EDGELESS

Cognitive edge-cloud with serverless computing

Benchmarking an EDGELESS Cluster for Serverless Edge Computing Applications

Claudio Cicconetti

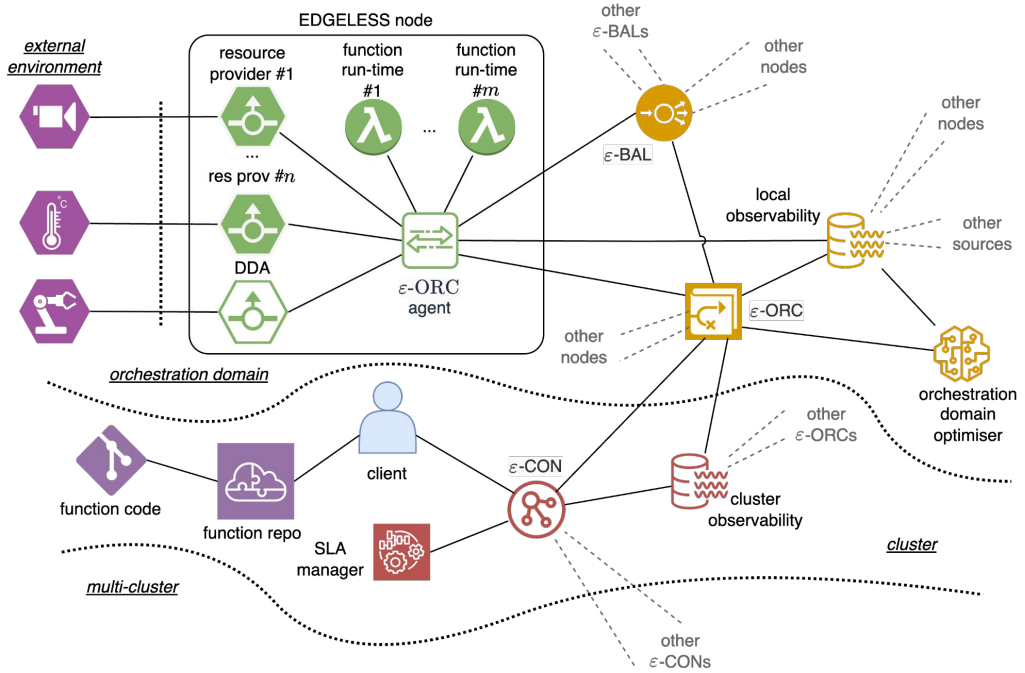
National Research Council of Italy

ISTITUTO DI INFORMATICA E TELEMATICA

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

EDGELESS as a complex system

- a lot of workflows, running in
- multiple domains, each hosting
- many nodes, each running
- several function/resource instances



This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

Problem

- How to evaluate the performance of an EDGELESS system in **controlled** and **repeatable** conditions?

Why?

Examples:

- Study the impact of a new algorithm (e.g., orchestration function)
- Provision the system
- Assess what-if scenarios

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

Solution

`edgeless_benchmark`

- Shipped with core EDGELESS software
- Interacts as a client with the real system
  - Benchmark flows can be mixed with real flows
- Creates workflows in a repeatable manner

