一、概念题（共30分）

1. k fault tolerant (3pt)

2. availability, Reliability (3pt)

3. recovery line (3pt)

4. continuous consistency ranges (3pt)

5. eventual consistency (3pt)
6. happens-before relation (3pt)

7. finger table (3pt)

8. out of band data (3pt)

9. MapReduce (3pt)

10. a scalable system (3pt)
二、简答题（共70分）
1. Q: What is the difference between a vertical distribution and a horizontal distribution? (5pt)

2. Q: Is a server that maintains a TCP/IP connection to a client stateful or stateless? (5pt)

3. Q: One way to handle parameter conversion in RPC systems is to have each machine send parameters in its native representation, with the other one doing the translation, if need be. The native system could be indicated by a code in the first byte. However, since locating the first byte in the first word is precisely the problem, can this actually work? (5pt)

4. Q: Routing tables in IBM WebSphere, and in many other message-queuing systems, are configured manually. Describe a simple way to do this automatically. (5pt)
5. Q: Is an identifier allowed to contain information on the entity it refers to? (5pt)

6. Q: When a node synchronizes its clock to that of another node, it is generally a good idea to take previous measurements into account as well. Why? Also, give an example of how such past readings could be taken into account. (5pt)

7. Q: Ricart and Agrawala's algorithm has the problem that if a process has crashed and does not reply to a request from another process to access a resources, the lack of response will be interpreted as denial of permission. We suggested that all requests be answered immediately to make it easy to detect crashed processes. Are there any circumstances where even this method is insufficient? Discuss. (5pt)

8. What kind of consistency would you use to implement an electronic stock market? Explain your answer. (5pt)
9. When using a lease, is it necessary that the clocks of a client and the server, respectively, are tightly synchronized? (5pt)

10. Q: Consider a nonblocking primary-backup protocol used to guarantee sequential consistency in a distributed data store. Does such a data store always provide read-your-writes consistency? (5pt)

11. Q: State-based leases are used to offload a server by letting it allow to keep track of as few clients as needed. Will this approach necessarily lead to better performance? (5pt)
12. With asynchronous RPCs, a client is blocked until its request has been accepted by the server. To what extent do failures affect the semantics of asynchronous RPCs? (5pt)

13. Q: Despite that GFS scales well, it could be argued that the master is still a potential bottleneck. What would be a reasonable alternative to replace it? (5pt)

14. Q: In NFS, attributes are cached using a write-through cache coherence policy. Is it necessary to forward all attributes changes immediately? (5pt)