



PGECons
PostgreSQL Enterprise Consortium

2020年度活動成果報告 PostgreSQL 13の性能傾向

2021年5月27日

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

定点観測/スケールアップ検証 成果紹介

検証概要

■ 目的

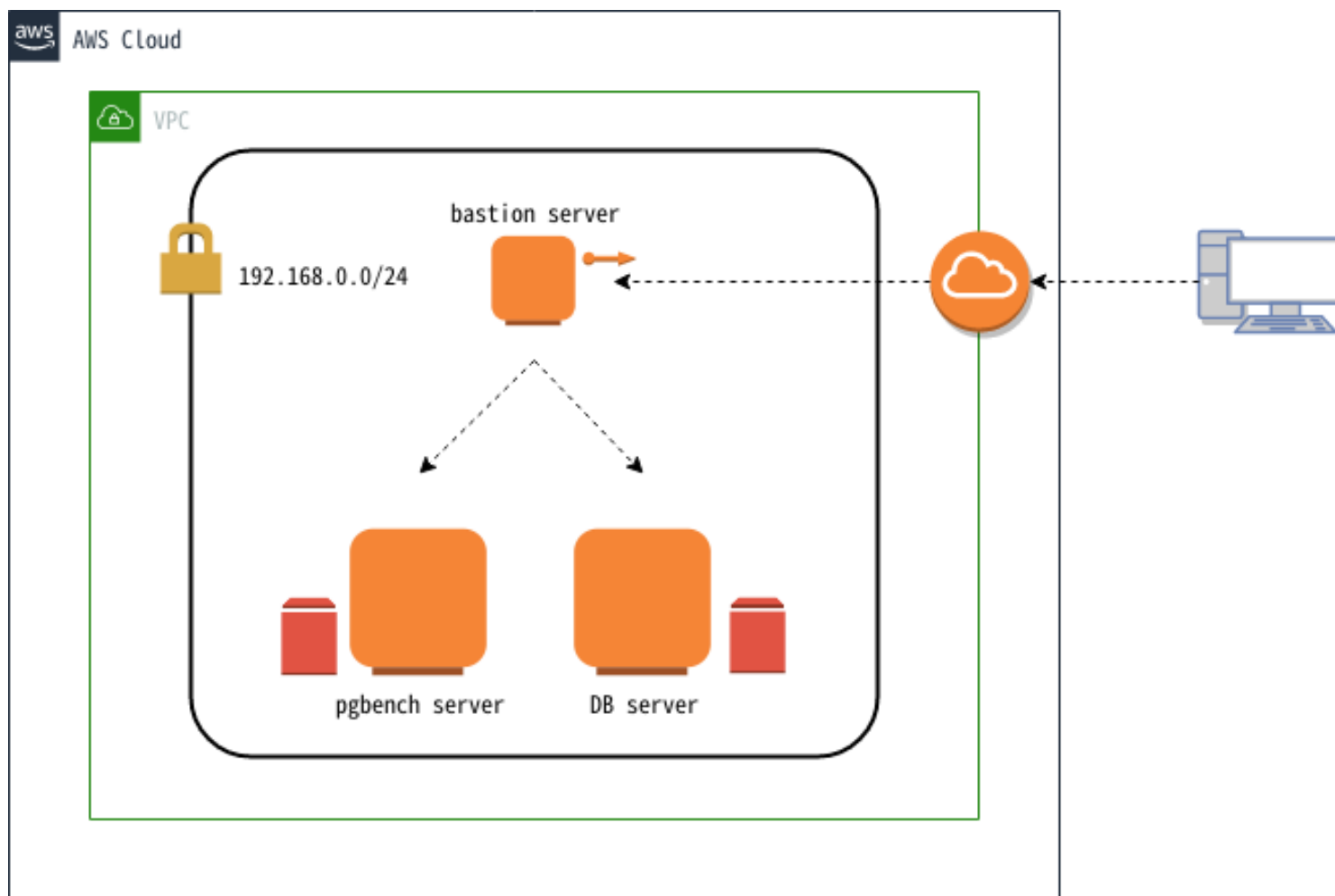
- メニーコアCPU上でのPostgreSQLのスケラビリティを検証
- 新バージョンのPostgreSQLの性能改善傾向を知る
 - PGECons発足当初 (2012年度、PostgreSQL 9.2) から継続的に実施 (定点観測)
 - 更新系性能に関する定点観測は2014年度から開始
 - バージョン間で性能差が現れたときはその要因も検証
- 今年度はV12とV13の比較を実施

