I have carried out the 150 test cases and 20 test cases given me the positive result.

When I combined the all those changes ,I got only 6 test cases that are giving me the good improvement.

**FINAL CODE CHANGES**

**1: Adding the compression technique to the files:-**

**Newer Code:-**

SET io.compression.codec.lzo.class com.hadoop.compression.lzo.LzoCodec

SET pig.tmpfilecompression true

SET pig.tmpfilecompression.codec lzo

**Details:-**

The pig stores the temporary output of the aggregate functions in the form of the files.

By using the compression technique we can compress the temporary files.

Advantages of the LZO compression:-

* Saving the hard disk space
* Speedy transfer or copy of the data

**2: Setting Mappers in the script:-**

**New Code:-**

SET mapred.min.split.size 524288;

**Details:-**

More number of the mapper more speed.

**3: Changed join rev\_dtc\_file11:-**

**Earlier Code:-**

rev\_dtc\_file11 = JOIN rev\_dtc\_file BY REV\_ITEM, hier\_item\_data\_required BY H\_ITEM\_ID using 'skewed';

**Newer Code:-**

rev\_dtc\_file11 = JOIN hier\_item\_data\_required BY H\_ITEM\_ID, rev\_dtc\_file BY REV\_ITEM;

**Details:-**

I have compared the data of the two files, and kept larger file last.

Generally, we have to adjust the files according to speed.

Also, I have removed the skewed keyword as data was not skewed.

**4: Changed the input\_join :-**

**Earlier Code:-**

input\_join = JOIN input\_req\_gen BY ITEM\_ID, sbt\_ord\_join1 BY sbt\_vend\_join1::filtered\_vend\_packs::shc\_item\_id;

**Newer Code:-**

input\_join = JOIN input\_req\_gen BY ITEM\_ID, sbt\_ord\_join1 BY sbt\_vend\_join1::filtered\_vend\_packs::shc\_item\_id using 'replicated';

**Details:-**

I have added the replicated keyword to the join.

When we have one table small and one table large then a small table can be fit into the memory and we can use the replicated join.

* It Increases the speed

**5: Adding the filtered\_vend\_packs after loading the corresponding file only.**

**Moved Code after:-**

smith\_\_idrp\_vend\_pack\_combined\_data =

LOAD '$SMITH\_\_IDRP\_VEND\_PACK\_COMBINED\_LOCATION'

USING PigStorage('$FIELD\_DELIMITER\_CONTROL\_A')

AS ($SMITH\_\_IDRP\_VEND\_PACK\_COMBINED\_SCHEMA);

filtered\_vend\_packs = FILTER smith\_\_idrp\_vend\_pack\_combined\_data by (TRIM(ksn\_purchase\_status\_cd) != 'U') ;

**Details:-**

If we use the filter after loading the file , it increases the speed and performance of the job.

**6: Adding parallel to test\_data\_1234:-**

**Newer Code:-**

test\_data\_1234 = UNION t\_data\_1, t\_data\_2, t\_data\_3 PARALLEL 500;

I have added the PARALLEL keyword here to increase the number of the reducers for the union.

**Details:-**

Reducers do join, union , aggregation and combination operations.

More number of the reducers 🡪 More number of data will process 🡪 Speedy OperationSklk