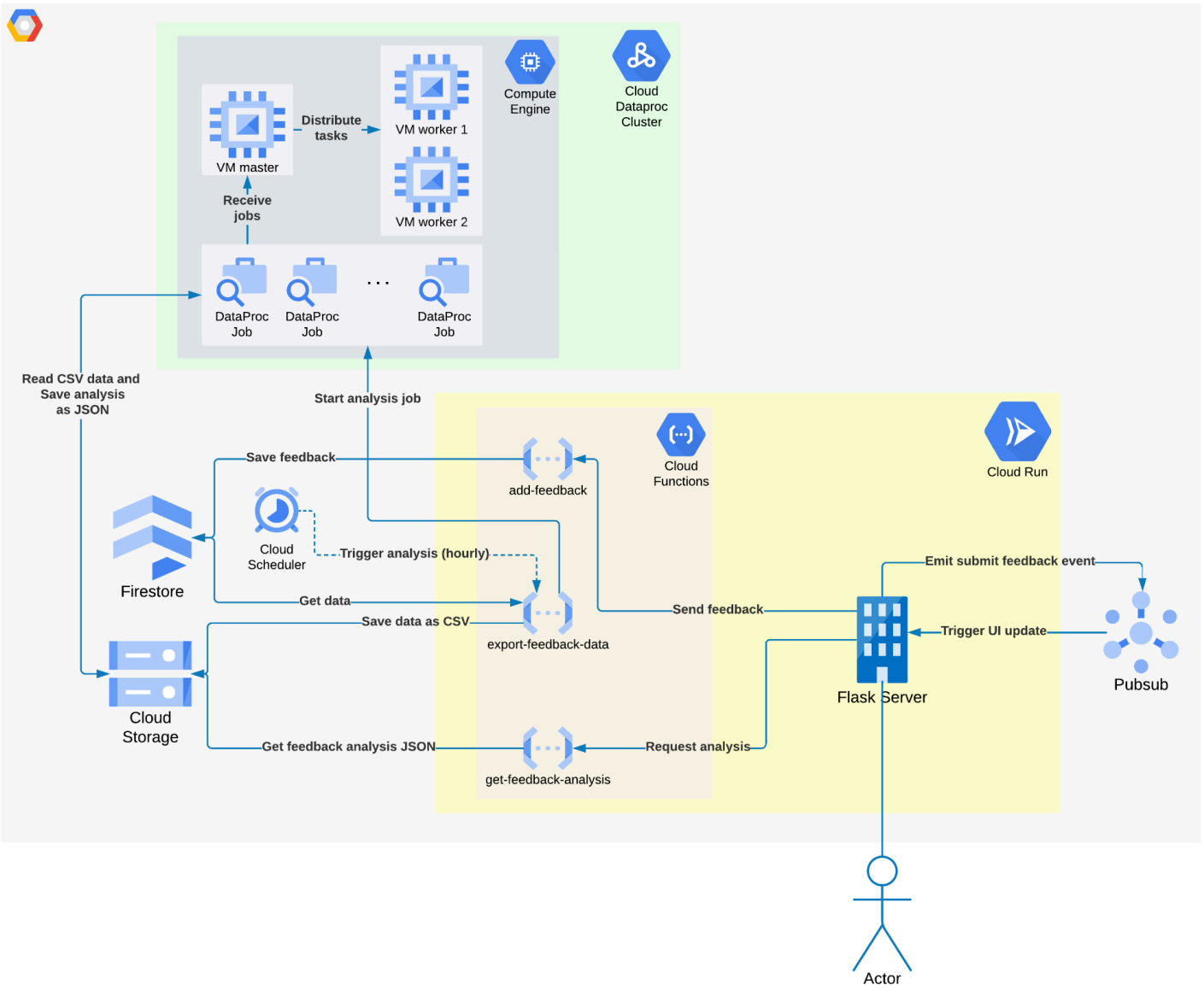
Distributed, real-time feedback and analytics system for restaurants

This system will allow users to give instant feedback on restaurants and receive real-time analytics about the reviews, like average ratings and common keywords. This application will utilize Cloud Scheduler, Cloud Functions, Firestore, DataProc, Cloud Run, and Pub/Sub, following a client-server and event-driven architecture.

System Overview:

The system is divided into three main components, each responsible for different tasks: feedback analysis, real-time feedback submission, and feedback validation.

