LFN Networking

LFN Developer & Testing Forum
EMCO: Edge Relocation using Temporal workflows

Grzegorz Panek, Piotr Matysiak: Orange
Sundar Nadathur, Srinivasa Addepalli: Intel
Anti-Trust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrustpolicy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.
Agenda

- Introduction to EMCO and Temporal
- Demo Setup
- Demo: Relocate an application from cluster A to cluster B
- Current and future work
Edge Multi Cluster Orchestrator

Major Trends:
- Multi-access Edge Computing + Cloud
  - Geo-distributed apps, Clusters at scale
  - Kubernetes everywhere for Cloud-Native Apps

EMCO
- Linux Foundation's project-emco.io
- Intent-based Lifecycle management of apps and network functions
- Various edge locations, cloud/on-prem DCs
- Highly extensible with in-tree or 3rd party controllers

New!
- Workflows: complex sequences of tasks in Go, Java, etc.
- A new way to extend EMCO for complex use cases
- Collaboration of Intel and Orange
- Available in EMCO 22.03
Temporal Workflow Engine

Temporal is an open-source workflow engine offering resilience and scale;

**Temporal Cluster + Worker Processes = Temporal Platform**

We can use Temporal SDK to develop Temporal Clients, Workflow Definitions and Worker Programs.

Then we can deploy all components and use Temporal Clients to communicate with Temporal Platform to Start a Workflows, Get result of a Workflow, etc.

At the end Workflow Workers would serve invoked Workflows and serve until they are done.

Temporal does not preclude EMCO integration with other workflow engines in the future.

„Less plumbing, more coding.”
EMCO + Temporal

EMCO + Temporal

EMCO

Admin

Ingress Gateway

EMCO

Scripts/Tools

Temporal Server

Temporal Containers

Workflow uService

Other EMCO uServices

http/gRPC server

Workflow Intent APIs

http server

Workflow client(s)

Worker containers

Worker:
- workflow
- activities

Worker starts when container is started

Query EMCO if needed

Monitor/query/cancel workflows

curl -X POST http://container-ip:9090/invoke/my-workflowclient
Example Scenarios for Workflows

- Relocate an application from edge cluster A to edge cluster B, because:
  - UE has roamed and cluster B is now closer to the UE.
  - Cluster A needs to be brought down for maintenance.
  - Cluster A is approaching its capacity limit.

- Update the network policies (firewall rules, load balancer policies, etc.) in an external device.
- Spin up a new Kubernetes cluster (infrastructure orchestration).
Problem statement according to ETSI standards

The user (UE) is consuming a service, while moving out of the coverage area of Source MEC Host (Cluster A). Later he/she enters the coverage area of Target MEC Host (cluster B) and expects to resume the same service. This requires a relocation of a service instance from cluster A to cluster B.

Requirements for application relocation (identified by ER WG):

- Service continuity must be assured to the UE;
- The new instance of the application must be declared to be 'ready' before we can steer the traffic to the new app instance;
- If there are several candidates for the target MEC cluster, the final choice should be made by MEC Orchestrator
Demo Scenario

1. EMCO deploys app
2. App published in DNS
3. Client queries DNS
4. Client traffic to server
5. Client roams to tower 2
6. EMCO starts workflow to migrate app
7. App updated in DNS
8. Client queries DNS

Client traffic to server In cluster 2!

Automate in future
Demo Layout and Steps

**DEMO STEPS**
1. Define app in EMCO with intents
2. Deploy app from EMCO
3. Define workflow intent in EMCO
4. Bring up worker and workflow client
5. Start workflow: app migrates
LIVE DEMO
The goal is to extend existing PoC of ER inside Stand-alone 5G + MEC System

- Using open-source 5G Core (Free5GC) and 5G RAN (UERANSIM)
- MEC architecture based on EMCO (Management Level) + Kubernetes (Host Level)

This requires to design and implement:

- Entity which will make decisions to relocate application (MEC side) or/and entity which would subscribe to 5G CN notifications;
- Entity which will be aware of MEC topology, which would select the optimal Edge Server (MEC Host) to place the MEC application, including decision algorithm;
- Traffic Steering mechanism (based on ETSI standardization or not)
- (optional) Mechanism to support relocation of stateful applications
THANK YOU FOR YOUR ATTENTION!

Project EMCO: project-emco.io
Edge Relocation Working Group: shorturl.at/vFSYZ