TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute for System Architecture, Operating Systems Group

Speculative execution in a distributed file system

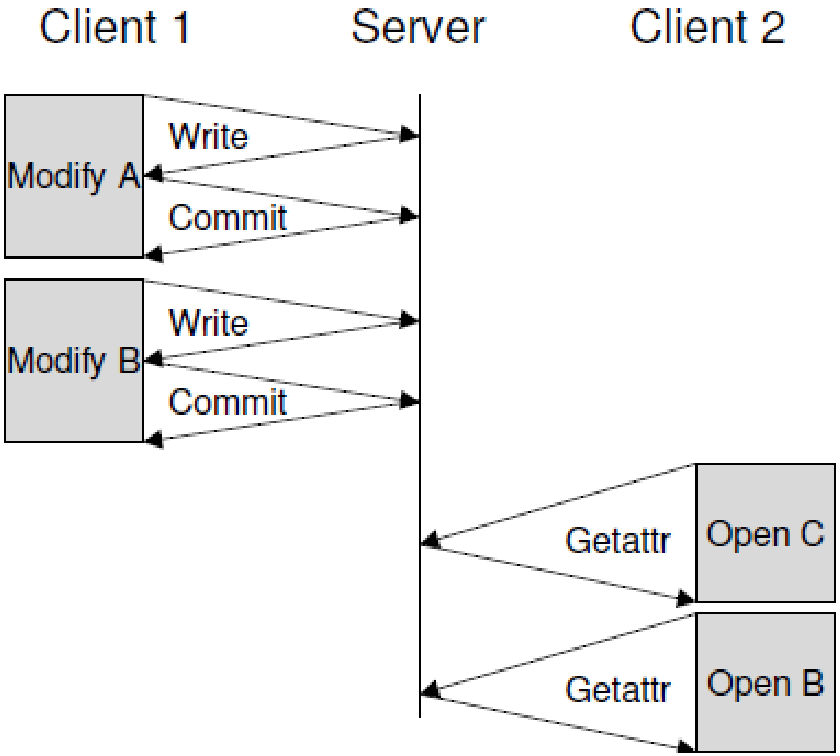
E. B. Nightingale, P. M. Chen, J. Flinn

Dresden, 2010-09-01

Distributed File Systems

- NFS (v3), AFS, Coda
- Carefully crafted protocols
  - need to handle concurrent accesses
  - synchronous → low performance
  - optimization: weaker consistency

Synchronous NFS



(a) Unmodified NFS

The case for speculation

- Concurrent access is an exception.
- FS client can normally predict the outcome of an operation.
  - Caches
- Cheap checkpointing/restart mechanisms
  - Often faster than network roundtrips
- Abundant resources
  - Can spend memory on checkpoints and cycles on bookkeeping.

Speculation interface for Linux

- Needs:
  - Prevent state externalization
  - Cheap checkpoint/restart mechanism
  - Track speculation dependencies across processes

- Spread function calls across the kernel
  - `create_speculation()`
  - `commit_speculation()`
  - `fail_speculation()`

- ~ 7.500 LoC

Checkpoint & restore

- Checkpoint
  - `fork()`
  - plus additional state (pending signals, locks, timers, ...)
  - don't make child runnable

- Restart
  - Force parent to exit silently
  - Modify forked child to look like parent at time of checkpoint (adapt PID, FDs, signal state, …)
  - Run child

Speculation

- Upon speculative system call
  - Create speculation data structure
    - track objects depending on this speculation
    - used later for process deps
  - Create undo log
    - for rollback on failure

- Optimization: use one log for a sequence of speculations
  - Rollback cost vs. bookkeeping cost

Preventing state externalization

- Goal: no one sees speculative state before it is committed.
  - Apart from speculative processes.
- Always block a process that tries to access speculative state.
  - Must do for non-speculative processes.
  - Can do better for speculative ones.

