PRACE-6IP

Hadoop

Introduction to MapReduce

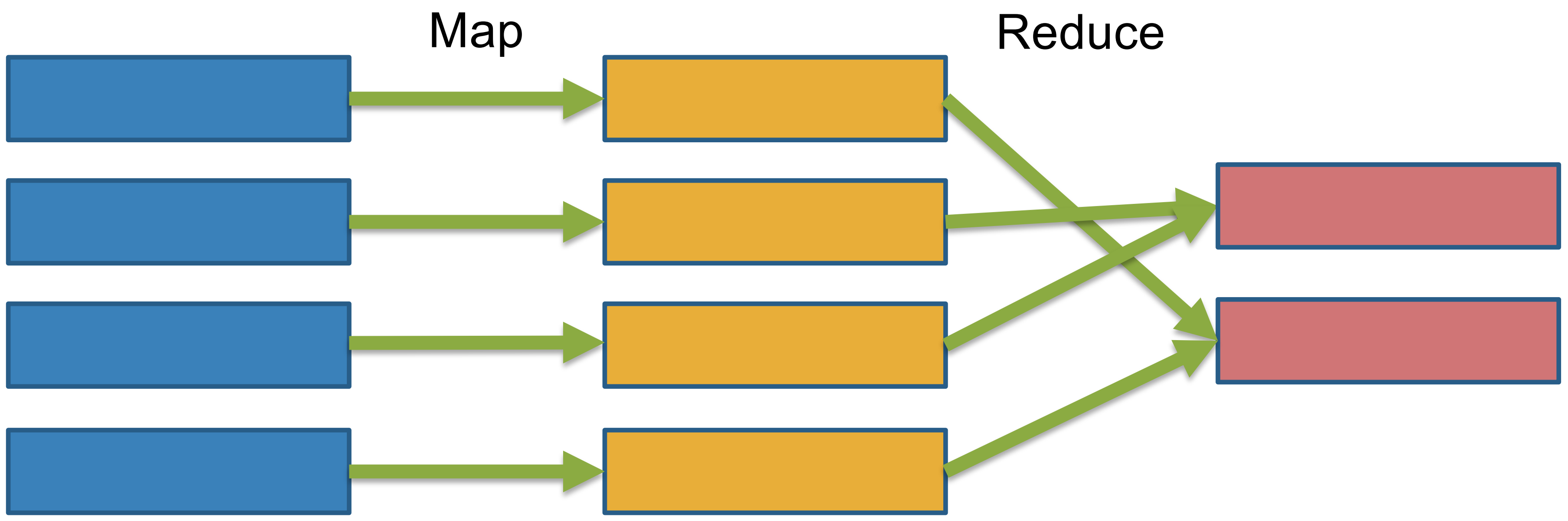
Amy Krause, Andreas Vroutsis
EPCC, The University of Edinburgh

Slides thanks to Ally Hume, EPCC

Overview

- MapReduce pattern
- Distributed computing of MapReduce tasks
- Joining multiple datasets

Map Reduce pattern



Map Reduce pattern

- Must provide stateless Map and Reduce functions:

	Input	Output
Map	<Key1 : Value1>	List(<Key2 : Value2>)
Reduce	<Key2 : List(Value2) >	List(<Key3 : Value3>)

- Framework groups by Key2 before calling reducers
  - Only one reduce call for each unique Key2 key

Map Reduce pattern

- To count words:

	Input	Output
Map	<223, “shop at my shop”>	[<shop,1>, <at,1>, <my,1>, <shop,1>]
Reduce	<shop, [1,1]>	[<shop, 2>]

	Input	Output
Map	<Integer : Text>	List(<Word : Integer>)
Reduce	<Word : List(Integer) >	List(<Word : Integer>)

Map Input	Map Output
<0 : “A boy drove a car”>	[<a,1>, <boy,1>, <drove,1>, <a,1>, <car,1>]
<1 : “A car drove at a bus”>	[<a,1>, <car,1>, <drove,1>, <at,1>, <a,1>, <bus,1>]
<2 : “Can a boy drive a car?”>	[<can,1>, <a,1>, <boy,1>, <drive,1>, <a,1>, <car,1>]
<3 : “A danger – a banana!”>	[<a,1>, <danger,1>, <a,1>, <banana,1>]

Reduce Input	Reduce output
<a,[1,1,1,1,1,1,1,1]>	<a,8>
<at, [1]>	<at,1>
<banana,[1]>	<banana,1>
<boy, [1,1]>	<boy,2>
<bus,[1]>	<bus,1>
<can,[1]>	<can, 1>
<car,[1,1,1]>	<car, 3>
<danger,[1]>	<danger,1>
<drive,[1]>	<drive,1>
<drove,[1,1]>	<drove,2>

