



**PGECons**  
PostgreSQL Enterprise Consortium

# 2022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）**

# 責任範囲

- 本資料は、PGECconsが独自に検証した結果であり、結果はPGECconsの責任の元、公開しています。



---

# 検証方法

# 検証概要

## ■ 目的

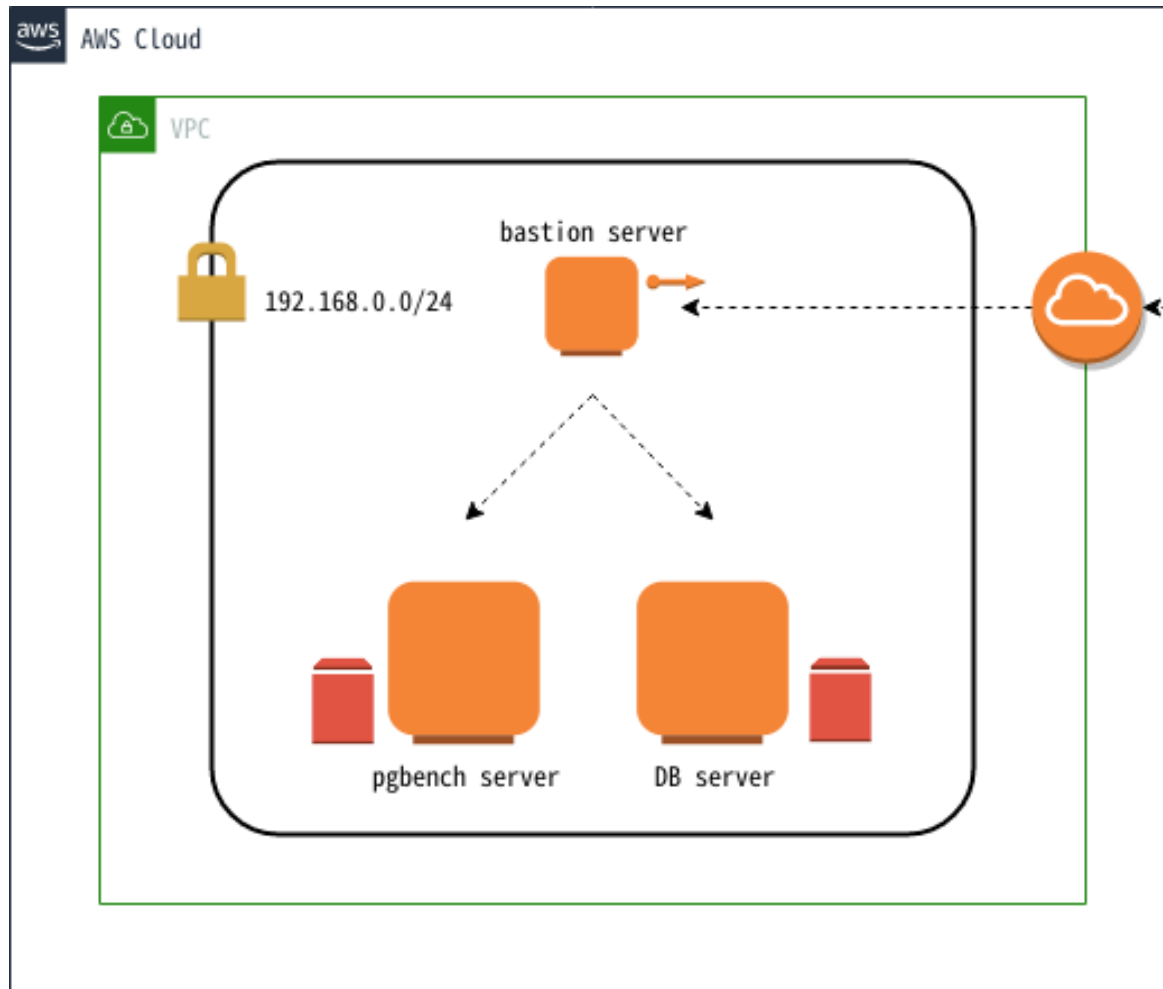
- メニーコアCPU上でのPostgreSQLのスケーラビリティを検証
- 新バージョンのPostgreSQLの性能改善傾向を知る
  - PGECons発足当初(2012年度、PostgreSQL 9.2)から継続的に実施(定点観測)
    - 更新系性能に関する定点観測は2014年度から開始
    - バージョン間で性能差が現れたときはその要因も検証
- 今年度はV14とV15の比較を実施

