Version 4: Asynchronous grading architecture

CS 744: Design and Engineering of Computing Systems Autumn 2023

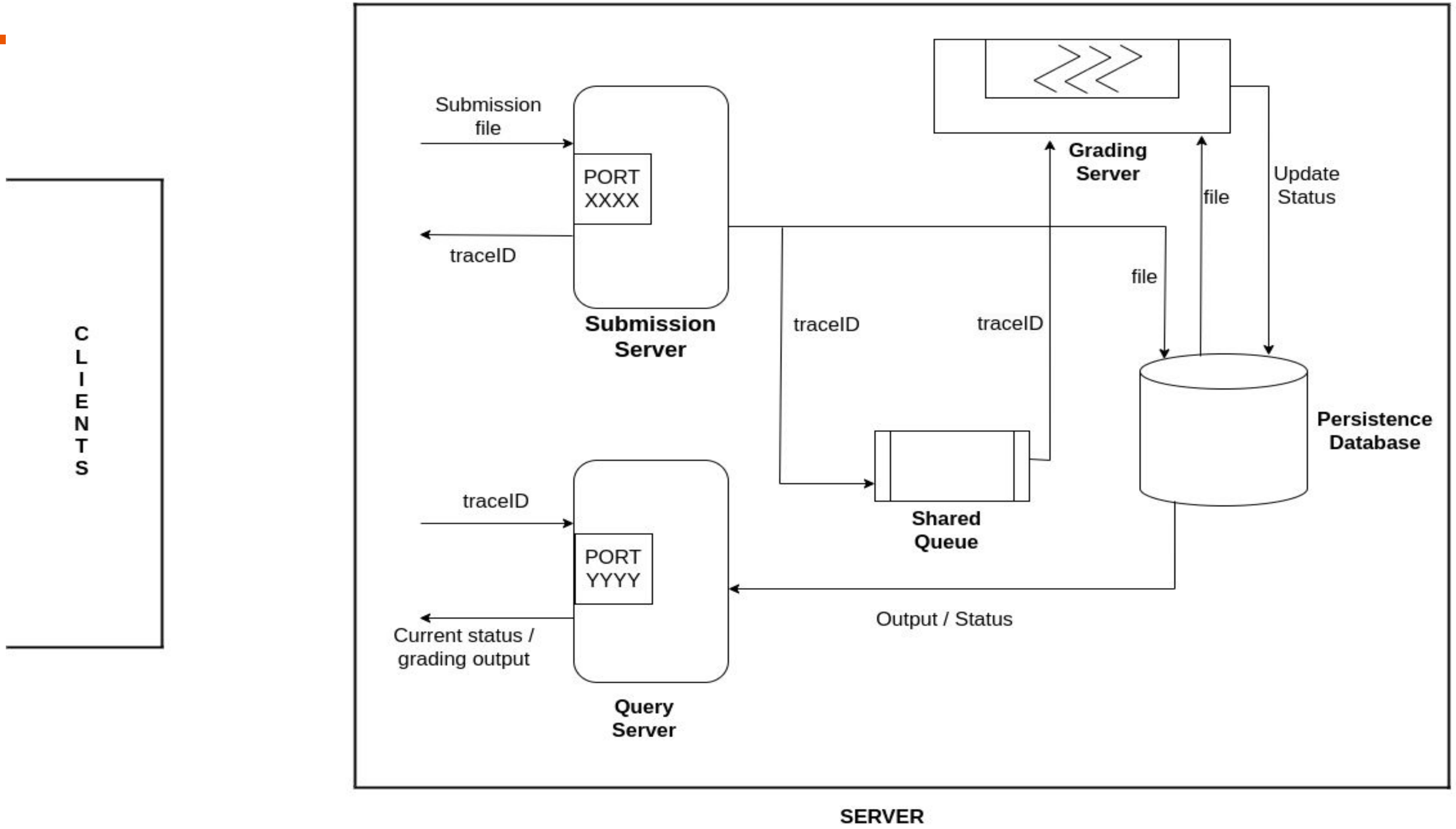
- Soumik Dutta (23m0826)
- Kumar Ankur (23m0829)

Goal of version - 4

The goal of this version is to turn the server architecture into an asynchronous grading architecture.

Now, the client is no longer waiting for the grading task itself. The grading task is what is happening ‘asynchronously’.

Autograder Overall Design



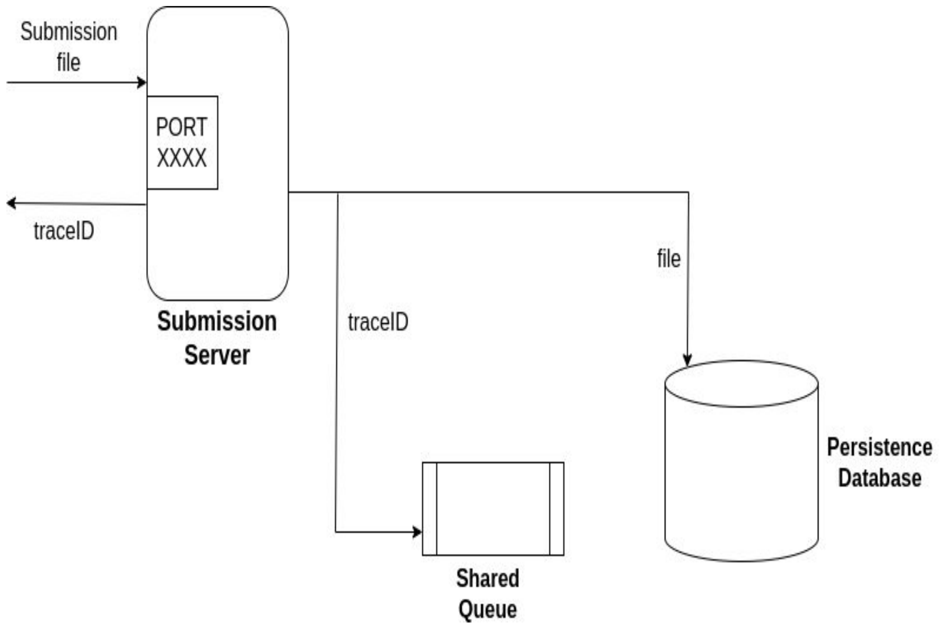
Server Design

- There will be 3 applications running on the server on different ports.

