



HAL
open science

Towards Container-layer-aware Scheduling Policies for Serverless-based Edge-Cloud Continuum

Anderson Andrei Da Silva, Yiannis Georgiou, Michael Mercier, Grégory Mounié, Denis Trystram

► To cite this version:

Anderson Andrei Da Silva, Yiannis Georgiou, Michael Mercier, Grégory Mounié, Denis Trystram. Towards Container-layer-aware Scheduling Policies for Serverless-based Edge-Cloud Continuum. DevConf.CZ 2023, Jun 2023, Brno, Czech Republic. hal-04244617

HAL Id: hal-04244617

<https://inria.hal.science/hal-04244617>

Submitted on 16 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Towards Container-layer-aware Scheduling Policies for Serverless-based Edge-Cloud Continuum

DevConfcz2023

Anderson Andrei Da Silva^{1 2}, Y. Georgiou², M. Mercier², G. Mounié¹, D. Trystram¹.

¹Université Grenoble Alpes, ²Ryax Technologies

Contact: anderson-andrei.da-silva@inria.fr, silva.andersonandrei@gmail.com

<https://andersonandrei.github.io/>

16-18 of Jun of 2023



PHYSICS



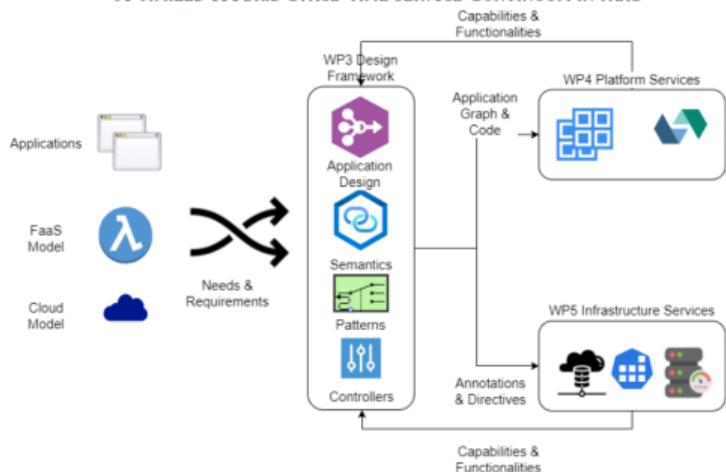
Project Goals

Visual programming environment to create serverless workflows with reusable patterns and increased semantics

Platform-level functionalities to orchestrate and deploy FaaS workflows and optimize cloud/edge interplay

Provider-local resource management mechanisms to offer competitive and optimized services execution

OPTIMIZED HYBRID SPACE-TIME SERVICE CONTINUUM IN FAAS



Friday, June 16 • 2:15pm - 3:35pm

✓ Writing a K8s Operator for Knative Functions

[Click here to remove from My Schedule.](#)

<https://sched.co/1MYp>

[Tweet](#)

[Share](#)

Feedback Submitted

Serverless and Function as a Service (FaaS) are getting more and more attention from customers and developers as a way to develop, run and manage applications functionality without the burden of infrastructure related knowledge. All big cloud providers offer them already, e.g., AWS Lambda, Google Cloud Functions or Microsoft Azure Functions. One of the most relevant upstream projects for serverless is Knative, which recently added support for functions (create, build, and deploy) on top of K8s clusters.

This workshop will introduce you to the FaaS model, as well as to building Kubernetes operators. You will implement a K8s Operator, using the operatorsdk framework, to provide the functionality of the Knative CLI. This will allow to easier create, build and deploy functions with Knative just by creating Kubernetes (CR) objects, and will help you learning the internals about how K8s Operators work in a real life example.

Speakers



[Luis Tomas Bolivar](#)
Red Hat



[Jose Castillo Lema](#)
Senior Software Engineer, Red Hat

Sunday, June 18 • 2:45pm - 3:20pm

✓ Design & Deployment of FaaS apps at the edge-cloud

[Click here to remove from My Schedule.](#)

<https://sched.co/1Mmi>

[Tweet](#)

[Share](#)

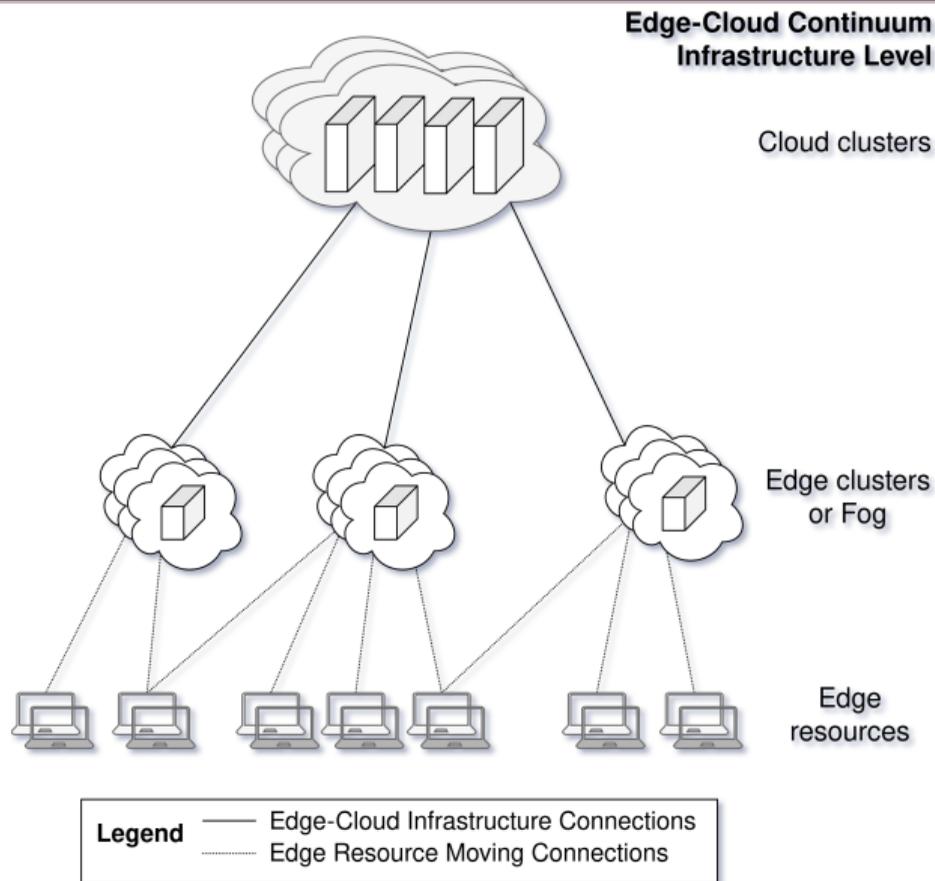
The talk will focus on a design and development environment coming from the H2020 PHYSICS project, that aims to ease application evolution to the new FaaS model. It uses the Node-RED open source tool as the main function and workflow runtime. The goal of the environment is to enable a more user friendly and abstract function and workflow creation process for complex FaaS applications. To this end, it provides an extendable, pattern-enriched palette of ready-made, reusable functionalities such as workload parallelization, data collection at the edge, function orchestration creation among others. The environment embeds seamless DevOps processes for generating the deployable artefacts of the FaaS platform (Openwhisk). Annotation mechanisms are also available for the developer to dictate diverse execution options towards the deployment stacks, including sizing and locality considerations, as well as abilities for dynamic FaaS applications to continuously leverage the edge-cloud continuum.

Speakers



Yiannis Georgiou

The edge-cloud continuum



From the cloud to the serverless

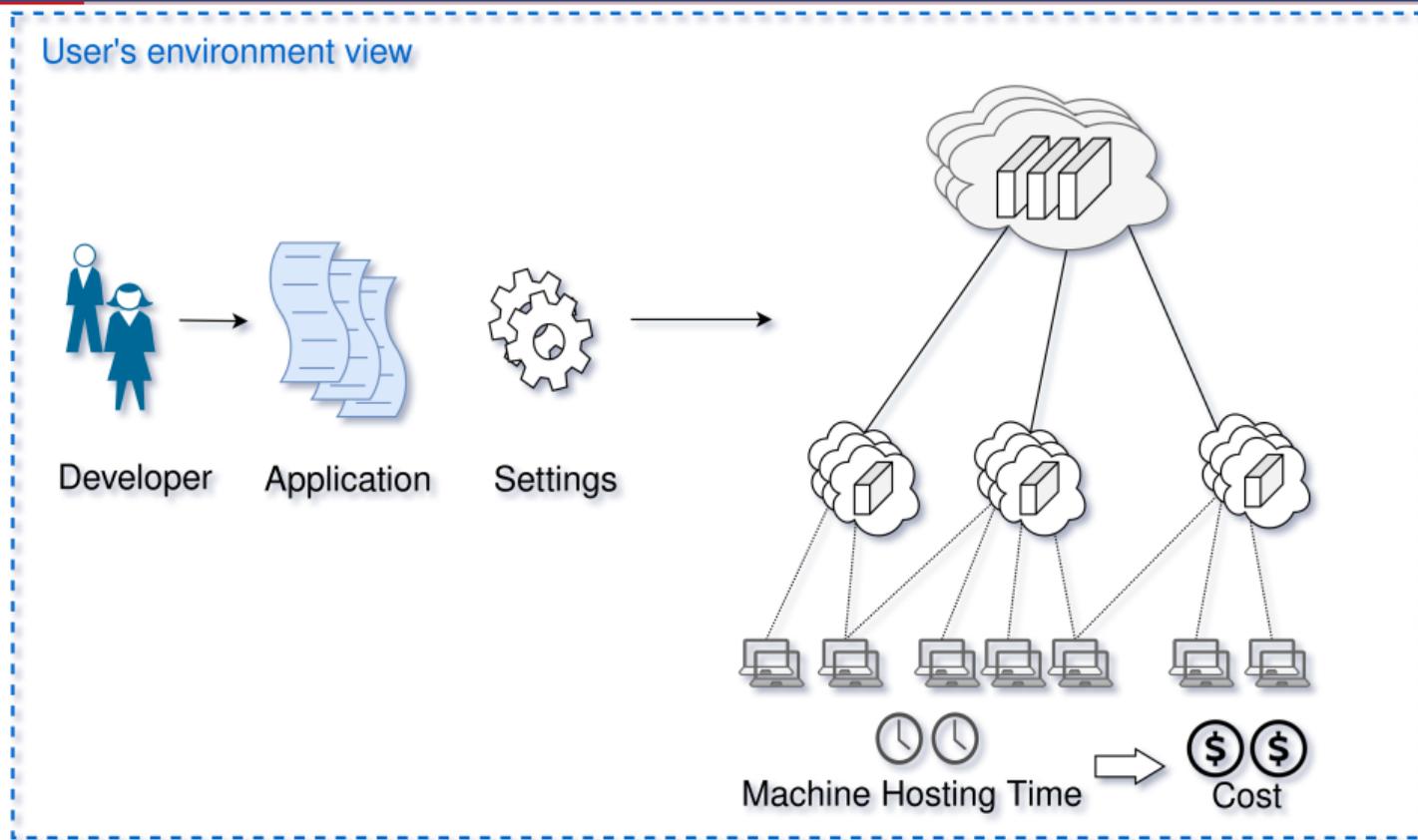


Figure 2: From the cloud to the serverless - step 1.