## ■ 検証内容

- 参照性能
- 更新性能

# 検証手続き > 環境

- 今年度も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

- 2012年度から同じ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		15.1
DB server	RHEL 7.9	14.6, 15.1	

- 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

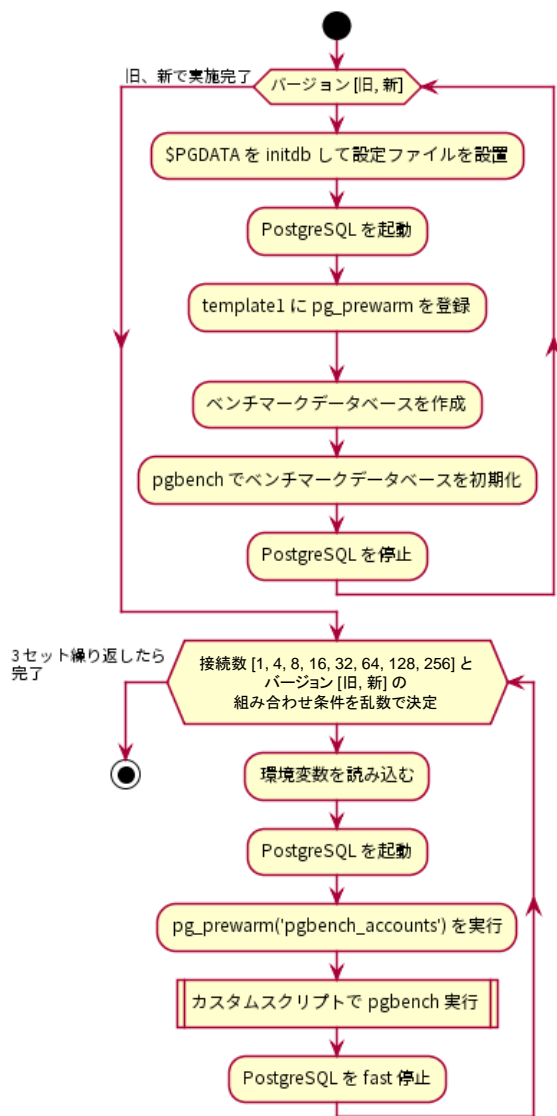
```
$ pgbench -i -s 2000 《データベース名》 -F 80
```

- 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 = 2000 # 頭打ちを観測するため、2021年度の500から増やした
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



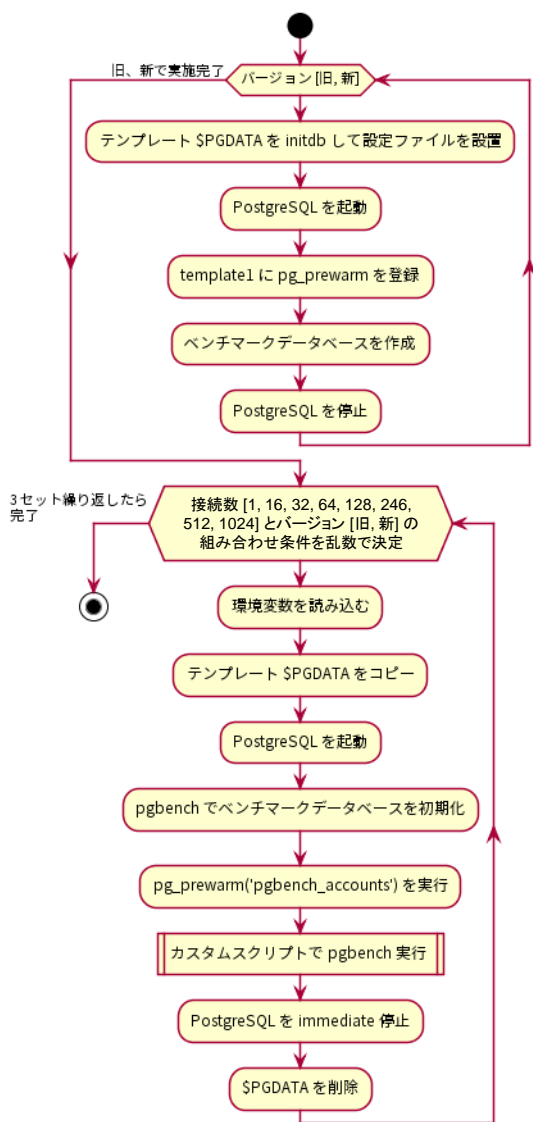
- PostgreSQLバージョン(2パターン)とクライアント同時接続数(8パターン)がTPSに及ぼす影響を計測
  - 2021年度、32接続で頭打ちになったため、その直前を詳細に計測
- $2 \times 8 = 16$ パターンのベンチマークをランダムな順序で3回繰り返す
- 得られたTPSの中央値を結果として採用

