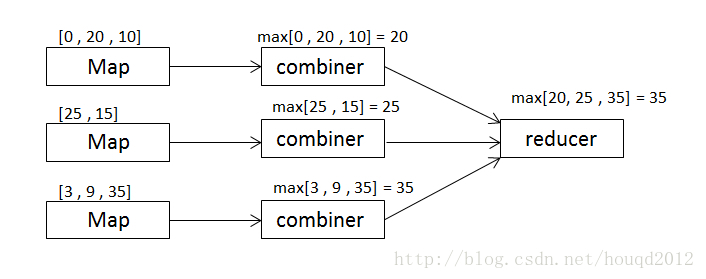
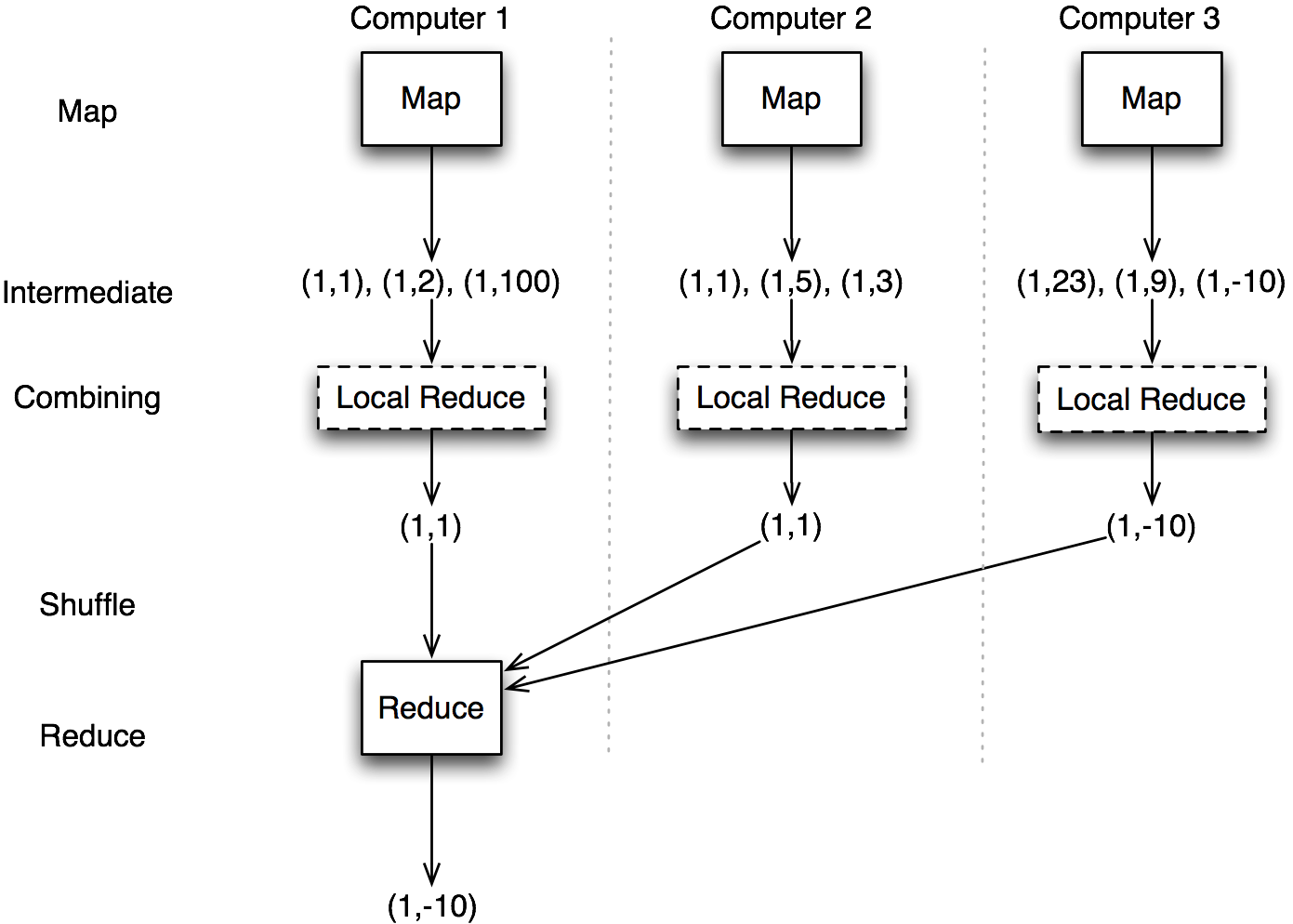
**COMBINER**

A Combiner, also known as a **semi-reducer,** is an optional class that operates by accepting the inputs from the Map class and thereafter passing the output key-value pairs to the Reducer class.



