CLAIM: a Lightweight Approach to Identify Microservices in Dockerized Environments

Kevin Maggi
University of Florence

Roberto Verdecchia
University of Florence

Leonardo Scommegna
University of Florence

Enrico Vicario
University of Florence

UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

{ST LAB} Software Technologies Laboratory

Introduction

➔ **Context**:

Mining studies in the field of Empirical Software Engineering

➔ **Motivation**:

Identifying microservices in a Microservices Architecture

➔ **Needs**:

Lightweightness in execution time and resources consumption
Language independence

Background

State of Art:

- Baresi *et al.*[1]: **static approach** *via* parsing Docker *compose* files

Related Work: e.g. μMINER, MicroArt, etc.

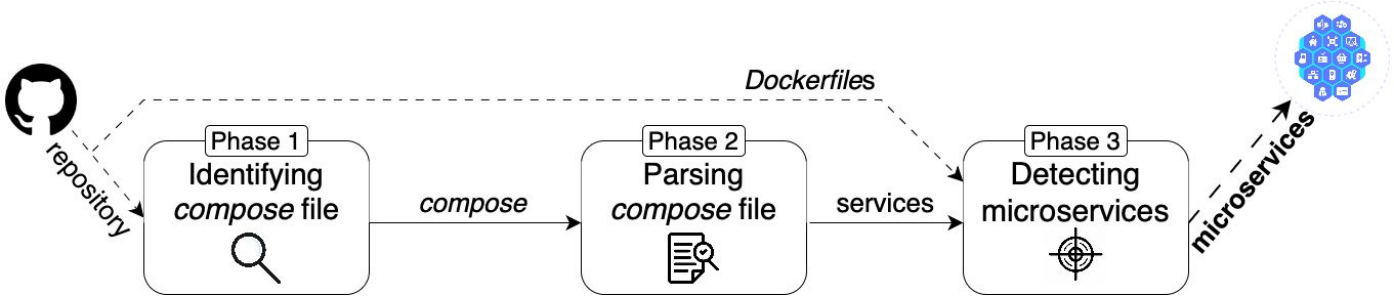
- recovery of the **entire architecture** (also architectural components)
- **dynamic** or **hybrid static-dynamic** approach → excessive effort
- **semi-automatic** approach (human intervention required)

CLAIM:

- **Convention-based empirical rules**
- **Refinement** and **extension** of Baresi *et al.*[1]
  - Identification and parsing of Docker *compose* files
  - Additional use of *Dockerfile* to improve effectiveness

[1] *"Microservice Architecture Practices and Experience: a Focused Look on Docker Configuration Files"* - L. Baresi, G. Quattrocchi, D. A. Tamburri (2022)

Approach



Phase 1

1. *compose* files collection
2. path-based filtering
3. path-based order
4. filename-based order
5. selection

Phase 2

1. variable interpolation
2. recursive inclusion resolution
3. inheritance reconstruction
4. services data extraction

Phase 3

1. *Dockerfiles* collection
2. extension-based filtering
3. path-based filtering
4. service-*Dockerfile* match
5. *Dockerfiles* checks

Research Questions

RQ_1 *What is the **effectiveness** of CLAIM in terms of microservice identification?*

Metrics:
- precision
- accuracy
- recall

RQ_2 *What is the **efficiency** of CLAIM in terms of execution time and memory consumption?*

Metrics:
- execution time per commit
- execution time per repository
- memory consumption per repository

Experiment

Experimental objects:

20 open source MSA repositories (13k commits, 1.7M SLOC, 160 microservices)

Ground truth:

6 *a priori* defined microservice Ground Truth
14 manually defined microservice Ground Truth

Experiment execution:

RQ_1	***compose* file selection**	(commit-wise)
RQ_1	**microservices identification**	(commit-wise)
RQ_2	**execution time and resource profiling**	(repository-wise)

Comparison:

Baresi *et al.*[1] tool

[1] *"Microservice Architecture Practices and Experience: a Focused Look on Docker Configuration Files"* - L. Baresi, G. Quattrocchi, D. A. Tamburri (2022)