From the cloud to the serverless

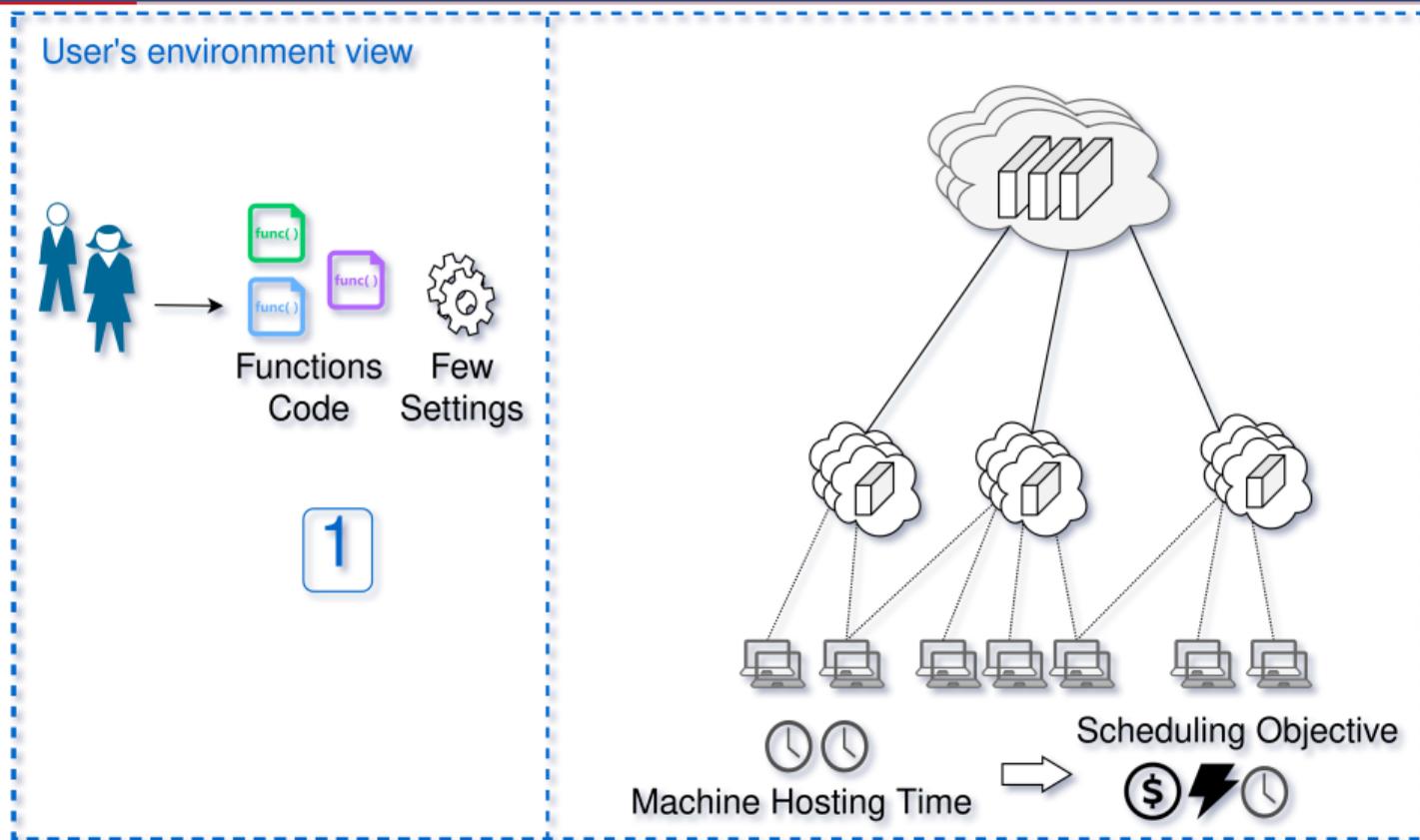
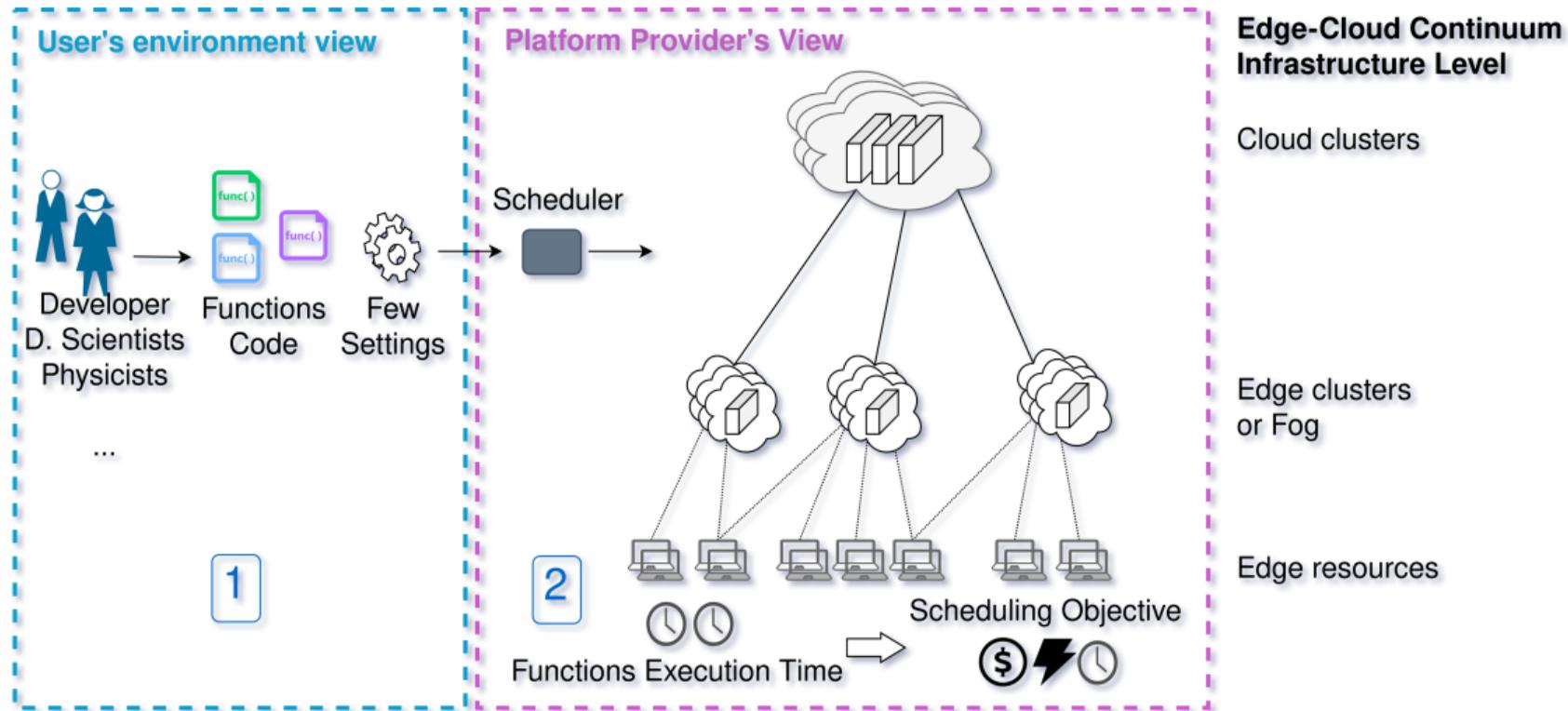


Figure 3: From the cloud to the serverless - step 2.

Serverless Computing and the Edge-Cloud Continuum



Towards a Multi-objective Scheduling Policy for Serverless-based Edge-Cloud Continuum

Towards a Multi-objective Scheduling Policy for Serverless-based Edge-Cloud Continuum

The 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing
Bangalore, India — May 1-4, 2023

L. Angelelli¹, **Anderson Andrei Da Silva**^{1 2}, Y. Georgiou²,
M. Mercier², G. Mounié¹, D. Trystram¹.

¹Université Grenoble Alpes, ²Ryax Technologies



Reproducible
Artifacts

<https://andersonandrei.github.io/>

<https://gitlab.com/andersonandrei/foa-a-multi-objective-scheduling-policy-for-serverless>

Thanks to the Intelligence of the Multi-disciplinary Institute on Artificial Intelligence MIAI at Grenoble Alpes (ANR-19-P3IA-0003), to the European Union's Project H2020 PHYSICS (GA 101017047), and to the Grid'5000 testbed (<https://www.grid5000.fr>).



