Advancing Serverless Computing for Scalable AI Model Inference: Challenges and Opportunities

Li Wang*
Yankai Jiang*
wang.li4@northeastern.edu
jiang.yank@northeastern.edu
Northeastern University
Boston, MA, USA

Ningfang Mi
ningfang@ece.northeastern.edu
Northeastern University
Boston, MA, USA

Abstract

Artificial Intelligence (AI) model inference has emerged as a crucial component across numerous applications. Serverless computing, known for its scalability, flexibility, and cost-efficiency, is an ideal paradigm for executing AI model inference tasks. This survey provides a comprehensive review of recent research on AI model inference systems in serverless environments, focusing on studies published since 2019. We investigate system-level advancements aimed at optimizing performance and cost-efficiency through a range of innovative techniques. By analyzing high-impact papers from leading venues in AI model inference and serverless computing, we highlight key breakthroughs and solutions. This survey serves as a valuable resource for both practitioners and academic researchers, offering critical insights into the current state and future trends in integrating AI model inference with serverless architectures. To the best of our knowledge, this is the first survey that includes Large Language Models (LLMs) inference in the context of serverless computing.

CCS Concepts

• **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Artificial intelligence**; • **Computer systems organization** → **Cloud computing**.

Keywords

Serverless Computing, LLMs Inference, DL Inference, ML Inference

ACM Reference Format:

Li Wang, Yankai Jiang, and Ningfang Mi. 2024. Advancing Serverless Computing for Scalable AI Model Inference: Challenges and Opportunities. In *10th International Workshop on Serverless Computing (WoSC10 '24), December 2–6, 2024, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3702634.3702950

1 Introduction

Artificial Intelligence (AI) model inference has become a crucial element in the deployment of AI applications, driving real-time analytics, recommendations, and decision-making systems across a wide range of industries. As AI models continue to grow in complexity and size, the need for efficient and scalable inference architectures has risen significantly. Serverless computing, a cloud-native paradigm that offers pay-per-use scalability while abstracting infrastructure management from developers, has emerged as a promising alternative for AI model inference. However, significant challenges persist in deploying AI models on serverless architectures, particularly when handling large-scale models such as large language models (LLMs), optimizing resource usage, and ensuring low-latency responses under dynamic inference requests.

Motivation. The rapid advancement in this field has made it challenging for both academic researchers and industry practitioners to keep pace with the latest innovations and identify the most promising solutions for real-world deployment. Previous research has attempted to analyze the challenges in deploying AI model inference serving systems on serverless platforms. For instance, [4] explores how serverless computing can support various stages of the machine learning pipeline, but primarily focuses on traditional ML models. It lacks in-depth coverage of the growing importance of Generative Artificial Intelligence models (e.g., LLMs) and only offers limited exploration of AI inference systems for real-time, large-scale applications. Similarly, [18] examines the challenges and optimization opportunities in deploying large-scale deep learning (DL) models and meeting strict Service Level Objectives (SLOs) for real-time inference. While it provides valuable insights into the role of GPU access in serverless architectures, the survey does not explore other essential aspects of serverless computing, such as memory management, checkpointing, or auto-scaling, all of which are particularly relevant for AI model inference tasks.

The limitations of existing surveys underscore the need for a comprehensive study that tackles the challenges of AI model inference on serverless platforms, particularly in the context of resource management, scalability, and optimization techniques. This new survey aims to bridge critical gaps in current research by providing an in-depth analysis of emerging LLMs, auto-scaling and scheduling strategies, resource utilization, and real-time performance optimization within serverless environments.

Contributions. In this survey, we methodically select and narrow down to 31 high-quality research papers exclusively focused on AI model inference serving systems, published between 2019 and 2024. Our selection criteria emphasize publications from prestigious machine learning, deep learning, and systems venues (e.g., ASPLOS, ATC, CCGrid, OSDI, SC), alongside influential arXiv submissions from prominent industry and academic research groups.

*Both authors contributed equally to this research.

WoSC10 '24, December 2–6, 2024, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-1336-1/24/12
https://doi.org/10.1145/3702634.3702950

Table 1: Statistics of top ten topics in selected papers.

