

CS 410/510 Data Management in the Cloud

Spring 2018 Quarter

Assignment 3 MapReduce and Version Reconciliation

Due: Thursday, 3 May 2018 at the beginning of class

You should do this assignment individually or in pairs of two students. (Pairs can be from different sections.) You should only talk to the instructor, the TA or your partner about this assignment. You may also post questions and comments to the course discussion list on Piazza.

Please turn in your completed assignments on paper. Put your last name, first name, the assignment number in that order in the first line of your assignment.

Part I: MapReduce (50 points)

This problem concerns the compilation of quotations for entries in the original Oxford English Dictionary (OED). There are links posted in the Homework section that describe the process, but feel free to look for further information on the web or ask questions on Piazza.

The first two questions have to do with viewing the compilation process as a map-reduce computation.

(a) What is a map task in this scenario? Be sure to describe the what the input “chunk” was for a map task, and what intermediate items looked like, including key and payload.

(b) What is a reduce task in this scenario? Be sure to describe the input that a reduce task receives and what its output is.

The next three questions have to do with different kinds of failures in the system.

(c) What is the consequence of a map task failing (for example, a reader loses a book)? How hard is it to recover from such a failure?

(d) What happens if all the slips for a word disappear from their pigeonhole? How hard is it to recover from such a failure?

(e) What happens if an editor quits while writing the dictionary entry for a word? How hard is it to recover from such a failure?

Part II: Reconciling Versions (50 points)

This question concerns update reconciliation for a system such as DynamoDB, where multiple versions of an object can exist, and an application must be prepared to merge the versions together.

Consider a movie-download service along the lines of Netflix. Suppose it maintains a “viewing list” for each subscriber, which is an ordered sequence of movie ids, each marked “watched” or “not yet watched”. (All the watched movies precede any not-yet-watched movies in the sequence.) Assume there are two operations that update a subscriber's viewing list:

Downloader: This operation downloads the first unwatched movie on the list to a user’s device, then marks that movie “watched” in the viewing list.

Chooser: This operation lets a user add a new unwatched movie to his or her viewing list or change the order of unwatched movies on the list.

Note: *These operations might happen simultaneously. For example, two family members might be access the subscription account at the same time, or there is a download on one device while choosing on another device.*

The following items concern an operation Display that displays a user's viewing list to him or her. Consider the situation where Display gets a user's viewing list and the storage system returns two versions of the list, V1 and V2, that reflect multiple independent updates (so they are branch versions).

Note: *You should assume that, as in DynamoDB, merge is only called on versions that are not causally related (incomparable vector clocks). Thus it is not appropriate to simply choose the version with the later timestamp.*

(a) Describe a specific method to merge V1 and V2 to get a single viewing list to display. That is, what could be the “business logic” for the merge?

(b) Give three non-trivial examples of possible versions V1 and V2, along with the result of your merge function on them.

(c) What guarantees does your merge method give in terms of preserving changes made by Downloader and Chooser?

(d) Are there cases where the user might be surprised by the contents of the merged list?

(e) Suppose Chooser also allows a user to remove an unwatched movie from the viewing list. Does your merge function handle this case appropriately?