- Two of them will be publicly accessible and one will be for internal use.

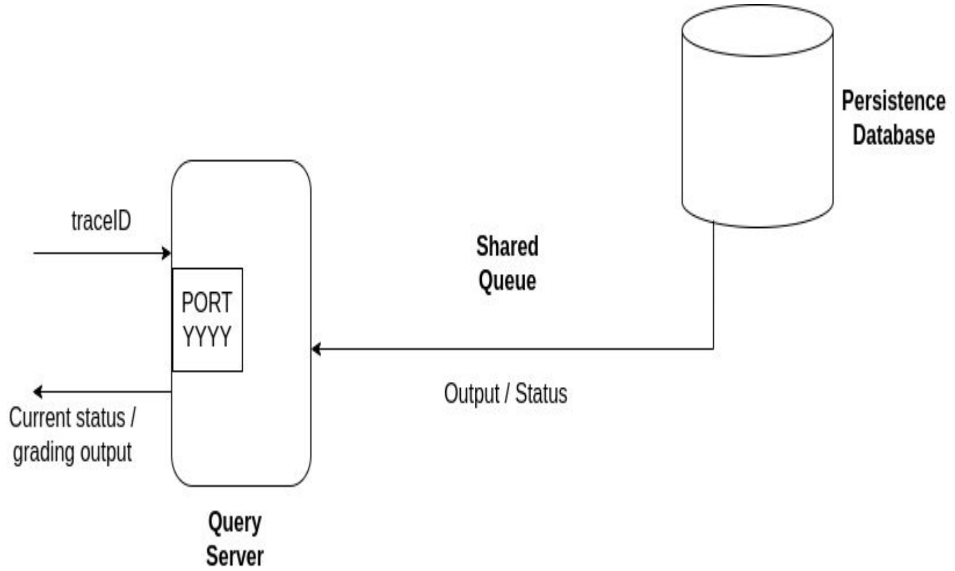
Submission server

- First application (SubmissionServer) is responsible for receiving new files for grading,pushing it into the queue, storing it into the database and then sending unique trace ID as response to the client.



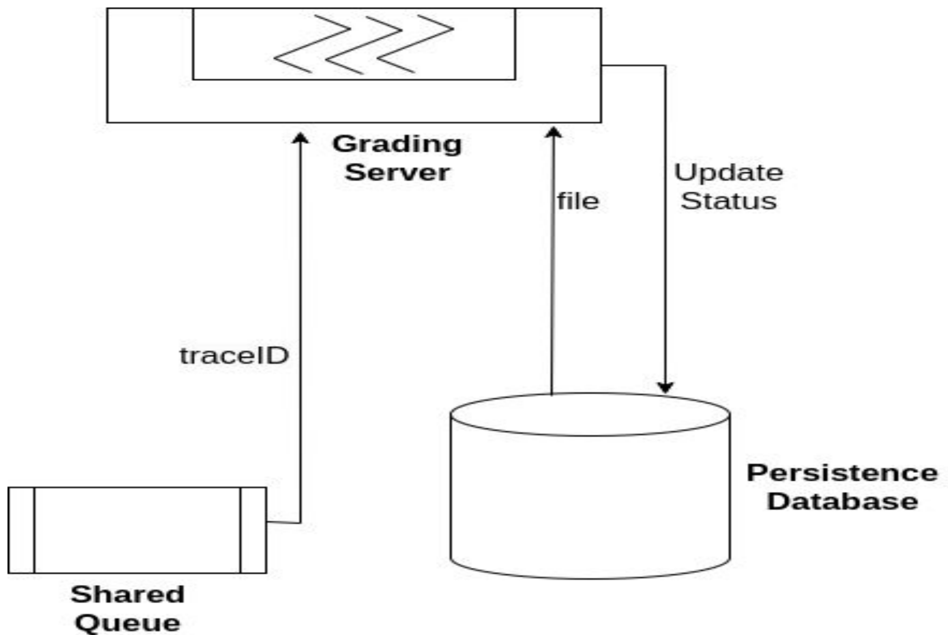
Query Server

- Second application (QueryServer) is responsible for handling grading status check request with respect to the given trace-ID by querying the database and then sending the current status to the client.



Grading Server

- Third application (GradingServer) does the actual internal work of grading. It one by one pulls the trace IDs from the shared queue, gets the grading file with respect to the trace-ID from the database, grades the file, then again stores the result back into the database.



Client Design

- Depending upon the input flag <**new**|**status**> client application will initiate the connection with server to the particular port of the server.
- If the input flag is <**new**> client will send the grading file to the server port where 'submission-server' application is running and in response gets a unique request ID .
- If the input flag is <**status**> the client queries to the particular port of the server where 'query-server' is running.

Applications Used

- Server code is written in C++
- Postgres DB is used as persistent database
- Redis queue is used as shared queue
- Bash script is used for load testing
- Python script is used for plotting graph

Performance analysis

Graphs plotted

1. Request rate
2. Successful request rate (goodput)
3. Timeout rate
4. Other Error rate
5. Response time
6. CPU utilization
7. Average number of active threads
8. Average Queue Length