15-440 Homework #3
Kesden/Fall 2013

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. How is the failure of a Mapper or Reduce managed?

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.

    i. The file system should be robust without involving any off-line backups, e.g. tape.
    ii. It should view the device's storage as a cache for the actual data, but not necessarily the primary copy.
    iii. It should assume a workload similar to what we see with these devices now, e.g. notes, calendars, photos
    iv. It should support user data, not necessarily user programs
    v. It should facilitate the migration from one device to another and the use of the data on at least one host computer
    vi. 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:

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

Security

13. Consider *Onion Routing* and the case of a compromised router. In this worst case, will it know the source of the message, the destination of the message, both? Why?

14. Consider *Onion Routing,* why is the path chosen in advance by an agent of the client, rather than the network hop-by-hop?

15. Kerberos enables a client to communicate credentials to a server. What guarantees that the server will be able to trust these credentials?

16. Kerberos uses *symmetric/secret key* cryptogrophy, rather than *asymmetric/public key* cryptopgraphy. Why?