Pluggable Storage 成果紹介

Pluggable Storage概要

- Pluggable Storageとは
  - PostgreSQL 12で導入された機能である。
  - BufferManager層とのやり取りを行う新しいレイヤとして抽象化されたテーブルアクセスメソッド(以降、AM)が導入された。

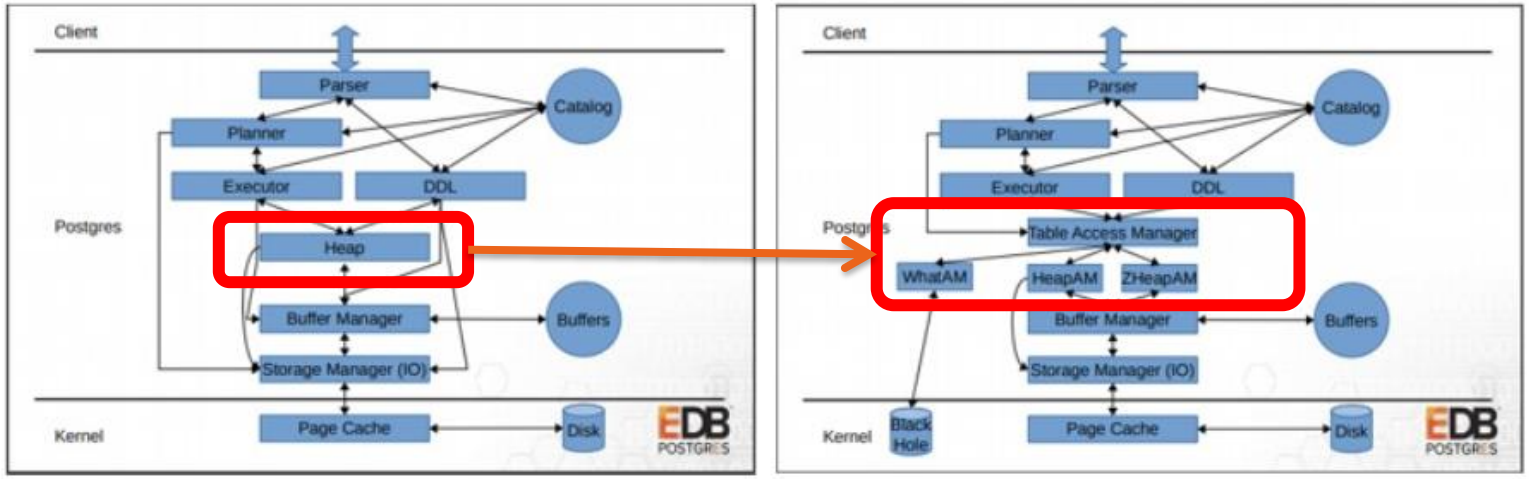


図 3.1 Route to access table data on Disk (Left: PostgreSQL 11 Right: PostgreSQL 12)

https://anarazel.de/talks/2018-10-25-pgconfeu-pluggable-storage/pluggable.pdf

システムのワークロードに合わせて、
テーブル毎に最適な方式を選択できる！

検証概要

- 目的
  - 開発中のAMによる効果の確認
    - zheap(UNDOログ形式)
      - テーブル膨張の抑止効果の確認
    - zedstore(カラムナ形式)
      - OLAP系の性能改善効果の確認
- 検証内容
  - 従来の形式であるheapAMとの性能比較
    - 参照性能、更新性能、各ベンチマーク(pgbench, SSB)
  - 設計、運用への影響確認(機能検証)

検証結果サマリ

heapAMとの性能比較結果

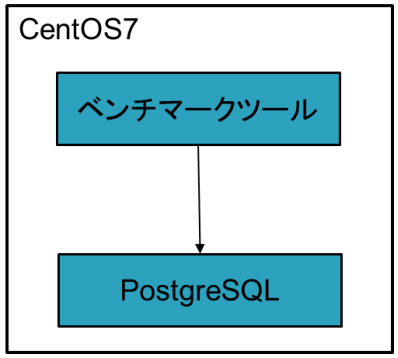
測定項目	zheap	zedstore
単一行/複数行に対する参照/更新処理	• 範囲検索：heapが高速 • 範囲更新：zheapが高速	• 想定通り更新系の処理は低速となった
特性に関する検証	• 肥大化の防止の効果を確認した • ヘッダ情報縮小によるテーブルサイズの圧縮を確認した	• 参照で発生するブロック数次第でheapAMが有利となるケースを確認した • OLAP系ベンチマークはheapAMの5倍の性能

