Grokking the System Design Interview
Search

- System Design Basics
  - Why System Design Interviews?
  - System Design Basics
  - Load Balancing
  - Caching
  - Sharding or Data Partitioning
  - Indexes
  - Proxies
  - Queues
  - Redundancy and Replication
  - SQL vs. NoSQL
  - CAP Theorem
  - Consistent Hashing
  - Long-Polling vs WebSockets vs Server-Sent Events (*New*)
- System Design Problems
  - System Design Interviews: A step by step guide
  - Designing a URL Shortening service like TinyURL
  - Designing Pastebin
  - Designing Instagram
  - Designing Dropbox
  - Designing Facebook Messenger
  - Designing Twitter
  - Designing Youtube or Netflix
  - Designing Typeahead Suggestion
  - Designing an API Rate Limiter (*New*)
  - Designing Twitter Search
  - Designing a Web Crawler
  - Designing Facebook's Newsfeed
  - Designing Yelp or Nearby Friends
  - Designing Uber backend
  - Design BookMyShow (*New*)
- Contact Us
  - Feedback

LearnTeach

- My Profile
  - View
  - Edit
- Logout

System Design Basics

Whenever we are designing a large system, we need to consider few things:

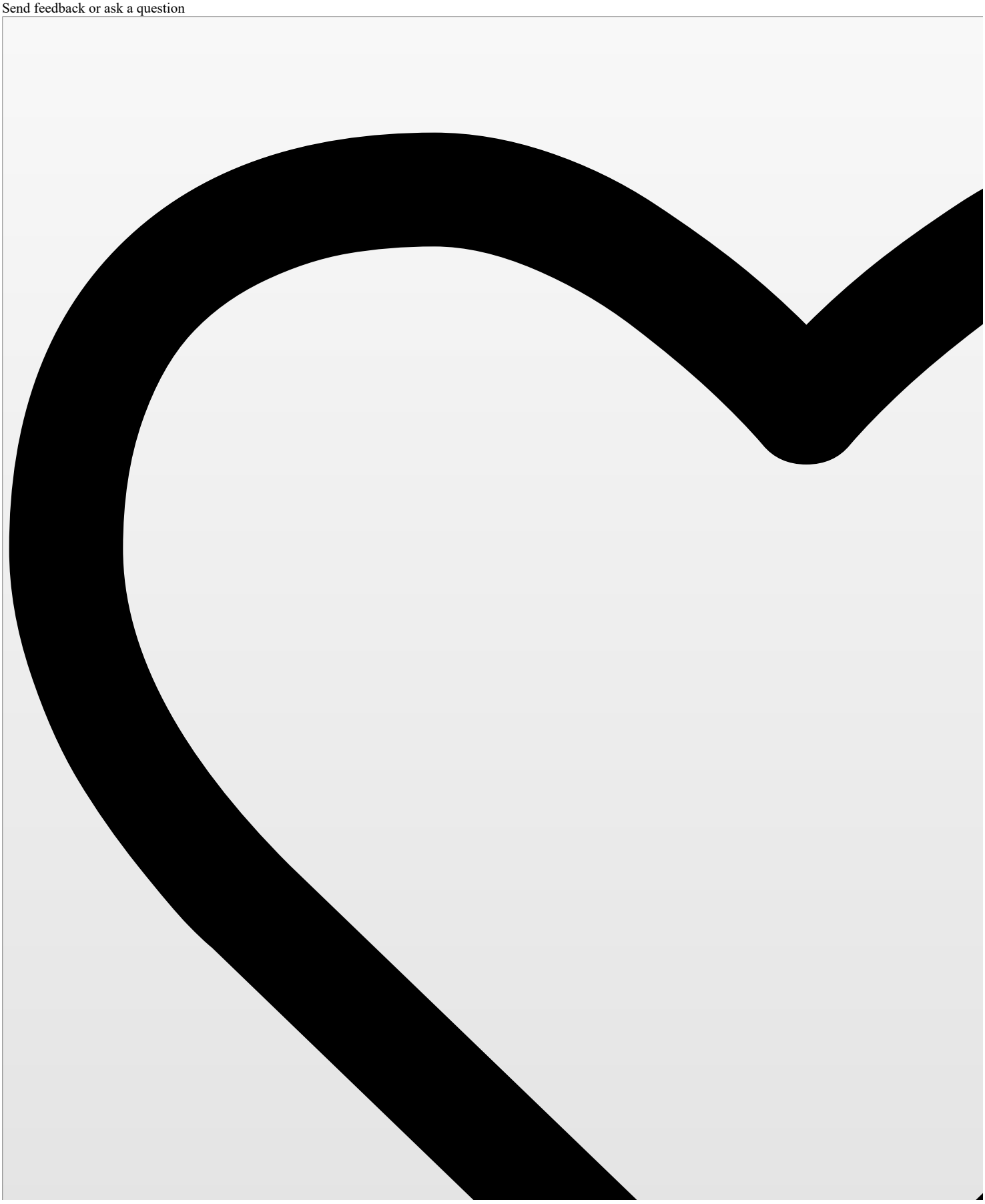
1. What are different architectural pieces that can be used?
2. How do these pieces work with each other?
3. How can we best utilize these pieces, what are the right tradeoffs?

Investing in scaling before it is needed is generally not a smart business proposition; however, some forethought into the design can save valuable time and resources in the future. In the following chapters, we will focus on some of the core building blocks of scalable systems. Familiarizing these concepts would greatly benefit in

understanding distributed system design problems discussed later. In the next section, we will go through Consistent Hashing, CAP Theorem, Load Balancing, Caching, Data Partitioning, Indexes, Proxies, Queues, Replication, and choosing between SQL vs. NoSQL.

Let's start with Load Balancing.

Mark as completed
← *Previous* Why System Design Interviews? *Next* → Load Balancing
Send feedback or ask a question



22 recommendations

- Home
- Featured
- Team
- Blog
- FAQ
- Terms of Service
- Contact Us