Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing

NOTE: Your slides/presentation need to cover the assigned sections and questions in a clear and well-organized manner. You are allowed to borrow contents from other resources, such as online slides, as long as you acknowledge them. For a slide that covers a given question, please print the question on the slide. However, you don't have to answer the question using a long paragraph of text on the slides. Instead, use bullet points, graph, animation, or oral explanation to answer the question. In your Q&A report, use text to more thoroughly answer the questions.

You only need to cover contents until Section 5.1.

(1) "…*individual RDDs are immutable*…" What does it mean by being "immutable"? What benefits does this property of RDD bring?

(2) When an RDD is being created (new data are being written into it), can the data in the RDD be read for computing before the RDD is completed created?

(3) *"This allows them to efficiently provide fault tolerance by logging the transformations used to build a dataset (its lineage) rather than the actual data." "To achieve fault tolerance efficiently, RDDs provide a restricted form of shared memory, based on coarse-grained transformations rather than fine-grained updates to shared state."* Why does using RDD help to provide efficient fault tolerance? Or why does coarse-grained transformation help with the efficiency?

(4) What is difference between transformation and action?

(5) "*In addition, programmers can call a persist method to indicate which RDDs they want to reuse in future operations*." What's the consequence if a user does not explicitly request persistence of an RDD?

(6) Explain Figure 1 about a lineage graph.