■ 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		13.2
DB server	RHEL 7.9	13.2, 12.6	

- セットアップはAnsibleで自動化

検証手続き > 環境

■ 試験データベースサイズ

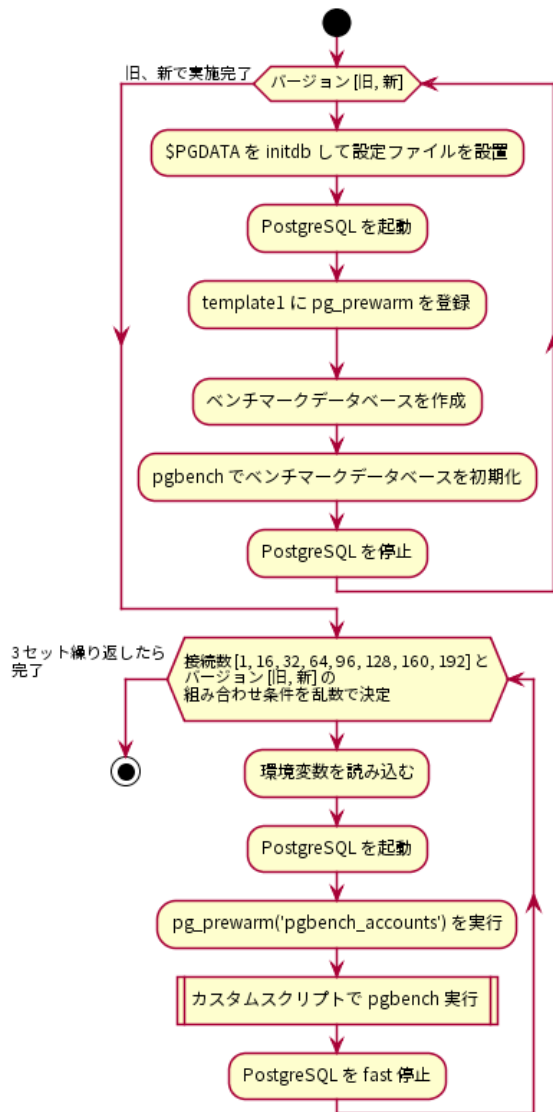
```
$ pgbench -i -s 2000 《データベース名》 -F 80
```

