IEEE NVMSA ’21

Designing a persistent-memory-native storage engine for SQL database systems

August 19, 2021

Shohei Matsuura
Yahoo Japan Corporation

Outline

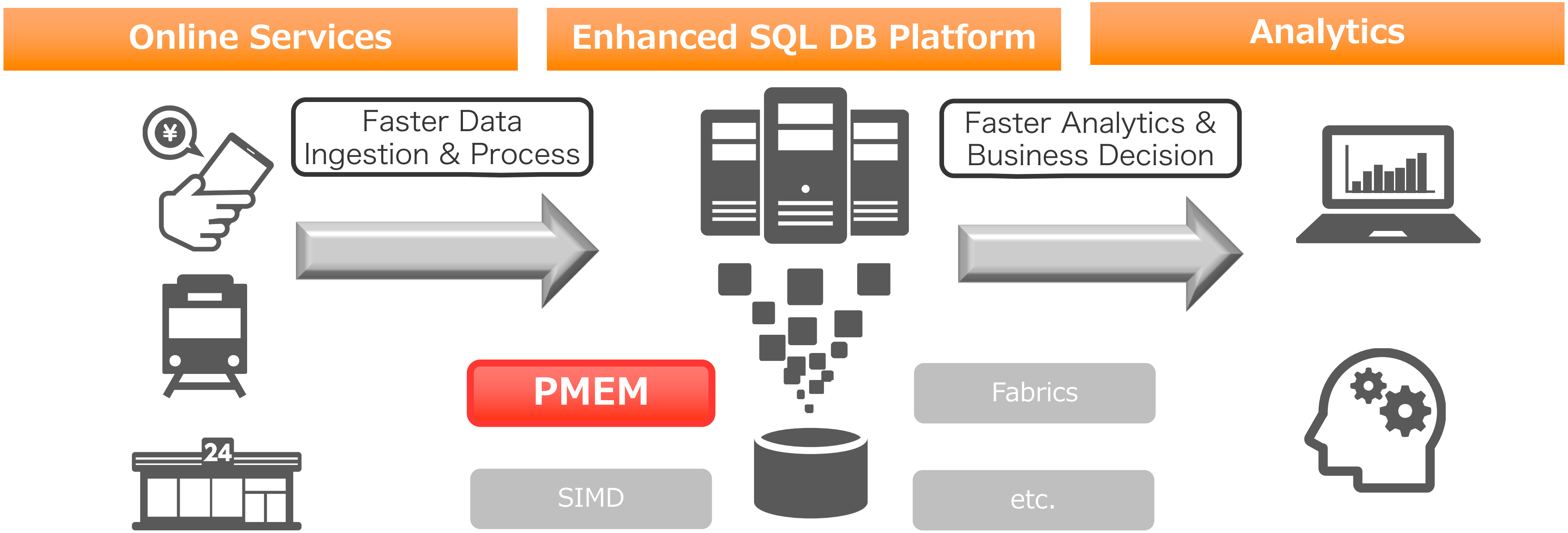
1. Background
2. Our Work
3. Evaluation
4. Conclusion & Future Work

History & Figures about Yahoo Japan Corporation

- Yahoo Japan Corp. is one of the major internet companies in Japan since 1996.
- It has 52 million user logins monthly and offers 100+ internet services ranging from e-commerce and online new media.

Research Motivation

- To Extend & enhance the capabilities of Yahoo! Japan's SQL database platform with the latest hardware technologies (i.e. Persistent Memory, or PMEM)

