TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Macroarea di Ingegneria

A QoS-Aware Broker for Multi-Provider Serverless Applications

Andrea Graziani
m. 0273395

Supervisor:	Valeria Cardellini
Tutor:	Gabriele Russo Russo

May 23, 2022

Serverless Computing: Overview

My thesis is focused on "**serverless computing**".

It is a development paradigm according to which:

- The provider takes care of all aspect of server management.
- Cloud application are abstracted as a group of so-called **serverless functions**, which are computation units implementing a business functionality.

A serverless function is executed inside a containerized environment: the so-called **function instance**.

The FaaS platform **automatically** scales the number of function instances.

FaaS platforms impose a **limit** on the number of function instance runnable at the **same time** called **concurrency limit**.

A delay is observed when a new function instance is started by the provider: this event is called **cold start**.

Serverless Computing: Overview

To invoke a **serverless function**, users have to specify a so-called **serverless function configuration**.

- Generally, the **amount of memory allocated** to a serverless function.

Configuration parameters **significantly** affect the **cost** and **response time** of serverless functions.

Serverless Computing: Problems

- Lack of support for application whose functions are hosted on **multiple providers**.
- Lack of support for serverless function implementations abstraction, that is, for the so-called **concrete functions**.
- Fulfillment of **non-functional requirements** concerning the **quality of service** (QoS) levels that should be guaranteed for **multi-provider serverless applications**.
- Fulfillment of **functional requirements** concerning the **orchestration** of multi-provider applications.

Thesis Goals

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Goal # 1

To guarantee the **satisfaction of QoS levels** for multi-provider serverless applications.

It was necessary to develop:

1. An **analytical model** to evaluate aforementioned class of applications.
2. A **methodological way** to find the "best" configuration to satisfy QoS constraints.

Goal # 2

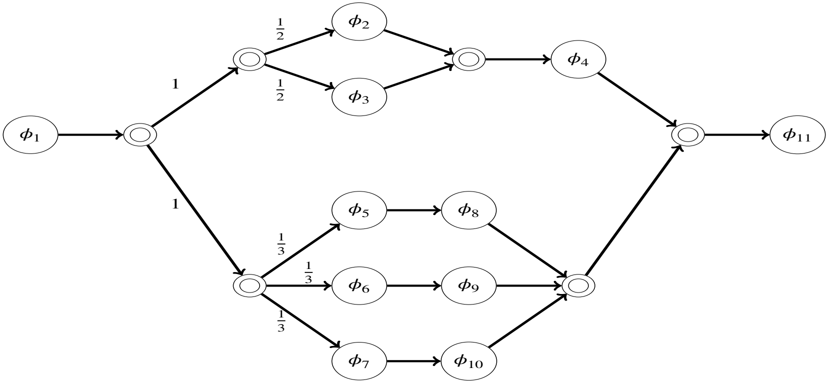
The **orchestration** for multi-provider serverless applications.

To achieve it, I had to build:

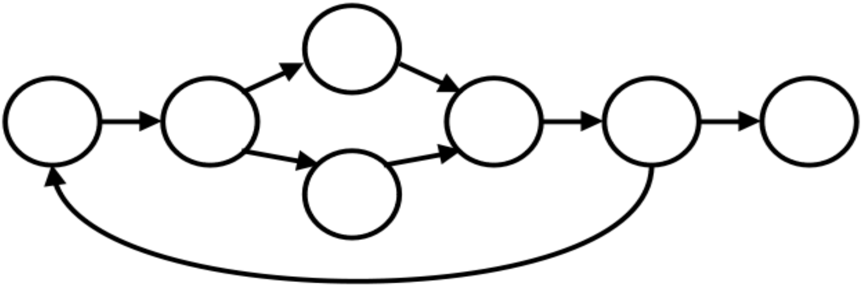
1. A **software framework**.
2. An extension to an already existent **representation scheme** to define a serverless application workflow.
   - Based on an existing language called **abstract function choreography language** (AFCL).