- 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

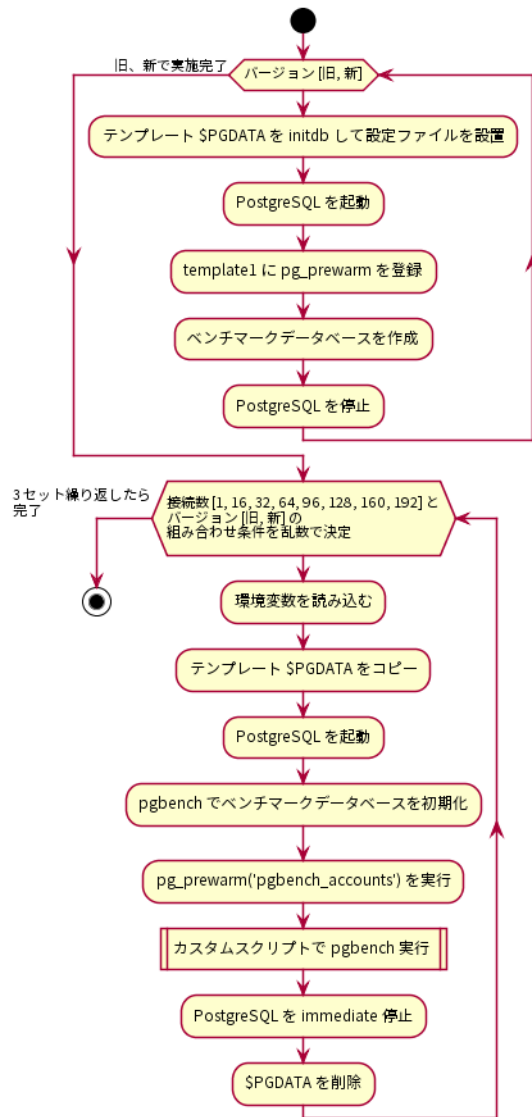
```
listen_addresses = '*' # クライアント用サーバからの接続用
max_connections = 500 # 多めに設定
shared_buffers = 40GB # 試験用データがすべてメモリに載るように設定
work_mem = 1GB
maintenance_work_mem = 20GB
checkpoint_timeout = 60min # 試験中にチェックポイントを発生させない
max_wal_size = 160GB # 試験中にチェックポイントを発生させない
logging_collector = on
log_checkpoints = on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

検証手続き > 更新系

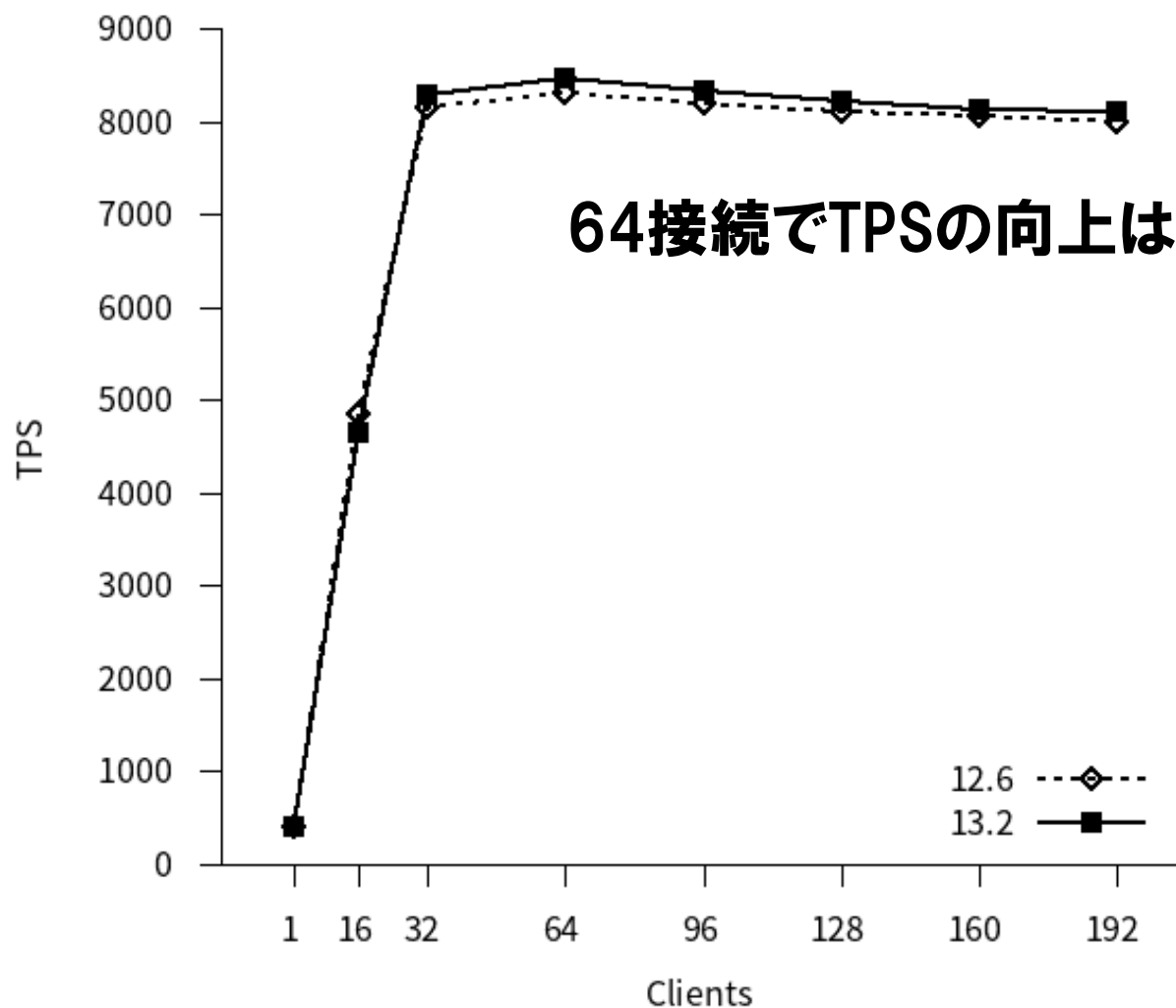


- 試行の都度\$PGDATAを作り直す（それ以外は参照系と同様）
- PostgreSQLバージョン(2水準)とクライアント同時接続数(8水準)がTPSに及ぼす影響を計測
- $2 \times 8 = 16$ 種類のベンチマークをランダムな順序で3回繰り返す
- 得られたTPSの中央値を結果として採用

検証結果

■ 参照系

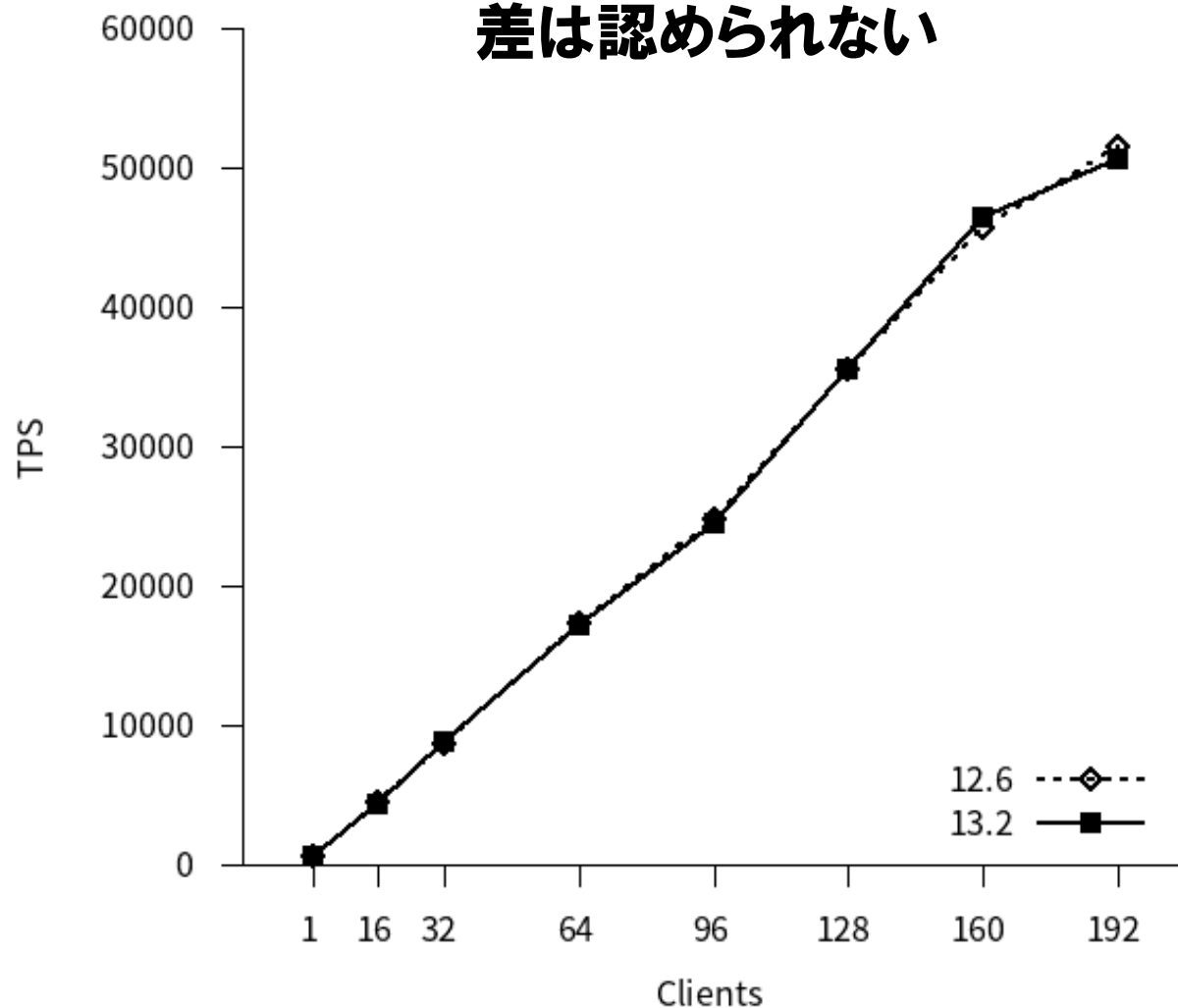
バージョン間に差は認められない



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



まとめ

- 参照性能、更新性能共にバージョン間で性能変化なし