Map Reduce exercise 1

- From US National Bureau of Economic Research
  - http://www.nber.org/patents/ (Cite75_99.txt)
- Lists patent IDs and the other patents they cite

```
"CITING", "CITED"
 3858241, 956203
 3858241, 1324234
 3858242, 1515701
 3858244, 956203
```

Map Reduce exercise 1

- Map Reduce task
  - Count the number of times each patent is cited
  - Tip: Do not need output for patents that are never cited
  - Tip: Reader is easily told to ignore the header row
  - Desired output:

```
956203,  2
1515701, 1
1324234, 1
```

	Input	Output
Map	<Key1 : Value1>	List(<Key2 : Value2>)
Reduce	<Key2 : List(Value2) >	List(<Key3 : Value3>)

Map Reduce exercise 1 answer

- Reader: key/value pair both of type integer
- Map: <Integer,Integer> → List(<Integer,Integer>)
  - Extracts the cited patent id and outputs it as key with value 1

Map Input	Map Output
<3858241, 956203>	[<956203,1>]
<3858241, 1324234>	[<1324234,1>]

Map Reduce exercise 1 answer

- Reduce <Integer,List(Integer)> → List(<Integer,Integer>)
  - Simply sums the values as outputs along with the input key

Reduce Input	Reduce output
<956203, [1, 1, 1, 1] >	<956203, 4>
<13242434, [1, 1]>	<13242434,2>

Map Reduce exercise 2

- Same citation data set

```
"CITING", "CITED"
3858241, 956203
3858241, 1324234
3858242, 1515701
3858244, 956203
```

- Map Reduce Task:
  - Invert citation data set to get for each patent the list of patents that cite it
  - Desired output:

```
956203,  3858241, 3858244
1515701, 3858242
1324234, 3858241
```

Map Reduce exercise 2 answer

- Reader: key/value pair both of type integer
- Map: <Integer,Integer> $\rightarrow$ List(<Integer,Integer>)
  - Extracts the cited patent id and outputs it as key with citing as value

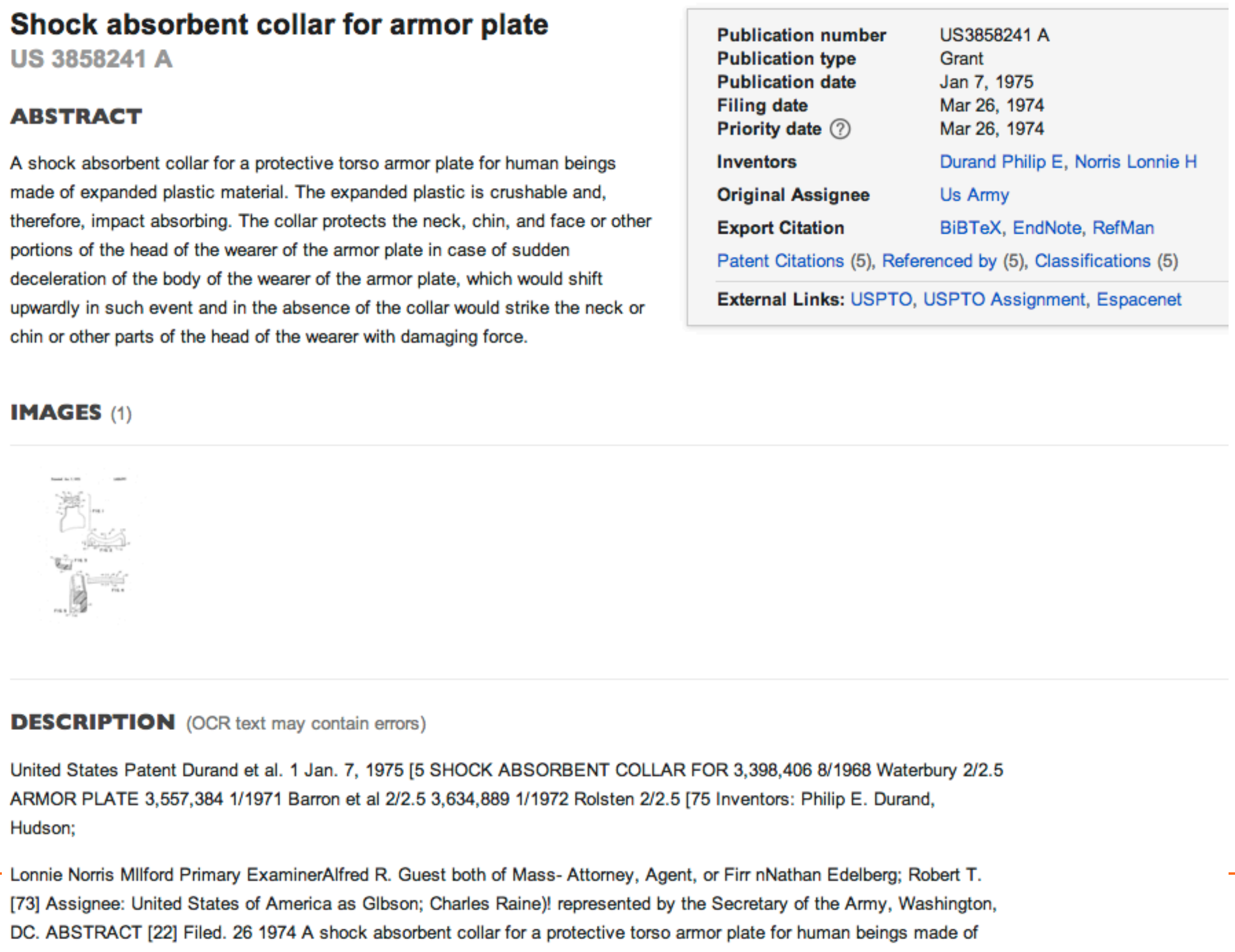
Map Input	Map Output
<3858241, 956203>	[<956203,3858241>]
<3858241, 1324234>	[<1324234,3858241>]