## 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パターン)とクライアント同時接続数(8パターン)がTPSに及ぼす影響を計測
  - 2021年度、432接続で頭打ちにならなかったため、1024接続まで計測
- 2×8=16パターン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;
```



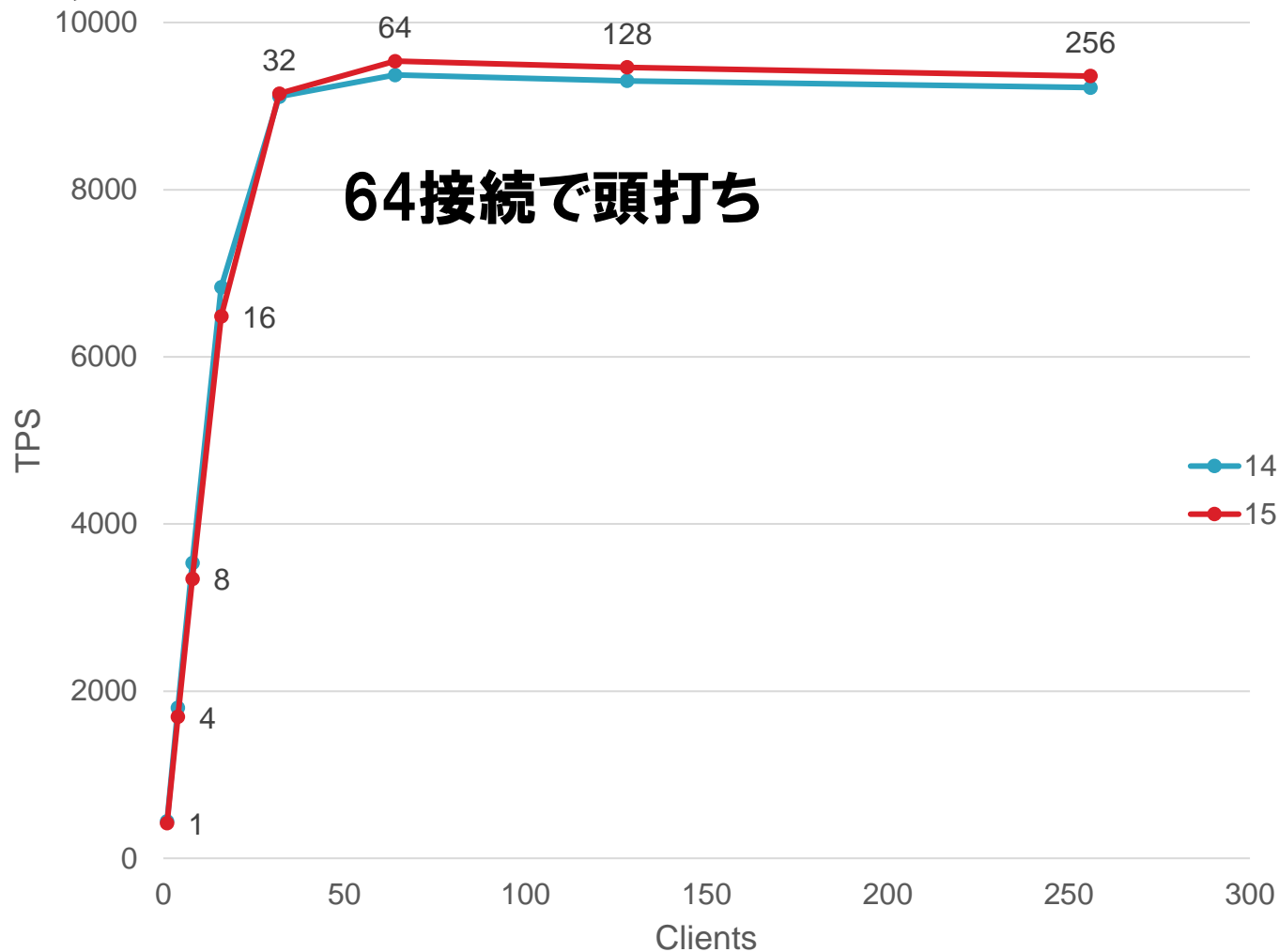
---

