Assignment 5. The design of an HTAP database

1 Overview

In this assignment, you are tasked with creating a design plan for an Hybrid Transactional and Analytical Processing (HTAP) database, custom-made to suit particular workloads and specifications. You will need to submit a **two-page** white paper (with reasonable formatting and spacing) which outlines a prototype design for this HTAP database. This design should reflect the knowledge and insights you have acquired over the course of this semester. In what follows, we provide the detailed instructions.

2 Design goals

You are tasked with the design of a relational DBMS, responsible for managing multiple relational tables. Generally, your focus will be on two primary types of workloads:

- **Rich join analytical queries**: One primary workload in this project involves analytical queries. These queries are read-only, i.e. they do not alter the underlying data; instead, they perform searches, and aggregations on the existing data. Furthermore, these queries are characterized by complex joins, often involving multi-way joins that span across numerous relational tables.
- **Non-uniform data updates**: In contrast to typical OLAP workloads, which usually do not involve data updates, here we consider the DBMS must also be able to efficiently manage continuous data updates and insertions. Especially, we consider
  - Some tables may only subject to insertion updates, that said every update to these tables is a insertion of a new record.
  - Updates typically occur within specific time frames, while during other periods, update workloads are relatively minimal. For example, in a 24-hour day, the majority of updates might occur between 10 am and 12 pm, and then again from 7 pm to 10 pm.
  - Even during the less active "cold" periods, there's a possibility of experiencing occasional bursts of updates being posted to the database.

Based on such workloads, your job is to design a DBMS that are optimized for two objectives:

1. The DBMS should be optimized for maximum efficiency in handling query workloads. You will need to explore all possible optimizations to enhance query efficiency. This involves improving both query latency, which is the elapsed time for processing a single query, and throughput, the number of queries processed within a given time frame.
2. The DBMS should strive to balance answering queries with the most recent data and the costs associated with applying updates to the underlying data. For example, if the DBMS is configured to update every time new data arrives, queries will always be answered with the most current information, but at a significant cost.

3 General requirements

In general, your design should aim to encompass the following aspects in your white paper.

Storage (30 pts). Your white paper should detail your approach to data storage design, including the selection of a storage model. This involves deciding between row-based or columnar storage, and considering whether to use a slotted table format or a log-structured file system, etc.

Update management (15 pts). You should discuss approaches on how to manage data updates, for instance, are you planning to make an update every time a request arrives, or do you plan to cache and update in a batch?

Query planning & optimization (30pts). Your white paper should also elaborate on optimizing query execution. This includes strategies for efficiently determining the optimal join order. Given the complexity of the rich join queries, it's crucial to consider how to reduce the search space and minimize the time spent on query planning. You should also elaborate on the specific statistics required and how to select the most effective statistics to aid in query optimization, particularly if you are limited to generating statistics for only a select number of attributes.

Query processing & parallelism (15 pts). Your white paper should also discuss ways on leveraging query parallelism for accelerating query execution.

Additional optimizations (10 pts). You should also discuss any possible additional optimizations that can assist with query processing, storage management, or query planning. For instance, consider how to conduct result caching, such as using materialized views.

4 Grading

The grading will highly depend on the quality of your white paper writing. While we do not have "expected outputs" for each design objective, any reasonable design can receive full credits. To get full credits, you should

1. Address all design objectives mentioned in Section 3.
2. You should not only provide your design but also justify **why this design is reasonable**.
3. Clear and concise technical writing is essential.
4. Include necessary references to support your statements (refs also counts into total page limits).
5. **LLM policy.** In this assignment, we will enforce a strict policy regarding the use of generative AI tools and large language models (LLMs). While the use of LLMs is permitted for auxiliary purposes, such as polishing sentences or refining writing, their use to generate large paragraphs, entire designs, or complete white papers is prohibited. You are required to document how you utilized LLMs in completing Assignment 5. To ensure compliance, we will use tools like GPTZero to validate your submissions. If AI-generated content exceeds 20% of your assignment or includes substantial generated sections, you will receive a score of 0 for Assignment 5.