Map Reduce exercise 2 answer

- Reduce <Integer,List(Integer)> → List(<Integer,String>)
  - Concatenates the values as strings and outputs along with the key

Reduce Input	Reduce output
<956203, [3858241, 3858244] >	[<956203, “3858241, 3858244”>]
<13242434, [3858241]>	[<13242434, “3858241” >]

Finding similar patents



Shock absorbent collar for armor plate

US 3858241 A

ABSTRACT

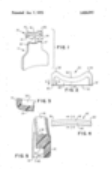
A shock absorbent collar for a protective torso armor plate for human beings made of expanded plastic material. The expanded plastic is crushable and, therefore, impact absorbing. The collar protects the neck, chin, and face or other portions of the head of the wearer of the armor plate in case of sudden deceleration of the body of the wearer of the armor plate, which would shift upwardly in such event and in the absence of the collar would strike the neck or chin or other parts of the head of the wearer with damaging force.

Publication number	US3858241 A
Publication type	Grant
Publication date	Jan 7, 1975
Filing date	Mar 26, 1974
Priority date	Mar 26, 1974
Inventors	Durand Philip E, Norris Lonnie H
Original Assignee	Us Army
Export Citation	BiBTeX, EndNote, RefMan

Patent Citations (5), Referenced by (5), Classifications (5)

External Links: USPTO, USPTO Assignment, Espacenet

IMAGES (1)



DESCRIPTION (OCR text may contain errors)

United States Patent Durand et al. 1 Jan. 7, 1975 [5 SHOCK ABSORBENT COLLAR FOR 3,398,406 8/1968 Waterbury 2/2.5 ARMOR PLATE 3,557,384 1/1971 Barron et al 2/2.5 3,634,889 1/1972 Rolsten 2/2.5 [75 Inventors: Philip E. Durand, Hudson;

Lonnie Norris MIlford Primary ExaminerAlfred R. Guest both of Mass- Attorney, Agent, or Firr nNathan Edelberg; Robert T. [73] Assignee: United States of America as GIbson; Charles Raine)! represented by the Secretary of the Army, Washington, DC. ABSTRACT [22] Filed. 26 1974 A shock absorbent collar for a protective torso armor plate for human beings made of

- Patent citation records:

```
“CITING”, “CITED”
3858241, 956203
3858241, 1324234
3858242, 151570
3858244, 956203
```

- How could you identify similar patents?

