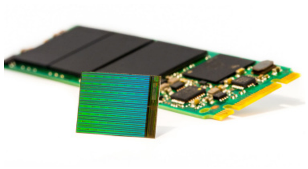
HOW TO BUILD A NON-VOLATILE MEMORY DATABASE SYSTEM

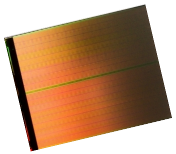
JOY ARULRAJ
CARNEGIE MELLON UNIVERSITY

ISTC
BIG DATA

NON-VOLATILE MEMORY (NVM)



DRAM



NVM

SSD

Like DRAM, low latency loads and stores

Like SSD, persistent writes and high density

Why we think NVM is happening for real this time?

#1: INDUSTRY STANDARDS

- Standard definitions of NVM technologies
  - *Form factors (e.g., JEDEC classification)*
  - *Interface specifications (e.g., NVM Express over Fabrics)*

JUNE 2016

JEDEC®

nvm EXPRESS®

#2: OPERATING-SYSTEM SUPPORT

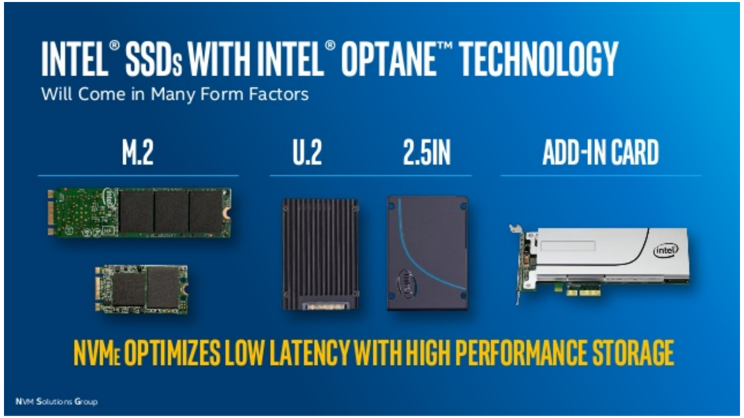
- Growing OS support for NVM
  - *Linux 4.8, e.g. NVM Express over Fabrics library*
  - *Windows 10, e.g. Direct access to files on NVM*

OCTOBER 2016

#3: PROCESSOR SUPPORT

- Intel’s Kaby Lake processor
  - *Support for 3D XPoint NVM technology*
  - *ISA updates for NVM management*

JANUARY 2017



INTEL® SSDs WITH INTEL® OPTANE™ TECHNOLOGY

Will Come in Many Form Factors

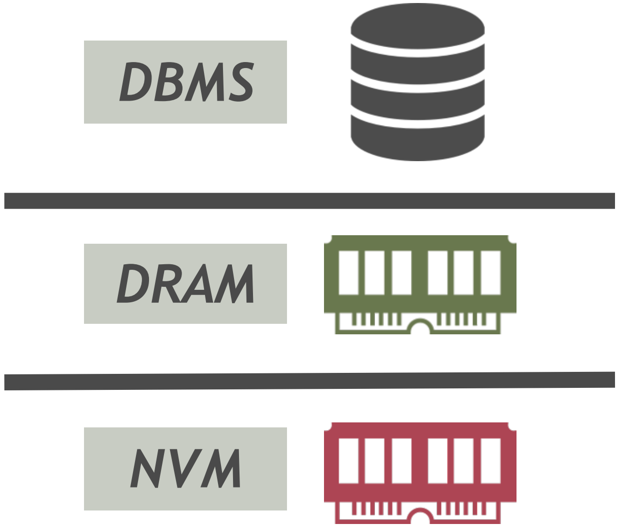
NVMe OPTIMIZES LOW LATENCY WITH HIGH PERFORMANCE STORAGE

NVM Solutions Group

How can we leverage NVM in a DBMS?