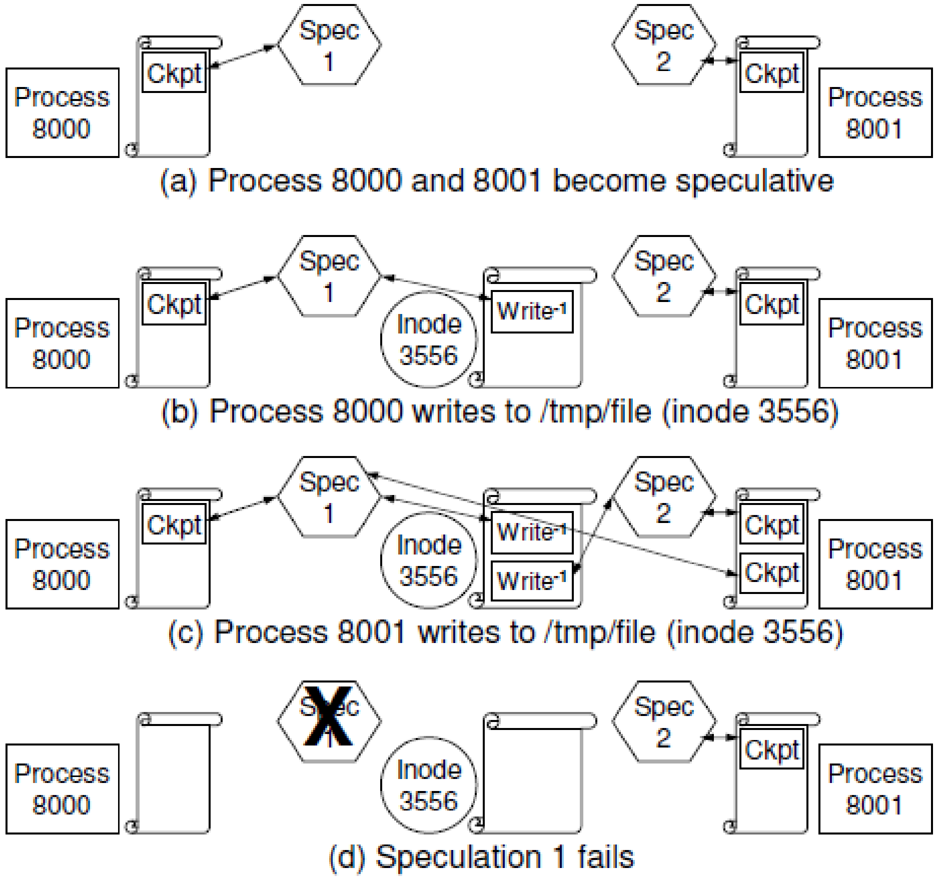
Avoiding blocking

- Allow
  - syscalls that don't modify state – `getpid`
  - syscalls that only modify process-local state – `dup2`
- Speculative operations on file systems
  - SPEC flag set upon open()
    - If set, try to speculate from cached data
    - Else, block
- Buffer I/O that would otherwise become visible, e.g. output to a TTY.

Tracking process dependencies

- Track propagation of speculative state through
  - pipes/FIFOs
    - log r/w operations
    - reader becomes speculative, too
  - sockets
    - buffer until committed
  - signals
    - make recipient speculative
    - might currently be in non-spec syscall
    - queue signal and deliver upon syscall return
    - deliver some signals immediately

Depending speculations



(a) Process 8000 and 8001 become speculative

(b) Process 8000 writes to /tmp/file (inode 3556)

(c) Process 8001 writes to /tmp/file (inode 3556)

