A Tiered Pricing Model for Serverless LLM Inference Based on Cold/Warm/Hot Starts

groda

October 2025

Abstract

Serverless computing offers cost-efficient Large Language Model (LLM) inference but incurs high cold start latency when loading model checkpoints into GPUs. While non-serverless setups achieve low latency through manual pre-caching to NVMe SSDs, serverless environments limit such control due to ephemeral resources. The ServerlessLLM framework mitigates this through GPU server RAM/SSD storage, optimized checkpoint formats, and token caching [4]. We propose a novel pricing model with three user-facing tiers—cold start (slow, low-cost), warm start (faster, medium-cost), and hot start (fastest, high-cost)—bringing non-serverless efficiency to serverless without dedicated GPUs. Unlike providers' opaque pre-caching and per-token pricing (e.g., AWS SageMaker Serverless, Google Cloud), our model uses ServerlessLLM's scheduler to guarantee latency tiers, inspired by Hadoop's data locality [3]. This approach provides user choice and transparency, addressing a gap in serverless LLM pricing.

1 Introduction

Serverless computing dynamically allocates GPU resources for Large Language Model (LLM) inference, minimizing costs but introducing significant latency during cold starts, where a model's checkpoint, such as the 130GB LLaMA-2-70B, is loaded into GPU memory. The *ServerlessLLM* article highlights two primary latency sources: large checkpoint sizes, requiring approximately 26 seconds to fetch from AWS S3 over a 5GB/s network, and complex loading processes, taking around 84 seconds with PyTorch's `torch.load()` [4]. In non-serverless environments, developers mitigate this by pre-caching checkpoints to local NVMe SSDs, achieving faster load times, but serverless's ephemeral resources prevent such manual optimization [1]. Current providers, such as AWS SageMaker Serverless Inference, Azure Machine Learning, and Google Cloud Vertex AI, employ opaque pre-caching (e.g., escrowing models to SSD) and charge based on per-token or per-millisecond rates, without offering user-selectable latency tiers [1, 6, 5]. AWS Lambda, while Amazon's flagship serverless platform, is less suited for large-scale LLM inference due to resource constraints, making SageMaker Serverless the more relevant comparison [2].

ServerlessLLM addresses these challenges by pre-caching checkpoints in GPU server's RAM (~50GB/s) or SSD (~7GB/s), using a custom checkpoint format and pipelined loading to reduce cold start latency from ~110 seconds to ~15–25 seconds, a 3.6–8.2× improvement over PyTorch. Additionally, token caching stores key-value (KV) caches in RAM/SSD to accelerate warm starts for similar queries [5]. Building on these innovations, we propose a pricing model with three tiers based on response speed: cold start (1 unit per query, slowest), warm start (5 units, faster), and hot start (20 units, fastest). This model brings the low-latency efficiency of non-serverless setups, such as pre-caching and token caching, to serverless computing without requiring dedicated GPUs, using ServerlessLLM's scheduler to route queries to servers with pre-cached checkpoints.

Inspired by Hadoop's data locality principle [3], it offers explicit latency guarantees, addressing a gap in serverless LLM pricing.

2 Related Work: Data Locality in Distributed Systems

The proposed model draws inspiration from Hadoop's data locality principle, where MapReduce tasks are scheduled on nodes storing relevant HDFS data blocks to minimize network transfers [3]. Similarly, ServerlessLLM pre-caches model checkpoints and KV caches in GPU server RAM or SSD, reducing inference latency by ensuring data proximity to compute resources [4]. While pre-caching to SSD is a standard practice in non-serverless ML, such as downloading checkpoints to NVMe before `torch.load()` [1], serverless environments limit manual caching due to resource ephemerality. Our model extends Hadoop's locality concept to serverless LLM inference, offering user-facing latency tiers that leverage ServerlessLLM's advanced pre-caching and scheduling, a novel approach compared to existing provider models.

