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

NOTE: Your slides/presentation only need to cover background information necessary to answer the given 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. The presentation should be mostly question-focused and proceed mostly in a Q&A format. Please include the questions in your slides. Don’t write detailed answers in the slides and read them to the class. Instead, use bullet points, graphs, or animations to explain your answers to the class.

In your Q&A report, use text to more thoroughly answer the questions. Include a short paragraph at the beginning of the report to summarize the paper.

(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 completely 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) “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?

(5) Explain Figure 1 about a lineage graph.