- 更新性能は192接続時点までほぼ線形にTPSが上昇
 - ・ 性能限界付近ではTPS値に差があった可能性は残る
 - ・ 次年度以降は性能限界まで接続数を増やして計測する



Windows版定点観測検証 成果紹介

検証概要

■ 目的

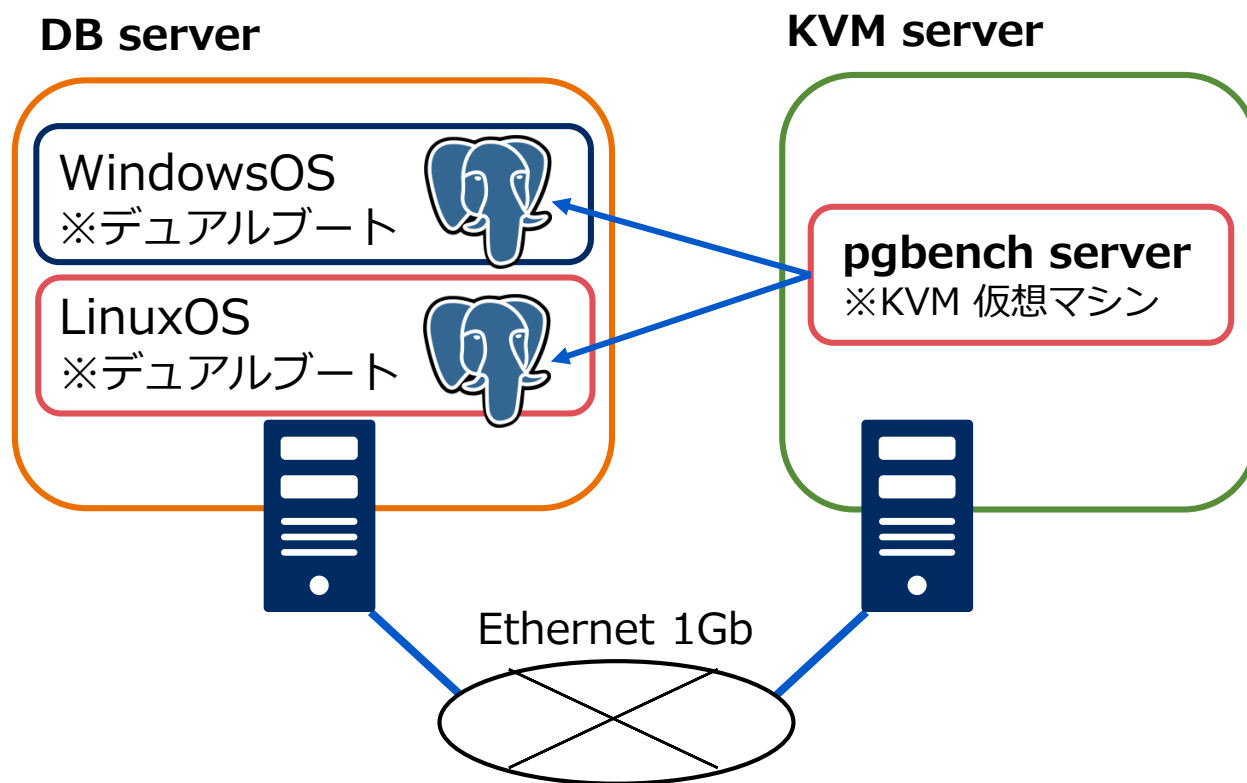
- 毎年実施している定点観測についてWindowsOSの傾向を確認する
 - バージョン間での性能差傾向を知る
 - 同条件でのLinuxOSとWindowsOS間の性能差を知る
- 今年度は昨年度未実施であったV11-V12検証を含め、V11、V12、V13の3Verで比較

■ 検証内容

- 参照性能
- 更新性能

検証手続き > 環境

- 物理サーバ1台に2種類のOSをデュアルブートで設定し
検証用PostgreSQLを構築、クライアントは仮想マシンを使用



検証手続き > 環境

■ マシンスペック

名称	CPU	メモリ	ストレージサイズ
DB server	Xeon E5-2630 v4(10C/20T) x 2	256GB	400GB SSD x 8 パーティションにより各OSで1.2TB設定
KVM server	Xeon E5-2630 v4(10C/20T) x 2	256GB	600GB HDD x 8
pgbench server	4vCPU	4GB	100GB

■ ソフトウェア

名称	OS	PostgreSQL	pgbench
DB server(WindowsOS)	Windows Server 2019	11.11 (EDB installer版) 12.6 (EDB installer版) 13.2 (EDB installer版)	
DB server(Linux OS)	RHEL 8.3	13.2 (ソースビルド版)	
