File system trace papers

- **The Zebra striped network file system. Hartman, J. H. and Ousterhout, J. K. SOSP '93. (ACM Digital Library)**

Technologies

- LFS to create logs, stripe across multiple storage servers
- RAID: various combinations of striping across multiple disks and redundant data
- Components:
  - Clients: create stripes and store data directly (in parallel) to storage servers
  - Storage servers: save segment fragments
    - Fragments: large block of data + identifier (client id+ per client sequence number + offset)

- Storage server options
  - Store a fragment (synchronous)
  - Append to a fragment (atomic)
  - Retrieve a fragment
  - Delete a fragment: invoked by stripe cleaner
  - Identify fragments: crash recovery

- File manager: stores and manage file metadata
  - Name lookup and cache consistency
- Stripe cleaner: log cleaner. Reclaim space from deleted or overwritten data

System operation

- Communication via deltas
  - Fragments are never overwritten (except for parity)
  - Delta tell clients, cleaner and file manager what changed
    - Stored in client logs
- Writing files: Distribute writes to storage servers. Client computes parity, loss of partial parity okay because client has parity
- Reading files: Storage manager for cache consistency. Fetch block pointers and then request blocks from storage managers
-

- Stripe cleaning: use deltas to compute liveness of segments (stripe status files)
- File access/cleaning conflicts: optimistic approach. Allow cleaner to issue cleaner delta. On update conflicts from users, file manager can create reject deltas for cleaner. Race condition for clients requesting open and read data.

- Crash
  - internal stripe consistency - partial fragment, use checksums. Missing fragments @storage servers: use parity to recover
  - stripes vs. metadata - file manager checkpoints metadata. After file manager crash, reprocess all deltas. Version numbers allows for ordering of deltas across all client logs.
  - stripes vs. cleaner – checkpoint state. Not as catastrophic
- Performance: 4-5x for large files. Small files 20%-3x
- Related: Server based striping (TickerTAIP)