

miscellaneous

- bugs in old lockd
 - unexpected packet could crash old lockd
- code reuse for NLM4
 - clone-and-tweak versus function pointer Hell

protocol holes

- GRANTED RPC
 - retransmission versus new call that looks like old one
- network partitions
 - is it dead, Jim? What (when) do I tell the user?
- F_GETLK sysid
 - hard to map protocol info to UNIX info
 - server's idea of sysid not same as client's?

performance and benchmarking

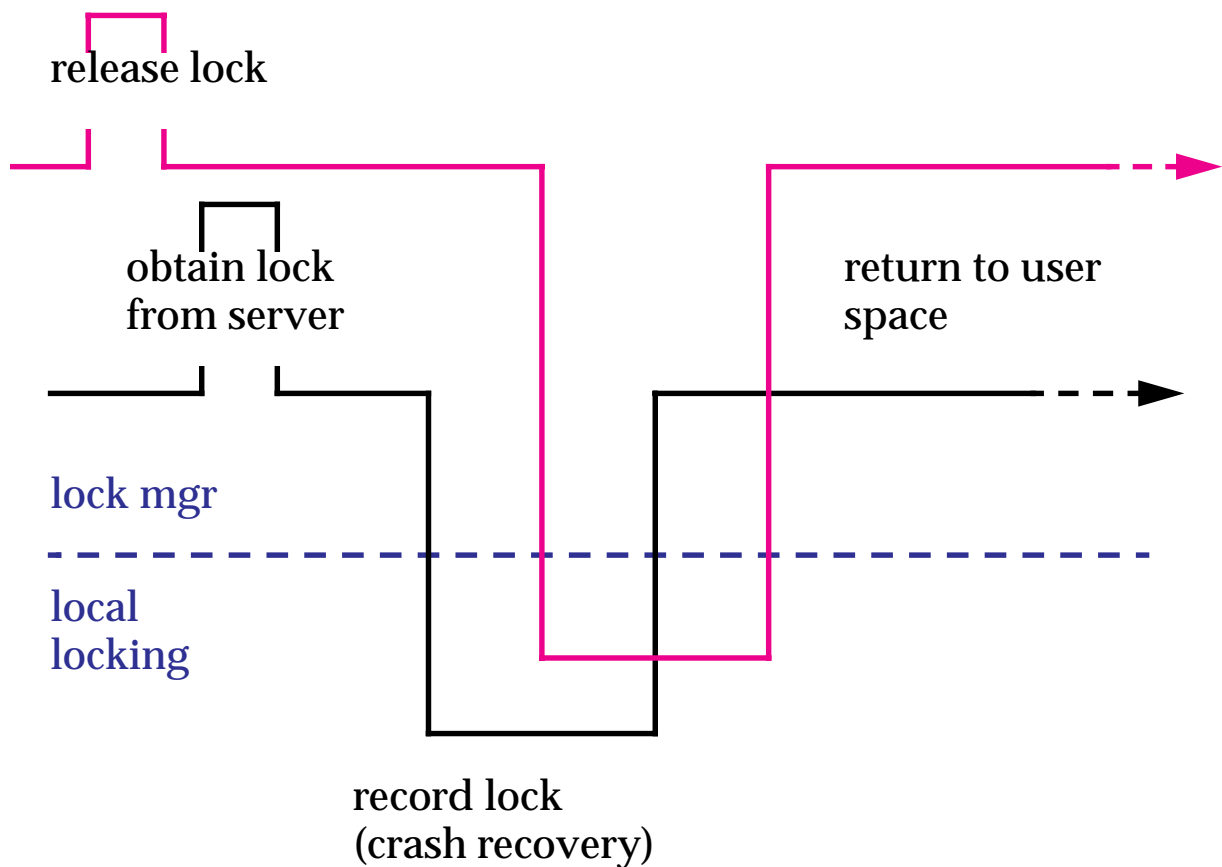
- faster locking, but potentially slower interactive response
 - more physical memory consumed
- hard to start lock manager on demand
- demand-load server-only NFS code
- need to be careful benchmarking

MT correctness (cont'd.)

