SE 464 Final Exam Study Guide

Some assorted questions we liked from the study activities:

Guest Lecture 1

1. Why are non-operational quality properties also critical to the success of a business?

2. Why is it important to intentionally choose architecture that maximizes specific quality attributes?

3. Name 4 quality attributes of a system.

4. What is reliability? Give an example of how reliability can be measured and affected.

5. Explain why you might want to host services in multiple regions, and why sometimes you might not want to do this.

6. Why is choosing what software language to use an important architectural design decision?

7. You notice your system's error budget is -25%. What does this tell you about the number of compliant and non-compliant events?

8. How does a Service Level Agreement (SLA) differ from a Service Level Objective (SLO)?

Midterm

1. Consider the full mesh topology structure. What are the pros and cons of this structure?

2. What is idempotency in the context of web requests?

3. What kind of data is Redis best used for?

4. Consider an application which provides cloud storage for files. The files can be arbitrarily large and contain arbitrary data. What data storage backend would be most suited to this task, and why: MySQL, MongoDB, Dynamo, S3, or Redis?

5. You are part of a team that is debating whether to prioritize the development of new features or improving existing NFPs in a software product. Evaluate the pros and cons of each approach.

6. What is the goal of evolvability?

7. Name 3 (of the 6) properties that result in a system with good dependability. Provide a short description of each.

8. Differentiate between the 2 different purposes of connectors of complexity as an NFP.

9. A system using network communication has the constraint that it must be implemented using different languages. Would gRPC be able to address this constraint?

10. You want to make and run a small app on your local machine to get some familiarity with basic features of a new programming language or framework that you're learning. Is a monolithic or microservice architecture better? Justify your answer.

11. (T/F) Are all designs considered to be architectures?

12. In a complex software architecture, where might you apply the Factory Method design pattern, and what benefits does it offer to the architecture, based on the content provided?

13. Explain why Facebook might choose to use a single-page JS app over a more traditional HTML app.

14. What is the difference between composition and aggregation?

15. Explain how the Three-Tiered Pattern could be implemented for a web application?

16. What is the purpose of a deployment diagram?

Final Study

1. System X uses shadow copies to support maintaining logs of specific files. However, the system needs to support better atomicity and have better performance. What type of logging can be introduced that addresses these specifications?

2. Justify the decision to use SSE over Web sockets and HTTP/1.1 with long polling by the RAMEN protocol.

3. Describe a real-word software scenario in which Kafka would be useful. List some system characteristics that Kafka would be useful to implement.
4. Compare and contrast the approach to quorum levels in CassandraDB and other NoSQL databases. Discuss the advantages and potential challenges of allowing programmers to choose the quorum level for each read/write in CassandraDB compared to the single read/write strategy in some other NoSQL databases.

5. A senior engineer at your company believes that replication of data is the best option to not lose data without having any tradeoffs. Evaluate why he might be incorrect.

6. In the lectures, there was only ever 1 view server for all clients. However, as we've seen previously, it's typically advantageous to have replicas all over the world to reduce latency for a global audience. How could you have multiple replicas of the view server all over the world, while still ensuring there is only 1 primary server (i.e. all view servers agree on the same primary).

7. Provide an example of a scenario where MapReduce can be applied for efficient data processing including what happens to the data at each phase.

8. You are building a cloud monitoring and analytics platform that provides real-time insights into the performance of applications and infrastructure. Where would you use batch processing and where would you use stream processing?

9. What are some common approaches to solving reliability problems?

10. Suppose a manufacturer is thinking of incorporating the Robotics Operating System (ROS) into their development of autonomous humanoid robots designed to perform high-endurance tasks. What's one advantage of incorporating such software?

11. You are a member of an engineering team developing a website for Oceanic Airlines. You have recently incorporated a feature enabling clients to select their seats during the flight booking process. One team member found a bug: booking two tickets simultaneously could cause the same seat to be assigned to two different tickets. Identify which ACID property is not respected by the booking service and propose a solution to prevent this issue.

12. Google Chrome focuses on 3 factors for security:
    a. reducing the severity of vulnerabilities
    b. reducing the window of vulnerability
    c. reducing the frequency of exposure

    Describe the method they use on how they address each factor.

13. Why use a trust management system?

14. Suppose you are a software engineer at a large company called Neta, and your company is planning to launch a new social media app called Strings. The company wants you to design a system that allows notifications (e.g., someone replied to your post, someone messaged you) to be delivered to users without much delay (i.e., a delay of a few minutes is acceptable). What method would you use to deliver notifications if you anticipate
    a. a low volume of notifications, or
    b. a high volume of notifications?

    Your solution should be cost effective given the expected traffic.

15. A company's mobile software architecture is suffering from backend degradation, and notices that the polling frequency for its push notification system varies, with polling calls potentially consuming the majority of the backend API gateway requests. How could they improve their push notification system to lighten the load on their APIs and deliver a better user experience?