The main function of a Combiner is to summarize the map output records with the same key. The output (key-value collection) of the combiner will be sent over the network to the actual Reducer task as input.

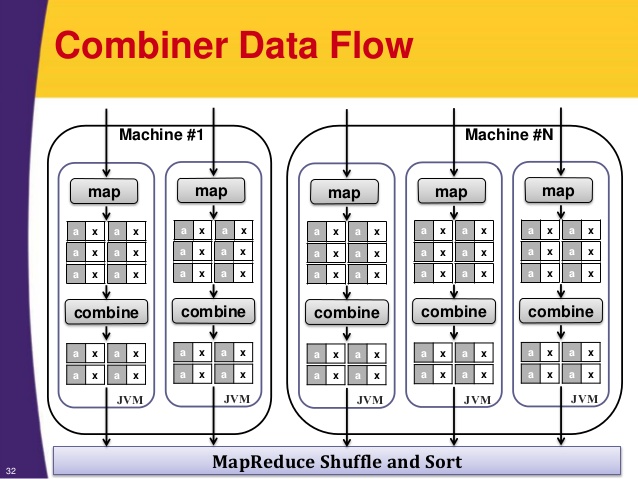
The advantages of combiner is, it minimizes the time taken for data transfer between mapper and reducer, combiner output is input to a reducer. Because of combiner reducer executes less amount of time which increases reducer performance, so let’s see how it works with diagram.



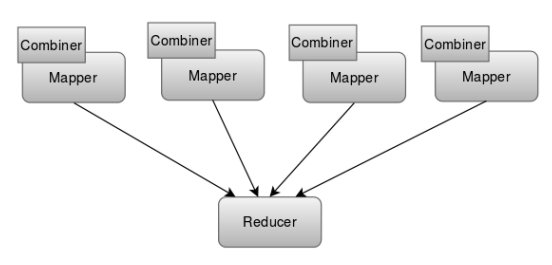
How Combiner Works?

Here is a brief summary on how MapReduce Combiner works −

* A combiner does not have a predefined interface and it must implement the Reducer interface’s reduce() method.
* A combiner operates on each map output key. It must have the same output key-value types as the Reducer class.
* A combiner can produce summary information from a large dataset because it replaces the original Map output.



Although, Combiner is optional yet it helps segregating data into multiple groups for Reduce phase, which makes it easier to process.

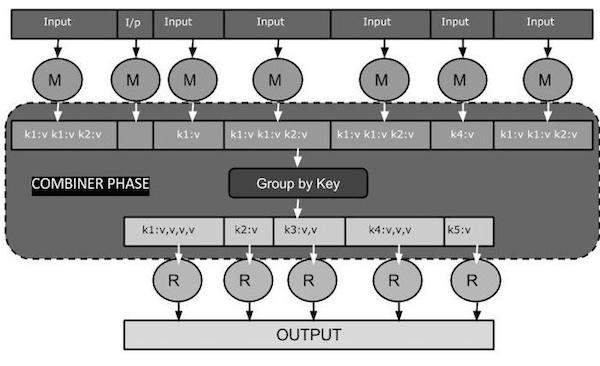


Although it implements the reducer class but it is executed at mapper level only. It helps in decreasing the number of iterations which are to be performed in reducer.

The main difference between combiner and reducer is in combiner we pass single entry for each whereas in reducer multiple entries are passed for a single key. Reducers can get data from multiple Mappers as part of the partitioning process. Combiners can only get its input from one Mapper.

### Combiner Phase:

The Combiner phase takes each key-value pair from the Map phase, processes it, and produces the output as **key-value collection** pairs.



