Spark implementation of slowly changing dimention

Implemented a slowly changing dimention type 2 using Scala Spark.

**Problem :** Given two dataset we have to derive a scd type 2 resultset. (At point we need to maintain a history of changes for the key with marking latest/current records based on the modified date with tagging new field ‘Y’)

Step 1 :  Creating two RDD from two textfiles. One text file holds base and other holds incremental records.

Step 2 : Creating a key,value(paired) RDD for the two dataset.

Step 3 : Subtract base paired RDD  with Incremental/delta paired RDD based on key which is ID in this example.

Step 4:  Tag the Resultset of step 3 with new field Y since no history will be found as resultset derived is subtracted from delta. And convert it to string.

Step 5 : Repeat the same steps by interchanging the subtract between base and delta paired rdd.

Step 6 : At end of step 5 we derived two dataset one will hold records that are not present in delta and present only in base. other is vice versa. Union these two resultset.

Step 7: Resultset dervied from Step 6 will hold no common keys and its elements from both delta and base dataset.

Step 8 :  Follow the steps mentioned as part of **Spark Use Case #5**to derive common keys and its element from base and delta datasets.

Step 9 : Join the base paired rdd with delta paired rdd created in step 2.

Step 10 :  Use map transformation in the joined dataset to pick the latest records for that row.

Step 11 : Use reduce transformation on resultset of step 10 to derive the latest record for the Key present in both dataset.

Note : Here we can avoid reduce step if we use cogroup which will give all the corresponding records for the Key from both dataset. so we can just use map transformation to arrive at the latest record for the key. Will be solved in next upcoming use cases.

Step 12 : Take the resultset of step 11 and tag the value for the key with new field ‘Y’ since it is compared on all the other common keys to find as a latest record based on modified date and convert it to string otherwise it will be a Array[String] which is not clean for saving/displaying.

Step 13 :  Rearrange the resultset of step 11 with key as Id and modified date and similarly for resultset arrived from step 8. So that we can subtract to find the old records for the key ignoring the latest record and mark the resultset with N as a new field and convert it to string.

Step 14 :  Unioning the resultset of Step 13(containing history of records for the Id without latest record) , the resultset of Step 12(containing latest record for the Id which was present in the both base and delta), the resultset of Step 6(holds the records which was not common in the two datasets)

Step 15 : Final resultset derived will be of scd type 2.