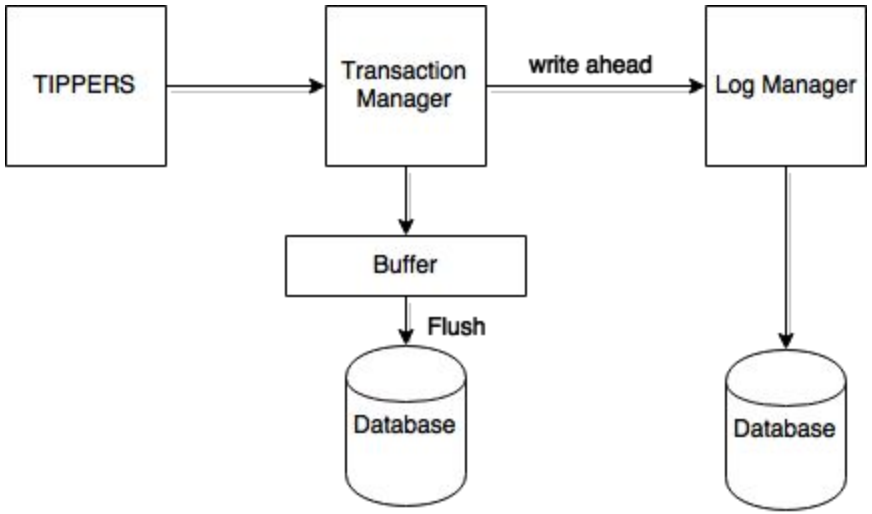
Transaction Manager for Policies

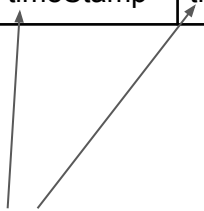
Team Deadlock
Chaitanya Kshirsagar, Nishad Gurav, Yadhu Prakash

Architecture



Policies Table

attributes	policyID	author	querier	from	to	entered	invalidated*
types	int	string	string	timeStamp	timeStamp	timeStamp	timeStamp



Time traversal and Recovery
based on these timestamps

Design Choices

- Use timestamp with policy versions.
  - Time traversal is much faster
  - Recovery is simpler
  - Cons: needs more storage

- Force Log @ Commit (FL@C)
  - Commit is not complete till all logs are flushed

- Periodic Flush with checkpoints
  - Logs and policies in buffer are flushed when capacity is reached. Capacity is configurable.
  - Checkpoint logged when flush is successful. Use checkpoints for redo.

- Using JSON to exchange data with TIPPERS for universality

Recovery Management

- If a TRXN is interrupted midway we could have the following cases.
  - Checkpoint log exists but no commit log before that -> UNDO
    - Log and policy table in sync, but trxn failed
  - Commit exits but no checkpoint -> REDO
    - TRXN failed during write to policy table.
  - Abort log exits without checkpoint -> UNDO
    - TRXN failed during write to policy table.
  - Random policy data logged last -> UNDO (till latest commit log)
    - TRXN failed during write to log table.
    - We still undo from policy table in case the TRXN was checkpointed earlier.