Introduction to Data Science

(Lecture 23)

Dr. Mohammad Pourhomayoun

Assistant Professor

Computer Science Department

California State University, Los Angeles

Map Reduce

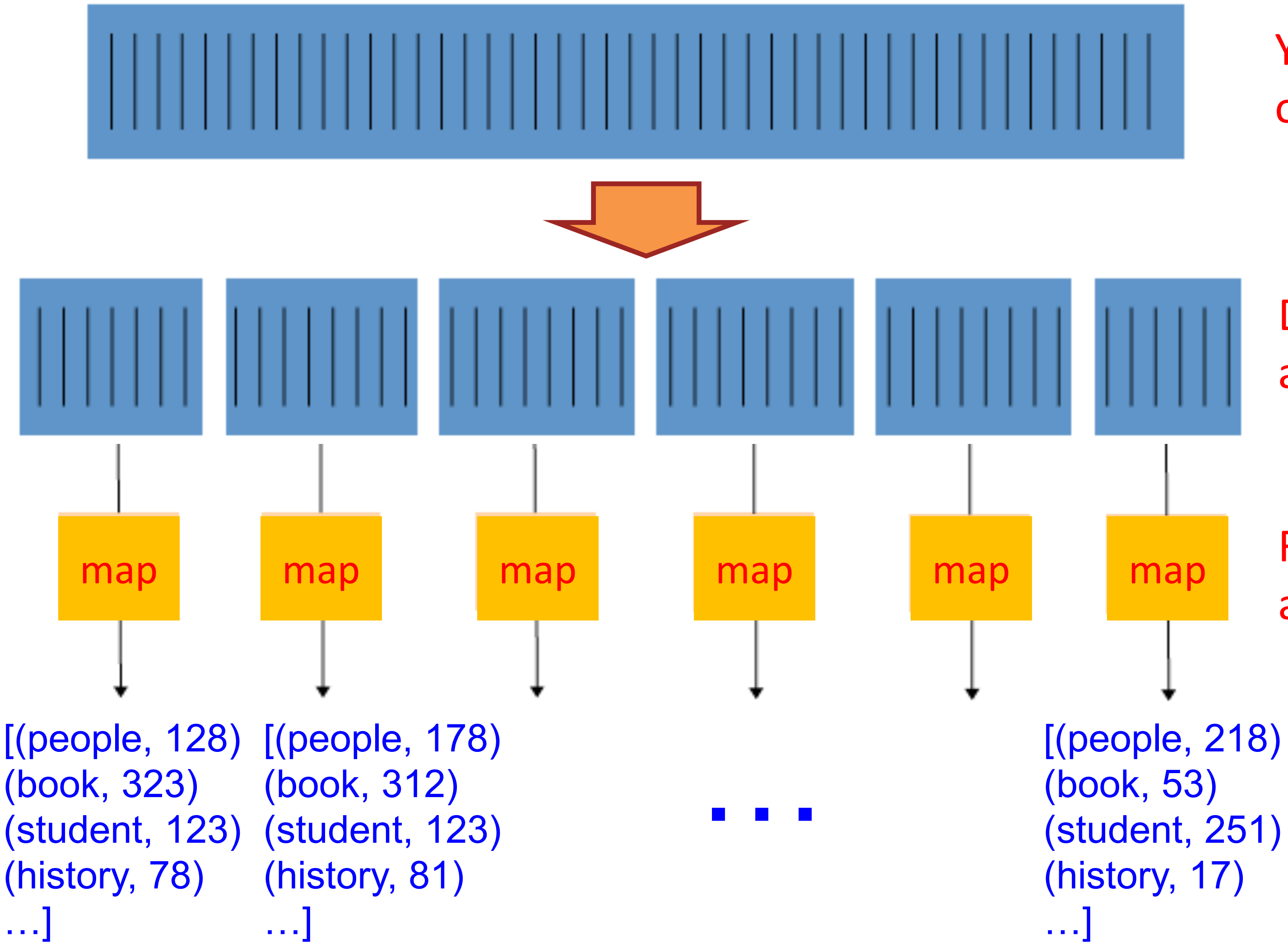
Map-Reduce

- **Map-Reduce** is a programming model for processing and generating big data sets with a parallel and distributed algorithm.