Let us consider some scenario’s and understand more about the combiners:

**Example-1: Retail Store**

Store id, item id, qty\_sold, price/unit

Block 1

11, 101, 20, 10

11, 102, 10, 5

12, 101, 30, 10

12, 102, 20, 5

Block 2

11, 101, 50, 10

11, 102, 30, 5

12, 101, 40, 10

12, 102, 10, 5

**Problem statement- To find quantity sold for item across all the stores**

input key- row no.

input value- entire

output key- item id

output value- qty\_sold

**BEFORE COMBINERS**-

Mapper 1 Mapper 2

101, 20 101, 50

102, 10 102, 30

101, 30 101, 40

102, 20 102, 10

Sort + shuffle phase – framework

101, (20,30,50,40)

102, (10,20,30,10)

Reducer

101, (20+30+50+40)

102, (10+20+30+10)

Output

101, 140

102, 70

**AFTER COMBINERS-**

Mapper 1 Mapper 2

101, 20 101, 50

102, 10 102, 30

101, 30 101, 40

102, 20 102, 10

101, 30 101, 20

102, 40 102, 10

Combiner 1 Combiner 2

101, 80 101, 110

102, 70 102, 50

Sort + shuffle phase-framework

101, (80,110)=190

102, (70,50)=120

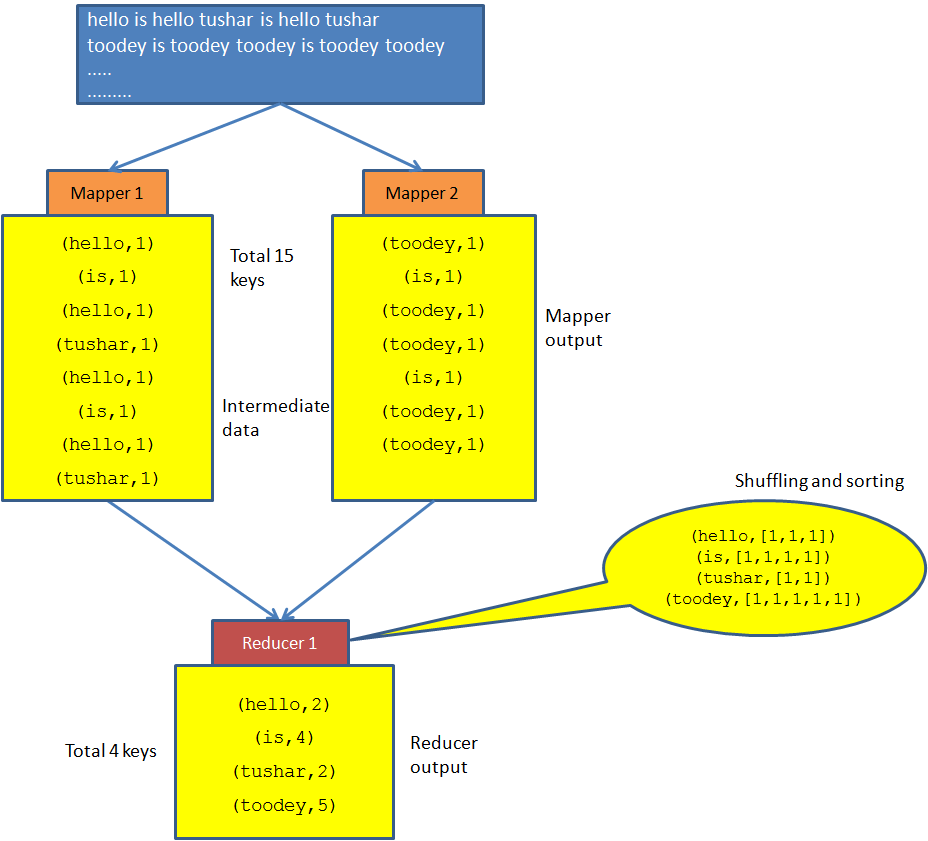
Sort + shuffle phase – framework

101, (20,30,50,40,30,20)=190

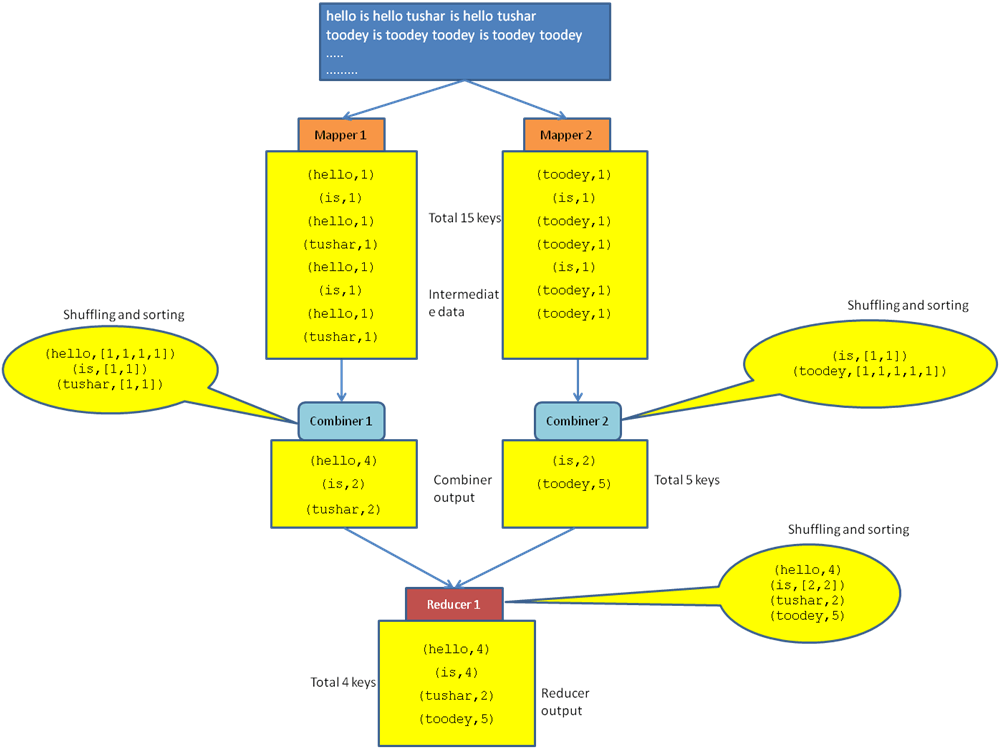
102, (10,20,30,10,40,10)=120

**Example-2:-**

**BEFORE COMBINER-**



**AFTER COMBINER-**



Combiners can only be used in specific cases which are going to be job dependent like finding the greatest among the given numbers, or highest price among given items for example:-

 Mapper 1 Mapper 2

101, 20 101, 50

102, 10 102, 30

101, 30 101, 40

102, 20 102, 10

101, 30 101, 20

102, 40 102, 10

Combiner 1 Combiner 2

101, 30 101, 50

102, 40 102, 30

Sort + shuffle phase-framework

101, (30,50)=50