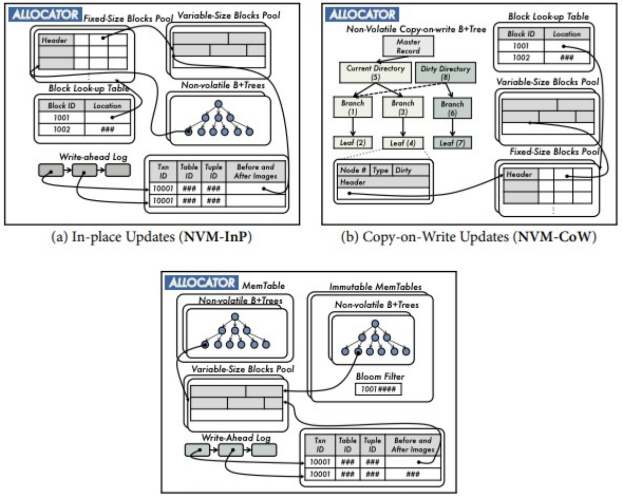


Existing Research & Application in Academia

Design Ideas for a PMEM-native Storage Engine

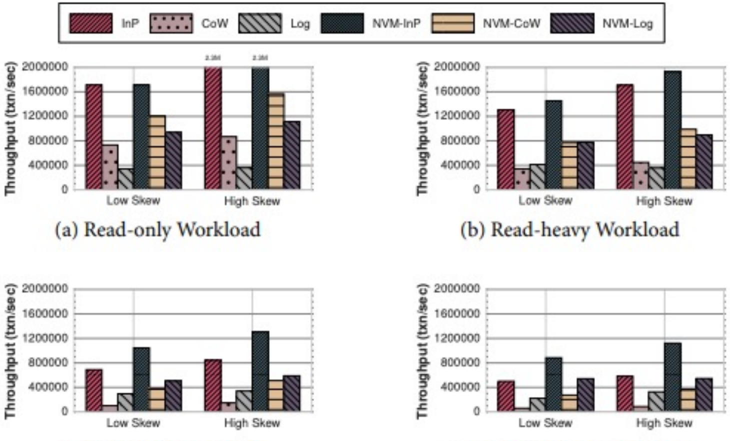
- Arulraj (2018), Arulraj & Pavlo (2019)
  - Present 3 design ideas for the storage engine: (1) NVM-InP, (2) NVM-CoW, (3) NVM-Log
  - Emulator-based performance study of the 3 designs with YCSB workloads

Storage Engine Design Ideas



(a) In-place Updates (NVM-InP) (b) Copy-on-Write Updates (NVM-CoW) (c) Log-structured Updates (NVM-Log)

Emulator-based Performance Study

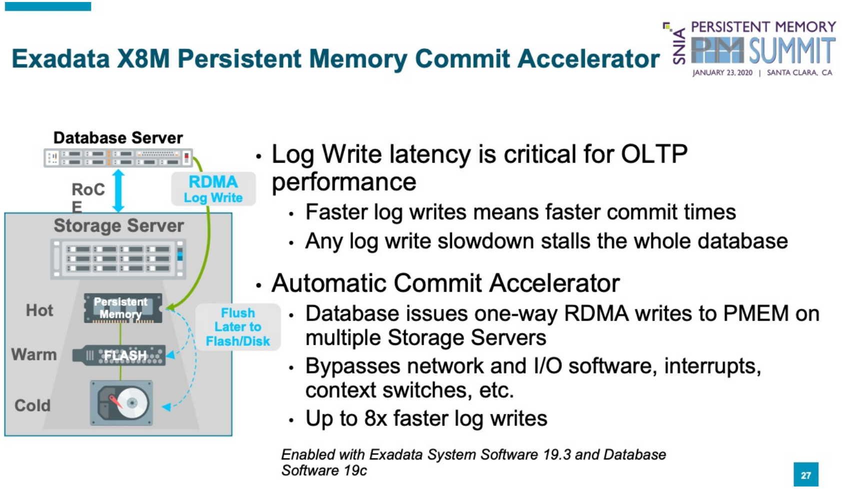


(a) Read-only Workload (b) Read-heavy Workload (c) Balanced Workload (d) Write-heavy Workload

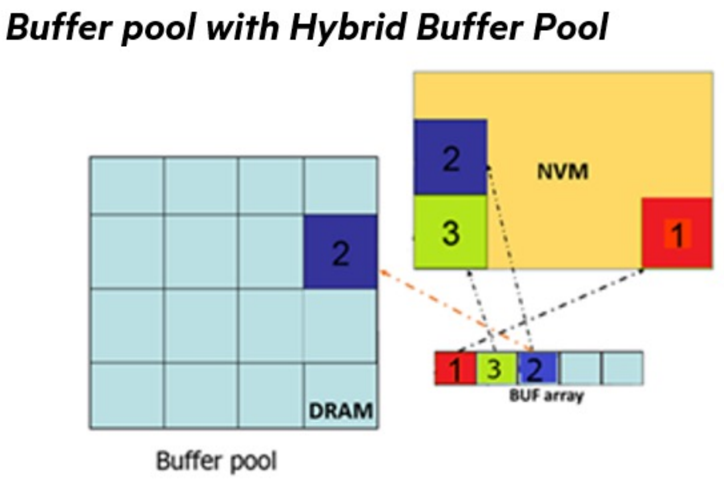
Existing Research & Application in Industry

Transaction Logging & Database Buffer Extension

- Oracle Exadata*
  - ➢ Accelerate Transaction Logging with PMEM



- Microsoft SQL Server**
  - ➢ Extend buffer pool & evicted pages direct read from PMEM (NVM)



