1. What is the theme of this research article?
   Answer: compare scale-up vs. scale-out
   They reached a conclusion that scale-up architecture could be better than scale-out in terms of the performance and other metrics.

2. In the introduction, authors mentioned two reasons why scale-up could be better than scale-out architecture. What are these two reasons?
   Answer: First, evidence suggests that the majority of analytics jobs do not process huge data sets.
   Second, hardware price trends are beginning to change performance points. Thus a scale-up server can now have substantial CPU, memory, and storage I/O resources and at the same time avoid the communication overheads of a scale-out solution.

3. What is the default assumption for Hadoop jobs today?
   Answer: scale-out is the only configuration that matters. Scale-up performance is ignored and in fact Hadoop performs poorly in a scale-up scenario. (at the end of page 1)

4. What are the four metrics used in this paper to compare scale-up with scale-out?
   Answer: performance, cost, energy, and server density. (at the beginning of page 2)

5. Why are the job sizes important for the comparison between scale-up and scale-out?
   Answer: job sizes are not large, thus broadly, whether judged on input data size or CPU time, a large number of jobs would seem to fit comfortably within a "big memory" scale-up server with 32 cores and 512GB of memory.
   Especially given the ever-decreasing price per gigabyte of DRAM, it seems worth while to consider whether such jobs are better fitted to a scale-up server rather than a scale-out cluster. (Second last paragraph of page 2)

6. What are the five optimization categories for Hadoop in this paper and what are only applied in scale-up configuration?
   Answer: storage, concurrency, network, memory and reduce-phase optimization
   Only applied in scale-up: local file system and reduce-phase optimization (2nd paragraph in Section 3)

7. What did authors do for storage optimization?
   Answer: using SSDs (Hence for our scale-up configuration we store the inputs on SSDs and access them via the local file system.) Authors also mentioned other possibilities (NAS, SAN, or cloud storage).

8. Why is increasing heap size helpful for improving the performance?
   Answer: By default, each Hadoop map and reduce task run in a JVM with a 200MB heap within which they allocate buffers for in-memory data. When the buffers are full, data is spilled to storage, adding overheads.
   We note that 200MB per task leaves substantial amounts of memory unused on modern servers. By increasing the heap size for each JVM (and hence the working memory for each task), we improve performance.

9. What are the three optimizations to speed up shuffle?
   Answer: First, authors modified Hadoop so that shuffle data is transferred by writing and reading the local file system; the default is for reducers to copy the data from the mappers via http.
   Second, they removed the limit on the number of concurrent connections that is allowed.
   Finally, they observed that the scale-up machine has substantial excess memory after configuring for optimal concurrency level and heap size. They use this excess memory as a RAMdisk to store intermediate data rather than using an SSD or disk-based file system.

10. When authors deployed their jobs on Cloud using Azure, they found performance decrease in scale-out configuration. Why did that happen?
    Answer: contention for shared network bandwidth with other cloud users. (first paragraph of page 10)

11. The authors used TeraSort to show the limitations of Scale-up. What were their conclusions?
    Answer: First, even with “big memory”, the scale-up configuration can become memory-bound for large jobs.
    Second, for TeraSort, scale-out is competitive at around the 100GB mark with current hardware. (at the end of page 10 and at the beginning of page 11)

12. Based on what you learned from this research article, what kind of scenarios are best for scale-up?
    Answer: small jobs so they can fit in memory; others mentioned in the article (e.g., using SSDs and local file system)

13. Although authors did not explicitly mention in the article, what kind of scenario do you think may be good for scale-out?
    Answer: large jobs so they cannot fit in memory any way in scale-up; possibly, jobs that do not require much shuffle (data transfer among different servers), so that network may not be the bottleneck.