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

Workload: arrival models

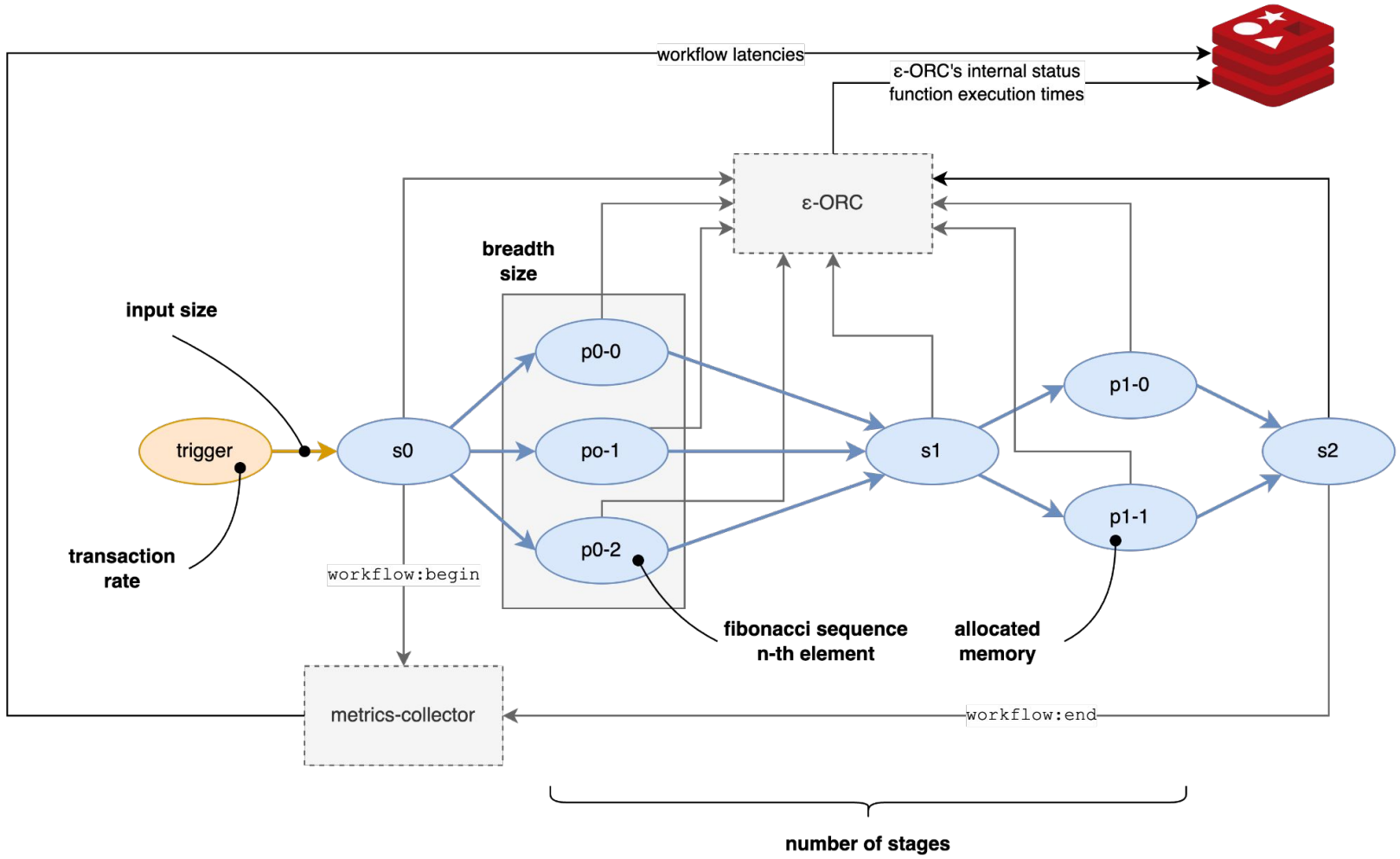
Arrival model	Description
poisson	Inter-arrival between consecutive workflows and lifetimes are exponentially distributed.
incremental	One new workflow arrive every new inter-arrival time, with constant lifetime.
incr-and-keep	Add workflows, with constant lifetimes, incrementally until the warm up period finishes, then keep until the end of the experiment.
single	Add a single workflow that lasts for the entire experiment.
trace	Read the arrival and end times of workflows from a file specified with `--workload-trace`, one workflow per line in the format `arrival,end_time`

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

Workload: workflow types

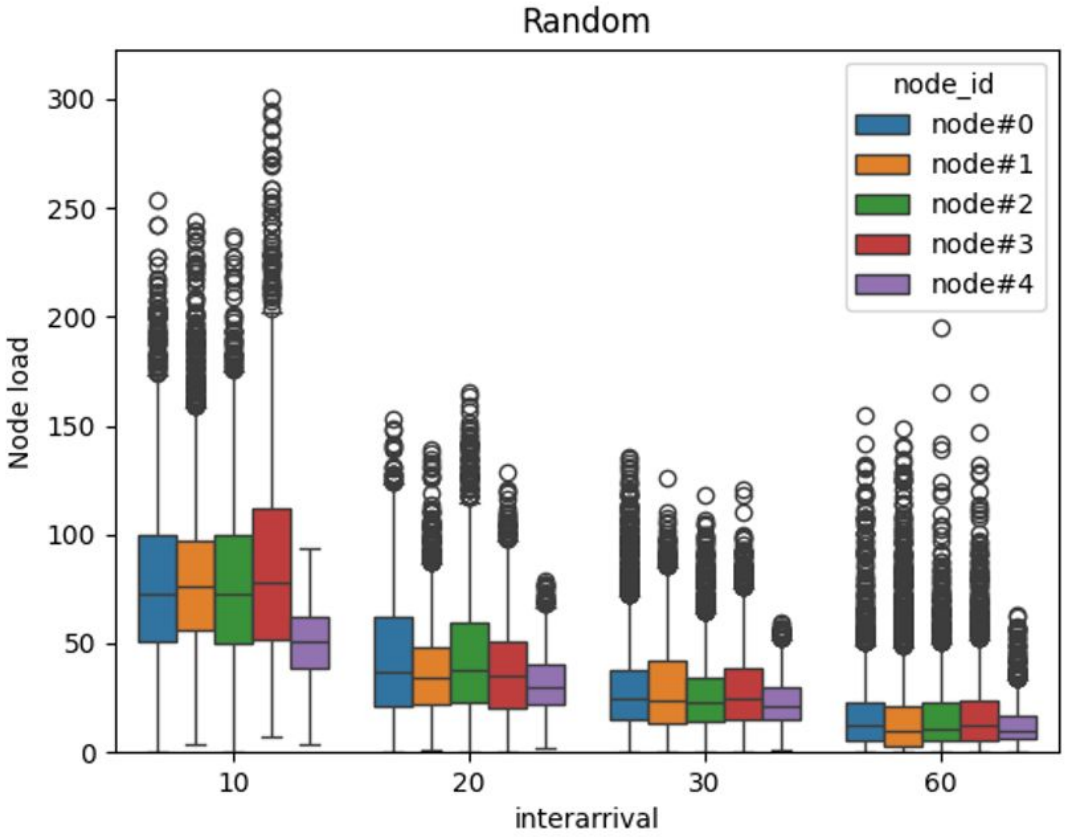
Workflow type	Description	Application metrics	Template
none	No workflow is created. This option is meant only for testing/troubleshooting.	None	N
single	A single function.	--	N
matrix-mul-chain	A chain of functions, each performing the multiplication of two matrices of 32-bit floating point random numbers at each invocation.	workflow,function	Y
vector-mul-chain	A chain of functions, each performing the multiplication of an internal random matrix of 32-bit floating point numbers by the input vector received from the caller.	workflow,function	Y
map-reduce	A workflow consisting of a random number of stages, where each stage is composed of a random number of processing blocks. Before going to the next stage, the output from all the processing blocks in the stage before must be received.	workflow	Y
json-spec	The workflow specified in the given JSON file. The string `@WF_ID` in the file is substituted with a sequential identifier of the workflow.	--	N

