OLTP ON NON-VOLATILE MEMORY:
WRITE-BEHIND LOGGING

JOY ARULRAJ

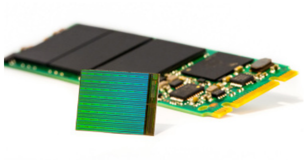
CARNEGIE MELLON UNIVERSITY

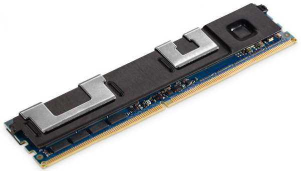
@HPTS 2017

CARNEGIE MELLON DATABASE GROUP

ISTC BIG DATA

NON-VOLATILE MEMORY (NVM)





DRAM

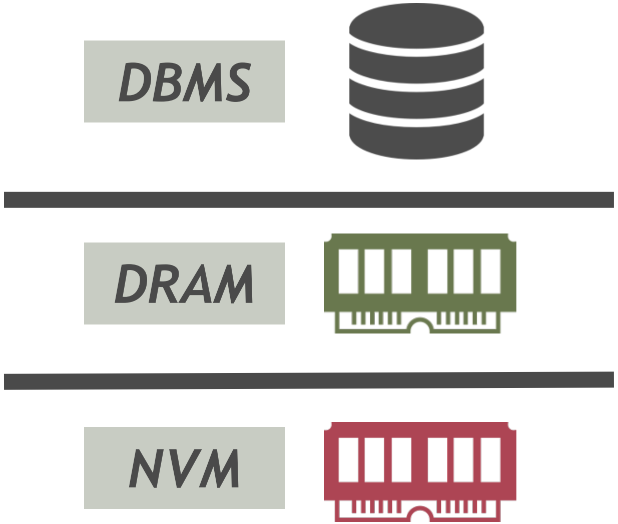
NVM

SSD

Like DRAM, low latency loads and stores

Like SSD, persistent writes and high capacity

CURRENT DATABASE SYSTEMS



DBMS

DRAM

NVM

Perform write-ahead logging to avoid random writes to durable storage

But, NVM supports fast random writes

TALK OVERVIEW

A new logging algorithm that is better suited for non-volatile memory

Inspired by research done at Wisconsin and Berkeley in the 1990s

THE UNIVERSITY of WISCONSIN MADISON

Berkeley UNIVERSITY OF CALIFORNIA

TALK OVERVIEW

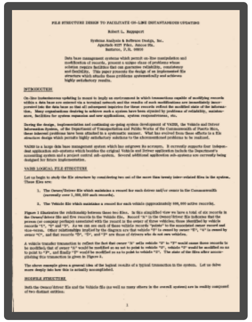
A new logging algorithm that is better suited for non-volatile memory

And research done at Puerto Rican DOT(!) and IBM Almaden in the 1970s

Physical Integrity in a Large Segmented Database

RAYMOND A. LORIE
IBM Research Laboratory

1. INTRODUCTION



FILE STRUCTURE DESIGN TO FACILITATE ON-LINE INSTANTANEOUS UPDATING

Robert L. Rappaport

Systems Analysis & Software Design, Inc.

