Lecture 29: Examples of end-to-end systems design
Recap: e-commerce system design

- Front end Web servers
  - CRUD: User profile management
  - REST: Product management
  - RPC: Shopping cart maintenance
    - Update product catalogue
    - Fetch user billing info
  - Order management (purchase, billing, shipping, cancellations, returns)
  - Recommendation algorithms
  - Pub/Sub: Message broker
  - Recommendation database

- User profile management
  - User profile data (RDBMS)

- Product management
  - Product catalogue

- Shopping cart maintenance
  - Shopping cart info (NoSQL)

- Order management (purchase, billing, shipping, cancellations, returns)
  - Order database
Video streaming: requirements

• Consider a simple video streaming system that lets users stream videos, and also create content. What functionality do we expect?
• Users should be able to upload content, which is made available in multiple resolutions and formats.
• Users should be able to download/stream content
• The system should recommend videos to watch based on history of which videos user click and watch
• Other services: store users profile information, authentication, billing for paid services, ..
• All these functions provided by multiple components, each having multiple micro-services
Video streaming system: example architecture
Social networking: requirements

• Social networking systems are hugely popular today, and handle large number of users exchanging huge volumes of data
• Users join a service, make friends, form a network
• Social network recommends more friends to add based on connections of existing friends
• Users post messages, which are disseminated within the user’s network
• Users see popular content posted by friends, and react to it
Social networking: example architecture

Front end Web servers

- User profile management
- User network management
- User message upload
- User likes and reactions
- Display personalized message feed
- Message broker

User profile data
RDBMS
Specialized graph data store
Social network graph

User generated content
NoSQL document store

Generate user feed
(popular posts in network)

User activity log
NoSQL

User feed of popular posts
NoSQL, in-memory

Separate micro-services for add friends, recommend friends
Instant messaging: requirements

• Users connect to server via persistent connections from mobile apps
• If two users are online and wish to communicate, messages exchanged between such users are delivered via the messaging service immediately
• If the recipient of a message is offline, the service buffers messages for later delivery. User’s client checks for such messages periodically.
• Users form groups of other users and post messages in groups, which are broadcast to all the users in the group.
Instant messaging: example architecture

Online users connected to server via persistent connections

Front end Web servers

Deliver messages to online users directly

User message upload

User message download

User group management

User post in groups
Broadcast to all users part of the group

Archive of all user messages

Buffer of messages to be delivered to user in future

Group database

NoSQL, in memory

RDBMS

REST API to fetch group info
Summary

• In this lecture
  • Systems design examples

• Look at designing more such simple systems on your own. Think of how you will make the systems more realistic by adding more complex features.