102, (40,30)=40

Sort + shuffle phase – framework

101, (20,30,50,40,30,20)=50

102, (10,20,30,10,40,10)=40

Combiners will not work in the case of function of same function like count(count), avg(avg) means we might or might not get the right result. But count function does work with sum as count will act as combiner and sum as reducer.

1.Count function:

Prblm statmnt- count the no. of transactions for each key

Output:

101,6

102,6

Mapper 1 Mapper 2

101, 20 101, 50

102, 10 102, 30

101, 30 101, 40

102, 20 102, 10

101, 30 101, 20

102, 40 102, 10

Combiner 1 Combiner 2

101, 3 101, 3

102, 3 102, 3

Sort + shuffle phase-framework

101, (3,3)=2

102, (3,3)=2

The count should be 6 but it doesn’t gave right result. If we use sum function along with count we can get the required output.

Sort + shuffle phase-framework

101, (3,3)=(3+3)=6

102, (3,3)=(3+3)=6

2. Average function:

Prblm statmnt- Too find the average of items from each store

Mapper 1 Mapper 2

101, 20 101, 30

102, 20 102, 30

101, 40 101, 30

102, 20 102, 10

101, 40 101, 20

102, 40 102, 10

101, 30

102, 40

1nd COMBINER: 130/4 = 35

2nd COMBINER: 80/3 = 26.67

REDUCER: (26.67+35)/2 = 30.83

WITHOUT COMBINERS:

101, (20,40,40,30,30,30,20) = 210/7 = 30

In this case we are not getting accurate output even after the use of combiner. So according to the requirement we have to use combiners in our program.

Properties of Combiner:

* Aggregate functionality at Mapper side
* Reduction in network IO
* Has same input/output key/value data types as reducer input key & value data types
* Optional
* No guarantee how many times it will be called