- **map function** processes input key/value pairs to generate a set of intermediate key/value pairs.
- **reduce function** merges all intermediate values associated with the same intermediate key.

[Ref]: Dean, Jeffrey & Ghemawat, Sanjay. (2004). MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM.

Example: Compute overall word frequency across 5M docs



You have millions of documents

Distribute the documents among k computers

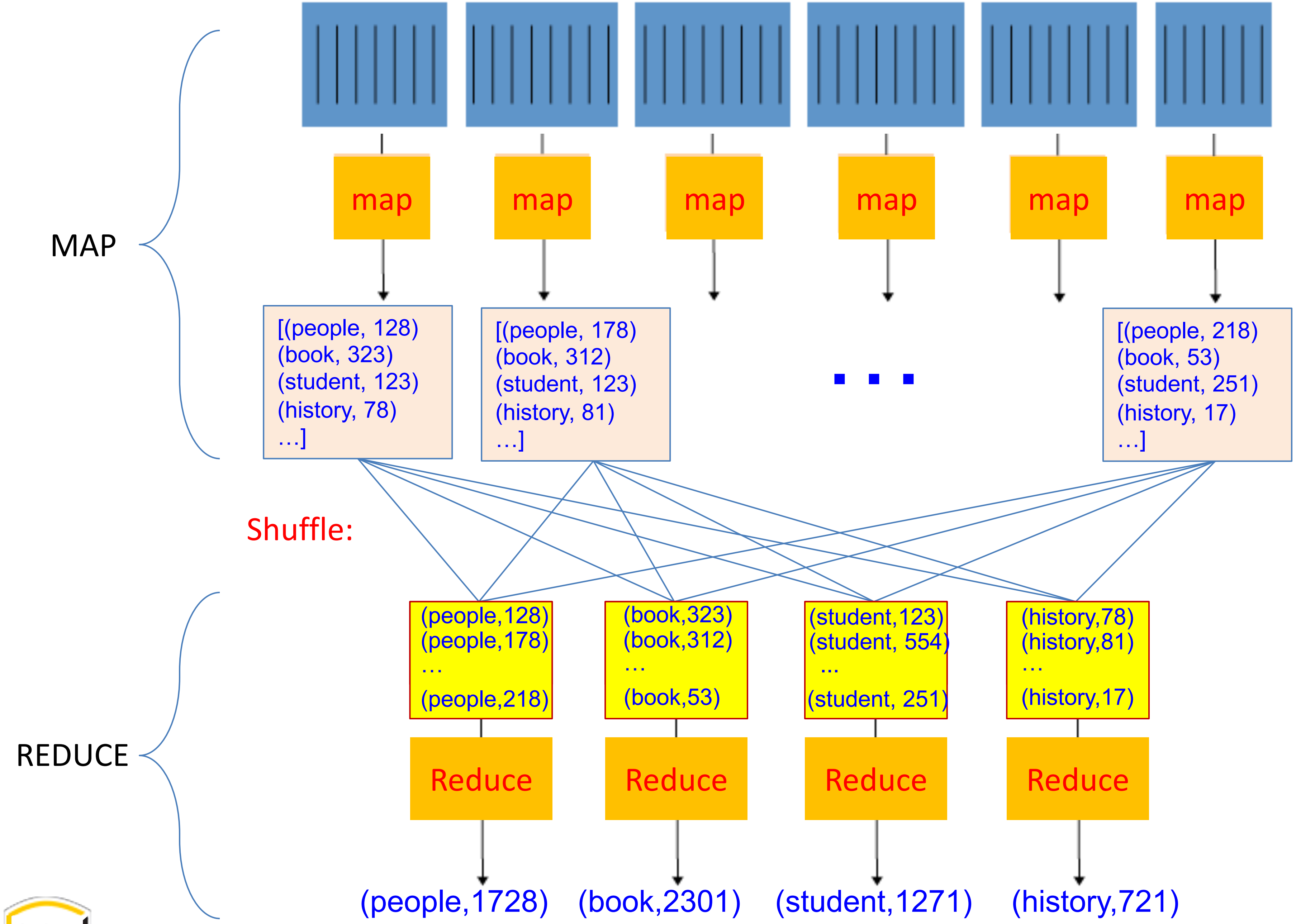
For each document f returns a set of **(word, freq)** pairs

