Questions on Reading: MapReduce Paper & Introduction to Distributed Systems

1. Name two possible uses for MapReduce besides WordCount. Do not list problems that calculate a specific attribute of the words in a document corpus, eg, DistributedGrep or WhitespaceCount. In your example, what are the map and reduce functions (including inputs and outputs)?
2. What kind of constraints does MapReduce place on its problem domain? Said in another way, what applications would *not* work in MapReduce? eg, could you use MapReduce's for Amazon's shopping cart? How about SETI@Home? An MMORPG?
3. Besides MapReduce, what is another possible strategy for parallelizing computation over a large document corpus? What are the strengths and weaknesses of that system relative to MapReduce?
4. When designing a distributed system from the ground up, what considerations would you make to improve the reliability of the networking component?
5. What are the scarce resources in a MapReduce system (eg, disk space, personnel, CPU)? What is the absolute limit for adding those resources into a MapReduce system (eg, if skilled personnel were the limiting resource, then the hard limit would be the number of skilled laborers)? Assuming that limit were reached, how could MapReduce be tweaked to reduce dependency on that resource?
6. "Introduction to Distributed System Design" lists seven characteristics of a reliable distributed system: fault-tolerant, highly available, recoverable, consistent, scalable, predictable performance and secure. For each of these, write a few sentences describing how MapReduce has been designed to exhibit that specific characteristic.
7. How does MapReduce improve the reliability of distributed systems?
8. Consider the example of shuffling the sentences in a document set by assigning each sentence a randomly-generated key. While this solution works in the theoretical MapReduce model, it is not guaranteed to work in practice. Why?
9. Is TCP/IP optimal for a reliable distributed system? Why do you think the Internet relies so heavily on it? In a perfect world, how would you change it to work in a closed distributed system?