機能検証結果

	zheap	zedstore
メンテナンス（VACUUM / ANALYZE / CLUSTER / REINDEX）	○（実行可能）	○
バックアップ/リカバリ	○	○
ストリーミング・レプリケーション	○	○
ロジカルレプリケーション	○	○
宣言的パーティショニング	○	○
パラレルクエリ	○	○
FILLFACTOR	○	○
追加モジュール（pgstattuple）	×（実行不可）	×

検証方法と環境

- 検証環境
  - クライアント、DBサーバ混在の1台構成



対象	OS	メモリ	CPU	ディスク
ゲストマシン	CentOS 7（Virtualbox）	3GB	3	SSD

  - Access Methodのバージョン
    - 2019年12月時点での最新版に対して評価を実施

Access Method	commit id
zheap	629b1a802abede64dfb2cb27f8210cd9e78ea4ae
zedstore	554cc36b025f7dc04d6e784555422caadf80aa27

  - PostgreSQLの設定
    - shared_buffers、checkpoint関連バラメータを中心にチューニング済み

検証結果（zheap）

- 単一レコードに対するheapAMとの比較（左：参照　右：更新）

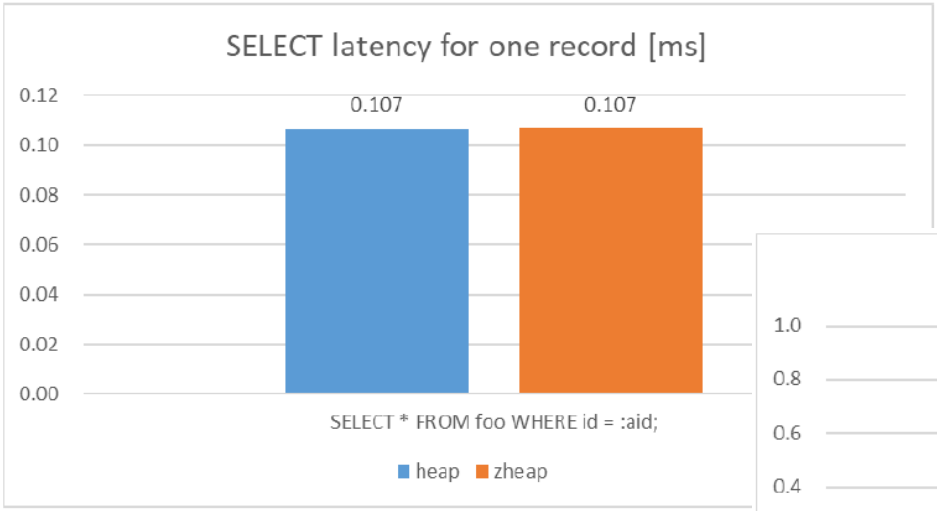


図 3.3 Latency of SELECT query (1 record)

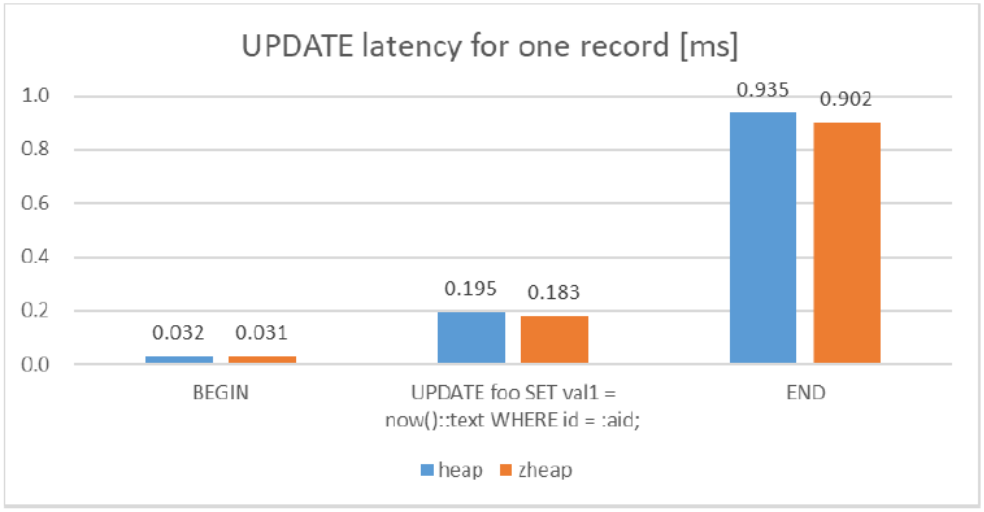


図 3.5 Latency of UPDATE query (1 record)