- example: server code for blocking locks
- example: client waiting for GRANTED call
- tools, stress tests helpful

MT correctness

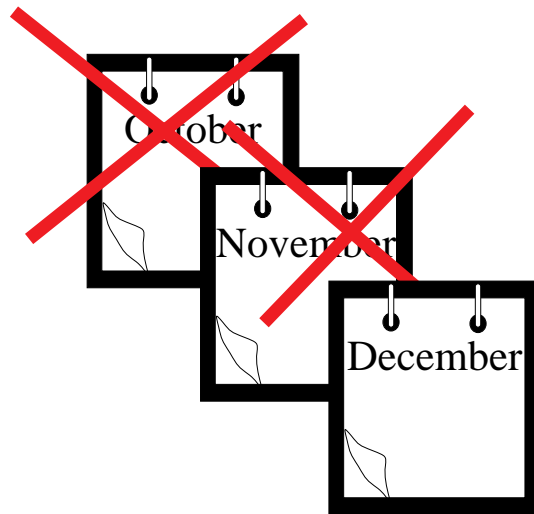
- example: registration of lock on client



features not required by NFS

- KRPC changes
 - service threads can block indefinitely
- portmapper support
 - understanding requirements
 - no convenient libraries

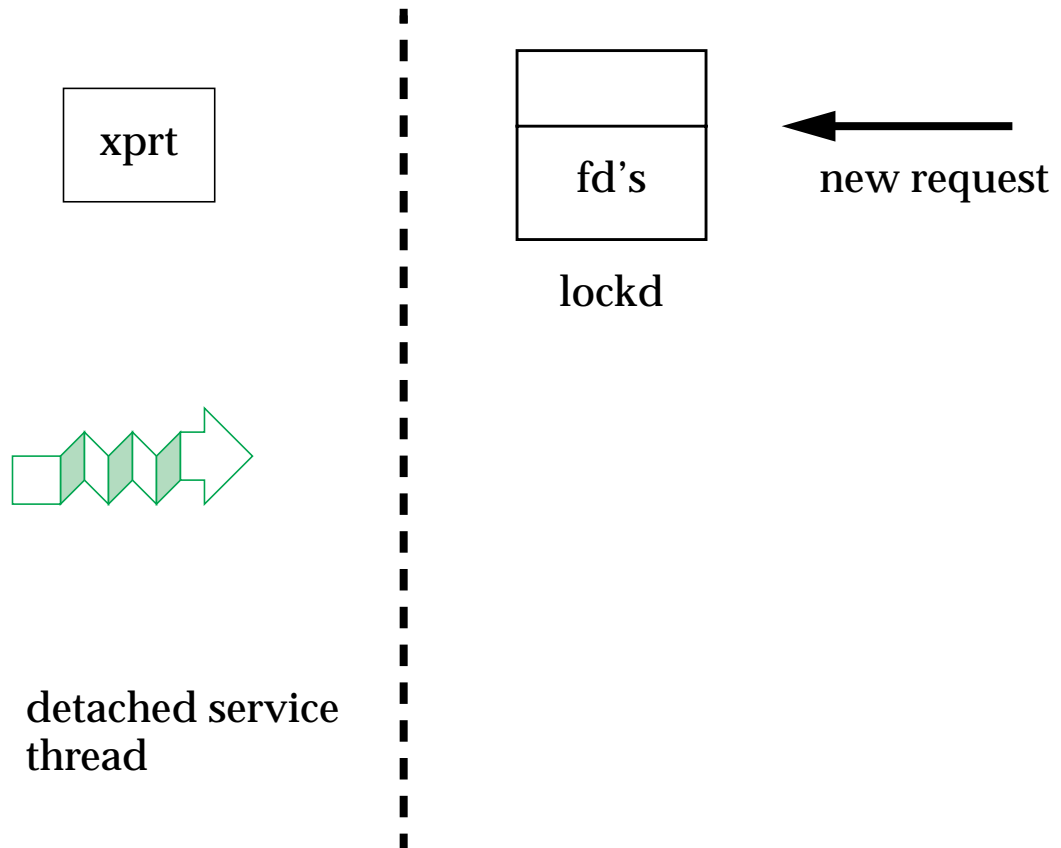
Learning experiences



KRPC changes (cont'd.)

