Big data analytics

Big Data Analytics (University of Mumbai)

Relational Algebra Operations using Mapreduce

→ Some of the relational algebra operations are:

1) Selection
2) Projection
3) Intersection
4) Union
5) Difference
6) Natural join
7) Grouping & Aggregation.

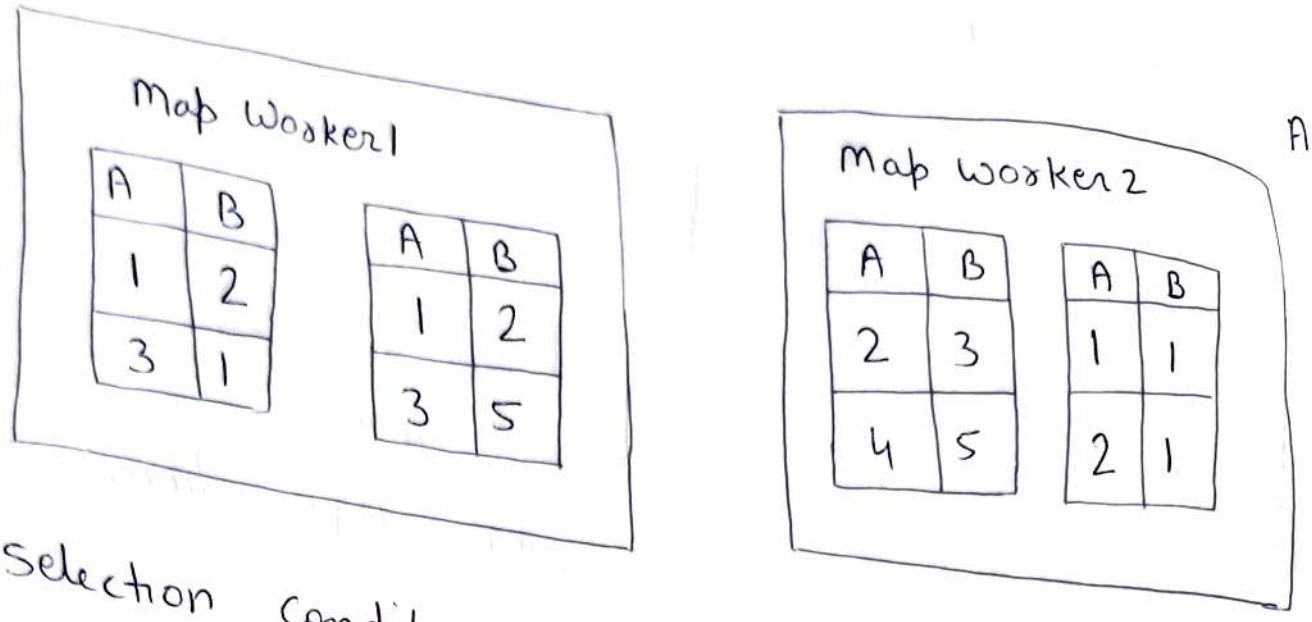
Selection using mapreduce

Selection Algo

```
map (Key, value):
    for tuple in value:
        if tuple satisfies condition:
            emit (tuple, tuple)

reduce (Key, values):
    emit (key, key)
```

Example:



Selection condition: $B <= 2$

Step 1: Create (key, value) pair of each and every record.

