



Fundamentals of Big Data Analytics

Lecture 16 – Introduction to MapReduce

Dr. Iqra Safder
Assistant Professor
FAST NUCES, Lahore



Quiz 3 – Thursday, 21st April

Lecture 11 - Lecture 15

Quiz 4 – Thursday, 28th April

Lecture 16 - Lecture 18

The problem:

- Big data means ...
lots of hard drives



The solution:

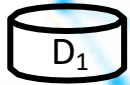
- Lots of data means we should...
bring computation to data!

Lots of disks:



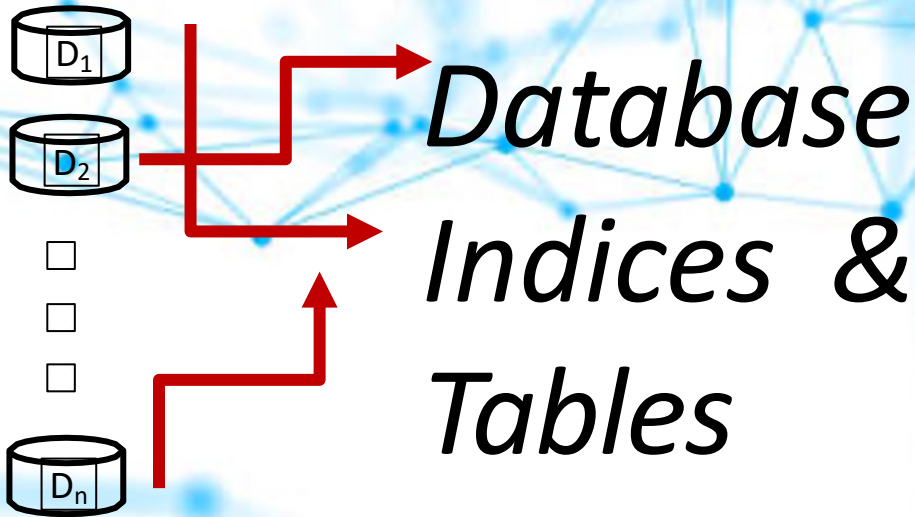
Possibilities:

- Case 1: data needs updating



Possibilities:

- Case 1: data needs updating so ...



Possibilities:

- Case 2: need to sweep through data

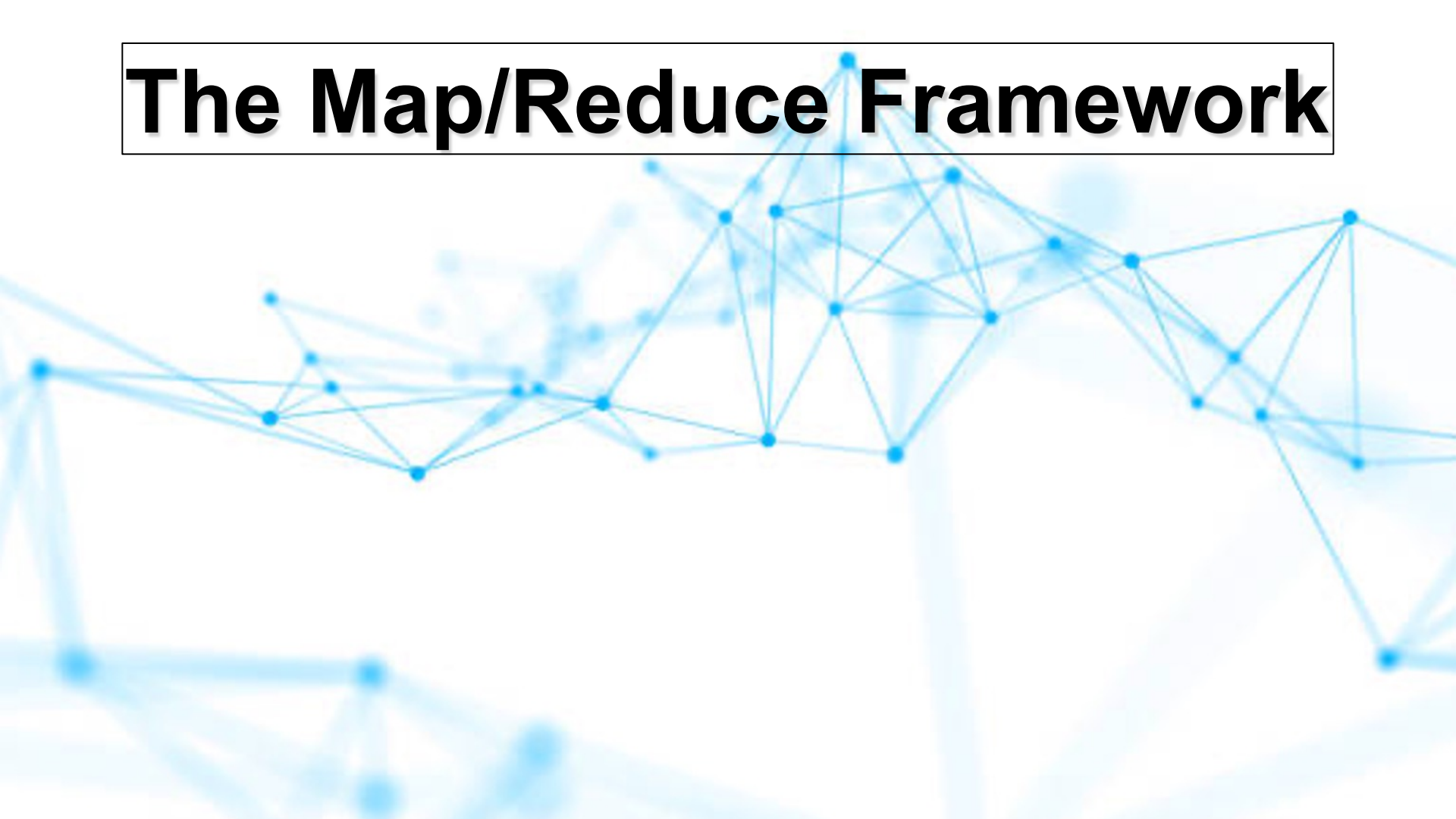


Possibilities:

- Case 2: need to sweep through data so...



The Map/Reduce Framework



The framework:

- User defines:
 - a. `<key, value>`
 - b. mapper & reducer functions
- Hadoop handles the logistics

The logistics:

- Hadoop handles the distribution and execution



Map/Reduce flow

- map() reads data and outputs $\langle \text{key}, \text{value} \rangle$

`map()` → $\langle \text{key}, \text{value} \rangle$

Map/Reduce flow

- User defines a reduce function

`reduce()`

Map/Reduce flow

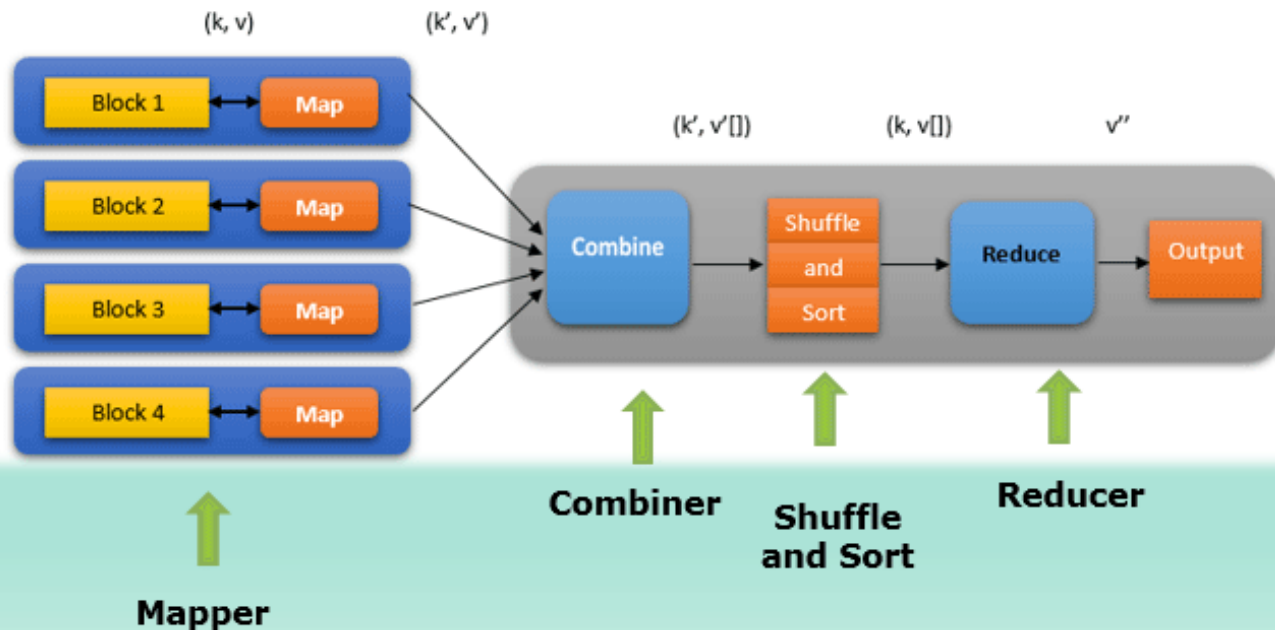
- `reduce()` reads `<key,value>` and outputs your result