# 検証結果

# 検証結果

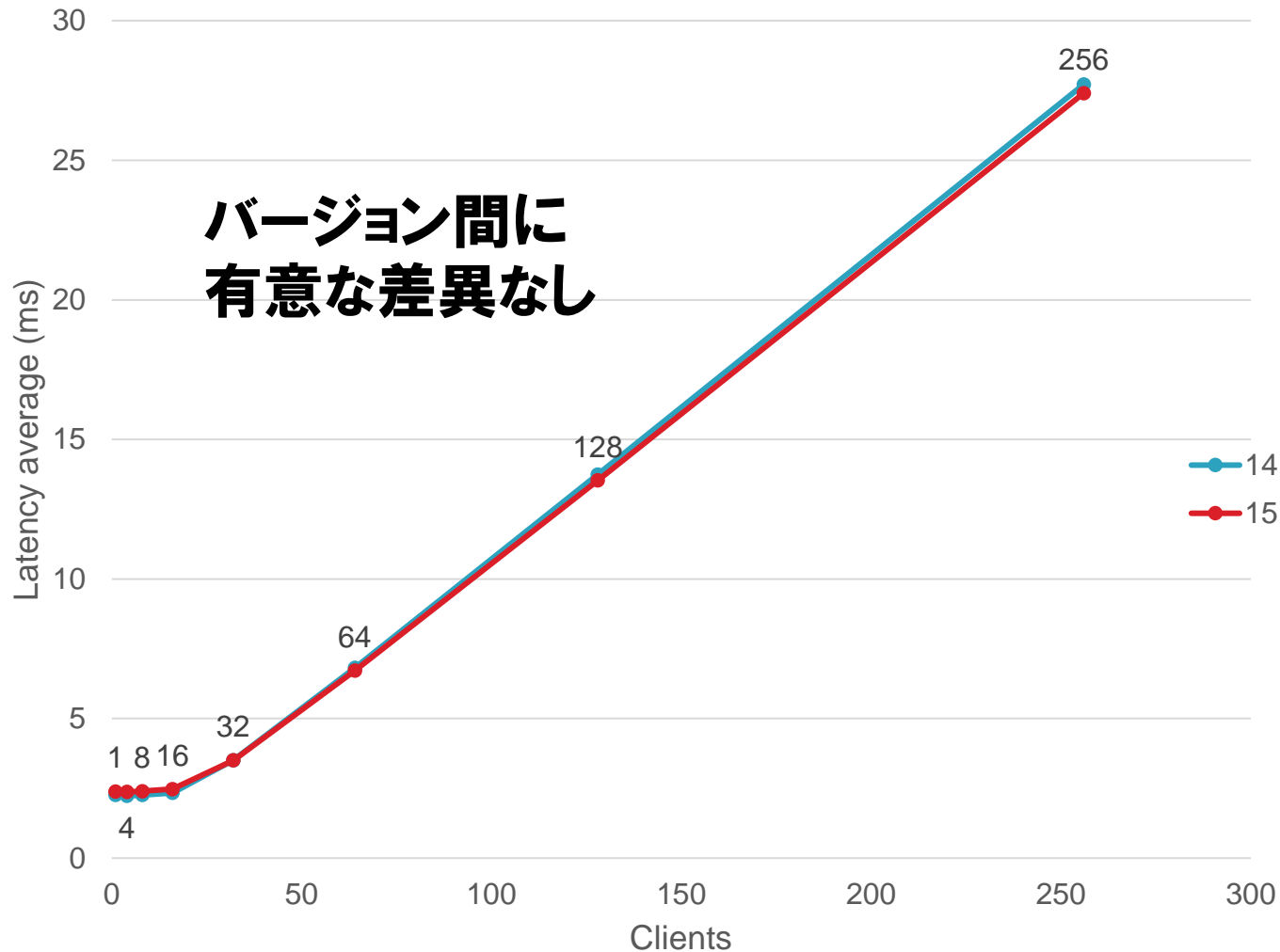
## ■ 参照系 TPS

バージョン間に有意な差異なし



# 検証結果

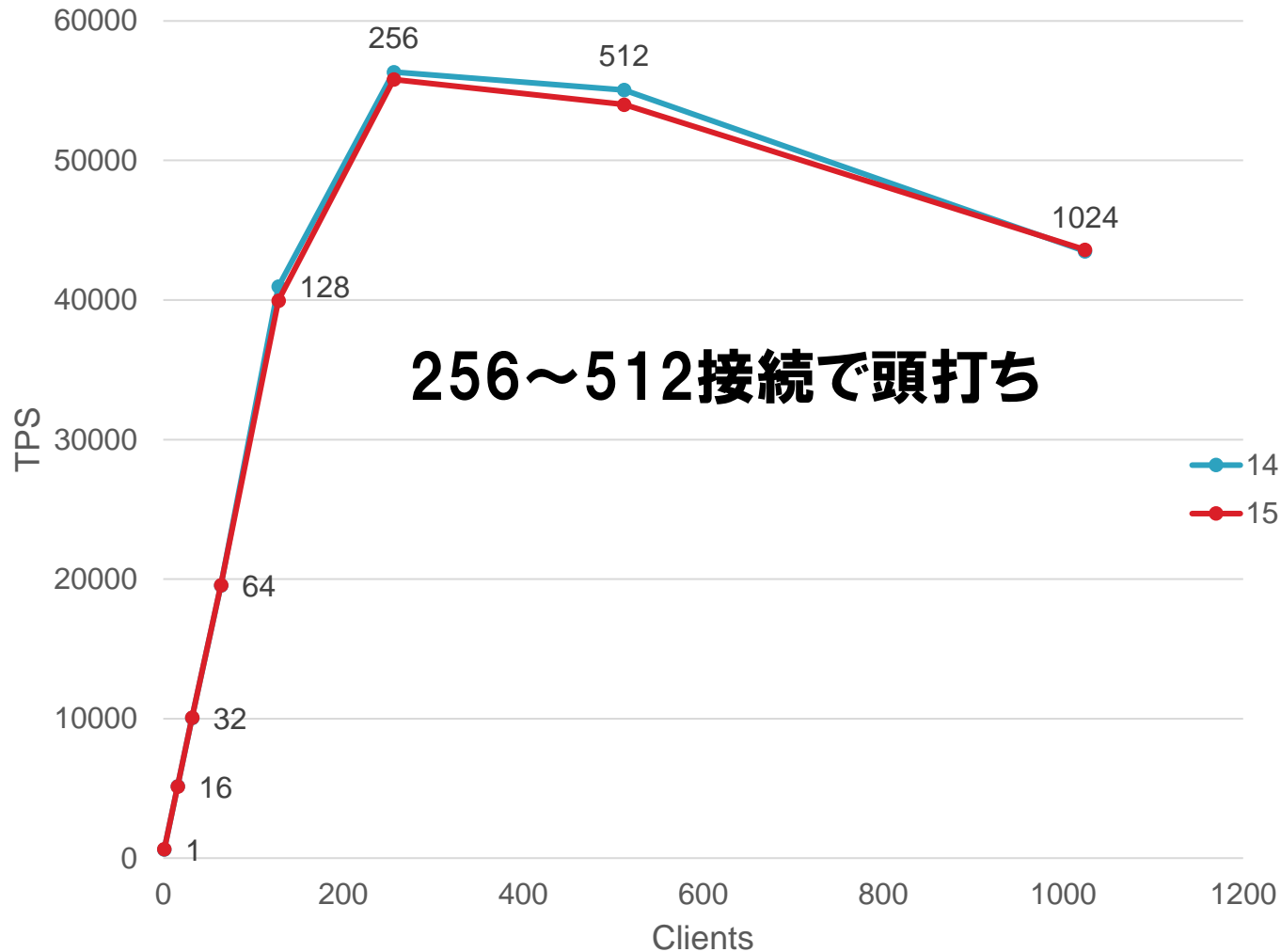
## ■ 参照系 レイテンシ



# 検証結果

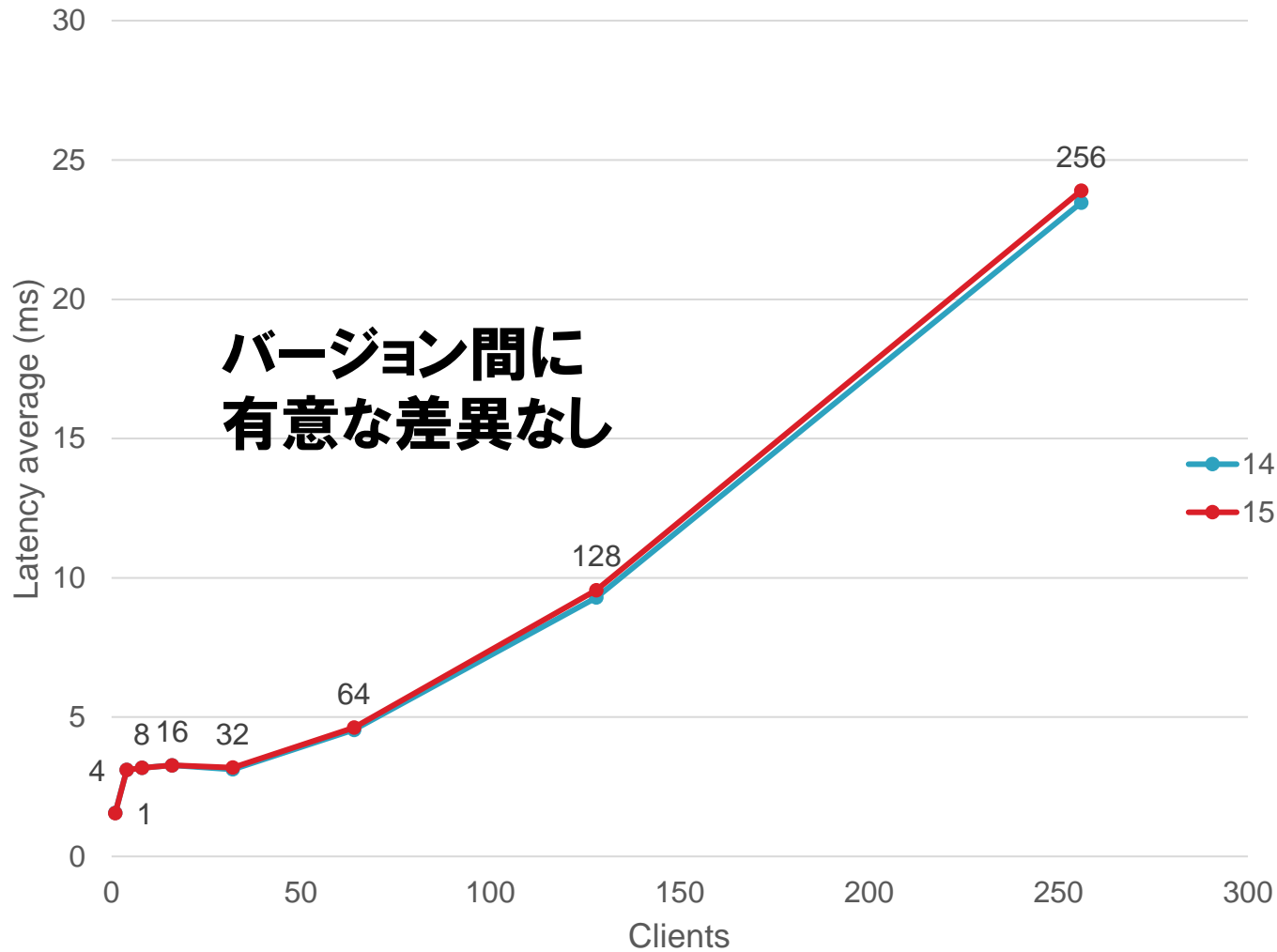
## ■ 更新系 TPS

バージョン間に有意な差異なし



