprehension, participant context interpretation, and creative effectiveness prediction. Realizing these operational benefits requires thoughtful integration that accommodates the stringent performance requirements intrinsic to auction environments.

2.4 Knowledge Gaps in Current Literature

Notwithstanding progress in both inference acceleration systems and marketplace platform implementations, considerable knowledge gaps persist regarding systematic coordination of neural language model invocation within immediate-response decision contexts [1][2]. Existing inference serving systems target optimization for generalized workload distributions without specific accommodation for the multi-phase deadline-governed characteristics inherent to marketplace processing architectures. Conversely, marketplace platform research frequently conceptualizes model inference as an abstract operation with presumed performance properties rather than a configurable computational system exhibiting intricate performance tradeoff surfaces. Addressing these knowledge gaps demands frameworks that explicitly formalize reasoning about invocation timing decisions, model selection criteria, and computation structuring to satisfy available temporal and economic constraints.

3. Policy-Driven Coordination Architecture

3.1 Coordination Policy Formulation

An effective analytical framework conceptualizes the coordination mechanism as a decision policy that maps incoming request attributes to inference execution specifications [1][2]. These specifications may designate selections among diverse model architectures, including determining whether to

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

employ compact parameter-efficient variants versus larger capability-maximized alternatives, configuring variable context window dimensions, or entirely bypassing neural language invocation when previously computed results or simpler prediction models furnish adequate analytical output. This policy-oriented methodology represents a fundamental conceptual shift from perceiving model inference as a predetermined computational sequence to conceptualizing it as a resource allocation determination requiring adaptive formulation based on observed operational conditions and forecasted performance outcomes.

Request Attribute	Assessment Criteria	Inference Plan Options	Coordination Logic
Economic Value	Revenue potential, conversion probability	Large model with full context, small model, cache only	Higher value justifies increased computation
Contextual Complexity	Semantic ambiguity, entity relationships	Extended context window, multi-step reasoning	Complex queries require deeper analysis
Temporal Constraint	Remaining latency budget, queue depth	Cached response, distilled model, async processing	Tight budgets force economical choices
Historical Pattern	Cache hit likelihood, user behavior consistency	Reuse cached context, incremental generation	Repetitive patterns enable maximum reuse
Risk Indicators	Fraud signals, policy violation patterns	Deep model with logging, multi-phase verification	Suspicious activity triggers thorough analysis

Table 3: Policy Decision Factors and Inference Plan Specifications [1][2]

3.2 Adaptive Scheduling Integration

Technical research examining scheduling infrastructure and serving architecture emphasizes that high-performance systems progressively adopt policy-governed rather than static operational modes, with scheduling determinations incorporating immediate awareness of request queue depths, computational resource occupancy levels, and service quality performance targets [3][4]. This adaptive methodology permits coordination system responsiveness to evolving operational conditions, including request volume fluctuations, hardware component performance degradation, or distributional shifts in request complexity characteristics. By establishing comprehensive visibility across the complete serving infrastructure state, the coordination mechanism can formulate informed determinations regarding expensive computation invocation versus economical alternative selection based not exclusively on request attributes but additionally on instantaneous resource availability and contention intensity.

3.3 Architecture Selection Logic

The coordination mechanism must traverse a multidimensional decision space regarding architecture selection that equilibrates analytical capability against cost implications and timing constraints [1]. For high-economic-value requests where additional analytical sophistication directly produces improved operational outcomes, invoking larger architectures with extensive contextual processing may economically justify elevated computational costs. For bulk request traffic where incremental decision quality enhancements minimally influence aggregate performance indicators, compact architectures or previously computed outputs may suffice. The fundamental technical insight recognizes that individual requests possess distinct economic value functions, and optimal

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

coordination treats architecture selection as a preference-matching problem rather than enforcing uniform operational approaches.

3.4 Hierarchical Context Management

This analytical reasoning suggests a stratified architectural design where the coordination mechanism administers pre-computed contextual states as primary computational resources [2]. Rather than employing binary logic regarding model invocation decisions, the policy evaluates necessary new computation volumes for incremental decision quality gains. For example, the system may leverage previously computed enriched contextual representations associated with specific users or content pages, synthesizing only compact incremental continuations tailored to immediate auction parameters, thereby conforming to available latency allocations. This compositional methodology for context assembly permits granular governance over the computation-quality exchange relationship.

