SIM GLOBAL EDUCATION

UNIVERSITY OF WOLLONGONG AUSTRALIA

ISIT312 – Big Data Management

Tutorial - MapReduce

Sionggo Japit

sjapit@uow.edu.au

8 April 2021

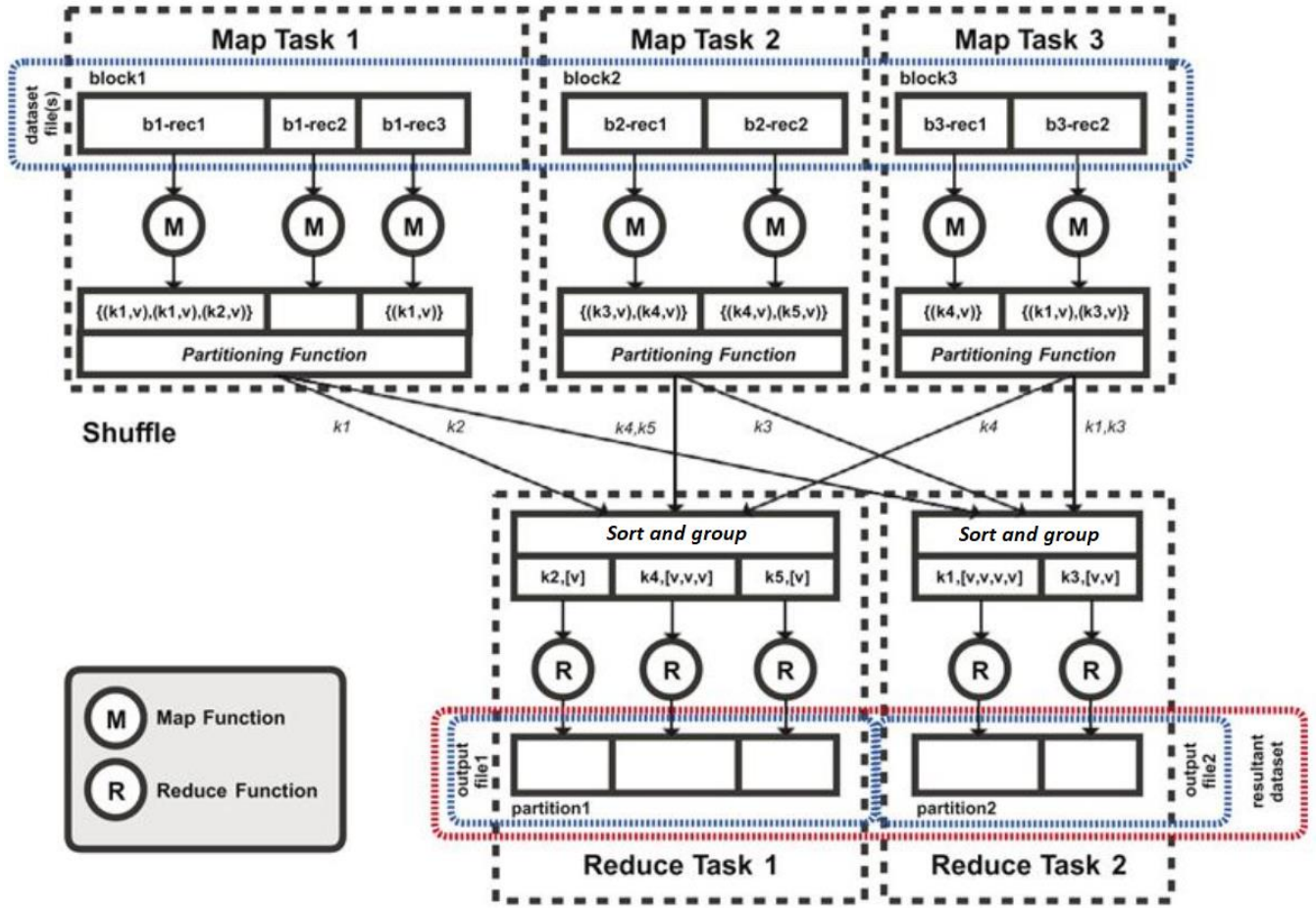
ISIT312-Big Data Management

MapReduce

What is a MapReduce?