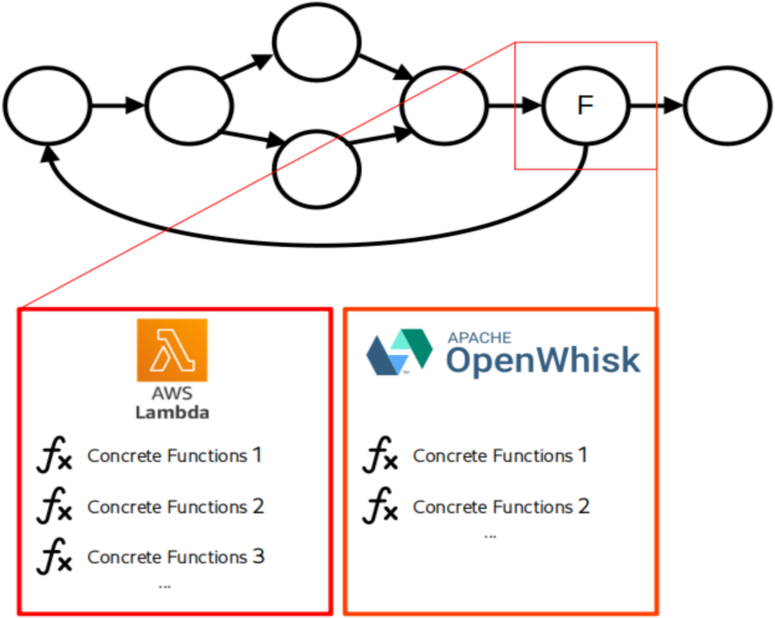
System Model

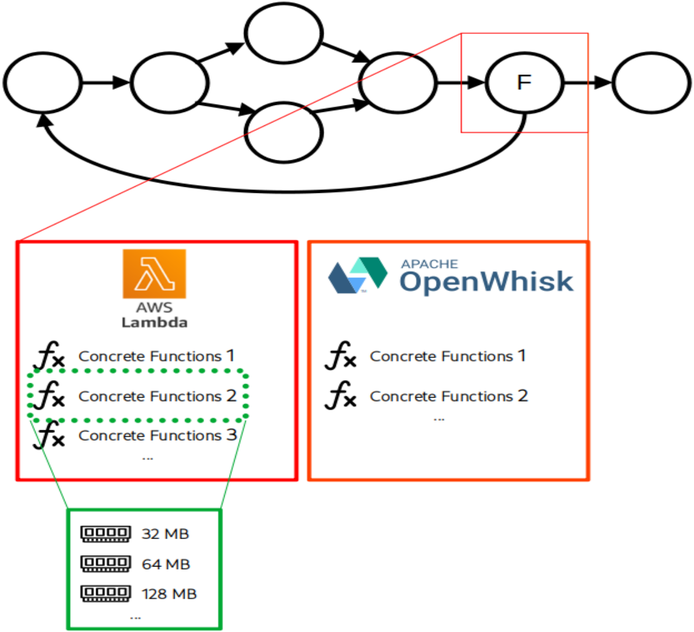
Serverless applications are abstracted to a **weakly connected weighted directed graph**.