AI Models	ML-based	DL-based	LLMs-based
Resource management	[1, 3, 6, 7, 10, 12, 13, 15, 19, 22, 23, 25, 28, 31]	[5, 8, 11, 16, 20, 24, 27, 30]	[2, 9, 14, 17, 21, 26]
Cost-effectiveness	[1, 6, 7, 10, 12, 13, 15, 19, 22, 23, 25, 28, 31]	[5, 8, 11, 20, 24, 27, 30]	[2, 9, 14, 21, 26]
Distributed inference	[15, 19, 22, 25, 27, 28, 31]	[5, 10, 11, 16, 27]	[9, 14, 17, 26]
Cold start latency	[1, 7, 12, 13, 25, 28]	[7, 8, 16, 20, 23, 30]	[9, 26]
GPU utilization	[1, 6, 7, 12, 13, 19, 31]	[11, 16, 20]	[9, 14]
Bursty workloads	[1, 6, 12, 13, 25, 31]	[5, 16, 24]	[9, 26]
Scheduling	[1, 6, 12, 13, 28, 31]	[5, 20, 24]	[9, 26]
Batching	[1, 6, 12, 28, 31]	[5]	[26]
Auto-scaling	[22, 31]	[5, 20]	[14]
Model partitioning	[10]	[8, 30]	N/A

Table 1 summarizes the key characteristics of selected papers, excluding three surveys ([4, 18, 29]). The analysis is categorized by AI model types—ML-, DL-, and LLM-based inference-and focuses on the top ten performance and optimization metrics as follows. *Resource management* and *cost-effectiveness* are prominent focuses across all AI models, with *distributed inference* and *cold start latency* following closely. The primary challenges lie in *GPU utilization* and *bursty workloads*. To mitigate these issues and enhance resource utilization, strategies such as *scheduling*, *batching*, *auto-scaling*, and *model partitioning* are employed, aiming to optimize performance and cost-effectiveness, especially under heterogeneous and dynamic inference requests. These topics are explored in detail in the following sections.

2 Background and Challenges

Serverless platforms provide high scalability, making them ideal for dynamic environments such as AI model inference. The pay-per-use billing model charges users only for the actual inference time, eliminating costs associated with idle server resources and minimizing operational overhead. While serverless inference offers significant benefits, it also presents several challenges, such as unpredictable workload arrival patterns, cold start latency, limited control over the underlying infrastructure, and resource constraints for deploying large models or handling high-performance requirements.

2.1 Serverless AI Models Inference Workflow

The serverless platform automatically scales resources in response to fluctuating workloads, making it ideal for dynamic environments. Users are relieved of the burden of provisioning or maintaining servers, thereby reducing operational overhead. With a "pay-as-you-go" model, users are billed only for actual inference time, which can be more cost-effective than traditional inference systems, as it eliminates the expense associated with idle server time. These features make serverless computing an excellent choice for deploying AI inference tasks in a scalable and cost-efficient manner.

In AI inference tasks, the serverless paradigm enables users to upload and execute AI models on a serverless infrastructure without managing underlying servers. An overview of this serverless inference workflow is depicted in Fig. 1. Users send inference requests with input data as stateless functions, which load pre-trained models and perform the inference. The serverless function is executed in a Docker container, and the inference results are returned to the user. This approach to AI model inference using serverless infrastructure offers numerous benefits, including scalability, flexibility, and reduced operational complexity.

2.2 Challenges in Serverless Inference Systems

Bursty Workloads. The challenge of bursty workloads in serverless inference systems arises when there are sudden and unpredictable spikes in the volume of requests. These systems are designed to scale automatically based on demand, but handling abrupt surges in traffic can lead to latency issues or degraded performance. The infrastructure must rapidly allocate resources to meet these demand spikes, which however can result in cold start delays as new instances take time to initialize. Additionally, over-provisioning resources to prepare for these bursts can lead to inefficiencies and higher costs, making it challenging to strike the right balance between responsiveness and cost-effectiveness [1, 5, 6, 9, 12, 13, 16, 24–26, 31].

Cold Start Latency. Cold start latency is a common issue in serverless computing, referring to the delay incurred when initializing a serverless function for the first time, or after it has been idle. The issue becomes particularly pronounced in DL and LLMs inference tasks, where large models need to be loaded into memory. These cold starts occur when a function must be initialized from scratch, resulting in delays that can degrade performance for real-time applications [1, 7, 7–9, 12, 13, 16, 20, 23, 25, 26, 28, 30].