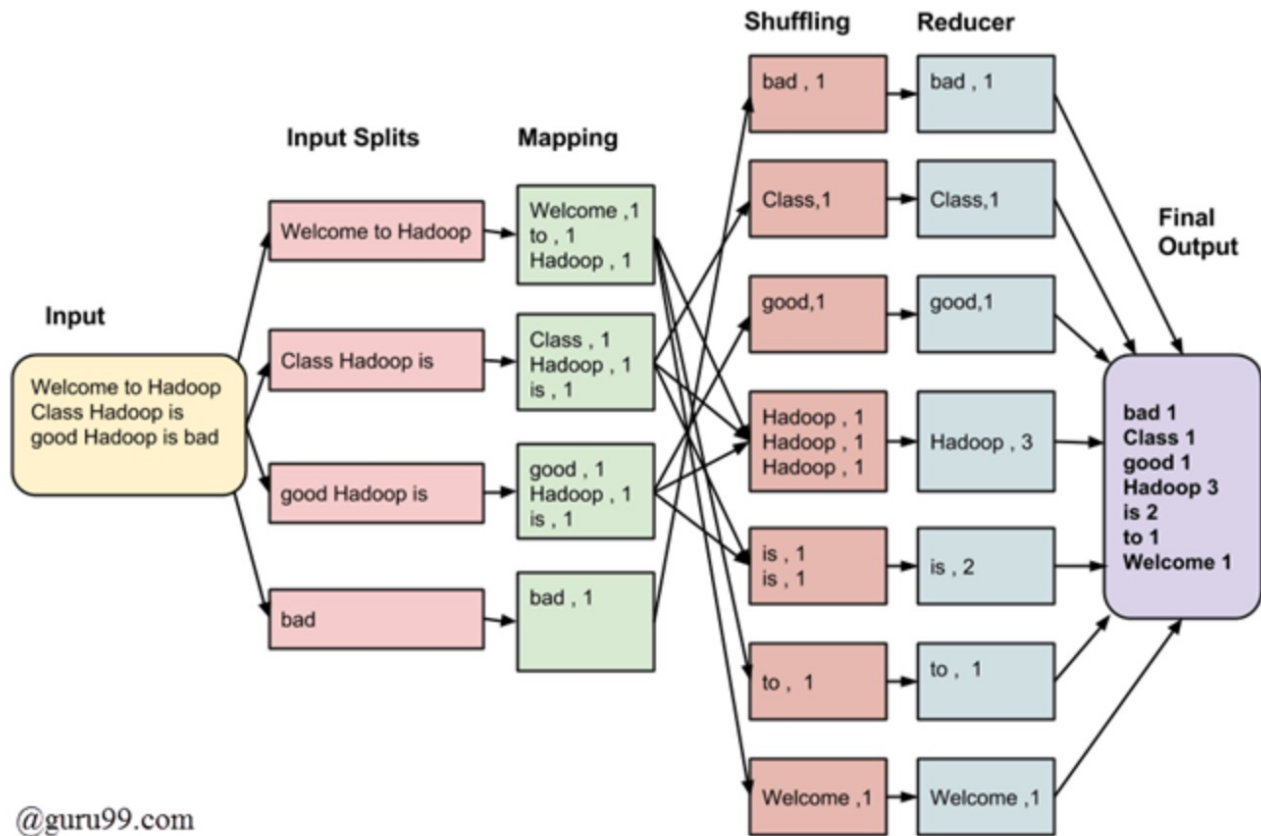
<key,value>

`reduce()`

result

How MapReduce Works





MapReduce Architecture

Wordcount task:

- ... books, blogs, and fan-fiction?



Wordcount task:

- ... books, blogs, and fan-fiction?

A photograph of a row of Star Wars comic books standing upright on a dark surface. A green rectangular box is overlaid on the middle of the row, containing the text 'use map/reduce of course' in white, italicized font. The comic book on the far right is clearly visible, showing the 'STAR WARS' title and a character illustration.

use map/reduce of course

Map/Reduce Strategy



- Keep it simple!

Wordcount Strategy

- Let $\langle \text{word}, 1 \rangle$ be the $\langle \text{key}, \text{value} \rangle$

Wordcount Strategy

An abstract graphic in the background consisting of a network of blue dots connected by thin blue lines, resembling a molecular structure or a data network. The dots are of varying sizes and are distributed across the slide, with a higher concentration in the upper right and lower left areas.