[(people, 128)
(book, 323)
(student, 123)
(history, 78)
…]

[(people, 178)
(book, 312)
(student, 123)
(history, 81)
…]

. . .

[(people, 218)
(book, 53)
(student, 251)
(history, 17)
…]



MAP

map map map map map map

[(people, 128)
(book, 323)
(student, 123)
(history, 78)
…]

[(people, 178)
(book, 312)
(student, 123)
(history, 81)
…]

. . .

[(people, 218)
(book, 53)
(student, 251)
(history, 17)
…]

Shuffle:

REDUCE

(people,128)
(people,178)
…
(people,218)

(book,323)
(book,312)
…
(book,53)

(student,123)
(student, 554)
…
(student, 251)

(history,78)
(history,81)
…
(history,17)

Reduce Reduce Reduce Reduce

(people,1728) (book,2301) (student,1271) (history,721)

Map: Count **All words in Each chunk of data**

Reduce: Count the occurrences of **Each word in the Entire data**

Map Reduce for Relational Database Operations

Example: Relational Join

- **Relational Join**: Stick the tuples of two relations together when they agree on common attributes (column names).
- Consider two database tables R(A, B) and S(B, C).
- **R JOIN S** : T(A, B, C) formed by joining rows (a, b) ϵ R and (b, c) ϵ S with matching b.
- Example: R(A,B) JOIN S(B,C) = {a,b,c | a,b is in R and b,c is in S}.

A	B
6	2
12	2
7	5

R

B	C
2	9
5	11
5	3
9	5

S

A	B	C
6	2	9
12	2	9
7	5	11
7	5	3

R JOIN S

The Map Function for Join

- Each tuple (a,b) in R is mapped to:

  key = b, value = (R,a).

  – Note: "R" in the value is just a bit to indicate "this value represents a tuple in R, not S."

- Each tuple (b,c) in S is mapped to:

  key = b, value = (S,c).

- After grouping by keys (shuffle), each reducer gets a key-list that looks like:

  (b, [(R,a_1), (R,a_2),…, (S,c_1), (S,c_2),…]).