Resource Under/Over-provisioning. Achieving an optimal balance between performance (latency, throughput) and cost is a persistent challenge. AI inference requires significant computational power, and improper scaling can lead to either under-provisioning (causing performance bottlenecks) or over-provisioning (leading to unnecessary costs). While serverless computing offers dynamic scaling, many platforms fail to efficiently utilize underlying resources such as GPUs and memory, particularly for computation-heavy DL and LLM workloads. This inefficiency can lead to increased costs and suboptimal performance [1, 6, 7, 9, 11–14, 16, 19, 20, 31].

Stateful Workflows. Serverless platforms are inherently stateless, which poses a challenge for models that require stateful processing or complex inter-model communication. This is particularly relevant for distributed AI models, where coordination across multiple instances is necessary [7, 10, 23].

3 Optimal Strategies

Serverless AI inference systems focus on maximizing performance while minimizing costs by employing various advanced techniques. Batching and scheduling are critical for optimizing performance, particularly during bursty workloads. Model partitioning addresses the resource demands of large AI models by dividing them into

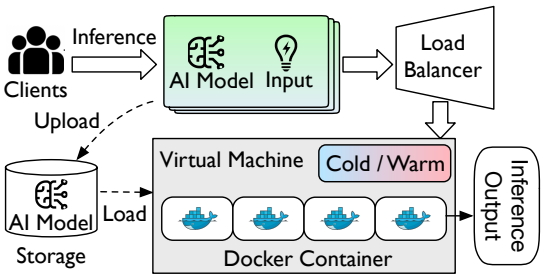

Figure 1: Workflow of serverless inference process.

smaller, manageable components that fit within the resource constraints of serverless platforms. Resource sharing strategies enhance the utilization of GPUs, memory, and containers across serverless functions, improving efficiency. Additionally, resource management systems optimize the allocation of resources in heterogeneous environments. Miscellaneous approaches further improve resource allocation and inter-process communication, enabling more efficient handling of large-scale AI inference tasks in serverless settings.

3.1 Batching and Scheduling

Efforts to mitigate cold start impacts and enhance throughput focus on dynamically batching requests and optimizing scheduling based on workload patterns, particularly for bursty workloads. BARISTA [5] tackles online resource configurations through a distributed, scalable dynamic resource allocation system that is composed of four main components: workload prediction, optimization formalization, suitable resource configuration, and resource allocation management. Similarly, AYCI [24] offers various open-source serverless DL inference environments, automating the evaluation of performance and assisting developers in estimating optimal serverless model serving configurations.

MArk [31] is a cloud-based system, which dynamically batches requests to improve resource utilization and reduce costs, using predictive auto-scaling to adjust resources based on traffic. MArk combines serverless (FaaS) for unpredictable workloads with IaaS for more stable tasks, recommending small IaaS instances with GPUs for low-latency inference. Building on MArk, BATCH [1] proposes a serverless framework with adaptive batching for efficient ML serving. This system's profiler monitors inference times and memory usage, while its performance optimizer calculates optimal batch sizes to ensure efficient utilization of resources and minimize data points required for model training. INFless [28] introduces batch processing with heterogeneous hardware support and the Long-Short Term Histogram (LSTH) strategy to reduce cold start times and resource wastage, significantly improving performance compared to BATCH and MArk. JointBatching [6] introduces a serverless ML inference system that employs batching and multiprocessing techniques, using a change-point detection algorithm to manage bursty workloads and Bayesian optimization to ensure latency SLOs.

3.2 Model Partitioning

To address the diverse resource demands of different layers in DL models, MOPAR [8] optimizes resource usage and reduces latency by vertically partitioning the model into slices of analogous layers. It exploits data compression and shared memory to accommodate varying resource configurations and cost requirements in different parts of the DL model. Similarly, Gillis [30] partitions large DL models across multiple serverless functions using latency-optimal and SLO-aware partitioning algorithms. MLModelDecomposition [10] introduces an efficient approach for decomposing ML models into slices, allowing the execution of large inference tasks as serverless functions. The process begins by verifying that the required software libraries and model payload fit within the serverless platform's constraints. The model is then decomposed into layers based on payload size, with an evaluation of storage and runtime memory requirements, allowing it to be executed as separate functions on the serverless platform.