#1: DISK-ORIENTED DBMSs

- Treat NVM like a faster SSD



DBMS

DRAM

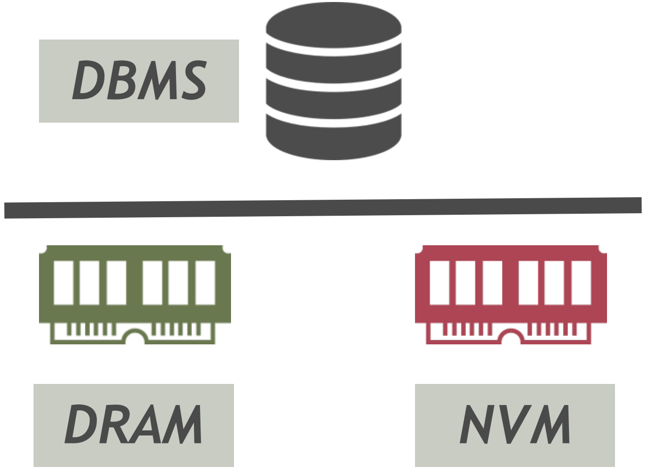
NVM

Designed to minimize random writes to NVM

But, NVM supports fast random writes

#2: MEMORY-ORIENTED DBMSs

- Treat NVM as extended memory



Designed to overcome the volatility of memory

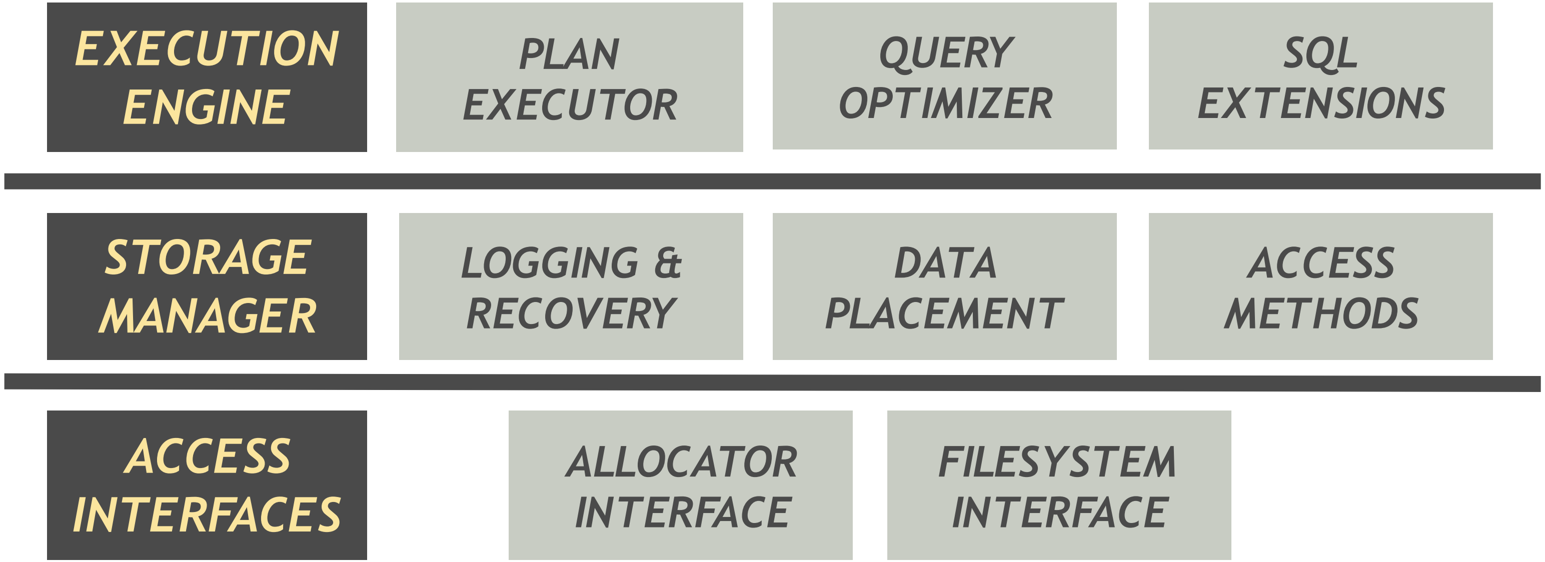
But, writes to NVM are persistent

DBMS OVERVIEW

LOGGING & RECOVERY

DATA PLACEMENT

NVM-AWARE DBMS OVERVIEW



EXECUTION ENGINE

PLAN EXECUTOR

QUERY OPTIMIZER

SQL EXTENSIONS

STORAGE MANAGER

LOGGING & RECOVERY

DATA PLACEMENT

ACCESS METHODS

ACCESS INTERFACES

ALLOCATOR INTERFACE

FILESYSTEM INTERFACE

HOW TO BUILD A NON-VOLATILE MEMORY DBMS
UNDER SUBMISSION

#1: ACCESS INTERFACES

- Allocator Interface
  - *Provide a durability primitive*
  - *Prevent persistent memory leaks*
- Filesystem Interface
  - *Direct access to files on NVM*
  - *Avoid extra copy in page cache*

