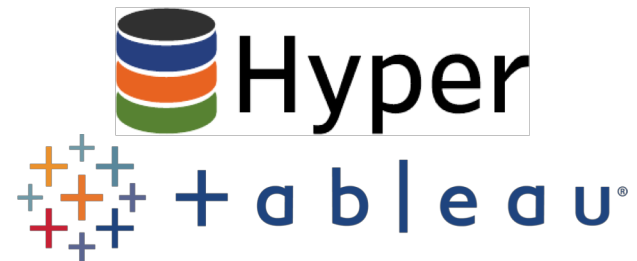


Modern techniques for transaction-oriented database recovery

Caetano Sauer





My pleas

1. Demand on-demand recovery

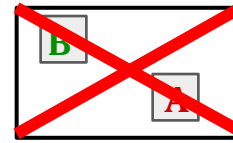
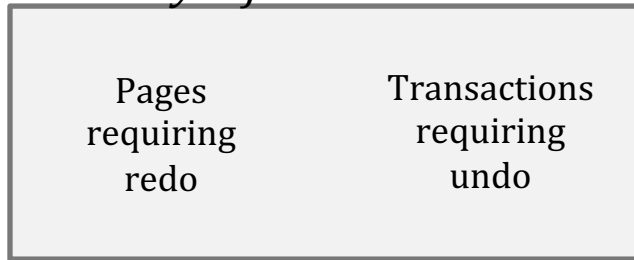
2. Make the log great again

Demand on-demand recovery

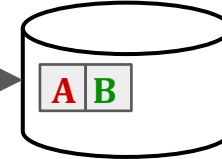
ARIES



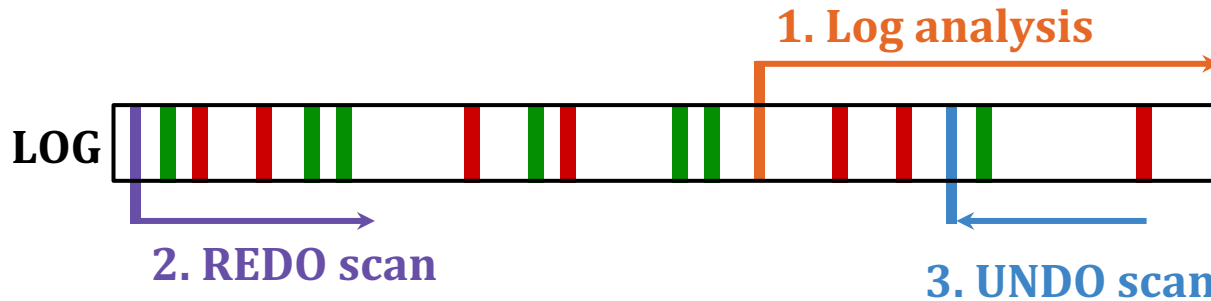
Recovery info



Buffer pool



Database

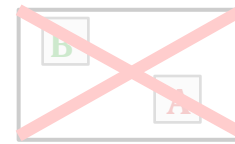
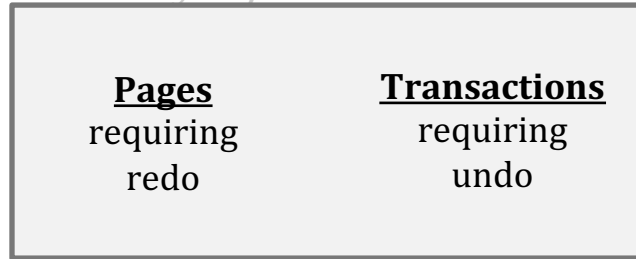


