AUTOMATIC GRADING OF PROGRAMMING EXERCISES USING THE INGINIOUS PLATFORM

EMOOCs 2015

Guillaume Derval
Anthony Gégo
Benjamin Frantzen
Pierre Reinbold
Peter Van Roy

May 19, 2015

Université catholique de Louvain

TABLE OF CONTENTS

MOTIVATION

CODE GRADING

```c
#include <stdio.h>
void f(int *i, long *l)
{
  printf("1. v=%ld\n", *l);
  *i = 11;
  printf("2. v=%ld\n", *l);
}
int main()
{
  long a = 10;
  f((int *) &a, &a);
  printf("3. v=%ld\n", a);
  return 0;
}
```

Code from the article "I do not know C" on Kukuruku.co

`http://kukuruku.co/hub/programming/i-do-not-know-c`

HOW TO GRADE CODE

Different ways:

- Ask for a report, and read it (manually)
- Read the code (manually)
- Run the code

HOW TO GRADE CODE

Different ways:

- Ask for a report, and read it (manually)
- Read the code (manually)
- Run the code (automatically, unit tests)

GETTING FEEDBACK

- Most of the time, students only get a grade...
- Does it allow them to improve?

GETTING FEEDBACK

- Most of the time, students only get a grade…
- Does it allow them to improve?
- Unit tests $\Rightarrow$ fine-grained feedback

REQUISITES

- Multiple courses(on-site & MOOCs)
- Usable for other types of tasks/courses
- Easy and fast grading
- Custom feedback
- Any language (even exotic ones)
- Security & scalability

EXISTING AUTOMATED CODE GRADING SOLUTIONS

- In EDX: only Python
- Lots of software for ACM-ICPC-like contests
- Some tools focused on one language
- Pythia

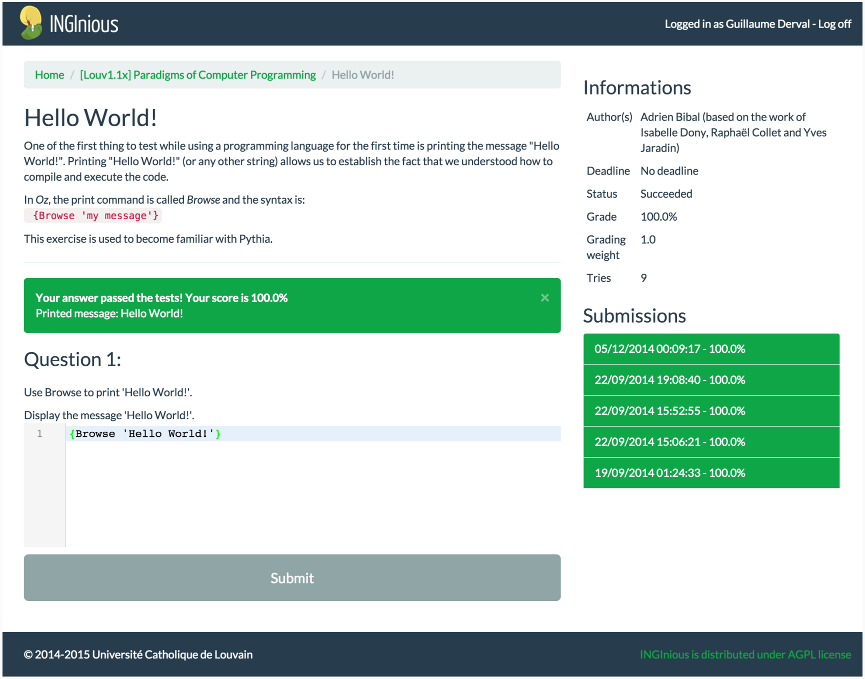
INGINIOUS

MAIN FEATURES

- Runs untrusted code
- Security: code is jailed
- Customisable limits
  - CPU time
  - Wall time
  - Memory
- Customisable environments
- Simple web interface

CUSTOMISABLE

- Runs containers
  - Any language
  - Any software
  - Simple to create
- Custom scripts
  - Made by assistants
  - Custom tests
  - Custom feedback
- Plugins



INGInious

Logged in as Guillaume Derval - Log off

Home / [Louv1.1x] Paradigms of Computer Programming / Hello World!