#2: STORAGE MANAGER

- Logging and Recovery
  - *Leverage NVM's ability to support fast random writes*
  - *Enable instantaneous recovery from failures*
- Access Methods
  - *Read and write latencies of NVM are asymmetric*
  - *Write-limited access methods such as B+tree*

UC San Diego

UNIVERSITY OF TORONTO

#3: EXECUTION ENGINE

- Plan Executor
  - *Write-limited sorting algorithm*
  - *Makes use of selection sort which takes multiple read passes*
- Query Optimizer
  - *Differentiate between reads and writes in cost model*
  - *Factor in byte-addressability of NVM*

THE UNIVERSITY OF EDINBURGH

TUM
TECHNISCHE
UNIVERSITÄT
MÜNCHEN

DBMS OVERVIEW

LOGGING & RECOVERY

DATA PLACEMENT

WRITE-AHEAD LOGGING

Can we avoid duplicating data in the log and the checkpoints ?

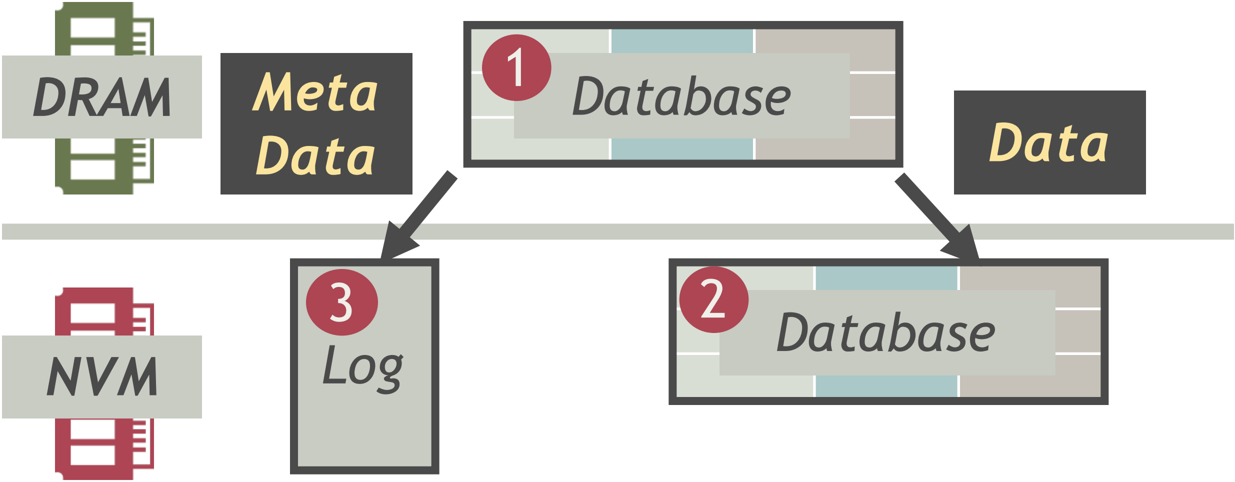
WRITE-BEHIND LOGGING

- Write-ahead log serves two purposes
  - *Transform random database writes into sequential log writes*
  - *Support transaction rollback*
- NVM supports fast random writes
  - *Directly write data to the multi-versioned database*
  - *Later, record metadata about committed txns in log*

