

Live Containerized Service Migration across Edge to Cloud Continuum

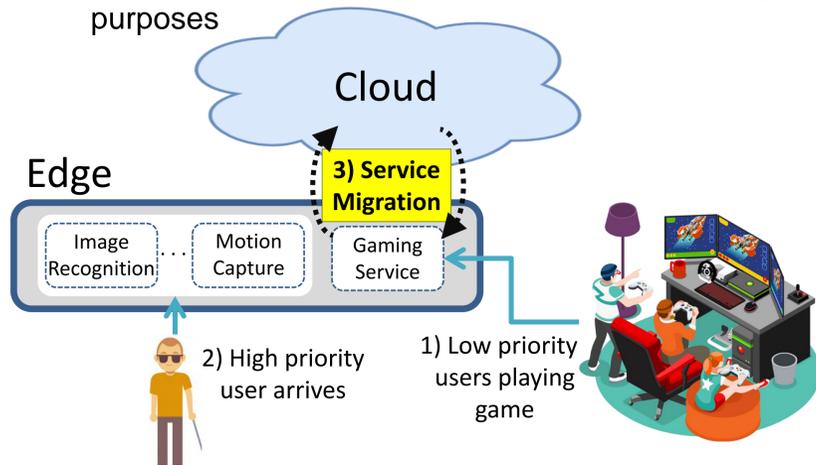
Thanawat Chanikaphon, Mohsen Amini Salehi

High Performance Cloud Computing (HPCC) Lab

School of Computing and Informatics, University of Louisiana at Lafayette

WHY LIVE SERVICE MIGRATION IS NEEDED?

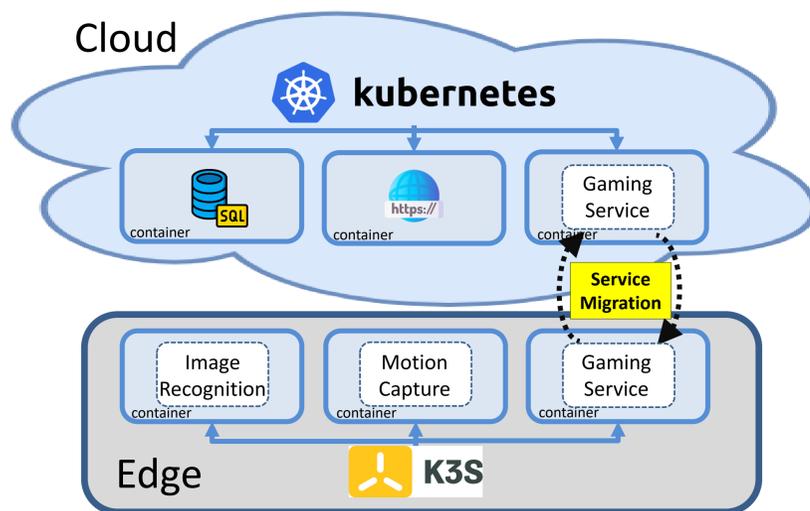
- To decrease cost, latency, and energy footprint
- For service mobility, elasticity, and load balancing purposes



- **Orchestrators do not support live container migration**

PROJECT GOALS

- Non-invasive support of live container migration
- Migrating across homogeneous and heterogeneous orchestrators



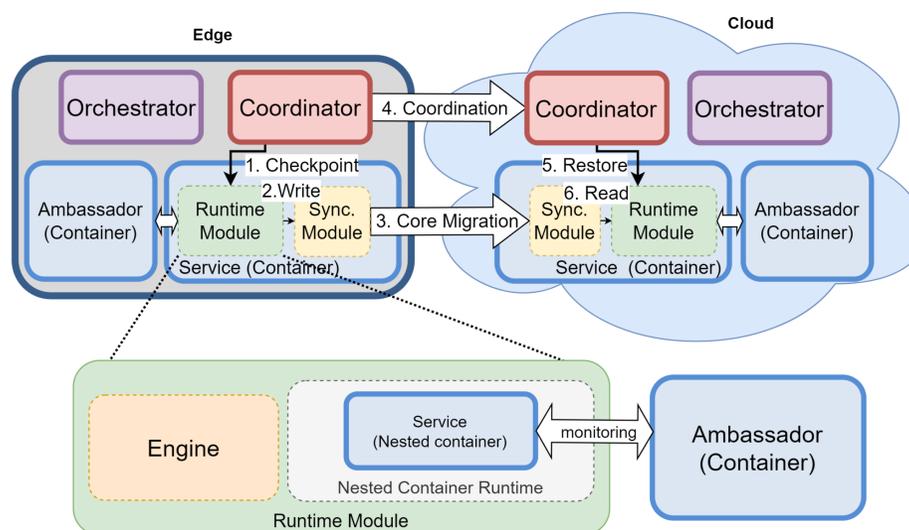
Bird-eye view of the live migration across the heterogeneous orchestrators

OBJECTIVES OF LIVE SERVICE MIGRATION

- Low time overhead
 - During normal operation (not migration)
 - Upon migration
- Transparent from users and developers
 - Migratable services should look like the non-migratable ones
- No change to the underlying orchestrator
 - System admin reluctant to change platform
- Integrate into existing container platforms
 - K3s
 - *Rancher Kubernetes Engine (RKE)*
 - *OKD (Openshift, Minishift)*
 - *Marathon (Apache Mesos)*

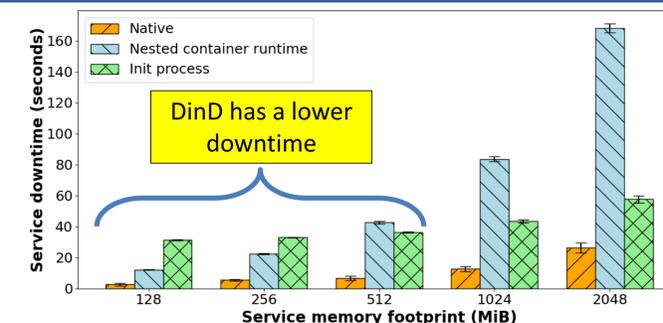
SOLUTION APPROACH

1. Leveraging nested containers (*Docker-in-Docker*)
2. Leveraging init process (*FastFreeze*)



Architectural overview of the live container migration system

EVALUATION RESULTS



Approach	Changed Required	Service downtime (seconds)				
		K8s	Mesos	K3s	Minishift	Minishift (diff. OS)
Native	Orchestrator	3.0	Infeasible	Infeasible	Infeasible	Infeasible
Docker-in-Docker	Nothing	12.2	12.1	12.1	12.8	Infeasible
FastFreeze	Application	31.3	31.3	31.4	31.3	32.6

Service downtime for migration from various orchestrators to Kubernetes

CONCLUSION AND FUTURE WORKS

- We provide non-invasive support for container migration across heterogeneous orchestrators using Docker-in-Docker
- For microservices under 512 MiB memory, Docker-in-Docker approach outperforms FastFreeze
- Future works
 - Decision making: what/when/where to migrate
 - Fault-safe migration

Source code:



Demo video:

