

UNIMORE

UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



Microservice performance in Container- and Function-as-a-Service architectures

Claudia Canali, Riccardo Lancellotti, Pietro Pedroni
University of Modena and Reggio Emilia

Motivation

- Function-as-a-Service model
 - Serverless approach
- Widely used in Cloud architectures
 - Highly efficient
 - Cost-effective
- Critical points
 - **Limited control** of infrastructure (Resource usage)
 - **Variable performance** (cold start)
- Contribution:
Analysis of state-of-the-art solutions



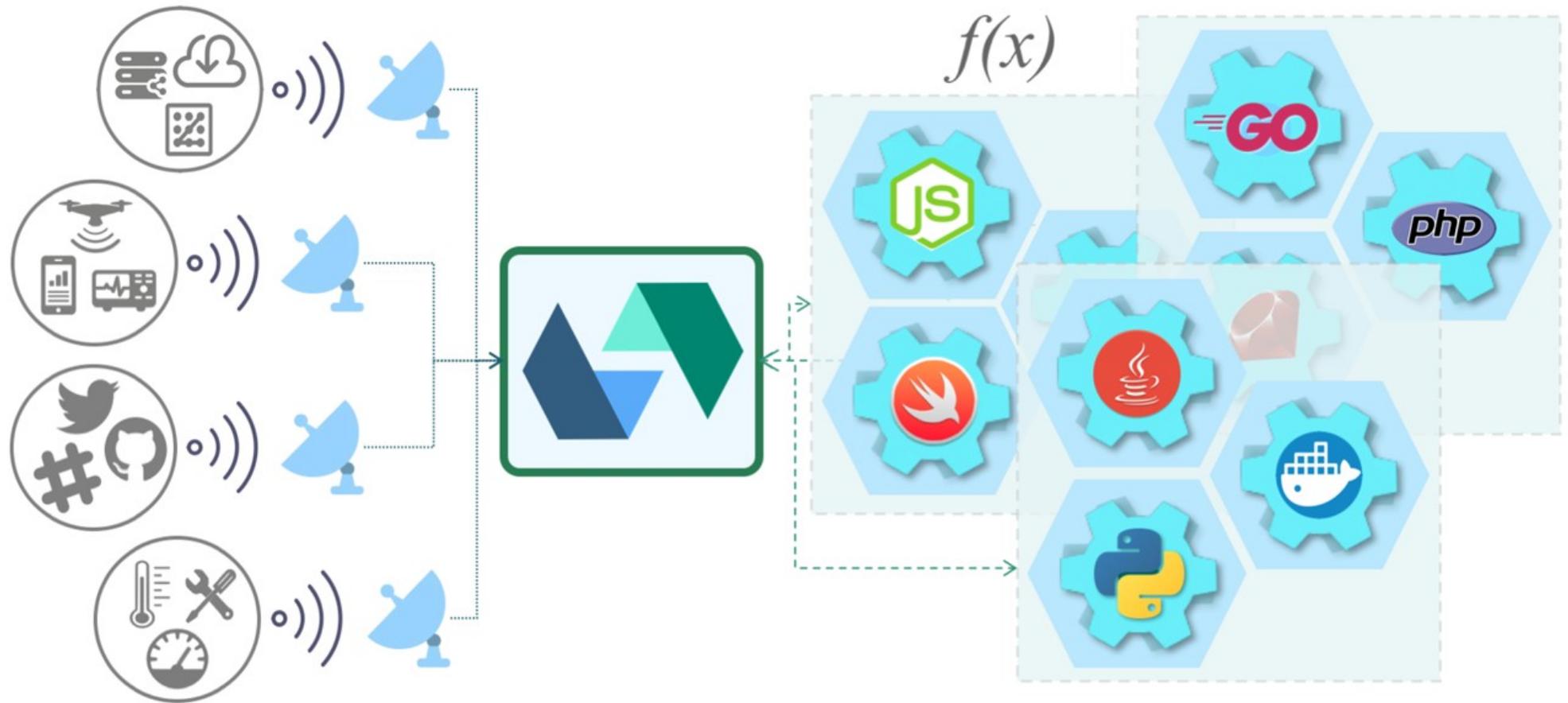
OPENFAAS



FaaS architecture

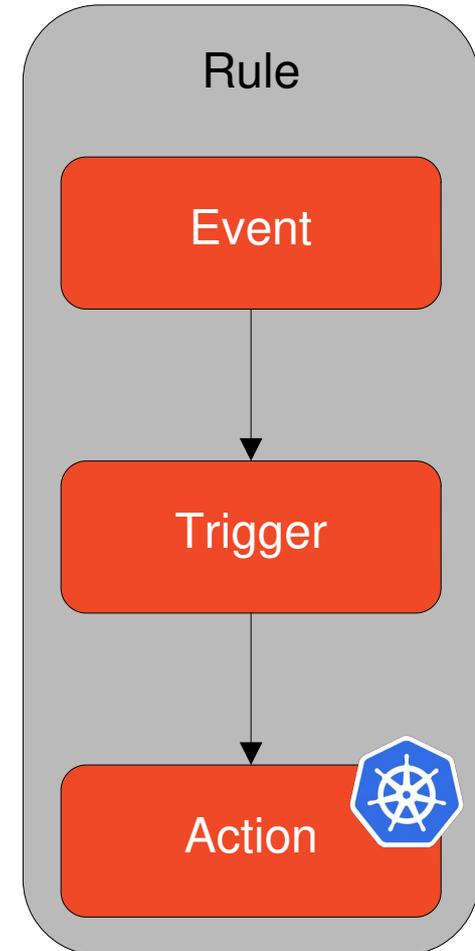
UNIMORE

UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



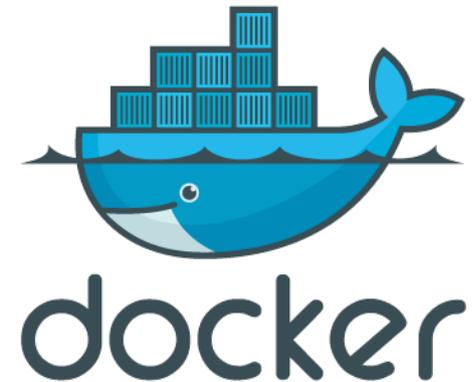
FaaS architecture

- Main concepts
 - Event (MQTT message, Web request, ...)