- reserved threads
 - will detach if blocked
 - need to check for thread ceiling before local locking call

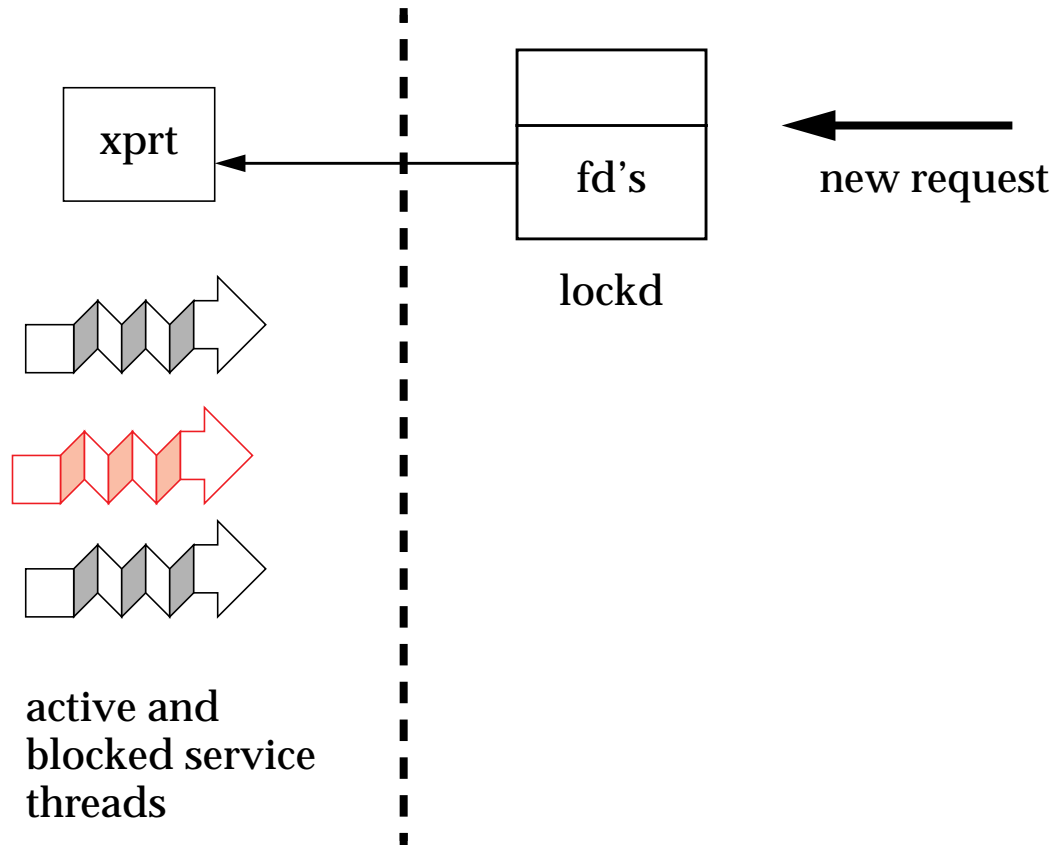
KRPC changes (cont'd.)