ryax
technologies



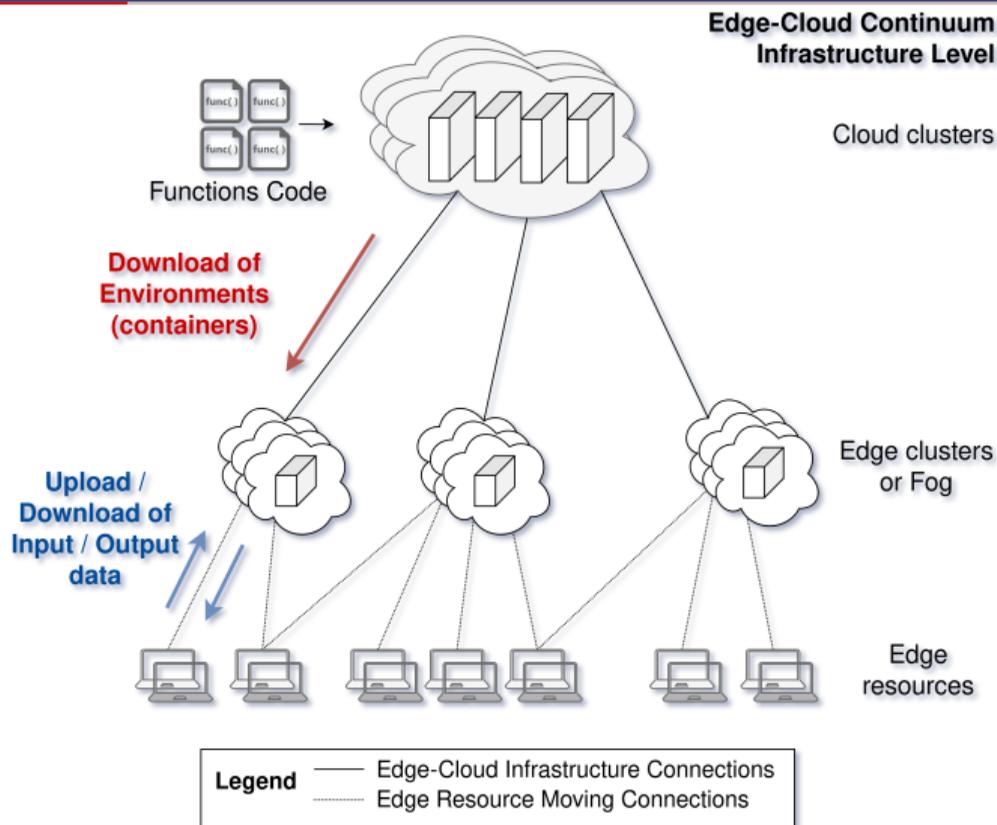
MIAI
Grenoble Alpes



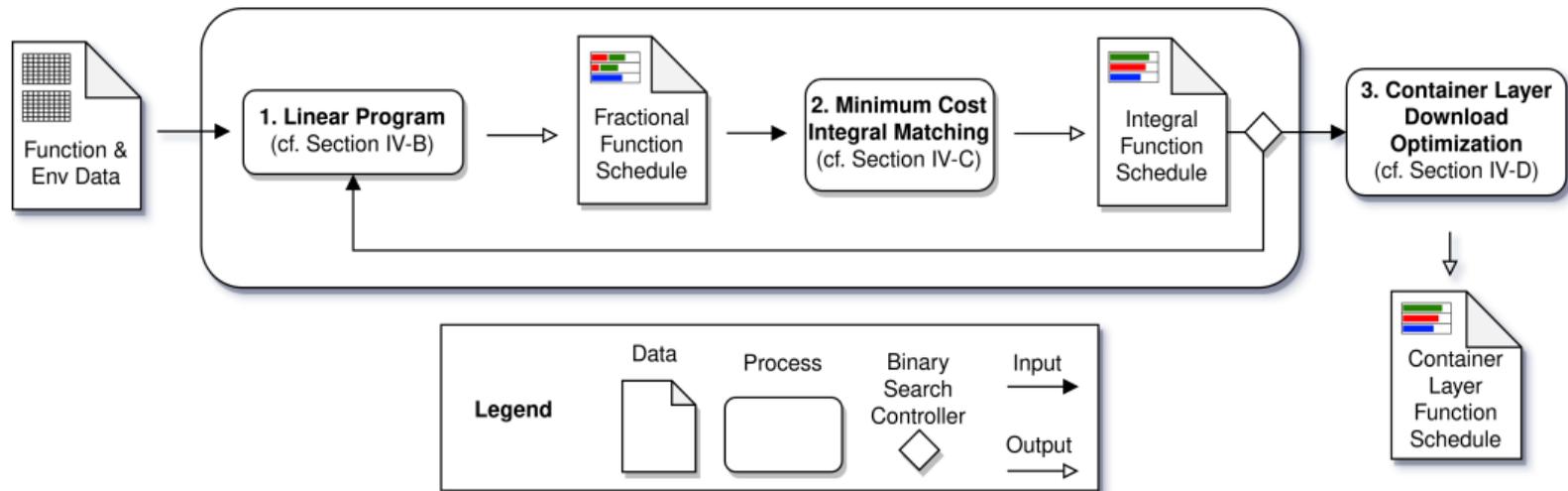
PHYSICS

- Presents an evolution in the sense of use of **containers**,
- **However**, containers' deployment time is **not negligible**,
- Nevertheless, container can **share layers**.
- Figure of Layers

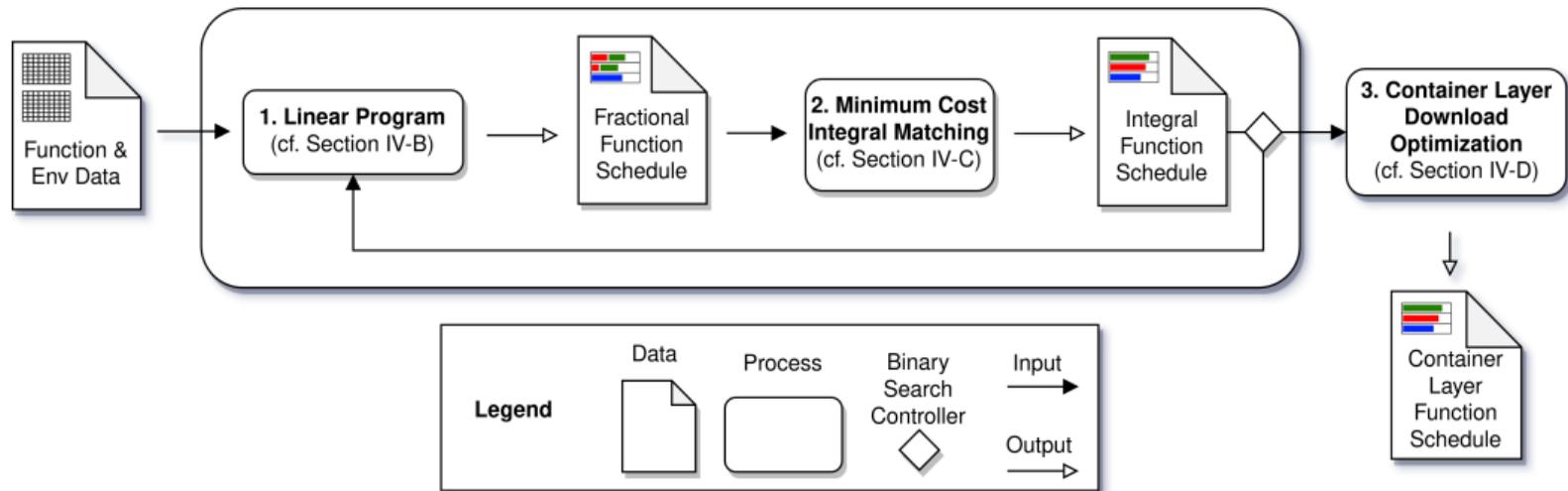
Motivation 1 and a multi-objective approach



FOA: Function Orchestration Algorithm



FOA: Function Orchestration Algorithm



The *Minimum Cost Integral Matching* phase uses the Dual-Approx. Algorithm of Shmoys and Tardos [D. B. Shmoys and E. Tardos, "An approximation algorithm for the generalized assignment problem," *Math. Program.*, vol. 62, no. 1–3, p.461–474, Feb. 1993].

Experimental Protocol

- Adaptation of the benchmark **FunctionBench** ¹,
- Deployment of **OpenWhisk** ² on **GRID5000** ³,
 - Measurement and calibration of results.
- Evaluation of scheduling policies FOA, K8S ImageLocality (baseline):
 - In a **simulated environment**,
 - On top of **Batsim/ Simgrid** [H.Casanova et.al, 2014].

¹ <https://github.com/kmu-bigdata/serverless-faas-workbench> ,

² <https://openwhisk.apache.org/>,

³ <https://www.grid5000.fr> .

Parameters	Values
Workload Size	200, 600, 1000
Platform Size	100, 300, 500
Heterogeneity Level	3, 5, 7
Scheduling Policies	FOA, <i>K8S ImageLocality</i>
Random Seeds	30

Table 1 Design of Experiments. Total of 1620 experiments.

Design of Experiments

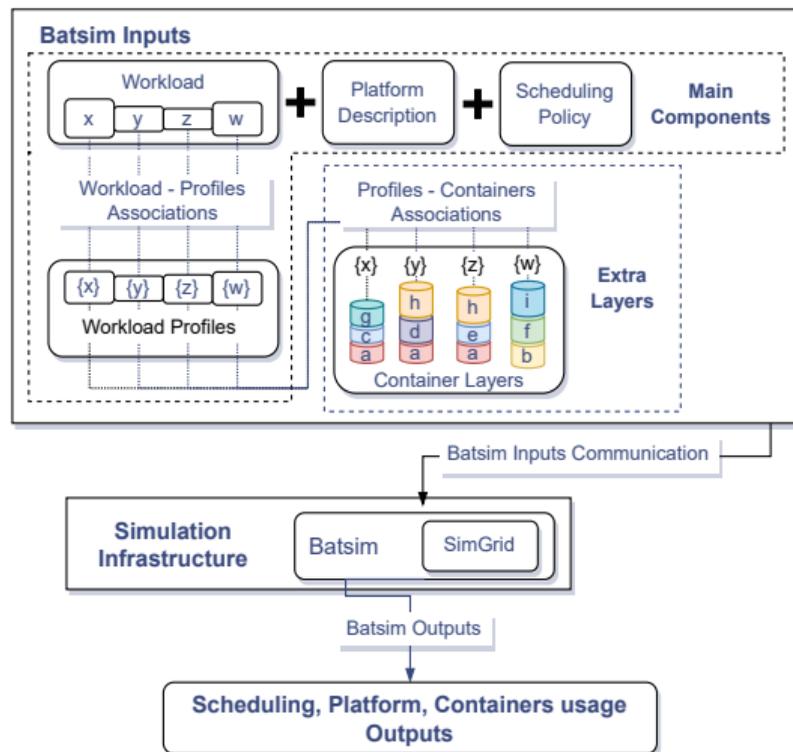
Parameters	Values
Workload Size	200, 600, 1000
Platform Size	100, 300, 500
Heterogeneity Level	3, 5, 7
Scheduling Policies	FOA, <i>K8S ImageLocality</i>
Random Seeds	30

Table 1 Design of Experiments. Total of 1620 experiments.