単一レコードに対する操作では明確な違いは見られなかった。

検証結果（zheap）

- 複数レコードに対するheapAMとの比較（左：参照　右：更新）

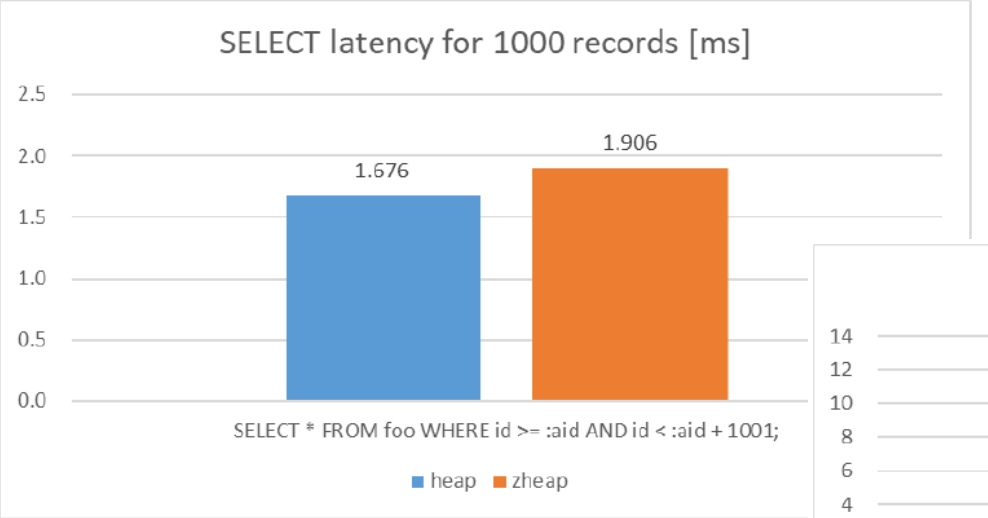


図 3.4 Latency of SELECT query (1000 records)

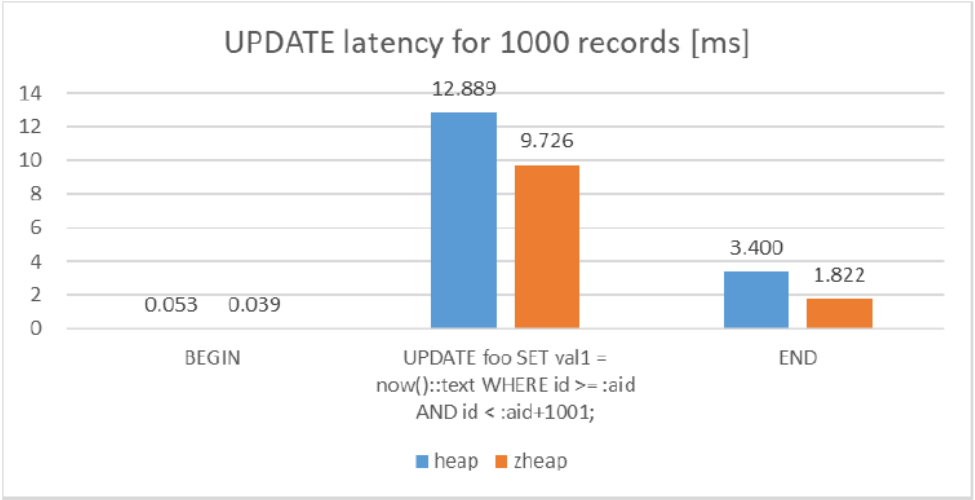


図 3.6 Latency of UPDATE query (1000 records)

