DATA-AWARE APPLICATION PLACEMENT AND ROUTING IN THE CLOUD-IoT CONTINUUM

JACOPO MASSA

jacopo.massa@phd.unipi.it

pages.di.unipi.it/massa
**RESEARCH**

**CONTEXT**

Cloud

Edge

IoT

Microservice-based architecture

(\(~\text{hundreds}\) ↓ processing ↓ storage)

(\(~\text{thousands}\) ↑ processing ↑ storage)

(\(~\text{billions}\) ↓ processing ↓ storage)

**PROBLEM**

Where to *place* application services and how to *route traffic* between them in a context- and QoS-aware manner?

**RELATED WORK**


Software, Hardware, IoT

Latency, Bandwidth

Security (only a few)

SDN Routing
WE AIMED TO:

• Devise a declarative modelling of Cloud-IoT infrastructures and multi-service applications, to determine eligible application placements and data traffic routings across Cloud-IoT resources in a context-, QoS-, and data-aware manner.

• Exploit continuous reasoning to speed-up decision making at runtime.

• Implement and assess the proposed solution in a Prolog open-source tool.
CONTINUOUS REASONING

• Adapt placement and routing at runtime.

• Triggered at each infrastructure/application change.

• Partial re-deployment, focusing only on suffering services.

• Speed-up the whole placement and routing search process.
Example application

Example infrastructure

DA-Placer Output
Example infrastructure

Example application

```
:- daplacer(museuMonitor, Placement, Routes).

% on(Service, Node)
Placement = [on(dataStorage, isp),
  on(controller, lifeSciences),
  on(interface, mannLab)],

% ((source, target), AllocatedBandwidth, Route)
Routes = [((dataStorage, controller), 18, [isp, firePolice, westEntry, mannLab, lifeSciences]),
  ((interface, controller), 40, [mannLab, westEntry, parkingServices, lifeSciences])
...
```

DA-Placer Output
NEXT STEPS

PROGRESS W.R.T THE STATE-OF-THE-ART

• Prolog prototype (https://github.com/di-unipi-socc/daplacer) that can be used to:
  • model data, services and IoT devices in a data-aware manner,
  • jointly place both data and services.

• security requirements

• runtime adaptation (continuous reasoning approach)

LIMITATIONS AND FUTURE WORK

• extending the model to account for serverless/FaaS

• multi-objective optimisation (evaluate the goodness of a solution, “greenness” included)

• identify interesting application contexts (AI applications, …)

• validate placement and routing solutions on real testbeds

• further management decision (scalability, undeploy, Osmotic)

• increase prototype usability (e.g. user-friendly tools)
THANK YOU FOR YOUR ATTENTION!