APIO 2020 Technical Report

APIO (Asia-Pacific Informatics Olympiad) is an IOI-like competition for delegations within the Asian and Western Pacific regions. APIO 2020 was hosted online by Indonesia, on 15-16 August 2020.

General Statistics

- 33 registered countries
- 852 registered contestants
    - 717 contestants actually started the contest

Contest System

We used Judgels as the contest system, at this commit. Judgels has been used for Indonesian national olympiad in informatics since 2015.

Tweaks

For APIO 2020, we added the following improvements/tweaks to Judgels:

- Increased MySQL's maximum concurrent connections to 500 (from the default value of 150).
- Added caching in NGINX for serving Raphael (the browser/client-side static assets of the contestant interface).

- Added capability to have IOI-style grader function problems in Java.

Load Test

We ran a load test of Judgels to ensure that it would be able to handle the load of the contest. Load test was done using JMeter 5.3. It was deployed on a separate DigitalOcean droplet (6 shared vCPU, 16 GB RAM) located in the same region as the Judgels droplets (Singapore). 1000 Judgels users were created and added as contestants to a single test programming contest containing a single batch-type problem.

The first scenario tested is read-only load, which is similar to load generated when contestants are joining the contest and viewing the problems. 750 JMeter threads were used to generate load simulating 750 users each doing a single home page load and login (including static frontend assets), then repeatedly viewing contest/problem/scoreboard/clarifications. In this scenario, submissions were not tested. Load was immediately increased and sustained for 10 minutes. We found that a 4 vCPU, 8 GB RAM droplet for the main Judgels droplet could handle the load, but with high CPU usage. RAM usage never exceeded 5-6 GB. We also tested an 8 vCPU, 16 GB RAM droplet, which could handle the load with some spare capacity left.

The second scenario is submission load, which is meant to verify that uploading many submissions in a short time would not cause any unpredictable load on the contest system. We simulated 100 users submitting 8000 submission files for a batch programming problem within 2 minutes, both with small files that is supposed to have an accepted verdict and with 50 kB files that would cause a compile error verdict. For this test, we used a single grader machine running Gabriel with 2 worker threads. We found that the submission uploads didn't increase the load significantly compared to the read-only load. All submissions were eventually graded correctly by the grader.

Deployment

Based on the above stress test, we deployed Judgels on DigitalOcean using Ansible scripts, with the following specifications:

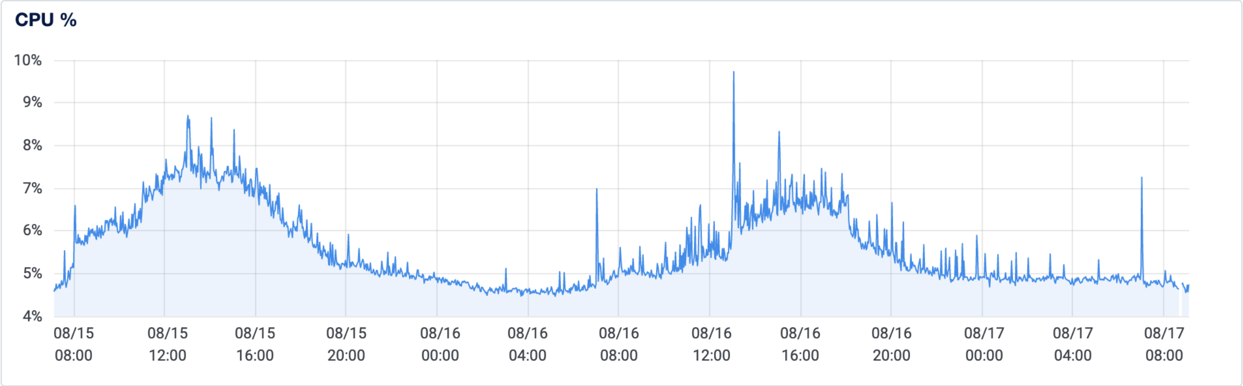
- 1 droplet for hosting Judgels app services (MySQL, Jophiel, Raphael, Sandalphon Sealtiel, Uriel, RabbitMQ)
    - Dedicated CPU, CPU optimized, 16 GB / 8 CPUs
- 15 droplets for hosting Judgels grader (Gabriel, 2 worker threads each)
    - Dedicated CPU, CPU optimized, 4 GB / 2 CPUs
- All droplets:
    - Singapore region
    - Ubuntu 18.04.3 LTS
    - Droplet monitoring turned on
- Floating IP for the main Judgels droplet to allow quick failover without having to wait for DNS TTLs to expire.

