1. Name(s) of the author(s): **Christian Riegger, Tobias Vincon, Robert Gottstein, Ilia Petrov**
2. Title of article: **MV-PBT: Multi-Version Index for Large Datasets and HTAP Workloads**
3. Title of journal, volume number, date, month and page numbers: **Proceedings of the 23rd International Conference on Extending Database Technology, {EDBT} 2020, Copenhagen, Denmark, March 30 - April 02, 2020, pages 217 to 228**
4. Statement of the problem or issue discussed: **On the same dataset and system, modern mixed (HTAP) workloads perform fast update-transactions and long-running analytical queries. Such workloads result in many short-lived versions and long version chains in multi-version (MVCC) systems, as well as increased and frequent maintenance overhead. For starters, periodic changes necessitate the creation of new versions on a regular basis, resulting in an increase in index maintenance overhead. Second, index scans require additional I/O to determine which of the resulting tuple-versions are visible to the executing transaction (visibility-check), since current designs only store version/timestamp information in the base table, not the index.**
5. The author’s purpose, approach or method: **We propose the Multi-Version Partitioned BTree (MV-PBT) as a version-aware index structure for MV-DBMS in this paper. MV-PBT is based on Partitioned B-Trees, a variant of B+Trees. The MV-PBT index structure is a version-aware index structure. It keeps track of version numbers and allows for index-only visibility checks. When compared to LSM-Trees, MV-PBT supports append-based write behaviour and has much lower write-amplification.**
6. Primary (evaluation) result: **MV-PBT has been implemented in PostgreSQL. The performance evaluation under HTAP workloads (CH-Benchmark) shows a 2x improvement in analytical throughput due to index-only visibility-checks, while improving transactional throughput by 15% compared to PostgreSQL's highly optimised B+Tree. MV-PBT outperforms TPC-C by 15%.**

**MV-PBT has also been implemented in WiredTiger (MongoDB). The performance evaluation indicates that YSC provides approximately 40% higher throughput.**