WRITE-BEHIND LOGGING
VLDB 2016

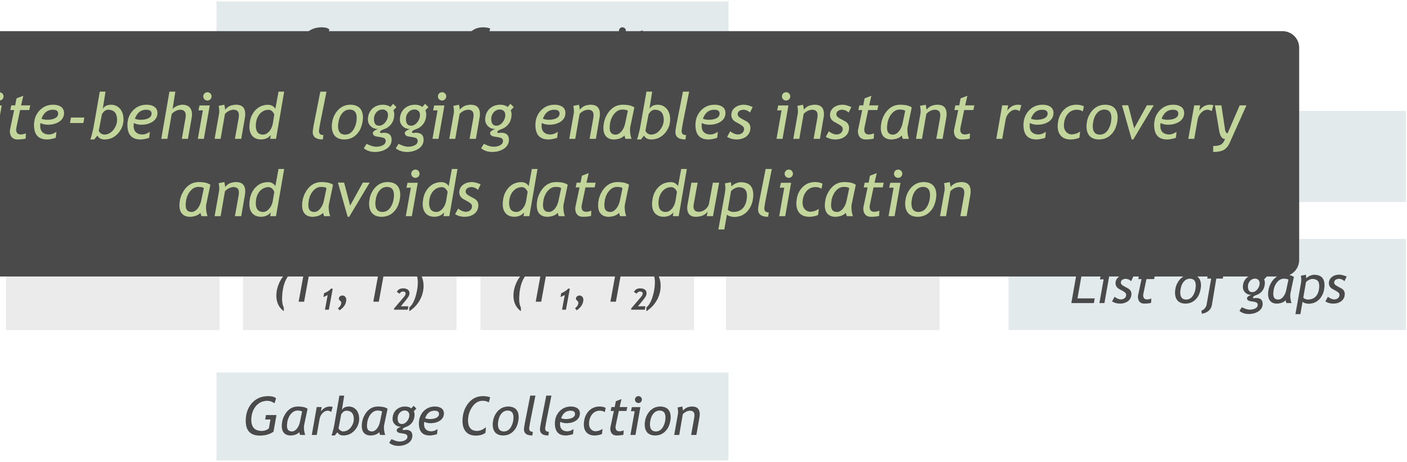
LET'S TALK ABOUT STORAGE AND RECOVERY METHODS FOR NON-VOLATILE MEMORY DATABASE SYSTEMS
SIGMOD 2015

WRITE-BEHIND LOGGING



METADATA FOR INSTANT RECOVERY

- Record failed group commit timestamp gap in log
  - *Use it to ignore effects of uncommitted transactions*



(T_1, T_2)

(T_1, T_2)

