

Lesson 4

MapReduce Detailed Functions
(Grouping, Shuffling, partitioning,
Sorting, Combining, and Reducing)

Mapper Mapping and sorting

Key showRoomID1:

(Date1, JaguarSales); (Date1, ZestSales);

(Date2, JaguarSales); (Date1, ZestSales);

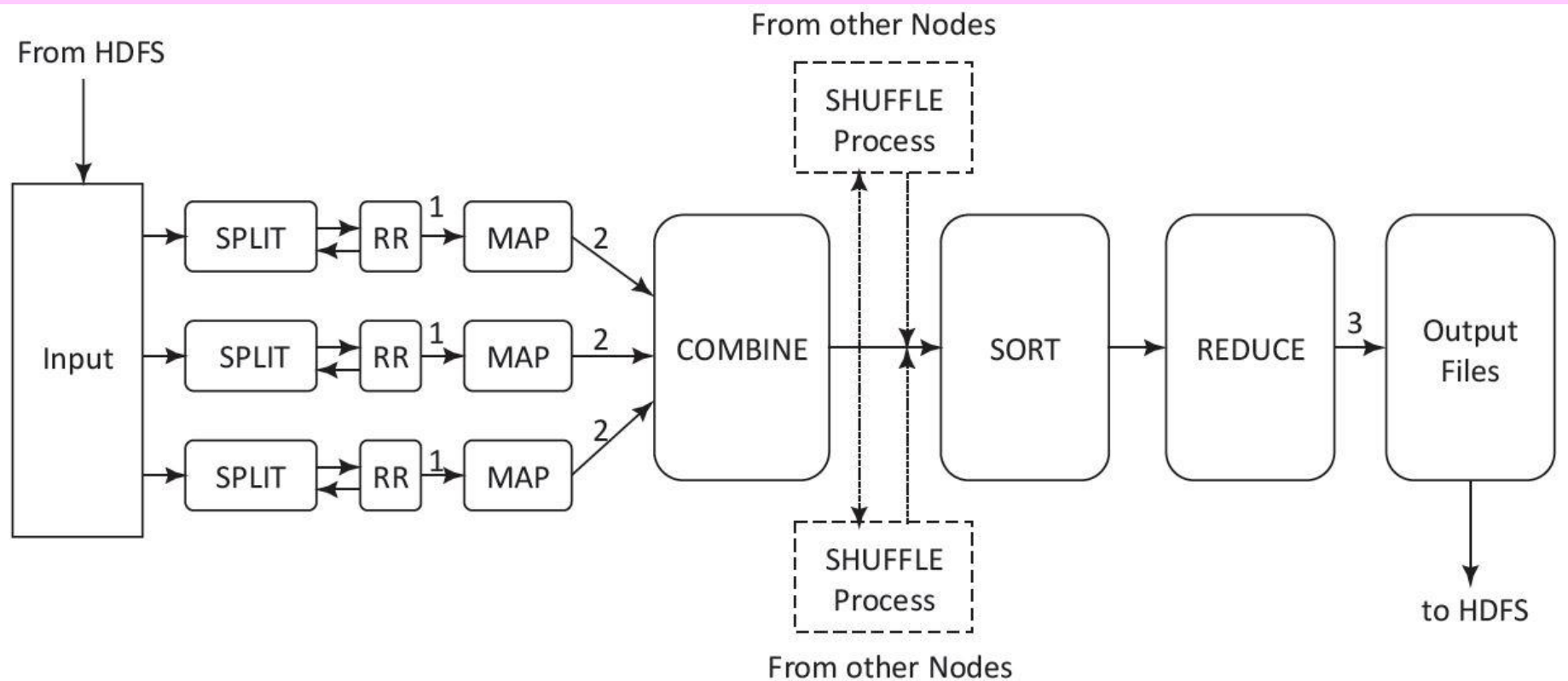
Key showRoomID2:

(Date1, ZestSales); (Date1, JaguarSales);

Reducer

- Combiner Key showRoomID1 Car Sales =
- Combiner Key showRoomID2 Car Sales =
- Reduce: Total all showroom Sales =
.....

