

Scalability of the Concert Web Service

The concert booking service is a system to allow clients to retrieve information about concerts and performers, to make and enquire about reservations, to register and authenticate users, and to download performer images. It is expected that the concert booking service will experience high load when concert tickets go on sale, therefore the web service must satisfy the scalability quality attribute. To make sure the system is scalable, the web service has been built with statelessness, efficient domain model design, optimistic concurrency control, and caching in mind.

The web service is built upon the stateless HTTP protocol and JAX-RS a framework for the development of RESTful Web Services. Due to the web service being stateless, the server does not store any state about the client's session on the server-side. The client session data is stored client-side, consequently, the server can service any client at any time as there are no sticky sessions because all the relevant session data is stored on the client and then passed to the server as needed. Therefore, any client can be serviced by any server replica on any server. This provides good scalability as many servers can be easily set up to handle server load right away.

As the web service does not store session data server side, the server uses cookies to help keep track of which client it is processing requests for. Therefore, any server replica can use the cookie to find the user (cookie can also be used for authentication after the first login). The cookie (token) is stored in the database so that all server replicas can use the token to identify clients.

The domain model was designed to make queries, updates, and storage of data into the relational database more efficient. Using unique foreign keys allows for faster queries when querying an entry associated with another entity. For example, with unique performer IDs allows for faster queries about performers from the concerts table. Databases are sorted by IDs, making queries very quickly. Using foreign key IDs reduce query and update speeds and the amount of duplicated entries in another table due to having only one table updated using the Table per class hierarchy strategy.

In this web service implementation, the Hibernate vendor of JPA is utilised. Hibernate lazy-loads as much of the object graph as possible which saves a lot of resources. As the database will increase over time due to entries populating tables, at a point in time the resources needed will be more than the server can handle. Lazy-loading helps reduce this problem by loading a part of the domain model at a time, therefore saving server resources. Though eager-fetching may be useful to reduce the number of transactions occurring on the database. With multiple clients, it is highly likely that clients will attempt to reserve seats at the same time. Naïve JPA implementations will throw an error to stop concurrent commits. To solve this problem, we can use JPA's optimistic concurrency control (OCC), so that it will allow clients to reserve seats at the same time. However, OCC will throw an error when clients attempt to reserve the same seats at the same time. This means that clients, for the most part, can concurrently reserve seats, thus increasing the web service's throughput of transactions occurring on the database.

Caching is used when retrieving concerts and performers, caching the concerts and performers helps decrease the load on the server as it will not have to query the database and load the same objects repeatedly. The server will just tell the client that the data has not been modified since the last GET request and will, therefore, use the cached copy.

The asynchronous publish/subscribe system is implemented on HTTP's stateless protocol. The asynchronous response allows the server to send news notifications without blocking other requests, as it utilises a separate pool of threads. However, news notification system is not stateless the server replica has to store a list of responses.

The concert web service overall has good scalability; though, the asynchronous news system is not stateless and should be implemented using the WebSocket protocol.