List of gaps

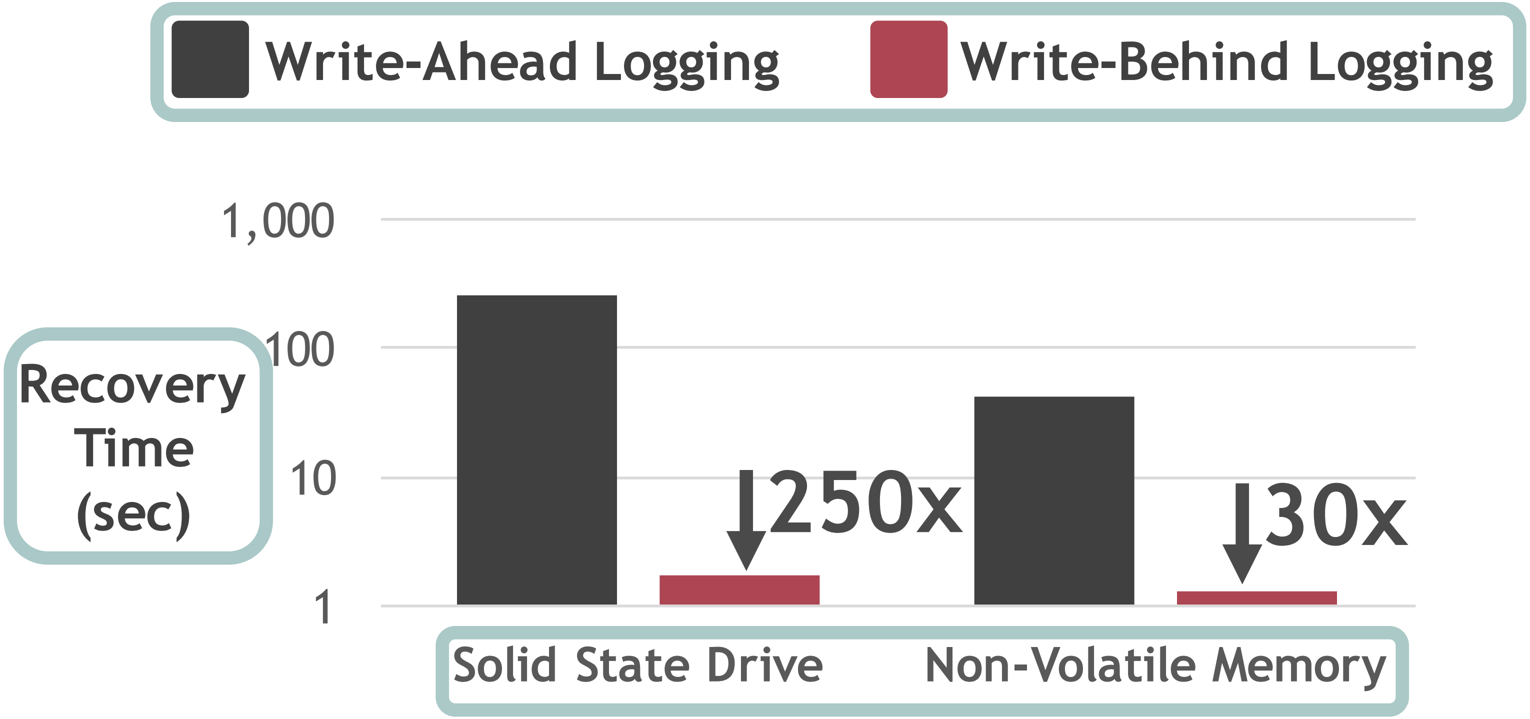
Garbage Collection

Write-behind logging enables instant recovery and avoids data duplication

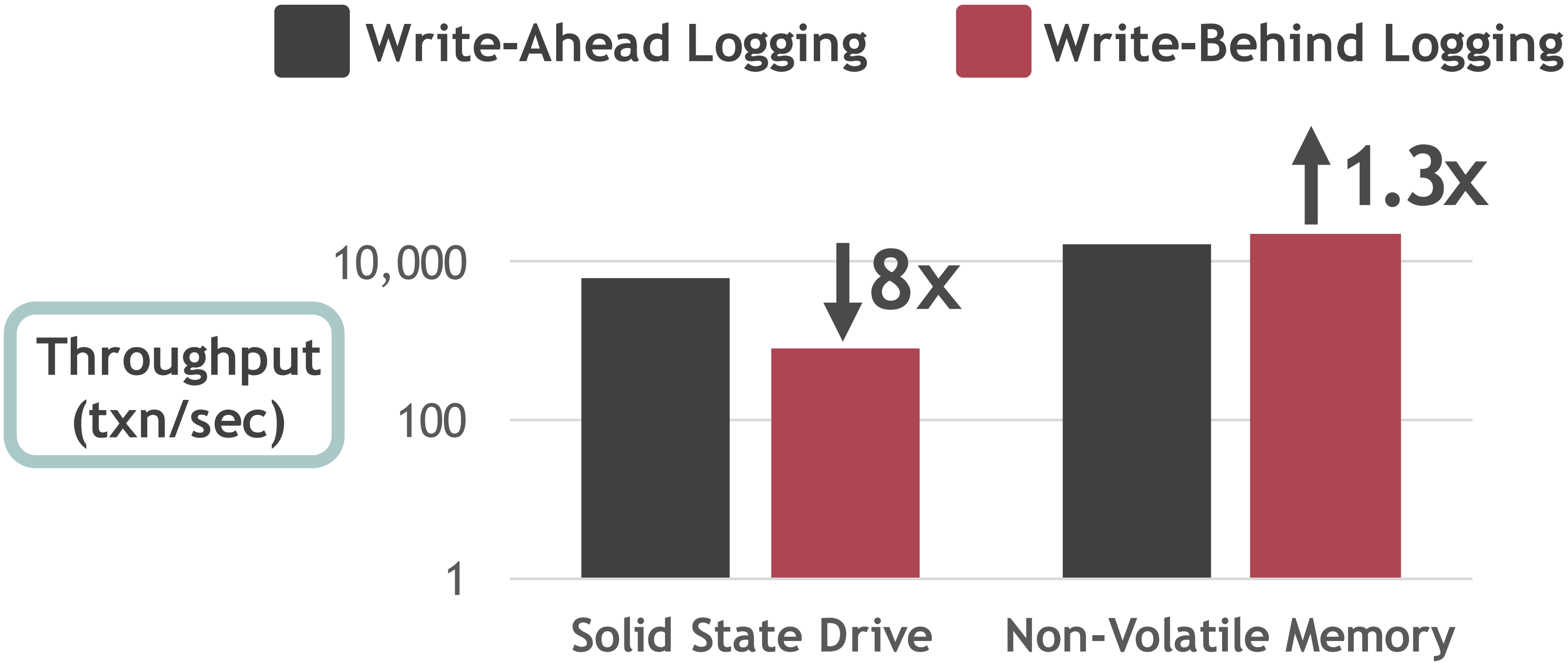
EVALUATION

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

DBMS OVERVIEW

LOGGING & RECOVERY

