OLxPBench

Real-time, Semantically Consistent, and Domain-specific Benchmark, Designing, and Implementing hybrid transactional analytical processing (HTAP) Systems

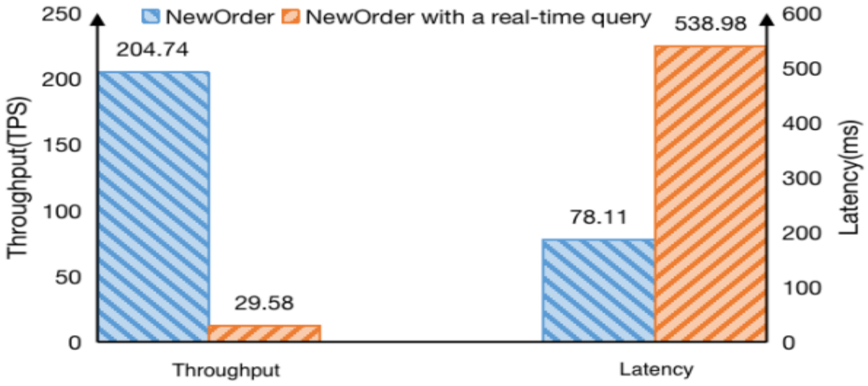
Name	عبدالعزيز صلاح محمد عبده
DR Name	سمير شاهين
Email	Abdelaziz132001@gmail.com
Paper Link	OLxPBench
Publishing date	9/5/2022

Preface

International Open Benchmark Council (BenchCouncil) is a non-profit international organization that aims to benchmark, evaluate, and standardize technologies like Big Data, AI, Chips, HPC, Metaverse, etc.

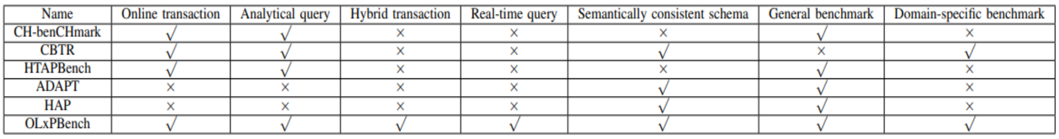
Since its founding, BenchCouncil bears four fundamental responsibilities: keep the benchmarks, evaluations, and standards community open, inclusive, and growing; define emerging or future challenges using benchmarks; promote benchmark-based quantitative approaches to tackle multidisciplinary and interdisciplinary challenges; establish unified benchmark science and engineering across multi-disciplines (https://www.sciencedirect.com/science/article/pii/S2772485922000515). You can visit them by clicking Here.

Introduction:



This paper mainly discusses the proliferation (The term "proliferation" refers to a rapid and widespread increase or growth in something) of hybrid transaction/analytical processing (HTAP) systems, highlighting the demand for fresher data in shorter durations for real-time customer analysis. It identifies three primary architectures for HTAP systems and emphasizes the importance of real-time analysis, semantically consistent schema, and domain-specific benchmarks in evaluating their performance. The text introduces an HTAP benchmark suite called OLxPBench, addressing the limitations of existing benchmarks and proposing new built-in hybrid workloads, a semantically consistent schema, and domain-specific benchmarks for sectors like retail, banking, and telecommunications. The paper also presents insights from experiments conducted on mainstream HTAP systems, MemSQL and TiDB, using OLxPBench, noting challenges such as poor performance isolation due to mutual interference between online transactions and analytical queries. HTAP stands for Hybrid Transaction/Analytical Processing. It refers to a type of database architecture that combines both transactional (OLTP - Online Transaction Processing) and analytical (OLAP - Online Analytical Processing) processing capabilities within a single system.

TABLE I
COMPARISON OF OLXPBENCH WITH STATE-OF-THE-ART AND STATE-OF-THE-PRACTICE BENCHMARKS.