4. Temporal Optimization Methodologies

4.1 Latency Allocation Strategies

Time allocation boundaries in marketplace operational contexts impose rigid constraint hierarchies that coordination systems must accommodate [5]. When auction processing sequences require completion within fifty-millisecond end-to-end intervals, the coordination mechanism may govern only a restricted fraction of that allocation after accounting for network transmission latencies and bidding computation durations. This decomposition necessitates explicit examination of temporal consumption patterns and identification of components amenable to optimization or removal. Individual pipeline components must justify their latency contributions through quantifiable outcome enhancements, establishing pressure for efficient architectural design and systematic prioritization.

Processing Component	Typical Duration Range	Optimization Techniques	Coordination Implications
Network Transmission	Variable, protocol-dependent	Connection pooling, edge deployment	Fixed overhead, non-compressible
Candidate Retrieval	Sub-millisecond to low milliseconds	Index optimization, pre-filtering	Minimal impact on neural budget
Context Assembly	Low milliseconds	Cache utilization, lazy loading	Critical for neural invocation feasibility
Neural Inference	Highly variable	All caching and optimization methods	Primary target for coordination policy
Bidding Logic	Low milliseconds	Algorithm optimization, approximation	Constrains available neural budget
Response Formation	Sub-millisecond	Template pre-compilation	Negligible impact

Table 4: Latency Budget Decomposition in Auction Processing [5]

4.2 Pre-Computed Attention State Management

Methodologies encompassing attention state pre-computation, intermediate value caching, and computational reuse mechanisms transition from discretionary performance enhancements to mandatory architectural components in resource-constrained operational contexts [5][6]. Technical research on attention state pre-computation reveals that substantial numbers of inference requests share extensive prompt prefixes encompassing system directives, structural templates, or standardized contextual metadata, and that pre-calculating attention states for these repeated segments can diminish time-to-first-token by considerable multiples without parameter modification.

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

This technical observation maintains particular significance for marketplace implementations where identical structural context manifests repeatedly across requests, varying only in specific elements including user identifiers or immediate auction parameters.

4.3 Intermediate Computation Preservation

Parallel technical investigations demonstrate how intermediate computation preservation enables self-attention mechanisms to exhibit approximately linear scaling relationships with input sequence length, substantially reducing latency while increasing memory footprint requirements [6]. The preservation mechanism stores intermediate attention computations for reuse across token generation steps, eliminating redundant calculations and accelerating sequential token production. For marketplace implementations where response characteristics frequently exhibit brevity and formulaic structure, this optimization permits architecture allocation of computation budget toward contextual comprehension rather than standardized text regeneration, effectively concentrating analytical capacity in understanding while automating generation.

4.4 Compositional Generation Approaches

The coordination mechanism can exploit these preservation mechanisms to implement compositional generation approaches that maximize computational reuse [5]. By maintaining preserved states at varying granularity tiers encompassing user-level context, content-level context, and transaction-level context, the system can assemble complete prompts from preserved components and synthesize only novel segments specific to individual requests. This methodology transforms the latency distribution from one dominated by complete architecture inference to one where preserved state retrieval handles primary computation and architecture invocation addresses exclusively unique characteristics of individual decisions.

5. Economic Optimization and Infrastructure Management

5.1 Accelerator Utilization Through Workload Integration

Processing throughput and operational cost considerations emerge as indirect factors in marketplace coordination design [7]. Systems engineering research demonstrates that strategic integration of immediate-response and background workloads on shared computational accelerators, governed by latency-conscious scheduling algorithms, can substantially elevate aggregate utilization while maintaining strict latency assurances for time-critical traffic. Equivalent principles apply in marketplace frameworks where background simulation, budget pacing computation, and creative exploration can share computational infrastructure with immediate bidding, provided the coordination mechanism maintains precise performance models characterizing how request batching and resource interference influence latency distributions.

5.2 Infrastructure Deployment Economics

From an economic analysis perspective, recent examinations of dedicated infrastructure versus hosted neural language model deployments formalize cost as an optimization problem spanning architecture scale, request volume distributions, and hardware capital amortization schedules [8]. For marketplace platforms, this analysis directly intersects with operational performance indicators encompassing return on advertising expenditure or net interest margin calculations. The determination regarding architecture deployment on dedicated infrastructure versus consumption as external services depends fundamentally on request volume characteristics and cost function geometry, with economic breakeven thresholds shifting according to utilization patterns and extracted value from architecture intelligence.

5.3 Cost-Value Relationship Quantification

An appropriate architectural design objective involves establishing a continuous intelligence cost metric tracking token consumption, accelerator utilization intervals, or energy expenditure per decision, coupled to quantifiable value metrics encompassing click-through rate elevation, fraudulent

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

traffic reduction, or risk-adjusted loss minimization [7][8]. This coupling mechanism enables the coordination system to formulate economically rational determinations regarding expensive architecture invocation, conceptualizing cost as capital investment requiring quantifiable returns. Absent this feedback mechanism, coordination policies risk either excessive expenditure on intelligence providing negligible value increments or insufficient investment in scenarios where architecture insights would generate substantial operational improvements.

