CSE 5350 Project Proposal

Background & Motivation:

Serverless computing has revolutionized how applications are deployed, enabling developers to focus on code rather than server management. This paradigm shift, while efficient, presents challenges in function scheduling to maintain Service Level Objectives (SLOs) due to the unpredictable nature of serverless workloads. Traditional OS schedulers like the Completely Fair Scheduler (CFS) are ill-suited for the ephemeral tasks in serverless functions. Our project introduces a novel scheduling framework to bridge the gap between the fleeting demands of serverless functions and the stringent SLOs.

Design:

We propose a two-tiered scheduling approach. The initial stage employs a FIFO strategy within a "filter-pool", where functions execute for a predetermined time slice S. Functions completing within this time are concluded and removed from the queue. If a function exceeds S in the filter pool, it's preempted and moved to the CFS pool. We incorporate SLO data from user-land to prioritize functions close to violating their SLOs by moving them back to the filter pool to ensure adherence to SLO requirements and optimize performance for shorter functions.

Objectives:

Our SLO-aware project will assess if the Service Level Indicator (SLI) meets the SLO consistently, impacting potential customer compensations per the Service Level Agreement (SLA). Runtime effectiveness (RTE) will gauge the system's efficiency with a majority of short jobs, defined by the ratio of service time to turnaround time. We'll use the Cumulative Distribution Function (CDF) against RTE and execution duration for quantitative system performance analysis. Additionally, queueing delay analysis will assess overload handling, while context switch ratio will measure scheduling overhead.