Name	Online transaction	Analytical query	Hybrid transaction	Real-time query	Semantically consistent schema	General benchmark	Domain-specific benchmark
CH-benCHmark	√	√	×	×	×	√	×
CBTR	√	√	×	×	√	×	√
HTAPBench	√	√	×	×	×	√	×
ADAPT	×	×	×	×	√	√	×
HAP	×	×	×	×	√	√	×
OLxPBench	√	√	√	√	√	√	√

And this is a comparison with the state-of-the-art benchmarks showing how OLxPBench is better than all of them in metrics such as (Online Transactions, Analytical queries, etc…).

We need such benchmark because It is mandatory to include real-time queries in HTAP, semantically consistent, and domain specific in benchmarking.

Benchmark Methodology:

HTAP DBMSs need to perform trade-offs considering different performance requirements of different workloads. Currently, there are three types of HTAP solutions, and the paper compares their pros and cons in the following subsections.

1. Separate DBMSs:

The first solution involves using distinct Database Management Systems (DBMSs) for online transactions and analytical queries. Online transactions typically use a row-based store for efficient record insert and update, while analytical queries prefer a column-based data warehouse for efficient data scans. However, the need for ETL (Extract, Transform, Load) processing to convert row-based data to column-based data introduces time delays, hindering real-time analysis.

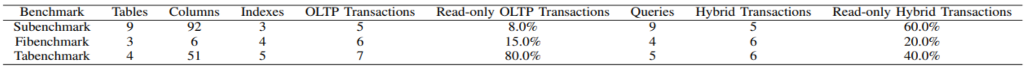
2. Lambda Architecture:

The second solution, known as the lambda architecture, combines real-time stream processing and batch processing systems. This approach allows for real-time analytics on incoming data but comes with high costs, including double write costs, increased development costs, and maintaining two separate systems.

3. Single STAP DBMS:

The third solution advocates for using a single HTAP DBMS to handle both online transactions and real-time queries. Various databases, such as Microsoft SQL Server, Oracle in-memory database, MemSQL, TiDB, IBM Db2 Analytics Accelerator, and VEGITO, implement strategies like lightweight propagation, dual-format storage, asynchronous log replication, and high availability mechanisms to support real-time queries on fresh data without the need for separate systems. This approach aims to provide a balance between performance and cost-effectiveness.

TABLE II
Features of the OLxPBench workloads.



Benchmark	Tables	Columns	Indexes	OLTP Transactions	Read-only OLTP Transactions	Queries	Hybrid Transactions	Read-only Hybrid Transactions
Subenchmark	9	92	3	5	8.0%	9	5	60.0%
Fibenchmark	3	6	4	6	15.0%	4	6	20.0%
Tabenchmark	4	51	5	7	80.0%	5	6	40.0%

OLxPBench consisting of a general benchmark (**subenchmark**) and two domain-specific benchmarks (**fibenchmark and tabenchmark**. OLxPBench incorporates nine built-in workloads, including *online transactions, analytical queries, and hybrid transactions, aiming at real-time customer analysis and simulating user behaviors*. The suite is configurable for various testing purposes, and this section details the schema model design, HTAP workloads, and integrity constraints considerations. Specific benchmarks (**subenchmark** and **fibenchmark**) are described, illustrating their inspiration, components, and characteristics.

1. **Su**benchmark:
   - It is inspired by TPC-C
   - Not bound specific benchmark.
   - Online workload is same as TPC-C (which is write heavy, and 8% read-only).
   - It includes 9 analytical queries, which keeps the essential characteristics such as complexity of operations.
2. **Fi**benchmark:
   - It aims the Bank scenarios.
   - It is a domain specific benchmark.
   - Contains three tables:
     - Account
     - Saving
     - Checking
   - Simulate most transactions occurring in banking.
3. Tabenchmark:
   - Related to telecom scenarios.
   - Inspired by TATP.
   - Domain specific.
   - Simulate a typical home location register (HLR) database used by mobile carrier.

Implementation:

OLxPBench is used for evaluating distributed HTAP DBMSs and other HTAP DBaaS systems that support SQL through Java Database Connectivity (JDBC). it is a Java-based API.