- Component 1: Automated feedback analysis with Cloud Scheduler, Cloud Functions, DataProc and Cloud Storage
- Component 2: Real-time feedback submission using Cloud Run, Firestore, Websockets, PubSub
- Component 3: Function as a service using Cloud Functions and Cloud Run



Component 1: Feedback Analysis

This component automates the analytics workflow, providing scheduled insights into customer satisfaction and service performance:

- Cloud Scheduler triggers Cloud Functions at specific intervals to process recent feedback from Firestore.
- The “export_feedback_data” Cloud Function exports feedback data from Firestore to a CSV file, uploads it to a Cloud Storage Bucket, and creates a DataProc Job.
- DataProc performs analytics on the CSV file to derive insights like average ratings for ambiance, food quality and service, and common keywords in feedback. After processing, it exports the results as a JSON file to the Cloud Storage Bucket.
- Cloud Storage stores both the raw feedback CSV files and the processed analytics JSON files for long-term analysis and historical records.
- The “get_feedback_analysis” Cloud Function returns the JSON file from the bucket, in order to display the ratings in the UI.

Component 2: Real-time Feedback Submission (Cloud-hosted)

- Cloud Run hosts our Flask-based backend service, capable of scaling automatically based on traffic. It provides API endpoints for the following: submitting feedback, retrieving lists of restaurants available for review along with existing reviews, and retrieving the feedback analysis via our Cloud Function.
- Firestore serves as the storage solution for our system, keeping track of the restaurants open for reviews and storing all feedback related to each restaurant.
- Cloud Functions is used to add new feedback into Firestore securely and to pull feedback analysis.
- Emits a new event to Pub/Sub with each feedback submission: This functionality guarantees that the application stays updated with the latest feedback, allowing for immediate UI refreshes. Each time a user submits feedback, a new event is dispatched to Pub/Sub, initiating a real-time update across the user interface.
- Connects with the UI through WebSocket and listens for events from Pub/Sub to update the UI once feedback is submitted: This feature ensures the UI receives real-time updates whenever a user submits feedback. The application uses WebSocket to create a live connection between the UI and the server. Upon receiving an event from Pub/Sub, indicative of new feedback, the application promptly updates the UI to display the latest feedback. This mechanism guarantees that users always have access to the most current feedback information.

Component 3: Feedback Validation

- A cloud function processes and validates incoming feedback requests. It checks for duplicate submissions.
- This function is triggered via HTTP requests from the real-time feedback submission component, ensuring each piece of feedback is valid and unique before storing it in Firestore.

Analysis & Discussion:

The system provides a comprehensive solution for collecting and analyzing customer feedback in real-time, using a combination of managed services for efficiency and scalability, showcasing different characteristics across several dimensions:

Performance

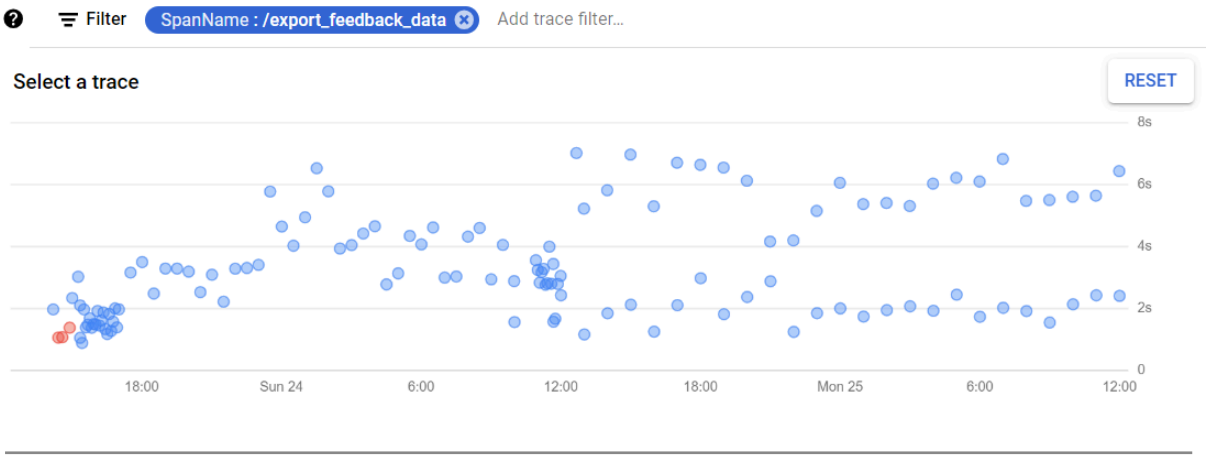
- High processing efficiency due to DataProc and automatic scaling via Cloud Run, ensuring the system can handle large datasets and traffic peaks efficiently

Latency

- Low latency in feedback submission and UI updates, thanks to Firestore and Pub/Sub, enabling near-instantaneous user interactions.

