### 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

### 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)

### PROJECT GOALS

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

### SOLUTION APPROACH

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

### EVALUATION RESULTS

<table>
<thead>
<tr>
<th>Approach</th>
<th>Changed Required</th>
<th>Service downtime (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native Orchestrator</td>
<td>K3s</td>
<td>Mesos</td>
</tr>
<tr>
<td>Docker-in-Docker</td>
<td>Nothing</td>
<td>12.2</td>
</tr>
<tr>
<td>FastFreeze Application</td>
<td>31.3</td>
<td>31.3</td>
</tr>
</tbody>
</table>

DinD has a lower downtime

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: [link]

Demo video: [link]