

System Design Interesting Reads

Basics:

[System Design Cheatsheet · GitHub](#)

[System Design Primer](#)

[How to Succeed in a System Design Interview](#)

[Approach a System Design Interview](#)

[System Design, Chapter 2: Sharding](#)

[System Design, Chapter 3: Load Balancing](#)

[System Design, Chapter 4: Caching](#)

[System Design, Chapter 5: Indexes in Databases](#)

[System Design, Chapter 6: Proxies](#)

[System Design, Chapter 7: Queues](#)

Milestone Research Papers:

[The Google File System](#)

[Dynamo: Amazon's Highly Available Key-value Store](#)

[MapReduce: Simplified Data Processing on Large Clusters](#)

[TAO: Facebook's Distributed Data Store for the Social Graph](#)

[Kafka: a Distributed Messaging System for Log Processing](#)

[Bigtable: A Distributed Storage System for Structured Data](#)

[Spark: Cluster Computing with Working Sets](#)

[Log Structured Merge Tree](#)

Popular Datastores:

[Choose the right data store - Azure Application Architecture Guide](#)

[Redis vs Aerospike](#)

[Elasticsearch vs MongoDB](#)

[MongoDB vs CouchDB](#)

Cassandra

[Cassandra writes in depth. Surprises you can expect from a... | by Andrzej Ludwikowski](#)

[Cassandra Data Modelling](#)

[Why Cassandra writes faster than Traditional RDBMS?](#)

[When to use Cassandra and when to steer clear](#)

[Log Structured Merge Trees](#)

[How is data read? | Apache Cassandra 3.0](#)

Redis

[Redis Persistence – Redis](#)

[Goodbye Cache: Redis as a Primary Database](#)

[Transactions in Redis](#)

[Partitioning: how to split data among multiple Redis instances. – Redis](#)

[Horizontal scaling in/out techniques for redis cluster](#)

[Redis Cluster](#)

[Key Problems with Redis Persistence](#)

Elasticsearch

[Elasticsearch from the Top Down](#)

[Uses of Elasticsearch, and Things to Learn](#)

[Elasticsearch from the Bottom Up, Part 1](#)

PostgreSQL

[Comparing Data Stores for PostgreSQL - MVCC vs InnoDB](#)

[Postgres MVCC](#)

[PostgreSQL 10: Partitions of partitions!](#)

Mongo

[How To Decide If MongoDB Is Right For You](#)

Interesting Blogs:

[All Company Blogs !!](#)

[Building Reliable Reprocessing and Dead Letter Queues with Kafka](#)

[A Practical Introduction to the Internals of Kafka Storage](#)

[How To Design A Scalable Rate Limiting Algorithm](#)

[Data Compression for Large-Scale Streaming Experimentation | by Netflix Technology Blog](#)

[WhatsApp-Engineering Inside-1. Real Time messaging are now an...](#)

[WhatsApp-Engineering Inside-2. In “WhatsApp-Engineering Inside-1” we...](#)

[NETFLIX system design. System Design](#)

[A Design Analysis of Cloud-based Microservices Architecture at Netflix](#)

[Microservice Architecture pattern](#)

[Resiliency: Cache Me If You Can](#)

[Microservices Design Patterns](#)

[Kafka Message Delivery Semantics](#)

[Replication In Depth - Kafka](#)

[Design Decisions for Scaling Your High Traffic Feeds](#)

[The Architecture Twitter Uses to Deal with 150M Active Users, 300K QPS, a 22 MB/S Firehose, and Send Tweets in Under 5 Seconds](#)

[Instagram Saves Switching to Cassandra from Redis](#)

[Jepsen Blog](#)

[Aerospike Developer Blog – Medium](#)

[Patterns for distributed transactions within a microservices architecture](#)

[A Guide to Atomikos](#)

[How to do distributed locking](#)

[Taming Garbage Collection](#)

[Whatsapp Scale Connections per server Tips](#)

[Modern Cache](#)

Real World Videos:

[Jeff Dean's talk at Stanford](#)

[Building Billion user Load Balancer at Facebook](#)

[Netflix Guide to Microservices](#)

[Amazon DynamoDB deep dive](#)

[Twitter: Timelines at Scale](#)

Advanced Techniques:

[Bloom Filter](#)

[Cache Stampede](#)

[Request Coalescing](#)

[Three Tier Caching](#)

[Consistent Hashing](#)

Actual Questions:

Easy

[Design Chess](#)

[Design Tic-Tac-Toe](#)

[Design Elevator System](#)

Medium

[Design Uber](#)

[Design Instagram](#)

[Design Facebook Messenger](#)

[Design URL Shortener](#)

[Design Dropbox / Google Drive](#)

[Design News Feed System / Quora](#)

[Design Price Surging Service](#)

[Design AirBnb](#)

Hard

[Design YouTube / Netflix](#)

[Design Web Crawler](#)

<https://medium.com/airbnb-engineering/avoiding-double-payments-in-a-distributed-payments-system-2981f6b070bb>