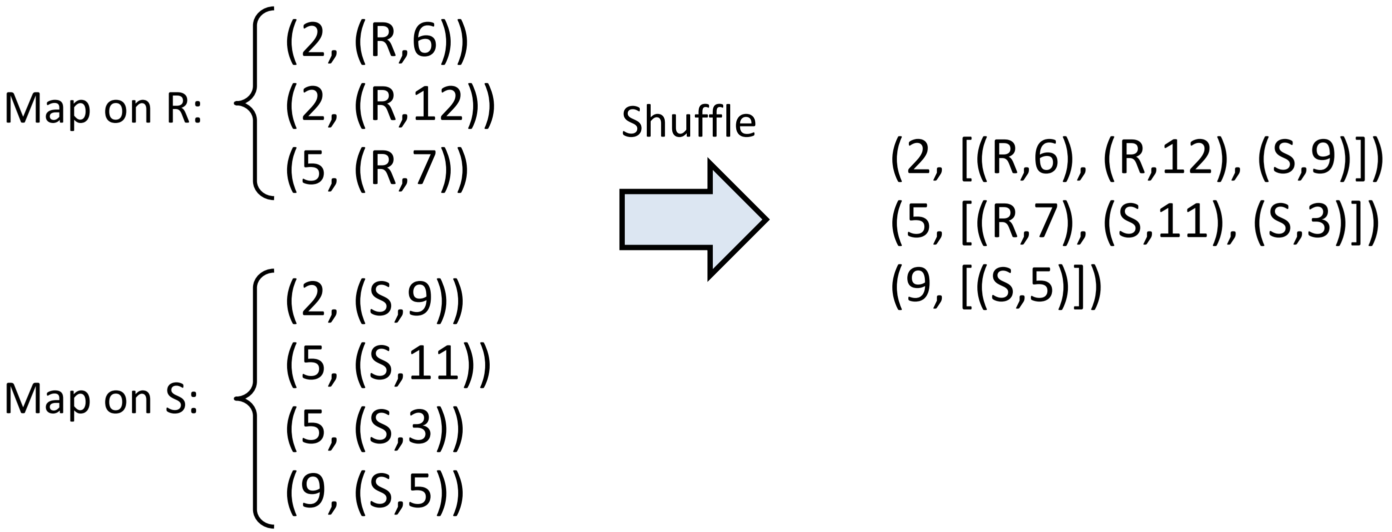
The Map Function for Join

R =

A	B
6	2
12	2
7	5

S =

B	C
2	9
5	11
5	3
9	5



Map on R:

(2, (R,6))
(2, (R,12))
(5, (R,7))

Map on S:

(2, (S,9))
(5, (S,11))
(5, (S,3))
(9, (S,5))

Shuffle

(2, [(R,6), (R,12), (S,9)])
(5, [(R,7), (S,11), (S,3)])
(9, [(S,5)])

The Reduce Function for Join

- After grouping by keys, each reducer gets a key-list that looks like:

$(b, [(R,a_1), (R,a_2),\ldots, (S,c_1), (S,c_2),\ldots])$.

- Reducer generates a tuple (a,b,c) for **each** pair of (R,a_i) and (S,c_j) on the list with key b.

The Reduce Function for Join

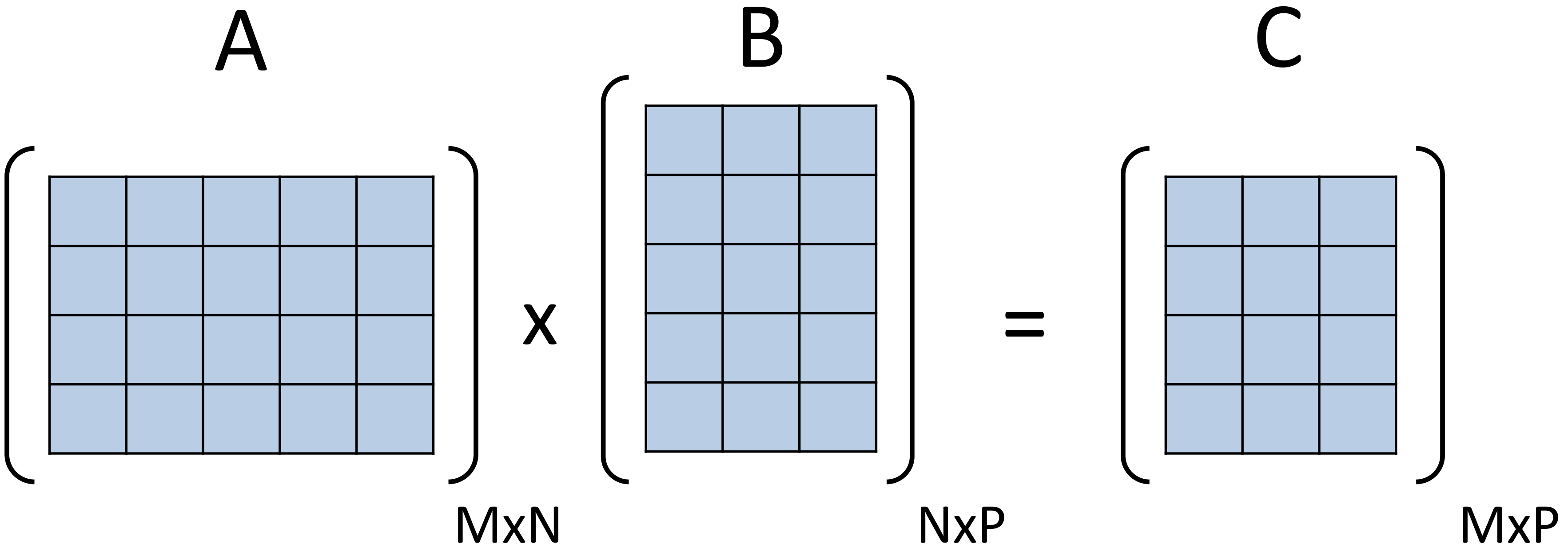
(2, [(R,6), (R,12), (S,9)])
(5, [(R,7), (S,11), (S,3)])
(9, [(S,5)])