Hello World!

One of the first thing to test while using a programming language for the first time is printing the message "Hello World!". Printing "Hello World!" (or any other string) allows us to establish the fact that we understood how to compile and execute the code.

In *Oz*, the print command is called *Browse* and the syntax is: `{Browse 'my message'}`

This exercise is used to become familiar with Pythia.

Your answer passed the tests! Your score is 100.0%
Printed message: Hello World!

Question 1:

Use Browse to print 'Hello World!'.

Display the message 'Hello World!'.

```
{Browse 'Hello World!'}
```

Submit

Informations

Author(s)	Adrien Bibal (based on the work of Isabelle Dony, Raphaël Collet and Yves Jaradin)
Deadline	No deadline
Status	Succeeded
Grade	100.0%
Grading weight	1.0
Tries	9

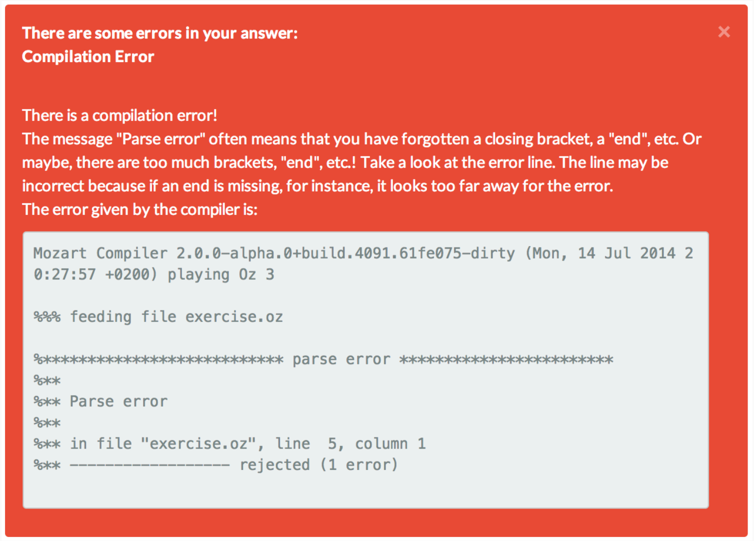
Submissions

05/12/2014 00:09:17 - 100.0%
22/09/2014 19:08:40 - 100.0%
22/09/2014 15:52:55 - 100.0%
22/09/2014 15:06:21 - 100.0%
19/09/2014 01:24:33 - 100.0%

© 2014-2015 Université Catholique de Louvain

INGInious is distributed under AGPL license

CUSTOM FEEDBACK



There are some errors in your answer:
Compilation Error

There is a compilation error!
The message "Parse error" often means that you have forgotten a closing bracket, a "end", etc. Or maybe, there are too much brackets, "end", etc.! Take a look at the error line. The line may be incorrect because if an end is missing, for instance, it looks too far away for the error.
The error given by the compiler is:

```
Mozart Compiler 2.0.0-alpha.0+build.4091.61fe075-dirty (Mon, 14 Jul 2014 2
0:27:57 +0200) playing Oz 3

%%% feeding file exercise.oz

%**************************** parse error ************************
%**
%** Parse error
%**
%** in file "exercise.oz", line  5, column 1
%** ------------------ rejected (1 error)
```

OTHER USEFUL FEATURES

- Syntax highlighting
- File upload
- Multiple choice questions
- Integration with edX
- Submissions from external scripts (REST API)
- INGInious studio
- ...

- Binary or integer grades
- Test suite
- Stats (exportable!)
- Scalable
- Tasks with long running time
- Compatible with Pythia v0 and v1
- ...

