



Schema Change

Final Presentation

Group 8: Shangjie Chen, Shaoxiong Zhang, Shilun Ding
May. 14. 2018



Motivation

Add support for ALTER TABLE operations



Project Goal

- 75%: Drop column and change column name
- 100%: Add column and change type
- 125%: Alter Table Benchmark



Project Goal

- 75%: Drop column and change column name ✓
- 100%: Add column and change type ✓
- 125%: Alter Table Benchmark (Compared with Postgres & MySQL) ✓



Implementation

- Rename Column
 - Update COLUMN_NAME in pg_attributes table
- Alter Table
 - Collect column changes in Parser
 - Construct new Schema object in Planner
 - Verification and locking in Executor
 - Actual work in Catalog



Implementation

Catalog

- Build empty table with new Schema
- Copy indexes
- Build column mapping between old table and new table
- Copy all the tuples from old table using SeqScan and InsertPlan
- Update pg_attributes table (delete and re-insert)
- Replace table and do garbage collection



Testing

Junit Test

- Functionality Test
 - Rename, Add/Drop columns, Change column type.
- Multi-Transaction Test
 - Concurrent update, Read before commit, etc
- Benchmark Test
 - Postgres, on-fly schema changes
 - MySQL, blocking schema changes



Issues

- @1356: Binder Infinite recursive calls of GetColumnObjects if the column tuple in pg_attribute is being modified and not committed
- E.g.
 - T1 Starts
 - T2 Starts
 - T1 alter table
 - T2 select *(infinite loop)

Currently addressed by pr @1327



Code Quality Assessment

- Code Style
- Comments and Documentation
- Logging
- Unit Test
- Strong part
 - Correctness & Consistency, we do the blocking way
- Weak part
 - performance



Benchmark of alter table

- Test alter table from different perspectives
 - 1. Tuple variance
 - 2. Data type variance
 - 3. Tuple number variance
- Currently workload only compares running time of alter table under single thread alter table operation, like database race.
- Type change only **compares inline -> not inline** and **not inline -> inline** and we take the average of them



Environment

macOS High Sierra

Version 10.13.2

MacBook Pro (Retina, 13-inch, Early 2015)

