



PGECons
PostgreSQL Enterprise Consortium

2021年度WG1活動報告 定点観測（バージョン間性能比較）

**PostgreSQLエンタープライズ・コンソーシアム
WG1（新技術検証WG）**

検証概要

■ 目的

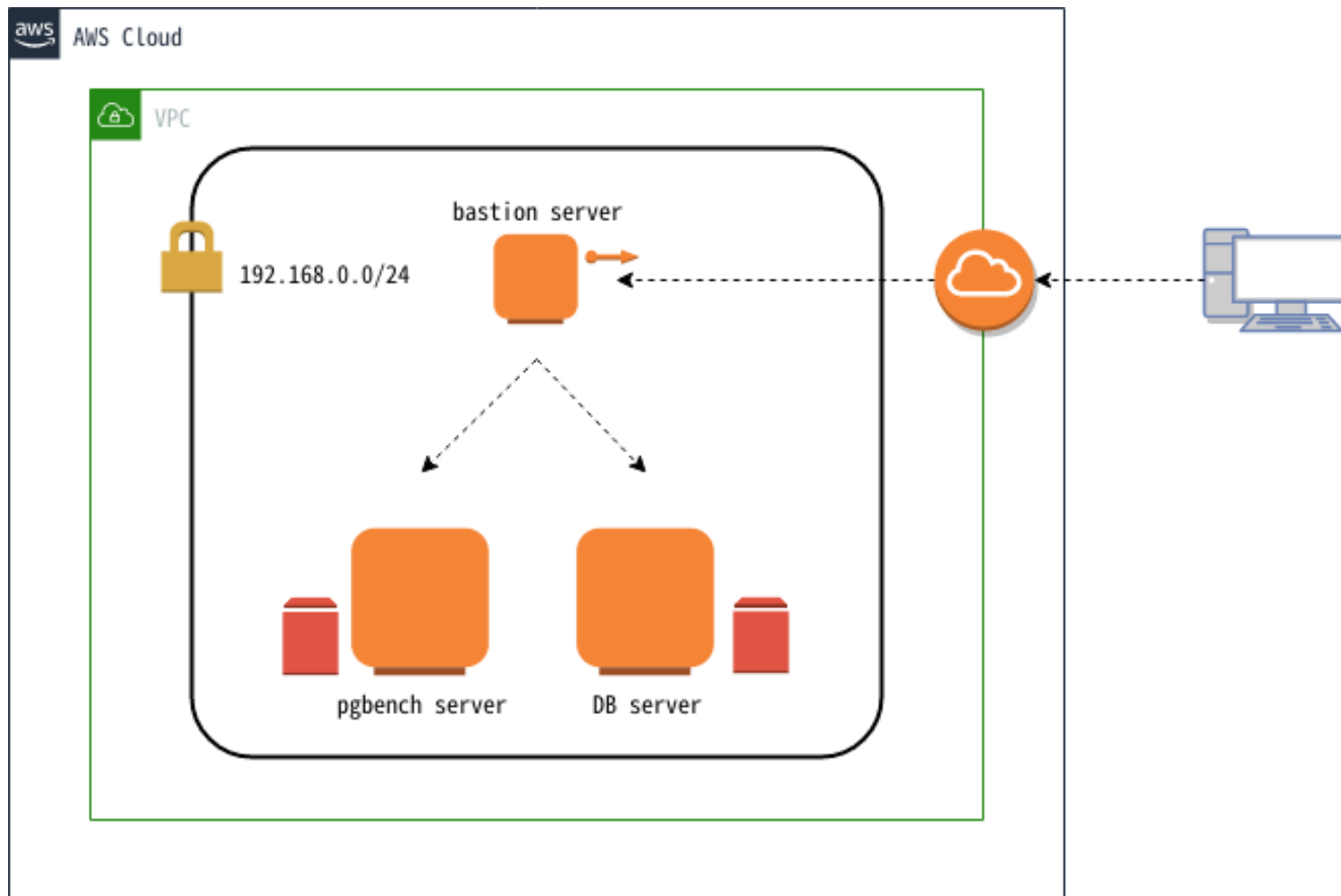
- メニーコアCPU上でのPostgreSQLのスケールビリティを検証
- 新バージョンのPostgreSQLの性能改善傾向を知る
 - PGECons発足当初(2012年度、PostgreSQL 9.2)から継続的に実施(定点観測)
 - 更新系性能に関する定点観測は2014年度から開始
 - バージョン間で性能差が現れたときはその要因も検証
- 今年度はV13とV14の比較を実施