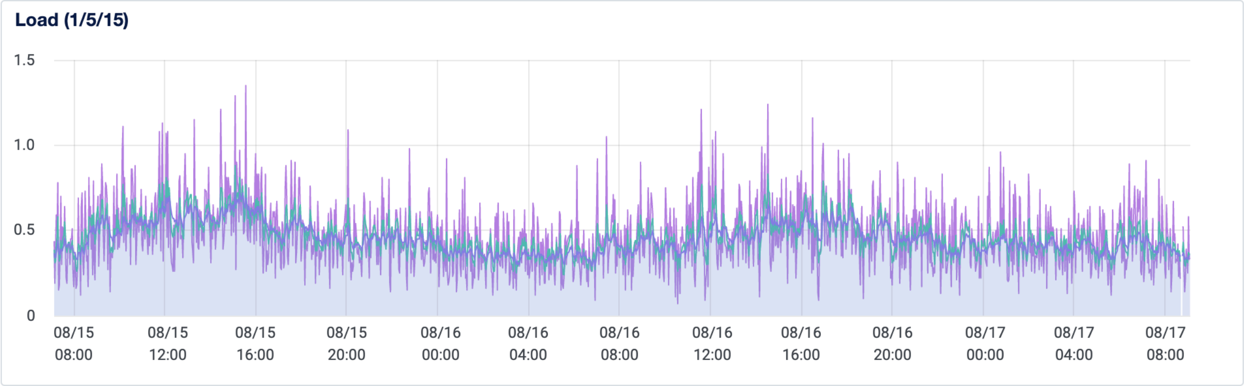
Contest Statistics

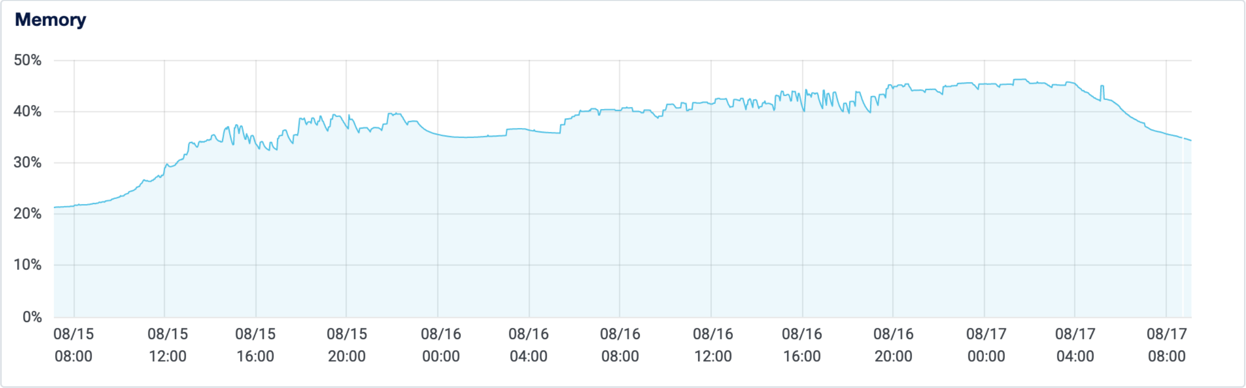
Droplet Performance

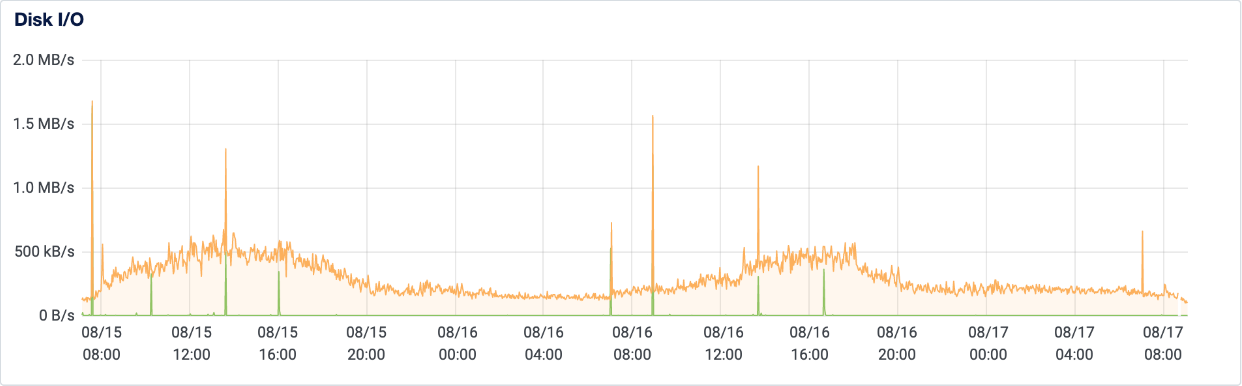
Here are the graphs of various stats of the