eSAAM 2023

on Cloud to Edge Continuum

FlexConnect: Mobile Computational Offloading

Alexander Droob, Daniel Morratz, Frederik Langkilde Jakobsen, Jacob Carstensen, Magnus Mathiesen, Rune Bohnstedt, Michele Albano (Aalborg University)

Sergio Moreschini (Tampere University)

Davide Taibi (University of Oulu)

Oct. 17, 2023

Ludwigsburg, Germany

Context

- Makerspaces provide AI-based computational support
- Aalborg requires makerspace to
  - Shut-down EVERYTHING when closed
  - Delete data before closing

Problem

*"How can we **provide computational power** to its users at **various locations without loss of progress** using **computational offloading?**"*

Goal

*How to develop an edge-based system that allows computational tasks to be executed and **interrupted/resumed** from edge nodes **without loss of progress**.*

Requirements – User and App

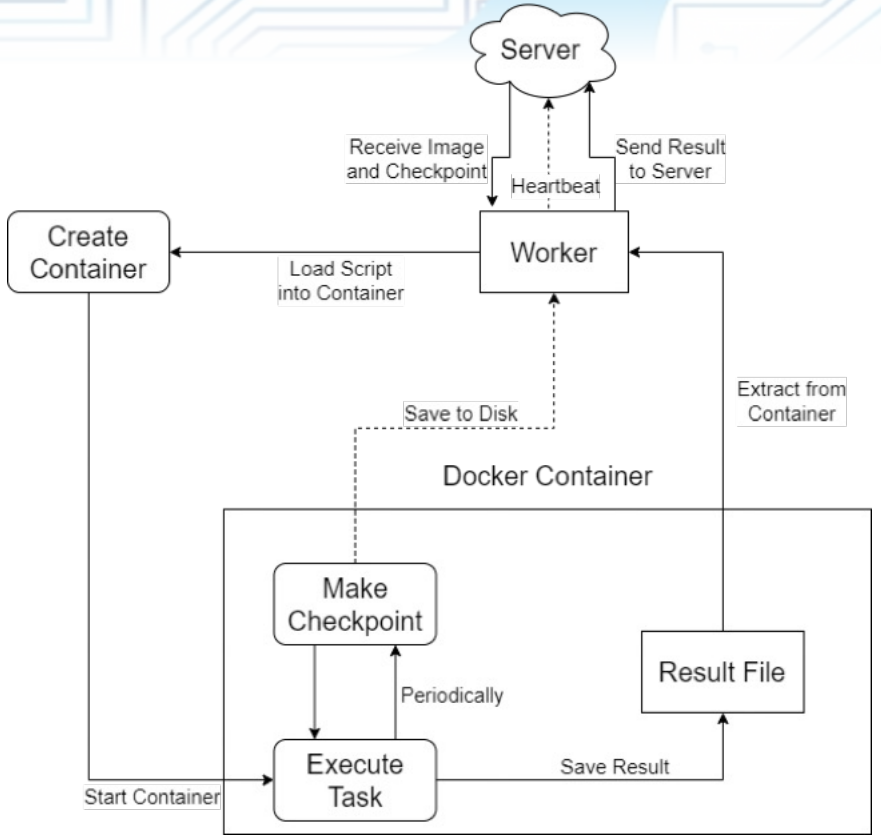
- A user can:
  - Upload a job to the system
  - Download the result from their job
  - Authenticate
  - Can see their job history
  - See the status of their job
  - Can pause and resume a running job
  - See the provider locations,
  - See the distance to provider locations.
- The application implements a user interface.

Requirements - System

- The system and application are connected wirelessly
- The system can:
  - Detect nearby users
  - Has a queuing system for jobs
  - Can handle data-intensive jobs
  - Will notify a user when a job is complete
  - The system must be also secure