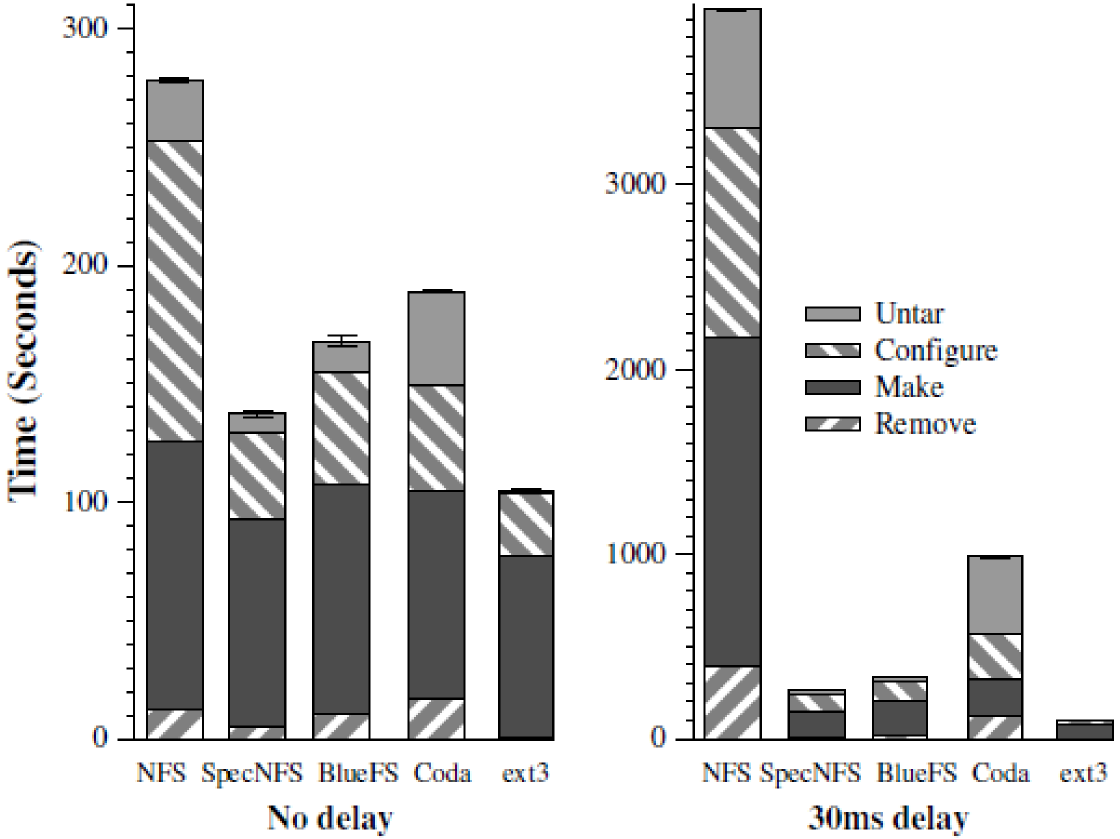
(d) Speculation 1 fails

Figure 3: Propagating causal dependencies

Speculative distributed FS

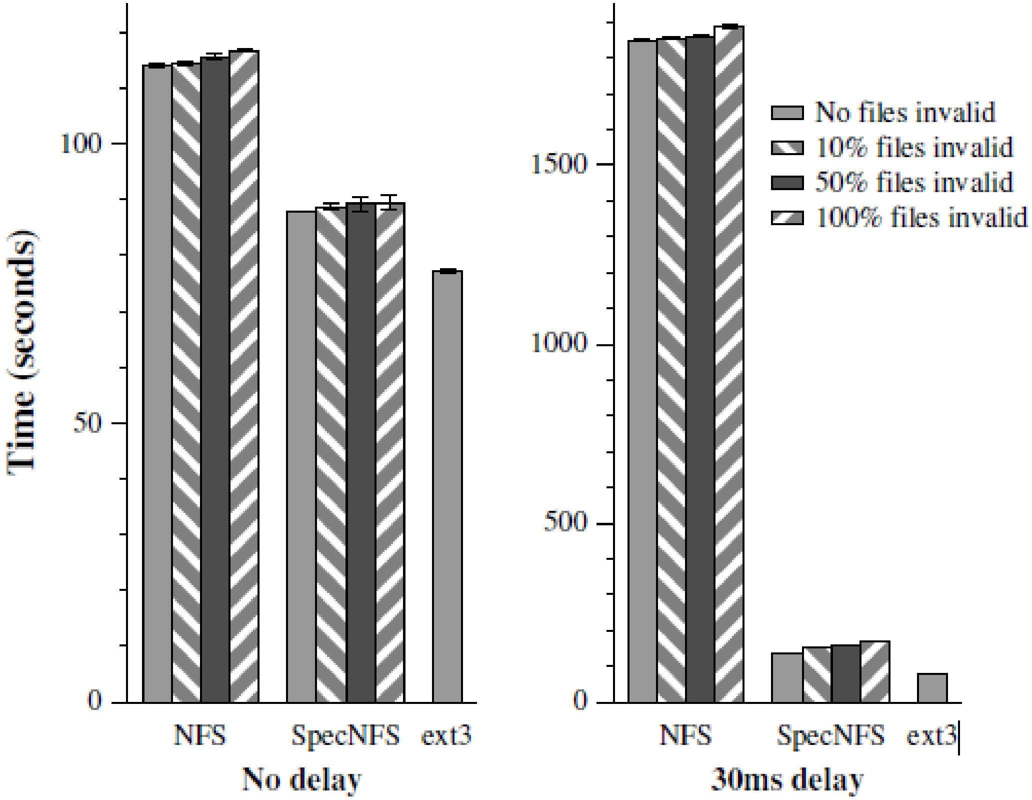
- Adapt server:
  - speculative calls include hypothesis
  - counter-check hypothesis before carrying out actions
  - keep speculation log at server
- Speculative group commits
- Implemented 2 FS: SpecNFS + BlueFS

Evaluation



This figure shows the time to untar, configure, make and remove the Apache 2.0.28 source tree. Each value is the mean of 5 trials—the error bars are 90% confidence intervals. Note that the scale of the y-axis differs between the two graphs.

Rollback cost



This figure shows the time to make Apache 2.0.28 with different percentages of files out-of-date in the client cache. Each value is the mean of 5 trials—the error bars are 90% confidence intervals. Note that the scale of the y-axis differs between the two graphs.

Discussion

- Review speculation in the context of
  - power → sync. protocols allow for turning off resources while waiting
  - 10GB ethernet
  - crashes
  - massively parallel applications
    - no IPC or shared memory support
    - Chen, Flinn @ ASPLOS 2010: *"Respec: Efficient Online Multiprocessor Replay via Speculation and External Determinism"*

- Group commit at server side w/o speculation?