- MapReduce
  - Software framework and programming model used for processing a very large amounts of data.
  - It works in two phases:
    - Mapping phase
      - In this phase, data is split and mapped into multiple groups.
    - Reduce phase
      - In this phase, the mapped data are shuffle and reduced individual groups.
  - MapReduce programs are parallel in nature, hence, they are useful for performing large-scale data analysis using multiple machines in the cluster.

MapReduce


Map Task 1
Map Task 2
Map Task 3
dataset file(s)
block1
b1-rec1
b1-rec2
b1-rec3
block2
b2-rec1
b2-rec2
block3
b3-rec1
b3-rec2
M
{(k1,v),(k1,v),(k2,v)}
{(k1,v)}
{(k3,v),(k4,v)}
{(k4,v),(k5,v)}
{(k4,v)}
{(k1,v),(k3,v)}
Partitioning Function
Shuffle
k1
k2
k4,k5
k3
k4
k1,k3
Sort and group
k2,[v]
k4,[v,v,v]
k5,[v]
k1,[v,v,v,v]
k3,[v,v]
R
output file1
output file2
resultant dataset
partition1
partition2
Reduce Task 1
Reduce Task 2
Map Function
Reduce Function

MapReduce

Input Splits:

- Data set is divided into fixed-size chunk (block) that is consumed by a single map.

Mapping:

- Data in each chunk is passed to a mapping function to produce counts of occurrences of each word, and prepare a list of key-value pair where key is the word, and value is the frequency of occurrences.

Shuffling

- Shuffling process will consolidate the relevant records from Mapping phases by clubbing together the same words and accumulate their frequency.

MapReduce

Reducing

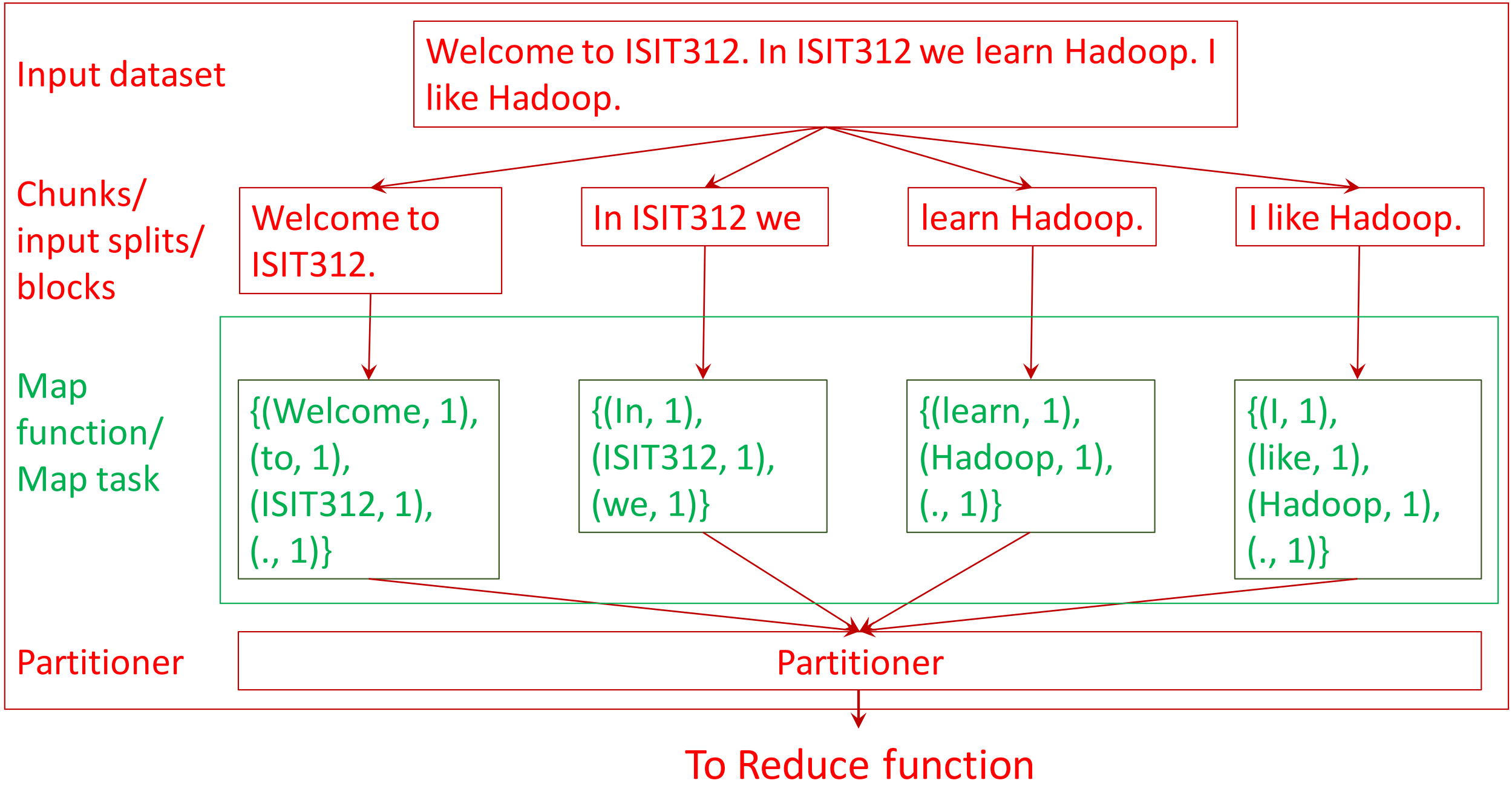
- In this phase, the output values from the Shuffling phase are aggregated, by combining all the words into a single output, that is, producing a complete dataset.