system mostly unavailable during three phases

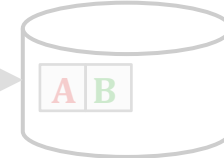
ARIES restart



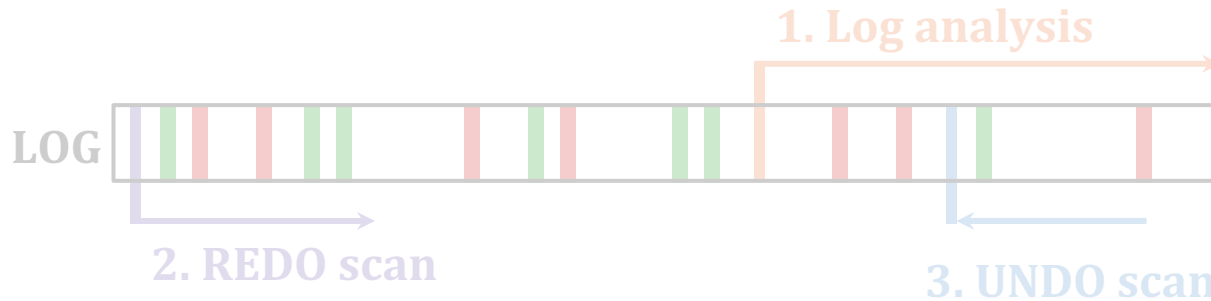
Recovery info



Buffer pool

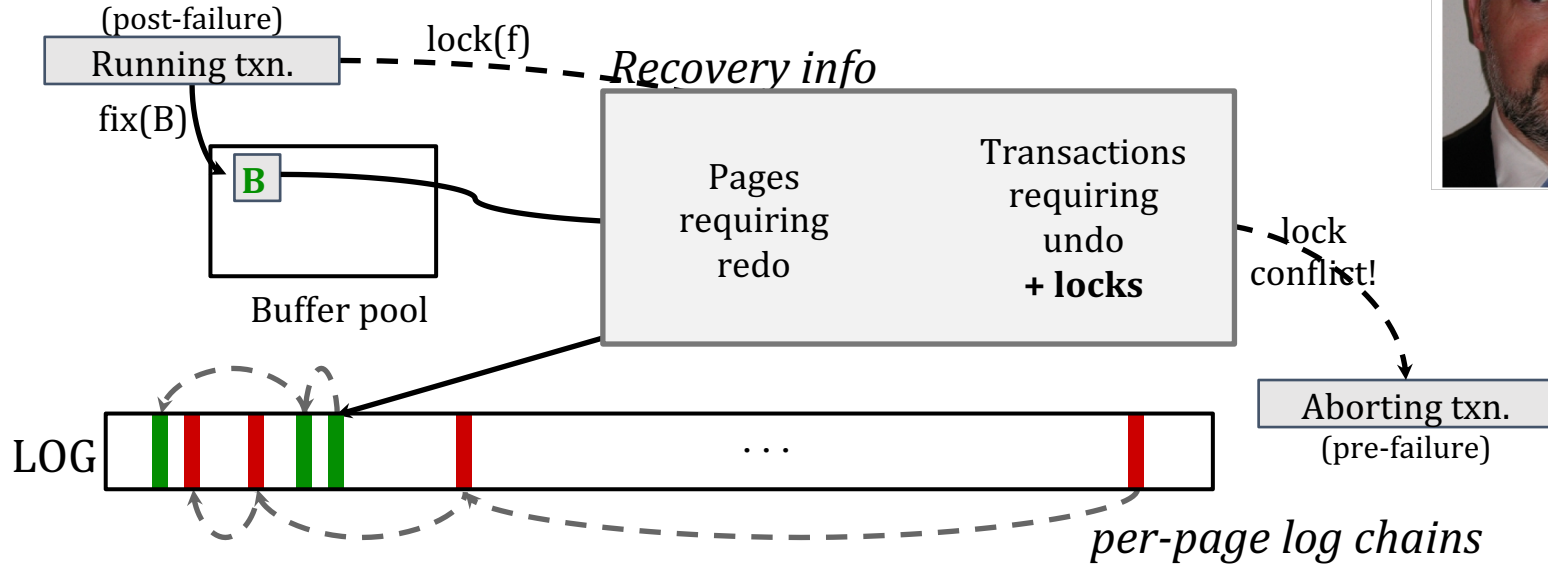
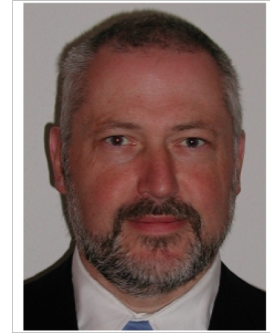


