

Airline Reservation System

Project for Big Data and Future of Computing

Amitabh Mehta, Prema Khatri, Ankit Gaur, Raghunath Reddy L, Sachin Sharma

We know from the CAP theorem that any system needs to compromise on one of the following: Concurrency, Availability and Partition Tolerance. For this project we demonstrate how a practical system can pick a different properties at different stages, thus providing best user experience and system performance.

We plan to demonstrate this tradeoff between consistency and availability using the example of an airline reservation system. The system will use cached data for high performance while most of the seats are availability, and then switch to a stricter consistency protocol after more than a certain fraction T of seats are booked. We also plan to run simulations to find out what may be a good value of T for given load conditions. Revision graph based technique will be employed to detect if a particular simulated history was consistent or not.

If time permits, we may also simulate client crashes and study it's effects on the value of T .

Deliverables:

1. **Airline booking server:** implementing APIs to book tickets for a flight.
2. **Caching server:** Which will cache available seats for the clients.
3. **Application Gateway:** Middleware that decides if cache server should be used to avoid database hits based on number of vacant seats.
4. **Clients:** App to book the tickets.
5. **History analysis tool:** Tool to analyze the transaction history to determine if consistency was violated.
6. **Simulation environment:** to run multiple simulations and capture the results.