Function	Input Values	Unit
Chameleon	$2 \cdot 10^3$, $3 \cdot 10^3$	Matrix size
Float operation	$10 \cdot 10^7$, $20 \cdot 10^7$, $30 \cdot 10^7$	Operations
Image processing	60, 40	Image size in MB
Linpack	$5 \cdot 10^3$, $6 \cdot 10^3$	Matrix size
Matrix Multiplication	$3 \cdot 10^3$, $4 \cdot 10^3$	Matrix size
Model Training	50, 100	Dataset size
Pyaes	$3 \cdot 10^3$, $4 \cdot 10^3$, $5 \cdot 10^3$	Length of message
Video Processing	30, 50, 100	Video size in MB

Table 2: Functions adapted from FunctionBench.

Simulation infrastructure



ddps-lab / serverless-faas-workbench PUBLIC

Code Issues Pull requests 15 Actions Projects Wiki Security Insights

Filters is:pr author:andersonandreil is:closed

Clear current search query, filters, and sorts

0 Open 1 Closed

Author Label Projects Milestones Reviews Assignee Sort

Functions adapted to the OpenWhisk platform #37 by andersonandreil was merged on Jan 17 • Approved

ddps-lab / serverless-faas-workbench PUBLIC

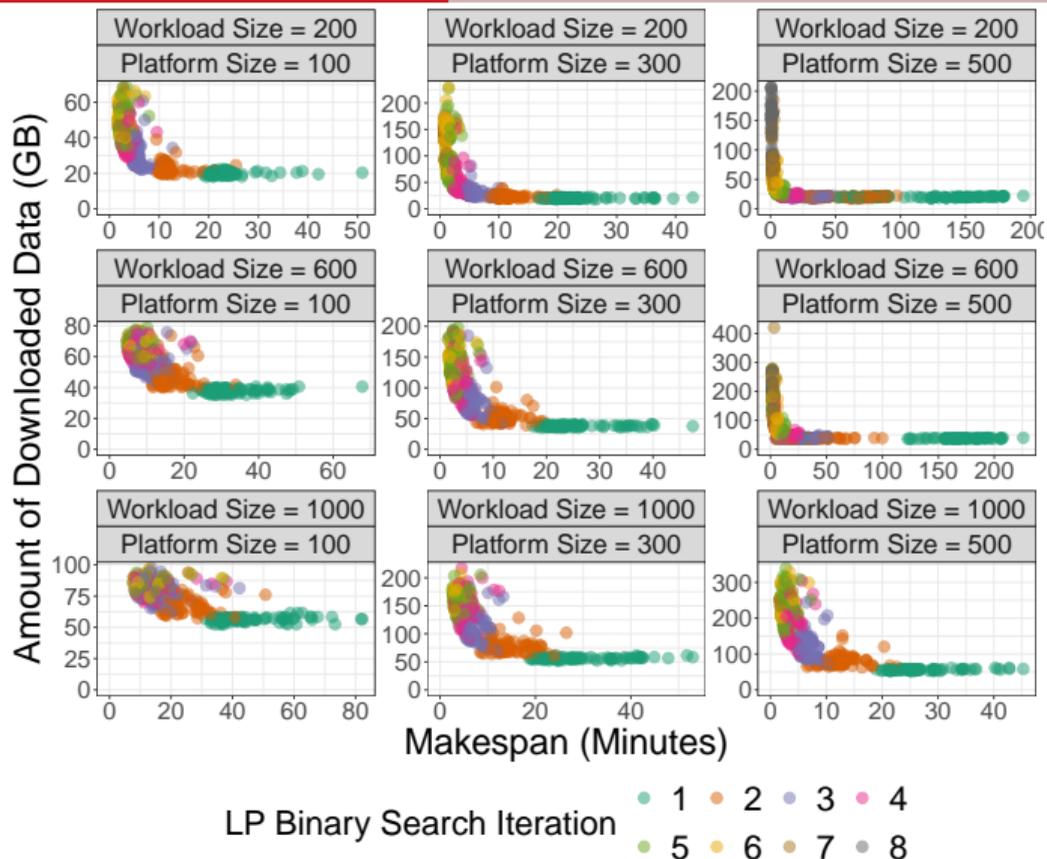
Code Issues Pull requests 15 Actions Projects Wiki Security Insights

serverless-faas-workbench / openwhisk / cpu-memory /

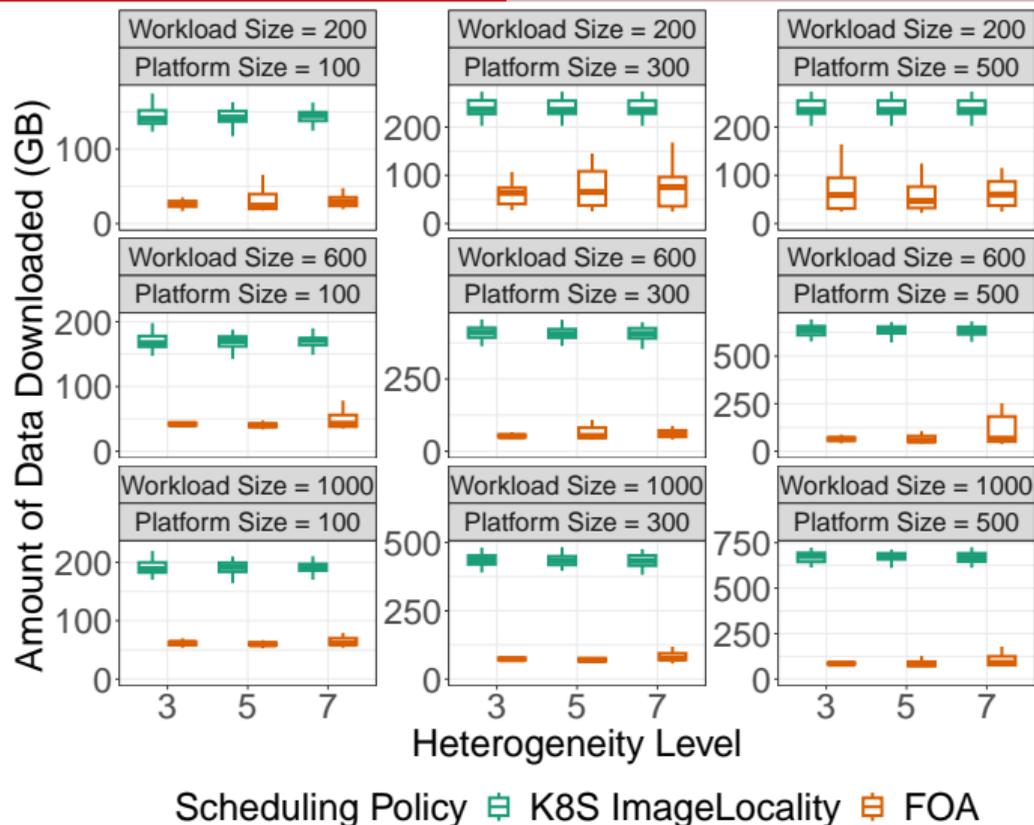
Sharlar076 added missing cpu-memory functions for openwhisk bb332f9 · 3 months ago History

Name	Last commit message	Last commit da...
..		
chameleon	Adapt functions to the OpenWhisk platform	5 months ago
feature_generation	added missing cpu-memory functions for openwhisk	3 months ago
float_operation	Adapt functions to the OpenWhisk platform	5 months ago
image_processing	Adapt functions to the OpenWhisk platform	5 months ago
linpack	Adapt functions to the OpenWhisk platform	5 months ago
mapreduce	added missing cpu-memory functions for openwhisk	3 months ago
matmul	Adapt functions to the OpenWhisk platform	5 months ago
model_serving	added missing cpu-memory functions for openwhisk	3 months ago
model_training	Adapt functions to the OpenWhisk platform	5 months ago

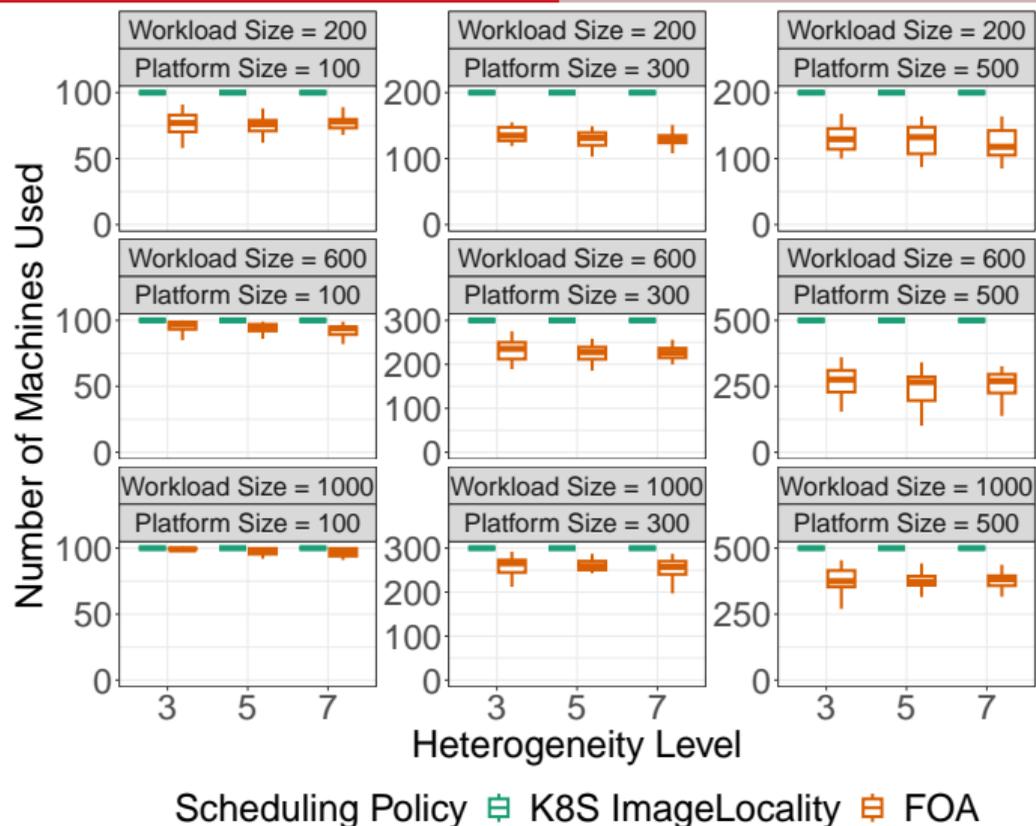
Experimental results: Linear program trade-off



Experimental results: Amount of data downloaded



Experimental results: Number of machines used



- Cloud **greedy algorithms** may **not** profit from heterogeneity.