- Let Hadoop do the hard work

Wordcount Map/Reduce:

Loop
Until
Done

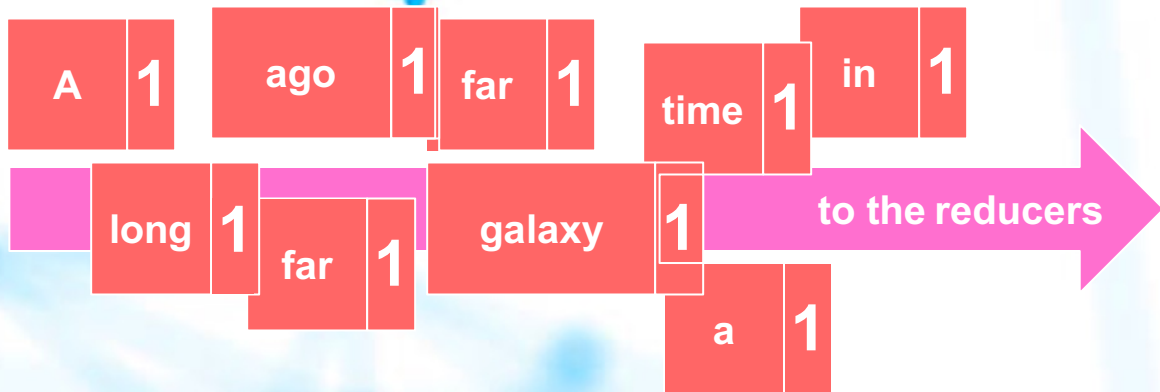
Get word

Emit <word> < 1>

What One Mapper Does

A long time ago in a galaxy far far ...

A	long	time	ago	in	a	galaxy	far	far
---	------	------	-----	----	---	--------	-----	-----



Wordcount Map/Reduce:

Loop
Over
key-
values

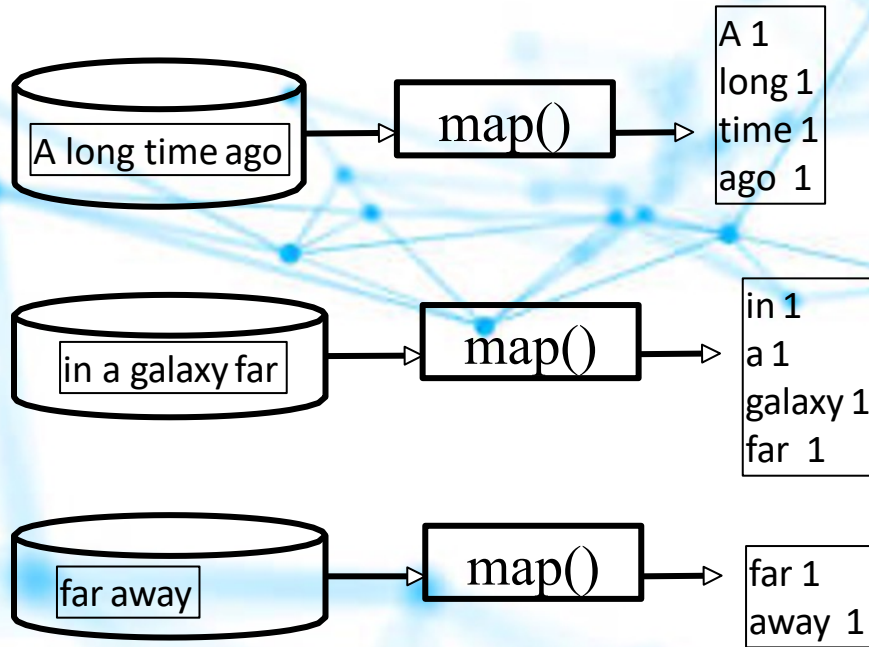
Get next <word><value>

***If <word> is same as previous word
add <value> to count***

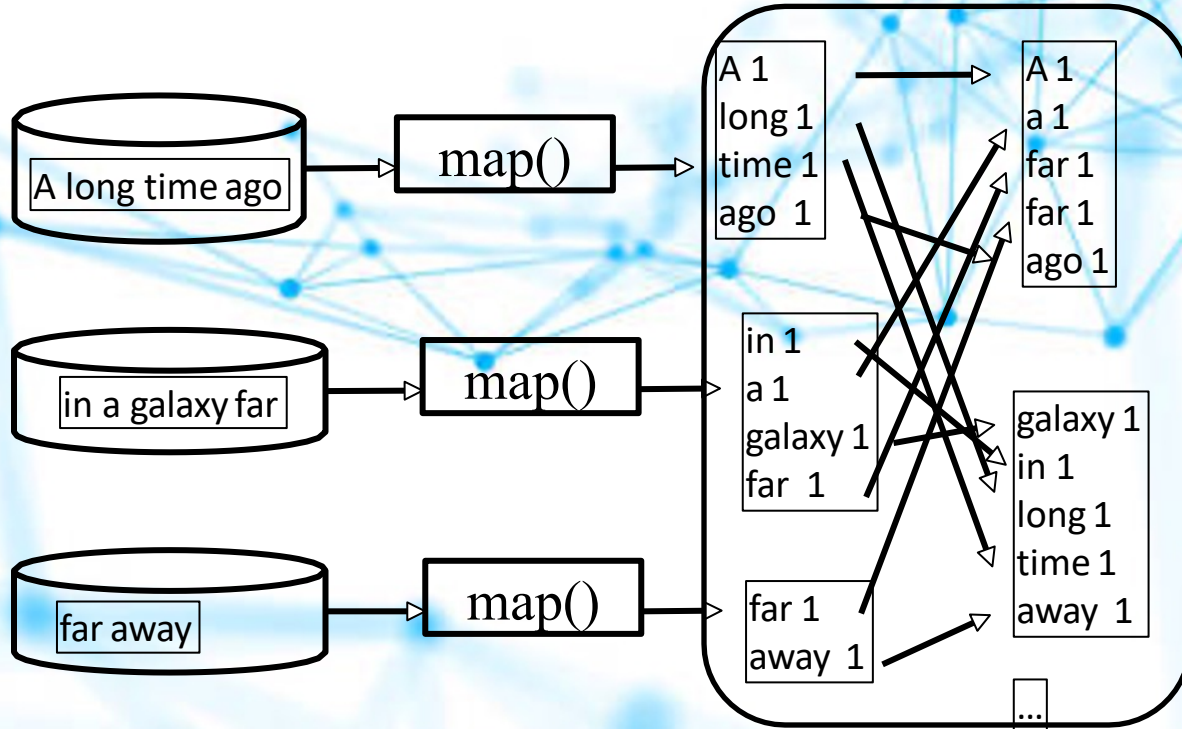
else

***emit <word> < count>
set count to 0***

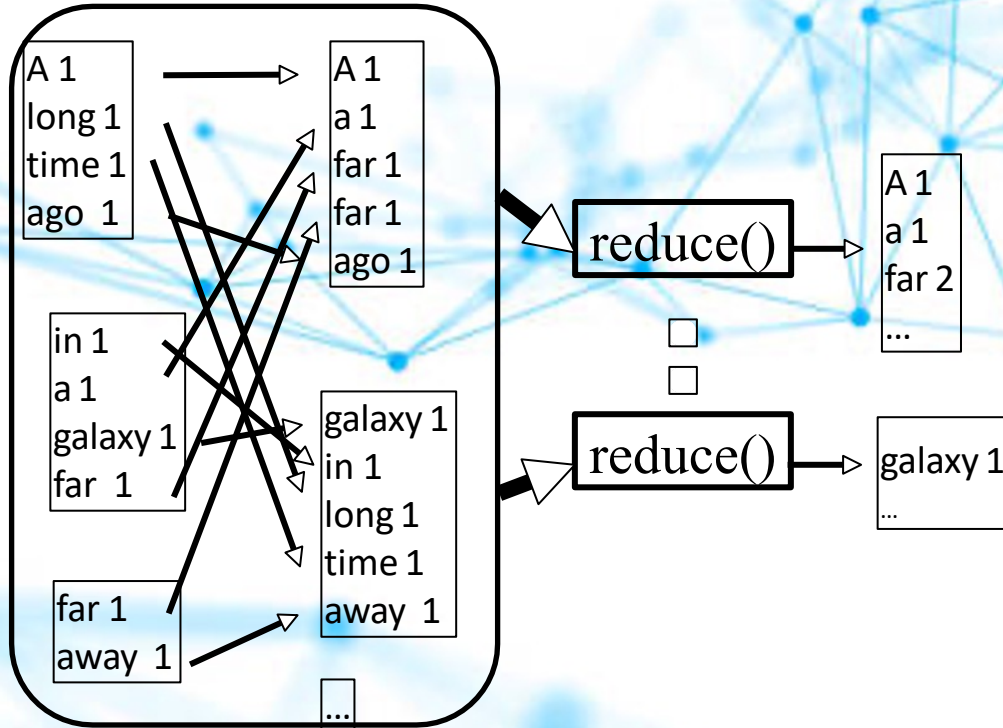
map() output

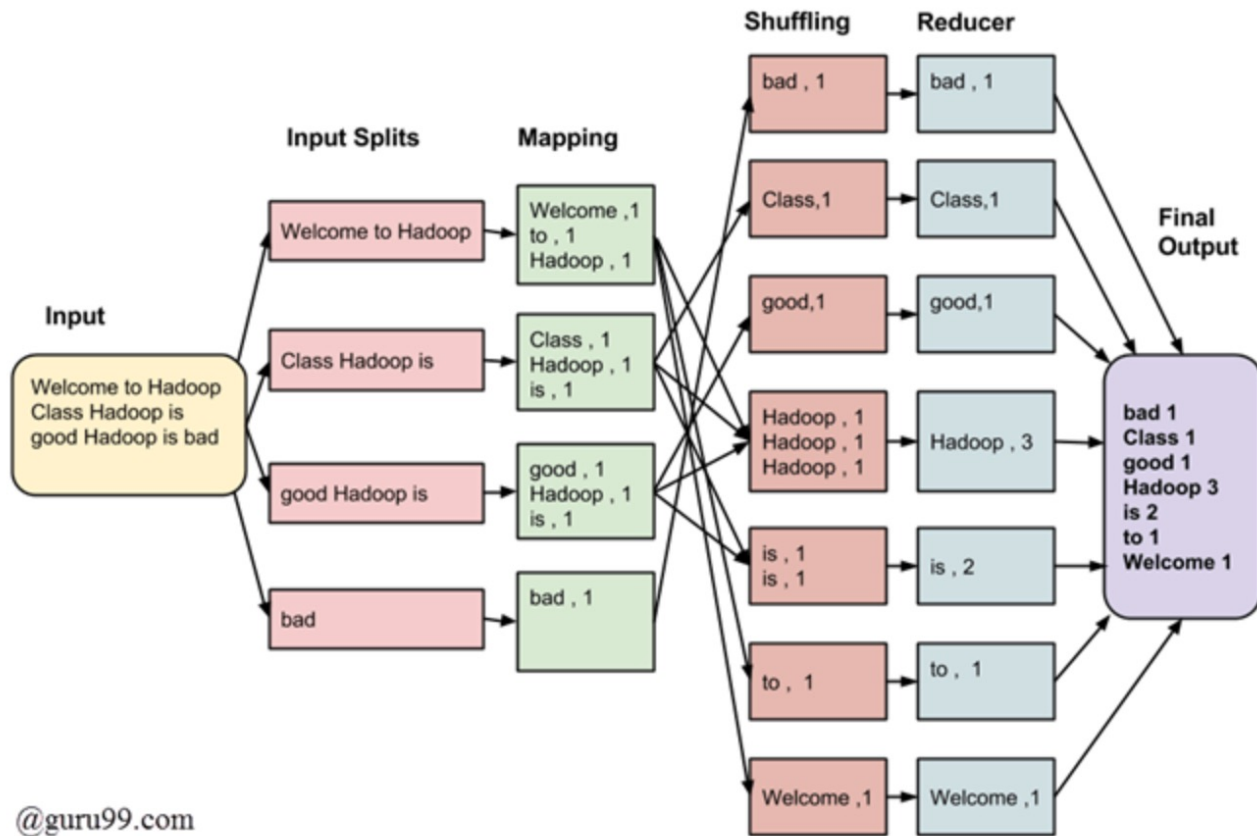


Hadoop shuffles, groups, and distributes



reduce() aggregates





@guru99.com

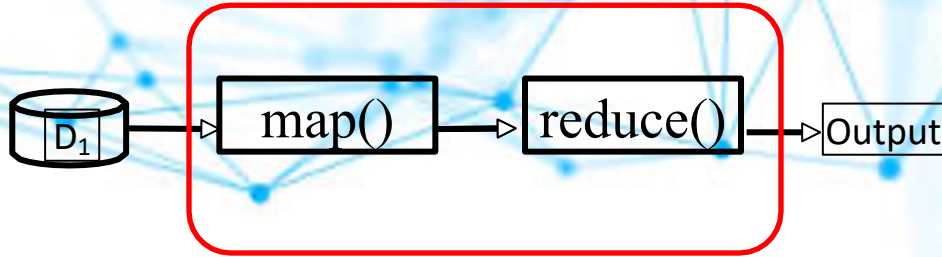
Introduction to Map/Reduce



Examples and Principles

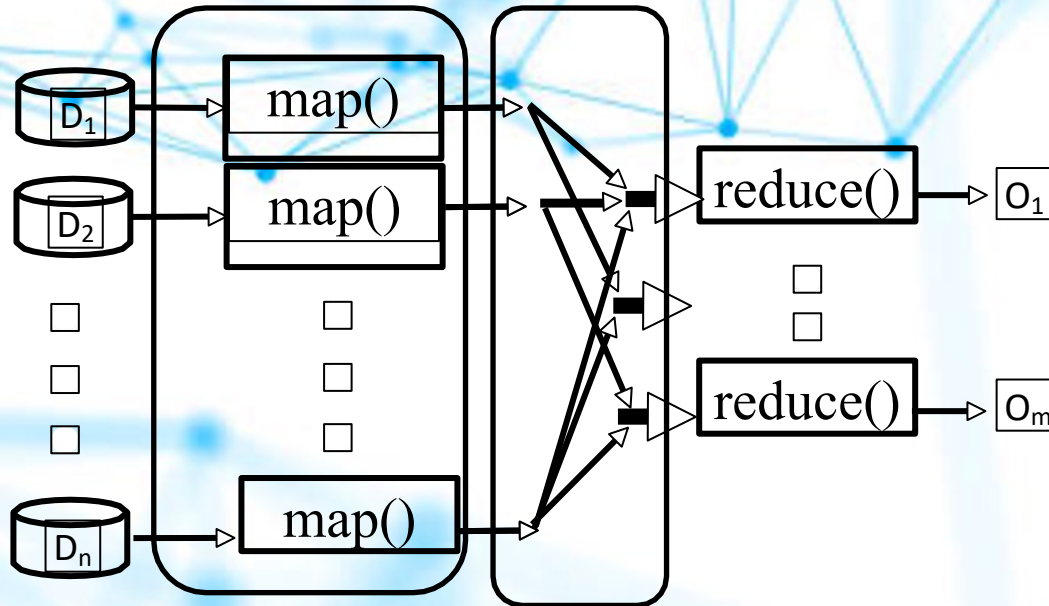
Recall the framework:

- User defines $\langle \text{key}, \text{value} \rangle$, mapper, and reducer



Recall the framework:

- Hadoop handles the logistics



Hadoop Rule of Thumb



- 1 mapper per data split (typically)

Hadoop Rule of Thumb

- 1 mapper per data split (typically)
- 1 reducer per computer core (best parallelism)

Hadoop Rule of Thumb

- 1 mapper per data split (typically)
- 1 reducer per computer core (best parallelism)

