

**École polytechnique de Louvain**

# **Improving the performance and the scalability of INGINIOUS**

Author: **Guillaume EVERARTS DE VELP**  
Supervisor: **Ramin SADRE**  
Readers: **Oliver BONAVENTURE, Anthony GÉGO**  
Academic year 2019–2020  
Master [120] in Computer Science

# Acknowledgements

Guillaume Everarts de Velp, 2020

# Abstract

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	INGInious . . . . .	4
1.1.1	Architecture . . . . .	4
1.1.2	Key features . . . . .	5
1.1.3	Bottlenecks . . . . .	6
1.2	Scaling . . . . .	6
1.2.1	Vertical scaling . . . . .	6
1.2.2	Horizontal scaling . . . . .	6
1.3	Intentions . . . . .	6
<b>2</b>	<b>State of the art</b>	<b>7</b>
2.1	Virtualization vs Containerization . . . . .	7
2.2	Containerization solutions . . . . .	7
2.2.1	Docker . . . . .	7
2.2.2	Podman . . . . .	7
2.2.3	runC . . . . .	7
2.2.4	SAND . . . . .	7
2.2.5	SOCK . . . . .	7
2.3	Virtualization solutions . . . . .	7
2.3.1	Firecracker . . . . .	7
2.3.2	Kata Containers . . . . .	7
2.3.3	LightVM . . . . .	7
<b>3</b>	<b>Measurements</b>	<b>8</b>
<b>4</b>	<b>Results</b>	<b>9</b>
<b>5</b>	<b>Conclusion</b>	<b>10</b>

# Chapter 1

## Introduction

### 1.1 INGINious

INGInious is a web platform developped by the UCL. It is a tool for automatic correction of programs written by students. It currently relies on Docker containers to provide a good isolation between the machine hosting the site and the execution of the student's codes. So that a problem in a program submitted by a student couldn't have any impact on the platform. Docker also allow to manage the resources granted to each code execution. For now INGINious can meet the demand and provide honest performances and responsiveness. But looking at the growing usage of the platform, we might soon come to a point where we reach the limits of the current implementation.

#### 1.1.1 Architecture

INGInious counts four main components:

1. The front-end: the website with which each student interacts when submitting a task.
2. The back-end: a queue of all the tasks that need to be graded.
3. The docker agent: responsible for the container assignment to the pending tasks when resources are available.
4. The docker containers: one for the student code and one for the teacher tests evaluating the student's code behaviour.

The journey of a task submitted on INGINious is represented on Figure 1.1.

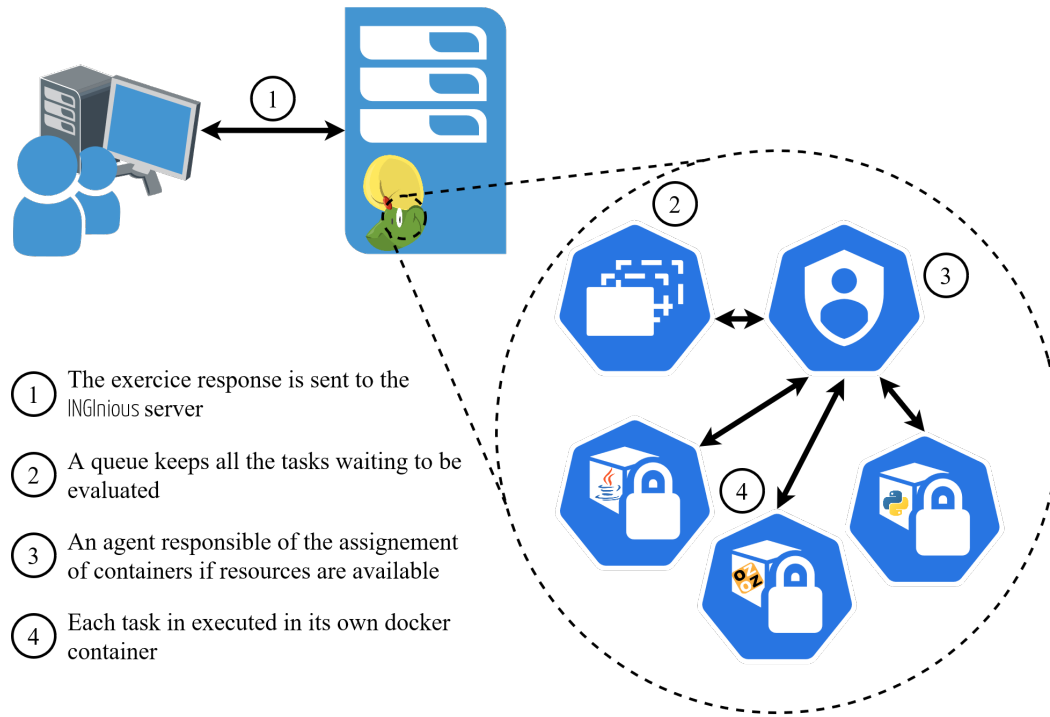


Figure 1.1: INGINIOUS global architecture

### 1.1.2 Key features

The key features of INGINIOUS, that allow it to meet the requirement of a code grading platform are the following:

- Isolation between the student's code and the platform.
- Resource limitation (CPU, RAM, Network) for the student's code execution.
- Modularity and versatility regarding the tasks that INGINIOUS can correct. Multiple programming language are supported, and new ones can easily be added.

### **1.1.3 Bottlenecks**

## **1.2 Scaling**

Currently, the resources provided to INGINious vary depending on the load that the platform is expected to be facing. Typically, when the grading of an exam is done by INGINious, the platform is scaled up, and during the holidays it is scaled down. In general, the scaling can be done in two directions, horizontally, or vertically.

### **1.2.1 Vertical scaling**

### **1.2.2 Horizontal scaling**

## **1.3 Intentions**

# Chapter 2

## State of the art

### 2.1 Virtualization vs Containerization

### 2.2 Containerization solutions

#### 2.2.1 Docker

#### 2.2.2 Podman

#### 2.2.3 runC

#### 2.2.4 SAND

#### 2.2.5 SOCK

### 2.3 Virtualization solutions

#### 2.3.1 Firecracker

#### 2.3.2 Kata Containers

#### 2.3.3 LightVM



## Chapter 3

# Measurements

# Chapter 4

## Results

# Chapter 5

## Conclusion

# Bibliography

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | [www.uclouvain.be/epl](http://www.uclouvain.be/epl)