5.4 Service Tier Differentiation

The interaction between temporal constraints and economic factors can be administered through differentiated service tier structures that align computational investment with opportunity economic value [7]. High-value low-volume transaction opportunities might economically justify larger architectures with extensive contextual processing and comprehensive attention state pre-computation, operating within generous latency allocations permitting exhaustive analysis. Bulk long-tail traffic might be serviced entirely from preserved computational outputs or compact architectures operating within restrictive latency constraints where speed supersedes analytical sophistication. Suspicious or elevated-risk signals might activate deeper architectures with multi-phase reasoning operating in asynchronous processing paths where analytical thoroughness outweighs temporal efficiency.

5.5 System Observability Infrastructure

An operational dimension provides a foundation for all optimization activities [8][9]. Mature neural language serving infrastructures conceptualize observability as a fundamental architectural concern, aggregating granular metrics for latency percentile distributions, queue depth measurements, preservation mechanism hit rates, and per-participant cost attribution, channeling this telemetry data back into scheduling and routing policy formulation. Without this feedback infrastructure, sophisticated coordination degrades to approximation as the divergence between presumed and measured performance characteristics expands over operational timescales. The observability infrastructure must capture both system-level metrics and operational outcome measures, enabling the coordination mechanism to identify which determinations generate economic value and which represent inefficient computation.

Conclusion

Economically conscious and temporally optimized coordination of neural language architectures in immediate-response marketplace decision frameworks constitutes a comprehensive control architecture rather than an isolated optimization technique. That control architecture synthesizes contributions from systems engineering regarding acceleration and state preservation, financial models characterizing deployment economics, and specialized domain knowledge encompassing auction theory and risk quantification. The framework articulated here conceptualizes coordination as a policy formulation problem where determinations regarding architecture invocation, context assembly, and resource allocation are established dynamically according to request attributes, system state observations, and economic value functions. When implemented with precision, neural language architectures transition from exotic performance-limiting additions to governed computational instruments in tightly managed decision infrastructures. The technical advancement trajectory requires persistent refinement of coordination policies, strengthened integration between serving infrastructure and operational performance metrics, and continuous adaptation to evolving marketplace requirements and architecture capabilities.

Journal of Information Systems Engineering and Management
2025, 10(63s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

References

[1] Baolin Li, et al., "LLM Inference Serving: Survey of Recent Advances and Opportunities," arXiv, 2024. Available: https://arxiv.org/abs/2407.12391
[2] Xupeng Miao, et al., "Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems," arXiv, 2023. Available: https://arxiv.org/abs/2312.15234
[3] Jiangsu Pan and G. Li, "A Survey of LLM Inference Systems," arXiv, 2025. Available: https://arxiv.org/abs/2506.21901
[4] Soheil Feizi, et al., "Online Advertisements with LLMs: Opportunities and Challenges," ACM SIGecom Exchanges, vol. 22, no. 2, 2025. Available: https://arxiv.org/abs/2311.07601
[5] In Gim et al., "Prompt Cache: Modular Attention Reuse for Low-Latency Inference," Proceedings of MLSys, 2024. Available: https://arxiv.org/abs/2311.04934
[6] Rohan Paul, "KV Caching in LLM Inference: A Comprehensive Review," May 2025. Available: https://www.rohan-paul.com/p/kv-caching-in-llm-inference-a-comprehensive
[7] Tianhui Sun, et al., "HyGen: Efficient LLM Serving via Elastic Online-Offline Request Co-location," arXiv, 2025. Available: https://arxiv.org/abs/2501.14808
[8] Guoming Pan and H. Wang, "A Cost-Benefit Analysis of On-Premise Large Language Model Deployment: Breaking Even with Commercial LLM Services," arXiv, 2025. Available: https://arxiv.org/abs/2509.18101
[9] Rick Hightower, "The Economics of Deploying Large Language Models: Costs, Value, and a 99.7% Savings Story," Cloudurable Blog, Jul. 2025. Available: https://cloudurable.com/blog/the-economics-of-deploying-large-language-models-costs-value-and-a-997-savings-story/
[10] Martin Reisenbichler, et al., "Applying Large Language Models to Sponsored Search Advertising," Marketing Science (in press), 2025. Available: https://pubsonline.informs.org/doi/10.1287/mksc.2023.0611