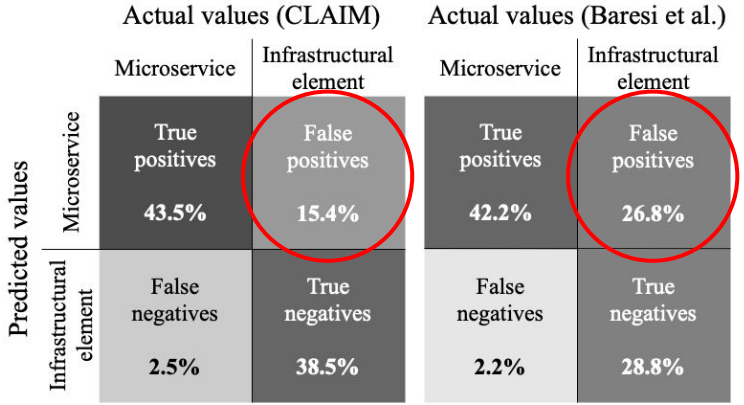
Results (RQ_1 - effectiveness)

compose file selection

	CLAIM	Baresi *et al.*[1]
Success rate	**99.2%**	94.8%

Microservices identification

		CLAIM	Baresi *et al.*[1]
Accuracy	(TP+TN)/(TP+TN+FP+FN)	**82%**	71%
Precision	TP/(TP+FP)	**73.8%**	61.2%
Recall	TP/(TP+FN)	94.5%	**95.0%**

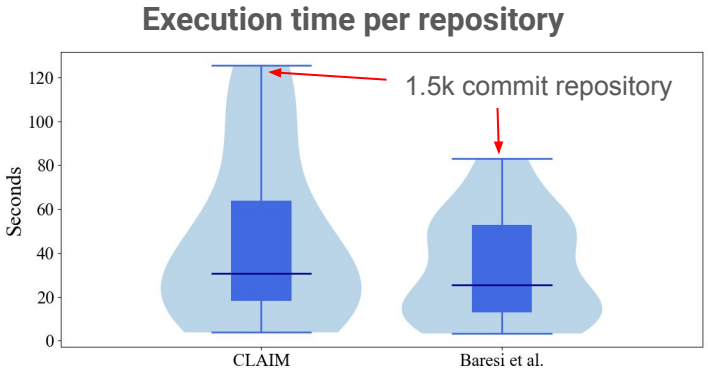


[1] *"Microservice Architecture Practices and Experience: a Focused Look on Docker Configuration Files"* - L. Baresi, G. Quattrocchi, D. A. Tamburri (2022)

Results (RQ_2 - efficiency)

Execution time

Execution time per commit	CLAIM	Baresi *et al.*[1]
Best case scenario	23 *ms*	**18 *ms***
Worst case scenario	266 *ms*	**216 *ms***
Median	61 *ms*	**38 *ms***



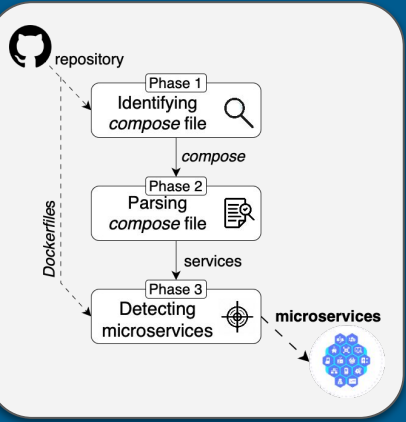
Memory consumption

	CLAIM	Baresi *et al.*[1]
Median	**20 MB**	30 MB

[1] *"Microservice Architecture Practices and Experience: a Focused Look on Docker Configuration Files"* - L. Baresi, G. Quattrocchi, D. A. Tamburri (2022)

CLAIM: a Lightweight Approach to Identify Microservices in Dockerized Environments

Summary



RQ_1: *What is the **effectiveness** of CLAIM in terms of microservice identification?*

RQ_2: *What is the **efficiency** of CLAIM in terms of execution time and memory consumption?*

Metric	Value
Precision	82%
Accuracy	73.8%
Recall	84.5%

Metric	Median value
Execution time (per commit)	61 *ms*
Memory consumption (per repository)	20 MB

- Valid and scalable option for microservices identification
- Stepping stone towards developing better techniques
- Conducting comparison also against dynamic approaches

Thank you, and… it's Q&A time!

Kevin Maggi
University of Florence
kevin.maggi@unifi.it

UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

https://github.com/STLab-UniFI/CLAIM_rep-pkg