 - Trigger
 - Action (Function)
- Function runs in containers
- Container lifecycle managed by FaaS
 - Risk of cold-start
 - Must start new container when trigger is activated



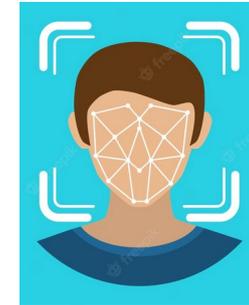
Container-based architecture

- Container always on-line
- Potential waste of resources
 - RAM, CPU
- Performance more stable
- Scalability provided by container
 - Docker → no scalability
 - Kubernetes → pluggable policies

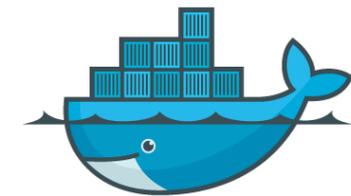


Experimental setup

- Two functions
 - Face recognition
 - Image conversion (transcoding)
- REST service
 - Python + Flask
- Two architectures
- FaaS → Open Whisk
 - Powerful and flexible tool
- Container-based → Docker
 - Focus on simplicity



Flask



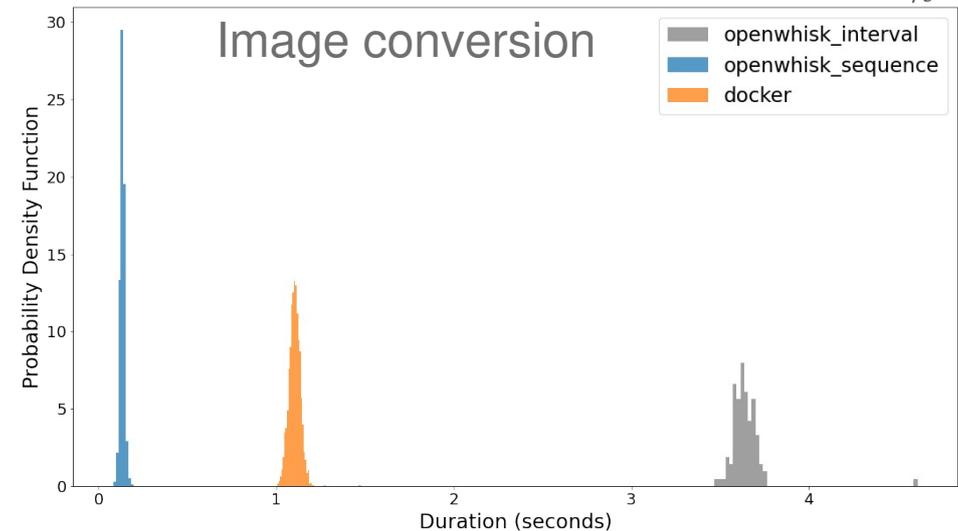
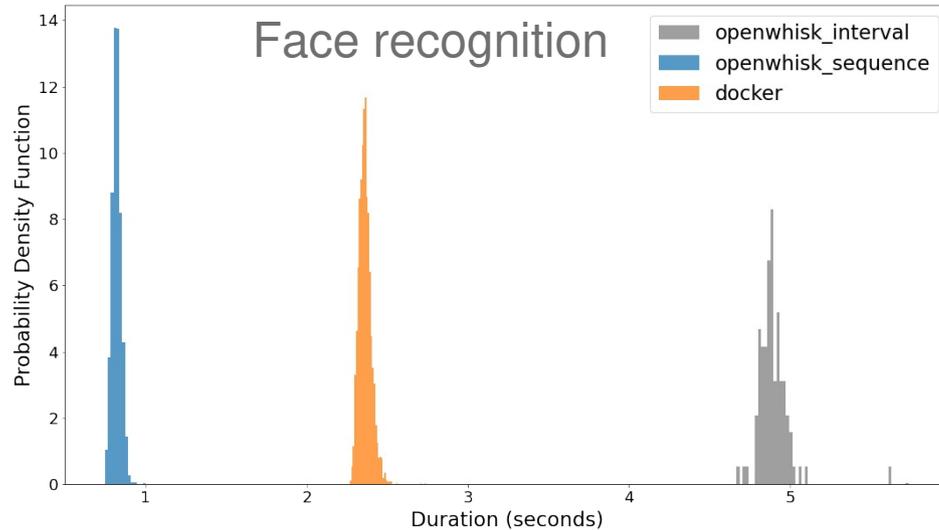
docker

