Thread-specific Database Buffer Management in Multi-core NVM Storage Environments

Tsuyoshi Ozawa, Yuto Hayamizu, Kazuo Goda, Masaru Kitsuregawa (UTokyo)

Summary

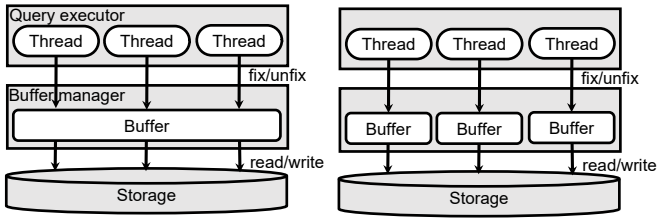
- Our preliminary experiment shows separated buffer mgmt. policy achieved *over 1M IOPS* on a real analytics database system

Introduction

- The emergence of high-performance NVM
  - 1M IOPS on single machine with NAND flash chip SSD / Optane

Database Buffer Management

- Buffer manager requires concurrency control
- Buffer management itself can be bottleneck



Baseline (shared) **Proposed (separated)**

Approach

- Eliminating concurrency control by separating buffers, targeting analytic queries

Experiment setup

- Workload
  - Modified TPC-H Query 3, scale factor 100
  - Buffer size: 1GB / Page size 16KB
- Buffer management policies
  - SHR (SHR-1, SHR-16, SHR-512) : shared buffer with 1/16/512 striped locks
  - SEP: separated buffer
  - All buffer managers use GCLOCK

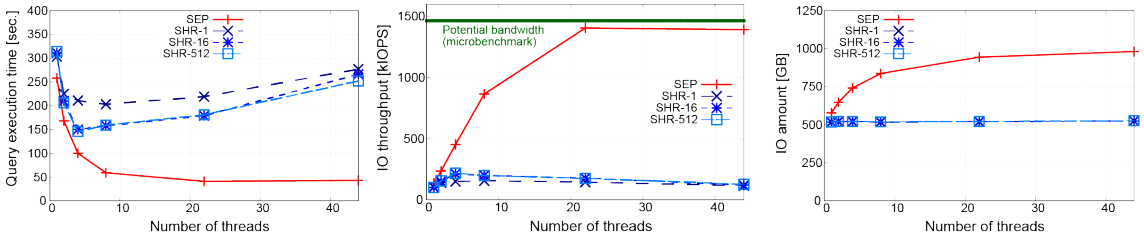
Machines for experiment

	NVMe flash machine	Optane machine
CPU	2x Xeon E5-2699 2.2GHz 22cores	2x Xeon E5-2699 2.2GHz 22cores
Memory	256GB	256GB
Storage	10x Intel DC P3700	10x Intel Optane DC 4800x

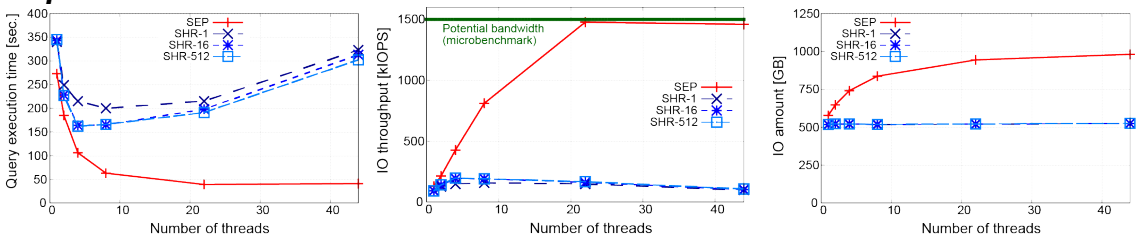
Experiment Results

- x6.37 faster query processing w/ SEP over SHR
- SEP achieves up to over 1.48 MIOPS (up to 98% device potential)

***NVMe flash* machine**



***Optane* machine**



Conclusion & Future work

- Our proposal (SEP) outperforms the baselines (SHR-*) to process a read-only query
- We will evaluate our systems with RW-mix queries and extend our study to explore lock-free techniques