pgbench server	RHEL 7.2		13.2 (ソースビルド版)
KVM server	RHEL 7.2		

検証手続き > 環境

■ 試験データベースサイズ

\$ pgbench -i -s 2000 《データベース名》

- pgbenchで30GBほどの試験用データベースを作成
- fillfactorの設定はデフォルト (テーブルのfillfactor=100)、unlogged tableの指定はなし
- PGDATA領域とWAL格納領域を別パーティションに分離する

■ 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 = 64GB	# システムメモリの25%
work_mem = 128MB	
maintenance_work_mem = 2047MB	# Windows版PostgreSQLでは上限値がLinux版よりも小さく、測定環境の限界値を設定した。
checkpoint_timeout = 1d	# 試験中にチェックポイントを発生させない
max_wal_size = 100GB	# 試験中にチェックポイントを発生させない
effective_cache_size = 128GB	# システムメモリの50%
autovacuum = off	# 試験中に I/O 処理を発生させない

検証方法(参照系)

■ 参照系検証

□ 測定方針

- 同時クライアント数(仮測定により策定)
 - 1, 2, 4, 8, 16, 24, 32, 40, 48 で実施(スレッド数は半分)
- 測定前に、キャッシュへの読み込みを実施
 - `SELECT pg_prewarm('pgbench_accounts');`
- 3回測定し、中央値を採用
- 測定はカスタムクエリ `accounts` からランダムに10000行読む