範囲検索ではheapAMが高速であり、範囲更新ではzheapが高速であることを確認した。

検証結果（zheap）

- pgbench実行後のテーブルサイズ比較
  - 肥大化を発生しやすくするため、HOT機能を無効化した環境で実施

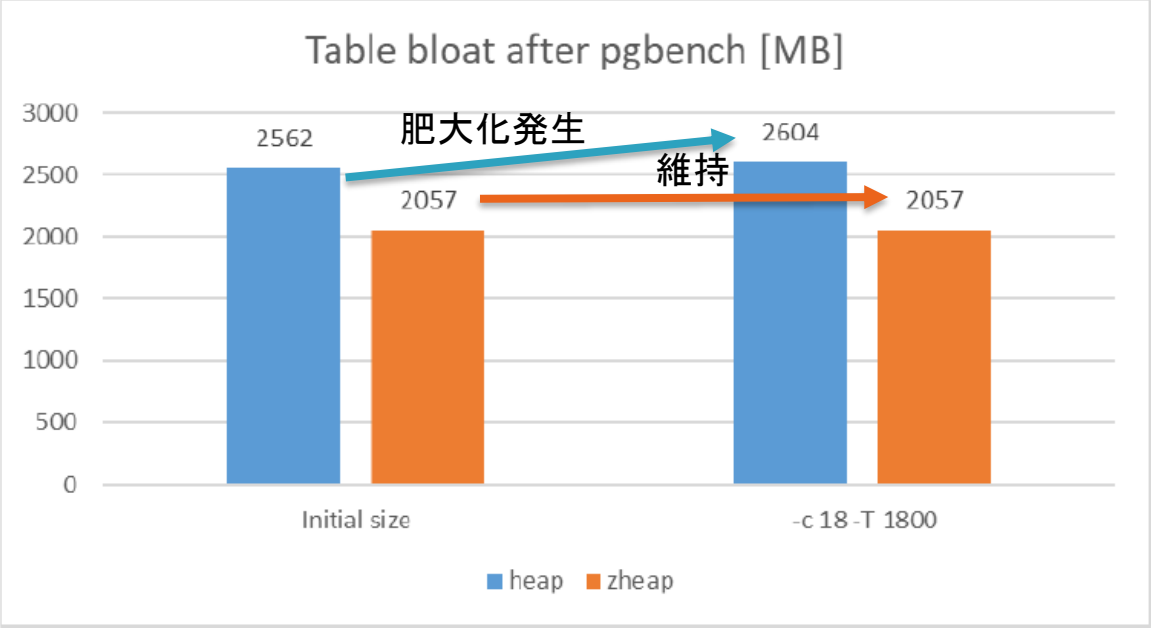


図 3.9 Table bloat size comparison

zheapではテーブルサイズの肥大化は発生しなかった。

検証結果(zedstore)

- 参照性能(単一カラムへのシーケンシャルスキャン)
  - テーブル定義:(id int, val1 text ,val2 text, ... , val20 text)