Conclusions - Part 1

- Cloud **greedy algorithms** may **not** profit from heterogeneity.
- FOA outperforms our baseline for both **data transfers** and **makespan** criteria, in addition to the **system utilization** by up to **two orders of magnitudes**.

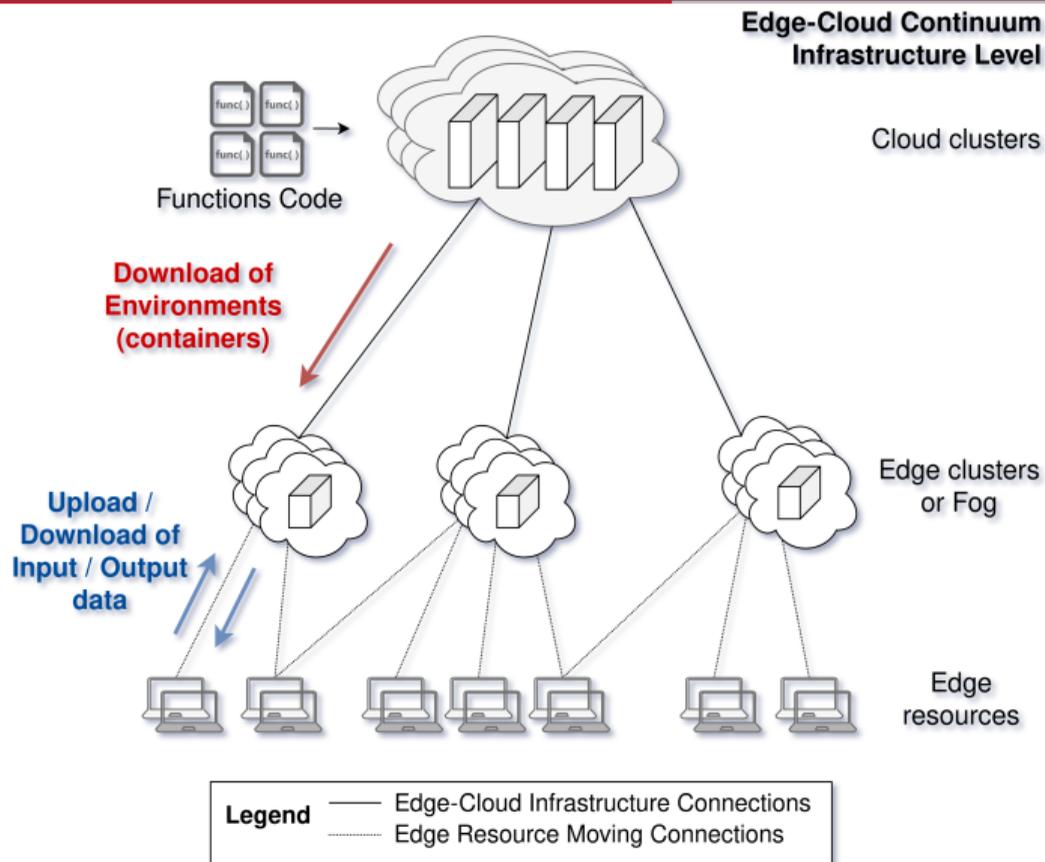
Conclusions - Part 1

- Cloud **greedy algorithms** may **not** profit from heterogeneity.
- FOA outperforms our baseline for both **data transfers** and **makespan** criteria, in addition to the **system utilization** by up to **two orders of magnitudes**.
- FOA minimizes **cold start delays** and **speed up** functions' **execution time**.

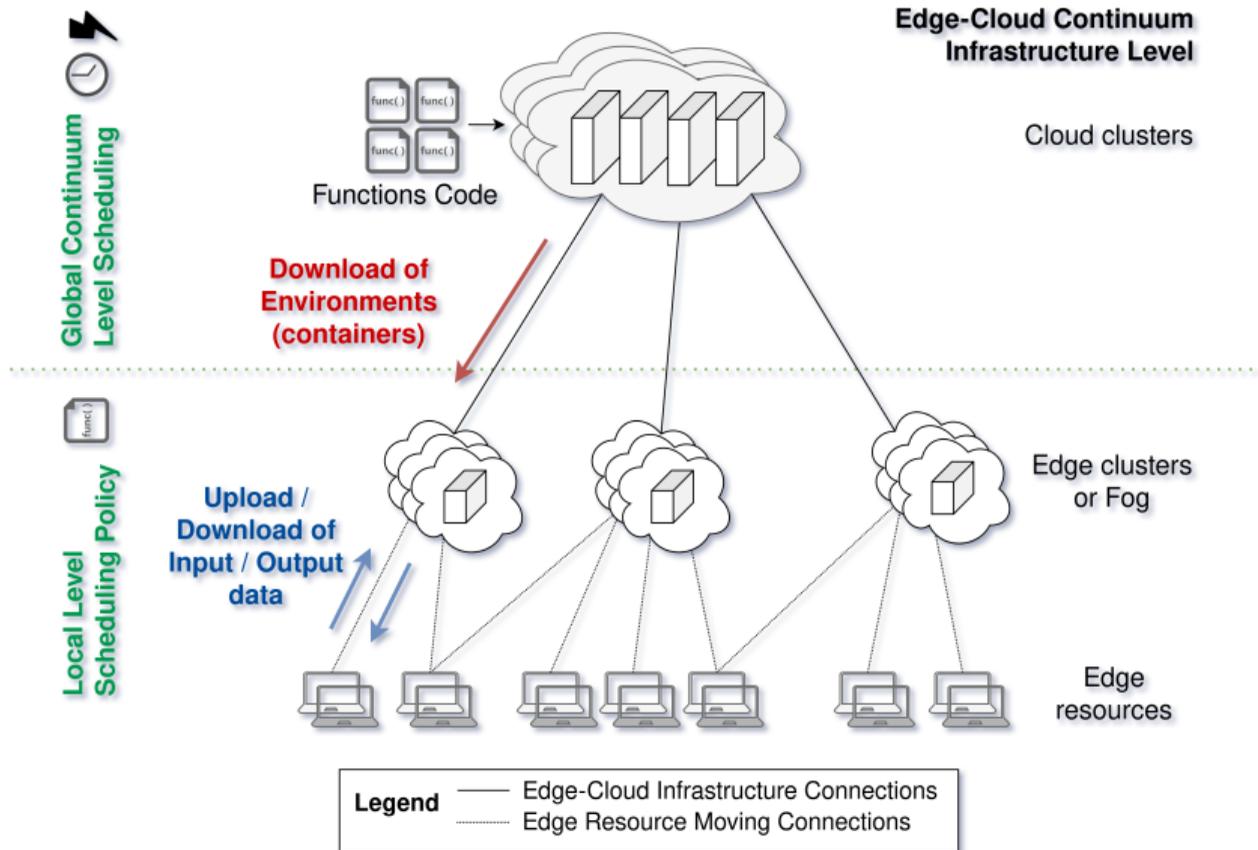
Conclusions - Part 1

- Cloud **greedy algorithms** may **not** profit from heterogeneity.
- FOA outperforms our baseline for both **data transfers** and **makespan** criteria, in addition to the **system utilization** by up to **two orders of magnitudes**.
- FOA minimizes **cold start delays** and **speed up** functions' **execution time**.
- However, FOA is very time-consuming.

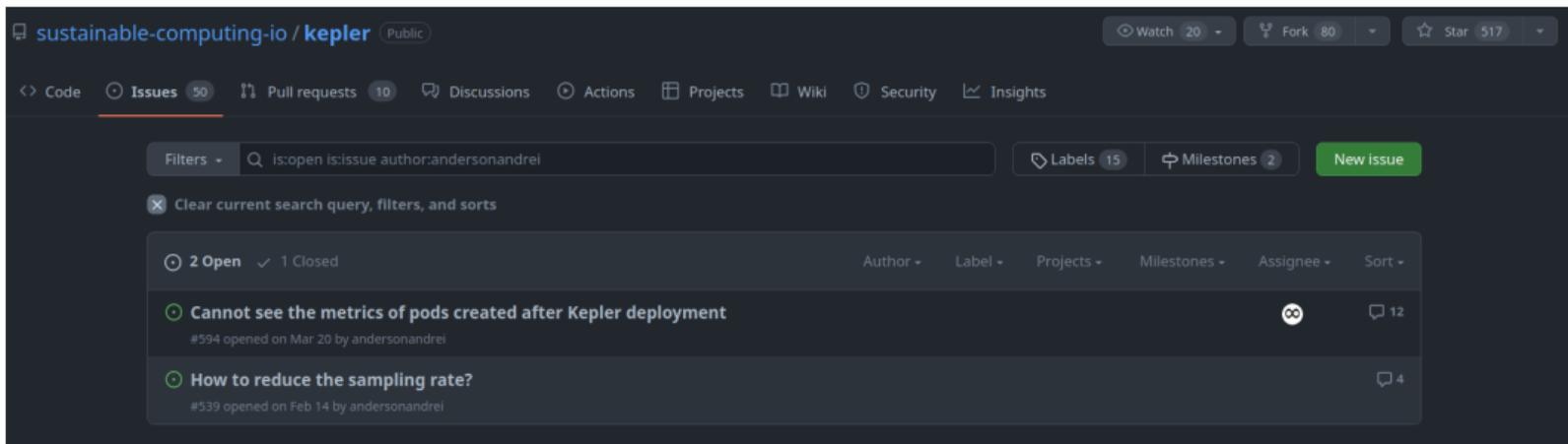
Motivation 2 = Motivation 1 + Energy Consumption



Motivation 2 = Motivation 1 + Energy Consumption



Energy Consumption on Serverless Platforms - Kepler



sustainable-computing-io / kepler Public

Watch 20 Fork 80 Star 517

Code Issues 50 Pull requests 10 Discussions Actions Projects Wiki Security Insights

Filters Labels 15 Milestones 2 [New issue](#)