droplet hosting Judgels app services during the competition windows. Unfortunately, we lost the stats of the droplet hosting Judgels grader.

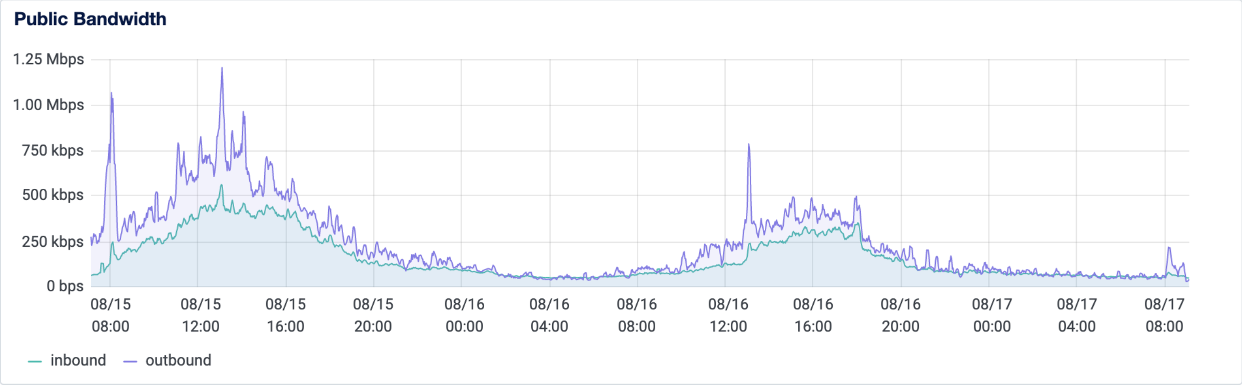
(All times in UTC+7.)