Figure 4.6 MapReduce execution steps



RR – RecordReader

1 – Input key-value pairs

2 – Intermediate key-value pairs

3 – Final Key-Value Pairs

Example 2

- Chocolate flavours (FL1, FL2, FL3, FL4 and FL5) KitKat, Milk, Fruit and Nuts, Nougat and Oreo
- Sold by number of Automatic Chocolate Vending Machines (ACVMs)
- Dates, Date1, Date2,, Date31 in a month

Example 2

- Consider key-value pairs Input from API (ACVM1, Date1, KitKatSales);
(ACVM2, Date1, MilkSales);
(ACVM2, Date2, NougatSales);
(ACVM1, Date2, MilkSales);
(ACVM1, Date1, NougetSales);

Mapper Mapping and sorting

Key ACVM1:

(Date1, KitKatSales); (Date2,
MilkSales); Date1, NougetSales;

Key ACVM2:

(Date1, MilkSales); (Date2,
NougatSales)

Reducer

- Combiner Key ACVM1
ChocolatesSales =
- Combiner Key ACVM2
ChocolatesSales Sales =
- Reduce: Total Sales =

Mapping and Grouping by Keys

1. When the map task completes—
 - Grouping the key-values of the Mapper output using Shuffle process aggregates (combines) all the Mapper outputs
 - The v_2 appends in a list of values.

“Group By” operation on intermediate keys creates new v_2 .

Shuffle and sorting phases

2. All pairs with the same group key (**k2**) collect and group together, creating one group for each key
3. Shuffle output format will be a **List of <k2, List(v2)>**.

Partitioning and Reducer Inputs

- A different subset of the intermediate key space assigns to each reduce node
- These subsets of the intermediate keys (known as "partitions") are the inputs to the reduce tasks

Partitioning

- Partitioner— is an optional class. Partitions functions as the semi-mappers in MapReduce
- MapReduce driver class can specify a Partitioner.

Partitioning

- A partition processes the output of map tasks before submitting it to reducer tasks
- Partitioner function executes on each machine that performs a map task

Partitioning

- An optimization in MapReduce that allow for **local partitioning** before reduce-task phase
- Typically, the same codes implement the Partitioner, Combiner as well as reduce () functions

Partitioner

- Functions for Partitioner and sorting functions at the mapping node
- A Partitioner split the map output records with the same key

Example 1: Mapping, Grouping-by Keys and sorting by map ()

```
{ Key showRoomID1, Date1 },  
    JaguarSales, ZestSales);
```

```
{ Key showRoomID1, Date2 }  
    (JaguarSales, ZestSales);
```

```
{ Key showRoomID2, Date1 },  
    (JaguarSales, ZestSales);
```


Reducer

- Reduce: Total Car Sales Number =
.....on all dates Date1 to Date31 and
all showrooms and all models

Mapper Mapping of Two keys and sorting

Key { **ACVM1**, **Date1** }:

(FL1Sales, FL2Sales);

Key { **ACVM1**, **Date2** } : (FL2Sales,
FL3Sales)

Key { **ACVM2**, **Date1** } : (),

Key { **ACVM2**, **Date2** } : (FL3Sales)

Scuffle and Sorting

- Combiner Key ACVM1
ChocolatesSales =on all dates
Date1 to Date31
- Combiner Key ACVM2
ChocolatesSales Sales =on all
dates Date1 to Date31

