Distributed Systems (H0N08A)

Report: Google App Engine

Dries Janse *(r0627054)*
Steven Ghekiere *(r0626062)*

November 28, 2019

1. Which hosts/systems execute which processes, i.e. how are the remote objects distributed over hosts, if run in a real deployment on the App Engine platform (not a lab deployment where everything runs on the same machine)? Clearly outline which parts belong to the front- and backend. Create a component/deployment diagram to illustrate this: highlight where the client and server are.

2. At which step of the workflow for booking a car reservation (create quote, collect quotes, confirm quotes) would the indirect communication between objects or components kick in? Describe the steps that cause the indirect communication, and what happens afterwards.

3. Which kind of data is passed between the front- and back-end? Does it make sense to persist data and only pass references to that data?

4. How have you implemented your backchannel? What triggers the backchannel to be used, how does the client use the backchannel and what information is sent?

5. How does your solution to indirect communication improve scalability?

6. Workers in GAE's Task Queues are by default set to run in parallel. While parallelism is usually a desirable property of a cloud service, as it enables scalability and thus faster overall processing, it may also endanger the application's state consistency. Assume a scenario in which two different clients try to confirm a couple of tentative reservations, i.e. their quotes are queued to be processed by the back end. Both include a tentative reservation for the last available car of a certain car type, so that, assuming correct behaviour of the car rental application, it should fail to confirm the quotes for one of them.

7. How does using a NoSQL database affect scalability and availability?

8. How have you structured your data model (i.e. the entities and their relationships)? Compared to a relational database, what sort of query limitations have you faced when using the Cloud Datastore?

9. What is the most critical difference between confirmQuote and confirmQuotes from the viewpoint of transaction management? Explain an alternative to your solution for the all-or-nothing semantics of confirmQuotes which is essentially different from the viewpoint of transaction management. Given the underlying distributed storage layer, what are the implications of each solution for performance and data consistency?