



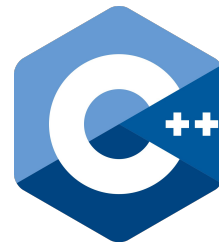
CS 4287: Final Project

Bao Nguyen, Blake Quigley, and Leqiang Wang



Motivations

- An online programming competitive game, specifically geared for educational use
- Help students learn programming faster and more effectively through competitive drive
- Get younger kids excited about programming through a fun videogame perspective
- Utilize cloud computing to compile and run many different programming languages without needing to install on your individual machine
- Potential users: STEM summer camps, AP Computer Science teachers, tech company on-campus visits, students preparing for engineering interviews, etc.



Example Use



1. Easily and quickly join classroom game with unique game pin
2. Present user with programming question (of teacher's desired difficulty level)
3. Using a variety of possible programming languages, users input code to solve problem
4. In the cloud, a random input is generated and solved with our own correct solution
5. The user's provided code executes with the same input, compares with the correct output, and generates useful statistics
6. In browser, user receives success/failure, correct output, user's output, stdout for debugging, execution time, etc.
7. Users are ranked based on statistics such as execution time, lines of code, first to successful result, etc.

Sort a list of 100 integers



| | |
|----------------|-------------|
| Lauren | 6650 |
| Natalie | 6200 |
| IZ | 6150 |
| Megan | 5900 |
| Schaeff | 5500 |

Cloud Computing Concepts

- Website is hosted in the cloud, so we don't have to buy or set up physical hardware
- Website is essentially infinitely scalable through the cloud
- AWS chosen for free education credits and to gain experience due to widespread industry use
- Kubernetes used to orchestrate tasks among Docker Workers executing the user's code and returning results
- Container concepts used for security with arbitrary code execution
- HTML/CSS/JS used for simple front-end website interface



kubernetes

Design of Docker Workers

- **Basic Goals**

- Execute code (Python to start) with custom input
- Limit the running time (in case of infinite loop or otherwise)
- Measure the running time
- Send back the output of the execution

Design of Docker Workers

- **Approaches**

- Use exec to execute Python in a Python based service
- Redirect input and output to StringIO
- Use time out signal to ensure running time limit
- Acquire system time at the start and end point of the execution
- Exception handling to handle the errors
- Only one docker instance run on every (virtual) machine, no concurrent execution.

Design of Docker Workers

- **Advanced goals**

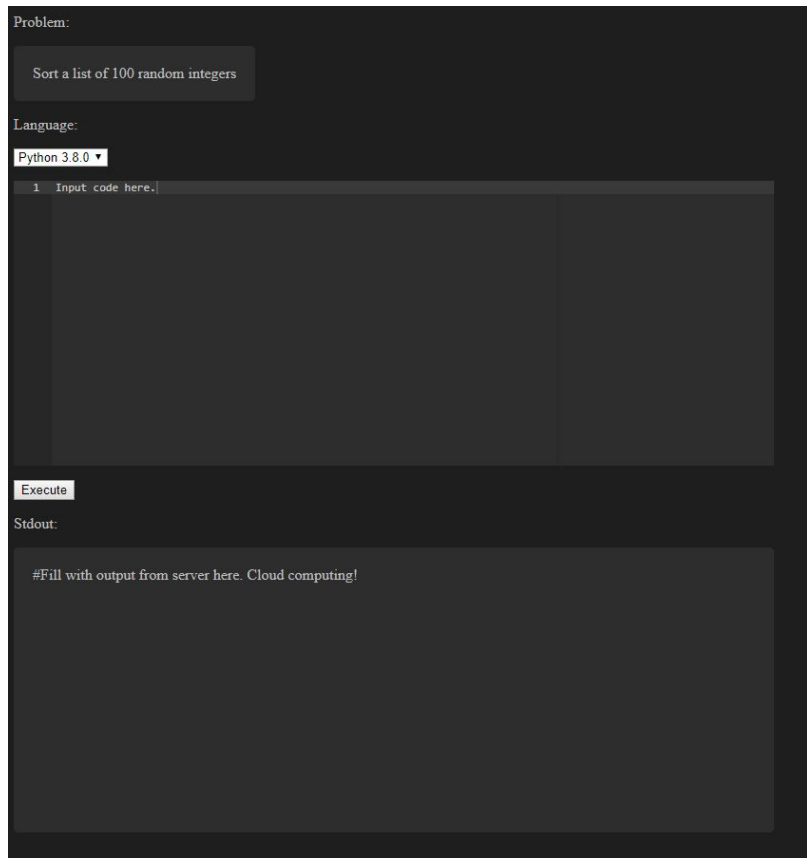
- Have test cases for certain problems and compare user's code output
- Properly schedule to worker nodes when the system is overloaded
- Data for test cases can be very large, may use a vm for a database server.
- Use cache to improve I/O efficiency (combine with node schedule)
- Support multiple languages (i.e. C++)
- Security check of the code (prohibit system calls)

Work Responsibility

| Bao Nguyen | Blake Quigley | Leqiang Wang |
|--|--|---|
| Back End | Front End | Back End |
| Worker nodes for executing code and communicating results to master / report | User interface / web server setup / PowerPoint / report | Master node for coordinating code execution tasks / report |

Current Status

- Created and iterated on project idea
- Architected design for implementation of project proposal
- Signed up for AWS with education credits
- Started experimenting locally with executing code from external source
- Simple website UI for testing (using Ace for code editor)



Next Steps

1. Finish website that can take user input, execute code in the cloud, and return stdout to the browser in several popular languages
2. Provide user with random sample problem to solve, check input code for correctness, and return useful statistics
3. Full, easily joinable competitive game ready for classroom use

Any Questions?