DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Scalable buffer-pools for CPU-GPU DBMS architectures

Keywords: DBMS, Concurrent Programming, Data Structures, Indexes, Trie

Problem: Disk-based DBMS uses buffer pools to manage data sets larger than available main memory. Buffer pools become the central point of coordination and contention for data access and, therefore, scalability bottleneck in multi-socket, multi-core hardware. LeanStore [1] employs a transparent pointer-swizzling-based design for scaling buffer pools. However, pointer swizzling restricts the page to be referenced by a `single owning swip` to avoid inconsistency where two pages own a reference and cannot synchronize the page swizzling, thereby only allowing hierarchical page dereferencing.
In heterogeneous (e.g., CPU-GPU) DBMS architectures [2], data may be replicated or moved across non-coherent memory hierarchies. Therefore, DBMS requires data ownership and move semantics across non-coherent memory hierarchies while guaranteeing read/write consistency. Further, in HTAP workloads, snapshot pages may be shared across query sessions, requiring buffer pools to synchronize page translations across query sessions.

Project: In this project, we will research and develop a buffer pool for heterogeneous DBMS architectures. The student will familiarize himself/herself/themselves with existing buffer pool designs in the literature. The student will implement a LeanStore-based buffer pool as a baseline and then extend it for heterogenous architectures (CPU-GPU) and workloads (HTAP). Essentially, making buffer-pool hardware- and workload-aware and managing data ownerships/references across the system.

Plan:

1. Literature review and interface design
2. Implement a pointer-swizzling-based buffer pool in Proteus [2].
3. Integrate buffer pool with OLTP storage engine.
4. Performance analysis for in-memory and larger-than-memory data sizes.
5. Extend the buffer pool to support GPU execution.

Supervisor:	Prof. Anastasia Ailamaki, anastasia.ailamaki@epfl.ch
Responsible collaborator(s):	Hamish Nicholson hamish.nicholson@epfl.ch, Aunn Raza aunn.raza@epfl.ch
Duration:	6 months

References

[1] Leis, Viktor, et al. "LeanStore: In-memory data management beyond main memory." 2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 2018.

[2] Proteus. https://proteusdb.com