- promise not to “use” transport
- ensure there is thread for new requests
- clean up when done

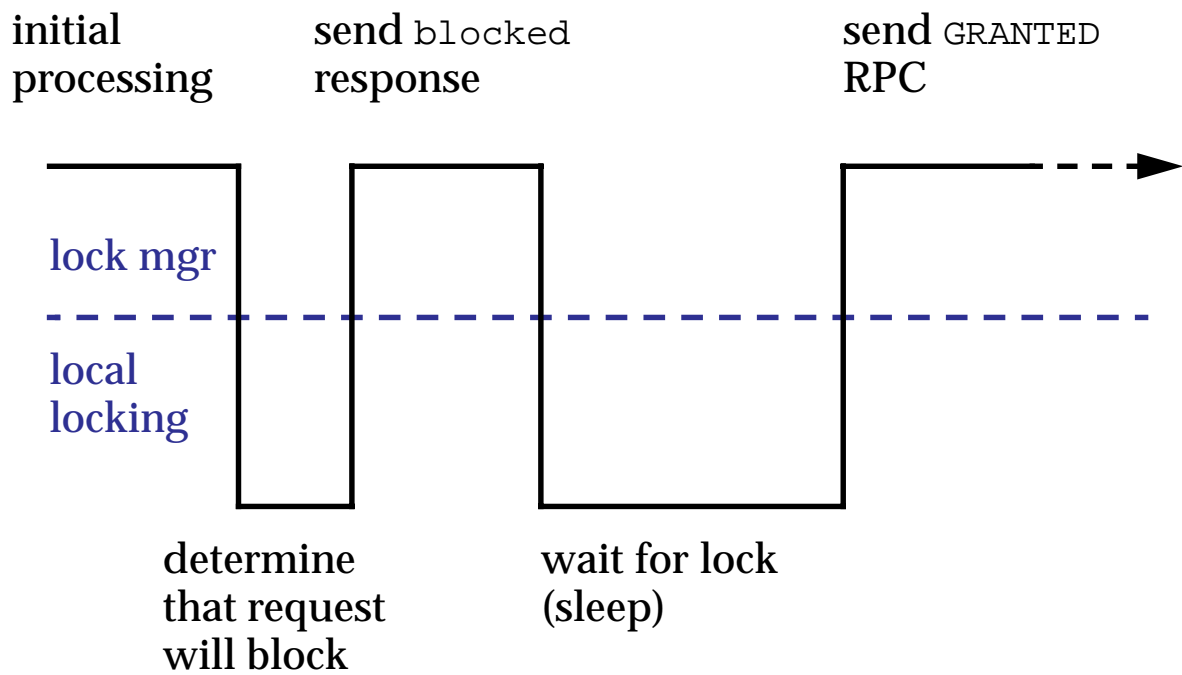
KRPC changes

- detached & reserved threads



local locking callback

- send blocked response without race



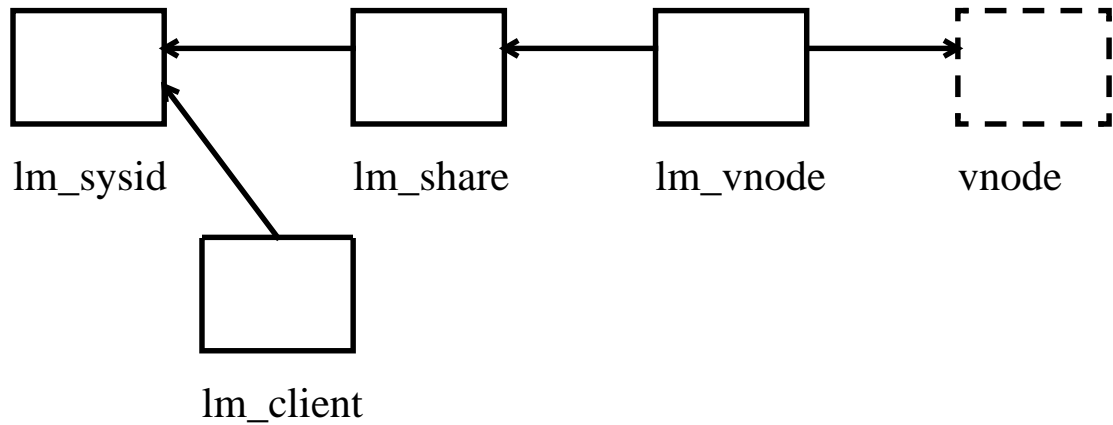
local locking changes

- remove hooks for old lockd
(GRANT_LOCK_FLAG, l_XXX, etc.)
- process blocks for blocked lock
- callback for blocking request
- query routines
- shutdown/restart
- support for unsigned ranges
- cosmetic improvements

data structures (cont'd.)