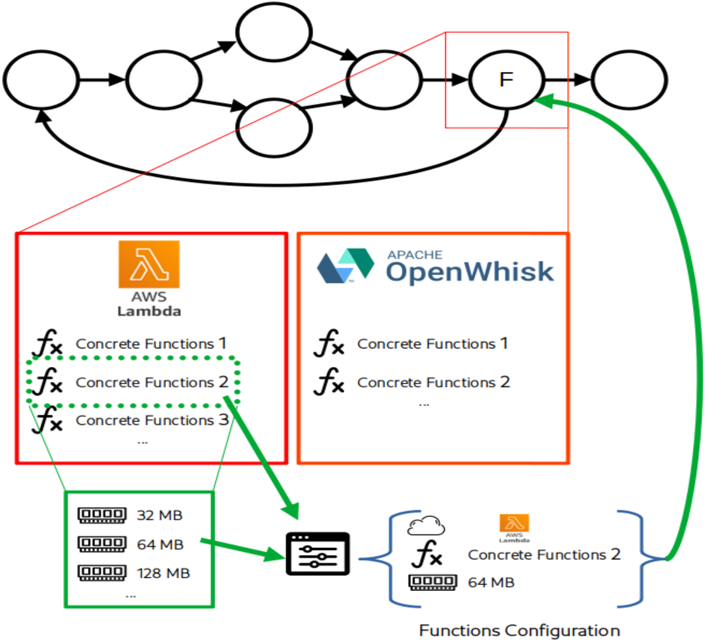


Application Configuration




F
AWS
Lambda
Concrete Functions 1
Concrete Functions 2
Concrete Functions 3
...
APACHE
OpenWhisk
Concrete Functions 1
Concrete Functions 2
...


F
AWS
Lambda
Concrete Functions 1
Concrete Functions 2
Concrete Functions 3
...
APACHE
OpenWhisk
Concrete Functions 1
Concrete Functions 2
...
32 MB
64 MB
128 MB
...


F
AWS
Lambda
Concrete Functions 1
Concrete Functions 2
Concrete Functions 3
...
APACHE
OpenWhisk
Concrete Functions 1
Concrete Functions 2
...
32 MB
64 MB
128 MB
...
AWS
Lambda
Concrete Functions 2
64 MB
Functions Configuration

System Model

- Estimations of **average response time** and **cost** under **any** possible configuration of all concrete functions is done using **exponential moving average** approach.
- An estimation of the **probability** according to which a request follows a cold start is required.
  - This is done using **Erlang-B** formula by modeling FaaS platform providers by **sets** of $M/G/K(t)/K(t)$ queueing systems.
    - $K(t)$: the number of function instances at time t.
- Performance estimations of the application depend on its workflow properties.

Optimization Problem

- To achieve our goal consisting in finding the best configuration to guarantee QoS constraints, we have to solve an **optimization problem**.
    - It is based on **multi-dimensional multi-choice knapsack problem formulation** (MMKP).

$$\max \quad \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot p_{\phi_{i_j}}(t) \tag{5.10}$$

$$\text{subject to} \quad \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot c_{\phi_{i_j}}(t) \leq C \tag{5.11}$$

$$\sum_{\phi_i \in \mathcal{F}_{\mathcal{E}}(\mathcal{C}) \setminus \mathcal{F}_{\mathcal{E}}(\widetilde{\mathcal{P}})} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot rt_{\phi_{i_j}}(t) + + \sum_{\phi_h \in \delta_{\mathcal{C}}} \sum_{j=1}^{|\mathbf{F}_{\phi_h} \times \mathbb{N}|} y_{\phi_{h_j}} \cdot rt_{\phi_{h_j}}(t) \leq RT \qquad \forall \delta_{\mathcal{C}} \in \Delta_{\mathcal{C}} \tag{5.12}$$

$$\sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot a_{(\phi_{i_j}, \omega_P^{(l)})} \leq l - R(\mathbf{Q}_{\omega_P^{(l)}}, t) \qquad \forall \omega_P^{(l)} \in \widetilde{\mathbf{S}}_{\mathcal{C}} \tag{5.13}$$

$$\sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} = 1 \qquad \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \tag{5.14}$$

$$y_{\phi_{i_j}} \in \{0, 1\} \qquad \begin{array}{r} \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \\ \forall j \in \mathbb{N} \cap [1, |\mathbf{F}_{\phi_i} \times \mathbb{N}|] \end{array}$$

$$max \quad \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot p_{\phi_{i_j}}(t) \tag{5.10}$$

$$\boxed{\text{subject to} \quad \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot c_{\phi_{i_j}}(t) \leq C} \tag{5.11}$$