mw1

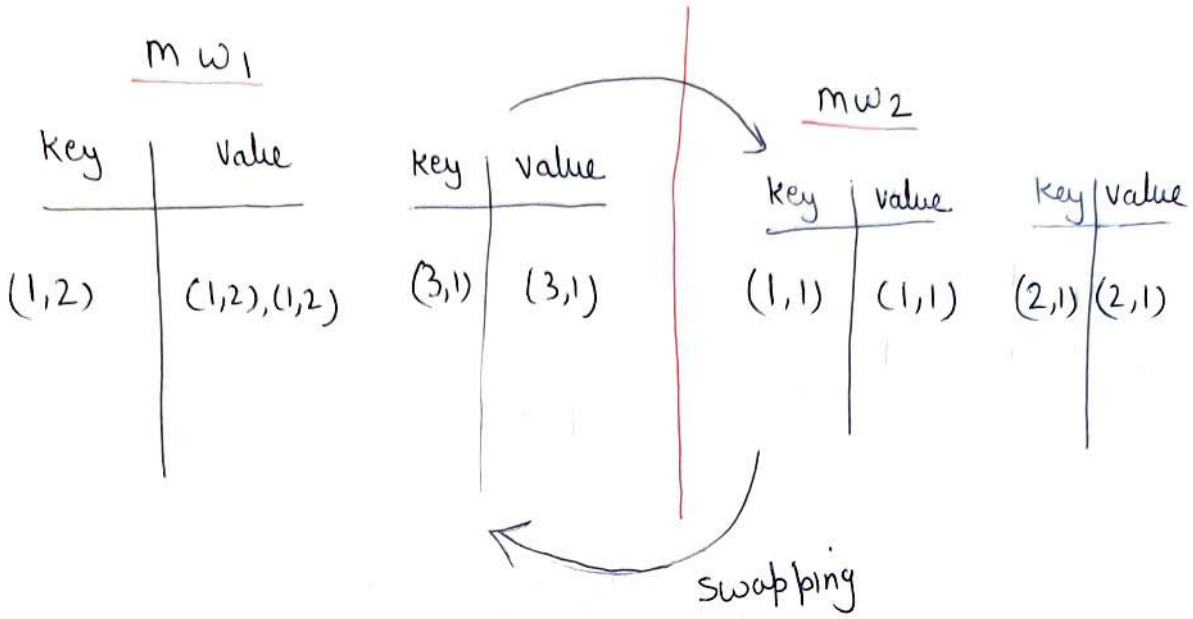
key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)

mw2

key	value
[illegible]	[illegible]
(1,1)	(1,1)
(2,1)	(2,1)

Step 2: After (key, value) conversion a particular hash function will be applied onto map workers. So after applying this hash function the map worker task is divided into two.

A hash function will be applied.



Swapping will be done to discard redundancy that can be caused by duplicate tuples.

Step 3 : swap table 2 and table 3.

RW1

Key	value
(1,2)	(1,2)(1,2)

Key	value
(1,1)	(1,1)

RW2

Key	value
(3,1)	(3,1)

Key	value
(2,1)	(2,1)

Step 4 : Now at RW1 & RW2 we have to club. When combine we have to check if there is any duplicate values.

RW1

Key	value
(1,2)	(1,2)(1,2)
(1,1)	(1,1)

RW2

Key	value
(3,1)	(3,1)
(2,1)	(2,1)

Step 5: The final output will be.

RW1	
A	B
1	2
1	1

RW2	
A	B
3	1
2	1

Projection using map reduce :

Projection Algorithm

let's say S be subset containing the selected attributes.

```
map (key, value):
    for tuple in value:
        ts = tuple with only attributes in S
        emit (ts, ts)

reduce (key, values):
    emit (key, ~~value~~ key)
```

Example

Project : (B,C)

Map worker 1

A	B	C
1	2	3
3	1	4

A	B	C
2	2	3
4	3	4

Map worker 2

A	B	C
5	7	2
7	1	1

A	B	C
3	2	5
4	1	0

Step 1

mw1

Key	Value
(2,3)	(2,3),(2,3)
(1,4)	(1,4)
(3,4)	(3,4)

mw2

Key	Value
(7,2)	(7,2)
(1,1)	(1,1)
(2,5)	(2,5)
(1,0)	(1,0)

Step 2

mw1

Key	Value
(2,3)	(2,3),(2,3)
(1,4)	(1,4)