Example

MapReduce (Without Combiner) – Map Phase

- The following example depicts a MapReduce function without a combiner in the Map function.
- Dataset is split into blocks/chunks of fixed size.
- Map function is then assigned to process each block, that is, each Map function operates only one block.
- Each Map function outputs (produces) sets of key-value pair records.
- If without combiner, the sets of key-value pair records are passed to *Patitioner*, which ensures each key-value pair record is passed to one and only one Reducer.

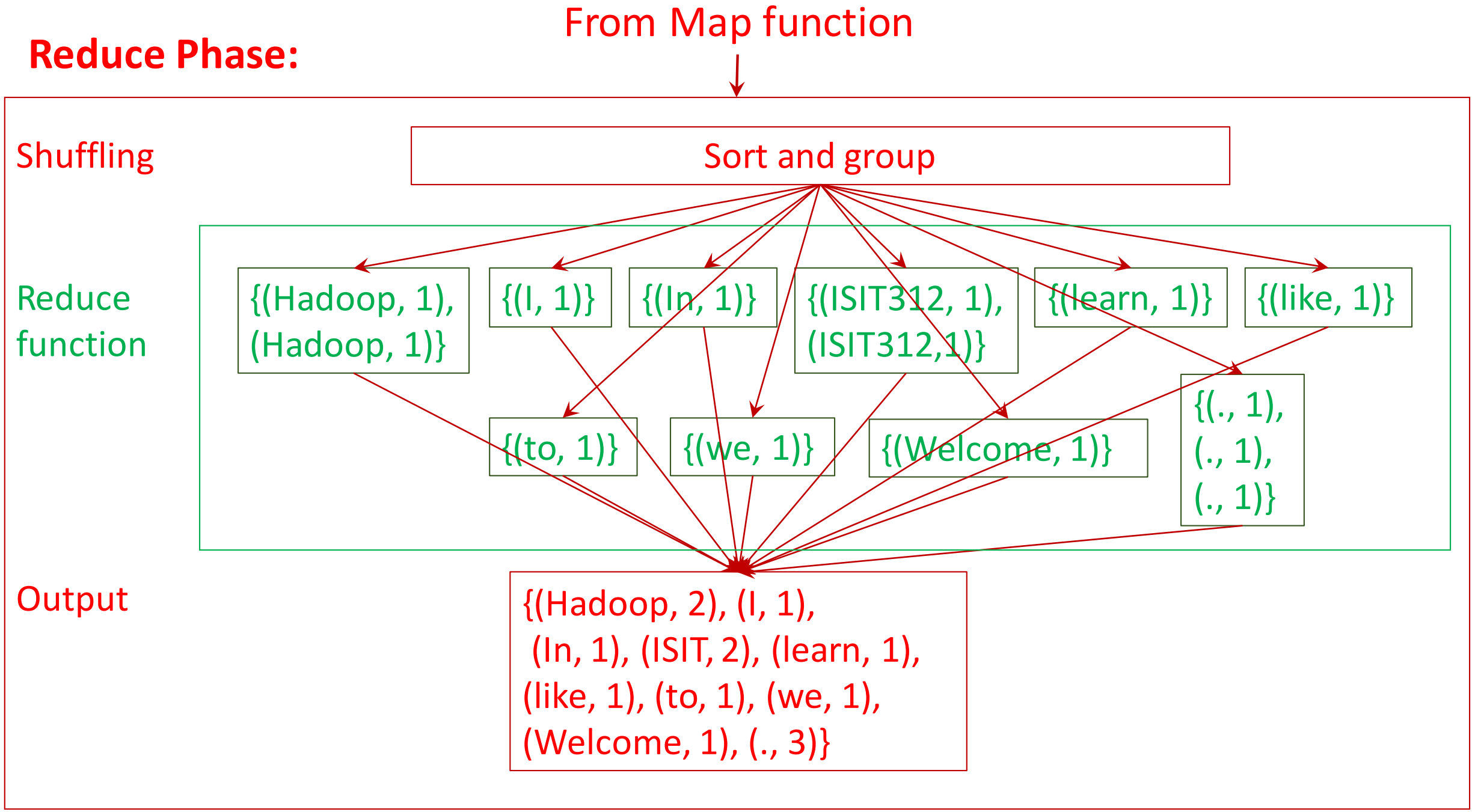
Map Phase:


Input dataset
Welcome to ISIT312. In ISIT312 we learn Hadoop. I like Hadoop.
Chunks/ input splits/ blocks
Welcome to ISIT312.
In ISIT312 we
learn Hadoop.
I like Hadoop.
Map function/ Map task
{(Welcome, 1), (to, 1), (ISIT312, 1), (., 1)}
{(In, 1), (ISIT312, 1), (we, 1)}
{(learn, 1), (Hadoop, 1), (., 1)}
{(I, 1), (like, 1), (Hadoop, 1), (., 1)}
Partitioner
Partitioner
To Reduce function

MapReduce (Without Combiner) – Reduce Phase

- Input to Reduce Phase is the output from Map Phase, that is, the *Partitioner* ensures each key-value pair is passed to one and only one Reducer.
- The Reduce will perform a sort and group function to sort and group the key-value pair by the key and accumulate the values for each key.
- The Reduce function will then output a set consisting of all the key-value pairs.
- Note: A reducer may be receiving input from multiple Map functions.

Reduce Phase:

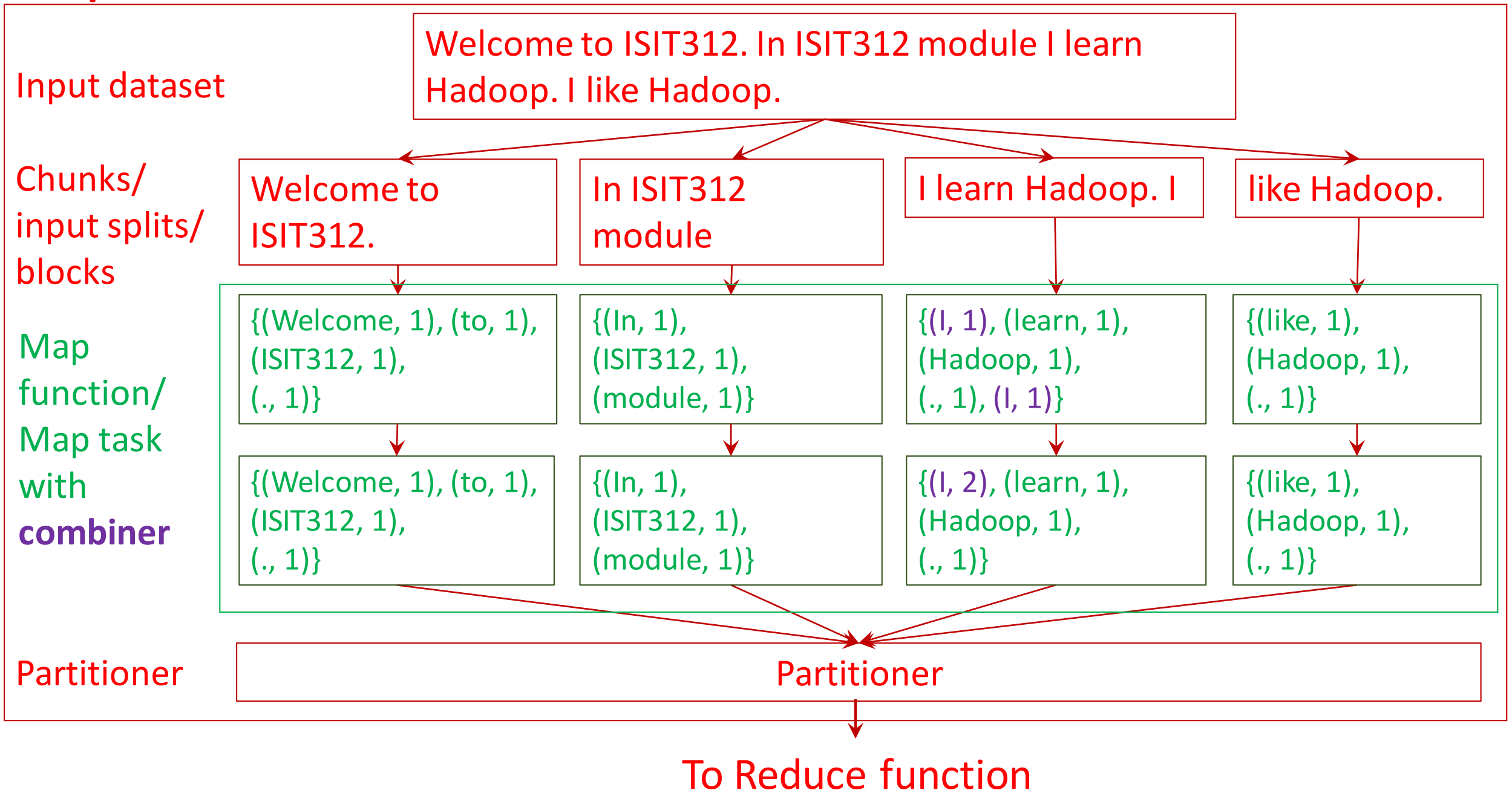


Example

MapReduce (With Combiner) – Map Phase

- The following example depicts a MapReduce function with a combiner in the Map function.
- Dataset is split into blocks/chunks of fixed size.
- Map function is then assigned to process each block, that is, each Map function operates only one block.
- Each Map function outputs (produces) sets of key-value pair records.
