

A Coherent Distributed File Cache with Directory Write-Behind
Mann, Birrell, Hisgen, Jerian, Swart
1993, SRC TR and ToCS

why are we reading this paper?

- it talks about consistency for a distributed system
- it focuses on crash-recovery semantics of use to apps
- it explores an alternate approach to logging
- it provides an efficient way to get those semantics
 - better than the usual sync writes approach
 - better than logging, since you don't have to force the whole log
- tokens are related to the locks in Labs 4 and 5

context

- lots of workstations sharing a file server
- each workstation has a cache
- for performance, performs writes into local cache,
 - send to server disk later
 - this is "write-behind"
- more aggressive than NFS's sync rename()&c and close-to-open
- trying to support sophisticated distributed apps (e.g. Vesta vcs)

what are the potential problems?

- if cache writes back only in response to server reclaiming tokens?
- vi does create(temp), write(), rename()
 - if I crash, might others see an empty file?
- I write foo.c, cc -o foo.o foo.c
 - if I crash, might others see new .o but old .c?

what are some simple designs?

- (writes only for token reclaim)
- sync updates (no write caching)
- close-to-open
- keep a log, flush prefix on token reclaim

what's their approach?

- client remembers order of update ops to each object
 - some ops affect more than one object
- writes all "prior" ops back before returning token

how does Echo's ordering help solve problem examples?

what does the implementation look like?

- do they keep pointer among block buffers in the client cache?
 - cycle example: write B1, B2, B1
 - or two renames between directories, opposite orders
- no: each vnode has list of ops w/ full arguments (?)
- each op refers to other affected files
- must flush some of those other files' ops when returning a token

what happens when a client workstation crashes?

- do apps on other workstations need to take action?
- i.e. does the system need to notify apps on other workstations?

what happens when a client loses its network connection?

- and it has holds write tokens & has ops in its write-back cache
- can server reclaim its tokens if another workstation wants them?

Cite as: Robert Morris, course materials for 6.824 Distributed Computer Systems Engineering, Spring 2006. MIT OpenCourseWare (<http://ocw.mit.edu/>), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

what if the unreachable client then reconnects?
will two workstations think they both have tokens?
what is reporting lost write-behind all about?