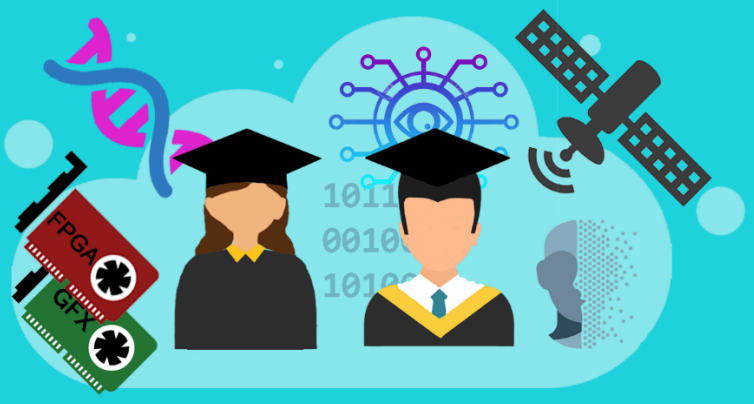
Diploma Thesis

Microproccessors and Digital Systems Laboratory



Serverless Computing Simulator

Serverless computing is an emerging cloud model where developers just write small functions and upload them, and the cloud provider takes care of “everything else”: creating containers or VMs, placing them on machines, scaling them up and down, and charging per request.

From the Service Provider’s, e.g., Amazon Web Services, Microsoft Azure, Google Cloud, perspective, in a real serverless platform there are many moving parts behind the scenes: a **scheduler** decides where to run each function instance, a **load balancer** routes incoming requests, an **autoscaler** decides how many instances to keep alive, and other components handle logging, monitoring, etc. All this complexity is hidden from the programmer, but it makes the platform itself quite complicated to understand and change.

Because of that complexity, it is hard for a student or researcher to experiment directly on a real serverless platform. Setting up open-source frameworks, i.e., Kubernetes, Knative, finding enough machines, and configuring monitoring already takes time, money, and operational skills. Going one step further and trying to change internal policies, like writing a new placement algorithm or autoscaling rule, usually means learning a large codebase, understanding many configuration options, and risking breaking the platform. This is why simulators [1,2] are so useful: they let us try out ideas offline, quickly and safely, without needing to deploy a whole cloud.

In recent years, cloud providers and researchers have released **many real-world traces** of serverless workloads, along with a wide range of ideas for **placement, dispatching, and autoscaling** policies. This creates a great opportunity to study how different strategies behave under realistic conditions.

In this thesis, the student will implement and use a **serverless simulator** to replay such traces and compare different methodologies for serverless deployments from the **provider’s point of view**. For example, how policies affect resource usage, cost, and quality of service.

References:

[1] Cao, Han, et al. "ServlessSimPro: A comprehensive serverless simulation platform." Future Generation Computer Systems 163 (2025): 107558.

[2] Lannurien, Vincent, et al. "HeROsim: An Allocation and Scheduling Simulator for Evaluating Serverless Orchestration Policies." IEEE Internet Computing (2024).

Contact:

Achilleas Tzenetopoulos, Ph.D. candidate NTUA: (atzenetopoulos@microlab.ntua.gr)
Dimosthenis Masouros, Post Doc NTUA
Sotirios Xydis, Ass. Prof., Microlab NTUA
Dimitrios Soudris, Professor Microlab NTUA