

CS550 Fall 2019
Final Exam Study Guide

1. Intel and AMD are the major manufacturers of general purpose processors today (2019). What is the most number of cores that a high-end server processor from AMD or Intel will have?
2. What properties does a distributed system have?
3. Which of these are examples of distributed systems?
4. What is the difference between a 6-core processor and a 28-core processor from the Intel Scalable Processor lineup?
5. Why is it cheaper to build a distributed system than a centralized system for a given quantity of computing capacity, memory, and storage?
6. What is the difference between a process and a thread?
7. What is the difference between a software thread and a hardware thread?
8. Why do we need synchronization locks (e.g. mutex) in a language like C/C++?
9. What reasons are there to use a virtual machine for?
10. What reasons are there to NOT use a virtual machine for?
11. What is the primary use of the program "ssh"?
12. If you use the "chmod" command to set the permissions on a file to 777, what permissions are you setting on the file?
13. Define what "bash" is in Linux.
14. What mechanism is used to search unstructured overlays networks?
15. What is the worst-case scenario for locating the object in an unstructured overlay network in terms of number of nodes visited, where n is the number of nodes?
16. Can unstructured overlay networks have network partitions?
17. What is one advantage of a non-blocking form of Send? What about one advantage of blocking communication?
18. What type of applications should use an unbuffered communication approach?
19. Is super-linear speedup is possible in practice? Explain your answer.
20. T/F: Multi/many-core is hard for architecture people, but pretty easy for software people.
21. T/F: Multi/many-core made it possible for Google to search the web.
22. T/F: Email uses persistent communication
23. T/F: Network routers use transient communication
24. T/F: Asynchronous communication allows for more scalable systems
25. T/F: Synchronous communication allows for simpler implementation
26. T/F: Asynchronous communication can be implemented using synchronous communication
27. T/F: Remote procedure call (RPC), remote method invocation (RMI), and web services (WS) are all abstractions to allow interprocess communication across a network to access remote resources.
28. T/F: Network sockets using TCP/IP are preferred over RPC/RMI/WS due to ease of use.
29. What are the key motivations for process migration?
30. What is a disadvantage of process migration across cores of a processor?
31. What are problems with centralized scheduling?
32. What are problems with distributed scheduling?
33. What is work stealing? What parameters influence the scalability of work stealing most (we discussed this in detail in the lecture)?
34. How is the namespace of a POSIX file system organized?
35. What is the best data structure to implement a POSIX file system metadata management that would minimize the memory/storage footprint and maximize performance?
36. What does the domain name space service do?
37. What is LDAP used for?
38. Why would someone choose NFS over GPFS?
39. T/F: Clock synchronization is hard in distributed systems.
40. What is today's approach to clock synchronization in distributed systems?
41. What are the advantages to logical clocks vs. real clocks?
42. What is the primary difference between structured and unstructured overlay networks?
43. What mechanism is used to search unstructured overlays? What is the worst case scenario for locating the object in question in terms of number of nodes visited, where n is the number of nodes?