*Processing
Time*

*Number
Output Files*

Wordcount Strategy

- Let $\langle \text{word}, 1 \rangle$ be the $\langle \text{key}, \text{value} \rangle$
- Simple mapper & reducer
- Hadoop did the hard work of shuffling & grouping

Good key-value properties

- Simple
- Enables reducers to get correct output

*Shuffling &
Grouping*

*Key-Value
simplicity*

Good Task Decomposition:

Mappers: simple and separable



Reducers: easy consolidation





Example: Trending Wordcount

Trending Wordcount



- Twitter Data: date, message, location, ... [other metadata]

Trending Wordcount

- Twitter Data: date, message, location, ... [other metadata]

Task 1 Get word count by day

Task 2 Get total word count

Trending Wordcount

The background of the slide features a complex, abstract network graph. It consists of numerous small blue circular nodes connected by thin, light blue lines. The nodes are distributed across the frame, with a higher density in the upper right and lower right areas, and a more sparse arrangement towards the left. The lines vary in length and orientation, creating a web-like structure that suggests interconnectedness and data flow.

Task 1: get word count by day

Trending Wordcount

Task 1: get word count by day

Design: *Use composite key*

Map/Reduce: <date word,count>

Trending Wordcount

An abstract background graphic featuring a network graph. It consists of numerous small blue circular nodes connected by thin, light blue lines. The nodes are distributed across the frame, with a higher density in the upper right and lower left areas, creating a sense of interconnectedness and flow.

Task 2: get total word count

Trending Wordcount



Task 2: get total word count

Easy way:

re-use previous wordcount

Trending Wordcount

Task 2: get total word count

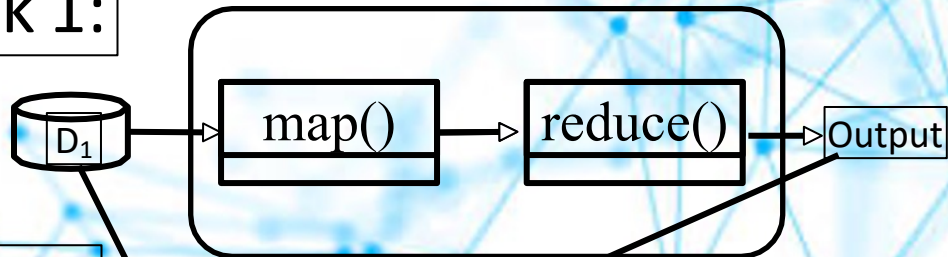
Alternatively:

use Task 1 output

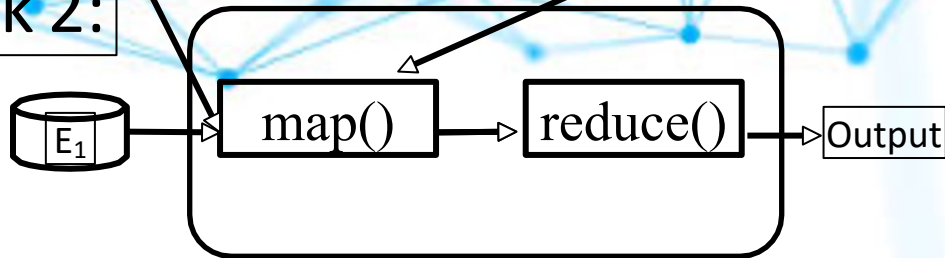
(it's partially aggregated)

Cascading Map/Reduce

Task 1:



Task 2:



Task 3 ...