Cost Constraint

$$\sum_{\phi_i \in \mathcal{F}_{\mathcal{E}}(\mathcal{C}) \setminus \mathcal{F}_{\mathcal{E}}(\widetilde{\mathcal{P}})} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot rt_{\phi_{i_j}}(t) +$$

$$+ \sum_{\phi_h \in \delta_{\mathcal{C}}} \sum_{j=1}^{|\mathbf{F}_{\phi_h} \times \mathbb{N}|} y_{\phi_{h_j}} \cdot rt_{\phi_{h_j}}(t) \leq RT \qquad \forall \delta_{\mathcal{C}} \in \Delta_{\mathcal{C}} \tag{5.12}$$

$$\sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot a_{(\phi_{i_j}, \omega_P^{(l)})} \leq l - R(\mathbf{Q}_{\omega_P^{(l)}}, t) \qquad \forall \omega_P^{(l)} \in \widetilde{\mathbf{S}}_{\mathcal{C}} \tag{5.13}$$

$$\sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} = 1 \qquad \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \tag{5.14}$$

$$y_{\phi_{i_j}} \in \{0, 1\} \qquad \begin{array}{r} \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \\ \forall j \in \mathbb{N} \cap [1, |\mathbf{F}_{\phi_i} \times \mathbb{N}|] \end{array}$$

$$max \quad \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot p_{\phi_{i_j}}(t) \tag{5.10}$$

$$\text{subject to} \quad \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot c_{\phi_{i_j}}(t) \leq C \tag{5.11}$$

Response Time Constraint

$$\boxed{\begin{aligned} &\sum_{\phi_i \in \mathcal{F}_{\mathcal{E}}(\mathcal{C}) \setminus \mathcal{F}_{\mathcal{E}}(\widetilde{\mathcal{P}})} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot rt_{\phi_{i_j}}(t) + \\ &+ \sum_{\phi_h \in \delta_{\mathcal{C}}} \sum_{j=1}^{|\mathbf{F}_{\phi_h} \times \mathbb{N}|} y_{\phi_{h_j}} \cdot rt_{\phi_{h_j}}(t) \leq RT \qquad \forall \delta_{\mathcal{C}} \in \Delta_{\mathcal{C}} \end{aligned}} \tag{5.12}$$

$$\sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot a_{(\phi_{i_j}, \omega_P^{(l)})} \leq l - R(\mathbf{Q}_{\omega_P^{(l)}}, t) \qquad \forall \omega_P^{(l)} \in \widetilde{\mathbf{S}}_{\mathcal{C}} \tag{5.13}$$

$$\sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} = 1 \qquad \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \tag{5.14}$$

$$y_{\phi_{i_j}} \in \{0, 1\} \qquad \begin{aligned} &\forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \\ &\forall j \in \mathbb{N} \cap [1, |\mathbf{F}_{\phi_i} \times \mathbb{N}|] \end{aligned}$$

$$\max \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot p_{\phi_{i_j}}(t) \tag{5.10}$$

$$\text{subject to} \sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot c_{\phi_{i_j}}(t) \leq C \tag{5.11}$$

$$\sum_{\phi_i \in \mathcal{F}_{\mathcal{E}}(\mathcal{C}) \setminus \mathcal{F}_{\mathcal{E}}(\widetilde{\mathcal{P}})} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot rt_{\phi_{i_j}}(t) +$$

$$+ \sum_{\phi_h \in \delta_{\mathcal{C}}} \sum_{j=1}^{|\mathbf{F}_{\phi_h} \times \mathbb{N}|} y_{\phi_{h_j}} \cdot rt_{\phi_{h_j}}(t) \leq RT \qquad \forall \delta_{\mathcal{C}} \in \Delta_{\mathcal{C}} \tag{5.12}$$

$$\boxed{\sum_{i=1}^{|\mathcal{F}_{\mathcal{E}}(\mathcal{C})|} \sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} \cdot a_{(\phi_{i_j}, \omega_P^{(l)})} \leq l - R(\mathbf{Q}_{\omega_P^{(l)}}, t) \qquad \forall \omega_P^{(l)} \in \widetilde{\mathbf{S}}_{\mathcal{C}}} \tag{5.13}$$