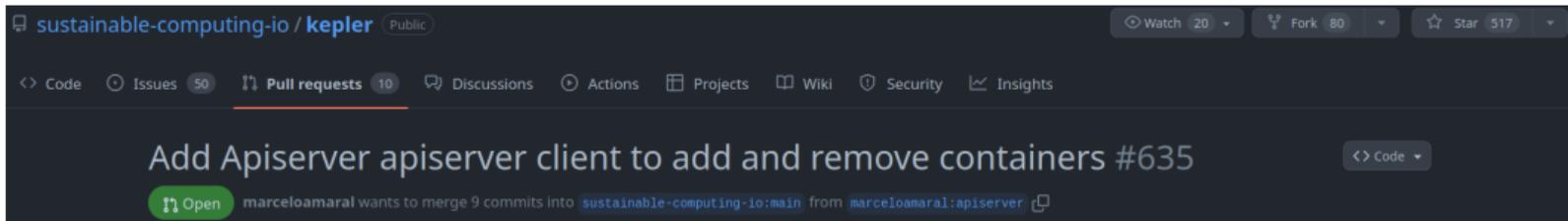
Clear current search query, filters, and sorts

2 Open 1 Closed

	Author	Label	Projects	Milestones	Assignee	Sort
Cannot see the metrics of pods created after Kepler deployment						12
How to reduce the sampling rate?						4

#594 opened on Mar 20 by andersonandrei

#539 opened on Feb 14 by andersonandrei



sustainable-computing-io / kepler Public

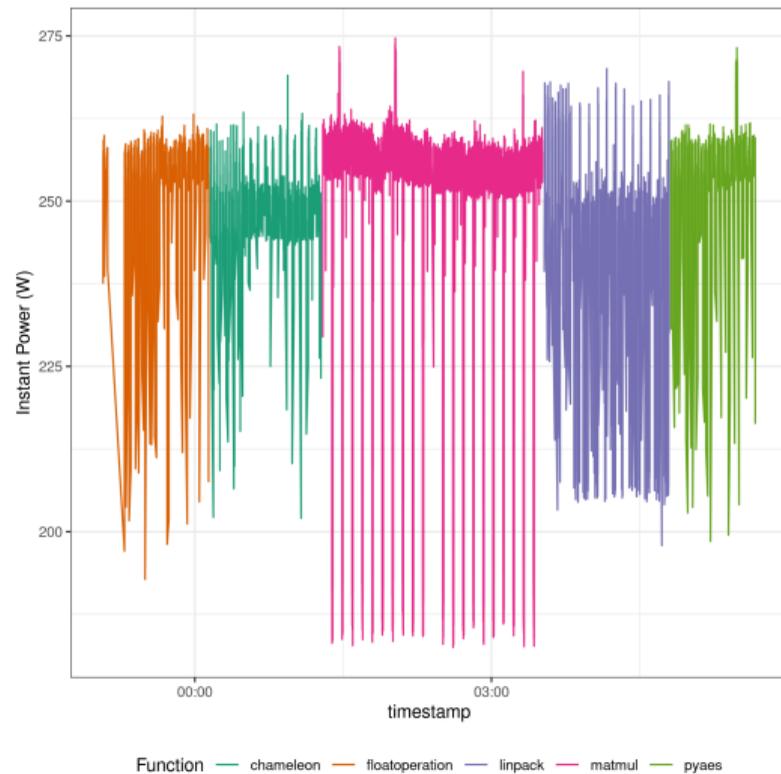
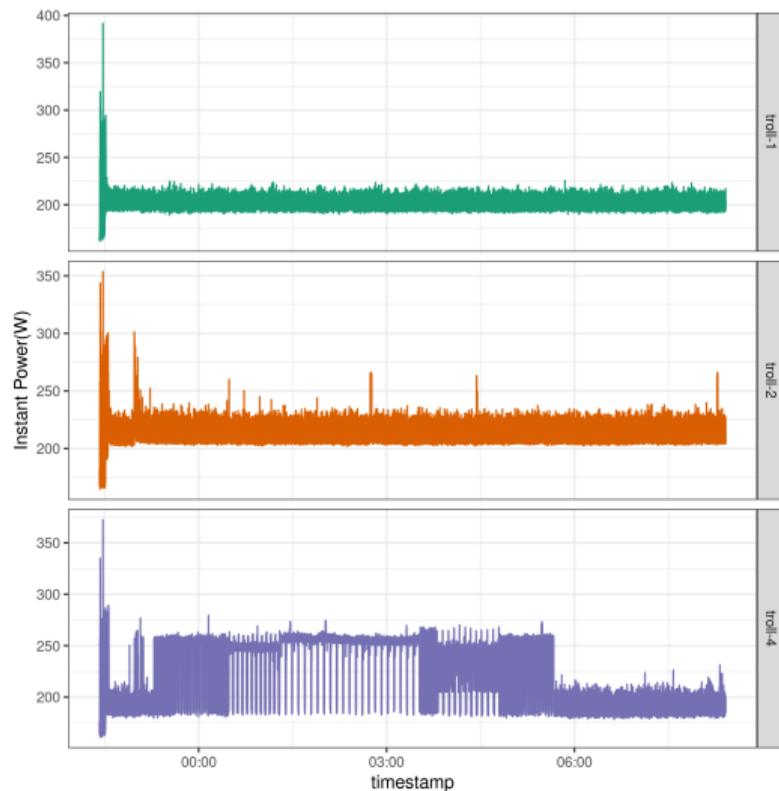
Watch 20 Fork 80 Star 517

Code Issues 50 Pull requests 10 Discussions Actions Projects Wiki Security Insights

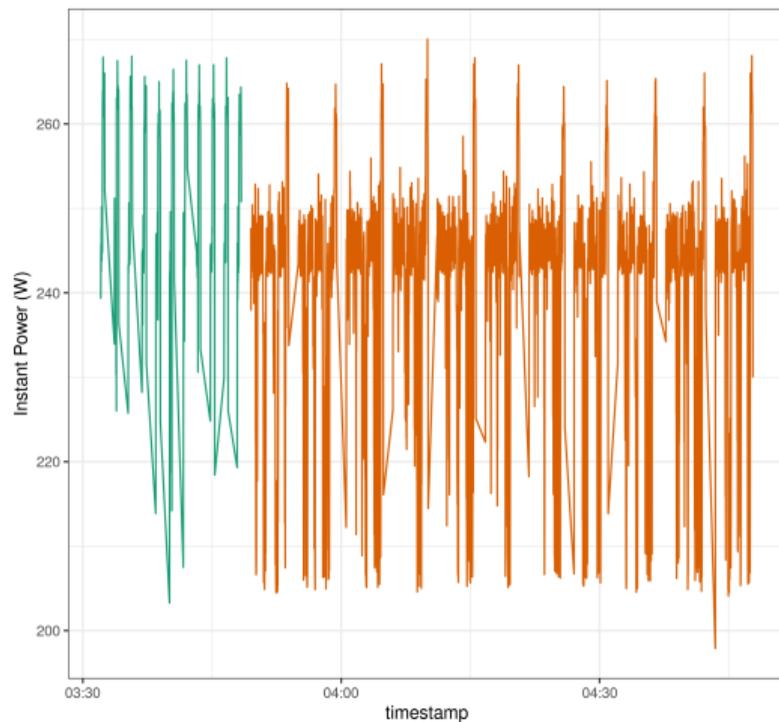
Add Apiserver apiserver client to add and remove containers #635

[Open](#) marceloamaral wants to merge 9 commits into sustainable-computing-io:main from marceloamaral:apiserver

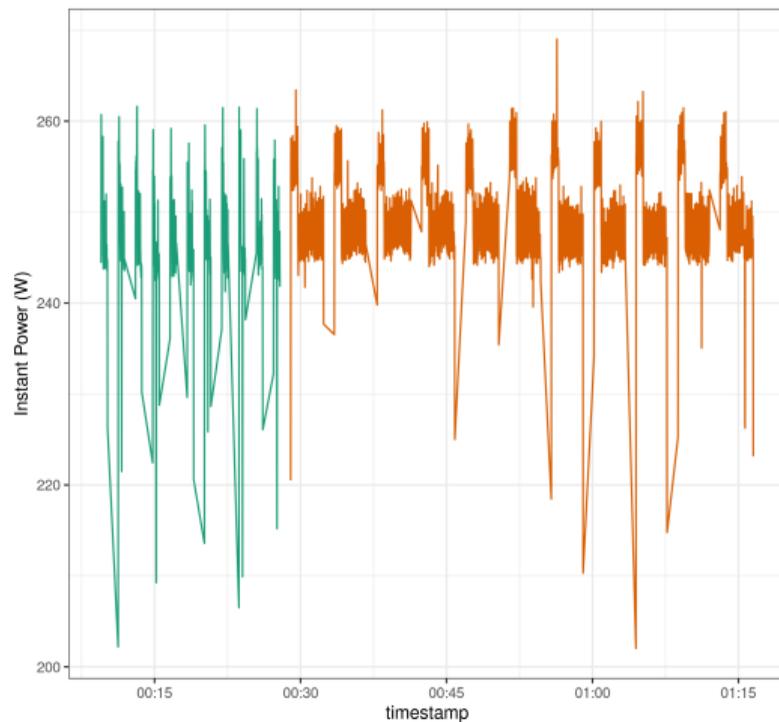
Energy Consumption on Serverless Platforms