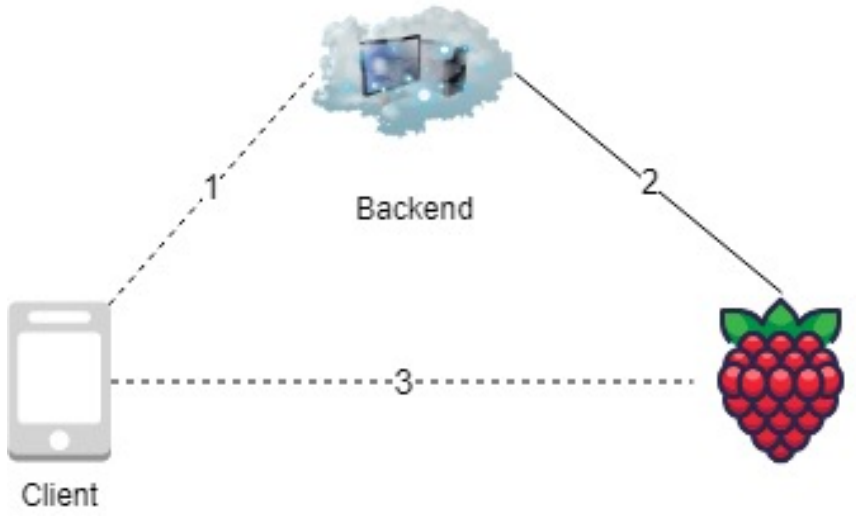
Proposed Solution - Workrs

- **Once the Job is received the worker creates the container**
- **Execute the tasks while:**
  - Creating checkpoints
  - Send heartbeat to the orchestrator
- **Once the process is done send the result to the orchestrator, which sends it to the server.**



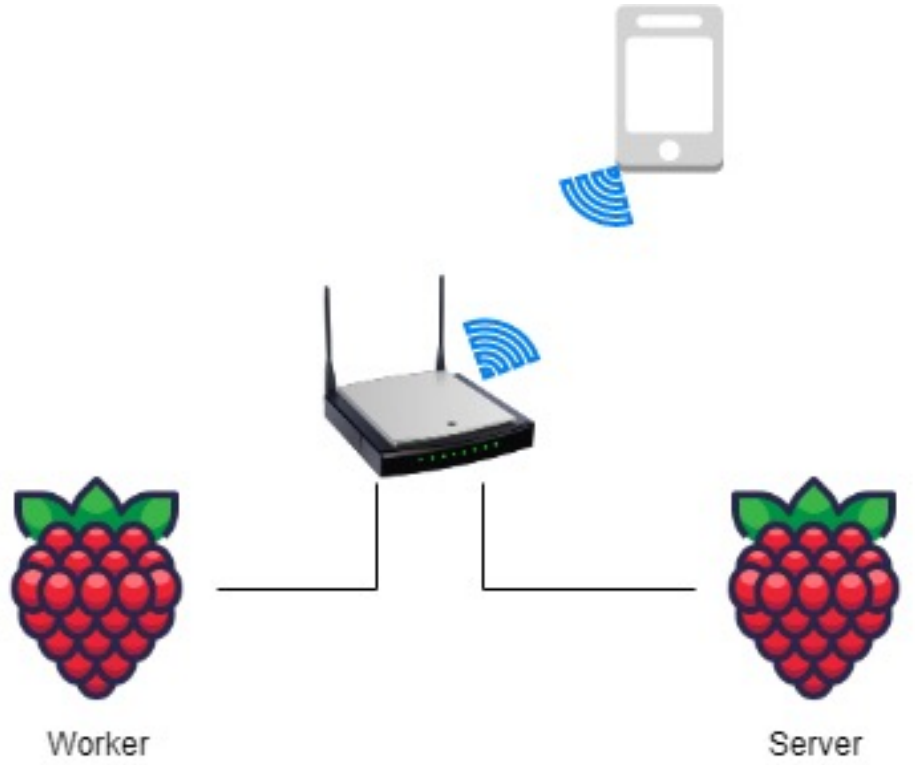
Proposed Solution - Communication

1. Initial connection to the backend: used for business logic such as user information, jobs overview, and starting or stopping jobs
2. Workers' connection to the backend: used for both logic and data flow connection
3. Direct connection from the users' mobile device to the worker: used for data flow, internet to the users' device.