3.3 Resource Sharing

Efficient GPU management techniques are essential for maximizing resource utilization while minimizing costs in high-performance AI inference tasks. FaST-GShare [11] comprises three key components: the FaST-manager, FaST-profiler, and FaST-scheduler, which work together to limit and isolate spatio-temporal resources for GPU multiplexing, monitor function performance, and allocate executions across GPU nodes to ensure maximum utilization while meeting SLOs. SMSS [7] tackles the limitations of serverless platforms in supporting stateful ML inference by employing a log-based workflow runtime and a two-layer GPU sharing mechanism. This design reduces cold start latency by facilitating both inter-model and intra-model GPU sharing.

Tetris [19] enhances resource usage through memory-efficient tensor sharing, which allows multiple ML models to reuse common tensors across different function instances. Tetris dynamically manages tensor sharing to balance performance and resource utilization, improving cost efficiency and scalability, especially in high-throughput, large-scale serverless deployments. GPUColdStarts [16] optimizes underutilized memory and network resources in serverless environments by leveraging remote memory pooling and hierarchical sourcing with a locality-aware autoscaler. This strategy cycles through GPU nodes in host machines before expanding to other hosts for cold start instantiations, minimizing redundant DL model transformations through download sharing.

Fifer [12] addresses microservice-agnostic scheduling and excessive container over-provisioning by optimizing resource management through efficient bin packing, function-aware container scaling, and LSTM-based request batching. Additionally, it proactively spawns containers to minimize cold start latency while ensuring SLO compliance. To further mitigate latency caused by model loading, Optimus [13] introduces an innovative inter-function model transformation mechanism within container operations. This mechanism enables rapid transitions between models in the same container using meta-operators specifically designed for AI models. Optimus also integrates advanced scheduling algorithms to manage model transformations efficiently, reducing delays and improving overall system performance.

3.4 Resource Management Systems

Pioneers have focused on developing efficient, high-performance resource management frameworks to recommend optimal resource

configurations for serverless AI inference workflows. These frameworks aim to address challenges in serverless ML inference, including diverse application requirements related to latency, cost, accuracy, and privacy, as well as the complexities of heterogeneous execution environments involving various hardware resources and accelerators at scale.

INFaaS [27] generates model variants and creates performance-cost profiles across different hardware platforms. It dynamically tracks the status of overloaded or interfered model variants using a state machine, enabling the efficient selection of the appropriate variant to meet specific application requirements. Similarly, AMPS-Inf [15] enables automatic customization of optimal execution and resource provisioning for large-scale distributed ML inference workloads. At its core, AMPS-Inf formulates and solves a Mixed-Integer Quadratic Programming (MIQP) problem to partition models and provision resources, minimizing costs while meeting SLO requirements.

3.5 Miscellaneous Areas

Several approaches have been developed to balance cost and performance for large-scale ML inference tasks. By leveraging cloud-based services such as publish-subscribe/queueing and object storage, FSD-Inference [22] enables efficient inter-process communication (IPC) for distributed ML inference workloads. This approach eliminates the need for traditional server-based solutions and offers significant cost savings and scalability, comparable to high-performance computing (HPC) setups while achieving high parallelism through Function-as-a-Service (FaaS).

In contrast, AsyFunc [25] optimizes resource allocation by separating resource-intensive tasks (e.g., model execution) from lighter ones (e.g., request handling) using asymmetric functions. It further enhances performance by implementing function fusion, which combines related tasks into a single flow, and predictive scaling to anticipate workload demands. These features reduce startup latency, making AsyFunc particularly effective for handling large-scale, unpredictable workloads that require both cost-efficiency and high-performance inference.

MLFaaS [23] generalizes ML inference pipelines by exposing the complete set of data path functions required by data scientists. It introduces an AI-based framework that recommends the optimal function compositions for serverless pipelines, aiming to improve Quality of Service (QoS) by minimizing the response time of ML inference tasks.

4 Emerging Research Fields

Recent advancements in serverless computing and AI inference have led to the emergence of several key research fields, including large language models (LLMs) inference, AI-driven scaling, and the exploration of security and privacy solutions in edge computing.

4.1 Serverless LLMs Inference

LLMs have gained widespread popularity for their powerful text-generation capabilities but face significant latency challenges during inference on serverless platforms due to the large size of model checkpoints and the complexity of loading them onto GPUs. Public cloud providers, open-source frameworks, and academic research are addressing these issues from various perspectives to optimize serverless LLM inference.

