Fast Logging and Recovery Support for Transactional Databases

Youhui Bai, Cheng Li, Yinlong Xu
School of Computer Science and Technology, University of Science and Technology of China
youhuibai0108@gmail.com, chengli7@ustc.edu.cn, ylxu@ustc.edu.cn

1. Background

- Failures are common → inconsistency → Money loss
  - Process crashes, kernel panics, power outage…

Delta Cancels 280 Flights Due to IT Outage

Says essential IT systems went down for hours

Travelers wait in line at the Delta check-in counter at LaGuardia Airport , August 8, 2016 in the Queens borough of New York City. Delta flights around the globe were

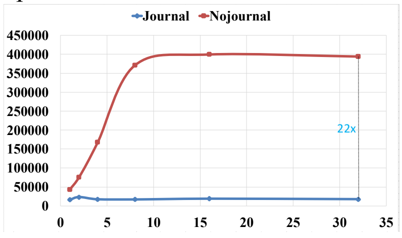
[source: http://www.datacenterknowledge.com]

  - Fast recovery is needed but often underestimated.
- Write-ahead Log (WAL) [Mohan, TODS'92]
  - Most transactional systems leverage a single WAL for data durability and consistency in presence of failures
    - Forward process: write to log first, then real place
    - Recovery: replay winners, rollback losers

Update
Log it
Data
Really change it

2. State-of-the-art solutions

- WAL is a performance bottleneck, due to the mismatch between a single centralized log and the increasing CPU capacity.
  - Threads contend for the log head while logging.
  - A sequential scan will be performed when recovering state despite of massive parallelism.



Fio benchmark on ext4 file system

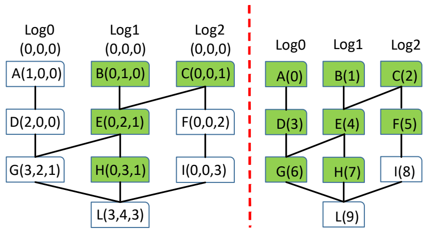
- Distributed log [Johnson, VLDB'12] didn't close the gap.
  - Too many dependencies between entries spanning multiple logs
    - cache line transfers due to thread contentions
    - cross-log synchronizations are needed during recovery

3. Our proposal

Challenge: significantly eliminating *dependencies* between log entries across multiple logs

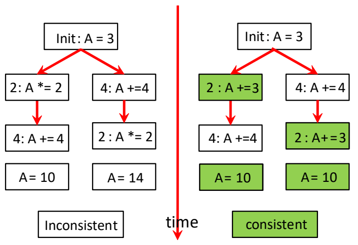
Key techniques:

- Using Vector Clock [Mattern, PDA'89] to maintain a partial order instead of a sequential order



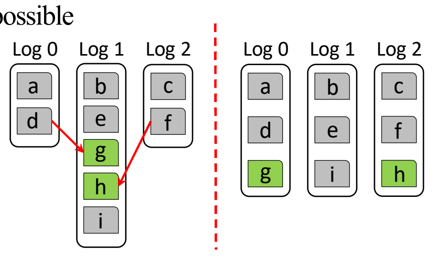
Vector clock vs Lamport clock

- Leveraging operation-level logging and conflicting-free replicated data types [Shapiro, SSS'11] to further reduce dependencies



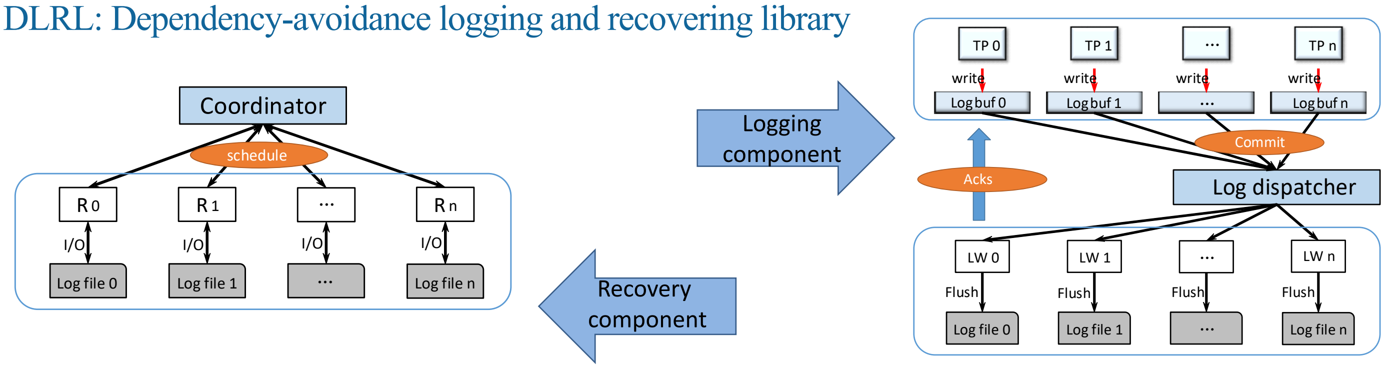
Conflict-free Replicated Data Types

- Optimizing log entries placement for striking a balance between reducing cross-log dependencies and uniformly distributing loads to logs as possible



Log entries placement

4. DLRL: Dependency-avoidance logging and recovering library



Each recovery worker processes as independently as possible while interacting with coordinator when the cross-log synchronization is needed.

The Log dispatcher builds a dependency graph of receiving log entries and makes decisions on their placement plans.

中国科学技术大学
University of Science and Technology of China