Latency (as the time that takes to execute the function) - each dot represents a single execution at a specific point in time; the color represents the success (blue) or failure(red) of the request:

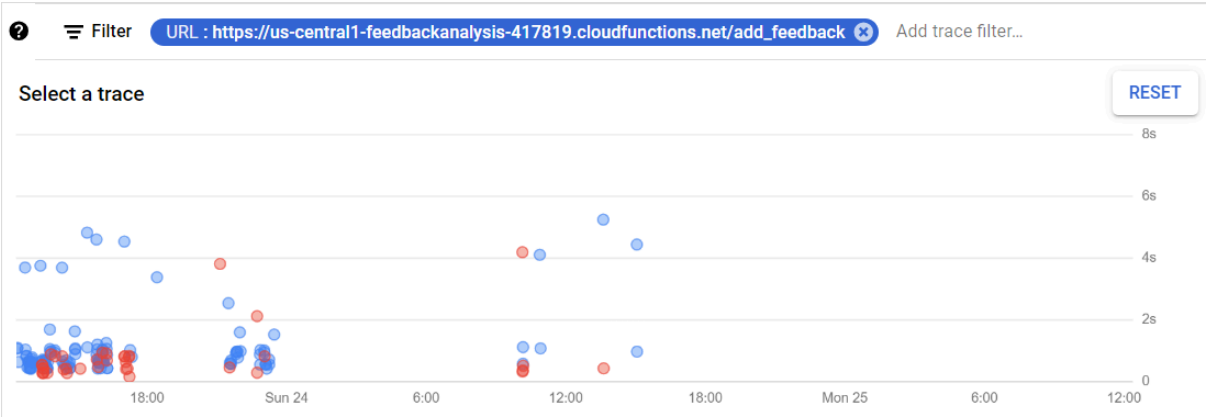
- export_feedback_data Cloud Function



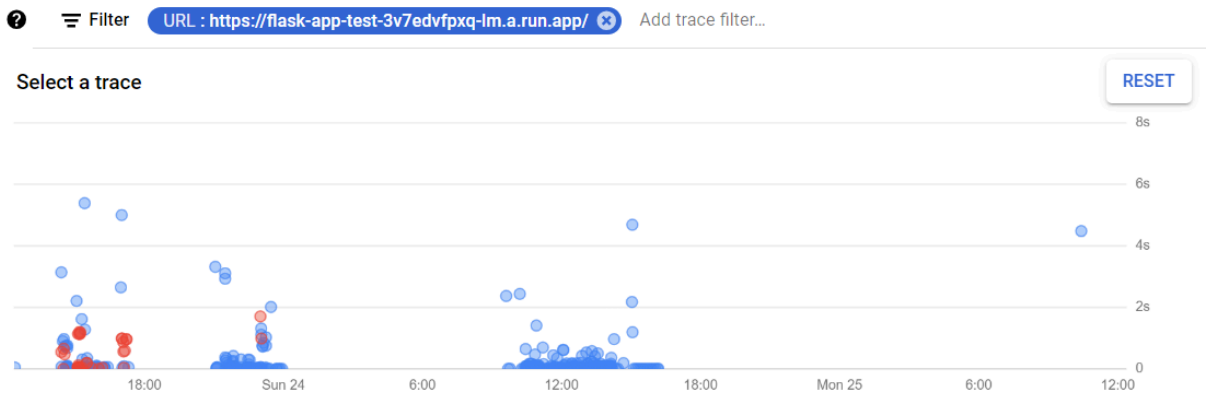
There's a range of execution times from about 2 seconds up to around 6 seconds. The cluster of dots at the lower end indicates that most executions are on the faster side, but there is variability.

The chart shows occasional spikes where the execution time increases, due to some factors such as increased data processing time, temporary resource contention in Firestore or Cloud Storage, or delays in starting the DataProc job.