Finding similar patents with Map Reduce

- Using ‘patents frequently cited together’ strategy
- First gather all citations made by each patent:

Map Input	Map Output
<“1111”, “9999”>	[<“1111”, “9999”>]

Reduce Input	Reduce Output
<“1111”, [“9999”, “2222”, “7777”] >	[<“1111”, “9999, 2222, 7777”>]

Finding similar patents with Map Reduce

- Next count all pairs that are cited together

Map Input	Map Output
<“1111”, “9999, 2222, 7777”>	[<“2222+9999”, 1>, <“2222+7777”, 1> , <“7777+9999”, 1>]

Reduce Input	Reduce Output
<“2222+9999”, [1, 1, 1, 1] >	[<“2222+9999”, 4>]

Finding similar patents with Map Reduce

- Using ‘patents frequently citing same patents’ strategy
- First gather all citations for each patent:

Map Input	Map Output
<“1111”, “9999”>	[<“9999”, “1111”>]

Reduce Input	Reduce Output
<“9999”, [“1111”, “3333”, “8888”] >	[<“9999”, “1111, 3333, 8888”>]

Finding similar patents with Map Reduce

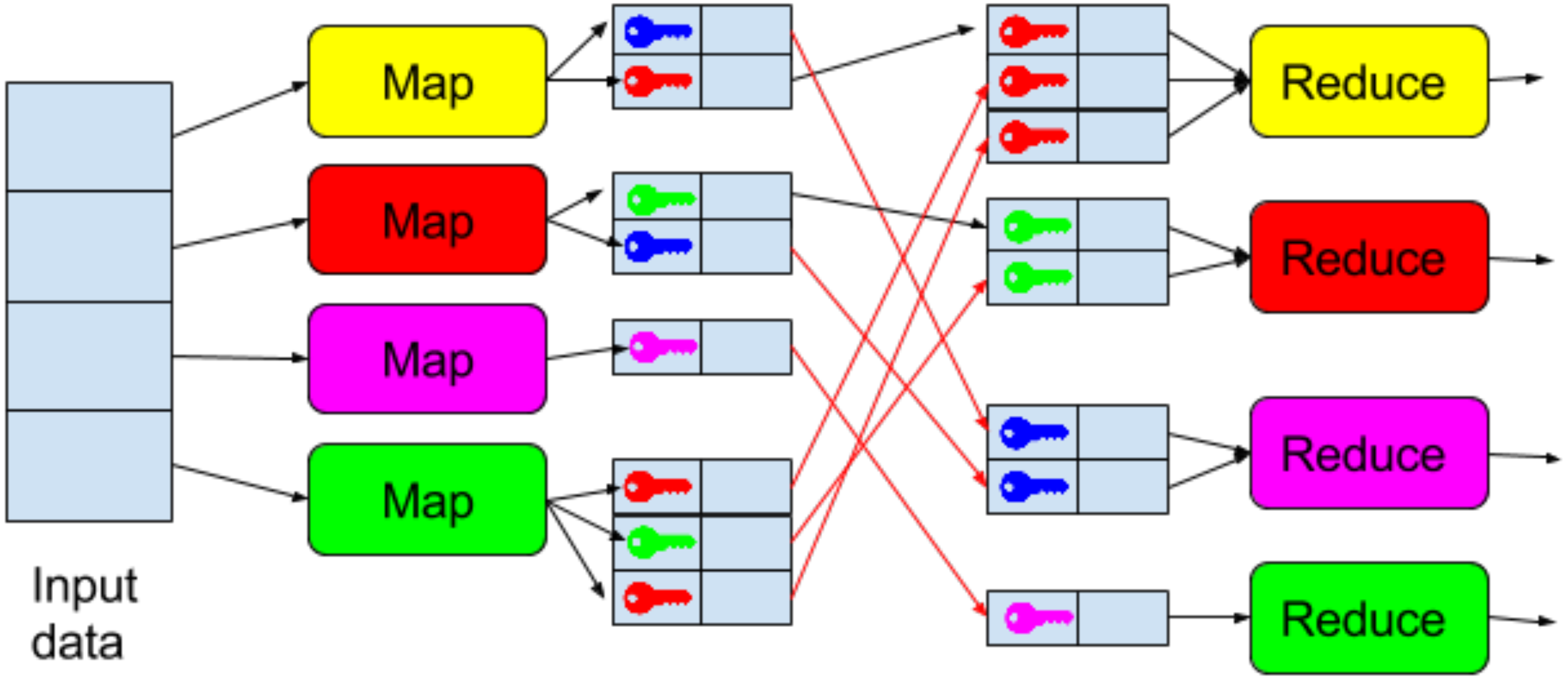
- Next count all pairs that are cited together

Map Input	Map Output
<“9999”, “1111, 3333, 8888”>	[<“1111+3333”, 1>, <“1111+8888”, 1>, <“3333+8888”, 1>]