* J.Shi, "Exadata with Persistent Memory: An Epic Journey." SNIA Persistent Memory Summit 2020.
https://www.snia.org/sites/default/files/PM-Summit/2020/presentations/11_PMEM_Jia_Shi_final_PM_Summit_2020_v2.pdf
** Microsoft Corporation, "SQL Server Hybrid Buffer Pool."
https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/hybrid-buffer-pool?view=sql-server-ver15

YAHOO! JAPAN

Research Goals

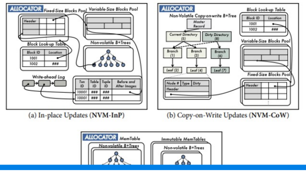
To state requirements for a practical storage engine that natively uses persistent memory for SQL database systems, and to illustrate how to design such a storage engine

Existing Research & Application in Academia

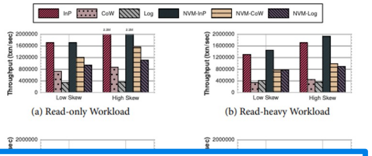
Design Ideas for a PMEM-native Storage Engine

- Arulraj(2018), Arulraj & Pavlo (2019)
  - Present3 design ideas for the storage engine: (1) NVM-InP, (2) NVM-CoW, (3) NVM-Log
  - Emulator-based performance study of the 3 designs with YCSB workloads

Storage Engine Design Ideas



Emulator-based Performance Study

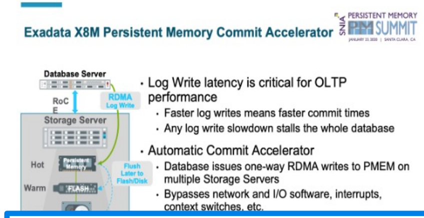


Making them practical is an open question!

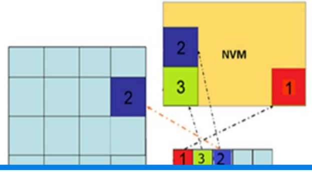
Existing Research & Application in Industry

Transaction Logging & Database Buffer Extension

- Oracle Exadata
  - Accelerate Transaction Logging with PMEM
- Microsoft SQL Server
  - Extend buffer pool & direct page read from PMEM



Buffer pool with Hybrid Buffer Pool



Partial use of persistent memory for database operations

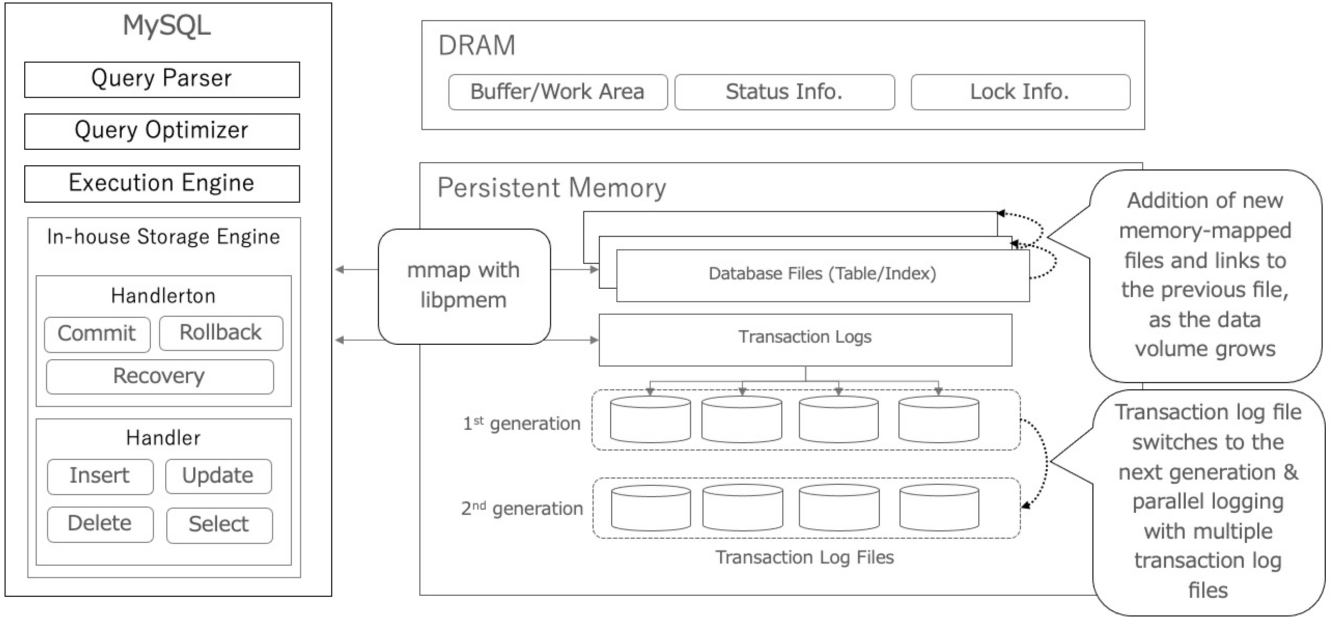
Requirements for a Practical PMEM-native Storage Engine