# 検証結果

## ■ 更新系 レイテンシ



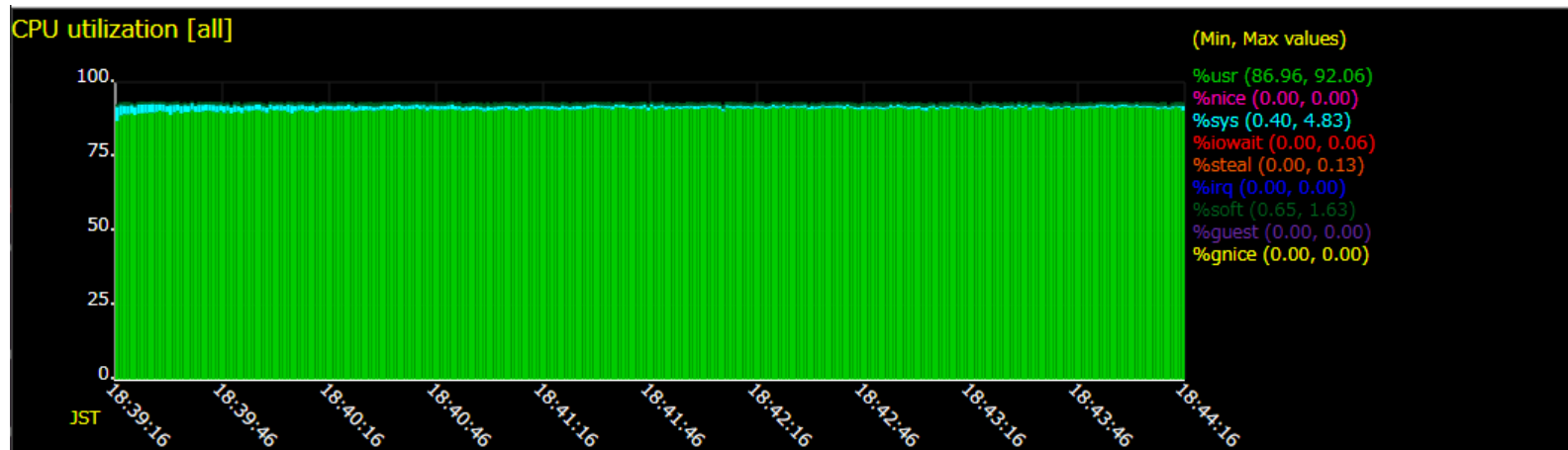
---

# まとめ

# まとめ (1/2)

- 参照、更新性能ともにバージョン間に有意な差異なし
- 参照性能は64接続で頭打ち
  - 全体のCPU使用率は32接続で90%、64接続で100%、ユーザがほとんどを占め、CPUがボトルネック

バージョン15、32接続、CPU使用率(全体)

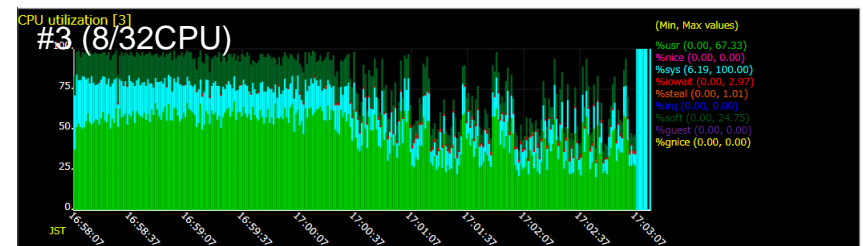