■ 検証内容

- 参照性能
- 更新性能

検証手続き > 環境

- 今年度はAmazon Web Services (AWS) の仮想マシンを使用



検証手続き > 環境

■ インスタンススペック

名称	インスタンスタイプ	vCPU	メモリ (GiB)	ルートストレージサイズ (GiB) / IOPS	追加ストレージサイズ (GiB) / IOPS
bastion server	t2.micro	1	1	10/100	N/A
pgbench server	m5a.8xlarge	32	128	20/100	20/100
DB server	m5a.8xlarge	32	128	20/100	200/600

- 前年度以前と同等のCPUコア数
- メモリは試験用データが載るサイズを確保
- ストレージはonキャッシュで試験を実行するので最低限

■ ソフトウェア

名称	OS	PostgreSQL	pgbench
bastion server	RHEL 7.9		
pgbench server	RHEL 7.9		14.0
DB server	RHEL 7.9	14.0, 13.4	

- セットアップはAnsibleで自動化

検証手続き > 環境

■ 検証データベースサイズ

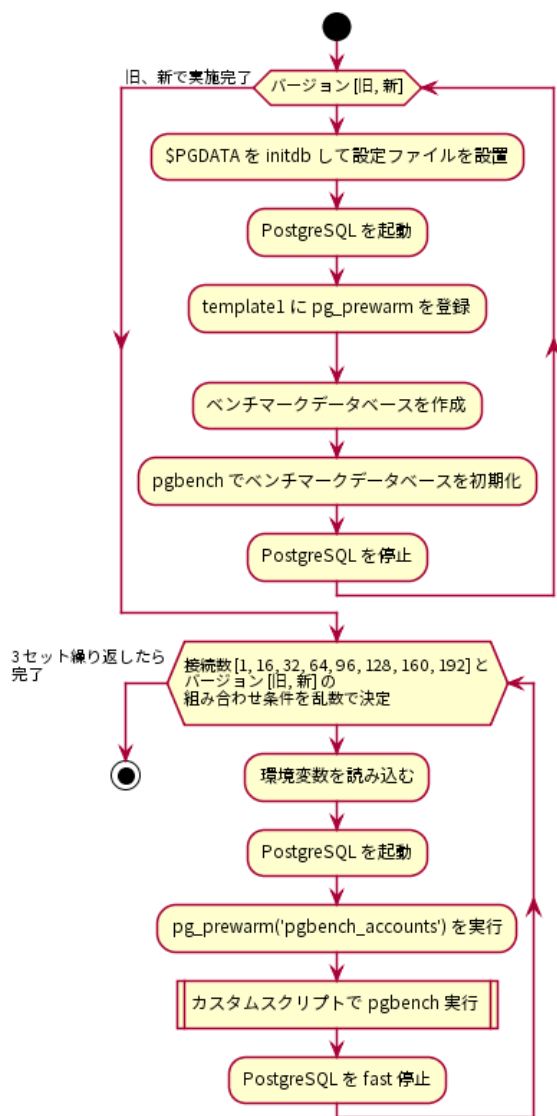
```
$ pgbench -i -s 2000 《データベース名》 -F 80
```

- pgbenchで30GBほどの試験用データベースを作成
- 更新系試験のみフィルファクタ80をオプション指定

■ postgresql.conf

```
listen_addresses = '*' # クライアント用サーバからの接続用
max_connections = 500 # 多めに設定
shared_buffers = 40GB # 試験用データがすべてメモリに載るように設定
work_mem = 1GB
maintenance_work_mem = 20GB
checkpoint_timeout = 60min # 試験中にチェックポイントを発生させない
max_wal_size = 160GB # 試験中にチェックポイントを発生させない
logging_collector = on
log_checkpoints = on
log_lock_waits = on
autovacuum = off # 試験中に I/O 処理を発生させない
```