Capacity Constraint

$$\sum_{j=1}^{|\mathbf{F}_{\phi_i} \times \mathbb{N}|} y_{\phi_{i_j}} = 1 \qquad \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \tag{5.14}$$

$$y_{\phi_{i_j}} \in \{0, 1\} \qquad \begin{array}{r} \forall i \in \mathbb{N} \cap [1, |\mathcal{F}_{\mathcal{E}}|] \\ \forall j \in \mathbb{N} \cap [1, |\mathbf{F}_{\phi_i} \times \mathbb{N}|] \end{array}$$

Heuristic Algorithm

I develop a custom heuristic algorithm based on **ant colony optimization** (ACO).

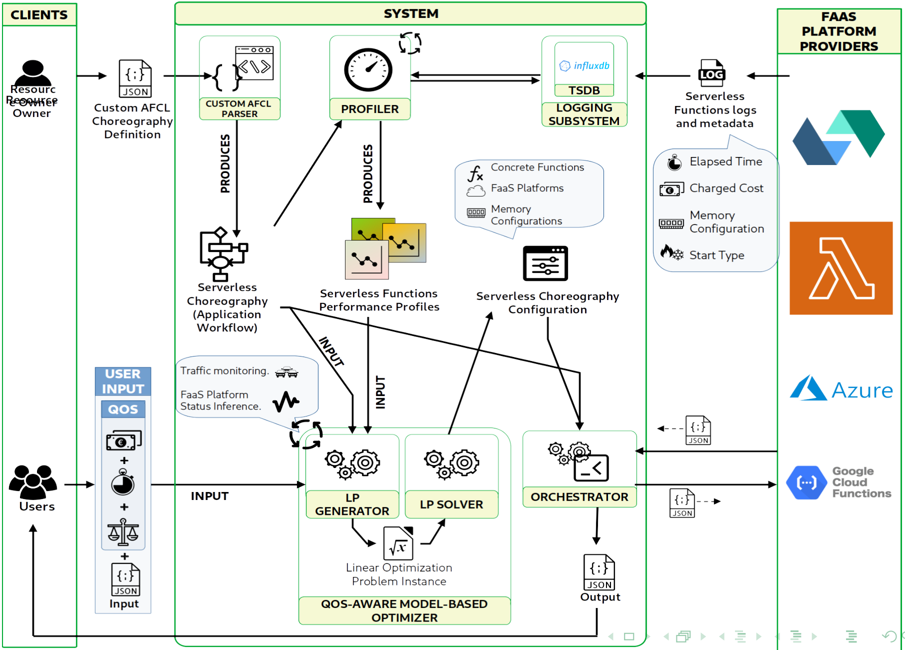
- It is based on a set of computational agents, called **artificial ants**, which **iteratively** construct a solution.
- At each iteration, each agent moves from a solution to another, applying a series of stochastic **local** decisions whose policy is based on following parameters:
  - **Attractiveness**.
  - **Pheromone trail**.

