Designing Data-Intensive Applications Notes

Contents

How to Use This Template

- Create a new **Chapter** section for each audiobook chapter.
- Capture key ideas, notable quotes, and takeaways in the provided subsections.
- Add action items or open questions you want to revisit.

1 Chapter # – Title

1.1 General

- Redis, Apache Kafka (boundaries blurred)
- Single tool can't any longer meet data needs
- work is broken down into tasks into multiple tools
- mem-cached, elastic search, solar – application code keeps all this synced
- API hides implementation details from clients
- how do you ensure data remains correct and complete?
- how do you scale?
- what API is best
- how do you have an issue but the client can still be fine
- Book considers 3 concerns: reliability (toyota style), scalability, maintainibility (chnaging people should be able to work on the system productively)
- following chapters: architectures and algos important to achieve these
- Reliability:
- faults and fault tolerant systems in reliability
- fault is one component of a system deviating from its spec
- failure: system as a whole stops providing the service to the user
- fault tolerance mechanisms are important
- Netflix Chaos Monkey
- hard disk MTTF (mean time to failure) is 10-50 years, 10000 disks, on average 1 disk to die per day

- mo machines mo problems
- Virutal Machine instances stop running without warning because it is sometimes needed somewhere else
- Software Errors:
- independent events generally, a large number don’t fail all at once
-

Key Ideas

-
-
-

Architecture / Systems Mentioned

-

Concepts and Definitions

- **Term**: Definition or explanation.

Diagrams / Mental Models

-

Notable Quotes

- “”

Questions / Confusions

-

Practical Takeaways

-

Action Items

-