⇨

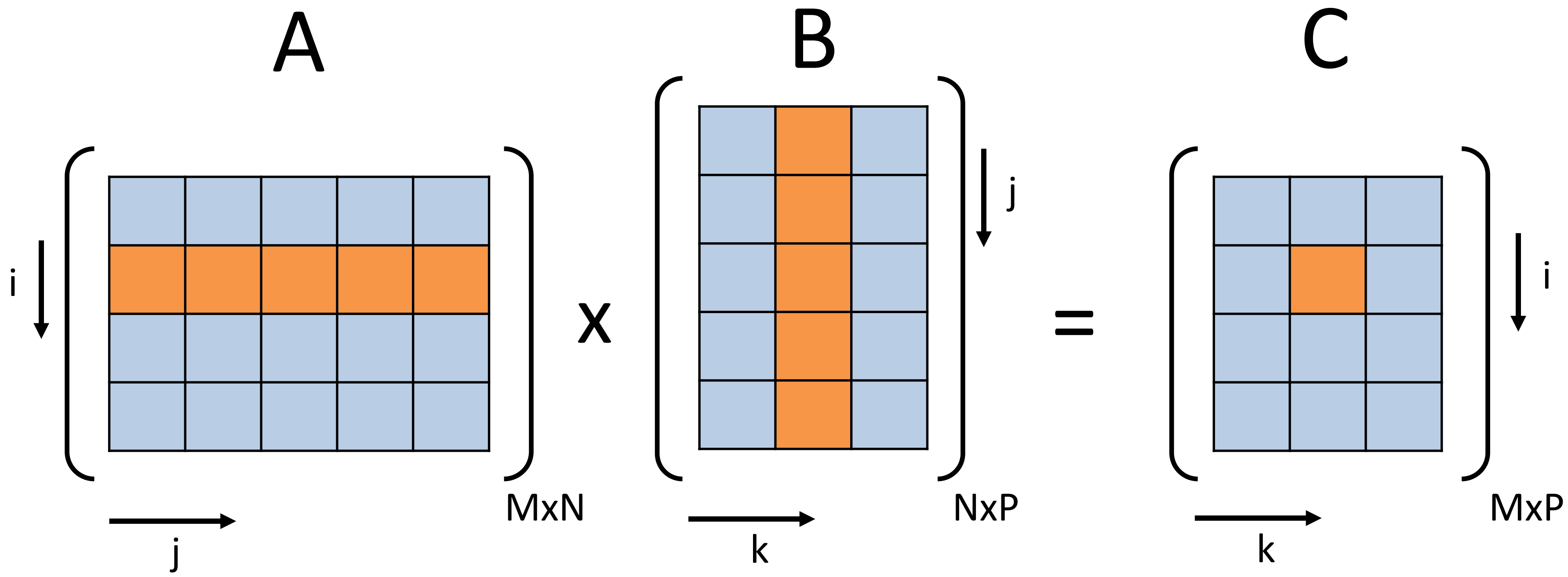
A	B	C
6	2	9
12	2	9
7	5	11
7	5	3

Map Reduce
for Matrix-Vector Multiplication

MapReduce for Matrix Multiplication



Matrix Multiplication



$$c_{ik} = \sum_{j} a_{ij} \times b_{jk}$$

Example

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 5 \\ 1 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2\cdot4+1\cdot1+3\cdot2 & 2\cdot5+1\cdot3+3\cdot1 \\ 4\cdot4+2\cdot1+1\cdot2 & 4\cdot5+2\cdot3+1\cdot1 \end{bmatrix}$$