INTRODUCTION

VADIS LOGICAL FILE STRUCTURE

MODFILE STRUCTURE

WRITE-AHEAD LOGGING

WRITE-BEHIND LOGGING

EVALUATION

MULTI-VERSIONED DBMS

TUPLE ID	BEGIN TIMESTAMP	END TIMESTAMP	PREVIOUS VERSION	TUPLE DATA
1	10	20	—	V-1
2	20	∞	1	V-2

Microsoft® SQL Server® Hekaton

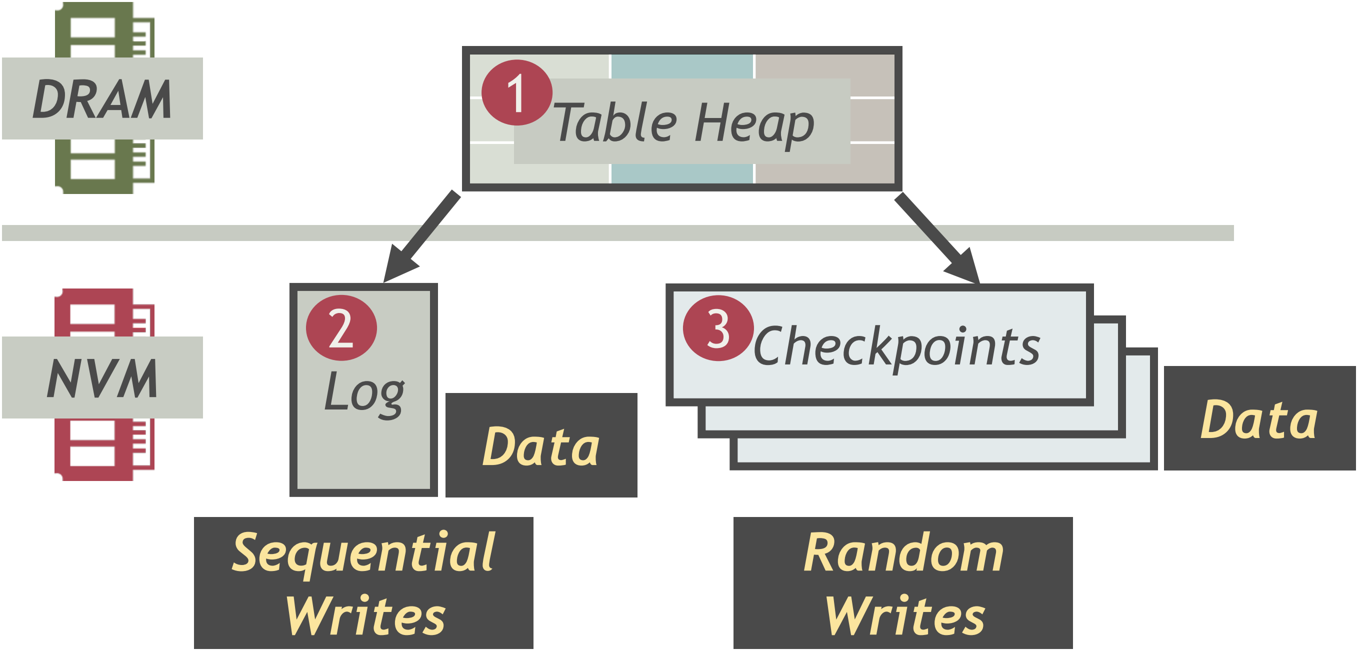
SAP® HANA

PostgreSQL

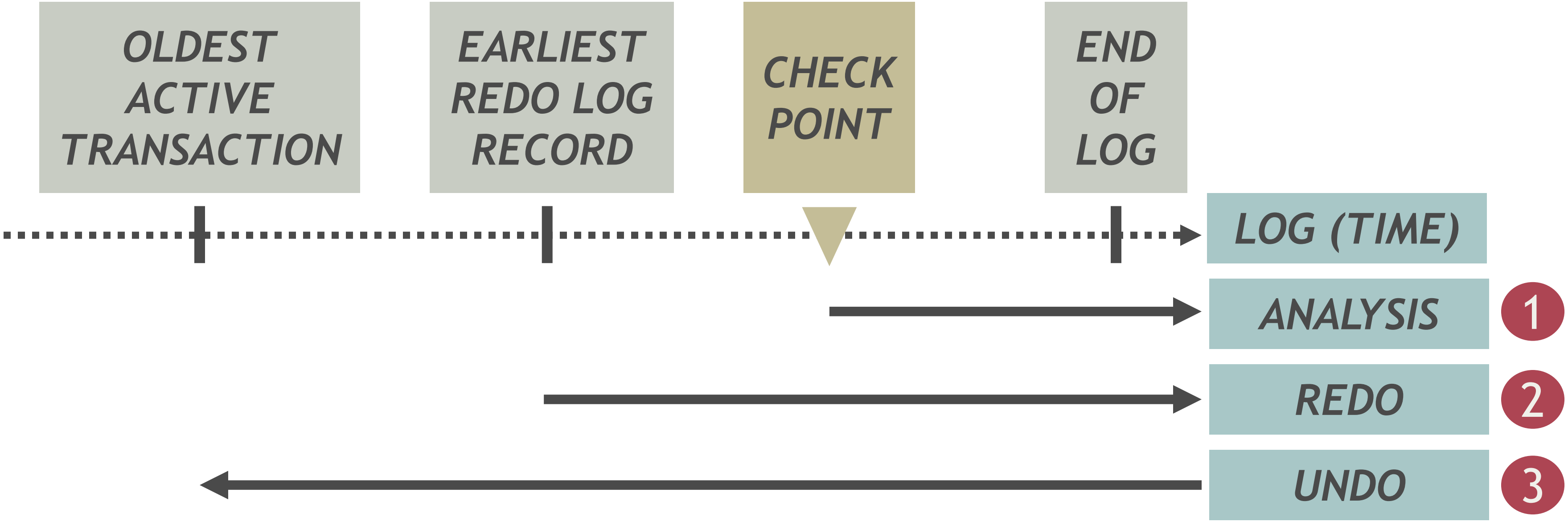
MySQL™

ORACLE®

WRITE-AHEAD LOGGING



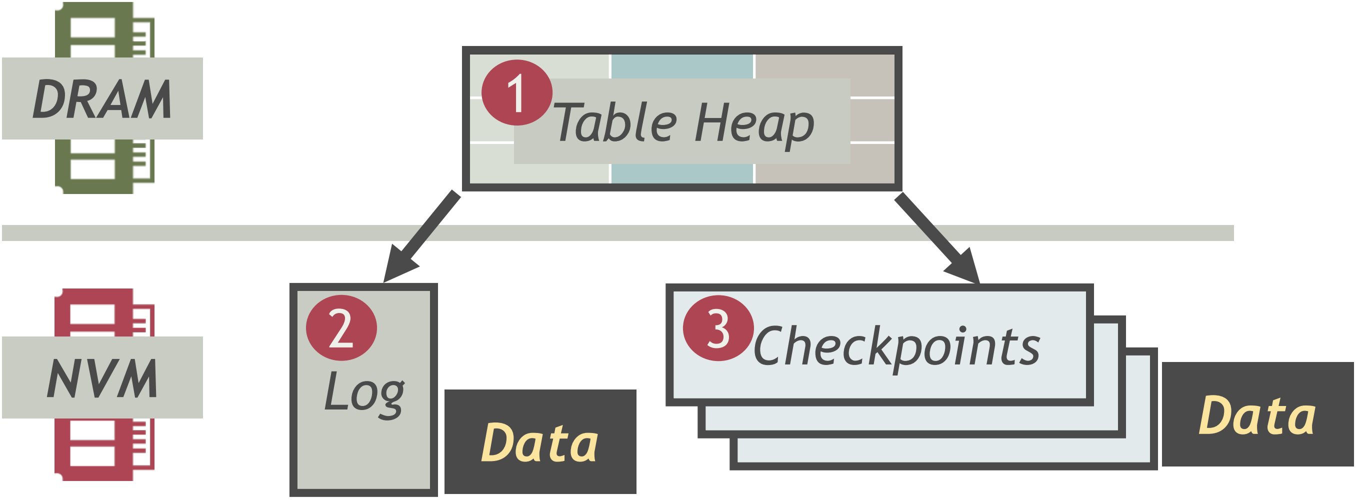
RECOVERY


OLDEST ACTIVE TRANSACTION
EARLIEST REDO LOG RECORD
CHECK POINT
END OF LOG
LOG (TIME)
ANALYSIS
1
REDO
2
UNDO
3

PROBLEM #1: SLOW RECOVERY

- Replaying the redo log takes time
  - *Recovery time depends linearly on the number of log records*