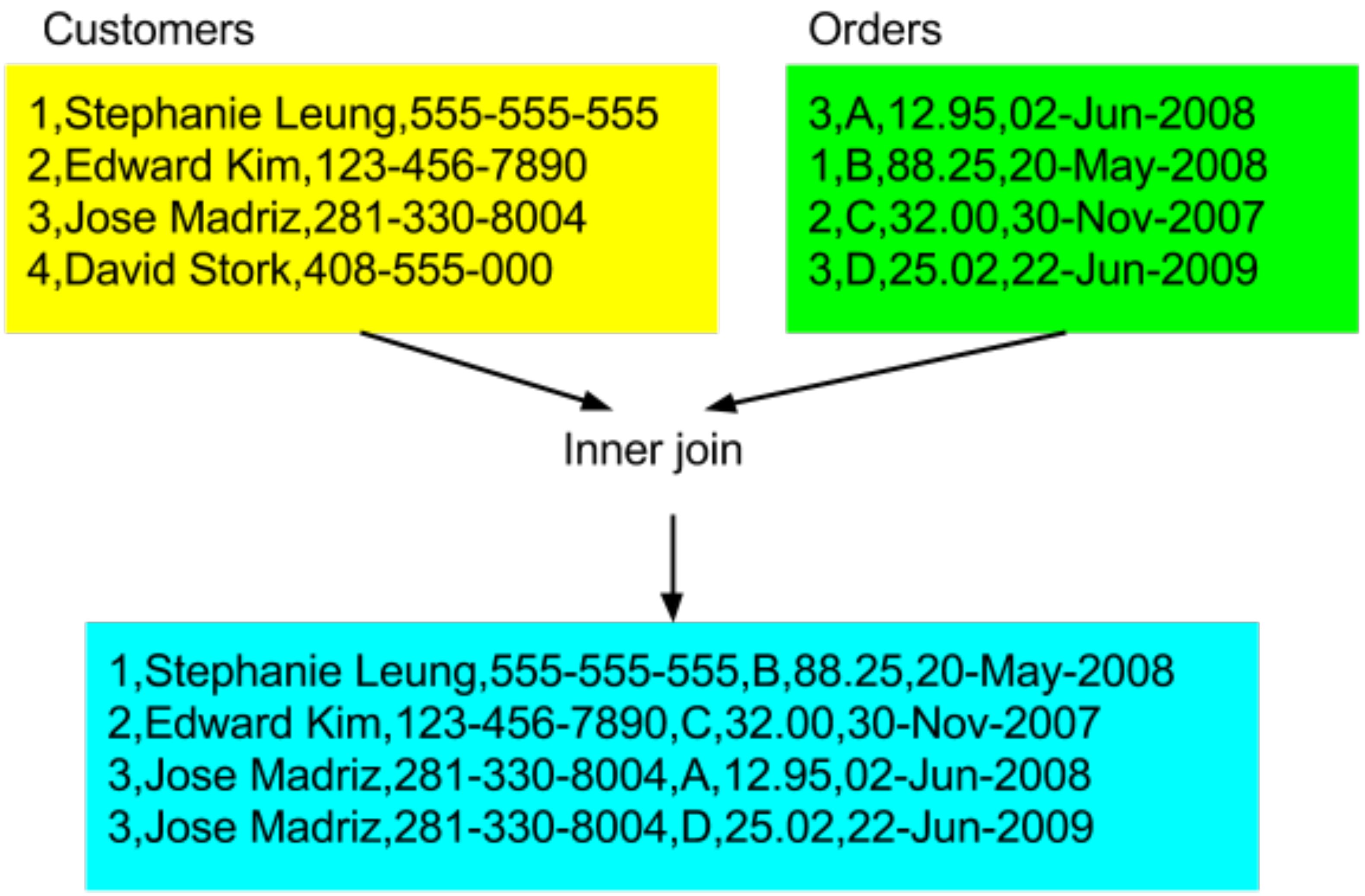
Reduce Input	Reduce Output
<“1111+3333”, [1, 1, 1, 1] >	[<“1111+3333”, 4>

Map Reduce at scale

- Stateless map and reduce functions allows massive parallelisation
- Between the Map and Reduce stages the grouping and moving data stage can be expensive



Joining multiple data sets: Inner Join



Customers:

```
1,Stephanie Leung,555-555-555
2,Edward Kim,123-456-7890
3,Jose Madriz,281-330-8004
4,David Stork,408-555-000
```

Orders:

```
3,A,12.95,02-Jun-2008
1,B,88.25,20-May-2008
2,C,32.00,30-Nov-2007
3,D,25.02,22-Jun-2009
```