44. Can unstructured overlay networks have network partitions?
45. What are advantages of a non-blocking form of Send?
46. Name some advantages of distributed systems over centralized systems.
47. In many layered protocols, each layer has its own header. Surely it would be more efficient to have a single header at the front of each message with all of the control in it than to have so many separate headers. Why is this not done?
48. In this problem you are to compare reading a file using a single-threaded file server with a multi-threaded file server. It takes 16 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming the data are in the block cache. If a disk operation is needed (assume a spinning disk drive with 1 head), as is the case one-fourth of the time, an additional 32 msec is required. What is the throughput (requests/sec) if a single threaded server is required, rounded to the nearest whole number? What is the throughput (requests/sec) if a multi-threaded server is required with 4-cores and 4-threads, rounded to the nearest whole number?
49. About how many cores/threads are expected to be in future commodity processors in the next 5 years?
50. What is an advantage to blocking communication?
51. In what situation would non-blocking communication improve performance most?
52. What type of application should use an unbuffered communication approach?
53. What overheads are you limiting when using buffered communication?
54. Name several unique feature of the TCP communication protocol.
55. Name a technique that can be used to handle out of order packets efficiently?
56. Identify the function that cannot be used to implement a concurrent server.
57. What does it mean for a system to be scalable?
58. Identify which system has a client-pull architecture.
59. Identify which system has a server-push architecture.
60. Which is not an OS state for a thread?
61. Why is threading useful on a single-core processor?
62. Identify what a thread has of its own (not shared with other threads):
63. What is the advantage of OpenMP over PThreads?
64. Do more threads always mean better performance?
65. What was the flaw in Amdahl's Law that made it inaccurate?
66. How does RPC/RMI/WS make improvements over sockets/threads?
67. What is the purpose of parameter marshaling?
68. What is an advantage to a connectionless oriented protocol compared to a connection-oriented protocol?
69. How are web services (WS) similar to RMI/RPC?
70. What form of communication is email considered?
71. What form of communication is UDP considered?
72. What form of a data stream is needed to handle archival video streaming?
73. What form of a data stream is needed to handle live 2-way audio streaming?
74. What is the purpose of code migration?
75. What is the purpose of the scheduler in a distributed system?
76. You have a cluster with 1000 compute nodes. You have a centralized scheduler that can schedule 10 tasks per second. What is the smallest granularity of task lengths that your scheduler can support in order to achieve high system utilization?
77. You have a cluster with 1000 compute nodes. You have a distributed scheduler that has 1000 schedulers, and each scheduler can process 1 task per second. What is the smallest granularity of task lengths that your scheduler can support in order to achieve high system utilization?
78. In distributed scheduling, why are dynamic neighbors better than static ones?
79. A user is in front of a browser and types in www.google.com, and hits the enter key. Think of all the protocols that are used in retrieving and rendering the Google logo and the empty search box. Select all that could apply (partial credit will be given).
80. Identify which algorithm is suitable for generating a unique name across a distributed system (hint: these operations would be done on each node, and do not involve any network communication)?
81. Why is time synchronization hard?
82. What makes time synchronization inaccurate?
83. Name a protocol used to synchronize time.

84. When synchronizing clocks, why is it a bad idea to change clocks infrequently a large amount?
85. What makes logical clocks more scalable than time synchronization?
86. What are the advantages of centralized locks?
87. What makes distributed locks hard to implement?
88. How is replication different than caching?
89. Why do we need replication/caching?
90. What guarantees does eventual consistency give?
91. Which operation(s) are most difficult in eventual consistency?
92. Why did processors from the 1980s not need cache-coherent processors?
93. Why is it sometimes so hard to hide the occurrence and recovery from failures in a distributed system?
94. Describe precisely what is meant by a scalable system.
95. Consider again an unstructured overlay network in which every node randomly chooses c neighbors. To search for a file, a node floods a request to its neighbors and requests those to flood the request once more. How many nodes will be reached?
96. Suppose that you could make use of only transient asynchronous communication primitives, including only an asynchronous receive primitive. How would you implement primitives for transient *synchronous* communication?
97. How could you guarantee a minimum end-to-end delay when a collection of computers are organized in a (logical or physical) ring?
98. Would it make sense to limit the number of threads in a server process?
99. Constructing a concurrent server by spawning a process has some advantages and disadvantages compared to multithreaded servers. Please explain.
100. Explain the difference between a hard link and a soft link in UNIX systems. Are there things that can be done with a hard link that cannot be done with a soft link or vice versa?
101. Consider a distributed file system that uses per-user name spaces. In other words, each user has his own, private name space. Can names from such name spaces be used to share resources between two different users?
102. Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?
103. One of the modern devices that have (silently) crept into distributed systems are GPS receivers. Give examples of distributed applications that can make use of GPS information.
104. 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.
105. Explain in your own words what the main reason is for considering relaxed consistency models.
106. What kind of consistency would you use to implement an electronic stock market? Explain your answer.
107. Consider a personal mailbox for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency would be most appropriate?
108. Name a few advantages and disadvantages of using centralized servers for key management.
109. How does the probability of failure of an entire distributed system (assuming all components are necessary to function properly) change as the number of independent components in the system grows?
110. What components in a computer system do we know how to make resilient, and what technique is used?
111. In the Two Army Problem and assuming an asynchronous unreliable communication channel, can the two generals ever reach consensus with 100% guarantee?
112. Data resilience through forward error correcting codes is an example what type of recovery mechanism? What is the advantage of Forward Recovery in storage systems?
113. Data resilience through replication is an example what type of recovery mechanism? What are some advantages of this recovery mechanism?
114. RAID (redundant array of inexpensive disks) is an example what type of recovery mechanism?
115. What is the technique called that allows applications to restart and recover from an intermediary point after the start of the application?