Experimental setup

- Two invocation policies
- **Sequence** of invocations
 - No cold start, service always warm
- **Interval** between invocations
 - Leave time to stop containers
 - Cold start

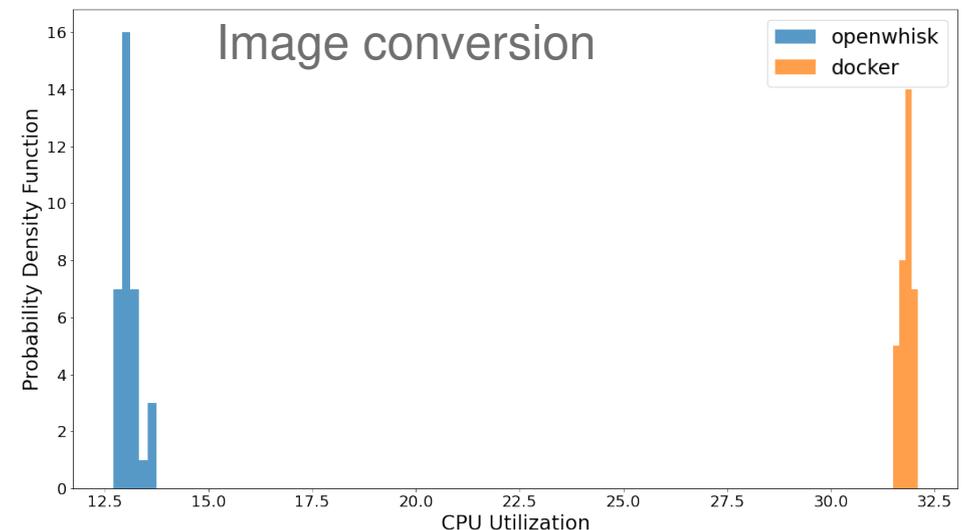
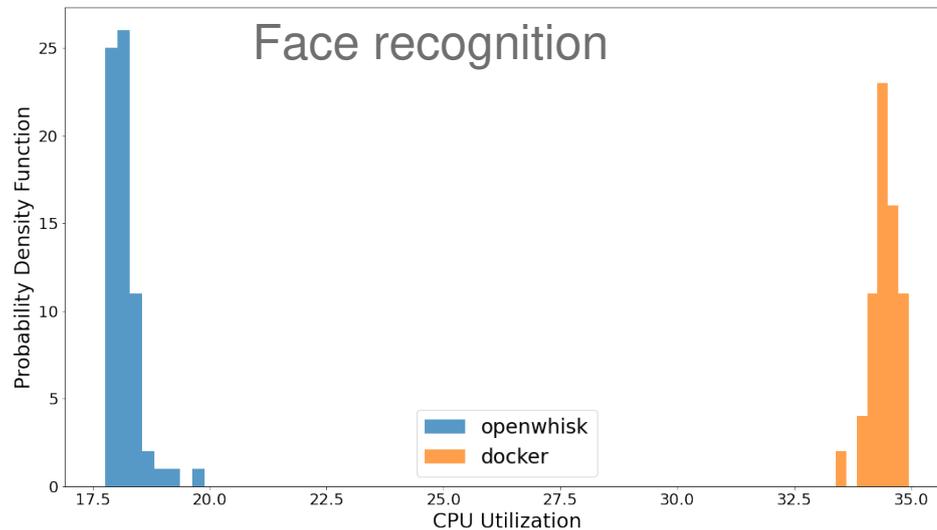