  - *Which in turn depends on frequency of checkpointing*
- Slow recovery hurts application availability
  - *Even with replication, bringing up new replicas takes time*
  - *This increases the vulnerability of the system*

PROBLEM #2: DATA DUPLICATION



How can we leverage NVM to support faster recovery and reduce data duplication?

WRITE-AHEAD LOGGING

WRITE-BEHIND LOGGING

EVALUATION

WRITE-BEHIND LOGGING

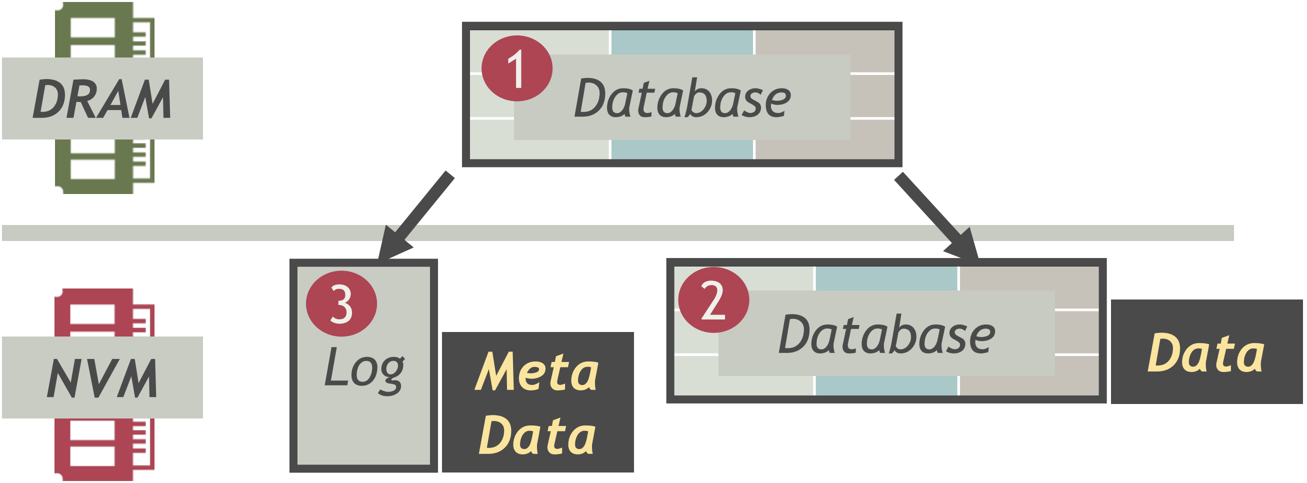
- Write-ahead log serves two purposes
  - *Transform random database writes into sequential log writes*
  - *Support transaction rollback*
- Designed for hard disks that can only slow random writes
  - *But, NVM supports fast random writes*

