Cloud-Native Serverless Machine Learning Inference on Kubernetes

Jinghao Sun[1]

Project Submission
202283910008@nuist.edu.cn

Abstract. Cloud computing has evolved rapidly from virtualized infrastructure to fully cloud-native architectures that emphasize scalability, elasticity, and automation. In recent years, the combination of Kubernetes, serverless computing, and machine learning (ML) has emerged as a powerful paradigm for deploying intelligent applications in the cloud. This paper explores cloud-native serverless machine learning inference using Kubernetes-based technologies, with a particular focus on Knative and KServe.

We present the motivation behind serverless ML inference, review related work in cloud-native ML platforms, and describe a reference system architecture for deploying machine learning models as serverless services. A practical implementation is demonstrated using KServe to deploy a scikit-learn model on Kubernetes with automatic scaling to zero and request-driven activation. Experimental observations highlight the benefits of this approach in terms of resource efficiency, scalability, and operational simplicity, as well as challenges such as cold-start latency and system complexity.

This work aims to provide an accessible yet technically accurate introduction to serverless ML inference for students and practitioners, while demonstrating how modern cloud computing technologies can be integrated to support intelligent, on-demand services.

Keywords: Cloud Computing · Kubernetes · Serverless Computing · Machine Learning Inference · KServe

1 Introduction

Cloud computing has transformed how applications are designed, deployed, and operated. Traditional monolithic applications hosted on fixed servers have gradually been replaced by microservices, containers, and orchestration platforms. Kubernetes has become the de facto standard for container orchestration, providing automated deployment, scaling, and management of containerized applications.

At the same time, machine learning workloads are increasingly deployed in cloud environments. While training often requires specialized hardware and long-running jobs, model inference typically involves serving predictions to users with low latency and high availability. Managing inference services manually can be complex and inefficient, especially when workloads are highly variable.

Serverless computing addresses this challenge by abstracting infrastructure management away from developers and enabling applications to scale automatically based on demand. When combined with Kubernetes, serverless platforms such as Knative allow services to scale down to zero when idle and scale up rapidly in response to incoming requests.

This paper investigates serverless machine learning inference in a cloud-native environment. The main contributions of this report are:

- A structured overview of serverless computing for ML inference in the cloud.
- A cloud-native system architecture based on Kubernetes, Knative, and KServe.
- A practical case study deploying a machine learning model as a serverless service.
- A discussion of benefits, limitations, and open challenges.

2 Literature Review

2.1 Cloud-Native Computing

Cloud-native computing emphasizes loosely coupled services, containerization, and continuous delivery. Kubernetes plays a central role in this ecosystem by providing primitives such as Pods, Services, and Deployments that enable scalable application management. Cloud-native design principles promote resilience, scalability, and automation across distributed systems.

2.2 Serverless Computing

Serverless computing, also known as Function-as-a-Service (FaaS), allows developers to deploy code without managing servers. In this model, resources are allocated dynamically based on workload demand. On Kubernetes, serverless capabilities are typically implemented using additional control planes that manage scaling, routing, and lifecycle events.

2.3 Machine Learning Inference Platforms

Machine learning inference platforms aim to simplify the deployment and operation of trained models. These platforms often abstract away low-level concerns such as container management, networking, and scaling, allowing data scientists and developers to focus on model development and experimentation.

3 System Architecture

3.1 Overview

The proposed system architecture is based on a layered cloud-native design. Kubernetes serves as the foundation, providing container orchestration and resource management. On top of Kubernetes, a serverless layer enables request-driven scaling and traffic management. Finally, a machine learning serving layer manages model loading and inference requests.

This layered approach improves modularity and allows each component to evolve independently while maintaining a consistent operational model.

3.2 Key Components

Kubernetes Kubernetes manages container scheduling, service discovery, and lifecycle management. It ensures that inference workloads are deployed reliably and can recover from failures.

Serverless Control Plane The serverless control plane enables autoscaling, including scaling workloads down to zero during idle periods. It also provides traffic routing and request activation mechanisms that are essential for on-demand services.

Machine Learning Serving Layer The ML serving layer is responsible for loading trained models, exposing prediction endpoints, and handling inference requests. It abstracts framework-specific details and provides a unified interface for different model types.

4 Experiment Setup and Evaluation

4.1 Experimental Environment

The experimental setup was based on a local Kubernetes cluster. The cluster was configured with a container runtime and a networking plugin suitable for service-to-service communication. Serverless and machine learning serving components were installed as extensions to the base Kubernetes system.

A simple classification model trained on the Iris dataset was selected as the inference target due to its simplicity and suitability for demonstration purposes.

4.2 Model Deployment

The trained model was deployed using a declarative configuration that specified the model type, storage location, and runtime environment. Once deployed, the inference service was automatically exposed through a network endpoint managed by the serverless layer.

The deployment process required minimal manual configuration, highlighting the usability benefits of cloud-native ML serving platforms.

4.3 Evaluation Observations

During evaluation, the inference service exhibited elastic behavior, scaling down to zero instances when idle and scaling up in response to incoming requests. This behavior demonstrates efficient resource utilization and aligns with the core principles of serverless computing.

However, a noticeable delay occurred when the service was activated after a period of inactivity. This cold-start latency is a known limitation of serverless systems and may affect latency-sensitive applications.

5 Discussion

5.1 Benefits

The cloud-native serverless inference approach offers several advantages:

- Reduced operational complexity
- Improved resource efficiency
- Automatic scalability based on demand
- Consistent deployment workflows for ML models

5.2 Limitations and Challenges

Despite its strengths, the approach introduces several challenges:

- Cold-start latency for infrequently used services
- Increased architectural complexity
- Higher learning curve for beginners

Balancing these trade-offs is essential when deciding whether to adopt serverless ML inference in production environments.

6 Conclusion

This paper examined serverless machine learning inference as an emerging trend in cloud computing. By combining Kubernetes-based orchestration with serverless execution models, machine learning models can be deployed as scalable, on-demand services.

The experimental observations confirm that this approach can significantly reduce resource waste while simplifying deployment workflows. At the same time, challenges such as cold-start latency and system complexity remain important considerations for future work.

Overall, cloud-native serverless ML inference represents a promising direction for building intelligent cloud applications and provides valuable insights into the evolution of modern cloud computing platforms.