

Lecture 15.2

Hadoop! Semantics

EN 600.320/420

Instructor: Randal Burns

4 April 2018



Department of Computer Science, *Johns Hopkins University*

Hadoop! Terms

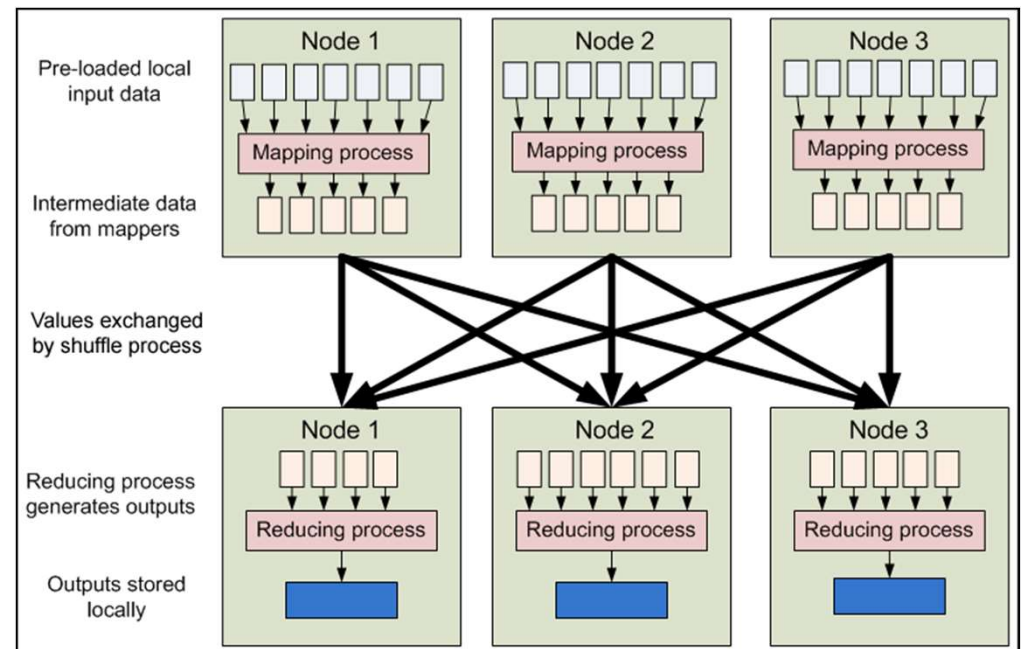
- Combiner
- Shuffle
- Partition`



Hadoop! Terms

- Combiner

- A reducer (like) function that runs at the mapper
- Used mostly with reducers that are composable to run the reducer many times
- Makes input to the shuffle smaller, reduce data handling
- Combiner typically more parallel