検証手続き > 参照系

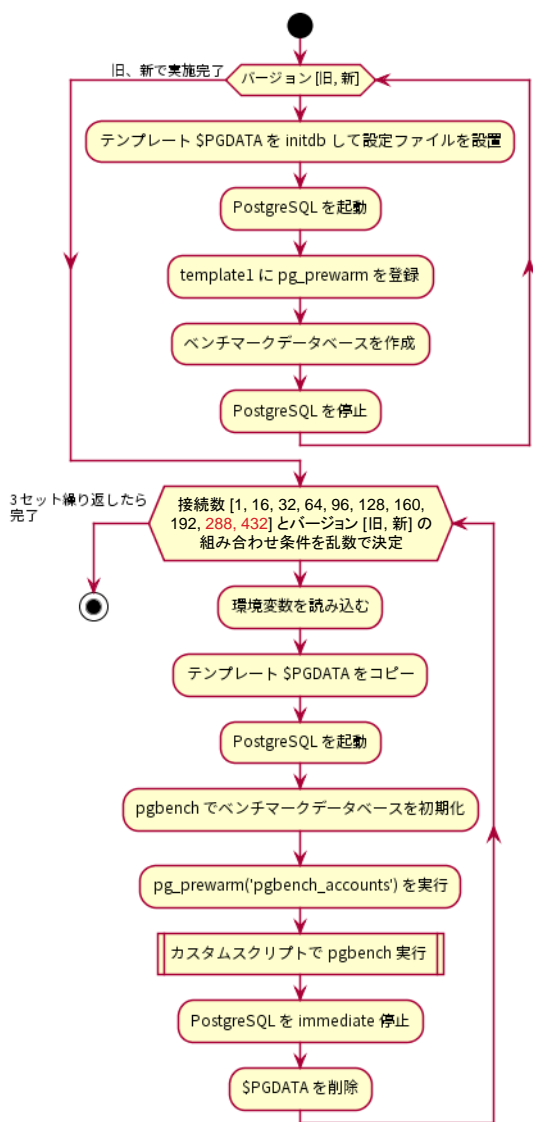


- PostgreSQLバージョン(2水準)とクライアント同時接続数(8水準)がTPSに及ぼす影響を計測
- $2 \times 8 = 16$ 種類のベンチマークをランダムな順序で3回繰り返す
- 得られたTPSの中央値を結果として採用

ベンチマークシナリオ

```
¥set naccounts 100000 * :scale
¥set row_count 10000
¥set aid_max :naccounts - :row_count
¥set aid random(1 :aid_max)
SELECT count(abalance) FROM pgbench_accounts
WHERE aid BETWEEN :aid and :aid + :row_count;
```

検証手続き > 更新系



- 試行の都度\$PGDATAを作り直す
(それ以外は参照系と同様)
- PostgreSQLバージョン(2水準)とクライアント同時接続数(10水準)がTPSに及ぼす影響を計測
 - 2020年度、192接続で頭打ちにならなかったため、288、432接続で計測
- 2×10=20種類のベンチマークをランダムな順序で3回繰り返す
- 得られたTPSの中央値を結果として採用