Why not directly write changes out to the multi-versioned database at runtime?

WRITE-BEHIND LOGGING

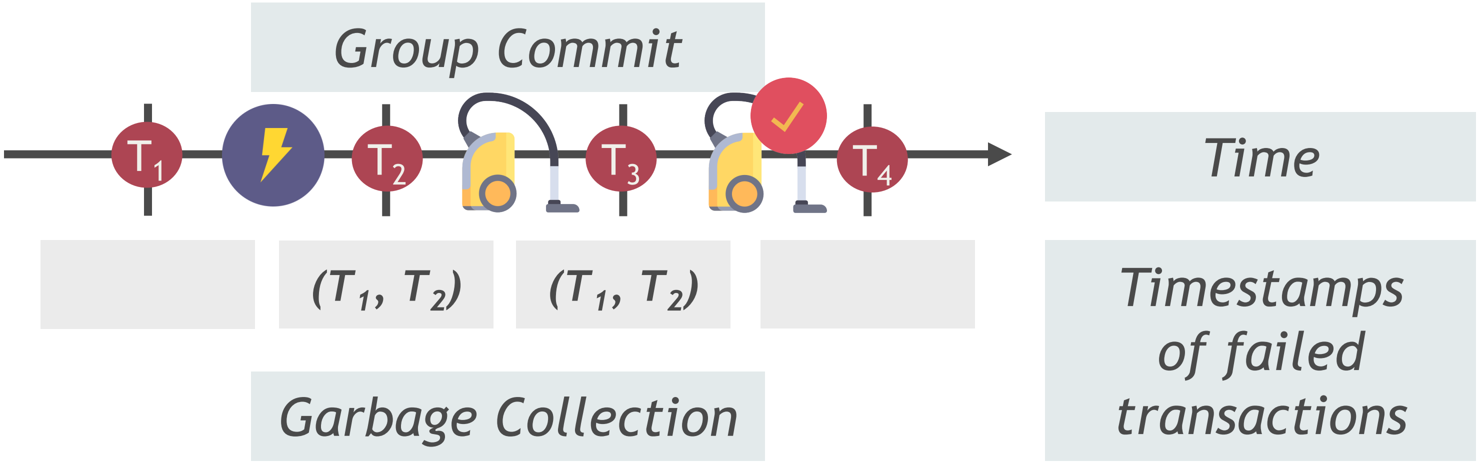
- No physical redo
  - *Directly writes changes to the database at runtime*
  - *Due to multi-versioning, it does not overwrite data*
- Logical undo
  - *Does not duplicate tuple data in the log and checkpoints*
  - *Instead, it only records transaction metadata*
  - *Sufficient to support transaction rollback*

WRITE-BEHIND LOGGING



METADATA FOR LOGICAL UNDO

- Record timestamps of failed transactions in log
  - *Ignore versions changed during those timestamps (logical undo)*



Group Commit

Time

(T_1, T_2)

(T_1, T_2)

