# Research Proposal

## Title

Scalable Serving System for Large Language Models with Kubernetes and Nvidia MIG

## Motivation

As the computing power of a single GPU continues to grow, many workloads today cannot fully utilize an entire GPU, which results in internal resources fragmentation. Therefore, technology that enables GPU sharing is needed. Multi-Instance GPU (MIG) is such a new feature introduced by Nvidia's A100 GPU that can physically partition one GPU into many MIG slices. With MIG, A100 can be the most efficient GPU ever.

On the other hand, with the rise of cloud computing, the new serverless platforms have emerged. Due to its resource management-free nature, auto-scaling, and cost-efficiency, serverless platform has become one of the preferred choices for users, making serverless a key focus in recent cloud computing development.

Therefore, we want to combine these two key features to develop a Scalable Serving System for Large Language Models with Kubernetes and Nvidia MIG, which can automatically scale up or out based on the throughput of LLM inference, at the same time maintaining high GPU utilization with MIG.

## Objective

The primary objective of this project is to design a serving system that automatically scales based on the runtime behavior of LLM inference, while using MIG to address external resource fragmentation, improving resource utilization and increasing the overall throughput of the serverless system.

There are 2 specific goals:

- Resource reconfiguration: Using MIG to reconfigure the GPU, addressing external fragmentation. For example, if there are $\frac{2}{7}$ and $\frac{1}{7}$ GPUs remaining but the incoming request requires $\frac{3}{7}$ of a GPU, MIG can combine the two smaller GPUs into one larger $\frac{3}{7}$ GPU.

- Auto scaling: Automatically scaling based on the inference server's runtime behavior. If too many resources are allocated, the system will scale down to reduce costs. If resources are insufficient, the system will scale up to achieve better performance.

## Related Work

In previous research, there has been little exploration of the combination of MIG and serverless platforms. As for the individual topics of these two areas, most prior research has focused on the following aspects:

- MIG GPU reconfiguration: The reconfigurable hardware is a brand new topic, many studies are focusing on how to reconfigure the MIG GPU to achieve better overall performance.

- Serving DNN on MIG GPU: Since MIG GPU can be partitioned into many smaller GPUs, many work have explored which kind of slices is best suited for different DNN models

- Serverless system for Machine inference: Machine learning workloads are not yet fully supported by current Serverless platform. Thus some studies have explored approaches to performing machine learning inference on serverless platforms.

- GPU sharing on serverless platforms: Some previous works have implemented systems that enable sharing the entire GPU on serverless platforms. However, such systems may suffer from fault intolerance without the support of MIG.

Hence, we aim to develop a system that can fully utilize the capabilities of MIG and serverless platforms. This system will scale automatically according to runtime behavior and enhance overall resource utilization rate. Furthermore, it will benefit from the fault tolerance support provided by MIG GPUs.

## Methodology

- Evaluation setup: Conduct real-system evaluations on a small scale. For large scale, performing an extensive simulation-based will be used to test the system is not limited to small scale.
- Workloads: We generate LLM requests in a bimodal distribution, meaning there are bursts of requests to test the system's scaling policies. The bursts will occur at regular intervals, allowing the auto scaler to scale in advance.
- Metrics: we use four widely used figure of merit: average job completion time, makespan, system throughput and resource utilization rate.
  - Average job completion time: The end-to-end service time of a job, including both the time spent waiting in the queue and the job execution time.
  - Makespan: The time from the start of the first job to the completion of the last job in a series of job requests.
  - System throughput: This metric measures how much faster jobs are progressing toward completion compared to a system without auto-scaling, a system without Nvidia MIG, and a system without both.
  - Resource utilization rate: This metric measures the time during which there are idle GPU fragments and queuing requests, allowing us to assess the improvement in resource utilization achieved through resource reconfiguration.

## Expected Results

The expected outcome of this project is to create a Scalable Serving System for Large Language Models with Kubernetes and Nvidia MIG. This system will address resource fragmentation issue through MIG while also providing fault tolerance. Furthermore, it will be able to scale automatically at runtime to reduce costs and achieve better resource efficiency.