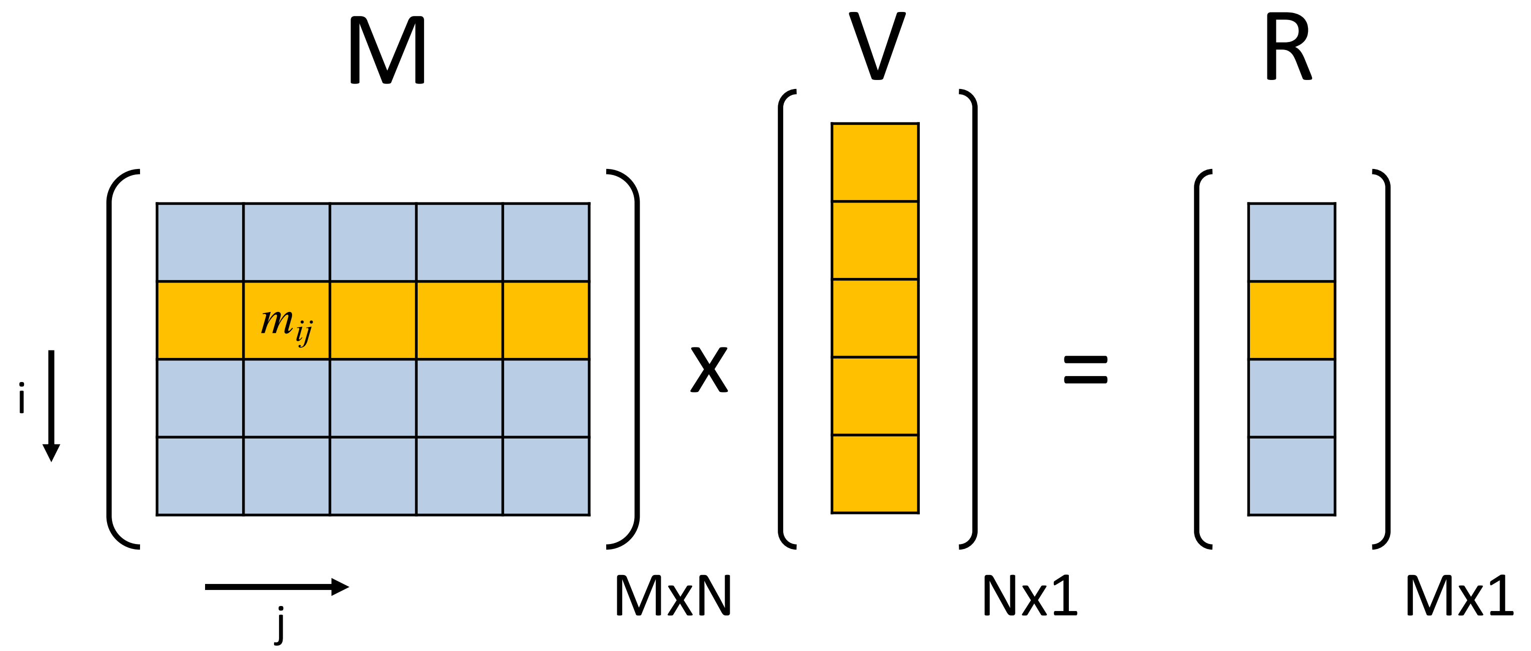
$$= \begin{bmatrix} 15 & 16 \\ 20 & 27 \end{bmatrix}$$

Simple Pseudo Code for Matrix Multiplication (general approach)

```
for i = 1 to n do
    for j = 1 to n do
        for k = 1 to n do
            C[i,j] = C[i,j] + A[i,k] x B[k,j]
        endfor
    endfor
endfor
```