Garbage Collection

Timestamps of failed transactions

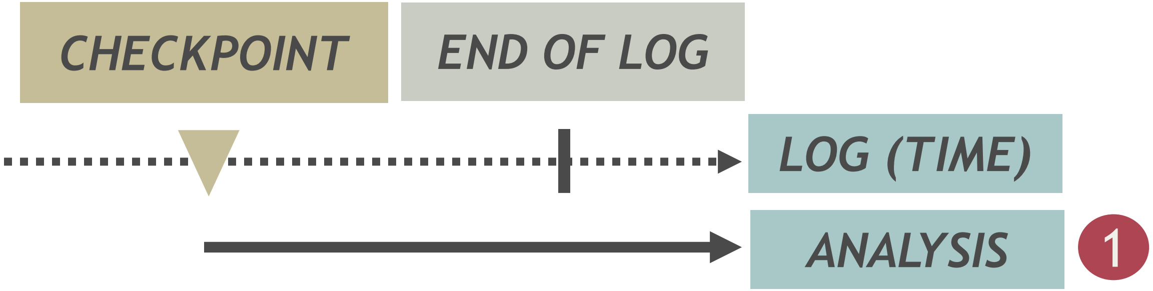
SOLUTION #1: NO DATA DUPLICATION

LSN	LOG RECORD TYPE	TIMESTAMPS OF FAILED TRANSACTIONS
1	GROUP COMMIT	$[T_1, T_2]$

Write-behind logging avoids data duplication by only recording transaction metadata in log

SOLUTION #2: INSTANT RECOVERY

- Single analysis phase to determine failed transactions
  - *No physical redo: All changes written to database at runtime*
  - *Logical undo: Skip reading effects of uncommitted transactions*



Write-behind logging enables instant recovery by eliminating redo and doing logical undo