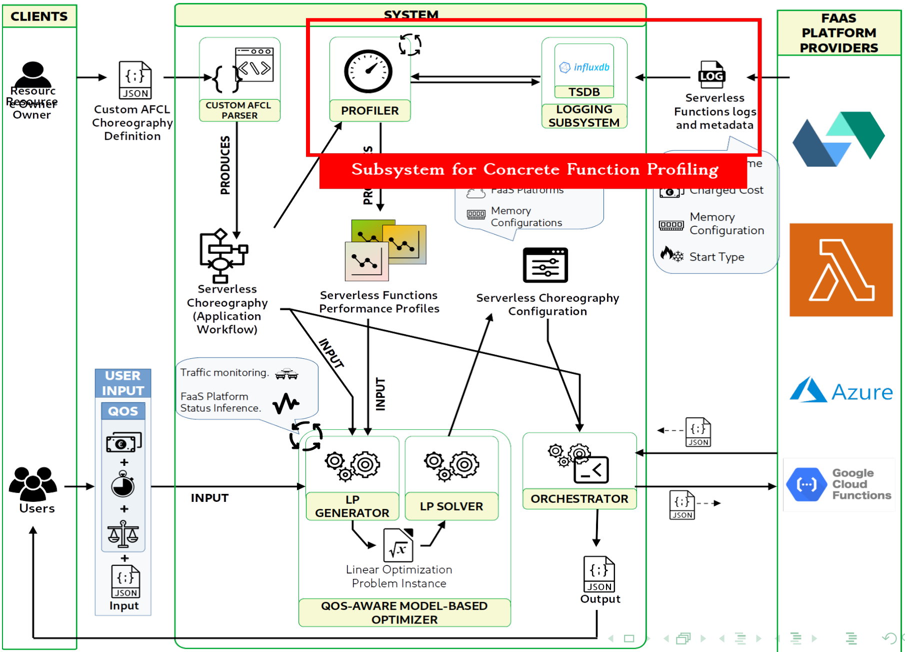
Pheromone trails are updated during **every iteration**.

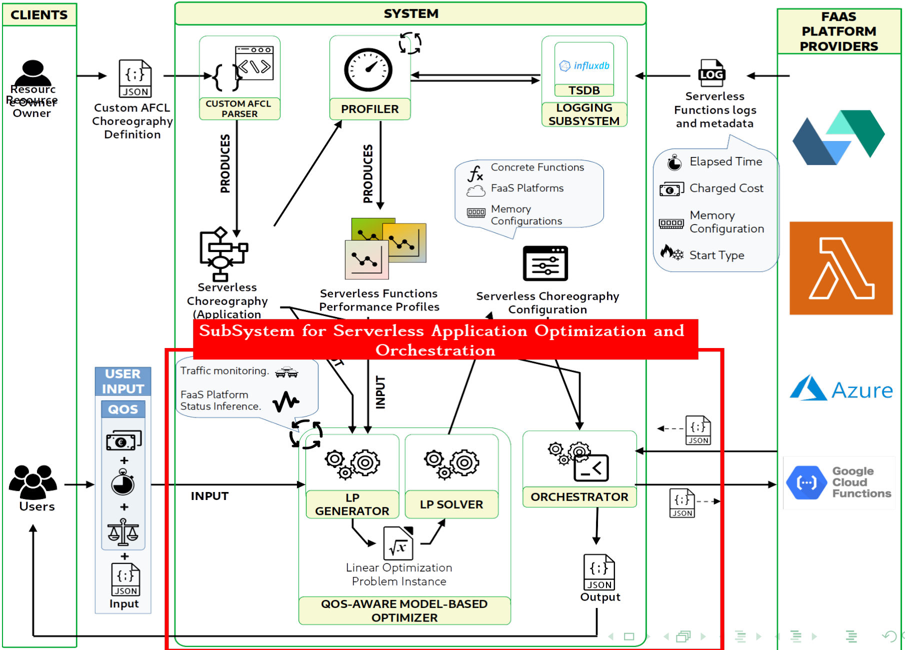
Pheromone trails are used to decide which solutions should be preferred during **next iterations**.

Main features of our software framework:

- **Client-server architecture**.
- **Cloud-native** application.
- Includes a set of **adapters** to interact with following FaaS providers:
  - AWS Lambda.
  - Apache OpenWhisk.
- **REST** architectural style.