Reducer

- Reduce: Total Sales =on all dates Date1 to Date31 and all ACVMs

Figure 4.7 MapReduce processing steps in ACPAMS application

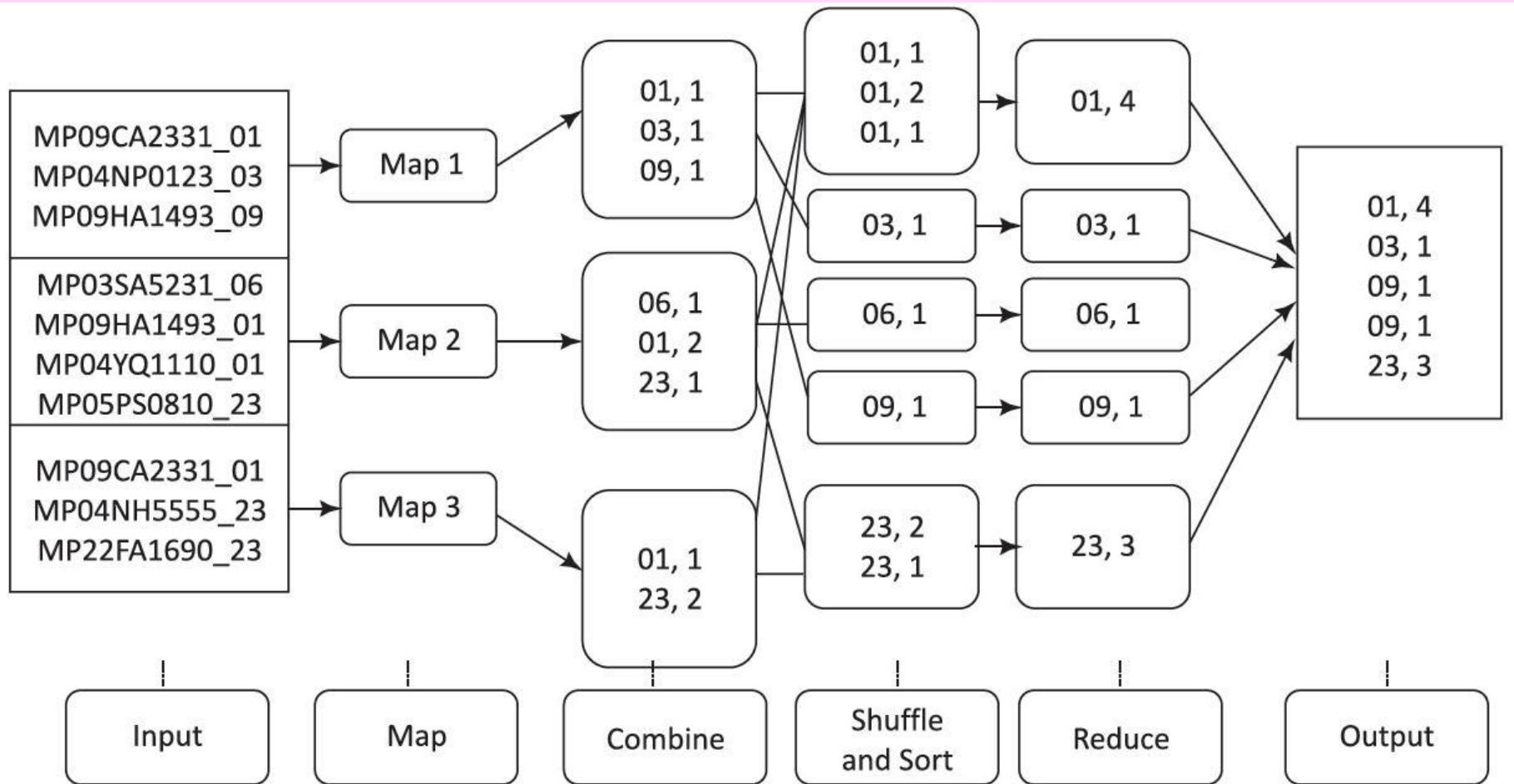


Figure 4.8 Sample Code for a map and reduce objects instance of Reducer Class

```
class Mapper {
    method Map (file id a; file f) {
        for all term i ∈ file f do {
            t = Substring (i, 2, After_)
            Emit (term t, count 1)}}}

class Reducer {
    method Reduce (term t, counts [c1, c2,....]) {
        sum ← 0
        for all count c ∈ counts [c1, c2, .....] do {
            sum ← sum + c}
        Emit (term t, count sum)}}}
```

Summary

We learnt :

- Grouping by Keys,
- Partitioners
- Shuffle and Sorting
- Combining
- Reducing

End of Lesson 4 on
MapReduce Detailed Functions
(Grouping, Shuffling, partitioning,
Sorting, Combining, and Reducing)