

Step 6: Detailed Design

High reliability to not lose files

- Create replica of image and image metadata storage
- Use load balancer between service and databases

High availability

- Create replica of services running upload and download applications
- Use load balancer between client and services

Data sharding

- Partition based on UserID
- Have to consider hot users that may have considerably more reads or uploads

Ranking and news feed generation

- Fetch latest, most popular photos of people user follows
- Assume fetch top 100 photos for a user's news feed
- App server will generate a list of user's follows and latest 100 photos from each user
- App server will use some algorithm to get top 100 photos
- Pre-generate news feed using dedicated servers.

How to send news feed contents to users

- Pull: client pull news feed contents regularly
- Push: servers push new data as its available (maintaining long poll request)
- Hybrid: move users who have high number of follows to pull-based

Cache

- Push content closer to user using geographically distributed photo cache servers (CDN)
- Consider using 80-20 rule to cache top 20% of photos