The Architecture used, is shown in the following diagram:

The architecture involves parsing configuration files at runtime to generate hybrid workloads, which are then placed in request queues. Configuration details such as request rates, transaction types, real-time query types, weights, and target DB configuration are specified in an XML file. Threads connect to the target database using a JDBC pool, pulling requests and measuring latency and throughput metrics. The statistics module aggregates these metrics and stores percentiles of latency in a user-specified file.

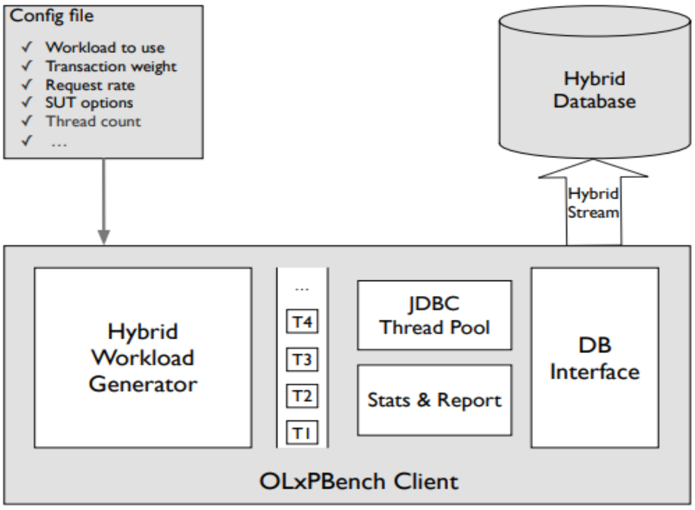


Fig. 2. OLxPBench architecture.

OLxPBench operates in open-loop and closed-loop modes, allowing precise request rate control and customizable weights for online transactions and analytical queries. Inspired by OLTP-Bench's OLTP module, OLxPBench includes additional analytic and hybrid modules. It supports three agent combination modes, including sequential and concurrent execution of online transactions and analytical queries, as well as sending hybrid transactions to simulate real-time user behavior. The OLxPBench client, a Java program, is easily extendable with new hybrid database back-ends.

Sample Results:

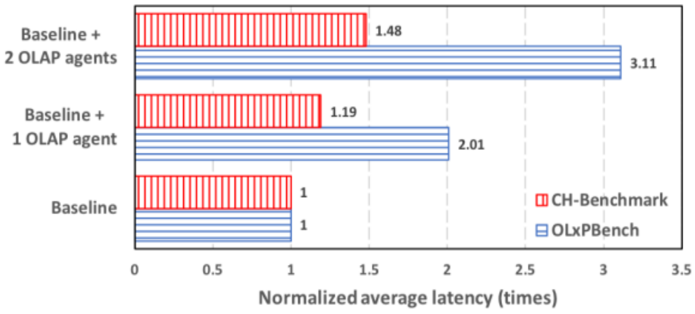
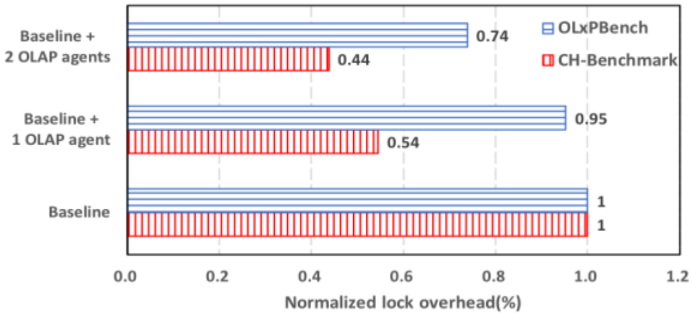
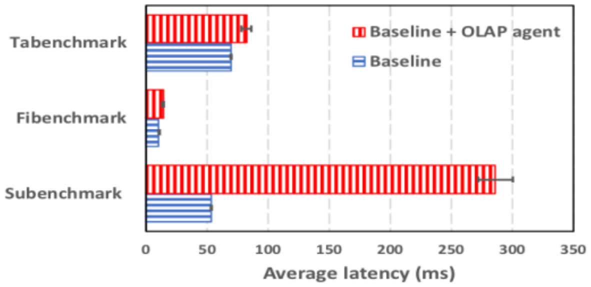
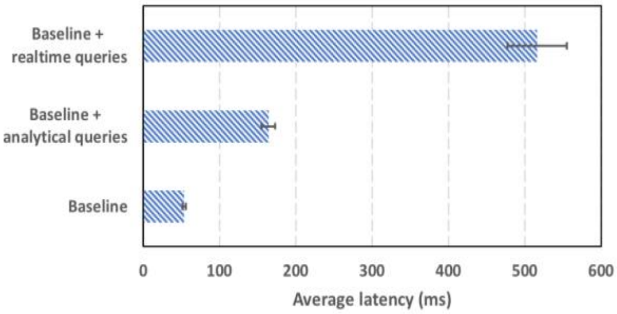