Example: Joining Data

Joining Data



- Task: combine datasets by key
 - A standard data management function

Joining Data

- Task: combine datasets by key
 - A standard data management function
 - In pseudo SQL

Select * from table A, table B, where
A.key=B.key

Joining Data

- Task: combine datasets by key
 - A standard data management function
 - In pseudo SQL
 - Select * from table A, table B, where
A.key=B.key
 - Joins can be inner, left or right outer

Joining Data



- Task: given two wordcount datasets ...

Joining Data

- Task: given two wordcount datasets ...

File A: <word, total-count>

able, 5

actor, 18

burger, 25

.

.

.

Joining Data

- Task: given two wordcount datasets ...

File A: <word, total-count> **File B: <date word, day-count>**

```
able , 5  
actor , 18  
burger , 25  
.  
.  
.
```

```
Jan-16 able , 2  
Feb-22 actor , 15  
May-03 actor , 3  
Jul-4 burger, 20  
.  
.  
.
```

Joining Data

- Task: combine by word

File A: <word, total-count> **File B: <date word, day-count>**

able, 5

actor, 18

burger, 25

.

.

.

Jan-16 able, 2

Feb-22 actor, 15

May-03 actor, 3

Jul-04 burger, 20

.

.

.

Joining Data

- Result wanted:

File AjoinB: <word date, day-count total-count >

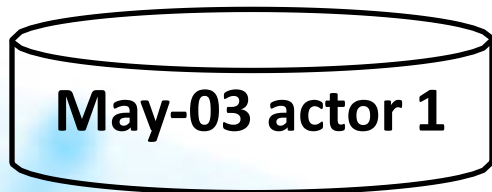
able	Jan-16,	2	5
actor	Feb-22,	15	18
actor	May-03,	1	18
burger	Jul-04,	20	25
.			
.			
.			

Joining Data

- Recall that data is split in parts



*How to gather
the right pieces?*



Key-Value & Task Decomposition

- Main design consideration:

Join depends on word
*(e.g. Select * where A.word=B.word)*

Key-Value & Task Decomposition

- For the join:
 - Let $\langle \text{key} \rangle$ = word
 - Let $\langle \text{value} \rangle$ = other info

$\langle \text{word}, \dots \rangle$

Key-Value & Task Decomposition

File A: <word, total-count>

```
able , 5  
actor , 18  
...
```

File B: <date word, day-count>

```
Jan-16 able , 2  
Feb-22 actor , 15  
...
```

Key-Value & Task Decomposition

- Note:

File A: <word, total-count>

able , 5
actor , 18
...

File B: <date word, day-count>

Jan-16 able , 2
Feb-22 actor , 15
...

word already the key

Key-Value & Task Decomposition

File A: <word, total-count>

```
able , 5  
actor , 18  
...
```

File B: <date word, day-count>

```
Jan-16 able , 2  
Feb-22 actor , 15  
...
```

date needs to be filtered out

Key-Value & Task Decomposition

File A: <word, total-count>

able , 5
actor , 18
...

File B: <date word, day-count>

Jan-16 able , 2
Feb-22 actor , 15
...

date needs to be filtered out
Where should date info go?

Key-Value & Task Decomposition

<word, date day-count total-count >



Task Decomposition

- Now data sets are:

File A: <word, total-count> **File B_new: <word, date count>**

```
able ,    5
actor ,   18
burger ,  25
.
.
.
```

```
able , Jan -16 2
actor , Feb-22 15
actor , May-03 3
burger , Jul-04 20
.
.
.
```

Task Decomposition

- How will Hadoop shuffle & group these?

File A: <word, total-count> **File B_new: <word, date day-count>**

```
able ,    5
actor ,   18
burger ,  25
.
.
.
```

```
able , Jan-16  2
actor , Feb-22 15
actor , May-03  3
burger , Jul-04 20
.
.
.
```

Task Decomposition

- How will Hadoop shuffle & group these?

Let's focus on 1 key:

actor , 18

actor , Feb-22 15
actor , May-03 3

Task Decomposition

- Hadoop gathers the data for a join

actor , 18

actor , Feb-22 15
actor , May-03 3

actor , Feb-22 15
actor , 18
actor , May-03 3

Task Decomposition

- Reducer now has all the data for same word grouped together

A number or date



```
actor , 18  
actor , Feb-22 15  
actor , May-03 3
```

Task Decomposition

- Reducer can now join the data and put date back into key



Example: Vector Multiplication