Concurrent Data Systems for Agents - End of Sem Update

Matthew Nelson, Nikhil Ghosh, Peter McNeeley COMS6998 - Topics in Cloud Computing (F'24)

Problem Statement

In agentic scenarios, there are often N many commands requested to run on some shared state, where only one should "win" and commit to the DB.

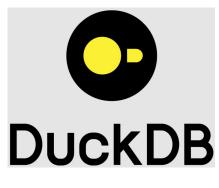
Project Idea

Performance benchmarks for serial/parallel handling of simultaneous transactions (Postgres, DuckDB), benchmarking on NeonDB using ntran.

Systems Evaluated



Serial transactions



Serial and parallel transactions



Parallel transactions

ntran System

```
ntran/
main.go
                 (Entry point, handles CLI)
policy/
                   Policy Interface
                 (Defines interface for all implementations)
    policy.go
     Implementations
                             (PostgreSQL)
         serialclient.go
         duckdbparallelclient.go (DuckDB)
                                              Share common test cases
         duckdbserialclient.go (DuckDB)
                                              and benchmarking logic

 coldneondbclient.go (NeonDB)

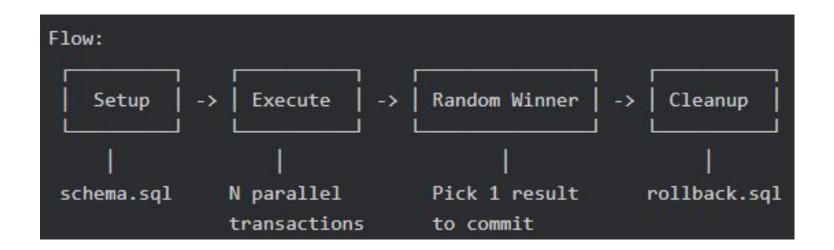
 prewarmneondbclient.go (NeonDB)

                  (Shared SQL test cases)
    queries.go
    benchmark.go (Performance measurement)
     experiment.go (Results collection)
schemas/
                 (Database schemas)
                   (Initial setup)
   - schema.sql

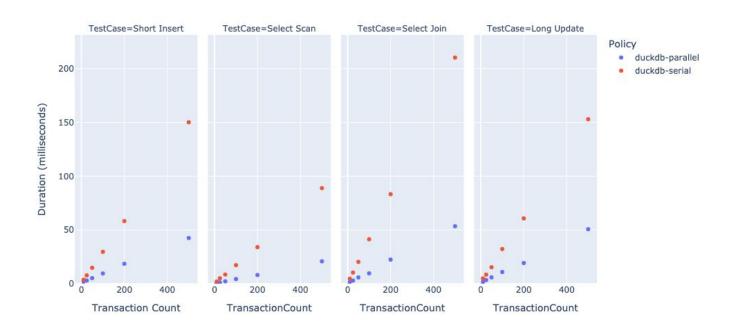
 rollback.sql (Cleanup)

results/
                 (Benchmark outputs)
 - *.csv
                 (Raw timing data)
```

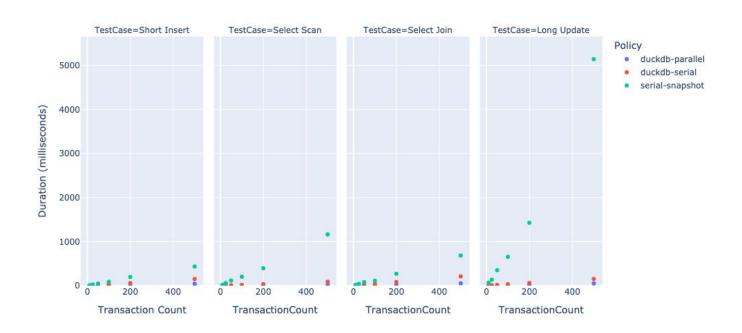
ntran Workflow



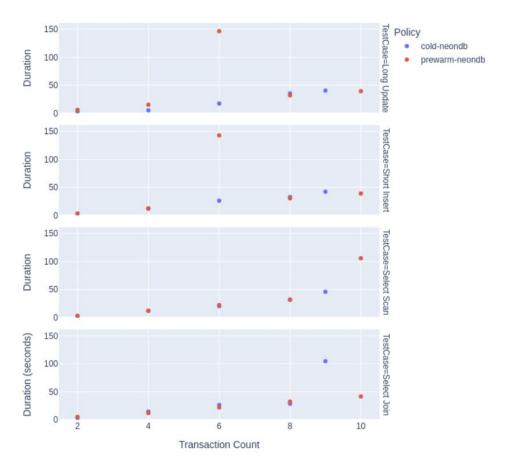
Results - DuckDB



Results - DuckDB vs pSQL (Serial)



Results - NeonDB



Key Findings + Limitations

- Findings
 - DuckDB COW issue within the driver, elaborate/reword, requires DuckDB driver update
 - Neon concurrency limit (Free tier)
 - Neon doesn't yet allow managing branches in parallel
- Artifacts
 - Benchmark results
 - ntran
- Future facing
 - Run Neon experiments with N>10
 - Further standardize testing environment