Database



system mostly unavailable during three phases

Instant recovery



redo and undo on demand, without waiting for log scans

same recovery actions, different schedule

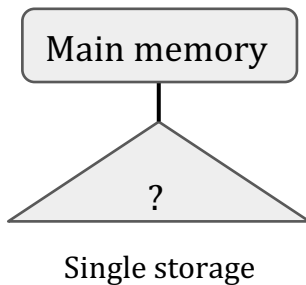
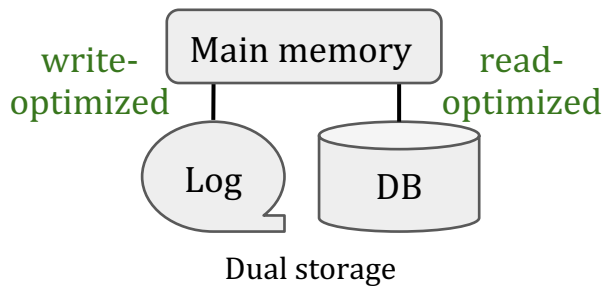
Failure classes

Failure class	Loss	Typical cause	Response
Transaction	Single-transaction progress	Deadlock, constraint violation	Rollback
System	Server process (in-memory state)	Software fault, power loss	Restart
Media	Persistent database contents	Hardware fault	Restore
Single page	Local integrity	Partial writes, wear-out	Repair

Make the log great again



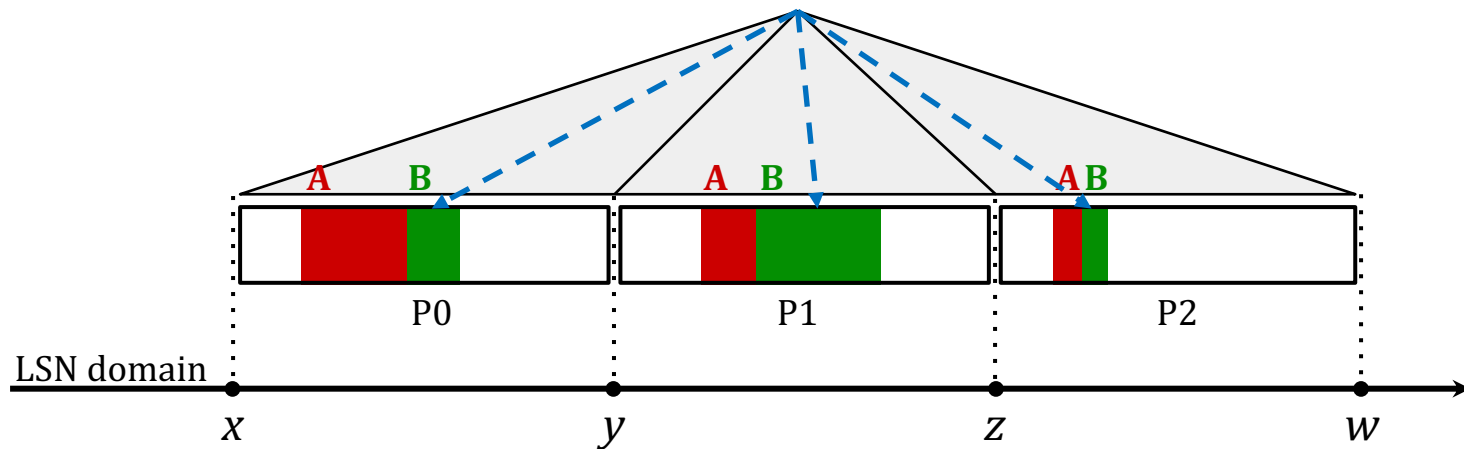
Throw away the database!



"... a DBMS is really two DBMSs, one managing the database as we know it and a second one managing the log."

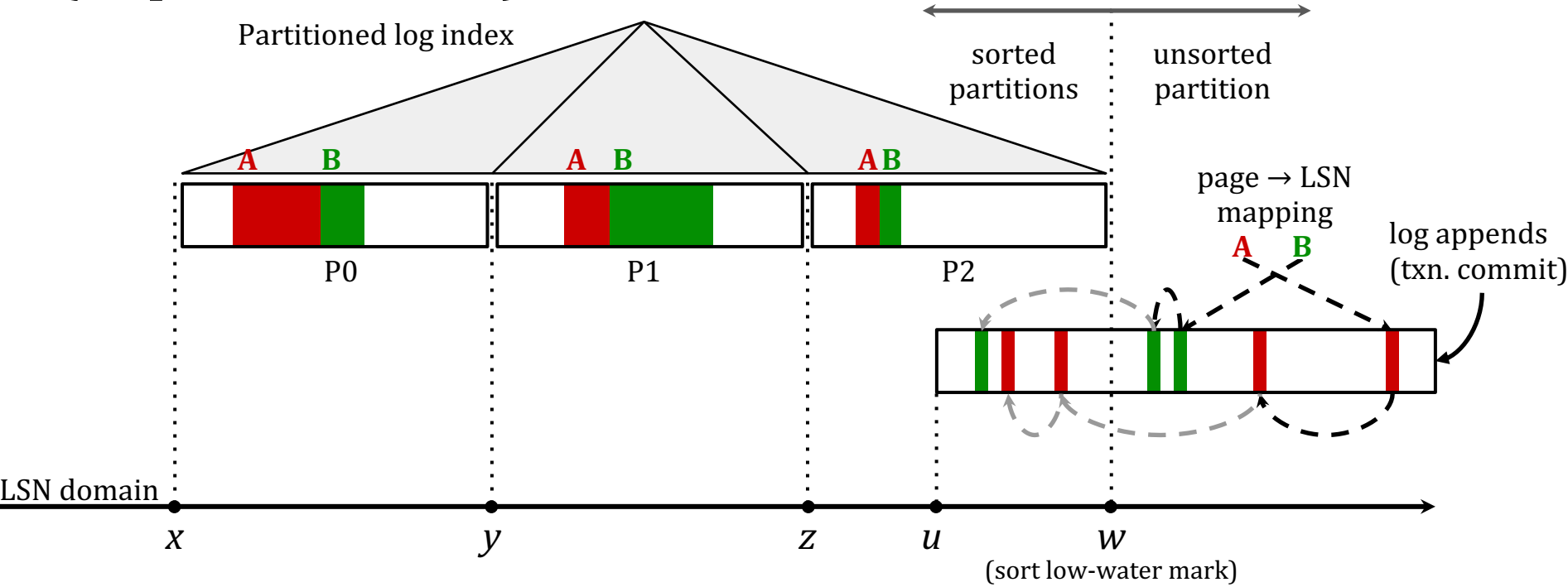
Michael Stonebraker

The log as a partitioned B-tree



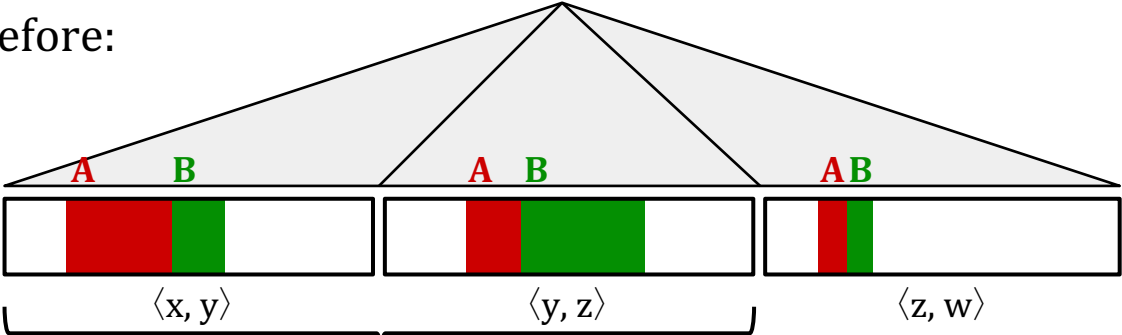
like an LSM, but with page identifiers and their log records

The log as a partitioned B-tree (implementation)