ベンチマークシナリオ

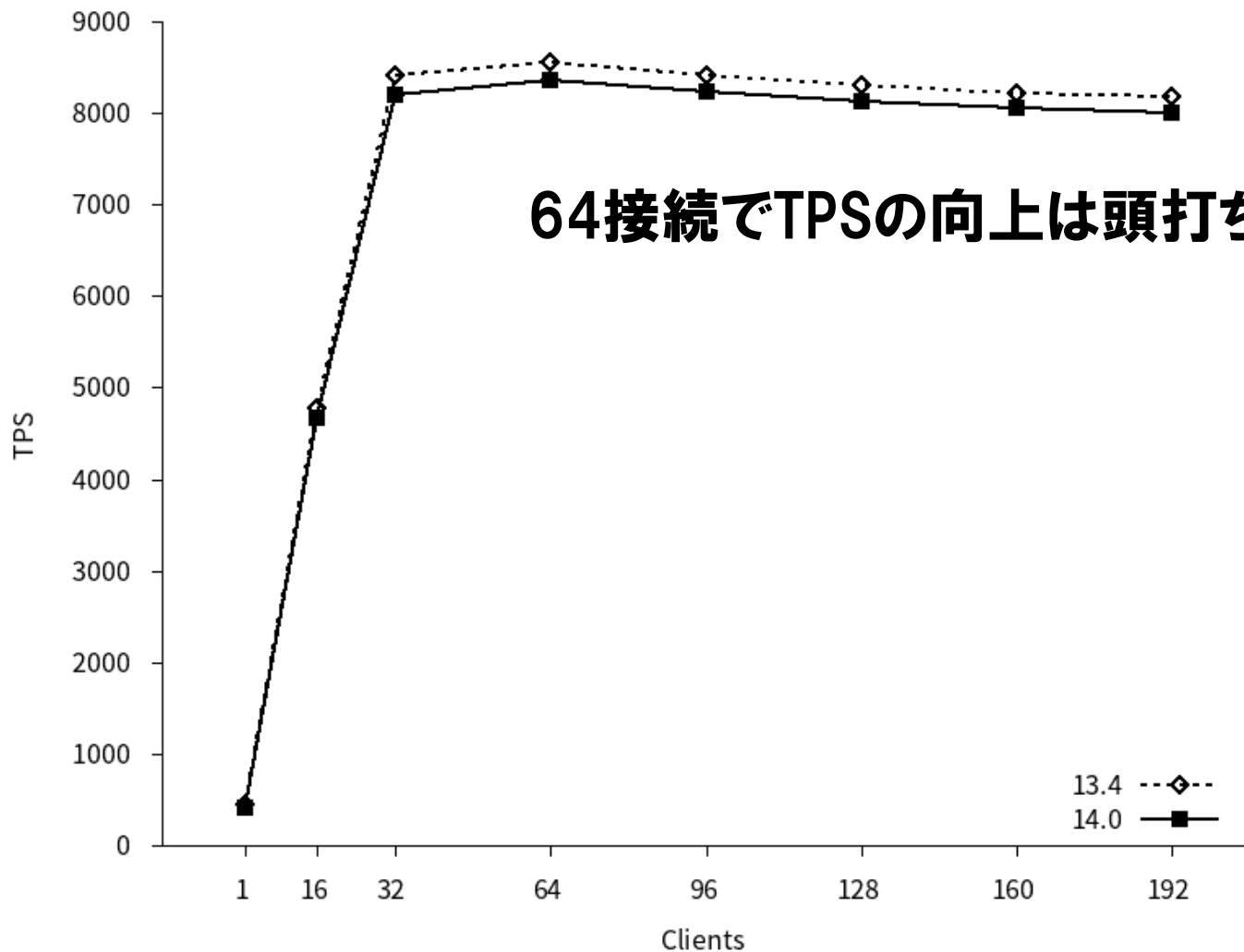
```
¥set naccounts 100000 * :scale
¥set aid_val random(1, :naccounts)
UPDATE pgbench_accounts
SET filler=repeat(md5(current_timestamp::text),2)
WHERE aid = :aid_val;
```