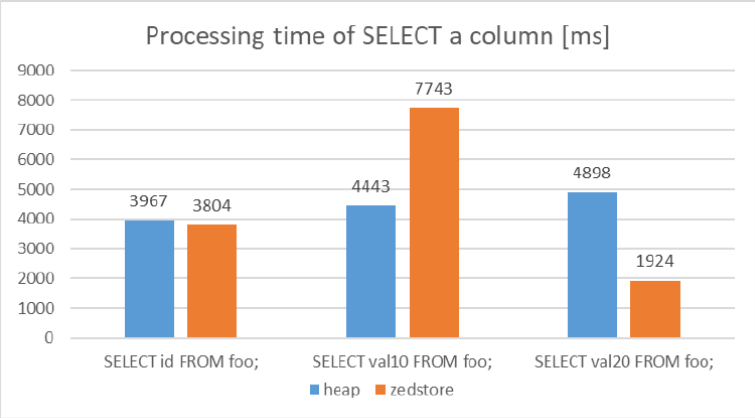


図 3.13 actual time of SELECT query

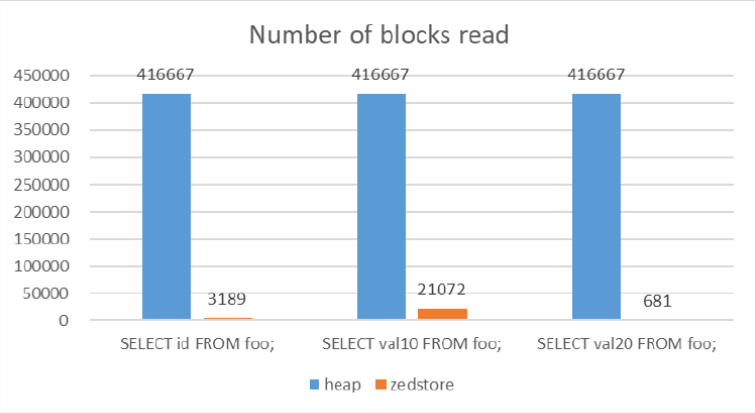


図 3.14 Number of read blocks

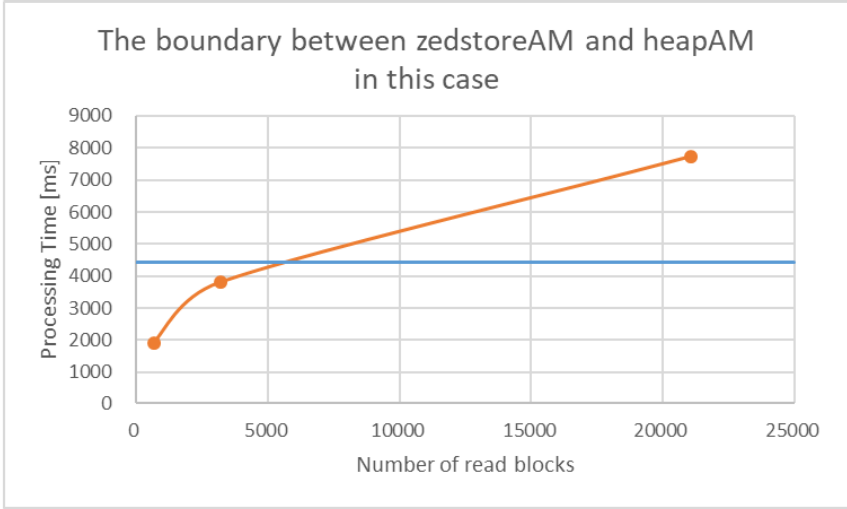


図 3.15 Performance boundary value (read block)

アクセスブロック数が少なければ、
zedstoreのほうが高速となる

アクセスブロック数がある程度多くなると
heapAMのほうが高速となる

検証結果(zedstore)

- 更新性能(単一行、範囲)

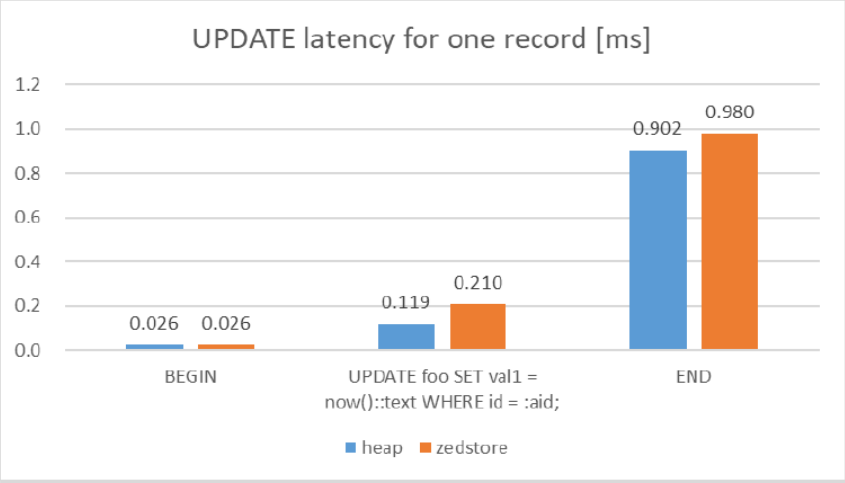


図 3.16 Latency of UPDATE query

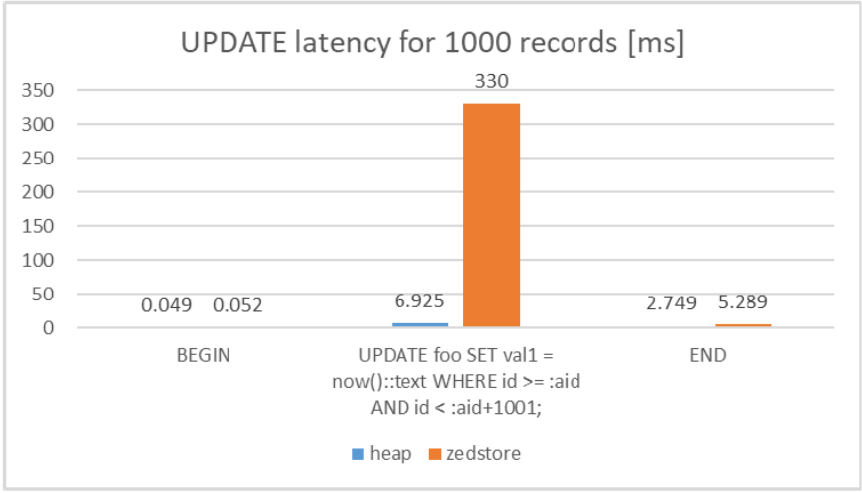


図 3.17 Latency of UPDATE query

カラムナの構造であるため、更新処理はheapAMと比べて、大幅に低速であることを確認できた。

検証結果(zedstore)

