Coda
CS306L: Mike Dahlin
April 3, 2002

1 Preliminaries
1.1 Review
1.2 Outline
1.3 Preview

2Hoarding
• Fetch data to local on-disk cache in preparation for disconnection
• Hoard walk every 10 minutes
  • Refresh on invalidate?
    • Yes: improve availability
    • No: repeated updates common in Unix
      • Coda's decision: No invalidate data files; keep stale directory
• Manual specification of important files (+ caching of recently used files)
• Tweak replacement — don't drop a directory until all children dropped

3Admin
• Snoreen talk Thursday 11AM
• Project checkpoint Friday
  • I am travelling Monday is OK,

4 Emulation
• Log updates to disk
• Optimization: Keep only latest version of multiply-updated file
• Metadata updates transactional (via "recoverable virtual memory" = nice)

5 Reintegration
• Replay log to server
  • common case: no conflicts server checks all accesses and updates files
• Conflict
  • Suppose a file is updated at server and disconnected client
  • Can this be avoided?
    • Yes: pessimistic consistency, one side of partition holds lock on all data. No updates
during partition if you are on the side that doesn't hold the lock. Disadvantage: worse
availability.

Which copy to keep?
• Options
  • Newer update
  • Update by preferred user or machine
  • Update with higher value
  • Random
  • Dialog box: user chooses
• Merge (e.g., directory operations on different file; CVS updates to different parts of
  text file)
• Both (+ notify user of conflict)
• Run "reconciliation" program

What about subsequent updates?
• Commit them = no conflicts, right?
• Abort them — they may not make sense without the "missing" update

Coda's decisions
• This paper: merge directory updates; abort on other conflicting writes
• Later (I think): merge directory updates; "keep both + notify user" conflicting writes

Other systems
• Bayou: application-specific reconciliation code
• Oracle 14 options
• Win32: "If someone else made changes to the same network file that you updated
  offline, you are given a choice of keeping your version, keeping the one on the network,
or keeping both. To save both versions of the file, give your version a different file
  name, and both files will appear in both locations."

  • Doesn't work for me...

  • ... Notice: write/read conflicts not detected
  • They worry about committing writes after a write/write conflict ("what if the later
    write depends on the earlier one?")
  • What about committing writes after a write/read conflict (later writes might only make
    sense in light of old value of data, e.g., I might read "price(X) = $1", and then write
    "order 10 X items")
• Coda semantics
  Suppose no write-write conflicts occur
  • Does Coda provide sequential consistency?
  • Does Coda provide FIFO consistency?
  • (Recall) FIFO consistency (aka PRAM consistency) write done by a single process
    are seen by all other processes in the order in which they were issued, but writes from
    different processes may be seen in different orders by different processes
  • Does Coda provide causal consistency?
  • Does Coda provide strict coherence?
  Notice: Coda does not worry about write read conflicts.

6 Today
• Coda available in Linux
• Win2K FS has most of features listed above