

15-440

Distributed Systems

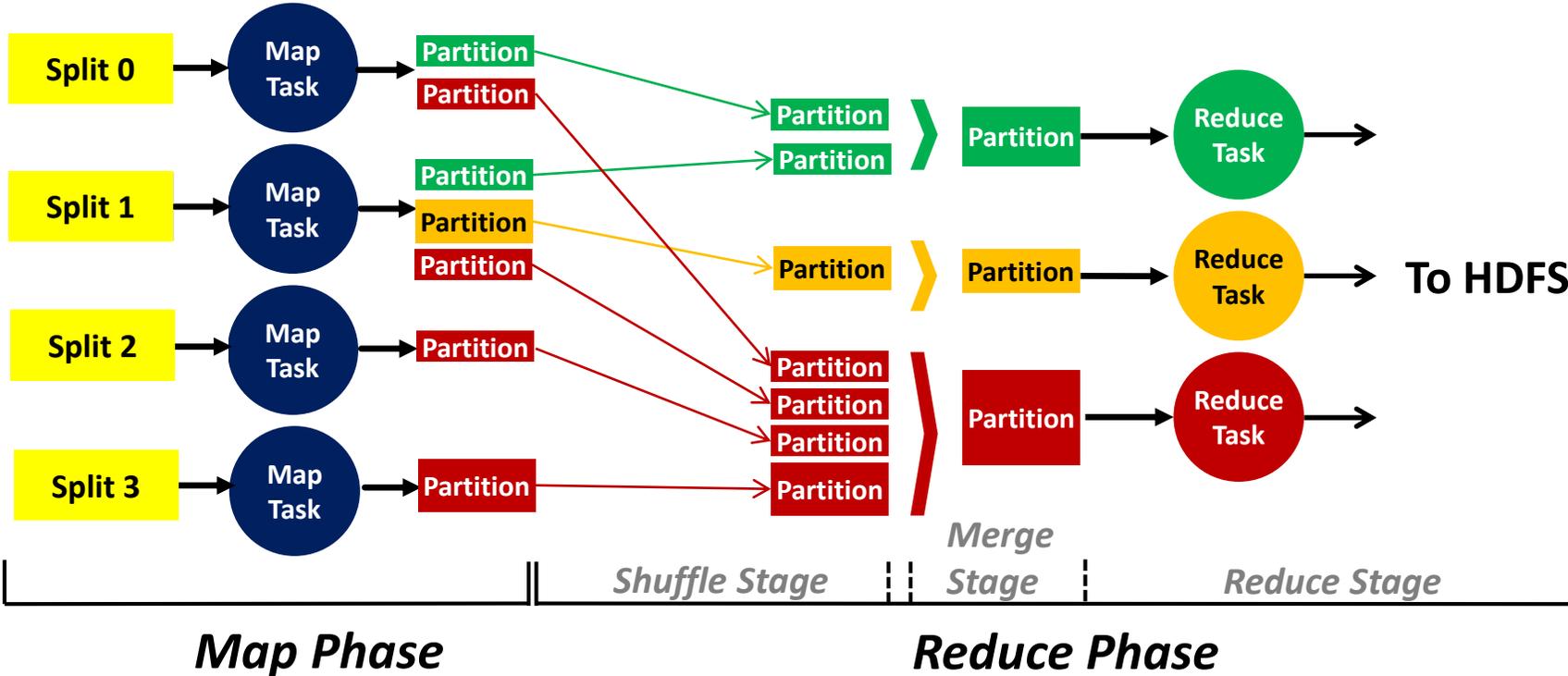
Recitation 10

Zeinab Khalifa

Project 4

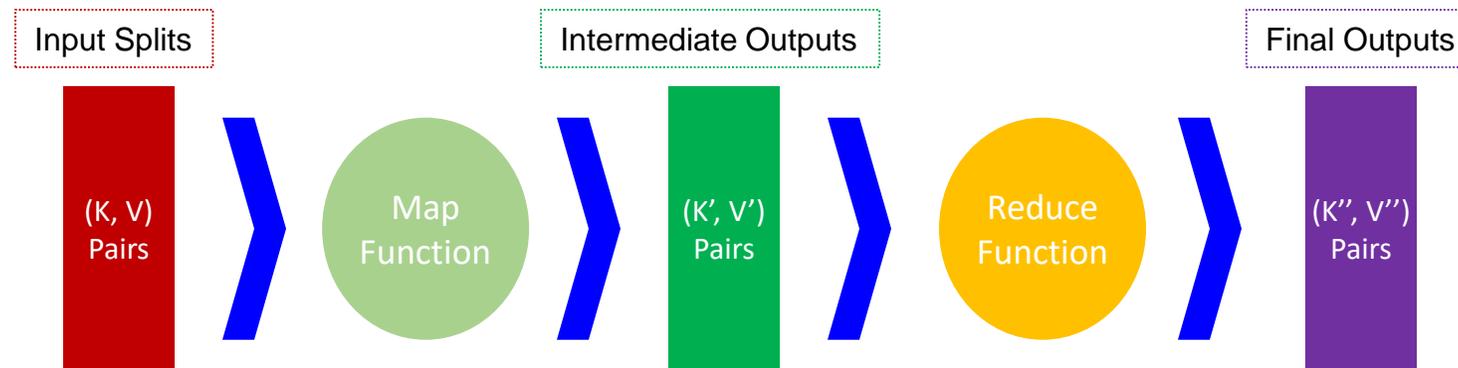
Apply **MapReduce** to cluster analysis, using the **K-Means** algorithm

MapReduce: A Systems View



Data Structure: Keys and Values

- In a MapReduce program, the programmer has to specify two functions: the **Map function** and the **Reduce function** that implement the Mapper and the Reducer, respectively
- In MapReduce, data elements are always structured as key-value (i.e., (K, V)) pairs
- Therefore, the Map and Reduce functions *receive* and *emit* (K, V) pairs



MapReduce: An Application View

A Chunk of File

Tamim is delivering a recitation to the 15-440 class

A Map Function

Key1	Value1
0	Tamim is
20	delivering a
38	recitation to
60	the 15-440 class

Parse & Count

Key2	Value2
Tamim	1
is	1
delivering	1
a	1
recitation	1
to	1
the	1
15-440	1
class	1

A Reduce Function

Iterate & Sum

Key	Value
Tamim	1
is	2
delivering	1
a	1
recitation	1
to	1
the	2
15-440	2
class	1
course	1
name	1
of	1
Distributed	1
Systems	1

A Chunk of File

The course name of 15-440 is Distributed Systems

A Map Function

Key1	Value1
0	The course
17	name of 15-440
40	is Distributed
58	Systems

Parse & Count

Key2	Value2
The	1
course	1
name	1
of	1
15-440	1
is	1
Distributed	1
Systems	1

WordCount.java (Helpers)

- **Scanner Object:**

- A Scanner breaks its input into tokens using a delimiter pattern, which matches whitespace by default.
- hasNext(): checks if the Scanner has another token in its input.
- next(): gets the next token

- **MR Text object:**

- .set(token): sets a token to a Hadoop Text object

- **OutputCollector<Text, IntWritable> object:**

- .collect(x, y) sets a text x and Int y (k,v) pair output to the reduce function