AWS Bedrock [2] offers a suite of high-performing foundation models to simplify the complexities of training LLMs and managing cloud infrastructure. Microsoft's Azure AI Studio [21] enables serverless LLM inference through a pay-per-use, token-based billing system, allowing users to deploy LLM models as serverless APIs without the need to host models within their subscriptions.

LoRA eXchange (LoRAX) [26] is a prominent open-source framework specifically designed to manage large-scale fine-tuned LLM inference tasks using shared GPU resources. This framework significantly enhances the overall performance of serverless LLM deployments by improving both throughput and latency in a scalable and resource-efficient manner.

ServerlessLLM [9] tackles the problem of long inference latencies in cloud-based LLM serving by introducing a novel checkpoint format and a multi-tiered loading system to speed up model loading. Additionally, it implements a locality-aware server allocation strategy to reduce cold start latency, ensuring faster model deployment in serverless environments. On the other hand, ENOVA [14] focuses on addressing the scalability and stability challenges of serverless LLM serving on multi-GPU clusters. ENOVA's service configuration and performance detection modules ensure optimal resource allocation and real-time monitoring of service quality, which are critical for accommodating diverse LLM inference tasks.

4.2 AI-based Scaling

Serverless platforms often employ reactive scaling mechanisms, which adjust resources based on current traffic patterns. Predictive scaling models, powered by AI-based workload forecasting, can improve performance by preemptively allocating resources based on expected demand. However, achieving accurate workload predictions remains challenging due to the dynamic nature of serverless environments.

To tackle the challenges of resource allocation and cold start latency in distributed AI inference tasks, ServingDI [20] introduces a hybrid scheduler that combines a greedy strategy with deep reinforcement learning (DRL) for optimal container allocation. Fifer [12] uses function-aware container scaling and LSTM-based request batching to proactively spawn containers to reduce cold start latency. Similarly, Gillis [30] encodes partitioning policies into a neural network, which is trained to iteratively optimize inference cost and latency.

4.3 Security and Privacy in Edge Computing

As serverless computing increasingly integrates into machine learning workflows, concerns surrounding security and data privacy have become paramount, particularly in edge computing. These paradigms aim to protect sensitive information while maintaining scalability and performance. Several key frameworks have emerged to address these concerns effectively.

TrustedLLMInference [17] proposes an innovative framework for securing distributed AI model inference using blockchain technology in edge computing environments. The use of blockchain also guarantees trust and verifiability in AI model inference, making TrustedLLMInference highly suitable for privacy-sensitive tasks

in distributed environments. MLEdge [3] allows for the efficient deployment of ML models closer to users, ensuring that sensitive data is processed locally. This approach thus minimizes the risks associated with data transmission to central servers, making it particularly beneficial for applications that require real-time, privacy-preserving inference.

5 Discussion

The rise of serverless computing has introduced a new paradigm for handling AI inference tasks, offering scalability, cost-effectiveness, and ease of deployment. While serverless architectures provide significant advantages, they also present unique challenges in serving LLMs inference, optimizing infrastructure usage, managing energy consumption, and fine-tuning AI inference pipelines. Nonetheless, several opportunities exist for improving AI model inference in serverless computing environments.

Serverless LLMs Inference. Current serverless inference frameworks face several limitations that make them less suitable for LLMs inference. A primary challenge is the latency caused by cold starts when loading large model checkpoints onto GPUs [9]. Additionally, resource allocation in serverless environments often falls short of meeting the resource-intensive demands of LLMs, especially for multi-GPU setups or large models that require substantial memory and compute power [9, 27]. Another critical issue is the lack of fine-grained control over GPU and memory resources, limiting the ability to optimize both throughput and cost efficiency [10, 16]. While public cloud providers such as AWS and Azure offer serverless LLM inference solutions [2, 21], these options come with limitations, including resource constraints, vendor lock-in, and high costs associated with scaling LLM workloads. The pay-as-you-go billing model can quickly become expensive due to the significant computational costs of loading and running LLMs, and the lack of tailored optimization for these models often results in suboptimal performance. To enhance the suitability of serverless inference frameworks for LLMs, improvements are required in cold start mitigation, efficient multi-GPU management, and dynamic scaling mechanisms that can predict and allocate resources preemptively, based on the dynamic LLM-specific workloads.