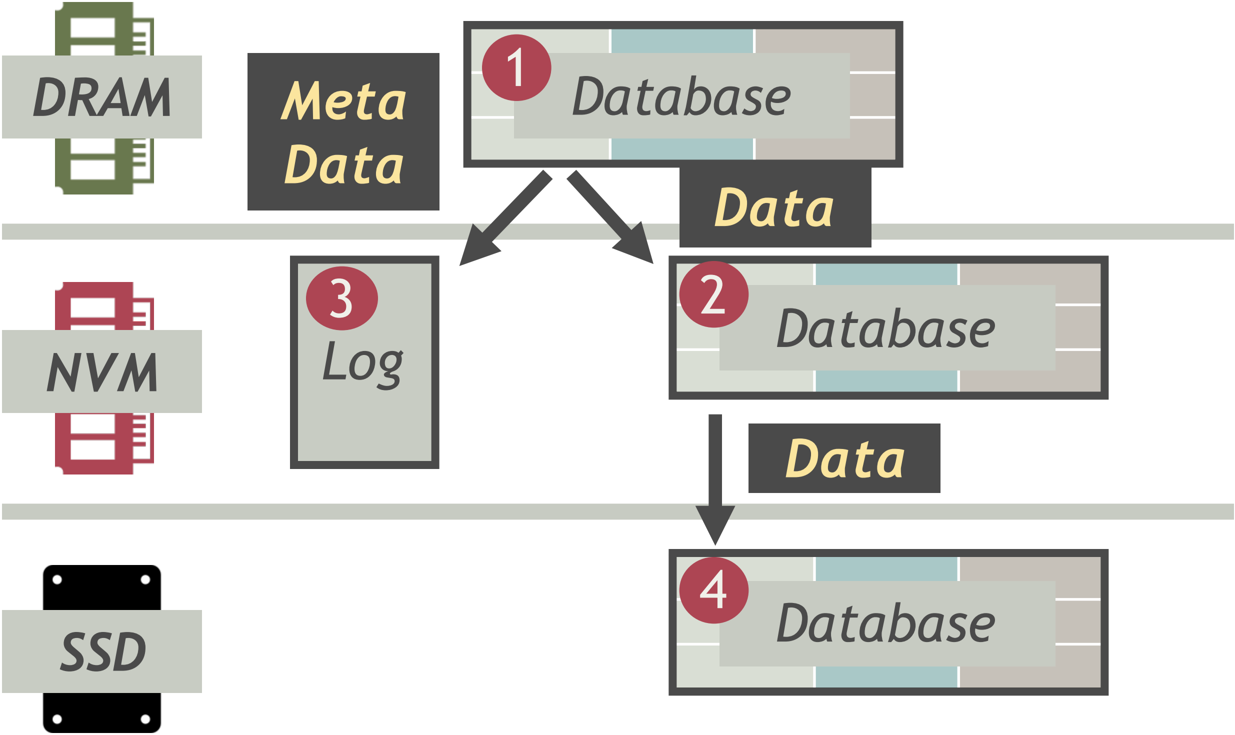
DATA PLACEMENT

NVM-AWARE DATA PLACEMENT

- Support analytics on a multi-tier storage hierarchy
  - *Cost of first-generation NVM devices*
  - *DRAM + NVM + SSD*

When should the DBMS migrate data between devices in storage hierarchy?