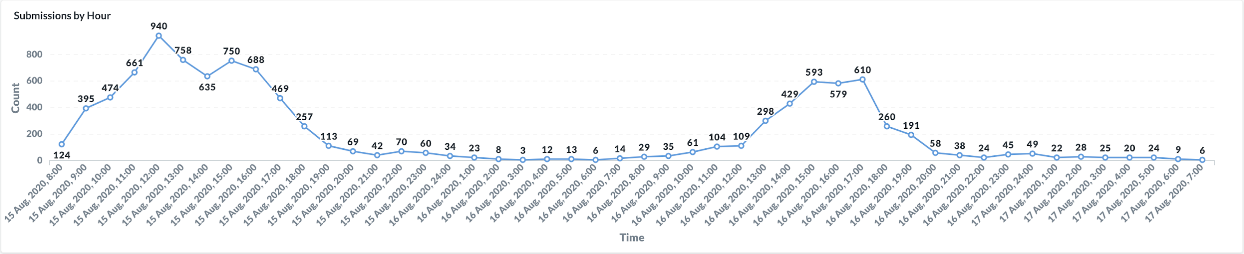




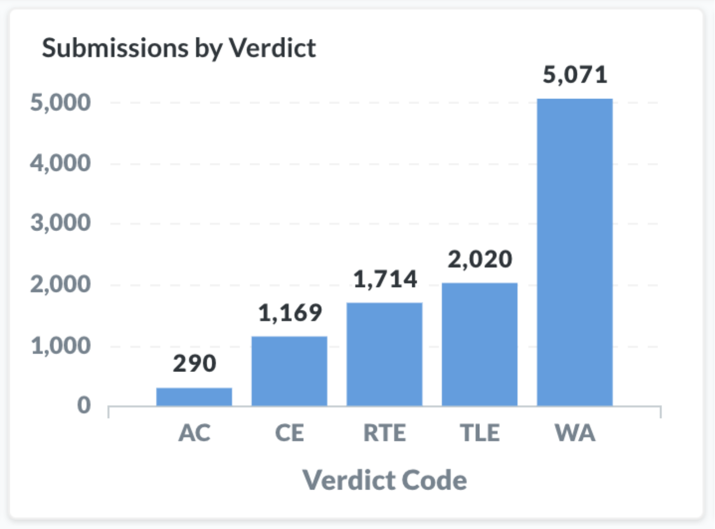


Submission Stats

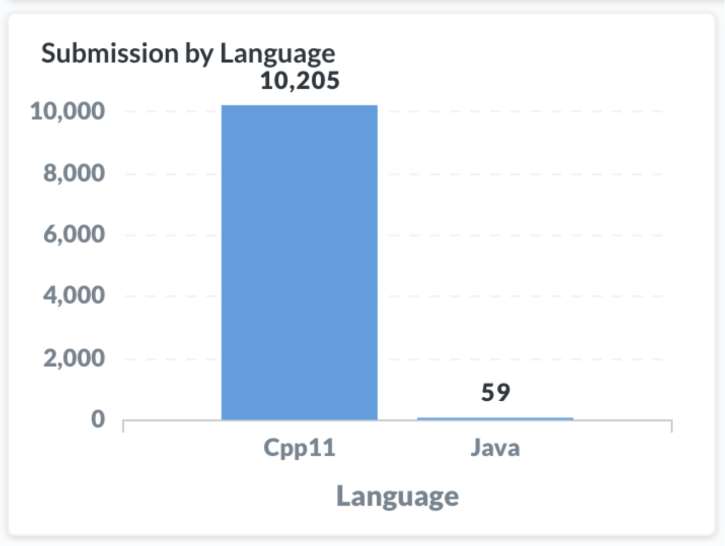
There were 10264 submissions for the contest proper. Here is the submission distribution over the two competition days, by hour:



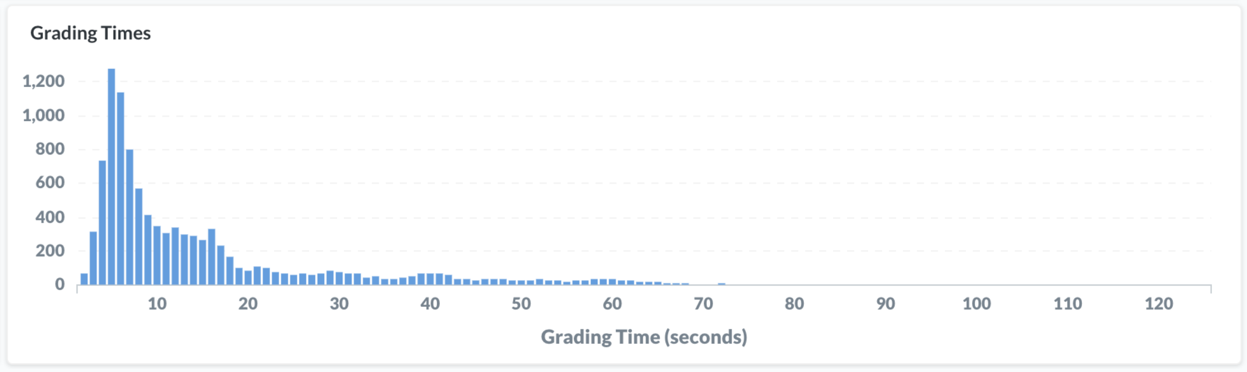
Here are the verdict distribution. The unusually high number of CE (Compile Error) submissions are due to wrong submission format. In APIO 2020, we use IOI-like grader function problems. During the contests, many contestants did not know how to submit such problems, even though we have provided practice session with similar problems.