```
¥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;
```

検証方法(更新系)

■ 更新系検証

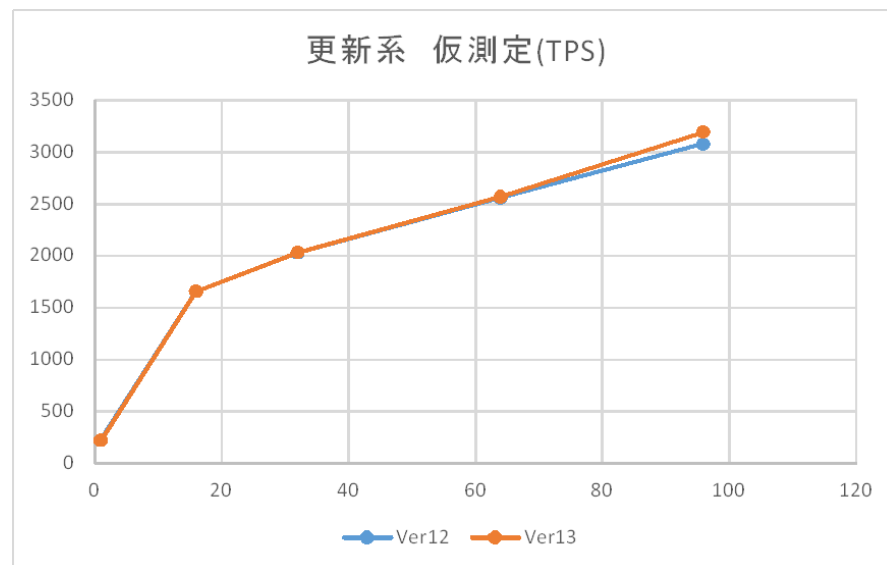
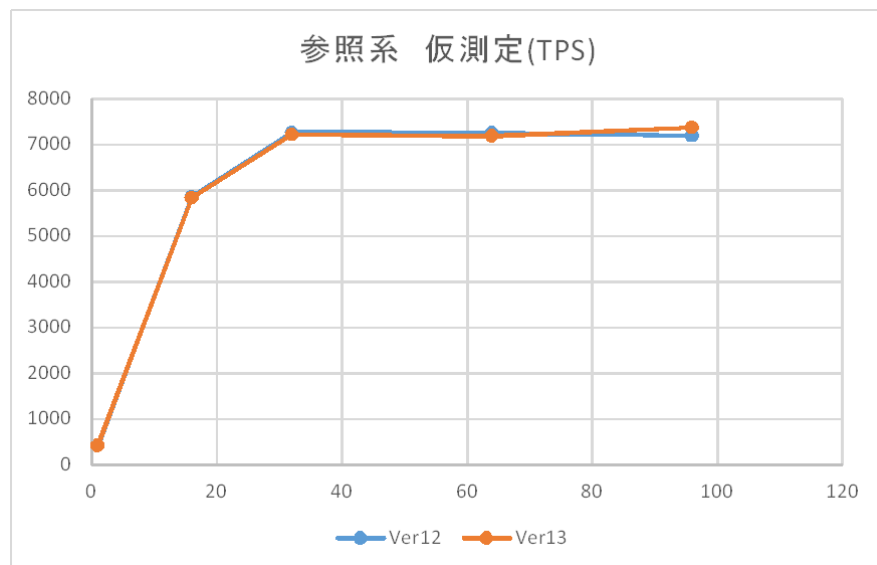
□ 測定方針

- 同時クライアント数(仮測定により策定)
 - 1, 16, 32, 64, 96, 128, 160, 192で実施(スレッド数は半分)
 - 上記順番で1回ずつ測定、これを3周する
 - 1回測定するたびにcheck pointとvacuumを実施する
- 各測定週の初めに、キャッシュへの読み込みを実施
 - `SELECT pg_prewarm('pgbench_accounts');`
- 各測定週の最後に検証データの再作成と
vacuum full, check point, ANALYZEを実施
- 3周測定した結果の中央値を採用
- 測定はカスタムクエリ accountsからランダムに選んだ行をUPDATE