- Based on the current requirements for SQL database systems at Yahoo! Japan, we impose the same and the following requirements for the storage engine:

1. Scale with Data
2. Transaction Support
3. Continuous Operation
4. Performance
5. MySQL Compatibility

Design for a Practical PMEM-native Storage Engine

- Place database & transaction log files on PMEM
- Storage expansion with linked database files, and transaction log switches for continuous operation
- WAL & Aries-based transaction support



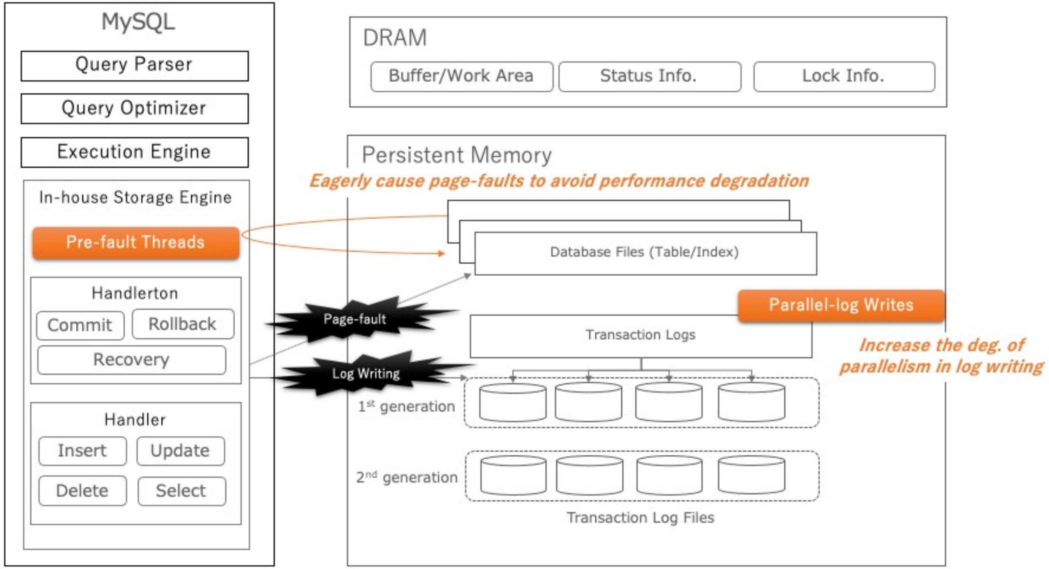
Two Important Design Features for Performance in the Storage Engine

1. Pre-fault

- Page-fault causes significant performance degradation when accessing mmap files [Choi & Kim, 2017]
- To avoid it during query processing, implement designated threads (pre-fault threads) to cause pre-fault before the storage engine main threads access the mmap files (database files & transaction log files)

2. Parallel-logging

- the state-of-the-art “parallel write-ahead logging algorithm” to increase the deg. of transaction log writing [Tanabe et al., 2018]



Evaluation: Environment & Workload

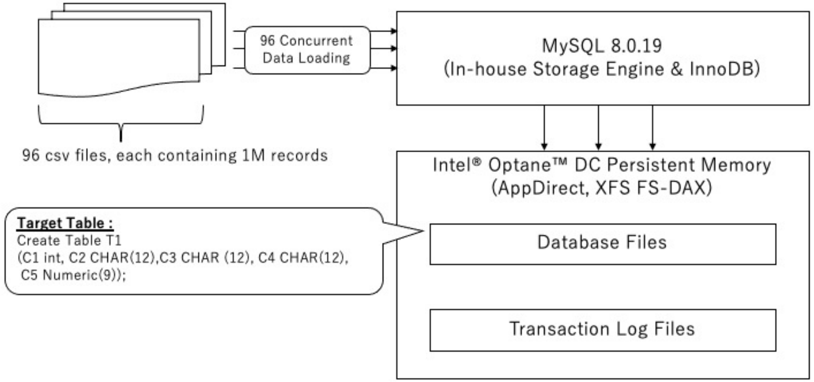
■ Evaluation Environment

- 2-Socket Server with 104-core
- Equipped with Intel® Optane™ DCPMM

CPU	Intel Xeon Gold 6230R 2.1GHz x 2 (Total 104 Cores)
DRAM	DDR4-192GB
Persistent Memory	Intel® Optane™ DC Persistent Memory
SSDs	SATA SSD 1.92TB (OS Boot, Load Data)
OS	CentOS 7.8
DBMS	MySQL 8.0.19 with the In-house Storage Engine

