Resource Allocation with Voronoi Diagrams in Cloud Lambda Functions

Executing lambda functions in cloud computing has gained popularity recently. As opposed to traditional computing, lambda functions are executed only on a request, whereas the request might come from a user (browser, mobile device) or another cloud service. Traditional client-server systems require a server application to be running all the time to respond the requests. With lambda functions, users do only pay for the fraction of the time their code is executed.

As lambda functions are executed per request, they need a preprocessing and execution time on the cloud providers virtual or physical servers. As lambda functions are stateless, in a geographically distributed cloud provider, the request can be scheduled to any of the available servers. To minimize the response time, the request should be scheduled to the closest server.

To find the closest server for lambda function execution, servers can be placed on a 2D space. Since the servers have static places, the voronoi diagram would be generated at initialization. After that, using the location information from user requests, a request can be mapped to the closest server. Since these lambda requests arrive in a high rate and take very short time to execute, we need the decision to select a server to be completed quickly. In my work, I will first try scheduling independent lambda functions using the voronoi diagram of the servers. If it leads to promising outcomes, I would also try higher order voronoi diagrams for dependent lambda function execution.