INGINIOUS IS A FOSS

- Free
- Open-source
- AGPL 3 license
- Sources on GitHub

`http://www.github.com/UCL-INGI/INGInious`

INGINIOUS IS IN PRODUCTION USE

Used in many courses at UCL

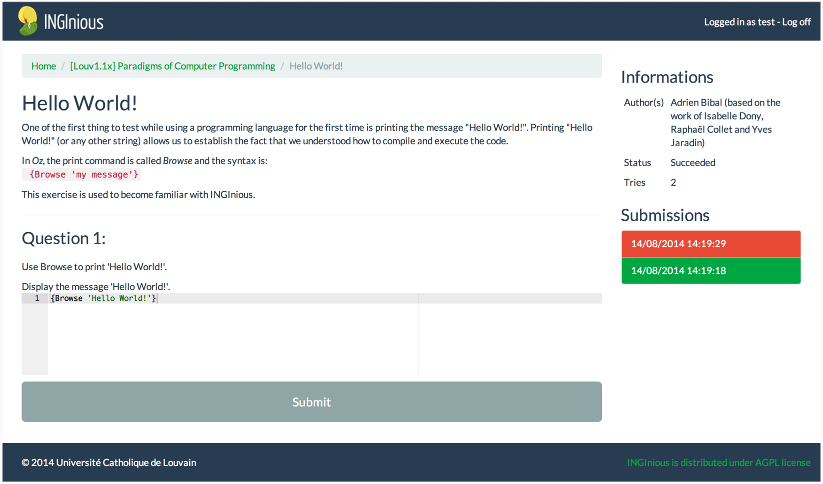
- On-site courses
  - `LFSAB1401`: Computer science 1
  - `LSINF1103`: Introduction to algorithmic
  - `LSINF1252`: Computer systems 1
  - `LINGI2132`: Language and translators
  - `LINGI2145`: Cloud computing
  - `LINGI2365`: Constraint programming
  - `LINGI2261`: Artificial intelligence
  - `LINGI2347`: Computer systems security
  - more courses next year
- MOOCs (@edX)
  - `Louv1.1x`: Paradigms of Computer Programming – Fundamentals
  - `Louv1.2x`: Paradigms of Computer Programming – Abstraction and Concurrency
  - These MOOCs are also given to on-campus students (`LFSAB1402`: Computer science 2)

USAGE

ORGANISATION

- Student see courses
- With tasks inside (unit of grading)
- With sub-questions
- Can read the task, answer
- Immediately get feedback

SUBMISSION... FROM THE FRONTEND



INGInious

Logged in as test - Log off

Home / [Louv1.1x] Paradigms of Computer Programming / Hello World!

Hello World!

One of the first thing to test while using a programming language for the first time is printing the message "Hello World!". Printing "Hello World!" (or any other string) allows us to establish the fact that we understood how to compile and execute the code.

In *Oz*, the print command is called *Browse* and the syntax is: `{Browse 'my message'}`

This exercise is used to become familiar with INGInious.

Question 1:

Use Browse to print 'Hello World!'.

Display the message 'Hello World!'.

```
{Browse 'Hello World!'}
```

Submit

Informations

Author(s)	Adrien Bibal (based on the work of Isabelle Dony, Raphaël Collet and Yves Jaradin)
Status	Succeeded
Tries	2

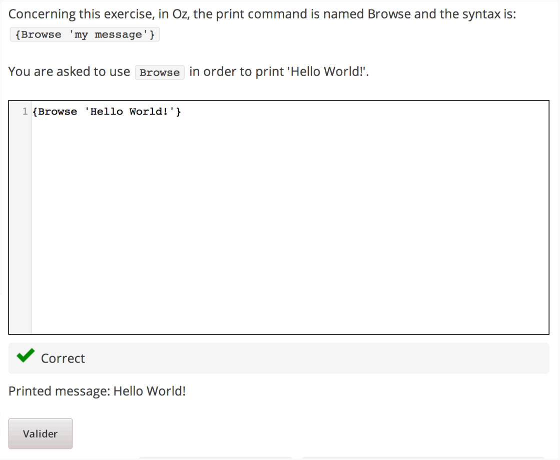
Submissions

- 🟥 14/08/2014 14:19:29
- 🟩 14/08/2014 14:19:18

© 2014 Université Catholique de Louvain

INGInious is distributed under AGPL license

SUBMISSION... FROM EDX



Concerning this exercise, in Oz, the print command is named Browse and the syntax is:

```
{Browse 'my message'}
```

You are asked to use `Browse` in order to print 'Hello World!'.

```
{Browse 'Hello World!'}
```

✓ Correct

Printed message: Hello World!

