**15-440 Final Exam Review Questions**
**Kesden/Summer 2014**

## Data-Intensive Scalable Computing (DISC)

1. How would it impact Hadoop's performance if it were to be implemented over AFS instead of HDFS? Why?

2. The output of a Mapper is written into the local filesystem instead of the global filesystem. Why?

   Your answer should explain both why writing into the global file system would be undesirable as well as why it would be of minimal benefit.

3. Why does Hadoop sort records en route to a Reducer? How would it affect things if these records were processed by the Reducer in the order in which they were received from the various Mappers?

4. What happens if a Mapper or Reducer fails?

**Distributed Computing Models**

5. If processor allocation is optimal, is it possible that migration will subsequently improve system performance? If not, why not? If so, how?

6. Why are periodic broadcast advertisements often considered to be a poor way of communicating information about resource availability? What is the risk?

7. Please explain two commonly used alternatives to the advertisements mentioned above and the relative costs and benefits.

8. When is a shared memory programming model typically a better approach than message passing? Why? Is anything special required or acutely helpful? Why?

9. What type of tasks would you expect to benefit from Hadoop over OpenMPI? Why?

**Distributed File Systems**

10. In class we observed that AFS and NFS manage consistency differently. AFS issues callbacks upon updates. NFS validates the client cache periodically.

    (a) Do either of these mechanisms eliminate the window of vulnerability? If so, how? If not, is possible to eliminate the window of vulnerability? Why or why not?

    (b) Which mechanism will result in less network traffic in the event that many dozens of clients have the same file open for high-frequency random-access reads?

11. Consider Coda's whole-file semantics, including the behavior of open() and close as well as of caching.

    (a) How does it enable disconnect and weakly connected operation, e.g. use of the file system even when network connectivity is poor or non-existent?

    (b) In what ways does it limit file system performance and capability?

**Design: Special Purpose Distributed File Systems**

12. Consider the design of a distributed file system for light-weight mobile devices, especially smart phones, such as common iPhone, BlackBerry, Android, and Windows Mobile devices.

    (a) The file system should be robust without involving any off-line backups, e.g. tape.
    (b) It should view the device's storage as a cache for the actual data, but not necessarily the primary copy.
    (c) It should assume a workload similar to what we see with these devices now, e.g. notes, calendars, photos
    (d) It should support user data, not necessarily user programs
    (e) It should facilitate the migration from one device to another and the use of the data on at least one host computer
    (f) User data should stay private to that user, but should be very quick to access, especially from the device, itself.

    Assume the following properties of the systems:

    (a) Files are generally "small", e.g. text messages, notes, short documents, and cellphone photos, but not long hi-res videos, databases, etc.
    (b) Latency is very high, e.g. 500mS
    (c) Bandwidth is modest, but not terrible for downloads, e.g. 1Mbps
    (d) Bandwidth for uploads can be outright bad, e.g. 0.20 Mbps
    (e) Although the data can be changed from multiple hosts, it will not be accessed concurrently, since there is only one user.
    (f) The storage on the each of the mobile device and host are large relative to the user's mobile needs
    (g) Files access generally requires the whole file, rather than random access to only some part of it.
    (h) Off-device storage is "Free"
    (i) The wired Internet is "Fast and wide"

(a) What are the most important challenges presented by these requirements?

(b) Please describe the architecture of your solution, include especially descriptions of caching, replication, checkpointing, and the protection of privacy.

**Virtual Machines**

13. Please identify and describe three way that virtual machines ease the implementation and management of distributed systems

14. What are the differences and relative advantages and disadvantages of simulators, emulators, and virtualizers?

15. What is a hypervisor? What role do they play in the management of computing.

16. Consider computation, disk I/O, network I/O, human input such as via keyboards and mice, and GPU computation. Please rank these in terms of the penalty encountered through virtualization on same-architecture virtual machines (consider VMware or KVM, if you'd like). Then, please explain the underlying reasons that each type of function experiences the relative VM penalty that it does.

**Cryptography, Cryptographic Protocols, Privacy, and Authentication**

17. What is the single biggest challenge in authentication? Why?

18. What are the relative costs and benefits of symmetric and asymmetric cryptography?

19. What is the role of a session key? Why is it helpful?

20. What is the role of a nonce? Why is it helpful?

21. In what critical way is a biometric key higher risk than a key of the user's choosing?

22. What about protocols, such as Kerberos, enables the user to present a ticket to the service – and the service to trust that it hasn't been forged by the presenter? Why does this work?

**Anonymous Communication**

23. What is the difference between anonymous communication and private communication?

24. Consider onion routing, why must the rout be selected in advance by the user, rather than by the network hop-to-hop?

25. Consider onion routing, what is accomplished during the initial “telescoping out” phase? Why is this important?

26. Please identify two clear weaknesses/brittle points in onion routing? In other words, if you wanted to identify users of an onion network, where would you start?