99%+ contestants used C++. Only 59 submissions were made using Java:



Most submissions (90%) were graded in 30 seconds or less:



Incidents

Overall, we faced no contestant-facing downtimes or server crashes during the whole competition window. No contestants asked or received additional time outside of their personal 5-hour contest timer.

However, there are several incidents worth mentioning:

MySQL user password changed after taking droplet snapshot

The night before the competition window started, we took the main app droplet down for a while to take a snapshot as a backup. When the backup was complete, we turned the droplet back on. Then, we noticed that all services could not connect to the database.

Upon further investigation, it turned out that the password got reset to the previous password we used during first Judgels deployment for testing purposes.

Timeline:

- Mid-July 2020
    - created a droplet for internal scientific committee testing.
    - deployed Judgels.
- 5 August 2020
    - dumped MySQL databases from the testing droplet, using mysqldump --all-databases command.
    - created a droplet for contest proper.
    - deployed Judgels, including MySQL (with different set of passwords and credentials).
    - restored database from the dump.
- 14 August 2020 (night)
    - took the droplet down to take a snapshot.
    - turned the droplet back on.
    - Judgels services started.
    - Judgels services failed to connect to DB (access denied).
    - figured out what happened.
    - changed the password to old password.
    - redeployed Judgels services with updated DB password.
    - all services worked correctly.

Root causes:

- Why did the DB password get changed?
    - Because it turned out that the dumped databases includes MySQL user accounts.
        - Why?

      - Because we used “--all-databases” option in “mysqldump” command.
- Why didn’t we realize this sooner?
  - Because the updated password didn’t take effect immediately.
    - Why?
      - Because MySQL only take updated grant/password after we flush or restart the MySQL process.
        - Why did the updated password take effect after taking droplet snapshot?
          - Because taking a droplet down and taking it back up would result in restarting the MySQL process.

Learnings:

- Do not use “--all-databases” when dumping MySQL databases; always selectively choose the desired databases to dump.
- Test restarting services / droplets as early as possible, so that when there is real production issue during contest, we are sure that restarting services / droplets won’t cause any new problem.

Chinese delegation reported that Chinese flag in contest system is wrong

During competition window, the Chinese delegation leader reported that some contestants noticed that the Chinese flag in contest system is wrong. (I.e., has slightly different shape of stars). Judgels is using react-flag-kit library to render country flags. Unfortunately, it turned out that there are several issues in FlagKit flag assets, including the reported Chinese flag.

FlagKit-style country flags are only present in the contest system and can only be seen by contestants during their personal 5-hour timer. They are not present anywhere else, including the official website and social media. Therefore, we decided not to do any modification to the contest system during the competition window.

We finally managed to fix the issue by using a different library called react-flag. This fix was released in time for the open contest, which was 1 day after the contest proper.

Non-Latin symbols not displayed properly in clarifications

We noticed that non-Latin symbols in the clarification text were replaced by placeholder symbols. This is caused due to the contest system databases' collation using the default MySQL collation (latin1_swedish_ci) that does not support all Unicode characters, as the collation was not explicitly set when creating the databases. We fixed the contest system deploy scripts to use utf8mb4_unicode_ci collation when creating databases.

This issue does not affect the display of problem statements as problem statements are not stored in the database.