```
¥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;
```

仮測定の実施

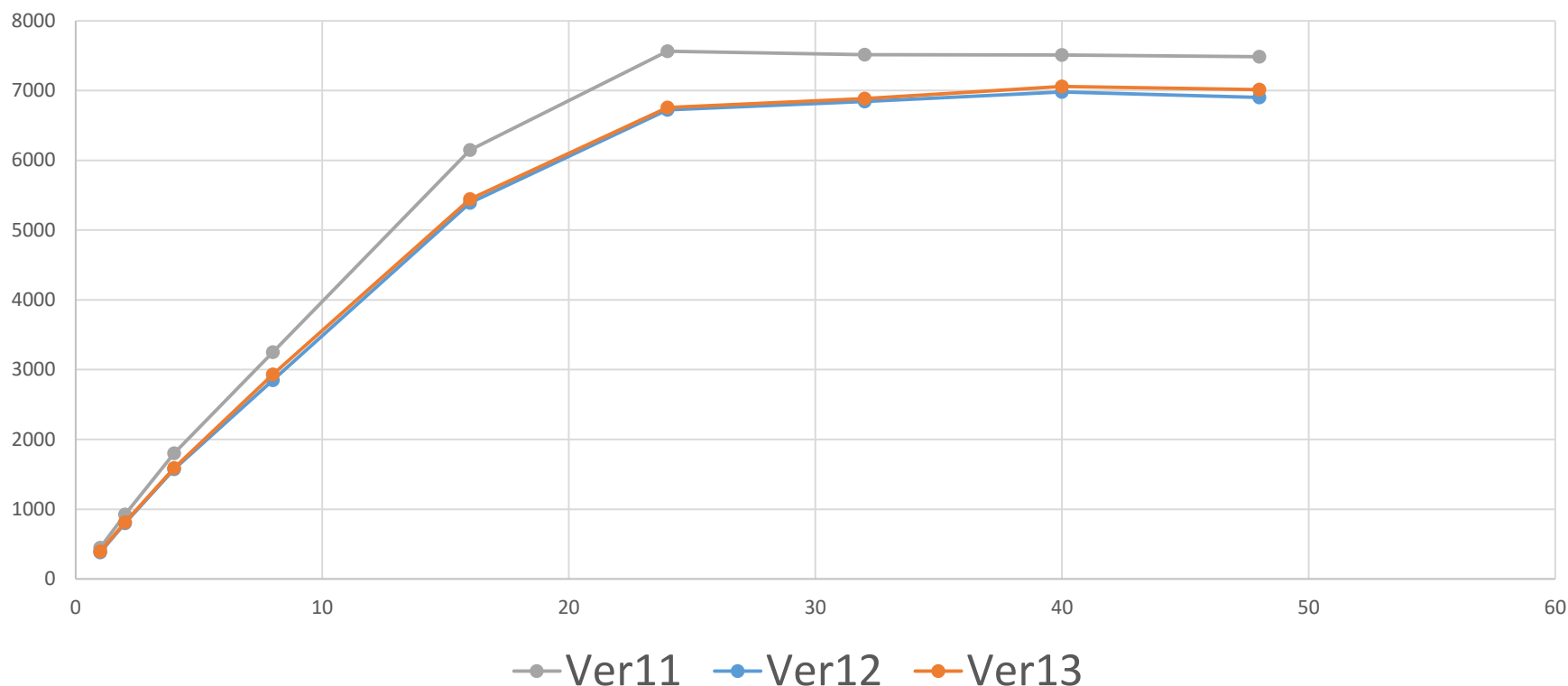
- 本測定の前に、今回の検証マシンのスペックでどの程度の同時実行数まで性能向上が期待できるかを仮測定で確認
 - 同時クライアント数は1, 16, 32, 64, 96



- 参照系は32接続で限界、更新系は96でも限界を迎える様子がないため、それぞれ1～48、1～192まで実施することを策定