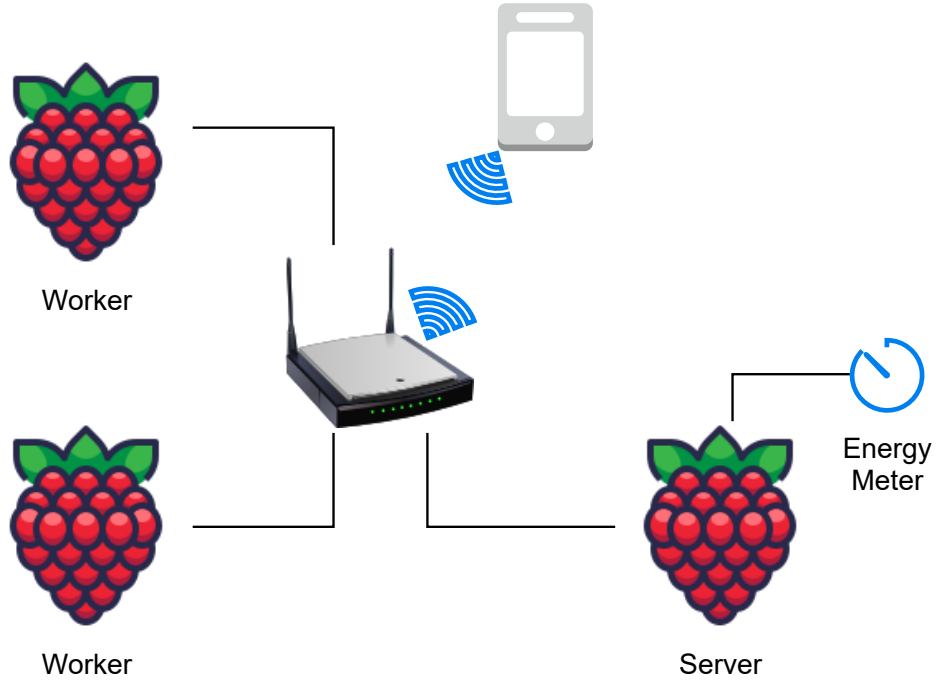
Implementation

- Physical Setup
  - 2x Raspberry Pi 4b (4GB Ram)
  - 1x Internet Router/Network Switch
  - 1x Mobile Android Device
- Raspberry Pi worker unit and FTP server
- Raspberry Pi backend and message broker
- Mobile device client unit
- In the prototype implemented instead of using a WebDAV connection, an FTP solution has been implemented.

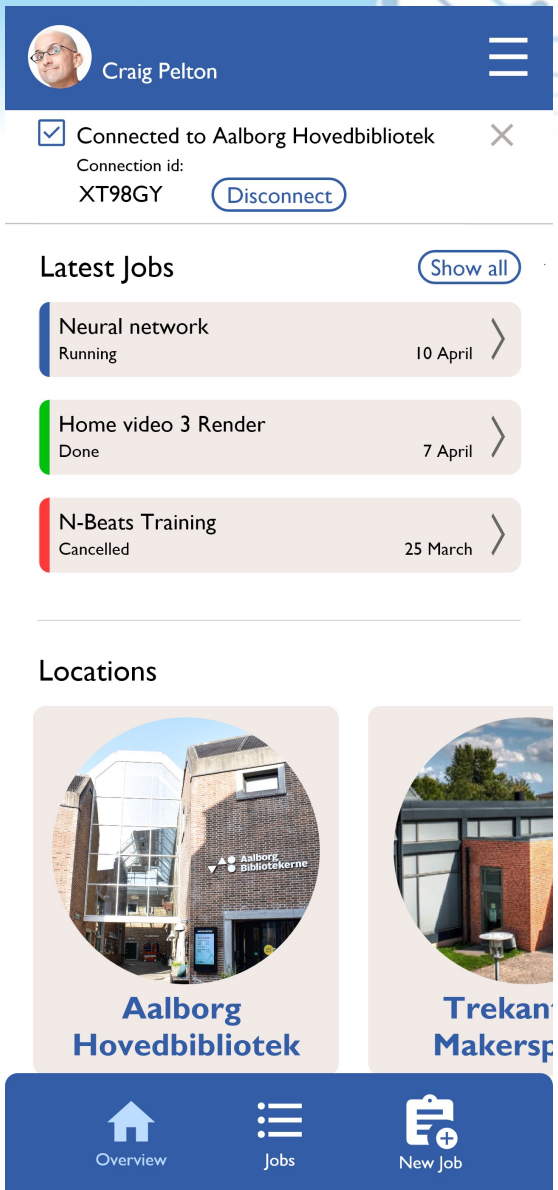


Testing Prototype

- Physical Setup
  - 2x Raspberry Pi 4b as workers
  - 1x Raspberry Pi 4b as Backend and Coordinator
  - 1x Internet Router/Network Switch
  - 1x Mobile Android Device
  - 1x Energy meter to measure energy consumption of checkpointing on the system.



The application Prototype



Testing

- **Test script** which multiplied matrices, simulating tasks performed by the worker unit.
- A **calculated failure time** was sent to the worker after which the worker would emulate a hardware failure.
  - Calculated through a semi-random calculation based on a given failure rate μ.

	Failure rate μ	Number of checkpoints
1.	0.0001	0
2.	0.0001	15
3.	0.0005	0
4.	0.0005	1
5.	0.0005	15
6.	0.003	0
7.	0.003	5
8.	0.003	15
9.	0.0131	0
10.	0.0131	11
11.	0.0131	15

Table 1: Test cases for the automatic testing system.

Conclusion

- Design and implementation of a mobile computational offloading system
- Ability to checkpoint user computations and resume progress at a later point.
- Users can divide their computational work into smaller chunks.
- MVP prototype created.

Future Work

- Implementation of missing features:
  - Queuing system
  - Increased security (Encryption)
  - Docker validation
  - Systematic testing process (ex. Stress test)
- GPU support
- Support for more workload types
- Support for Federated Learning

eSAAM 2023

on Cloud to Edge Continuum

Thank You

Sponsored by:

Organized by: