Abstract der fertigen Arbeit

Edge Intelligence is a paradigm that promises to bring highly-responsive AI applications to the end users. Edge computing is the main enabler paradigm for this scenario, where computational resources are pushed from the cloud to the edge of the network. These heterogeneous clusters consist of devices that offer different capabilities, which range from general purpose CPUs, to GPUs, to application specific hardware accelerators. Cloud systems are comparatively homogeneous in terms of computing infrastructure. This heterogeneity poses a problem for users, which have to decide the hardware for their applications. To mitigate this issue, serverless computing may be a solution to this problem. Serverless computing helps abstract the underlying infrastructure from users away —allowing users to simply upload functions and offers the convenience of automatic scaling and pay-per-use cost model. Research proposes the idea of merging both paradigms, resulting in serverless edge computing. The problem is, that current serverless computing platforms use schedulers that were developed for homogeneous clusters. Therefore, heterogeneous clusters pose a challenge for serverless container schedulers to find optimal placements for containers.

In this thesis, we propose a solution that matches application requirements with appropriate node capabilities made tractable with the help of machine-learning based workload characterization. Our approach builds on existing serverless platforms such as Kubernetes and OpenFaaS. OpenFaaS is a Function-as-a-Service platform that uses Kubernetes as its deployment platform. The Kubernetes scheduler uses simple heuristics to schedule containers to cluster nodes. We extend the scheduler to make it workload-aware by adding three scheduling constraints that focus on: (1) performance, (2) preventing resource contention and (3) matching applications with appropriate nodes. We enable these constraints by (1) extensive profiling, (2) subsequent workload characterization and (3) solving the problem of matching applications with appropriate nodes.

We evaluate our approach by running simulations with three different scenarios and focus on AI-based applications. The results show that in edge computing scenarios the Function Execution Time (a key performance indicator) can be reduced by 33% to 68%. Moreover, performance degradation, caused by resource contention, can be reduced by 45% to 57%.