Energy Consumption on Serverless Platforms

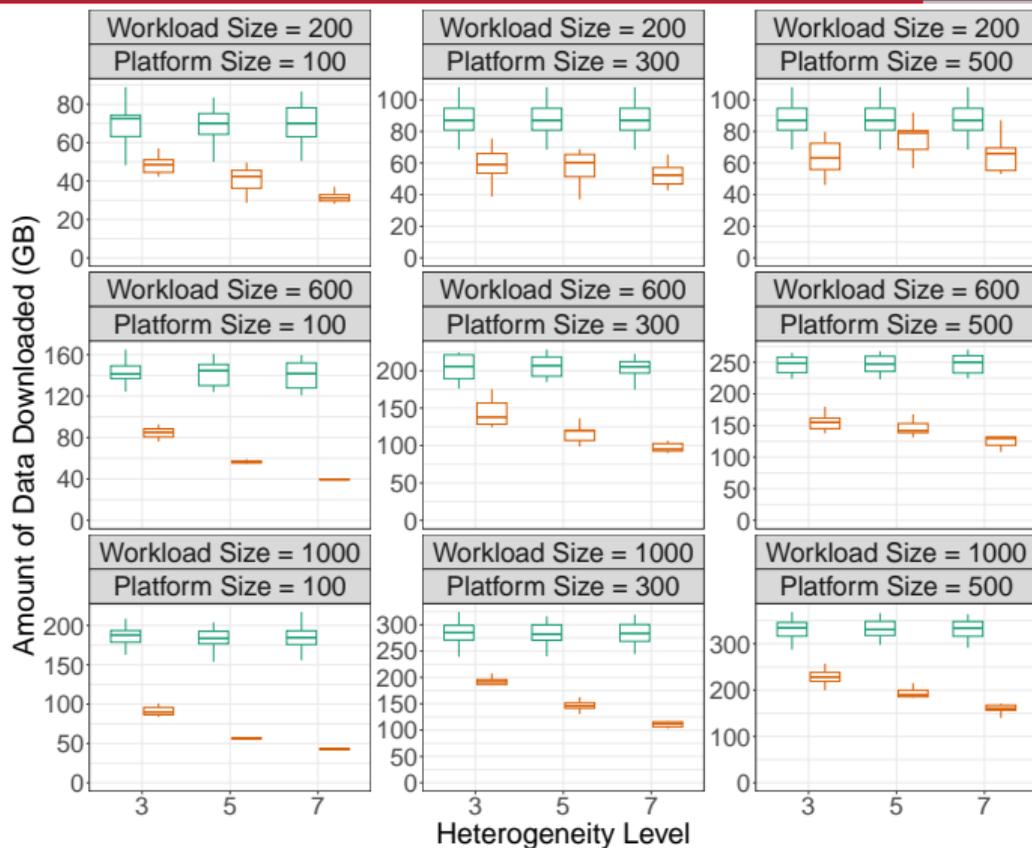


Function — linpack_5000 — linpack_6000



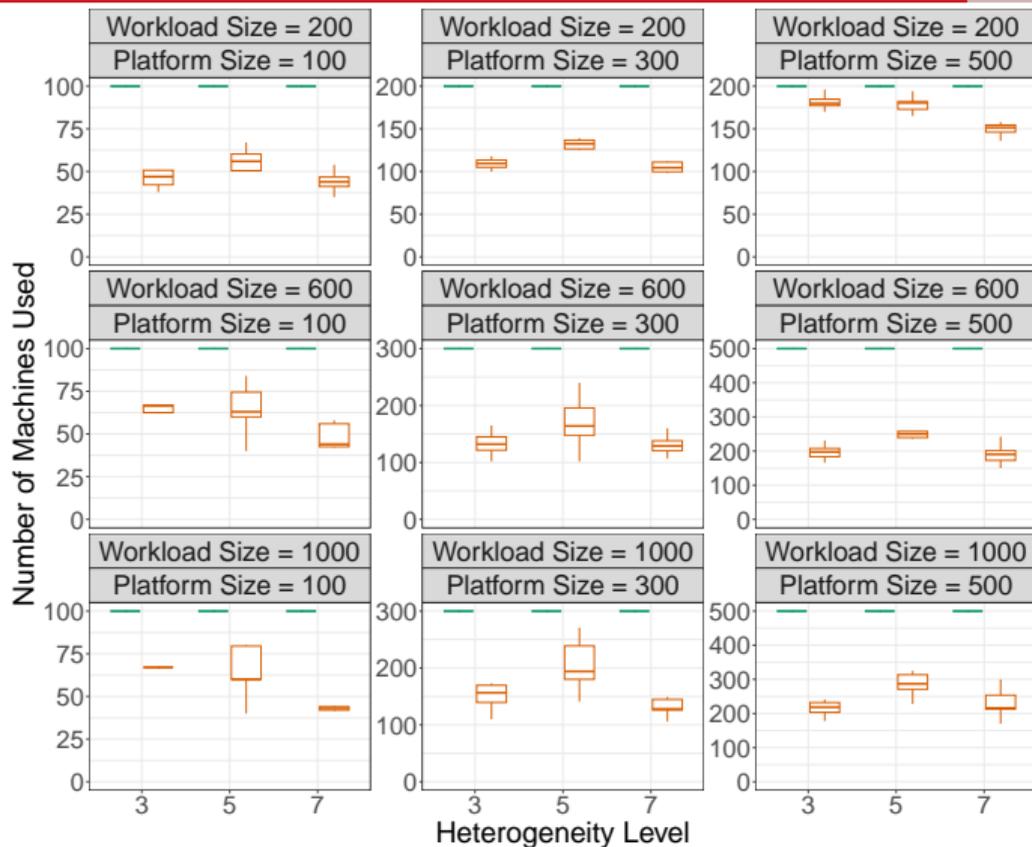
Function — chameleon_2000 — chameleon_4000

Experimental results: Amount of data downloaded



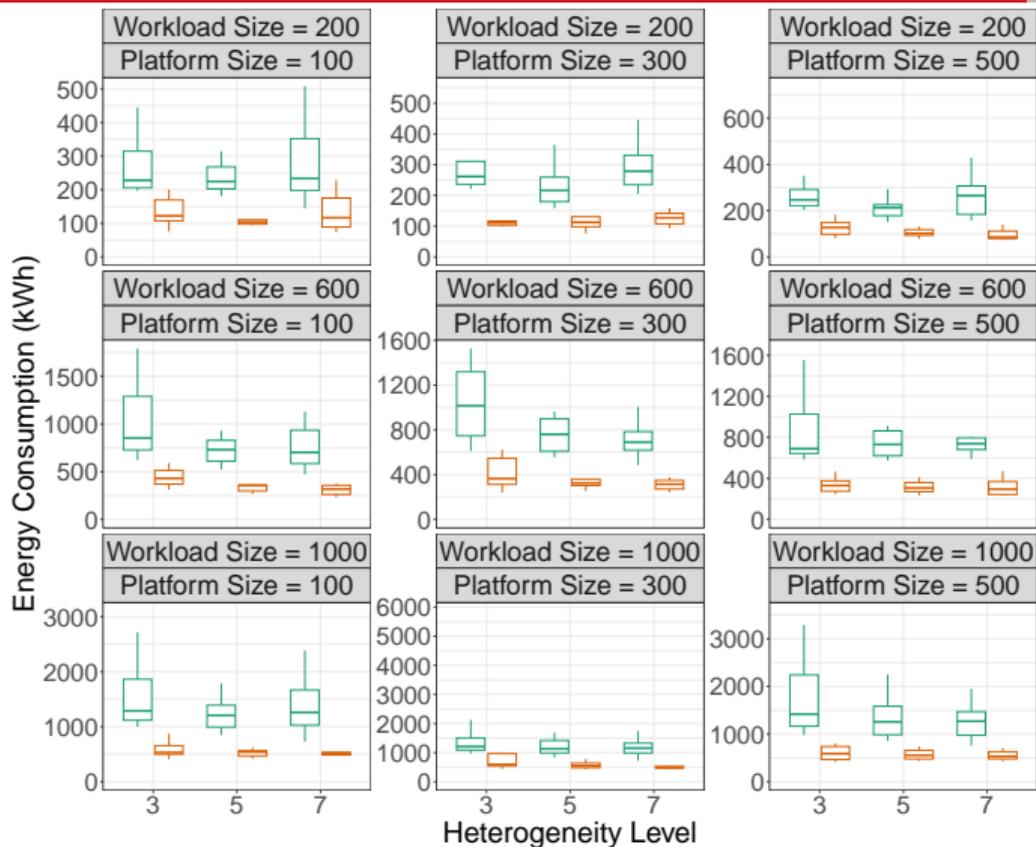
Scheduling Policy ■ FCFS + ImageLocality ■ FOA-e

Experimental results: Number of machines used



Scheduling Policy FCFS + ImageLocality FOA-e

Experimental results: Energy Consumption



Conclusions - Part 2

- Cloud **greedy algorithms** may **not** profit from heterogeneity.

Conclusions - Part 2

- Cloud **greedy algorithms** may **not** profit from heterogeneity.
- FOA-e outperforms our baseline for both **data transfers** and **energy consumption** criteria, in addition to the **system utilization**.

Conclusions - Part 2

- Cloud **greedy algorithms** may **not** profit from heterogeneity.
- FOA-e outperforms our baseline for both **data transfers** and **energy consumption** criteria, in addition to the **system utilization**.
- FOA-e minimizes **cold start delays** and **speed up** functions' **execution time**,

Conclusions - Part 2

- Cloud **greedy algorithms** may **not** profit from heterogeneity.
- FOA-e outperforms our baseline for both **data transfers** and **energy consumption** criteria, in addition to the **system utilization**.
- FOA-e minimizes **cold start delays** and **speed up** functions' **execution time**,
- FOA-e has a scheduling time in order of seconds.

- To improve FOA's **model** and try other linear programs' **solvers**,

Future Work

- To improve FOA's **model** and try other linear programs' **solvers**,
- To study applications that can be modeled as **workflows** of serverless functions,

Future Work

- To improve FOA's **model** and try other linear programs' **solvers**,
- To study applications that can be modeled as **workflows** of serverless functions,
- To include and use Kepler for a continuous measurement of the energy consumption,

Future Work

- To improve FOA's **model** and try other linear programs' **solvers**,
- To study applications that can be modeled as **workflows** of serverless functions,
- To include and use Kepler for a continuous measurement of the energy consumption,
- To continue the investigations towards the reduction of energy consumption on serverless,

Future Work

- To improve FOA's **model** and try other linear programs' **solvers**,
- To study applications that can be modeled as **workflows** of serverless functions,
- To include and use Kepler for a continuous measurement of the energy consumption,
- To continue the investigations towards the reduction of energy consumption on serverless,
- To add both schedulers in the PHYSICS platforms.

Thank you for your attention! Any question?