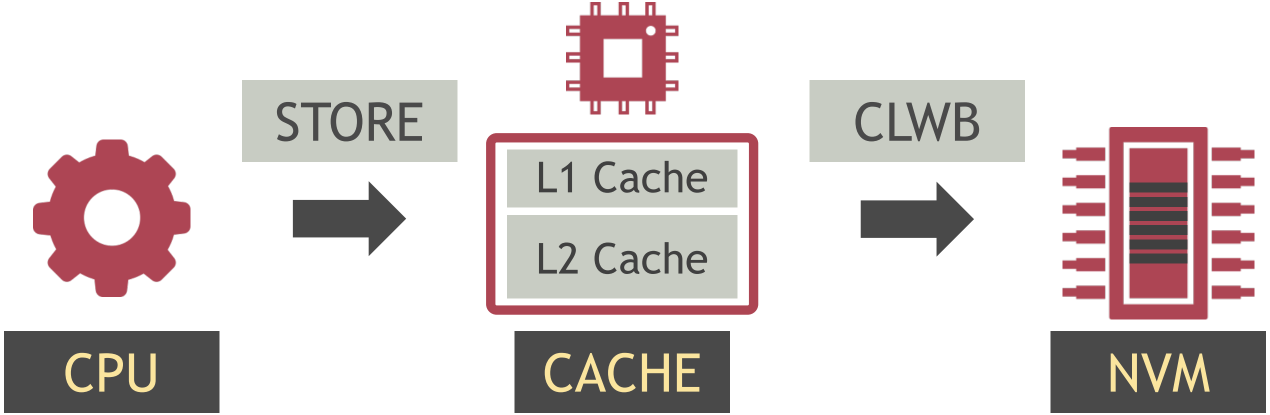
WRITE-AHEAD LOGGING

WRITE-BEHIND LOGGING

EVALUATION

NVM HARDWARE EMULATOR

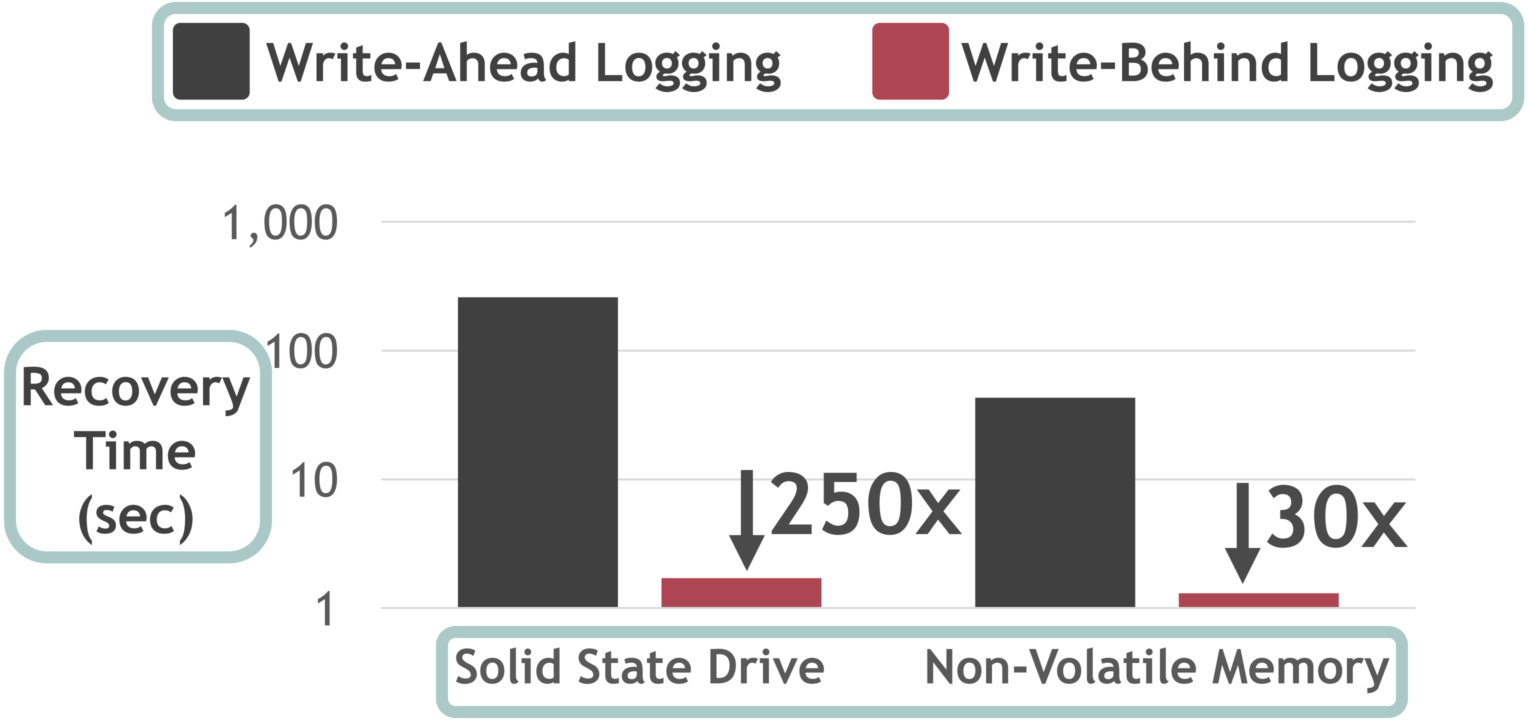
- Special CPU microcode to add stalls on cache misses
  - *Tune DRAM latency to emulate different NVM technologies*
- New assembly instructions for managing NVM
  - *Cache-line write-back (CLWB) instruction*



