Patrick Trinkle

CMSC621 Fall 2009

Homework 3

### Exercise 7.10

Consider a personal mailbox for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency would be most appropriate?

With client-centric consistency, data items have an associated owner--the only process permitted to update the item.

* Monotonic Reads--When the user opens their mailbox it will always be the last view or a more update to date one.
* Monotonic Writes--When the user moves an email into a folder or creates a new email these writes/updates must be executed in order.
* Read Your Writes--Before an item X is read, all writes are performed on X. So if you sent an email on local copy at location A and then read your email at B you'd be guaranteed to see the changes.
* Writes Follow Reads--guarantees that when you write on an item x that your write goes to a value of x that is the same that you last read.

Any of these consistency models would be appropriate, however the most appropriate I feel would be read your writes. It guarantees that whenever you log into the check your mail your view is updated and also has all the updates the client (user) made previously.

### Exercise 7.11

Describe a simple implementation of read-your-writes consistency for displaying Web pages that have just been updated.

Assuming the browser caches the web pages it can hash them and query the web server if the page has been updated. The web page itself may not entirely be updated; pieces may be more likely to update more frequently and these pieces can be polled for updates specifically.

### Exercise 7.19

To implement totally ordered multicasting by means of a sequencer, one approach is to first forward an operation to the sequencer, which then assigns it a unique number and subsequently multicasts the operation. Mention two alternative approaches, and compare the three solutions.

One alternative approach to a sequencer is a quorum-based approach, whereby the client seeking to update the file contacts a majority of the replica servers and have the replica servers distribute the updates. This removes the need for multicasting the updates while maintaining versioning and sequence with updates.

Another alternative approach is using Lamport's logical clocks. However, this approach doesn't scale. (pg 311)