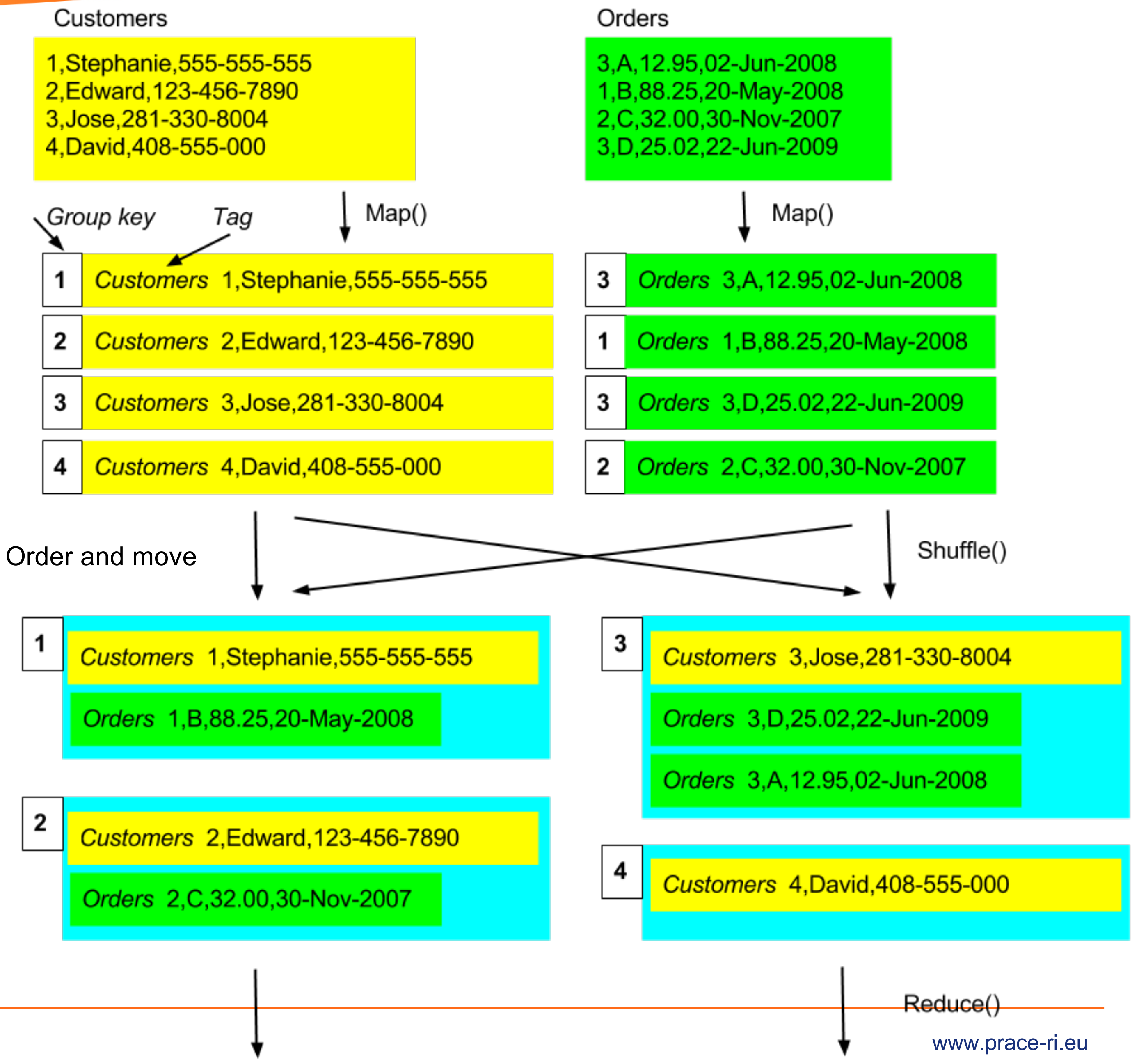
Inner join:

```
1,Stephanie Leung,555-555-555,B,88.25,20-May-2008
2,Edward Kim,123-456-7890,C,32.00,30-Nov-2007
3,Jose Madriz,281-330-8004,A,12.95,02-Jun-2008
3,Jose Madriz,281-330-8004,D,25.02,22-Jun-2009
```