■ Evaluation Workload

- 96 concurrent data loading
- Good workload to observe the effects of the pre-fault & the parallel logging as it always accesses a new region of a mmap file and generates transaction logs



Evaluation: Effects of Pre-fault and Parallel-logging Features

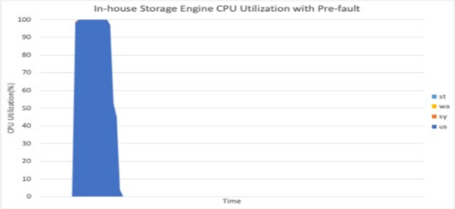
Pre-fault Feature

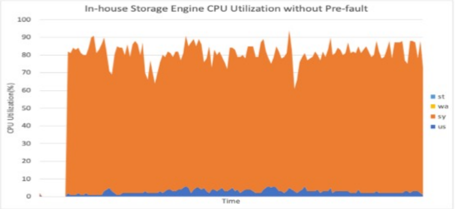
- More than 5x performance improvement
- Significant improvement in CPU utilization

Normalized Data-loading Time

Storage Engine	In-house Storage Engine with Pre-fault Feature	In-house Storage Engine without Pre-fault Feature
Loading Time	1	5.86

CPU Utilization & Perf Output






```
98.83% 0.00%   mysqld mysqld          [.] mysql_execute_command
        |
        |---mysql_execute_command
               |
               |--98.83%--Sql_cmd_load_table::execute
                      |
                      |--97.90%--Sql_cmd_load_table::read_rep_field
                             |
                             |--81.65%--write_record
                                    |
                                    |--79.20%--handler::ha_write_row
                                           |
                                           |--71.79% in_house_se:write_row
                                                  |
                                                  |--71.67%--row_to_heap
```

```
98.90% 0.00%   mysqld mysqld          [.] mysql_execute_command
        |
        |---mysql_execute_command
               |
               |--98.90%--Sql_cmd_load_table::execute
                      |
                      |--98.51%--Sql_cmd_load_table::read_rep_field
                             |
                             |--94.78%--write_record
                                    |
                                    |--94.50%--handler::ha_write_row
                                           |
                                           |--89.74% in_house_se:write_row
                                                  |
                                                  |--87.01%--page_fault
```

Parallel-logging Feature

- 30%+ performance improvement
- Increasing the deg. of parallel log write too much also causes performance degradation

Normalized Data-loading Time

Number of Parallel Log Write	1	2	4	8	10	12	16
Loading Time	1.31	1.06	1.00	1.17	1.18	1.05	1.28

Evaluation: Overall Performance Improvement by the Storage Engine

■ Overall Performance Comparison

- ➢ More than 50x performance improvement with our in-house storage engine than InnoDB running on PMEM
- ➢ In-house storage engine run with the pre-fault feature enabled and the parallel log write=4

Normalized Data-loading Time

Storage Engine	In-house Storage Engine with Pre-fault & Parallel-logging Features	InnoDB on Persistent Memory
Loading Time	1	58.29

Conclusion & Future Work

- **Conclusion:**
  - ✓ Presented & discussed five requirements for a practical PMEM-native storage engine that satisfies industry requirements
  - ✓ Two important design features, (1) pre-fault and (2) parallel-logging, to make a storage engine performant on PMEM
  - ✓ Overall, our designed in-house storage engine achieves 50x+ performance in write-workload on PMEM compared to InnoDB on PMEM
- **Future Work:**
  - ✓ Data Tiering to handle more data than PMEM capacity
  - ✓ High-Availability feature to ensure database operations can continue even in the case of a data center failure

Trademarks & Registered Trademarks

- Oracle and MySQL are registered trademarks of Oracle and/or its affiliates.
- Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Intel® and Intel® Optane™ are trademarks of Intel Corporation or its subsidiaries.

Thank you very much! ☺

Please send us your feedback and questions to:

Contact: Shohei Matsuura
Email: shmatsuu@yahoo-corp.jp
Yahoo Japan Corporation