Key	Value
(3,4)	(3,4)

mw2

Key	Value
(7,2)	(7,2)
(1,1)	(1,1)

Key	Value
(2,5)	(2,5)
(1,0)	(1,0)

Step 3

RW1

Key	value
(2,3)	(2,3),(2,3)
(1,4)	(1,4)

Key	value
(7,2)	(7,2)
(1,1)	(1,1)

RW2

Key	value
(3,4)	(3,4)

Key	value
(2,5)	(2,5)
(1,0)	(1,0)

Step 4

RW1

Key	value
(2,3)	(2,3) (2,3)
(1,4)	(1,4)
(7,2)	(7,2)
(1,1)	(1,1)

RW2

Key	value
(3,4)	(3,4)
(2,5)	(2,5)
(1,0)	(1,0)

Step 5

RW1

B	C
2	3
1	4
7	2
1	1

RW2

B	C
3	4
2	5
1	0

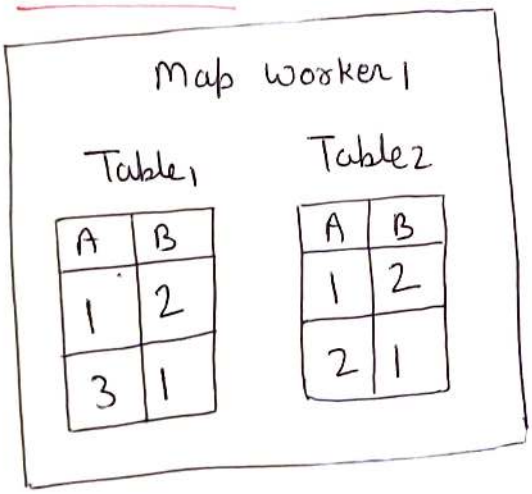
Union using map reduce :

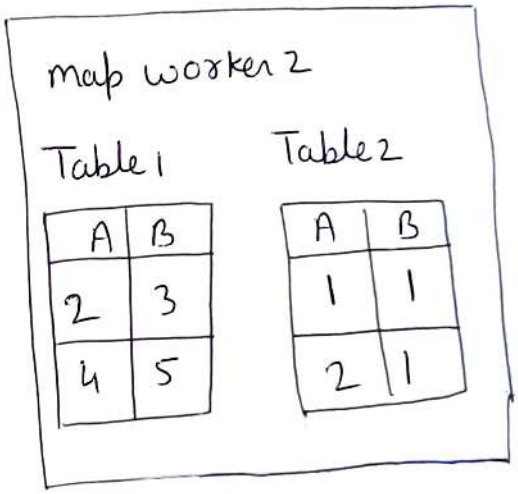
Union algo :

```
map (key, value):
    for tuple in value :
        emit (tuple, tuple)

reduce (key, values):
    emit (key, key)
```

Example





Step 1

MW1

Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)
(2,1)	(2,1)

MW2

Key	value
(2,3)	(2,3)
(4,5)	(4,5)
(1,1)	(1,1)
(2,1)	(2,1)

Step 2

Step 5

mw_1

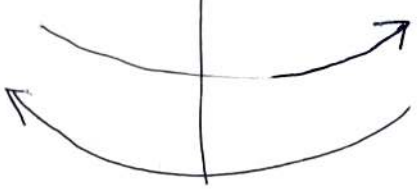
Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)

Key	Value
(2,1)	(2,1)

mw_2

Key	value
(2,3)	(2,3)
(4,5)	(4,5)

Key	value
(1,1)	(1,1)
(2,1)	(2,1)



Step 3

Rw_1

Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)

Key	value
(2,3)	(2,3)
(4,5)	(4,5)

Rw_2

Key	value
(2,1)	(2,1)

Key	value
(1,1)	(1,1)
(2,1)	(2,1)

Step 4

Rw_1

Key	value
(1,2)	(1,2) (1,2)
(3,1)	(3,1)
(2,3)	(2,3)
(4,5)	(4,5)

Rw_2

Key	value
(2,1)	(2,1), (2,1)
(1,1)	(1,1)

step 5

RW1

A	B
1	2
3	1
2	3
4	5

RW2

A	B
2	1
1	1

Intersection using map reduce :

Intersection algo:

```
map (key, value):
    for tuple in value:
        emit (tuple, tuple)

reduce (key, values):
    if values == [key, key]
        emit (key, key)
```

example

map worker 1

Table 1

A	B
1	2
3	1

Table 2

A	B
1	2
2	1

map worker 2

Table 1

A	B
2	3
4	5

Table 2

A	B
1	1
2	1

Step1

mw1

Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)
(2,1)	(2,1)

mw2

key	value
(2,3)	(2,3)
(4,5)	(4,5)
(1,1)	(1,1)
(2,1)	(2,1)

Step2 Apply hash function

m w1

Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)

Key	value
(2,1)	(2,1)

mw2

Key	value
(2,3)	(2,3)
(4,5)	(4,5)

Key	value
(1,1)	(1,1)
(2,1)	(2,1)

Step3 swapping

RW1

Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)

Key	value
(2,3)	(2,3)
(4,5)	(4,5)

RW2

Key	value
(2,1)	(2,1)

Key	value
(1,1)	(1,1)
(2,1)	(2,1)

Reply

RW1

Key	value
(1,2)	(1,2),(1,2)
(3,1)	(3,1)
(2,3)	(2,3)
(4,5)	(4,5)

RW2

Key	value
(2,1)	(2,1),(2,1)
(1,1)	(1,1)

Step 5 If length of "values" is greater than 1, then emit.

RW1

Key	value
(1,2)	(1,2)(1,2)

RW2

Key	Value
(2,1)	(2,1) (2,1)

Step 6

RW1

A	B
1	2

RW2

A	B
2	1

Difference using mapreduce:

Difference algo

```
map (key, value):
    if key == R :
        for tuple in value :
            emit (tuple, R)
    else :
        for tuple in value :
            emit (tuple, S)

reduce (key , values):
    if values == [R] ;
        emit (key, key).
```

example

map worker 1

Table 1

A	B
1	2
3	1

Table 2

A	B
1	2
2	1

map worker 2

Table 1

A	B
2	3
4	5

Table 2

A	B
1	1
2	1

R → Table 1

S → Table 2

Step 1

mw1

Key	Value
(1,2)	[T1, T2]
(3,1)	[T1]
(2,1)	[T2]

mw2

Key	Value
(2,3)	[T1]
(4,5)	[T1]
(1,1)	[T2]
(2,1)	[T2]

Step2 apply hash function.

mw1

Key	value
(1,2)	[T1, T2]
(3,1)	[T1]

Key	value
(2,1)	[T2]

mw2

Key	Value
(2,3)	[T1]
(4,5)	[T1]

Key	value
(1,1)	[T2]
(2,1)	[T2]

Step3

Rw1

Key	value
(1,2)	[T1, T2]
(3,1)	[T1]

Key	value
(2,3)	[T1]
(4,5)	[T1]

Rw2

Key	value
(2,1)	[T2]

Key	value
(1,1)	[T2]
(2,1)	[T2]

Step4

Rw1

Key	value
(1,2)	[T1, T2]
(3,1)	[T1]
(2,3)	[T1]
(4,5)	[T2]

Rw2

Key	value
(2,1)	[T2]
(1,1)	[T2]

* eliminate keys that are present in both table i.e T_1 & T_2 as well as only in T_2.

~~Step 5~~

RW_1

Key	value
(3,1)	$[T_1]$
(2,3)	$[T_1]$
(4,5)	$[T_1]$

RW_2

Key	Value
NiL	

Step 5

RW_1

A	B
3	1
2	3
4	5

RW_2

A	B