Processor 2.7 GHz Intel Core i5

Memory 8 GB 1867 MHz DDR3

Startup Disk Macintosh HD

Graphics Intel Iris Graphics 6100 1536 MB

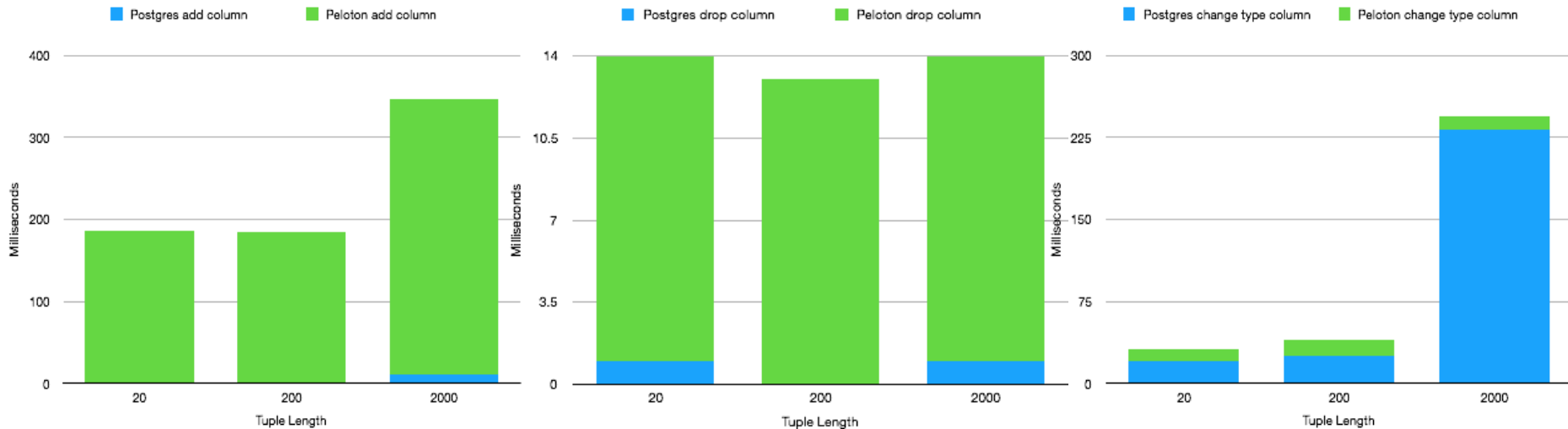
Serial Number C02QWBTFVH5

Peloton/Postgres/MySQL stays default settings.

Benchmark tries to keep things in memory by select * before actual running.

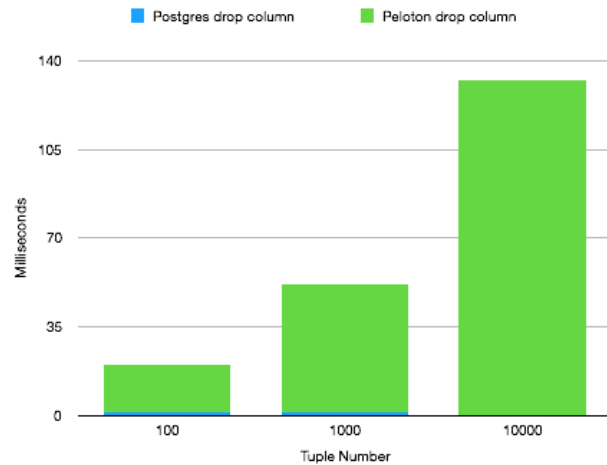
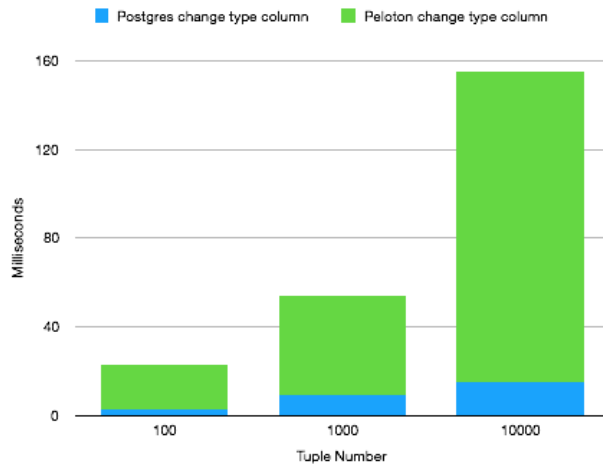
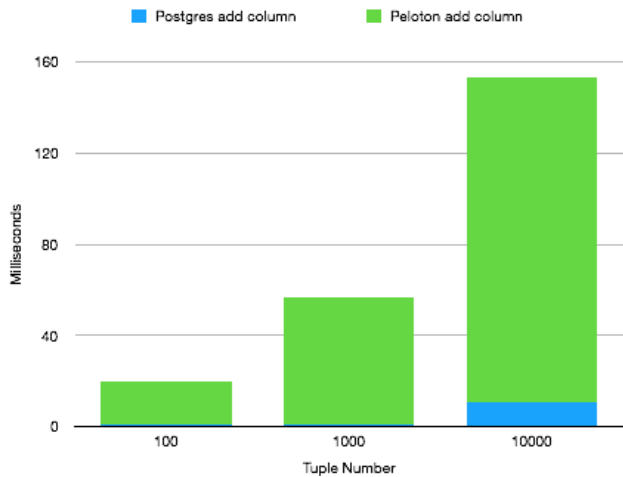


Different Tuple Length





Different Tuple Number





DEMO



Future Work

- Performance Optimization
- Foreign Key constraints
- Thorough multi-transaction test
- Merge!



Thank you!