Thesis Title: Methods and Apparatus for Container Resource Management in Fog-Cloud Orchestration

Name: Shubha Brata Nath

Roll Number: 16CS91R01

Thesis Abstract: In the age of the Internet of Things (IoT) devices, fog-cloud computing has emerged as an essential computing paradigm for faster user response as well as for managing a large number of clients. Fog-cloud orchestration is used for efficiently managing the user services for meeting its quality of service. From the conceptualization of Cloud Computing and Fog Computing, several researchers have proposed different types of architectures to facilitate their implementations in the various application context. Nevertheless, there are many open rooms still available and pertinent for improving fog-cloud systems to boost their performance to support many users with better resource management. This thesis analyzes the fog-cloud orchestrations with optimizations for better system performance.

Fog-cloud system constitutes of different fog and cloud servers with dynamic resource availability. In the presence of various services with different requirements, the fog-cloud system should have improved designs. The services are deployed in the fog-cloud system as a light-weight virtualized instance called containers. However, the deployment of containers may not be always better in terms of response time, energy consumption as well as service consolidation. This raises the need to analyze the fog-cloud system for better deployment of the services. It motivates one to dig deeper to understand how the existing fog-cloud systems work and the specific measures to be adopted for its improvement.

Fog computing is one of the emerging technology and it enables the IoT micro-services. The IoT environment is comprised of numerous devices which execute IoT micro-services. Hence, a deep dive into how the micro-services can be placed is expected to yield answers to the improvement of response time. As the literature lacks an in-depth analysis of the fog based micro-service, in this work, we have proposed Pick-Test-Choose (PTC), a dynamic micro-service deployment and execution model that considers such time-varying primary workloads and workload spikes in the fog nodes. We have performed thorough experiments to understand the same. We use our in-house testbed to deploy the micro-services with the aim to minimize response time and have subsequently analyzed them. Our analysis has found that the dynamic resource variation of the fog devices can be monitored and efficient deployment of the micro-services can happen. We have also observed that the deployment algorithm needs to take care of the migration of the micro-services. A further scalability study with an emulated setup over Amazon Elastic Compute Cloud (Amazon EC2) further confirms the superiority of PTC over other baselines.

Containers are a lightweight virtualization platform to support large-scale application deployments over cloud-based systems. Due to its low-overhead deployment capacity, containers are predominantly used to execute services in a consolidated edge-cloud system. However, unplanned deployment of containers across edge-cloud systems can result in significant resource wastage, particularly in terms of energy footprint. Therefore, container deployments over an edge-cloud orchestration platform should carefully model the deployed

services' energy footprint while ensuring their quality of experience. So, in the second work, we first show that proper consolidation of containers over an integrated edge-cloud orchestration framework can significantly reduce the energy footprint. We propose an algorithm named Energy-Aware Service consolidation using baYesian optimization (EASY). It is based on the Bayesian Optimization framework to decide container placements while reducing the execution platform's energy footprint. The proposed EASY algorithm selects the edge and the cloud servers to distribute the containers while considering both the application's response time requirements and the energy footprint due to container execution. The experiments included observations with delay-sensitive and non-delay-sensitive services in fog-cloud computing. The study has shown that the proposed approach i.e. EASY can minimize the energy consumption in fog-cloud orchestration.

In the thesis's final contribution, we have developed a system named Container State Management for Deployment (CSMD) that increases consolidation ratio i.e. the number of services per host in the cloud. We found that the containers remain in an idle state when no user request is coming. Again, it is important to increase the consolidation ratio for ensuring that less number of servers are used in a cloud data center. However, existing systems focus on service deployment. CSMD uses an algorithm to checkpoint the idle containers so that the resources held by it can be released. The new containers are deployed using the released resources in a server. CSMD also has an algorithm for checking the container status. The containers are resumed from the checkpointed state when user requests come for it. We performed the evaluation of the CSMD system in Amazon EC2 by performing efficient state management of the containers. The experiments in Amazon EC2 show that the proposed CSMD system is superior to the existing algorithms for ensuring the increase in the consolidation ratio of data centers.

In summary, this thesis contributes to the literature from two different aspects. First, it analyzes the existing fog-cloud systems to check and explore their performance. Second, we propose two advancements over the current systems – (a) from the context of optimization of system response time and (b) for improving the energy consumption and increasing the consolidation ratio. Although this thesis opens up and explores new research in fog-cloud computing, there are still many unexplored areas, which can be taken up for extending the research and developments in this direction.

Keywords: Fog-Cloud Orchestration; Container Resource Management; Container Placement; Service Consolidation