CLIENTS
SYSTEM
FAAS PLATFORM PROVIDERS
Resource Owner
Custom AFCL Choreography Definition
JSON
CUSTOM AFCL PARSER
PROFILER
influxdb
TSDB
LOGGING SUBSYSTEM
LOG
Serverless Functions logs and metadata
Elapsed Time
Charged Cost
Memory Configuration
Start Type
PRODUCES
PRODUCES
Concrete Functions
FaaS Platforms
Memory Configurations
Serverless Choreography (Application Workflow)
Serverless Functions Performance Profiles
Serverless Choreography Configuration
INPUT
INPUT
INPUT
Traffic monitoring.
FaaS Platform Status Inference.
USER INPUT
QOS
Users
Input
LP GENERATOR
LP SOLVER
ORCHESTRATOR
Linear Optimization Problem Instance
QOS-AWARE MODEL-BASED OPTIMIZER
Output
Azure
Google Cloud Functions


Subsystem for Concrete Function Profiling
CLIENTS
Resource Owner
Custom AFCL Choreography Definition
JSON
SYSTEM
CUSTOM AFCL PARSER
PRODUCES
PROFILER
influxdb
TSDB
LOGGING SUBSYSTEM
LOG
Serverless Functions logs and metadata
FaaS Platforms
Memory Configurations
Charged Cost
Memory Configuration
Start Type
Serverless Choreography (Application Workflow)
Serverless Functions Performance Profiles
Serverless Choreography Configuration
Traffic monitoring.
FaaS Platform Status Inference.
INPUT
USER INPUT
QOS
Input
Users
LP GENERATOR
LP SOLVER
ORCHESTRATOR
Linear Optimization Problem Instance
QOS-AWARE MODEL-BASED OPTIMIZER
Output
FAAS PLATFORM PROVIDERS
Azure
Google Cloud Functions

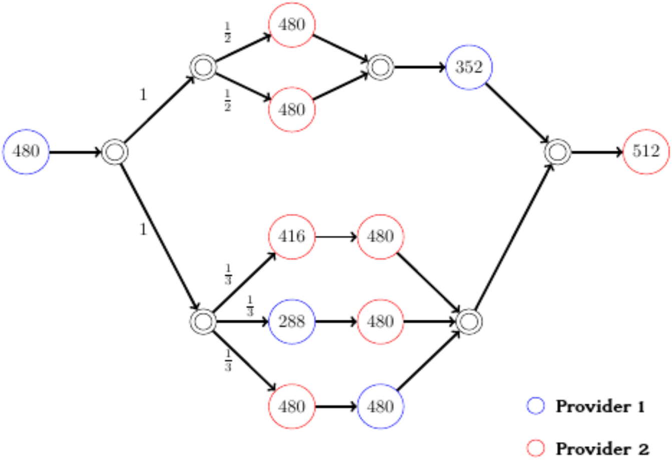

CLIENTS
SYSTEM
FAAS PLATFORM PROVIDERS
Resource Owner
Custom AFCL Choreography Definition
JSON
CUSTOM AFCL PARSER
PROFILER
influxdb
TSDB
LOGGING SUBSYSTEM
LOG
Serverless Functions logs and metadata
Elapsed Time
Charged Cost
Memory Configuration
Start Type
PRODUCES
PRODUCES
Concrete Functions
FaaS Platforms
Memory Configurations
Serverless Choreography (Application
Serverless Functions Performance Profiles
Serverless Choreography Configuration
SubSystem for Serverless Application Optimization and Orchestration
Traffic monitoring.
FaaS Platform Status Inference.
INPUT
INPUT
USER INPUT
QOS
+
+
+
Input
Users
LP GENERATOR
LP SOLVER
ORCHESTRATOR
Linear Optimization Problem Instance
QOS-AWARE MODEL-BASED OPTIMIZER
Output
Azure
Google Cloud Functions

Validation Test

We validate our model through several experiments using an image-processing serverless application.