THREE-TIER STORAGE HIERARCHY

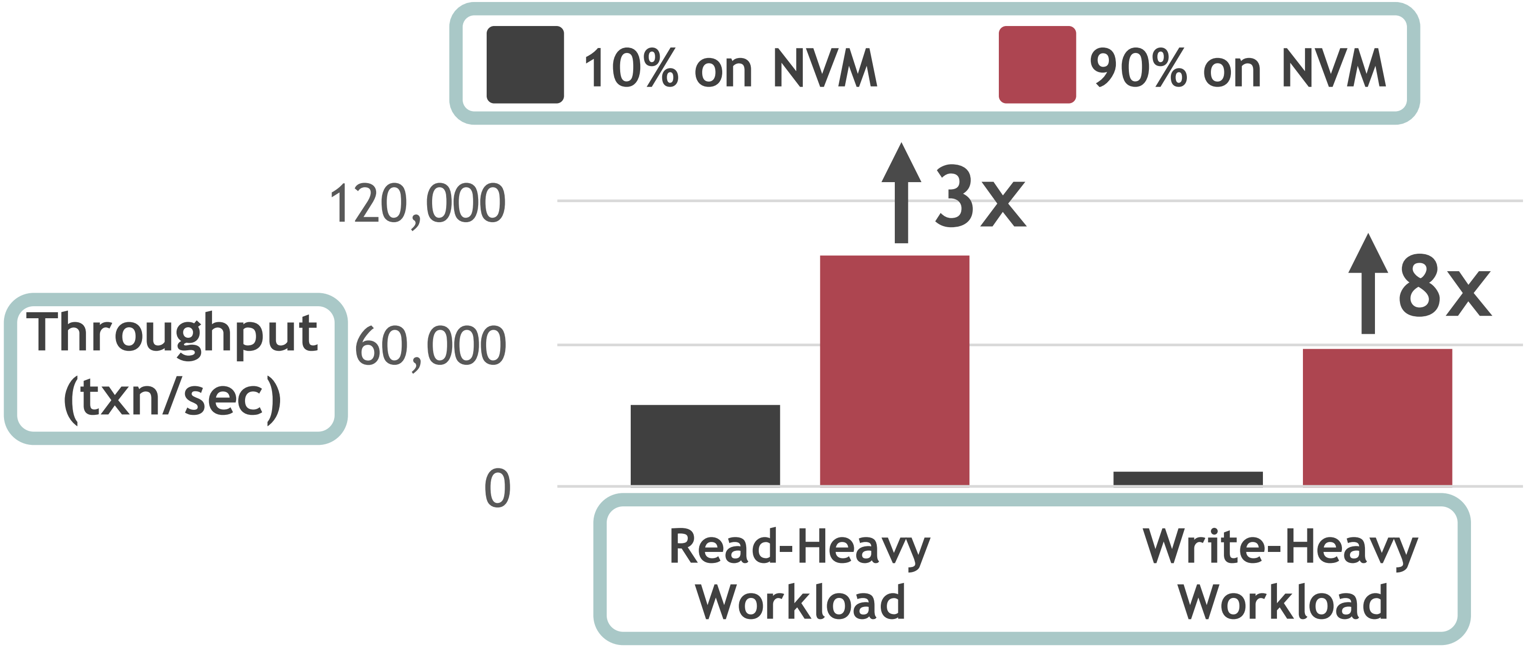


DATA PLACEMENT

- Can directly read data from NVM
  - *No need to copy data over to DRAM for reading*
- Cache hot data in DRAM
- Dynamically migrate cold data to SSD
  - *And bring back warm data to NVM*

DATA PLACEMENT IN NON-VOLATILE MEMORY DATABASE SYSTEMS
WORK IN PROGRESS

THREE-TIER STORAGE HIERARCHY



DBMS
OVERVIEW

LOGGING &
RECOVERY

DATA
PLACEMENT

THE HOME STRETCH

- #1: NVM-aware B+tree (with Microsoft Research)
  - *Write-limited design for NVM*
- #2: Data placement in multi-tier storage hierarchy
  - *Data migration policies*
- #3: Replication
  - *NVM Express over Fabrics library*

PELOTON

http://pelotondb.org

- NVM Ready
- Autonomous
- Apache Licensed

END

@joy_arulraj