- Combiner performs sum or count function to combine each key and its value before passing the key-aggregateValue pairs to *Patinioner*, which ensures each key-aggregateValue pair record is passed to one and only one Reducer.

Map Phase:


Input dataset
Welcome to ISIT312. In ISIT312 module I learn Hadoop. I like Hadoop.
Chunks/ input splits/ blocks
Welcome to ISIT312.
In ISIT312 module
I learn Hadoop. I
like Hadoop.
Map function/ Map task with combiner
{(Welcome, 1), (to, 1), (ISIT312, 1), (., 1)}
{(In, 1), (ISIT312, 1), (module, 1)}
{(I, 1), (learn, 1), (Hadoop, 1), (., 1), (I, 1)}
{(like, 1), (Hadoop, 1), (., 1)}
{(Welcome, 1), (to, 1), (ISIT312, 1), (., 1)}
{(In, 1), (ISIT312, 1), (module, 1)}
{(I, 2), (learn, 1), (Hadoop, 1), (., 1)}
{(like, 1), (Hadoop, 1), (., 1)}
Partitioner
Partitioner
To Reduce function

MapReduce (With Combiner) – Reduce Phase

- Input to Reduce Phase is the output from Map Phase, that is, the *Partitioner* ensures each key-value pair is passed to one and only one Reducer.
- The Reduce will perform a sort and group function to sort and group the key-value pair by the key and accumulate the values for each key.
- The Reduce function will then output a set consisting of all the key-value pairs.
- Note: A reducer may be receiving input from multiple Map functions.

Reduce Phase: