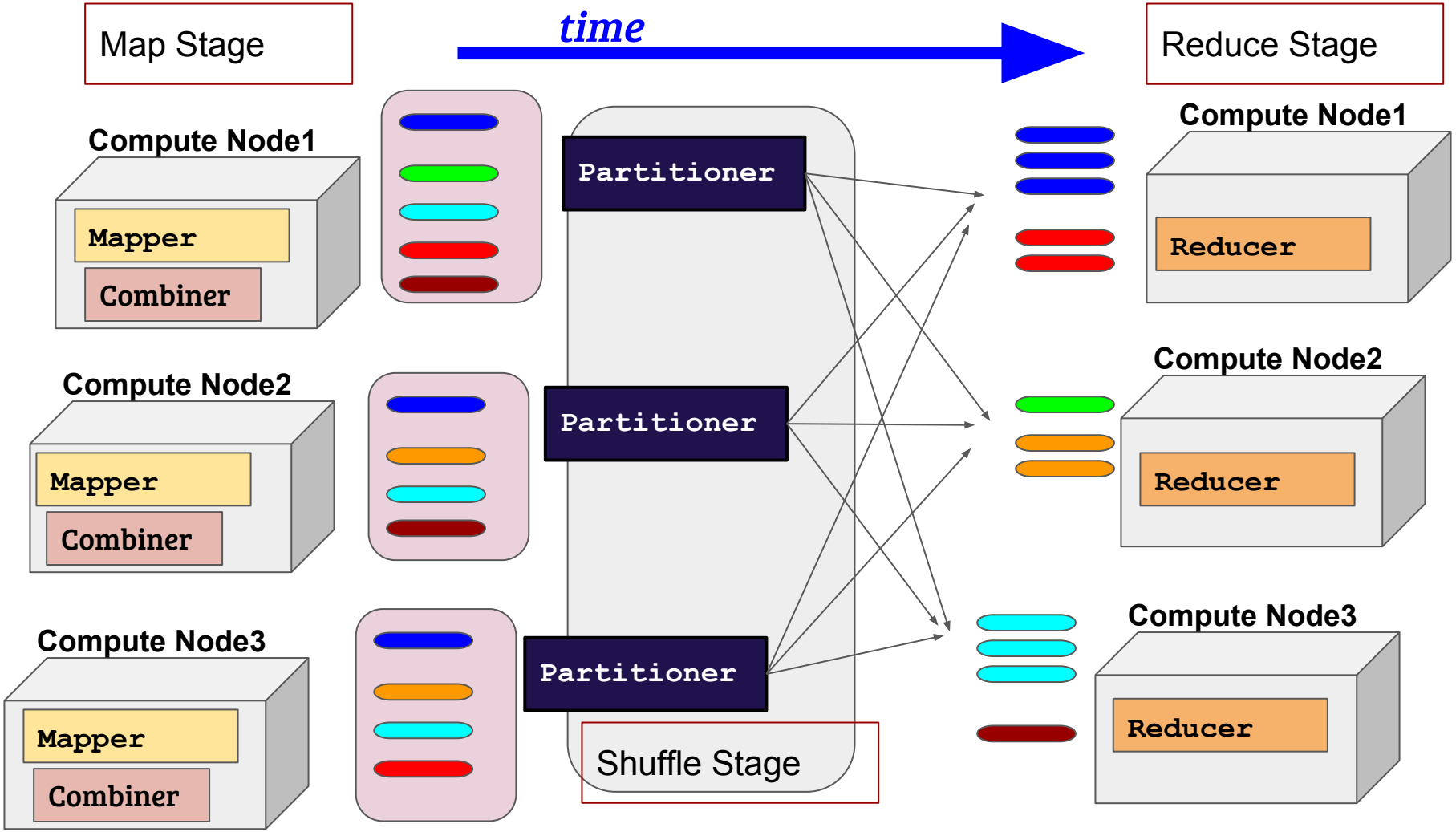
CSC 369

Introduction to Distributed Computing


Map Stage
time
Reduce Stage
Compute Node1
Mapper
Combiner
Compute Node2
Mapper
Combiner
Compute Node3
Mapper
Combiner
Partitioner
Partitioner
Partitioner
Shuffle Stage
Compute Node1
Reducer
Compute Node2
Reducer
Compute Node3
Reducer

Combiner Function

- A **combiner function** is executed between the map and reduce functions.
  - Purely for optimization.
  - Input and output of the combiner must be of the same type.

- Values with the same key are aggregated at each of the map nodes before data is sent to the reducer.
  - Decrease traffic and make the process faster.
  - Unnecessary if the mapper and reducer are on the same node.

Combiner Example - Average Temperature

Input: Date Temperature
(multiple readings per day)

2022-04-21 60
2022-04-21 71
2022-04-21 75
2022-04-22 54

2022-04-22 66
2022-04-22 68
2022-04-22 71
2022-04-22 55
2022-04-23 57

2022-04-23 64
2022-04-23 70
2022-04-23 71
2022-04-23 50
2022-04-21 71

Output: Date, Average Temperature

2022-04-21 67.0
2022-04-22 62.8
2022-04-23 62.4
2022-04-24 61.6
2022-04-25 59.6

Combiner Example - Average Temperature

- On each map node, we will compute the sum of all temperatures and the temperature count for each day.
- We will need a new container class to store (temperature sum / temperature count).
- Send this data to the reduce function (sending less data).
- The reduce functions will sum all the temperatures and their counts and divide the two numbers.

Combiner Example - Average Temperature

https://github.com/amigler/369-hadoop3

Combiner Summary

- The type of the key or value can be a custom class.
- Create custom classes by implementing the `Writable` interface (`WritableComparable` for keys).
- Combiner aggregates output of the map function locally before being sent (i.e., shuffled) to the reduce nodes.
- Combiner methods are just for optimization and they *may be skipped* by the system (e.g., mapper and reducer node are the same).

Partition Function

- A partitioner controls how the (key,value) output of the mapper is distributed among the reducer nodes.
  - The default partition function partitions the output based on the hash code of the key.

- Custom partitioning is useful when we don't want to partition based on the key (e.g., we want to look at a subset of the key).
- We need to define the `getPartition` function.
  - Takes as input (key,value) pair as returned by mapper, and the number of partitions (n).
  - Returns the ID (i.e., 0, 1, ..n-1) of the reducer node where the record should be sent.

Sorting / Grouping in Partitioner

- By default, the input to each reducer is **sorted** by key.
- Sometimes useful to redefine how keys are sorted during shuffle stage

- By default, all records with the same key are **grouped** together and passed as input to a single call to the reduce method.
- We can override this behavior (e.g., only look at a subset of the key).

Partitioner Example

- Consider the following data (year, month, day, temperature)
  - 2021,01,01,5
  - 2021,01,02,45
  - …
- We want to group by year-month and print all the temperatures sorted in ascending order (i.e., we ignore the day).
  - Expected output:
  - 2021-01: 5,45 ...

Partitioner Example - Monthly Temperatures

2021, 01, 01, 5
2021, 01, 32, 5
2021, 01, 02, 45
2021, 01, 03, 35
2021, 01, 04, 10
2021, 11, 01, 46
2021, 11, 02, 47
2021, 11, 03, 48
2021, 11, 04, 40
2019, 08, 20, 50
2019, 08, 21, 52
2019, 08, 22, 38
2019, 08, 23, 70

Ascending sort (by temperature reading) →

Ascending sort (by month) ↓

2019-08 38, 50, 52
2021-01 5, 5, 10, 35, 45
2021-11 40, 46, 47, 48

Natural and Composite Keys

- The **natural key** is the (year, month) pair. This is what the final result is grouped by
- The **composite key** is (year, month, temperature). This is what will be the output key of mapper. We will also define the sorting order on the composite key.
- Usually, the natural key is a subset of the composite key.
- We create a composite key when we want to sort on something more than just the natural key.

Composite Key Class

- Note that we can only specify a sorting order on the key of the output of the mapper.
- If we want the sorting to include more attributes, than we need to **extend** the key.
- In our example, we cannot define the key to be just year-month because then we will not be able to sort the result on temperature.
- The custom class for the composite key will implement Writable (all custom key classes need to) and `WritableComparable` (to define sort order on the composite key).

Partitioner Implementation

- Takes as input the (k,v) that is returned by mapper.
- Returns the partition where to send the record.
- Guarantees that we are partitioning based on YearMonth (the natural key) and not based on YearMonthTemperature (the composite key).
- That is, everyone with the same value for year and month should go to the same reducer node.
- We define the getPartition function that returns the partition number.

Comparators - Sort & Grouping

- Sort based on composite key (year-month-temperature)
- Group based on natural key (year-month)

Temperature Sort Reducer

- All the data will come sorted by year, month, temperature to each reducer node. (guaranteed by sorter comparator).
- The data will have the same year and month (guaranteed by grouping comparator) in each call to reduce.
- Each reducer will keep the year and month and concatenate all the temperatures (the temperatures will be sorted).
- The reducers do not need to sort the temperatures because they will arrive in sorted order.

TemperatureSort Summary

- **mapper**: sends records with composite key = (year, month, temperature) value = temperature
- **sorter**: sorts the records by the composite key
- **partitioner**: partitions the data to the reducer nodes based on (year, month)
- **grouping comparator**: says that each reducer call will take as input the records with the same value for (year, month).
- **reducer**: takes as input the records with the same value for year, month. Makes (year, month) the key and it concatenates all the temperatures to form the value.
- Result of each mapper will be sorted by (year, month, temperature). For each (year, month), temperatures will be sorted.