3 Proposed Tiered Pricing Model

The pricing model introduces three tiers based on response speed, abstracting infrastructure complexities such as checkpoint storage and token caching, while providing clear latency-cost trade-offs. Unlike dedicated GPU setups, it uses ServerlessLLM's scheduler to dynamically route queries to servers with pre-cached checkpoints, preserving serverless's resource-sharing efficiency. The tiers are:

- **Cold Start Tier (1 unit per query)**: This tier offers the slowest response, fetching checkpoints from remote cloud storage (e.g., AWS S3) without prior caching or token reuse. It incurs a latency of ~60–120 seconds (e.g., 26 seconds to fetch a 130GB checkpoint, 84 seconds to load), leveraging low-cost storage (~$0.023/GB/month for S3). It suits budget-conscious users, such as hobbyists, who tolerate longer wait times.

- **Warm Start Tier (5 units per query)**: This tier provides faster responses by pre-caching checkpoints on GPU server SSDs (~7GB/s) using ServerlessLLM's optimized checkpoint format. Cold starts take ~15–25 seconds (less than 5 seconds to fetch, 10–20 seconds to load), while warm starts, enabled by token caching on SSD or RAM, achieve ~50–100ms per token for similar queries. Priced to reflect SSD costs (~$0.10–0.125/GB/month, ~4–5× S3), it targets semi-interactive applications like customer support bots.

- **Hot Start Tier (20 units per query)**: This tier delivers the fastest response, pre-caching checkpoints in GPU server RAM (~50GB/s) with optimized loading. Cold starts require ~10–15 seconds (less than 2 seconds to fetch, 10–13 seconds to load), warm starts achieve less than 100ms with token caching, and true hot starts (pre-loaded models in GPU memory) reach ~10–50ms. Reflecting high RAM costs (~10–20× SSD), it serves real-time applications like interactive chatbots or live analytics.

3.1 Cost Basis

Costs are expressed in relative units (1, 5, 20), reflecting infrastructure pricing ratios: remote storage (e.g., S3 at ~$0.023/GB/month), SSD (~4–5× S3), and RAM (~10–20× SSD). Actual prices depend on provider rates and query volume.

4 Alignment with ServerlessLLM

The pricing model is tightly integrated with ServerlessLLM's technical innovations to deliver its promised latency guarantees. ServerlessLLM pre-caches model checkpoints in GPU server RAM or SSD, utilizing a custom checkpoint format and pipelined loading to achieve cold start times 3.6–8.2× faster than PyTorch's `torch.load()` [4]. This enables the warm and hot tiers to deliver significantly reduced latencies compared to standard pre-caching practices. Additionally, ServerlessLLM's token caching stores key-value (KV) caches in RAM or SSD, accelerating warm starts for queries with overlapping tokens, a critical feature for interactive workloads [5]. The system's scheduler plays a pivotal role by routing queries to servers where the required model is pre-cached in RAM (hot tier) or SSD (warm tier), rather than relying on dedicated GPUs, thus maintaining serverless's multi-tenant efficiency. By leveraging the unused capacity of GPU servers, which typically have hundreds of gigabytes of DRAM and terabytes of SSD storage, ServerlessLLM supports the storage of multiple model checkpoints and KV caches, ensuring scalability across diverse workloads.

5 Benefits of the Pricing Model

The proposed pricing model offers several advantages for both users and cloud providers. It empowers users to choose between speed and cost, enabling hobbyists to opt for the low-cost cold start tier (1 unit) while real-time applications, such as chatbots, can utilize the high-performance hot start tier (20 units). This flexibility mirrors the efficiency of non-serverless setups, where developers manually pre-cache models and optimize inference, but delivers it within a fully managed serverless environment, justifying the higher costs of warm (5 units) and hot tiers. For providers, the model optimizes revenue by charging premiums for faster tiers, offsetting the higher costs of SSD and RAM infrastructure. The cold/warm/hot-start terminology provides transparency, clearly communicating latency expectations (e.g., less than 15 seconds for hot starts), unlike the opaque caching mechanisms of existing providers. By leveraging ServerlessLLM's advanced pre-caching and token caching, the model introduces a novel approach to serverless LLM inference, offering guaranteed performance that bridges the gap between non-serverless efficiency and serverless scalability.