Example from: Hadoop in Action, Chuck Lamb

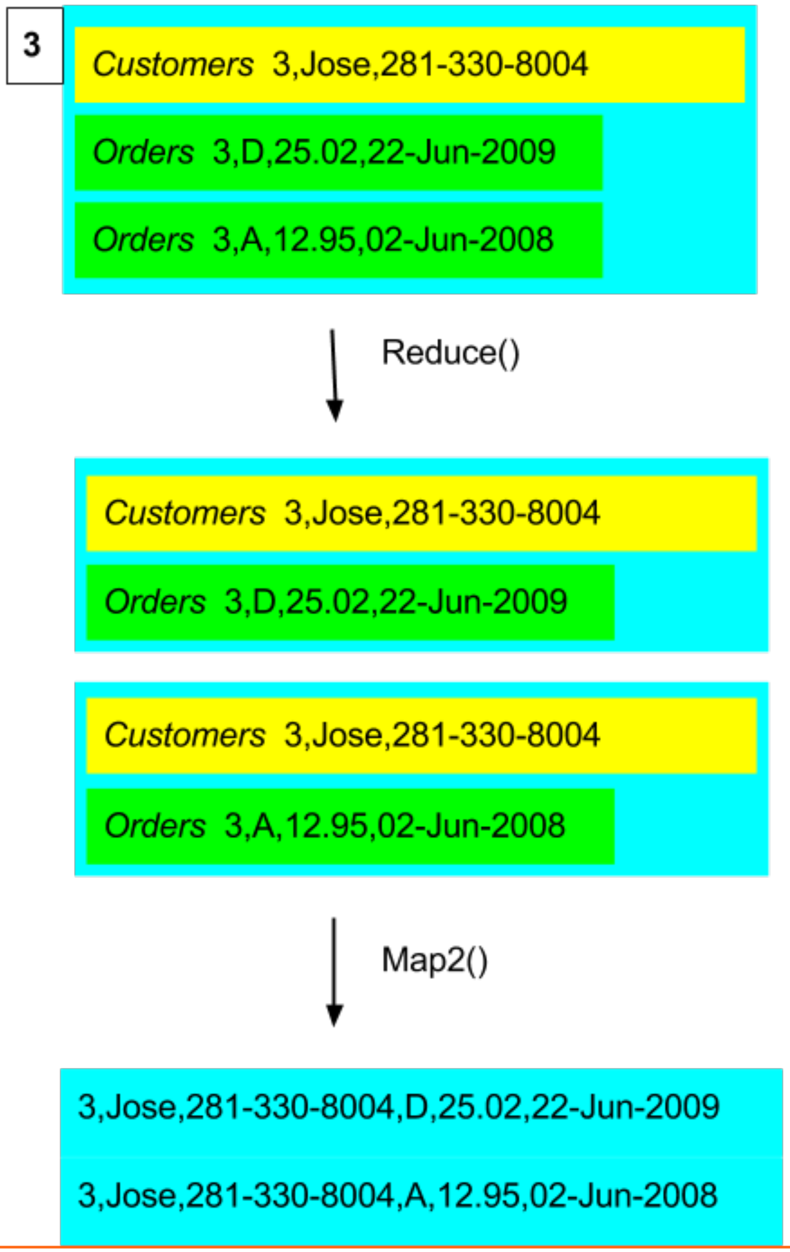
Reduce side join: repartitioned join 1

- Add a tag to store data source filename along with each record


Customers
1,Stephanie,555-555-555
2,Edward,123-456-7890
3,Jose,281-330-8004
4,David,408-555-000
Orders
3,A,12.95,02-Jun-2008
1,B,88.25,20-May-2008
2,C,32.00,30-Nov-2007
3,D,25.02,22-Jun-2009
Group key
Tag
Map()
Map()
1 Customers 1,Stephanie,555-555-555
2 Customers 2,Edward,123-456-7890
3 Customers 3,Jose,281-330-8004
4 Customers 4,David,408-555-000
3 Orders 3,A,12.95,02-Jun-2008
1 Orders 1,B,88.25,20-May-2008
3 Orders 3,D,25.02,22-Jun-2009
2 Orders 2,C,32.00,30-Nov-2007
Order and move
Shuffle()
1 Customers 1,Stephanie,555-555-555
Orders 1,B,88.25,20-May-2008
2 Customers 2,Edward,123-456-7890
Orders 2,C,32.00,30-Nov-2007
3 Customers 3,Jose,281-330-8004
Orders 3,D,25.02,22-Jun-2009
Orders 3,A,12.95,02-Jun-2008
4 Customers 4,David,408-555-000
Reduce()

Reduce side join: repartitioned join 2

- Reduce produces cross-product of records with a single instance of each tag in each output
- Second Mapper implements join style (inner, outer etc).
- Hadoop has classes that support such join patterns.


