CSE 291 Homework #2

Spring 2016 (Kesden)

Data-Intensive Scalable Computing (DISC)

1. HDFS is implemented as a user-level file system vs an in-kernel file-system. (a) What is the advantage of this in the context of Hadoop?
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?
5. In a typical Map-Reduce graph algorithm, what data structure is used to represent the graph? Why?
6. In a typical Map-Reduce graph algorithm, how many Map-Reduce phases are typically necessary before the graph can be traversed? Why?

Processor Allocation and Migration

1. If processor allocation is optimal, is it possible that migration will subsequently improve system performance? If not, why not? If so, how?
2. Why are periodic broadcast advertisements often considered to be a poor way of communicating information about resource availability? What is the risk?
3. Please explain two commonly used alternatives to the advertisements mentioned above and the relative costs and benefits.

Distributed File Systems

1. In class we observed that AFS and NFS manage consistency differently. AFS issues callbacks upon updates. NFS validates the client cache periodically.
2. 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?
3. 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?

Security

1. 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?
2. Consider *Onion Routing,* why is the path chosen in advance by an agent of the client, rather than the network hop-by-hop?
3. Kerberos enables a client to communicate credentials to a server. What guarantees that the server will be able to trust these credentials? (Covering this on Monday)
4. Kerberos uses *symmetric/secret key* cryptogrophy, rather than *asymmetric/public key* cryptopgraphy. Why? (Covering this on Monday)