6 Challenges and Solutions

Implementing the proposed pricing model presents several challenges, primarily due to the resource constraints and dynamic nature of serverless environments. GPU server RAM, typically limited to around 512GB, restricts the number of model checkpoints that can be pre-cached for the hot tier. To address this, ServerlessLLM's scheduler can prioritize high-demand models for RAM storage, relegating less frequently used models to SSD or remote storage, ensuring efficient resource utilization. Another challenge is the variability in warm and hot start performance, as the availability of pre-loaded models or cached tokens depends on workload patterns. This can be mitigated by guaranteeing maximum cold start latencies for each tier (e.g., 25 seconds for warm, 15 seconds for hot), treating warm and hot starts as performance bonuses when token caching or pre-loaded models are available. Calibrating the pricing ratios (1, 5, 20 units) also requires careful consideration to align with infrastructure costs, such as RAM (~$1–2/GB) versus SSD (~$0.10/GB). Providers can address this by analyzing cost structures and query patterns to fine-tune these multipliers, ensuring economic viability. Finally, users may need clarity on the latency implications of each tier. To facilitate adoption, providers can integrate user interface indicators, such as "Hot: <15 seconds," into their platforms, enhancing transparency and ease of use.

7 Comparison to Existing Models

Pre-caching checkpoints to NVMe SSD is a standard practice in non-serverless ML, where developers download models before `torch.load()` to reduce latency [1]. However, in serverless environments, ephemeral resources limit manual control. AWS SageMaker Serverless Inference, a key platform for serverless ML, employs opaque pre-caching (e.g., escrowing models to SSD) and charges per inference duration (e.g., $0.0003/second with Provisioned Concurrency) [1]. Similarly, Azure Machine Learning and Google Cloud Vertex AI use per-token or per-millisecond pricing with features like context caching (75% discount for cached tokens), but lack user-selectable latency tiers [6, 5]. AWS Lambda, Amazon's flagship serverless platform, supports lightweight workloads but is less suited for large-scale LLM inference due to resource constraints (e.g., 10GB memory limit) [2]. Our model leverages ServerlessLLM's advanced pre-caching, optimized loading, and token caching to offer explicit cold/warm/hot-start tiers in serverless, delivering non-serverless efficiency without dedicated GPUs.

8 Conclusion

The proposed cold/warm/hot-start pricing model, with costs of 1, 5, and 20 units per query, introduces a user-centric approach to serverless LLM inference. By leveraging ServerlessLLM's pre-caching in RAM/SSD, optimized checkpoint loading, and token caching, it delivers the low-latency efficiency of non-serverless setups in a fully managed serverless environment, justifying premium prices for faster tiers. Unlike providers' opaque caching and consumption-based pricing, the model offers explicit latency guarantees using ServerlessLLM's scheduler to route queries to servers with pre-cached checkpoints, without requiring dedicated GPUs. Inspired by Hadoop's data locality, it extends pre-caching to user-facing tiers for real-time AI workloads. Future work could refine pricing ratios, explore dynamic warm start policies, and pilot the model with providers like AWS or xAI, potentially redefining serverless AI pricing.

References

[1] Amazon Web Services. Amazon sagemaker serverless inference pricing. `https://aws.amazon.com/sagemaker/pricing/`, 2025.

[2] Amazon Web Services. Aws lambda pricing. `https://aws.amazon.com/lambda/pricing/`, 2025.

[3] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[4] Yao Fu, Leyang Xue, Yeqi Huang, Andrei-Octavian Brabete, Dmitrii Ustiugov, Yuvraj Patel, and Luo Mai. Serverlessllm: Low-latency serverless inference for large language models. *arXiv preprint arXiv:2407.19554*, 2024.

[5] Google Cloud. Vertex ai pricing. `https://cloud.google.com/vertex-ai/pricing`, 2025.

[6] Microsoft Azure. Azure machine learning pricing. `https://azure.microsoft.com/en-us/pricing/details/machine-learning/`, 2025.