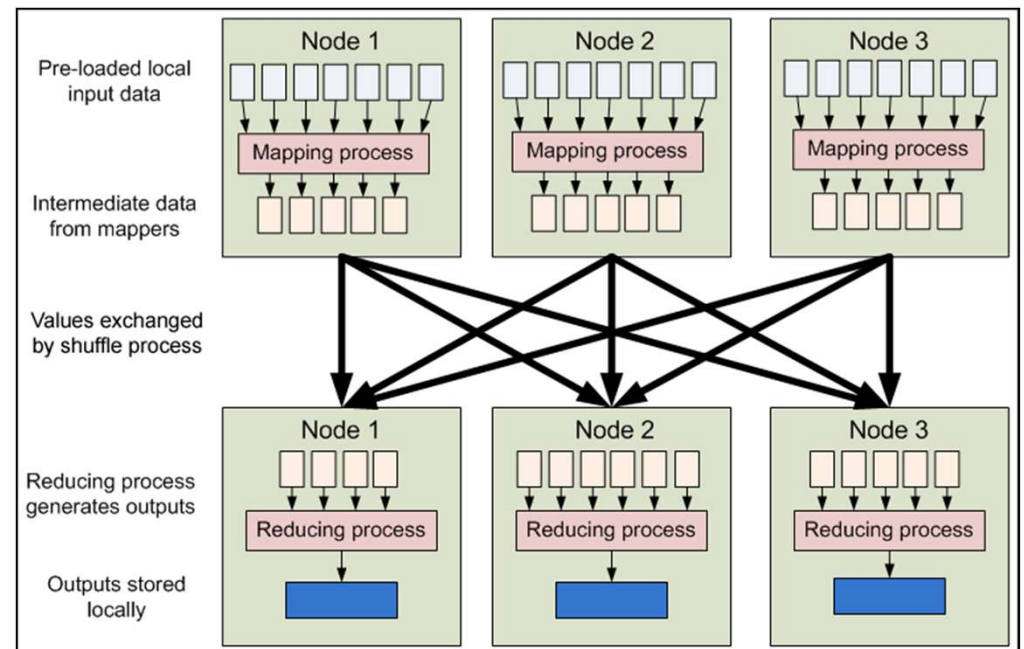
Hadoop! Combiner

- Combiner executes on the mappers $\langle \text{key}, \text{value} \rangle$ output while in memory at the mapper
 - It is possible to write unique combiner and reduce classes
 - It is common to use the reducer as a combiner
- Typical use: pre-compute aggregates or extrema
 - Word count: compute sum output by mapper for each key and send a single aggregated value to reducers
 - Maximum: compute maximum value for each key. Reducer computes a maximum of maxima.
- Caution!!! Your homework assignment cannot use a combiner. I will ask you why?



Hadoop! Terms

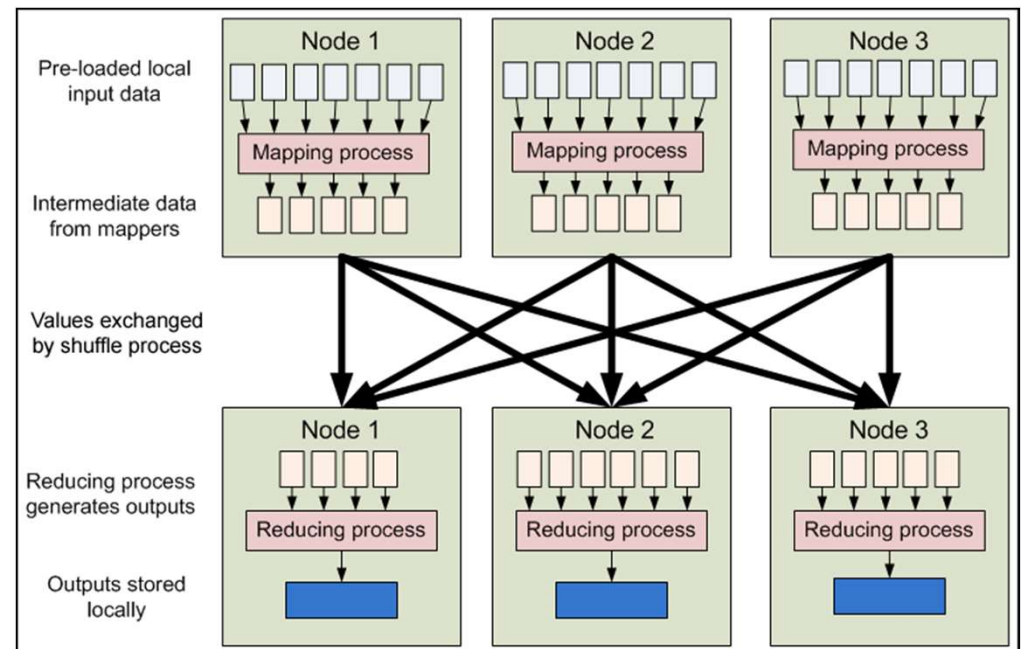
- Shuffle
 - The process of routing mapper outputs to the reducer inputs



Hadoop! Terms

- Partition

- The portion of data associated with a parallel execution unit
- Output partition: the part of the result written by each reducing process



Sorting by Key

- Map: extract a sorting key from the value
 - $\langle \text{key}, \text{value} \rangle \rightarrow \langle \text{sort_key}, \text{value} \rangle$
- Reduce: use shuffling properties of reduce
 - $\langle \text{sort_key}, \text{value} \rangle \rightarrow \langle \text{sort_key}, \text{value} \rangle$
- What's going on?

4.2 Ordering Guarantees

We guarantee that within a given partition, the intermediate key/value pairs are processed in increasing key order. This ordering guarantee makes it easy to generate a sorted output file per partition, which is useful when the output file format needs to support efficient random access lookups by key, or users of the output find it convenient to have the data sorted.



Observations about Sorting

- The ordering guarantees sort within partitions
- To sort completely:
 - All output to a single partition (use one reducer)
 - Customize the shuffle function (quite complex)
 - The default shuffle uses hashing (for load balance)
- The Google paper optimizes sort by customizing shuffle so that partitions are ordered, not randomized
 - Run a M/R job to learn the key distribution
 - Specify a shuffle based on evenly partitioning the key distribution
 - This is also how Hadoop!'s sort record worked

