AUTOMATED CODE AND CONTEST EVALUATION SYSTEM

Kadir Emre Oto **Muhammed Burak Buğrul** **Süheyl Emre Karabela**

Advised by **Assistant Professor Ayşe Tosun**

Istanbul Technical University Faculty of Computer and Informatics Engineering

ACCES

Try it yourself | Get Report | Contact us

What is ACCES?

Problem: Most university departments started to assign coding problems in student homework and exams, but the evaluation and grading process takes a long time such that students may need to wait more than four weeks to see their grades, and academicians have to spend a considerable part of their time on evaluating them instead of doing research projects.

Solution: ACCES is built to solve this problem, it is a system for universities. ACCES enables academicians to create online homework/exams, and students to implement and submit their assessments online. When a submission is done, it will be evaluated by the system automatically. For that matter, students can see their results as soon as they submit their codes, and academicians do no longer have to put in time on grading process.

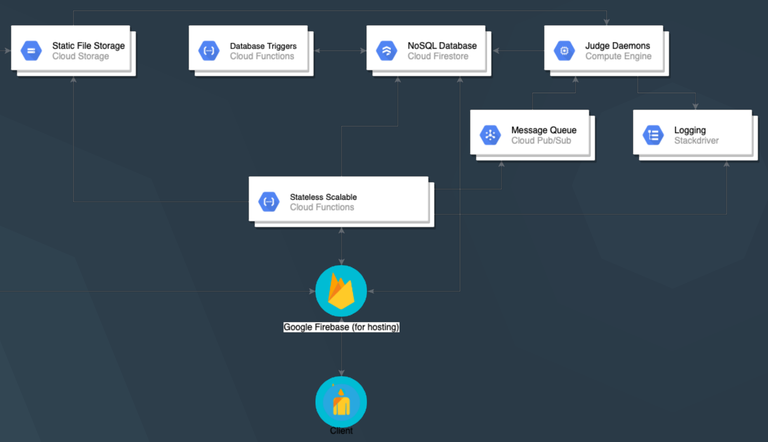
Technology Stack

- Python 3.7
- Google Firebase
- IOI Isolate Sandbox
- NodeJS / ReactJS
- Docker

Utilities

- Netlify
- Trello
- Google Calendar
- Gitlab
- Draw.io

Project Structure



Currently Supported Languages

- C++ 98
- C++ 17
- Python 2.7
- Python 3.6
- C98
- C11
- Java 8
- Java 11

Performance Measurements

We use Google Cloud Functions for Firebase as the back-end solution. After we deploy the code, Google creates some instances to execute the function when the corresponding event triggers. As the load of the function increases or decreases, number of instances are adjusted accordingly to handle changing volume of requests. We can observe the speedup in execution time by measuring the request per second of the function with repeated tests.

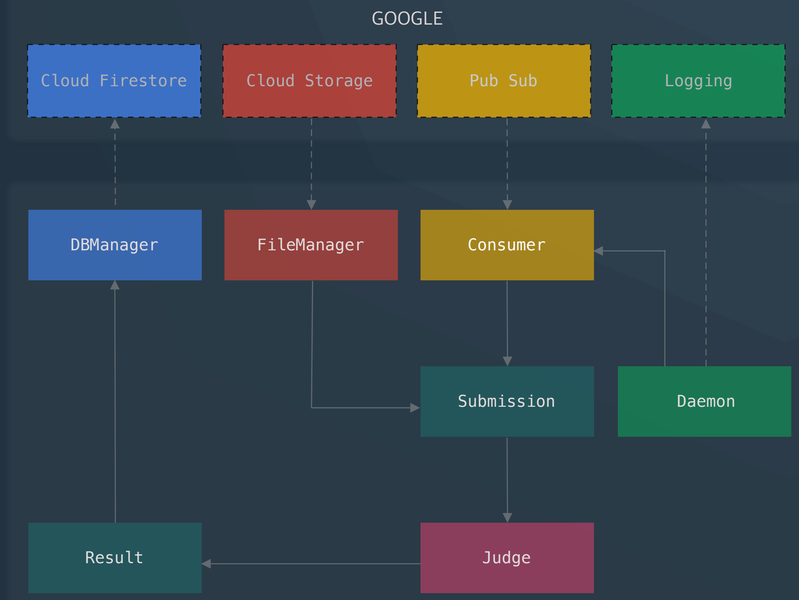
Batch Number	Request per Second
1	350
2	1540
3	4060
4	12050

Judge daemons are crucial for the system. They are loaded with heaviest part of the system: evaluation of user submitted codes. These codes can be anything. They may be time consuming, there may be memory leaks etc. We tested a single daemon with random memory, CPU, time inefficient and efficient codes. To complete the test, we produced these submissions into the message queue in batches. Then we measured run times per submission in the daemon.

#Active Test Cases	Average Evaluation Time
486	0.3 seconds
990	0.4 seconds
4992	0.6 seconds

How an ACCES Judge Works?

Purpose of a Judge daemon is to get tasks to evaluate from the message queue. When a student submits a solution for a task, it will be sent to the message queue after back-end checks and adjustments. A daemon listens to the message queue and takes submissions according to its' computing power. When the evaluation is done, it writes the results to the storage and the database. All of the daemons work independently. So, any number of daemons can be added to the system easily.



ACCES Daemon

System Outcomes

- Automatic evaluation of assignment submissions
- A single button for re-evaluating all submissions in case of any change in the test data set
- Opportunity for organizing coding exams
- Analysis of every student task by task for an assessment
- An online code editor that saves the students' work and provides an autocomplete feature.
- Real time detailed assignment submission result analysis for students.