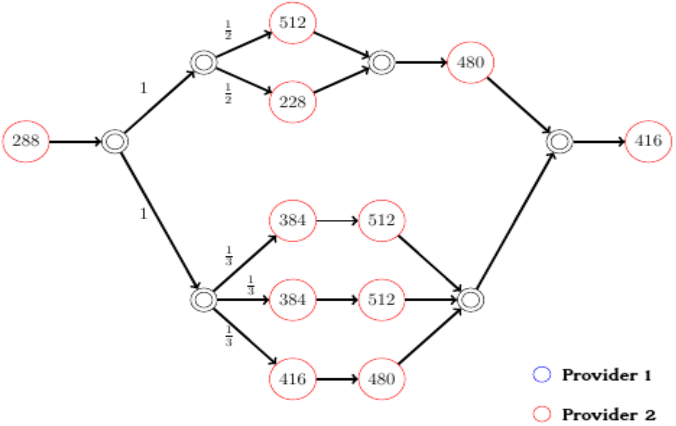
- Firstly, we check the respect of user specified QoS constraints in a static way thorough several **sequential invocations**.
- Then, we test the model in a dynamically way thorough **several concurrent and parallel invocations**.
  - That experiment aimed to **run out of capacity** on one FaaS provider and so force our prototype to schedule the concrete function on the other.

- Application configuration produced by our system **before** Provider 1 runs out of capacity...

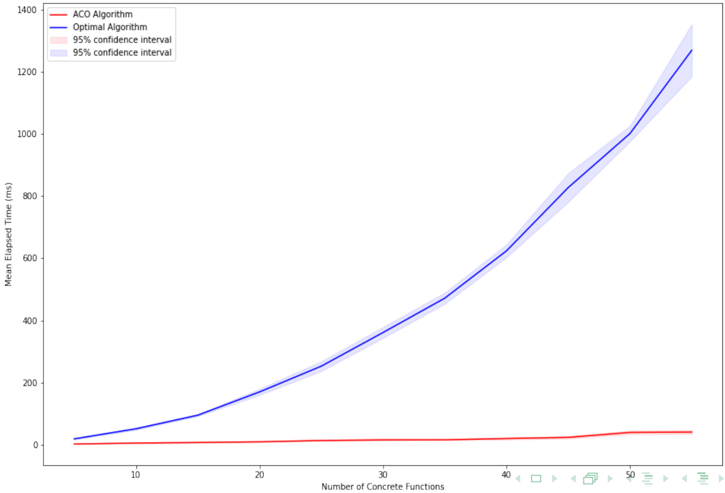


Validation Test

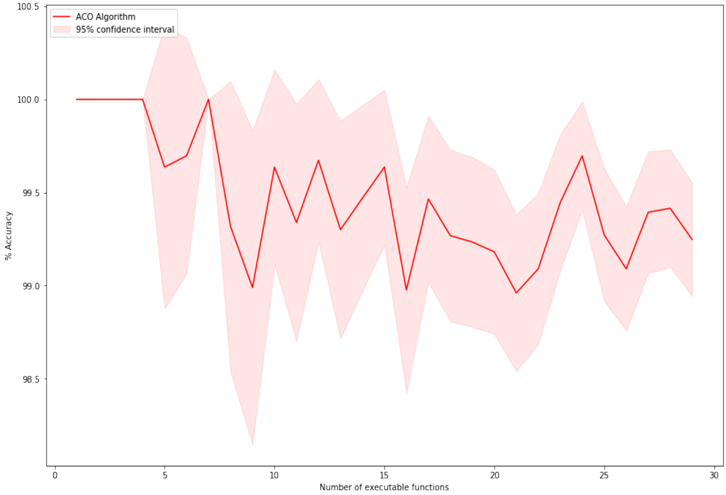
- …and **after**, when Provider 1 runs out of capacity.



- We compared **execution time** of our heuristic algorithm with that of optimal algorithm through several experiments.



- According to my results, **accuracy** is above **98%**.



Conclusions

1. I presented an analytical model to evaluate the performance of multi-provider serverless application.
2. I defined an optimization problem formulation to address the problem of finding a suitable configuration in order to meet user defined QoS constraints.
3. I developed a heuristic algorithm to rapidly solve it.
4. I validate proposed solution through test and experimental evaluations using my prototype.

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Thanks for your attention!

Questions?