



CODECO's Validation and Experimentation Challenges



Field	Details
1. Name of Challenge	CODECO deployment and scalability
2. Partners	ICOM (contact Marinela Mertiri at marmert@intracom-telecom.com), FOR (contact person is Rute C. Sofia at sofia@fortiss.org)
3. Submission Specifications	<p>Requirements: OSS Basic Toolkit (Single - Cluster)</p> <p>Description</p> <p>The main goal of this challenge is to test the deployments' times and scalability of CODECO, following the proposed steps. The specific goals are the next ones:</p> <ol style="list-style-type: none">1. Scale to different topologies (a single cluster with 10 nodes, 50 nodes and 100 nodes) and watch for possible issues2. For each topology, calculate the deployment and deletion times (to see how they're affected by the number of nodes) and compare them against plain Kubernetes (?)3. In the same scope, perform a simple workflow in each topology and calculate how much time it takes for CODECO to deploy the respective services. The goal is again to see if and how the number of nodes increases the time required for the decision making in CODECO, and, thus, the overall time for deploying a workflow, which should be the same in all cases.4. Analyze which component contributes the most to latency, in all

the above tests. A way to do that is to have in each component some logs with timestamps, showing when a request is received and when it's served.

Main steps

M1 – Setup CODECO in a single cluster

1. Prepare 3 simulated, testing environments (using KIND or kubeadm). These environments correspond to a small cluster (10 nodes), a medium one (50 nodes) and a large one (100 nodes)
2. Train the GNN models for each environment (other ML models need pre-training?)
3. If not already done, update the key components (i.e. ACM, PDLC, MDM, NetMA and SWM) with timestamp logging in events like request received, processing start, decision taken and response sent.
4. Deploy full CODECO on each setup. This deployment is for testing purposes only (to see if it's configured and deployed correctly) and no measurements will be taken at this step.
5. Ensure component-level logging is enabled (e.g., with timestamps at request/response points)
6. Set up basic observability (e.g. Prometheus for metrics, central point for viewing the logs)

M1 - Deployment /Deletion Time

Measurements

1. For each setup, measure time taken for:
 - Full OSS Basic Toolkit deployment (from starting the ACM deployment to having all components "Running")
 - Toolkit teardown (from deletion to cleanup)
2. Repeat the tests 3–5 times per topology to get average and variance
3. **Baseline comparison:**
 - Perform the same deploy/delete cycle using

	<p>only Kubernetes-native equivalents</p> <ul style="list-style-type: none"> • Compare with OSS Basic Toolkit's performance to assess overhead <p>M2 – Workflow Deployment Test</p> <ol style="list-style-type: none"> 1. Define a simple multi-service application (e.g., 3–5 pods with interdependencies) 2. For each cluster: <ul style="list-style-type: none"> • Deploy the workflow using OSS Basic Toolkit (via ACM) • Measure time from submission to pod scheduling, time from scheduling to all pods "Running" and Total end-to-end workflow deployment time. The logs and timestamps will assist in these measurements 3. Repeat 5 - 10 times (for each setup) for reliability <p>M2 – Analyze Results</p> <ol style="list-style-type: none"> 1. Collect all results from above experiments and analyze them in order to <ul style="list-style-type: none"> • Compare measurements for different topologies • Identify delays per component and possible bottlenecks <p>M3 – Summarize Findings and Propose improvements</p> <ol style="list-style-type: none"> 1. Prepare charts and plots (e.g. deployment time vs. cluster size, workflow execution time vs. cluster size, latency per component) 2. Document observed performance patterns and unexpected slowdowns or errors 3. Propose improvements (e.g., parallelized scheduling, optimizing message passing, use faster but sub-optimal solutions for decision making in large clusters)
<p>4. Platforms to be used</p>	<p>KinD, latest version of OSS basic Toolkit</p> <p>Applicants are expected to follow up with the listed contact persons in order to obtain</p>

	the necessary material and then upload their submission via a Zip File.
--	-------------------------------------------------------------------------