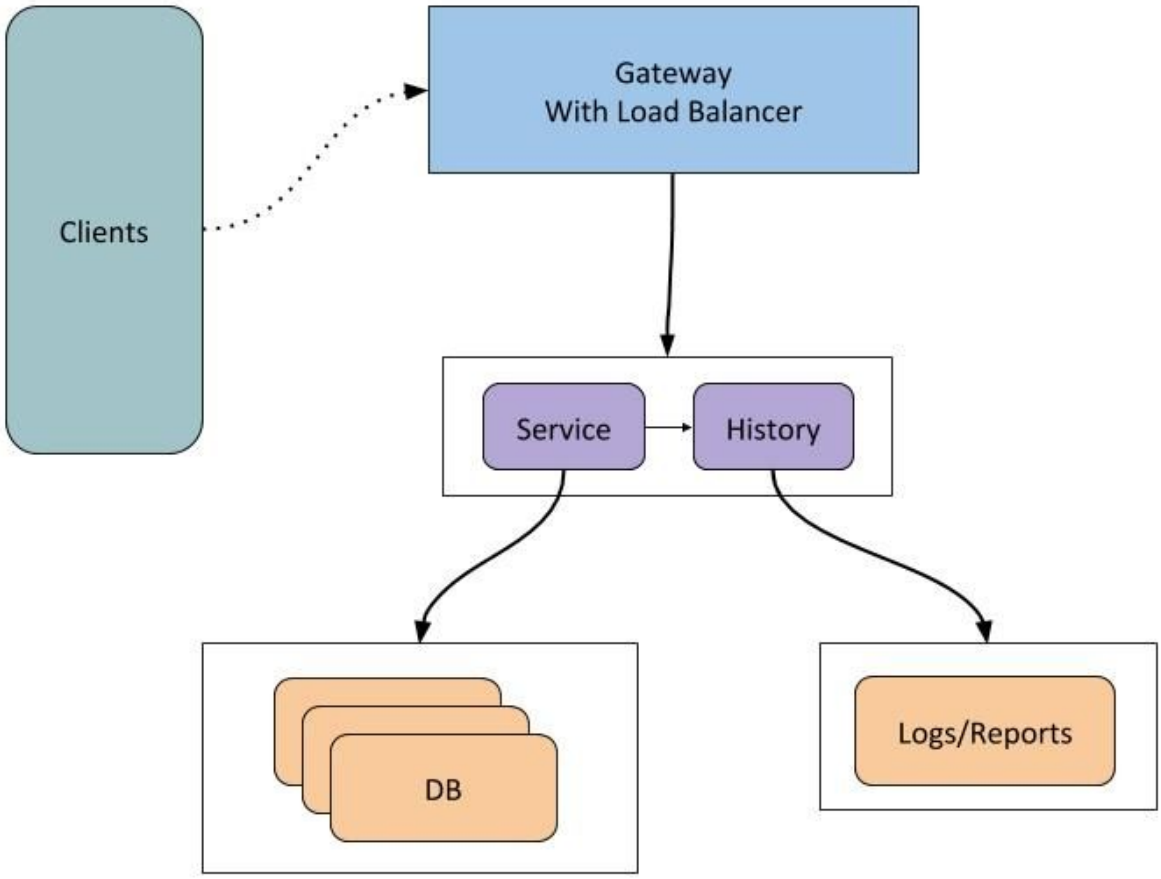

Proposal for Analytic System Design

Caveats

- I assume that the company has given me “All the money” that I want. Open-source will be used when possible, but there are uncertain cases and any open-source tools will be presumed as options in the proposal
- This is a “High-level” description, meant to give an idea without a deep dive into particulars such as data migration, data policies, etc.
- This is only my second design proposal, so please feel free to let me know what I missing for next time! :)

Description

A load-balanced gateway will take in requests for analytic data, to which the Service is called and will handle the request by calling into the DB. When the request is finished, the response is sent back to the user. History is used to help with metadata or heuristics of the data, along with

records with regards to bugs or errors that occurred when calling the system. The backend is thought of in a microservice sort of architecture, with the boxes representing various clusters or containers that could be scaled with regards to service load.

Why microservice?

- Adds the benefits of scaling with more lightweight objects and classes
- Allows for resources to be allocated to where they are needed most (need more service power than DB space? Create more service nodes and reduce the DB cluster. Having more of a load on the customers looking for historical data? Up the History cluster for better throughput)
- Allows for services to be functional in one regard when other services are down, rather than a traditional monolith

Database Flavor

- Cloud Spanner
    - Highly scalable, cloud based SQL solution
    - Provides all the power of SQL databases with cloud based scale options
    - Very expensive, but for large data stores could be a good option
    - Helps with regards to the Historical data aspect of the system needs
- Cassandra/MongoDB/NoSQL
    - Highly responsive, allows for more malleable data
    - Is open source, and can be used with a non-commercial license in some regards
    - Allows for some transactional data and queries, but the queries have to be simpler.

Given the way the data needs to be hammered at by millions of customers at the same time, I would recommend Spanner just for the ability to have data consistency and transactions done in batches. Mongo/Cassandra/etc would work, but there would need to be additional design constraints considered.