SUBMISSION... FROM ANYWHERE

Plugin system can do everything!

- Other MOOC platforms
- External javascript
- BlueJ
- Claroline
- Moodle
- ...

... you just need to develop the plugin ;-)

TASKS

Task are composed of two files:

- `task.yaml`
- the `run` file

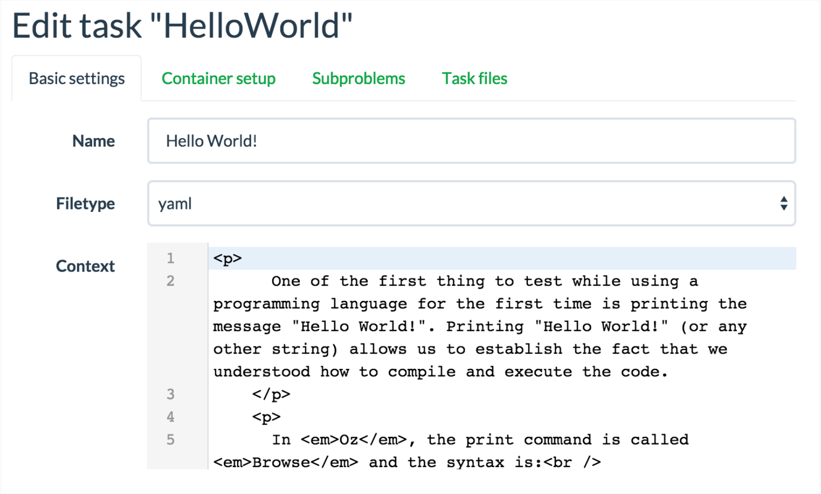
task.yaml: TASK DESCRIPTOR

Contains the description of the task

```
author: Me
context: |-
    A bit of reStructuredText to explain what students have to
    do
environment: php
limits:
    memory: 100
    time: 30
name: Hello World!
problems:
    student_response:
        language: oz
        header: |-
            Please, print hello world!
        name: ''
        type: code
```

I don't want to open a text editor

We have a nice interface for that:



Edit task "HelloWorld"

Name: Hello World!

Filetype: yaml

Context:

```
<p>
      One of the first thing to test while using a programming language for the first time is printing the message "Hello World!". Printing "Hello World!" (or any other string) allows us to establish the fact that we understood how to compile and execute the code.
    </p>
    <p>
      In <em>Oz</em>, the print command is called <em>Browse</em> and the syntax is:<br />
```

THE RUN FILE

- An executable (shell, python, ...)
- named `run`
- that you provide
- contains all the intelligence
- does the grading

INGInious provides a very nice API to help you

AN EXAMPLE OF RUN FILE

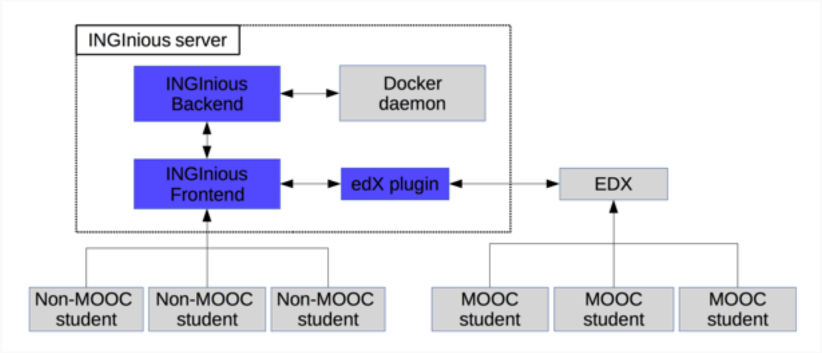
```
#! /bin/bash

# This line parses the template and put the result in studentcode.py
parsetemplate ——output studentcode.py template.py

# Verify the output of the code...
output=$(python studentcode.py)
if [ ”$output” = ”Hello World!” ]; then
# The student succeeded
feedback ——result success ——feedback ”You solved this difficult task!”
else
# The student failed
feedback ——result failed ——feedback ”Your output is $output”
fi
```

