Sequences

15721 Final Project Presentation

Bowen Deng[bowend1]
Peixin Sun[peixins]
Kangyan Zhou[kangyanz]

Goals

- 75% & 100%: Implement Create/Drop Sequence,
  Select nextval(‘seq’)/currval(‘seq’)
  - Session-specific behaviors

- 125%: Implementing SERIAL in peloton requires too many changes
  - Using function as default values is not fully supported in peloton

Review of Sequence Behaviors in Postgres 10

- Sequence name must be unique within a database
- No rollback for an aborted transaction
  - nextval is always incrementing
- Cannot create sequences with invalid arguments
  - e.g MINVALUE > MAXVALUE
- Progress in nextval is shared across sessions
  - Terminal 1: nextval -> 2, Terminal 2: nextval -> 3, Terminal 1: nextval -> 4
- In contrast, currval is session-specific
  - Terminal 1: currval -> 2, Terminal 2: nextval -> 3, Terminal 1: currval -> 2

Demo

- CREATE SEQUENCE seq;
- SELECT nextval(‘seq’);
- SELECT currval(‘seq’);
- DROP SEQUENCE seq;

Implementation Details

- Overview of Changes
  - Parser, Create/Drop Statement/Plan/Executor
  - SequenceCatalog, SequenceCatalogObject
  - SequenceFunctions(similar in project 1)
  - Put Sequence into CatalogCache(per txn)

Test

- Unit Test
  - Parser test
  - Nextval, currval functionality test
  - CREATE SEQUENCE argument validation test
  - DROP Sequence

- Integration Test
  - JDBC test

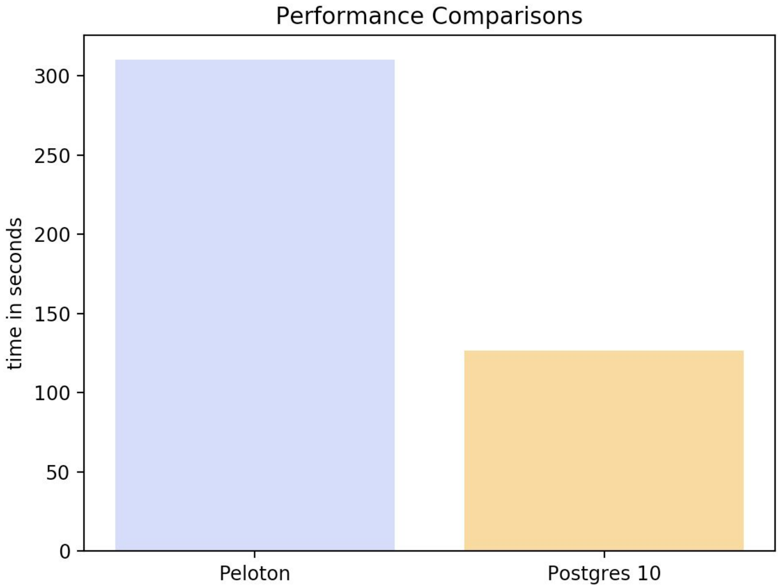
Code Quality

- Strong
  - Sequence cache, mini txn
  - Everything else
- Weak
  - Pass ExecutorContext to SequenceFunction
    - Stateless vs Stateful: Need the txnContext, database name
  - Session-binding for single-stmt transaction
    - Cache in SequenceCatalog

Benchmark

- We are comparing Peloton with Posgres 10
- For each trial, we ran nextval() for 5,000,000 times and timed it
- We ran 10 trials and computed the average

Result & Analysis



Postgres 10 runs 2.5 times speedup

The use of cache in Postgres 10 boosts the performance a lot !!!

Future Work

- A better way to deal with session binding
  - Create temp tables during each session
- How to cache sequence nextval results in in-memory databases
  - Create version chain without writing timestamp beforehand
  - Claim the empty version with CAS during nextval call

Acknowledgement

- Andy for providing suggestions
- Temp table team for providing useful reviews and the session binding