Transcript name: MapReduce – Part 3 – Mergesort/Shuffle

English
Let us examine how the map and reduce operations work in sequence on your data to produce the final output. In this case, we have a job with a single map step and a single reduce step. The first step is the map step. It takes a subset of the full data set called an input split and applies to each row in the input split an operation you have written, such as the "multiply the value by two" operation we used in our earlier map example. There may be multiple map operations running in parallel with each other, each one processing a different input split. The output data is buffered in memory and spills to disk. It is sorted and partitioned by key using the default partitioner. A merge sort sorts each partition. The partitions are shuffled amongst the reducers. For example, partition 1 goes to reducer 1. The second map task also sends its partition 1 to reducer 1. Partition 2 goes to the other reducer. Each reducer does its own merge steps and executes the code of your reduce task. For example, it could do a sum like we used in the earlier reduce example. This produces sorted output at each reducer. This lesson is continued in the next video.