Project L2 Part A

R09922029 彭梓瑄

i. What does this Map-Reduce code below achieve? Fill in the 2 words in ___ ___ computation.

Input: doc_id with the corresponding words in the doc_id: w1, w2, ... E.g., doc_1 has {word1, word2, ...}

Mapper: map((doc_id, {w1, w2,...})) -> list((w1, doc_id), (w2, doc_id), ...)

Reducer: reducer(word, [doc1, doc2, ...]) -> (word, [doc1, doc2, ...])

ANS:
Inverted index

ii. Describe how you can improve this Map-Reduce code.
ANS:

- Use Shuffler:
  Shuffle the (word, doc_id) tuples, and group tuples by the word.
  It will send values of the same key to a particular Reducer machine, and group (and sort) the values with the same key.
- Use Combiner to improve performance, which reduce the network traffic between Mappers and Reducers.
  Combine output data at local machine, before emitting to Shuffler.
  A combiner does not have a predefined interface and it must implement the Reducer interface' s reduce() method.
  A combiner operates on each map output key. It must have the same output key-value types as the Reducer class.
  A combiner can produce summary information from a large dataset because it replaces the original Map output.
  Although, Combiner is optional yet it helps segregating data into multiple groups for Reduce phase, which makes it easier to process.