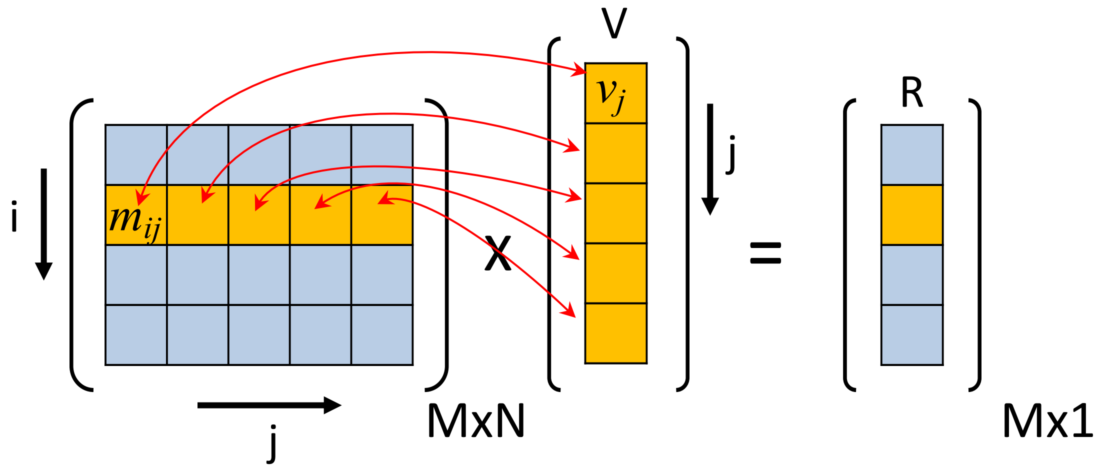
Special Case: Matrix-Vector Multiplication

- Map-Reduce originally developed by **Google** in order to compute the **PageRank** vector.



Map for Matrix-Vector Multiplication

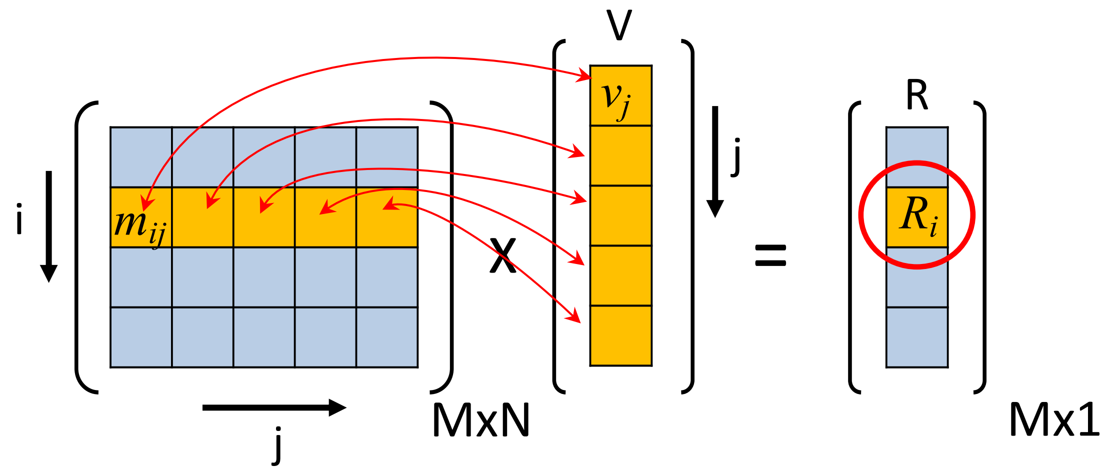
- **Map Function:**
  - N mappers process row i of the matrix at a time
  - Mapper j Maps $((i, j), m_{ij})$ to $(i, m_{ij} v_j)$
  - Note that We assumed that each mapper can load vector v.



Reduce for Matrix-Vector Multiplication

- **Reduce function:**
  - M reducers calculate elements of vector R
  - Reducer i receives $(i, [m_{i1}\, v_1, \ldots m_{iN}\, v_N])$, sums all values of the list of a key i, and produces (i, R_i):

$$R_i = \sum_j m_{ij} \times v_j$$



Thank You!

Questions?