INTERNALS

ARCHITECTURE



INTERNAL ORGANIZATION

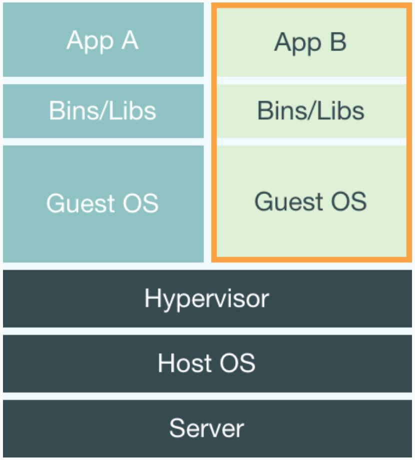
- Backend
  - Grades
  - Python module
  - Standalone
  - $\leftrightarrow$ Docker
- Frontend
  - Displays and stores
  - ”Stateless”
  - $\leftrightarrow$ MongoDB

DOCKER AND ”VIRTUAL MACHINES”

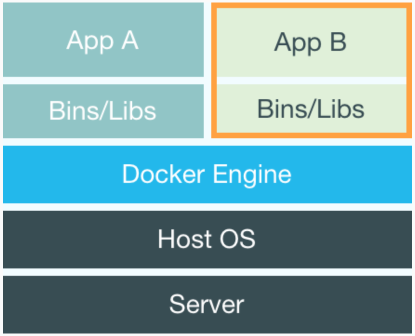
- Powered by Docker
- Containers, not VMs
- Simple to create, launch and manage
- Security

WHY DOCKER (1): CONTAINERS VS. VMS

Virtual machines



Container



schemas from `http://www.docker.com/whatisdocker`)

why docker (2): it's too simple

- I want to install and start a container on CentOS:

  ```
  $ docker run centos
  ```
- Ubuntu?:

  ```
  $ docker run ubuntu
  ```
- I want a basic container with MariaDB installed:

  ```
  $ docker run mariadb
  ```

From 1+ hours to 1 second.

WHY DOCKER (3): IT'S WAY TOO SIMPLE

A custom container with PHP for dummies

- Create a file named "Dockerfile", with this content:

```
FROM     ingi/inginious-c-default
RUN      yum install -y php
```

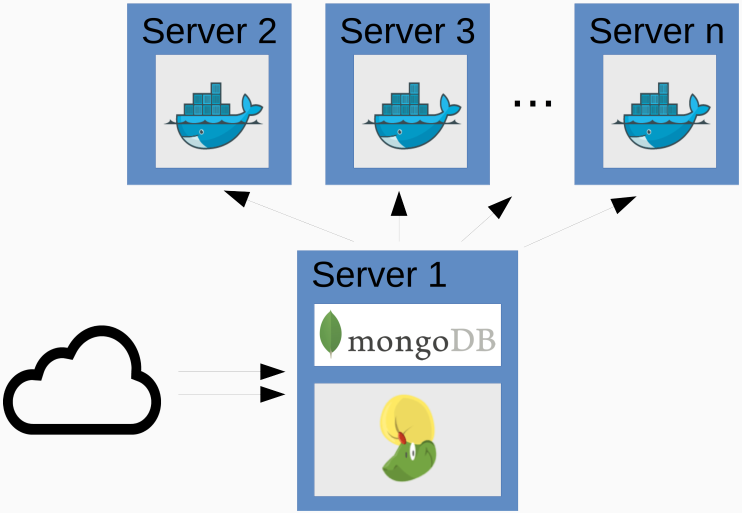
- `$ docker build -t a_name .`
- And you are done!