Infrastructure Advancement. AI inference workloads, especially for large models such as DL and transformer-based LLMs, require considerable computational resources for real-time processing. Traditional serverless environments, characterized by their ephemeral nature and resource-limited function instances, often struggle to meet these demands [14]. The stateless nature of serverless functions limits the ability to perform large-scale inference, particularly when models require continuous access to memory or GPUs for acceleration [9, 14, 26]. The delays in provisioning resources during function invocations can significantly impact latency-sensitive AI inference, such as real-time recommendations or autonomous systems. While some cloud providers offer GPU-enabled serverless platforms, these options are not always available in conventional FaaS offerings [2, 21]. This creates a gap in performance optimization for AI inference workloads. Advancements in hardware acceleration, particularly the integration of GPUs and specialized AI inference chips in serverless environments, present opportunities for significant performance improvements. By enabling fine-grained control over hardware resources, cloud providers can offer tailored serverless environments for AI inference, improving both resource utilization and cost-efficiency.

Energy Efficiency. The energy consumption of AI inference tasks is a growing concern, particularly as models become larger and more complex. AI models, such as transformers used in natural language processing (NLP), can have billions of parameters, leading to considerable power consumption during inference. AI inference tasks often rely on GPUs to accelerate computations, but GPUs are also energy-intensive. Serverless platforms typically lack fine-grained control over GPU resources, leading to potential inefficiencies in energy use [11, 12, 28]. Serverless functions scale horizontally based on demand, which may lead to redundant resource allocation and energy waste, particularly during bursty workloads or idle periods. The increased demand for real-time AI inference further drives up data center energy usage, requiring more efficient cooling systems and energy-efficient hardware configurations [2, 21]. Consequently, energy-efficient AI model inference is becoming a priority, driven by the environmental impact of large-scale AI workloads. Innovations such as energy-aware scheduling, efficient model partitioning, and intelligent resource allocation can help reduce the carbon footprint of AI inference tasks. Techniques such as model quantization and pruning [26] can also lower the energy requirements when deploying large AI models in serverless environments.

AI Model Inference Pipelines. Serverless AI inference pipelines introduce several unique challenges due to their modular, stateless, and event-driven nature. Efficiently deploying, scaling, and managing AI inference workloads requires careful consideration of various pipeline stages [4, 15, 23]. Inference tasks often involve multiple stages, from data preprocessing to model loading and prediction generation. Serverless platforms must effectively orchestrate these stages while minimizing the latency between function invocations. Furthermore, large AI models cannot easily be loaded and executed in a single serverless function due to resource constraints. Therefore, efficiently partitioning models across functions and managing inter-function communication remains an ongoing challenge. Furthermore, AI inference tasks, such as those involving LLMs or ensemble models, may require complex orchestration across multiple serverless functions, which increases the risk of latency spikes or bottlenecks. AI-based optimization techniques can help enhance serverless inference pipelines [23], particularly in workload prediction and resource management. For example, ML models can be used to predict traffic patterns and proactively scale resources, thus reducing both latency and costs. Additionally, reinforcement learning can dynamically adjust function configurations to improve the overall efficiency of the serverless AI inference pipeline.

AI-based Optimization Strategies. AI-based optimization strategies present opportunities to enhance performance and cost-efficiency in serverless AI inference workflows. However, implementing these strategies introduces several challenges, including resource unpredictability and the need for real-time adaptability. The rise of edge computing presents an opportunity to offload AI inference tasks from centralized cloud servers to edge devices, reducing both latency and energy consumption. By integrating serverless architectures with edge computing, AI inference tasks can be processed closer to the data source, improving performance

and reducing the load on central cloud infrastructure. Federated learning and distributed AI models also offer avenues for privacy-preserving inference at the edge [3, 17].

6 Conclusion

This paper has surveyed the key challenges and solutions in serverless computing for AI model inference. By addressing issues such as cold start latency, resource utilization, cost optimization, and scalability, recent research has paved the way for more efficient and scalable serverless inference systems. Innovative frameworks and techniques, including adaptive batching, hybrid scheduling, model partitioning, and GPU sharing, are driving significant improvements in the performance and cost-effectiveness of serverless platforms. As the demand for large-scale AI inference continues to grow, advancements in serverless computing will be essential to meet the requirements of real-time, high-performance applications. The intersection of infrastructure management, energy consumption, AI inference pipelines, and AI-driven optimization strategies offers rich opportunities for innovation in improving serverless AI inference workflows. By addressing these challenges, the serverless paradigm can evolve into a viable solution for real-time AI inference at scale, enabling a new generation of AI applications.