Towards a Multi-objective Scheduling Policy for Serverless-based Edge-Cloud Continuum

The 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing
Bangalore, India — May 1-4, 2023

L. Angelelli¹, **Anderson Andrei Da Silva**^{1 2}, Y. Georgiou²,
M. Mercier², G. Mounié¹, D. Trystram¹.

¹Université Grenoble Alpes, ²Ryax Technologies



Reproducible
Artifacts

<https://andersonandrei.github.io/>

<https://gitlab.com/andersonandrei/foa-a-multi-objective-scheduling-policy-for-serverless>

Thanks to the Intelligence of the Multi-disciplinary Institute on Artificial Intelligence MIAI at Grenoble Alpes (ANR-19-P3IA-0003), to the European Union's Project H2020 PHYSICS (GA 101017047), and to the Grid'5000 testbed (<https://www.grid5000.fr>).



ryax
technologies

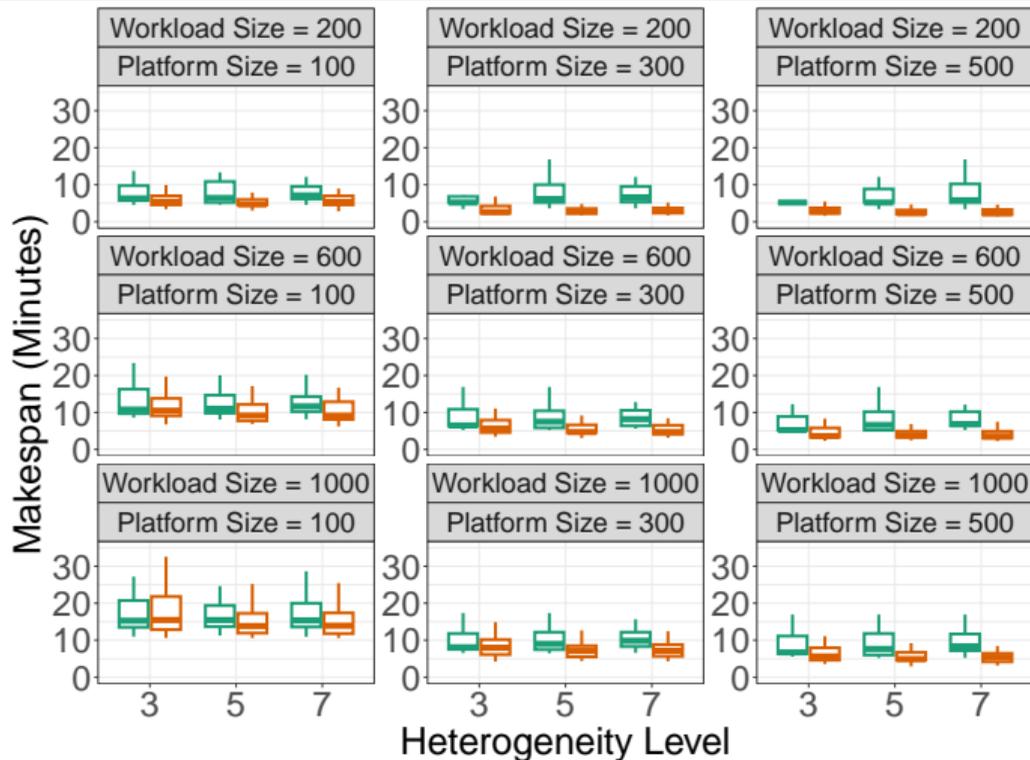


MIAI
Grenoble Alpes



PHYSICS

Experimental results: makespan



Scheduling Policy  K8S ImageLocality  FOA

FOA's Linear Program - Assumptions and Variables

Notation	Description
M	Number of machines
N	Number of scheduled functions
K	Number of container environments
c_{ij}	Cost of the j -th function on the i -th machine
p_{ij}	Execution time of the j -th function on the i -th machine
d_{ik}	Cost of the k -th environment on the i -th machine
b_{ik}	Execution time of the k -th environment on i -th machine
$env[j]$	Environment id of the j -th function
x_{ij}	Placement of the j -th function on the i -th machine
e_{ik}	Placement of the k -th environment on the i -th machine

Assumptions:

- 1) Functions and environments are independent and known in advance;
- 2) Their cost and processing time on each machine are known;
- 3) The dependency between functions and environments is known.

Table 1: FOA's list of notation and descriptions.

FOA's Linear Program

The Linear Program minimizes the cost under a makespan constraint of value T

$$\text{Min } \sum_{i=1}^M \sum_{j=1}^N c_{ij} \times x_{ij} + \sum_{i=1}^M \sum_{k=1}^K d_{ik} \times e_{ik} \quad (1)$$

$$\forall i \leq M, \quad \sum_{j=1}^N p_{ij} \times x_{ij} + \sum_{k=1}^K b_{ik} \times e_{ik} \leq T \quad (2)$$

$$\forall j \leq N, \quad \sum_{i=1}^M x_{ij} \geq 1 \quad (3)$$

$$\forall i \leq M, \forall j \leq N, \quad x_{ij} \leq e_{i,env[j]} \quad (4)$$

$$\forall i \leq M, \forall k \leq K, \quad \sum_{j=1}^N x[i][j] \leq N_{ik} \times e[i][k] \quad (5)$$

$$\forall i \leq M, \forall k \leq K, \quad e_{ik} \leq 1 \quad (6)$$

Exp. results: System utilization and makespan

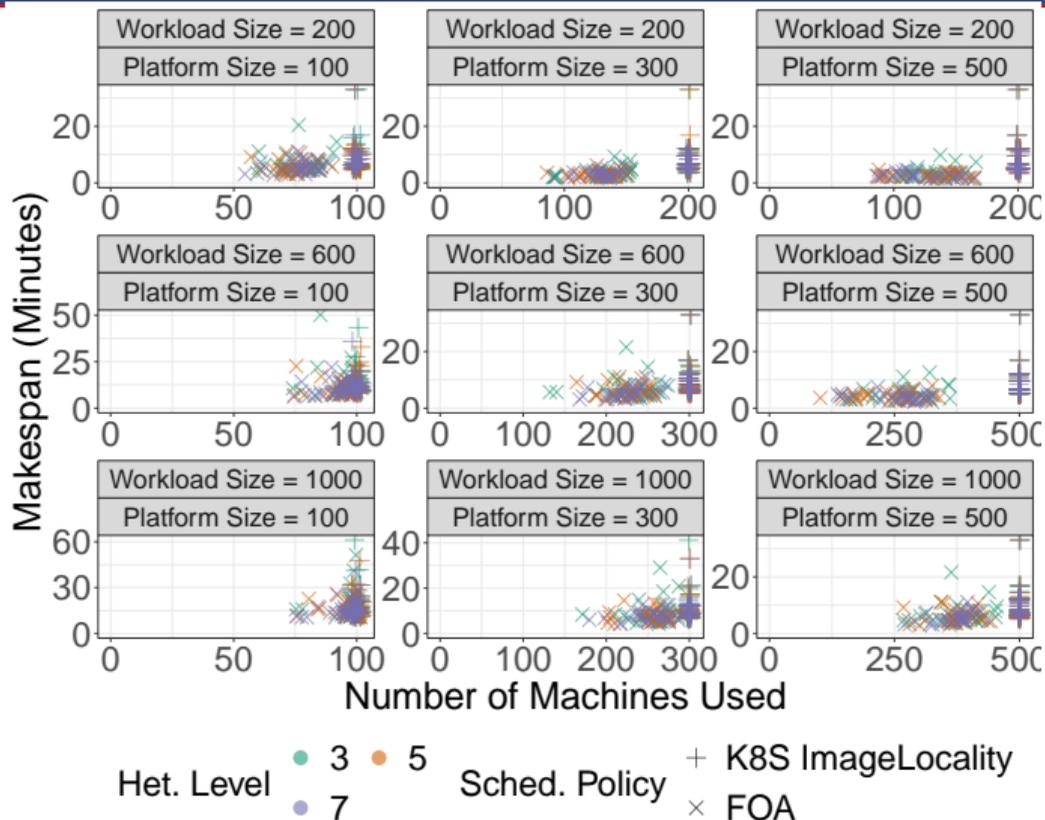


Figure 10: The makespan (y-axis) against the number of machines used (x-axis).

Exp. results: System utilization and data transfers

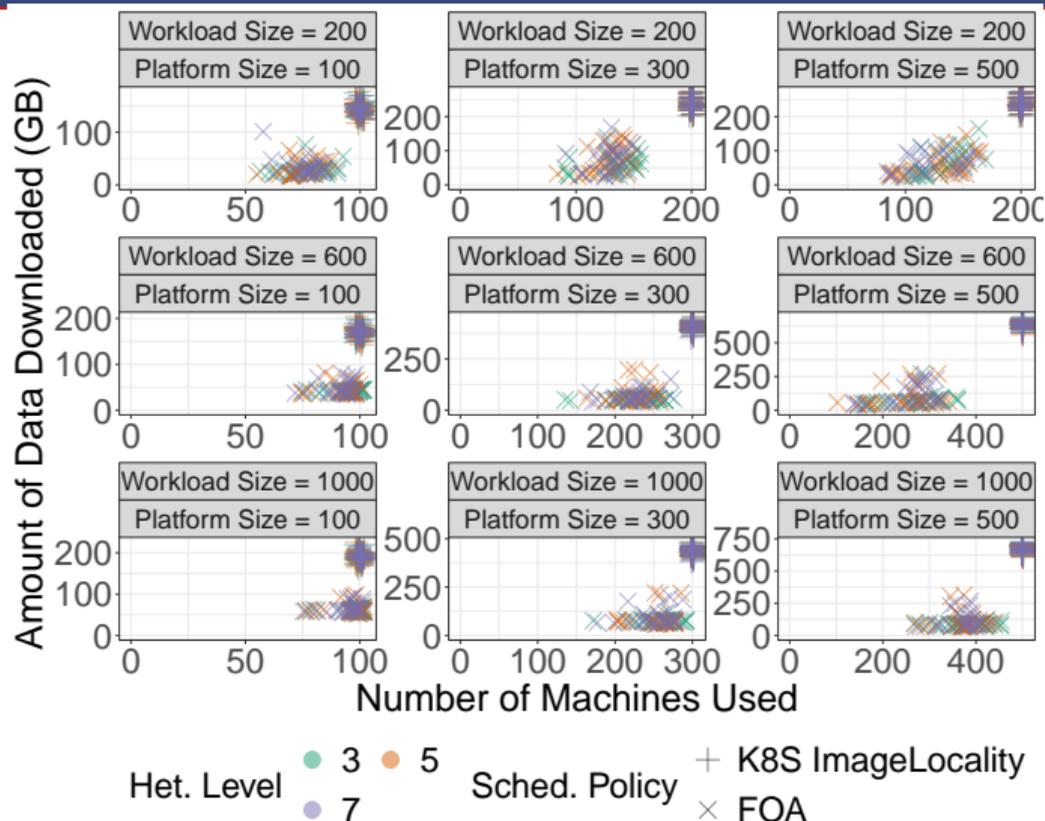


Figure 11: The amount of data downloaded (y-axis) against the number of machines used (x-axis).