Question. MapReduce

Definition

- A style of programming designed for
  - Easy parallel programming
  - Invisible management of hardware and softwarefailures
  - Easy management of very-large-scale data
- Implementations
  - Hadoop
  - Spark
  - "MapReduce" The original Google implementation

Distributed computing programming model

Componments

- Mapper: the Map function
  - Input
    - from the disks
    - <key, value> pairs
  - Output
    - Another set of intermediate <key. value> after production
- Reducer: the Reduce function
  - Input
    - <key, value> pairs
  - Output
    - <key. value> pairs

3 Steps

- 1. Map
  - Apply a user written Map function to each input element
    - Mapper applies the Map function to a single element
    - Many mappers grouped in a Map task(the unit of parallelism)
  - The output of the Map function is a set of 0, 1 or more key-value pairs

Read input and produces a set of key-value pairs

- 2. Group by key: Sort and shuffle
  - System sorts all the key-value pairs by key, and outputs key-(list of values) pairs

Collect all pairs with same key (Hash merge, Shuffle, Sort, Partition)

- 3. Reduce
  - User-written Reduce functions applies to each key-(list of values)

Collect all values belonging to the key and output

Example Question True of False

- Past Year Question
- Other

Each mapper/reducer must generate the same number of output key/value pairs as it receives on the input.	False. Mappers and reducers may generate any number of key/value pairs (including zero).
The output type of keys/values of mappers/ reducers must be of the same type as their input.	False. Mapper may produce key/value pairs of any type.
The inputs to reducers are grouped by key.	True. Reducers input key/value pairs is grouped by the key.
It is possible to start reducers while some mappers are still running.	False. Reducer's input is grouped by the key. The last mapper could theoretically produce key already consumed by running reducer
Mappers input key/value pairs are sorted by the key.	False. Mapper's input is not sorted in any way.
Reducer is applied to all values associated with the same key.	True. Reducer is applied to all values associated with the same key.
Reducers input key/value pairs are sorted by the key.	True. Reducers input key/value pairs are sorted by the key.
Each reducer must generate the same number of key/value pairs as its input had.	False. Reducer may generate any number of key/value pairs (including zero).
Reducers output key/value pair must be of the same type as its input.	False. The statement is false in Hadoop and true in Google's implementation.