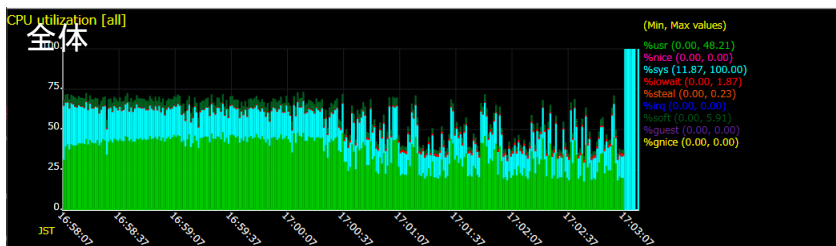




# まとめ (2/2)

- 更新性能は256～512接続で頭打ち
  - 2021年度頭打ちの432接続付近も測定すべきだった
  - 全体のCPU使用率は1024接続で70%、ユーザが3分の2、カーネルが3分の1、ソフトウェア割り込みがわずかを占める
  - 32個中8個のCPU使用率はソフトウェア割り込みが25%を占め、100%近くに達する時間あり、ソフトウェア割り込みがボトルネックと推測
  - ソフトウェア割り込み発生CPUの偏りはirqbalanceが負荷分散を行うため
    - 特定CPUを対象外にできるが、全CPUを対象にはできない
    - 全CPUを対象にすると、効率が悪くなる可能性あり
  - ソフトウェア割り込み発生原因は特定できず、今後の課題として持ち越し

## バージョン15、1024接続、CPU使用率



■ %soft: ソフトウェア割り込み、■ %sys: カーネル、■ %usr: ユーザ



# PGECons

PostgreSQL Enterprise Consortium