- `lm_sysid`: per-{host, netconfig} address, status
- `lm_client`: RPC client handle
- `lm_vnode`: vnode reference, filehandle map
- `lm_share`: PC file sharing
- `lm_async`: message-style call/response
- `lm_sleep`: wait for GRANTED call
- `lm_xprt`: thread/transport map
- `lm_config`: transport/netconfig map

data structures



lm_async



lm_xprt



lm_config

server



client



lm_sysid

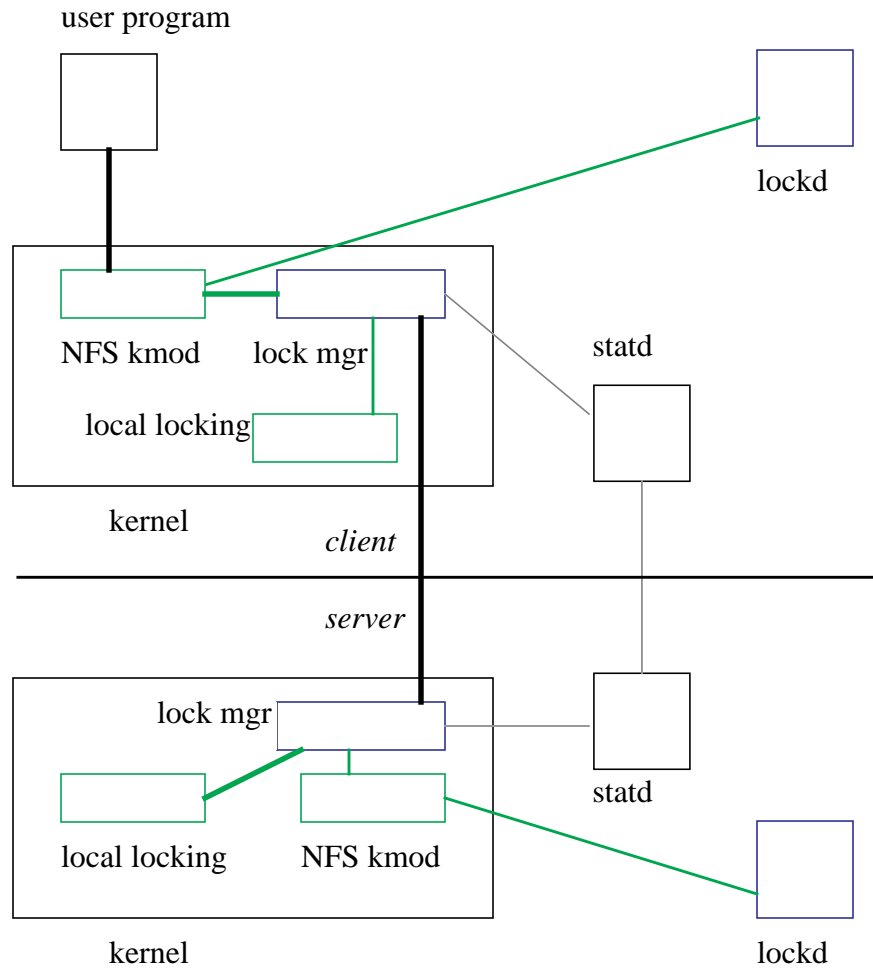


lm_client



lm_sleep

How it works



Why you should care

- cleaner implementation, many bugs fixed
- avoids “message-style” RPCs
 - easier cross-domain locking
 - avoid meltdown from name service or portmapper error
- supports 32-bit offsets, lengths over the wire (PCs)
- faster

Overview

- Why you should care
- How it works
- Learning experiences

The New NFS Lock Manager

Mike Kupfer

kupfer@Eng.Sun.COM