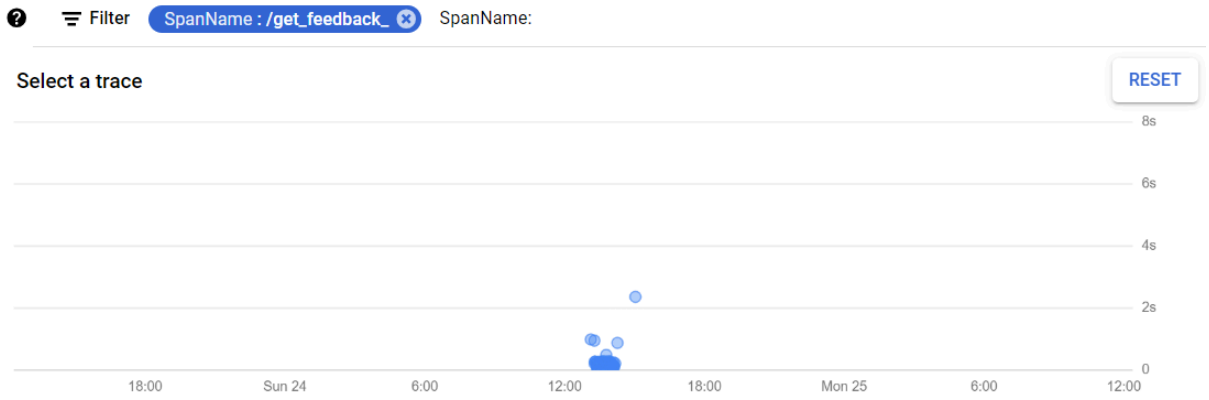
- add_feedback Cloud Function



- flask server



- get_feedback Cloud Function



- DataProc job: 55 seconds average

Job ID	Status	Region	Type	Cluster	Start time	Elapsed time
80276f05-2c3f-4758-bf17-81d5b2b8af57	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 10:00:03 PM	56 sec
4fb21e5e-b655-4776-b16a-8eaa6a129f24	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 9:59:06 PM	58 sec
cdc09157-4c03-42b9-9838-b3f3003946ea	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 9:00:02 PM	52 sec
d9b27a23-ae3b-452c-a965-e157352c07b8	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 8:59:08 PM	52 sec
d2a0f03f-b229-4320-96c3-8b9a5268448b	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 8:00:01 PM	55 sec
5b0e84de-26dd-469e-bd86-54aa50aca7e1	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 7:59:05 PM	51 sec
97f4031c-5733-406f-bf1c-1534dc336235	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 7:00:03 PM	55 sec
105c69bb-03d3-4dbc-bb1c-8b18021e2124	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 6:59:07 PM	55 sec
7c8a942f-f897-4da0-8412-cd0f1548de17	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 6:00:02 PM	53 sec
16c72f02-72f9-4f04-9ffb-073292456af6	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 5:59:06 PM	55 sec
bd5ff06e-d1c8-444c-b84b-da2eb9afd4e8	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 5:00:02 PM	56 sec
98a4823c-ec8c-4f8e-a5cb-555bacc46473	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 4:59:06 PM	1 min
a0d93410-df82-4cd6-847d-d26ec5f3c2bd	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 4:00:02 PM	52 sec
95615dd4-d895-4285-b642-c8668d392469	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 3:59:06 PM	52 sec
709f881a-7d0a-4b47-a019-61f51fde1e86	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 3:00:02 PM	54 sec
a7dd79cc-7971-4051-b4dc-7c368d6059d3	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 2:59:05 PM	56 sec
ab3cabae-7da1-4f3d-9c4b-d8dacd323c77	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 2:00:03 PM	54 sec
4d3bcd93-f3b8-45db-8125-3b52d798a2c7	Succeeded	us-central1	PySpark	feedback-analysis-cluster	Mar 25, 2024, 1:59:05 PM	56 sec

Reliability

- Increased reliability through the use of managed cloud services like Cloud Functions, Cloud Run, Firestore, and DataProc, which offer built-in fault tolerance and redundancy.

Transparency

- The system provides clear insights into customer satisfaction and feedback through its analytics and real-time UI updates.

Scalability

- Designed for scalability, with services like Cloud Run and Firestore automatically adjusting to demand, ensuring the system can accommodate growth without significant changes.

Conclusion

In conclusion, our three-component distributed real-time feedback and analytics system demonstrates the advantages and challenges of using modern cloud-based components and architectural patterns.