3
Customers 3,Jose,281-330-8004
Orders 3,D,25.02,22-Jun-2009
Orders 3,A,12.95,02-Jun-2008
Reduce()
Customers 3,Jose,281-330-8004
Orders 3,D,25.02,22-Jun-2009
Customers 3,Jose,281-330-8004
Orders 3,A,12.95,02-Jun-2008
Map2()
3,Jose,281-330-8004,D,25.02,22-Jun-2009
3,Jose,281-330-8004,A,12.95,02-Jun-2008

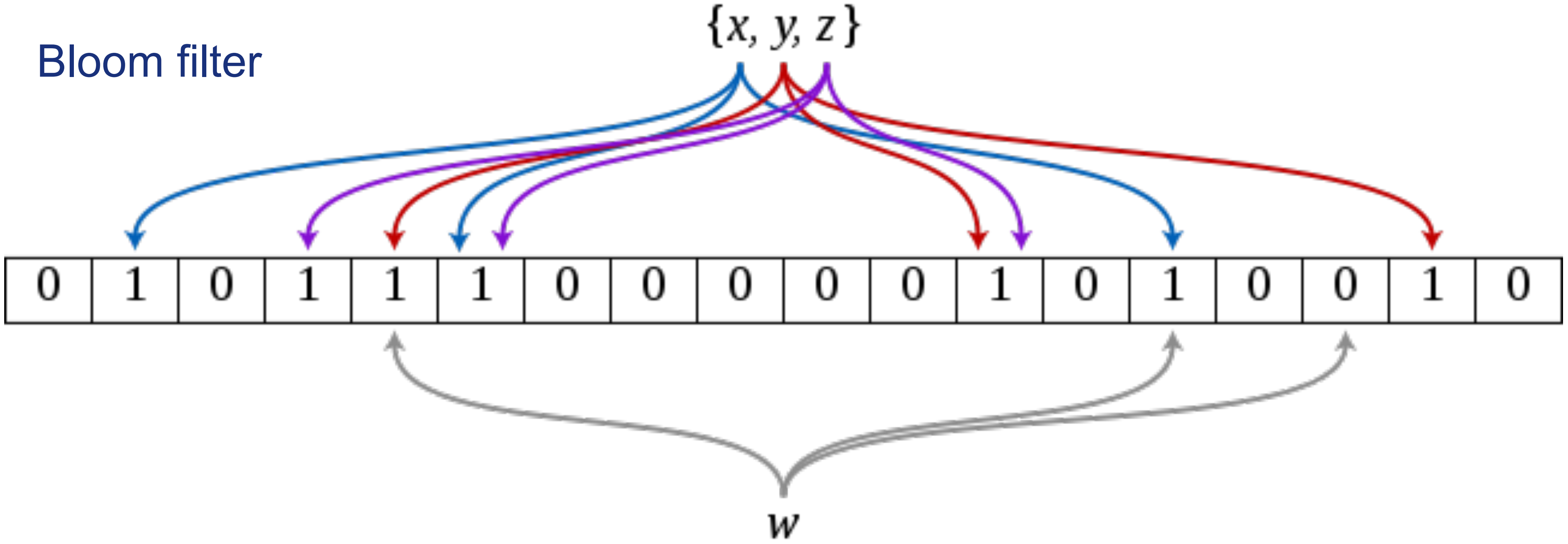
Map side join: replicated joins

- Reduce-side joins require lots of expensive data transfer in shuffle phase.
- If joining one large dataset and one small dataset it may be more efficient to move small dataset to all nodes and then execute the join at the Map stage (and eliminate the Shuffle and Reduce stages).
- Hadoop provides a Distributed Cache to distribute files to all nodes in the cluster.

Alternatives to replication join

- Sometimes data sets are just too big for replication join
- Reduce data transfer by map-side filtering
  - Reduce amount of data transfer by filtering to only those records of interest, e.g. only those customers who live in Scotland.
    - Note: applying such a filter may make the data set small enough to use the replicated join strategy.
  - Replicate only the join keys rather than the whole records
    - Thus only data which will actually be joined is transferred
  - If join keys are still too large consider a smaller data structure that gives an approximate answer, e.g. Bloom filter
    - BloomFilter.contains(x) – returns true if x is in the filter
    - BloomFilter.contains(x) – returns either true or false if x is not in the filter.
    - Level of false positives related to the size of the filter.

Bloom filter



By David Eppstein - self-made, originally for a talk at WADS 2007, Public Domain, https://commons.wikimedia.org/w/index.php?curid=2609777

THANK YOU FOR YOUR ATTENTION

www.prace-ri.eu