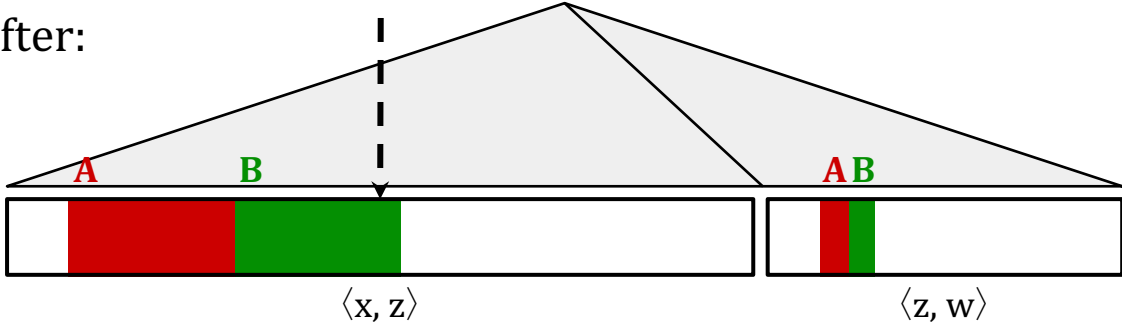
Merging

Before:



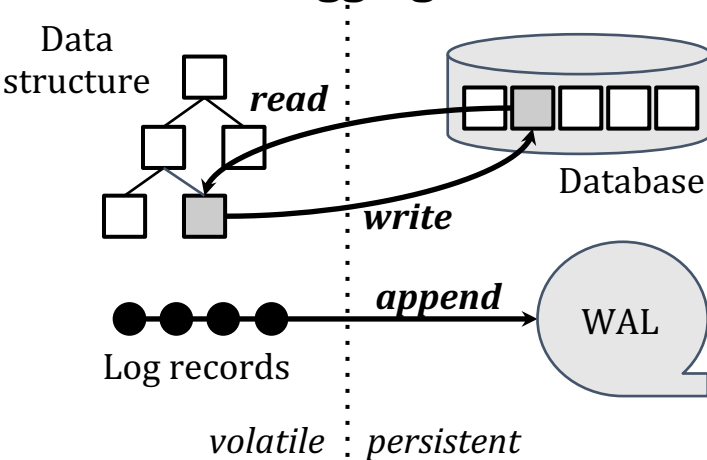
merge

After:

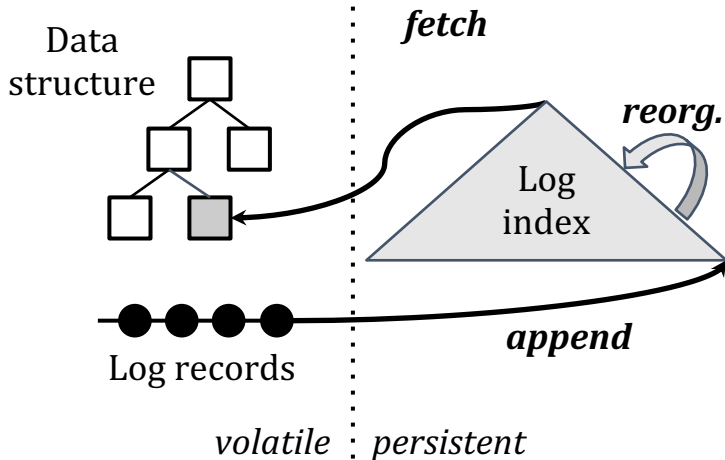


FineLine

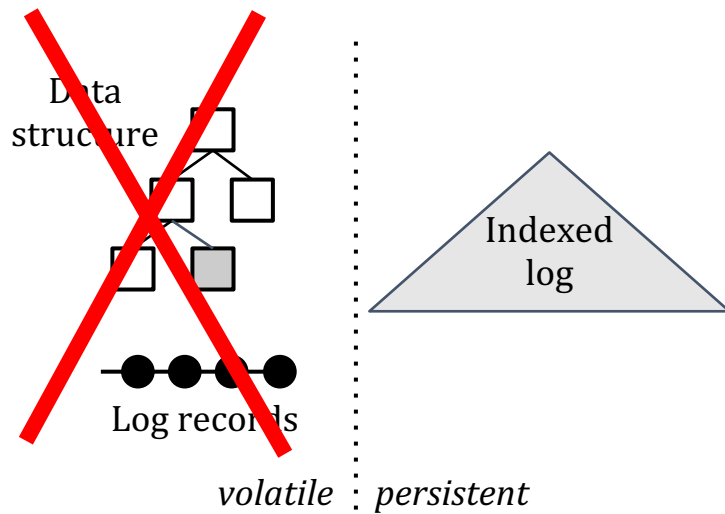
Write-ahead logging:



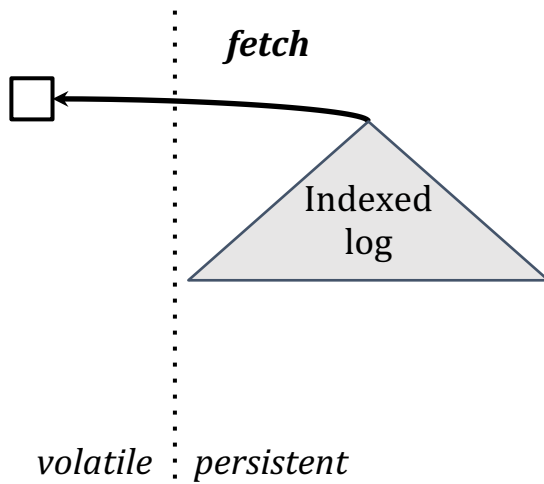
FineLine:



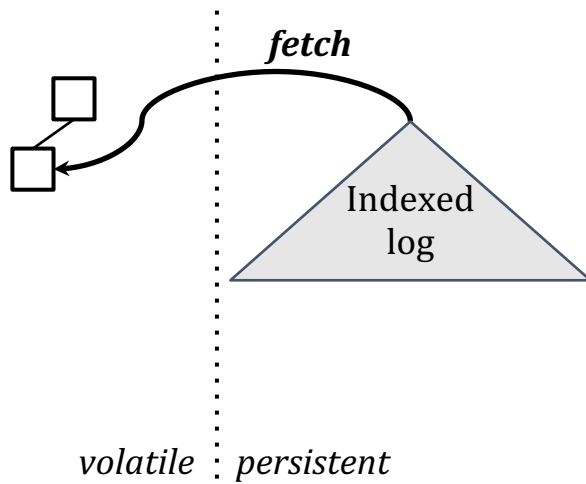
FineLine recovery



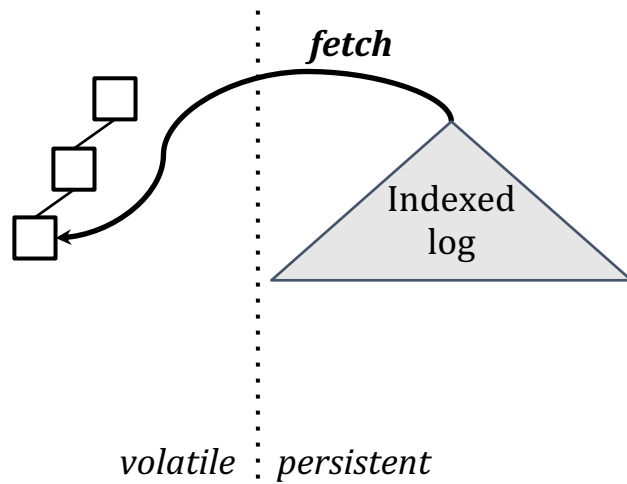
FineLine recovery



FineLine recovery

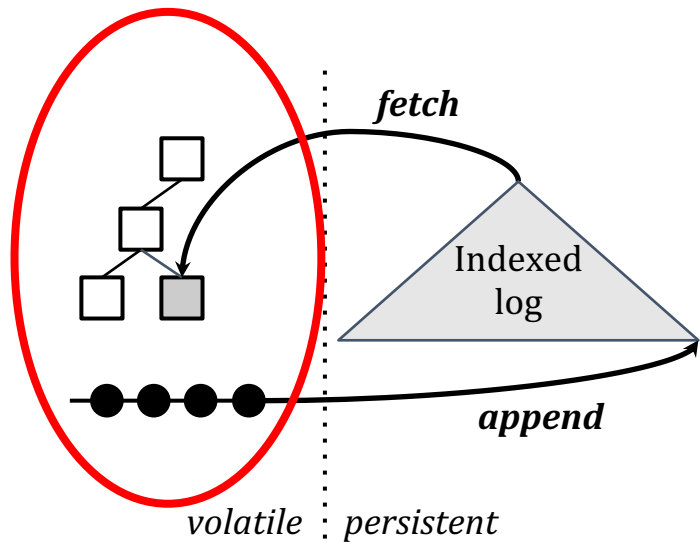


FineLine recovery

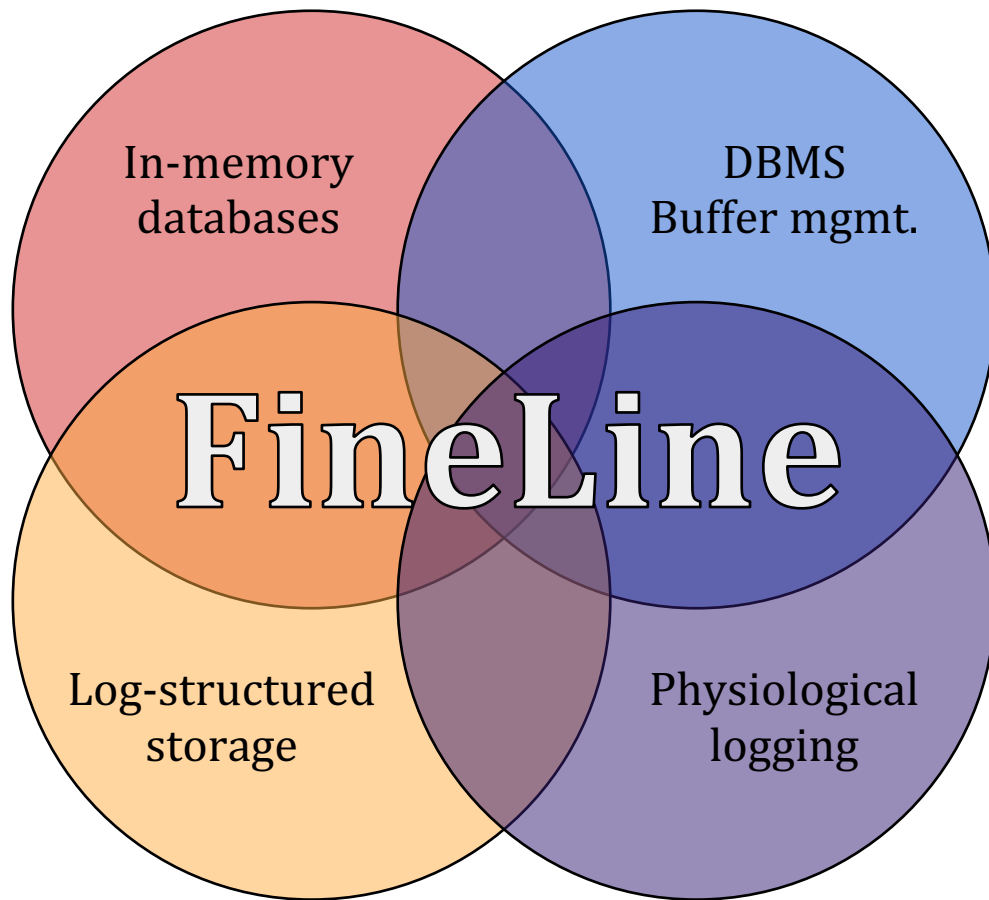


FineLine

In-memory database



- nodes recovered automatically during fetch
- volatile structures = in-memory database
- no undo, no dirty pages, no checkpoints, no offline log scans



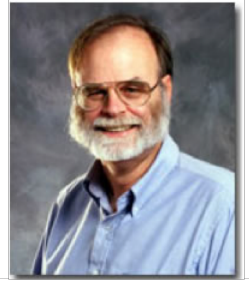
My pleas

1. Demand on-demand recovery

2. Make the log great again

Standing on the shoulders of Giants





5.8.6.2. How to Get Perfectly Reliable Data Management

watch this space (a coming attraction)

Thank you!
csauer@tableau.com