検証結果

■ 参照系

2%程度性能低下

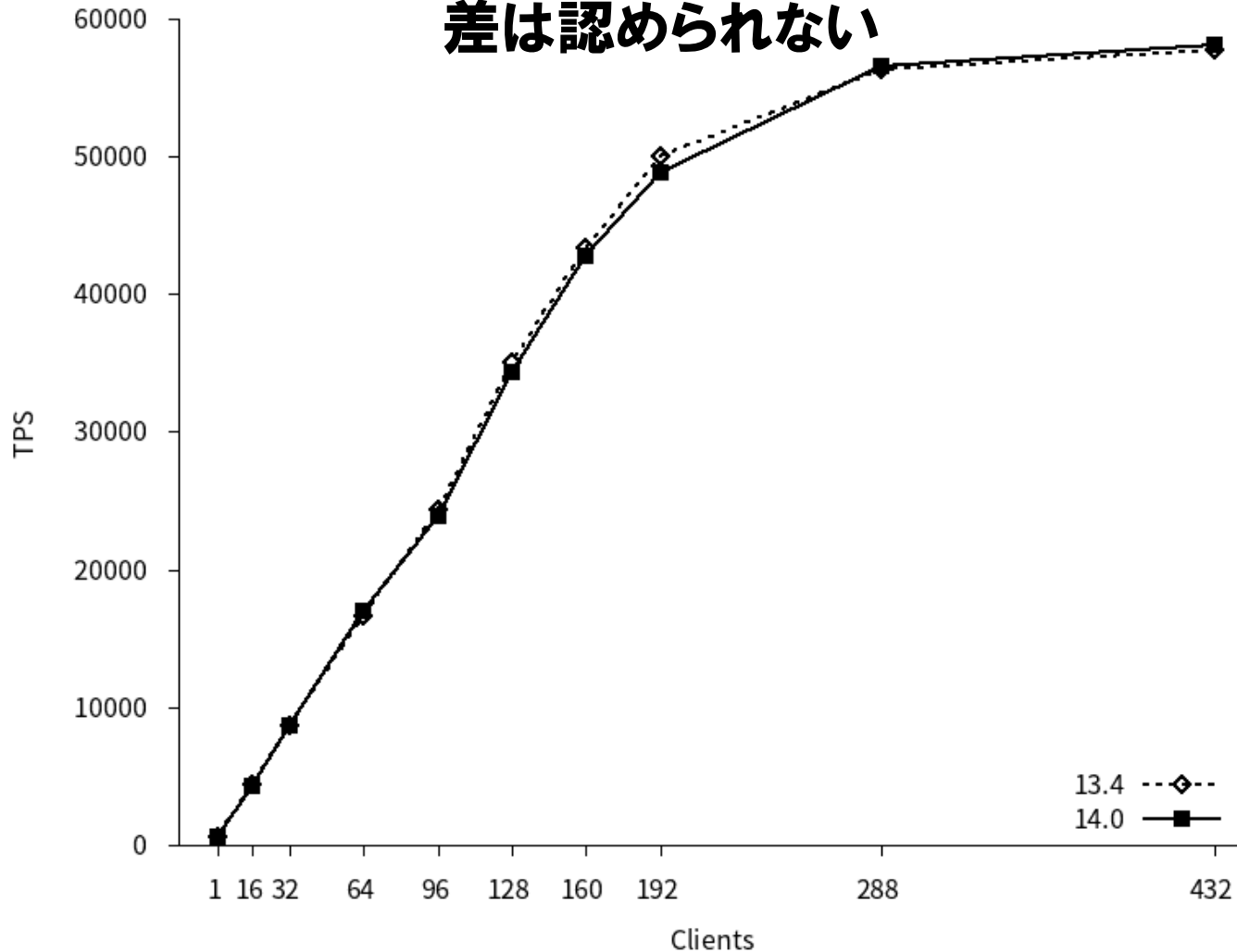
64接続でTPSの向上は頭打ち



検証結果

■ 更新系

検証した範囲ではバージョン間に
差は認められない



参照系の追加検証

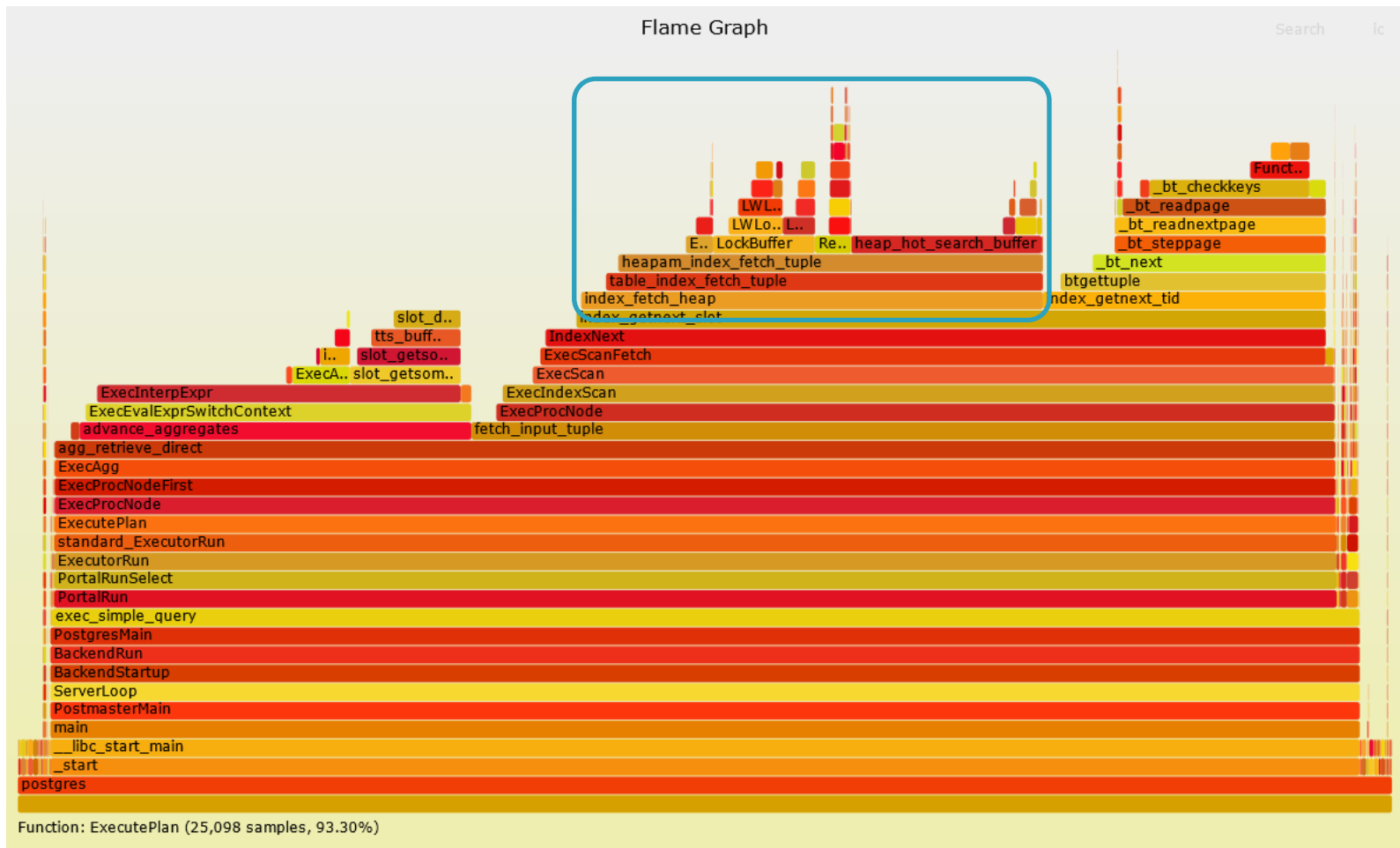
- バージョン13と14の参照系検証結果から、14に性能劣化が確認された
- 前バージョン比で性能劣化の割合が最も大きかった、クライアント数1の条件で追加検証
- バージョン間で内部的に呼び出される関数、関数ごとのCPU使用率の違いを確認するため、perfを元にFlameGraphを生成