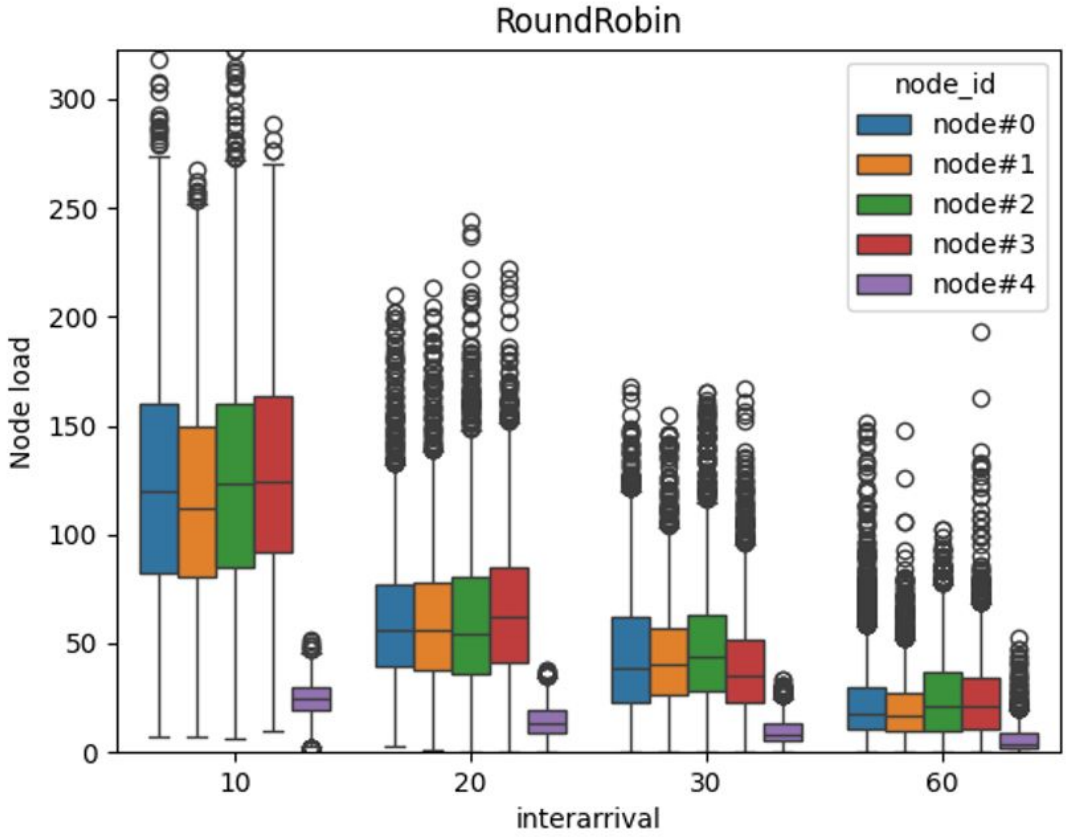
Workflow example: map-reduce



This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

Example of results (20 experiments, 1 hour each)





This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.

Thank you for your attention

Talk is cheap, here's the code:

`https://github.com/edgeless-project/edgeless`

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement No 101092950.