Fig. 3. Comparing the schema model of OLxPBench and CH-benCHmark on TiDB cluster.



Our evaluation illustrates the effectiveness of OLxPBench. In Section V-B, we compare OLxPBench with the state-of-the-practice work, testing the key features of OLxPBench and reporting the standard deviation of absolute value. From Section VI-A to Section VI-D, we evaluate the mainstream distributed HTAP DBMSs using OLxPBench and pinpoint the bottlenecks of two mainstream distributed HTAP DBMSs. In Section VI-E, we evaluate the scaling capability of TiDB, MemSQL, and OceanBase



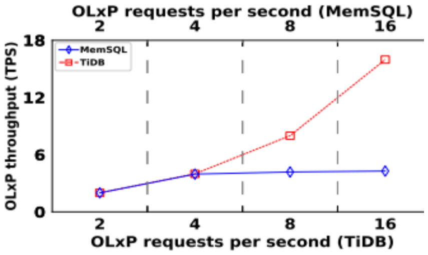


Here we can clearly see some statistics of the average latency of OLxP, and how it works faster than other benches such as OLAP and so on.

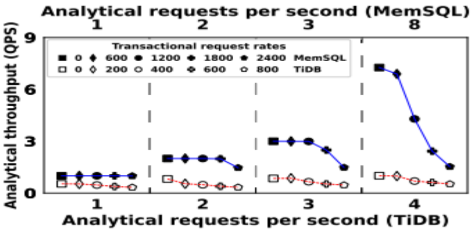
The subench is approximately getting 6X faster than the normal agent.

Also we can see that we are achieving an acceptable rate on the real time queries, as just need 500 ms which is half a second.

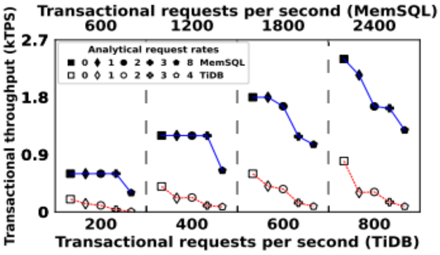
This is on applying different kind of pressures.



(c) Throughput of OLxP.



(b) Throughput of OLAP.



(a) Throughput of OLTP.

And in these graphs we show how the OLxP is performing in transactional requests per second against other benchs such as OLTP, and OLAP.

Conclusion:

This paper quantitatively discloses that the previous HTAP benchmarks provide misleading information in evaluating, designing, and implementing HTAP systems. We design and implement an extensible HTAP benchmarking framework named OLxPBench. OLxPBench proposes the abstraction of a hybrid transaction to model the widely-observed behavior pattern – making a quick decision while consulting real-time analysis; a semantically consistent schema to express the rela tionships between OLTP and OLAP schemas; the combination of domain-specific and general benchmarks to characterize diverse application scenarios with varying resource demands. Extensive experiments demonstrate its merit and pinpoint the bottlenecks of the mainstream distributed HTAP DBMSs.