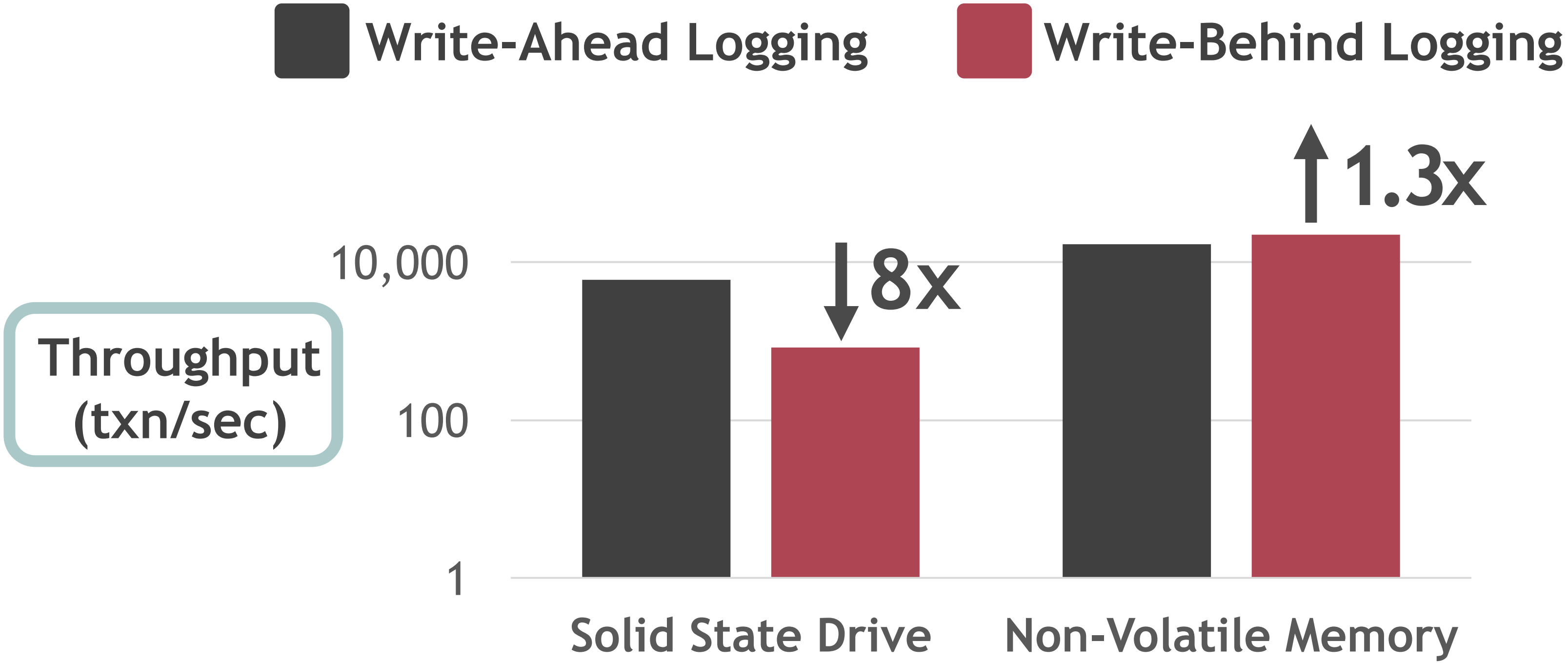
EVALUATION

- Compare logging protocols in Peloton
  - *Write-Ahead logging*
  - *Write-Behind logging*
- TPC-C benchmark
- Storage devices
  - *Solid-state drive*
  - *Emulated non-volatile memory*

RECOVERY TIME



THROUGHPUT


Write-Ahead Logging
Write-Behind Logging
Throughput (txn/sec)
10,000
100
1
↓8x
↑1.3x
Solid State Drive
Non-Volatile Memory

TAKEAWAYS

- Write-behind logging
  - *Enables instant recovery from failures*
  - *Illustrates importance of rethinking algorithms for NVM*
- NVM upends key assumptions about storage
  - *This impacts all the layers of a DBMS*
  - *It’s time for complete system rewrite*