HANDLING LOAD

INGInious is made to scale horizontally

- Stateless frontend
- Multiple Docker daemons
- MongoDB

SCALING HORIZONTALLY WITH DOCKER

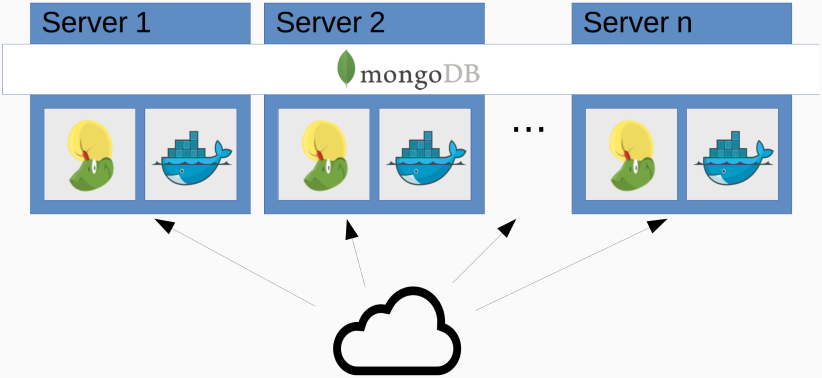


Scale horizontally the grading, not the frontend.

MONGODB

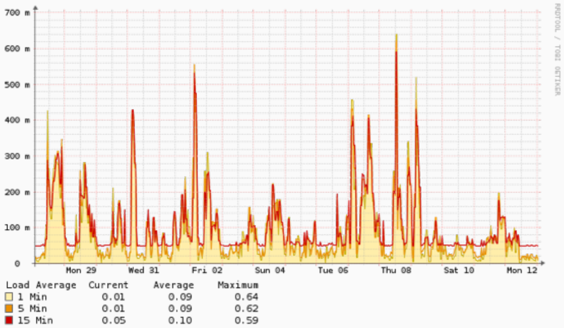
- NoSQL
- Document-oriented
- Perfect to store submissions
- Replication & Sharding

SCALING HORIZONTALLY WITH MONGODB



INGInious is not even aware of that!

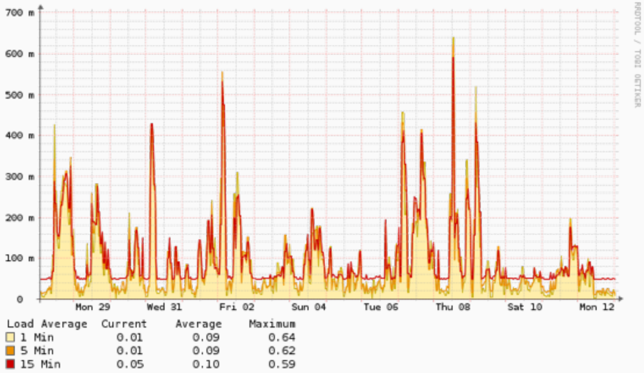
YOU WON'T NEED HORIZONTAL SCALING



Load Average	Current	Average	Maximum
1 Min	0.01	0.09	0.64
5 Min	0.01	0.09	0.62
15 Min	0.05	0.10	0.59

- Load average: mean of total CPU usage for all CPUs on 1, 5, 15 minutes
- During Louv1.1x exams - $\sim$ 600 students
- Single machine, 6 (v)CPUs, 8 GB ram
- Max sustainable load average: 5-6
- Higher bound on the graph: 0.7 (m is for milli...)

YOU WON'T NEED HORIZONTAL SCALING (2)



Load Average	Current	Average	Maximum
☐ 1 Min	0.01	0.09	0.64
🟧 5 Min	0.01	0.09	0.62
🟥 15 Min	0.05	0.10	0.59

No need to invest 500€/month on Amazon Cloud for grading

INGInious is Free & Open-source

`http://www.github.com/UCL-INGI/INGInious`

guillaume.derval@student.uclouvain.be
anthony.gego@student.uclouvain.be

Gitter: UCL-INGI/INGInious