Acknowledgments

We extend our gratitude to Professor Devesh Tiwari and the reviewers for their valuable feedback. This work is partially supported by the National Science Foundation Award CNS-2008072 and the National Science Foundation Award OCA-2417715.

References

[1] Ahsan Ali, Riccardo Pinciroli, Feng Yan, and Evgenia Smirni. 2020. BATCH: Machine Learning Inference Serving on Serverless Platforms with Adaptive Batching. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–15.

[2] Amazon. 2024. *AWS Bedrock*. https://docs.aws.amazon.com/bedrock/

[3] Ta Phuong Bac, Minh Ngoc Tran, and YoungHan Kim. 2022. Serverless Computing Approach for Deploying Machine Learning Applications in Edge Layer. In *2022 International Conference on Information Networking (ICOIN)*. 396–401.

[4] Amine Barrak, Fabio Petrillo, and Fehmi Jaafar. 2022. Serverless on Machine Learning: A Systematic Mapping Study. *IEEE Access* 10 (2022), 99337–99352.

[5] Anirban Bhattacharjee, Ajay Dev Chhokra, Zhuangwei Kang, Hongyang Sun, Aniruddha Gokhale, and Gabor Karsai. 2019. BARISTA: Efficient and Scalable Serverless Serving System for Deep Learning Prediction Services. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*. 23–33.

[6] Shen Cai, Zhi Zhou, Kongyange Zhao, and Xu Chen. 2023. Cost-Efficient Serverless Inference Serving with Joint Batching and Multi-Processing. In *Proceedings of the 14th ACM SIGOPS Asia-Pacific Workshop on Systems* (Seoul, Republic of Korea) *(APSys '23)*. 43–49.

[7] Zinuo Cai, Zebin Chen, Ruhui Ma, and Haibing Guan. 2023. SMSS: Stateful Model Serving in Metaverse With Serverless Computing and GPU Sharing. *IEEE J.Sel. A. Commun.* 42, 3 (dec 2023), 799–811.

[8] Jiaang Duan, Shiyou Qian, Dingyu Yang, Hanwen Hu, Jian Cao, and Guangtao Xue. 2024. MOPAR: A Model Partitioning Framework for Deep Learning Inference Services on Serverless Platforms. *ArXiv* abs/2404.02445 (2024).

[9] Yao Fu, Leyang Xue, Yeqi Huang, Andrei-Octavian Brabete, Dmitrii Ustiugov, Yuvraj Patel, and Luo Mai. 2024. ServerlessLLM: Low-Latency Serverless Inference for Large Language Models. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*. USENIX Association, Santa Clara, CA, 135–153.

[10] Adrien Gallego, Uraz Odyurt, Yi Cheng, Yuandou Wang, and Zhiming Zhao. 2024. Machine Learning Inference on Serverless Platforms Using Model Decomposition. In *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing* (Taormina (Messina), Italy) *(UCC '23)*. Article 33, 6 pages.

[11] Jianfeng Gu, Yichao Zhu, Puxuan Wang, Mohak Chadha, and Michael Gerndt. 2023. FaST-GShare: Enabling Efficient Spatio-Temporal GPU Sharing in Serverless Computing for Deep Learning Inference. In *Proceedings of the 52nd International Conference on Parallel Processing* (Salt Lake City, UT, USA) *(ICPP '23)*. 635–644.

[12] Jashwant Raj Gunasekaran, Prashanth Thinakaran, Nachiappan C. Nachiappan, Mahmut Taylan Kandemir, and Chita R. Das. 2020. Fifer: Tackling Resource Underutilization in the Serverless Era. In *Proceedings of the 21st International Middleware Conference* (Delft, Netherlands) *(Middleware '20)*. 280–295.

[13] Zicong Hong, Jian Lin, Song Guo, Sifu Luo, Wuhui Chen, Roger Wattenhofer, and Yue Yu. 2024. Optimus: Warming Serverless ML Inference via Inter-Function Model Transformation. In *Proceedings of the Nineteenth European Conference on Computer Systems* (Athens, Greece) *(EuroSys '24)*. 1039–1053.