PELOTON

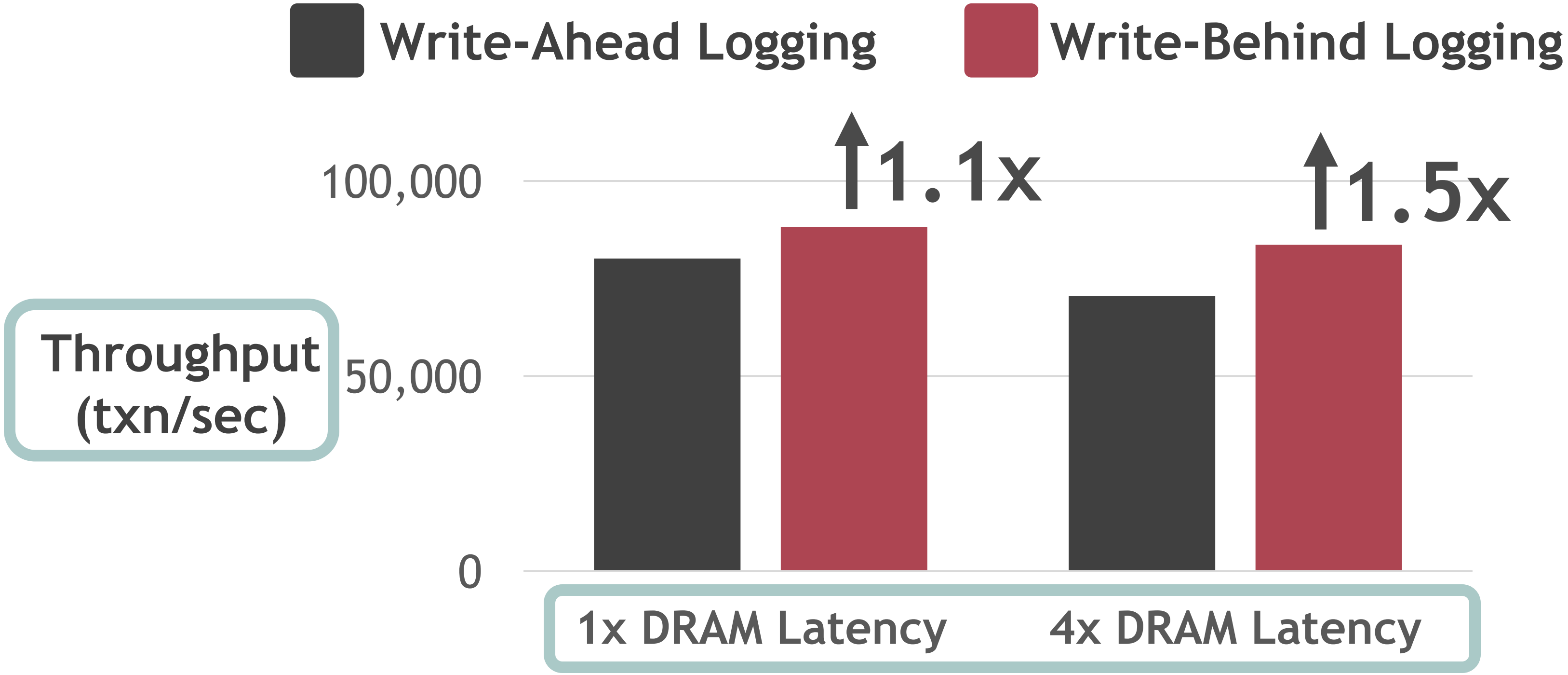
http://pelotondb.org

- NVM Ready
- Autonomous
- Apache Licensed

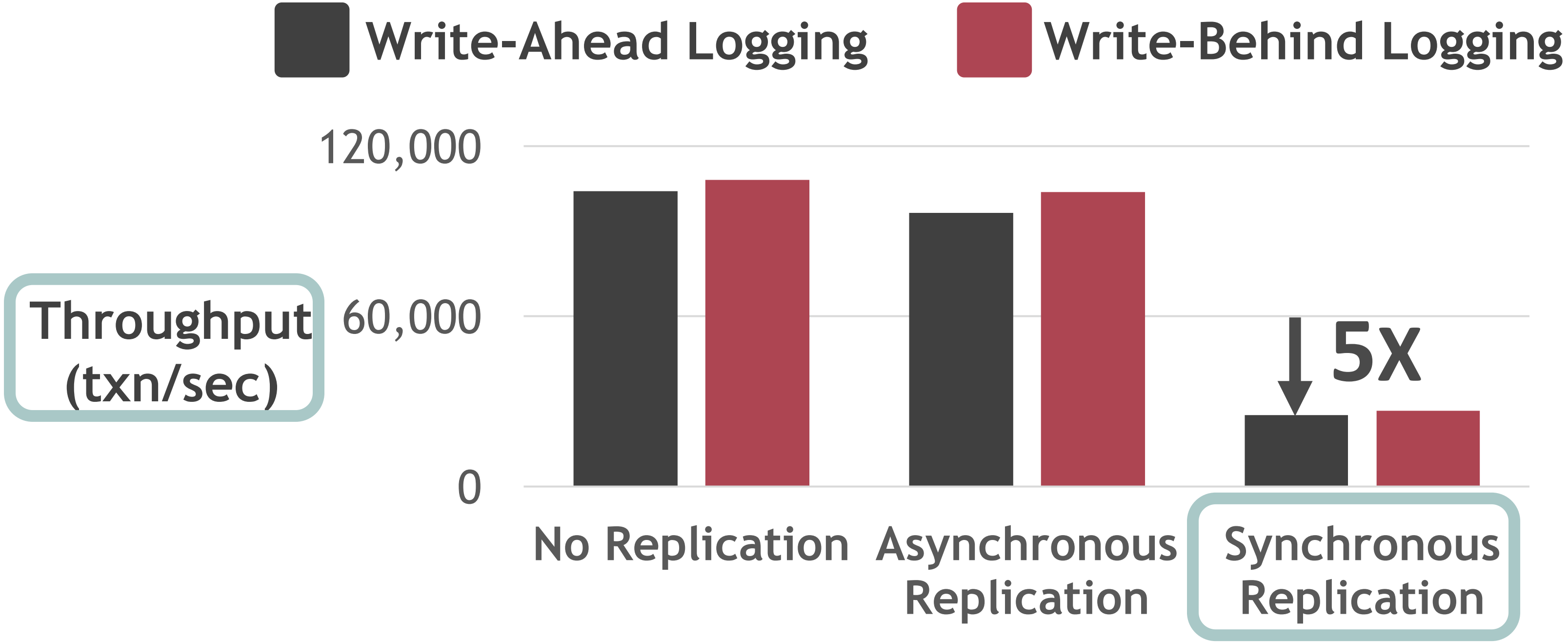
END

@joy_arulraj

IMPACT OF NVM LATENCY



REPLICATION


Write-Ahead Logging
Write-Behind Logging
Throughput (txn/sec)
5X
No Replication
Asynchronous Replication
Synchronous Replication