- OLAP (SSB)

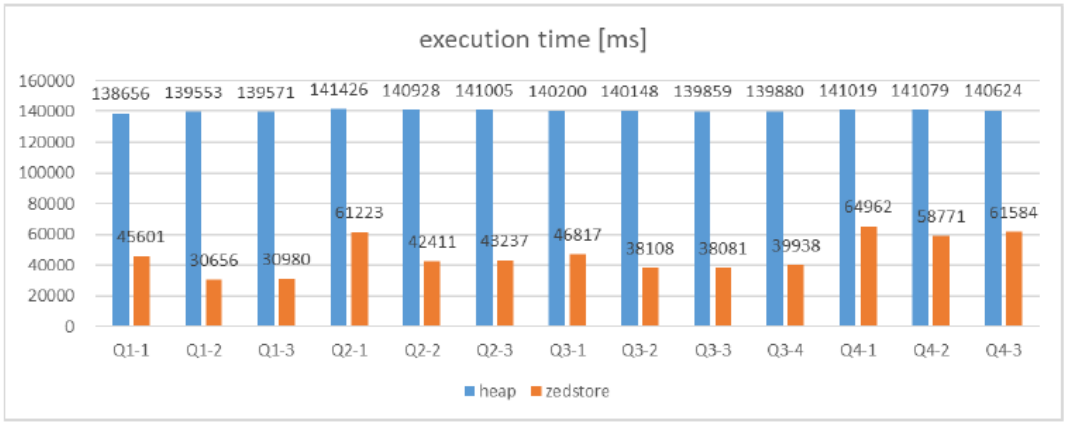


図 3.19 Execution time on Star Schema Benchmark

- テーブルサイズ

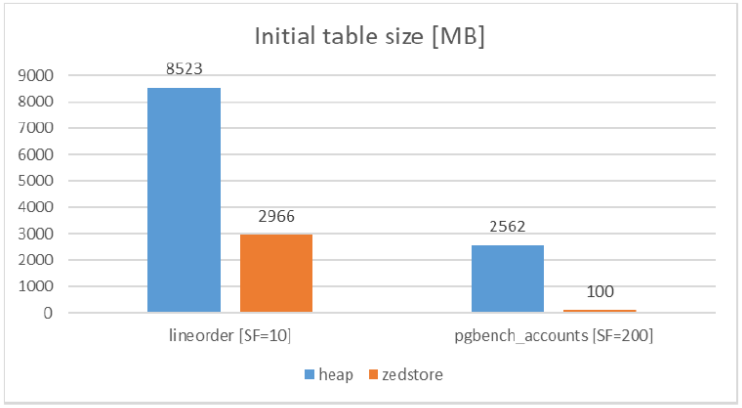


図 3.20 Table size comparison

OLAP系ベンチマークでは最大でheapAMの5倍程高速化された
圧縮効果により、データ次第では大幅にテーブルサイズが圧縮された

まとめ

- zheap
  - 肥大化を発生しやすい状況においても、肥大化が発生しないことを確認でき、性能についても明確な遅延がないことを確認できた。
    - ただし、ロールバック数の制限とか、ロールバックの性能に関しては見れていないので、その周りについても検証が必要。
- zedstore
  - カラムナ型として期待する結果を確認できた。
    - 読み込みブロック数によって、heapに比べて遅くなるため、ワークロードの見極めが重要となることをが確認できた。
    - OLAP系ベンチマークSSBではheapに比べて最大5倍程度高速化することが確認できた。
- 本検証は開発中のAMに対して実施したものであり、正式版のリリース時には今回の結果と異なる傾向が出る可能性がある。
  - UNDOログについてはコミュニティにおいて構造について再度議論が行われており、動向は継続して確認が必要である。

PGECons

PostgreSQL Enterprise Consortium