Experimental results



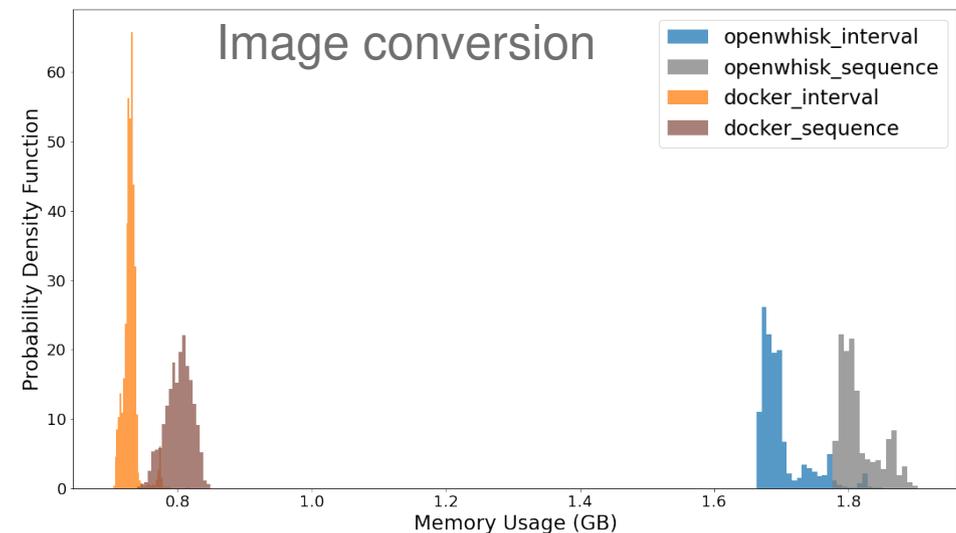
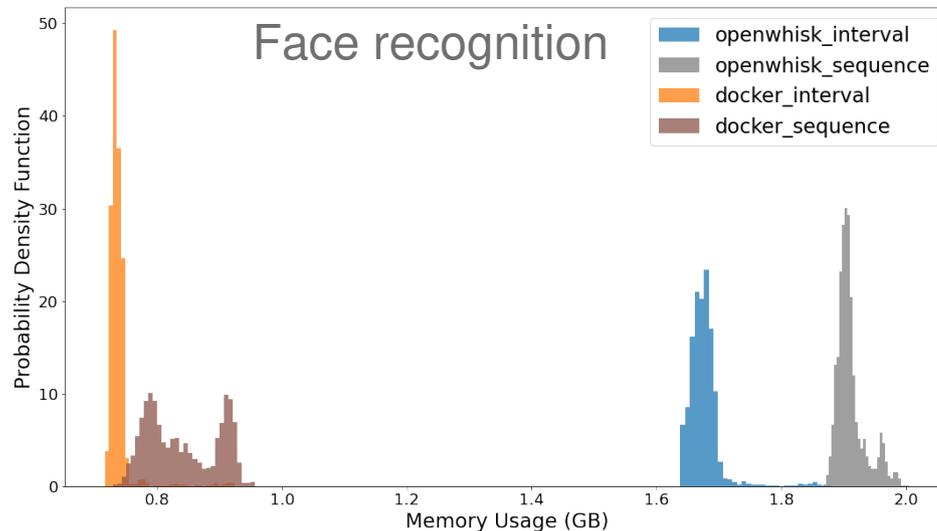
- Docker
 - No difference between cold and warm start
- FaaS
 - Major difference between cold and warm start
 - Cold start → very slow
 - Warm start → faster than docker (function prepared)

Experimental results



- CPU utilization (only for warm start)
- FaaS is more efficient than Docker
 - Internal optimizations
 - Data caching

Experimental results



- Memory utilization

- Larger footprint of FaaS approach compared to Container (more software layers, data caches)
- Impact of memory cache and garbage-collection (more memory for sequential invocation)

Conclusion

- Performance analysis of FaaS architectures
- Comparison with container-only approach
- Highly optimized response time for warm start
- Significant overhead for cold start
- Significantly higher memory footprint

UNIMORE

UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



Microservice performance in Container- and Function-as-a-Service architectures

Claudia Canali, Riccardo Lancellotti, Pietro Pedroni
University of Modena and Reggio Emilia