[14] Tao Huang, Pengfei Chen, Kyoka Gong, Jocky Hawk, Zachary Bright, Wenxin Xie, Kecheng Huang, and Zhi Ji. 2024. ENOVA: Autoscaling towards Cost-effective and Stable Serverless LLM Serving. *arXiv preprint arXiv:2407.09486* (2024).

[15] Jananie Jarachanthan, Li Chen, Fei Xu, and Bo Li. 2021. AMPS-Inf: Automatic Model Partitioning for Serverless Inference with Cost Efficiency. In *Proceedings of the 50th International Conference on Parallel Processing* (Lemont, IL, USA) *(ICPP '21)*. Article 14, 12 pages.

[16] Justin San Juan and Bernard Wong. 2023. Reducing the Cost of GPU Cold Starts in Serverless Deep Learning Inference Serving. In *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 225–230.

[17] Rabimba Karanjai and Weidong Shi. 2024. Trusted LLM Inference on the Edge with Smart Contracts. In *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 1–7.

[18] Kamil Kojs. 2023. A Survey of Serverless Machine Learning Model Inference. *arXiv preprint arXiv:2311.13587* (2023).

[19] Jie Li, Laiping Zhao, Yanan Yang, Kunlin Zhan, and Keqiu Li. 2022. Tetris: Memory-efficient serverless inference through tensor sharing. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*.

[20] Kunal Mahajan and Rumit Desai. 2022. Serving distributed inference deep learning models in serverless computing. In *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*. 109–111.

[21] Microsoft. 2024. *Microsoft Azure AI Studio*. https://learn.microsoft.com/en-us/azure/ai-studio

[22] Joe Oakley and Hakan Ferhatosmanoglu. 2024. FSD-Inference: Fully Serverless Distributed Inference with Scalable Cloud Communication. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 2109–2122.

[23] Efterpi Paraskevoulakou and Dimosthenis Kyriazis. 2023. ML-FaaS: Toward Exploiting the Serverless Paradigm to Facilitate Machine Learning Functions as a Service. *IEEE Transactions on Network and Service Management* 20, 3 (2023), 2110–2123.

[24] Subin Park, Jaeghang Choi, and Kyungyong Lee. 2022. All-you-can-inference: serverless DNN model inference suite. In *Proceedings of the Eighth International Workshop on Serverless Computing* (Quebec, Quebec City, Canada) *(WoSC '22)*. 1–6.

[25] Qiangyu Pei, Yongjie Yuan, Haichuan Hu, Qiong Chen, and Fangming Liu. 2023. AsyFunc: A High-Performance and Resource-Efficient Serverless Inference System via Asymmetric Functions. In *Proceedings of the 2023 ACM Symposium on Cloud Computing* (Santa Cruz, CA, USA) *(SoCC '23)*. 324–340.

[26] Predibase. 2024. LoRAX: Multi-LoRA inference server that scales to 1000s of fine-tuned LLMs. https://github.com/predibase/lorax.

[27] Francisco Romero, Qian Li, Neeraja J. Yadwadkar, and Christos Kozyrakis. 2021. INFaaS: Automated Model-less Inference Serving. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*. USENIX Association, 397–411.

[28] Yanan Yang, Laiping Zhao, Yiming Li, Huanyu Zhang, Jie Li, Mingyang Zhao, Xingzhen Chen, and Keqiu Li. 2022. INFless: a native serverless system for low-latency, high-throughput inference. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Lausanne, Switzerland) *(ASPLOS '22)*. 768–781.

[29] Zhisheng Ye, Wei Gao, Qinghao Hu, Peng Sun, Xiaolin Wang, Yingwei Luo, Tianwei Zhang, and Yonggang Wen. 2024. Deep Learning Workload Scheduling in GPU Datacenters: A Survey. *ACM Comput. Surv.* 56, 6, Article 146 (jan 2024), 38 pages.

[30] Minchen Yu, Zhifeng Jiang, Hok Chun Ng, Wei Wang, Ruichuan Chen, and Bo Li. 2021. Gillis: Serving Large Neural Networks in Serverless Functions with Automatic Model Partitioning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. 138–148.

[31] Chengliang Zhang, Minchen Yu, Wei Wang, and Feng Yan. 2019. {MArk}: Exploiting cloud services for {Cost-Effective},{SLO-Aware} machine learning inference serving. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. 1049–1062.