参照系ベンチマーク結果

クライアント数	TPS		前バージョン比
	V13.4	V14.0	
1	437.2413	412.9373	94%
16	4778.467	4668.131	98%
32	8404.637	8193.761	97%
64	8546.134	8361.549	98%
96	8405.358	8229.714	98%
128	8310.025	8120.542	98%
160	8218.915	8055.056	98%
192	8188.156	8000.038	98%

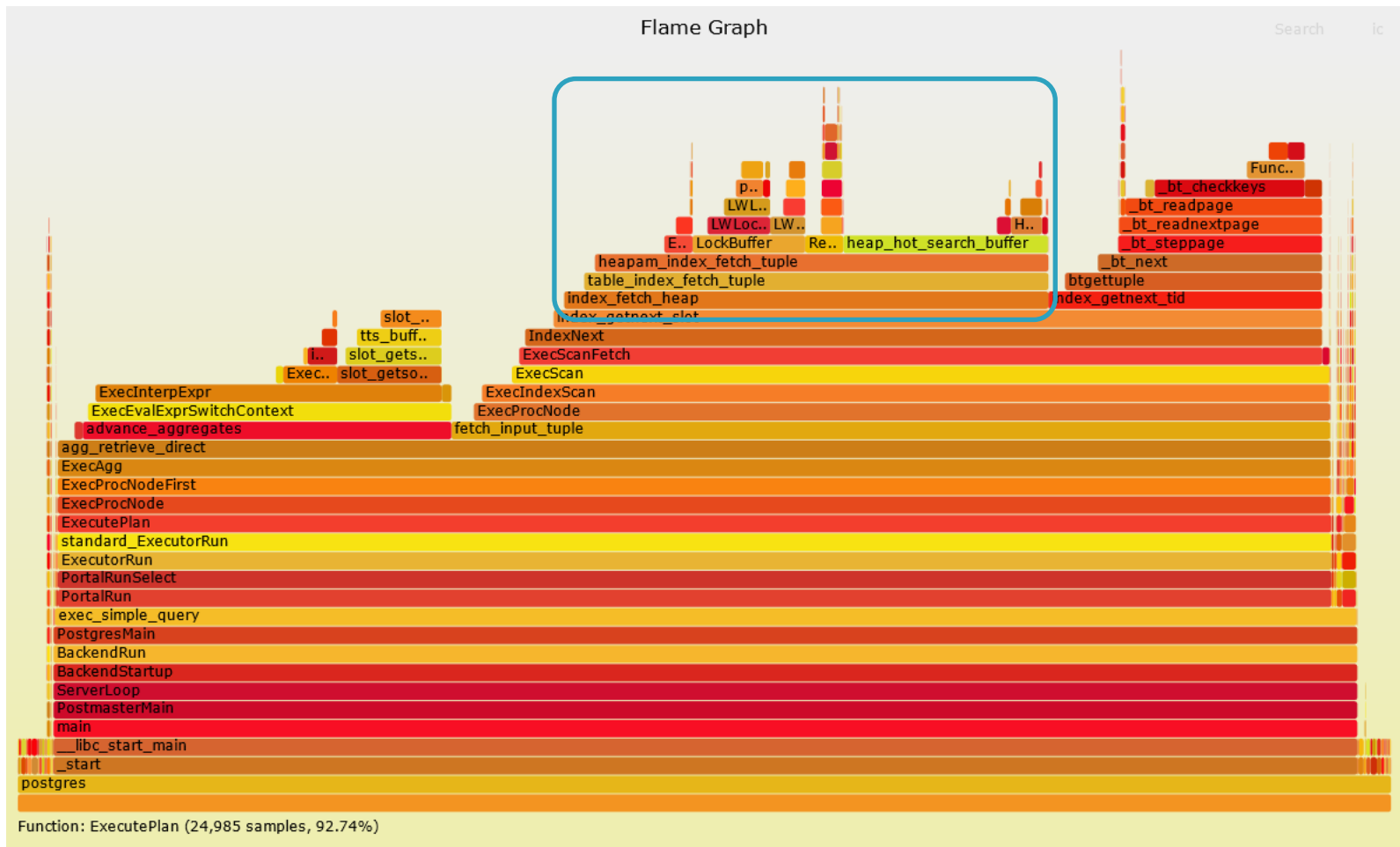
参照系の追加検証結果

■ PostgreSQL 13



参照系の追加検証結果

■ PostgreSQL 14

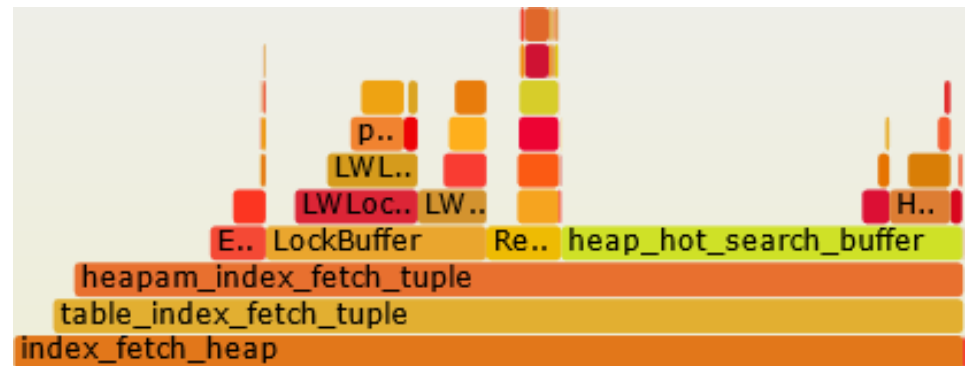


参照系の追加検証結果

- 呼び出されている関数に大きな違いなし
- `index_fetch_heap`の割合が少し増えている



- 共有バッファアクセスの品質改善に伴ってオーバヘッドが増えた
 - 更新性能はアクセスが少なく、影響が小さい



PostgreSQL 14.0 Release Note: <https://www.postgresql.org/docs/release/14.0/>

E.3.3.12. Source Code

- Various improvements in valgrind error detection ability (Álvaro Herrera, Peter Geoghegan)

まとめ

- **今回のケースでは参照性能は2%程度低下**
 - ・ 共有バッファアクセスの品質改善によるオーバヘッドが関与の可能性あり
 - ・ すべてのケースで性能低下するわけではない
 - ・ 新機能が有効に働けば、性能向上するケースあり
 - ・ Btree Index肥大化防止
 - ・ LZ4によるデータ圧縮
 - ・ 多セッションにおけるオーバヘッド軽減
 - ・ Nested Loopアルゴリズム改善
 - ・ 正規表現の性能向上 など
- **更新性能は変化なし**
 - ・ 432接続まで増やし、性能限界が見えてきた



PGECons

PostgreSQL Enterprise Consortium