- 仮測定の時点ではV12とV13の差がないことを確認

検証結果 (参照系)

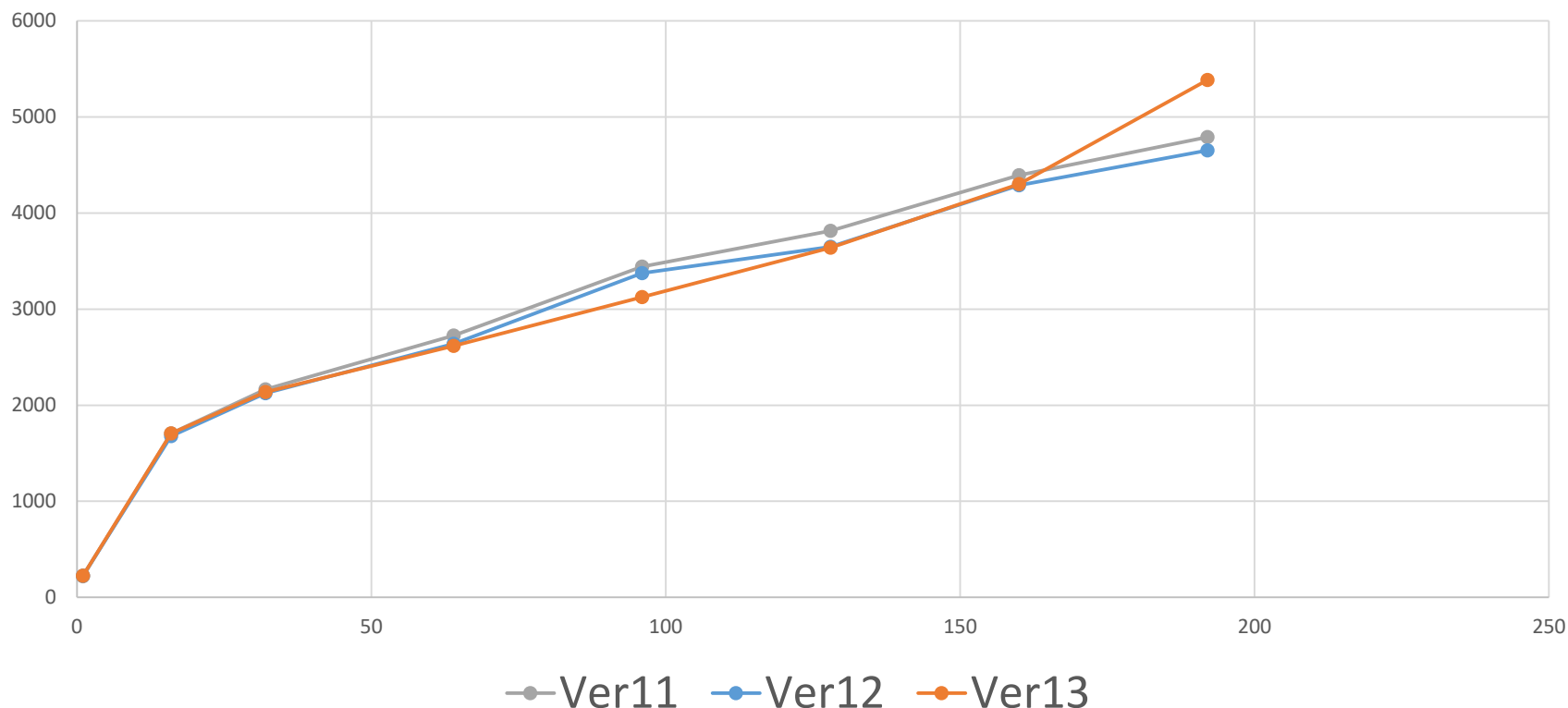
参照系 TPS



- 物理コア数20、論理コア数40の環境では24接続まで線形増加、40接続で収束することを確認。CPU使用率は40接続で100%付近に
- V12とV13は差が認められず。V11⇒V12の劣化は2019年度検証でLinuxでも観測されており、11/12/13のVer間性能差はLinuxと同様と言える

検証結果（更新系）

更新系 TPS



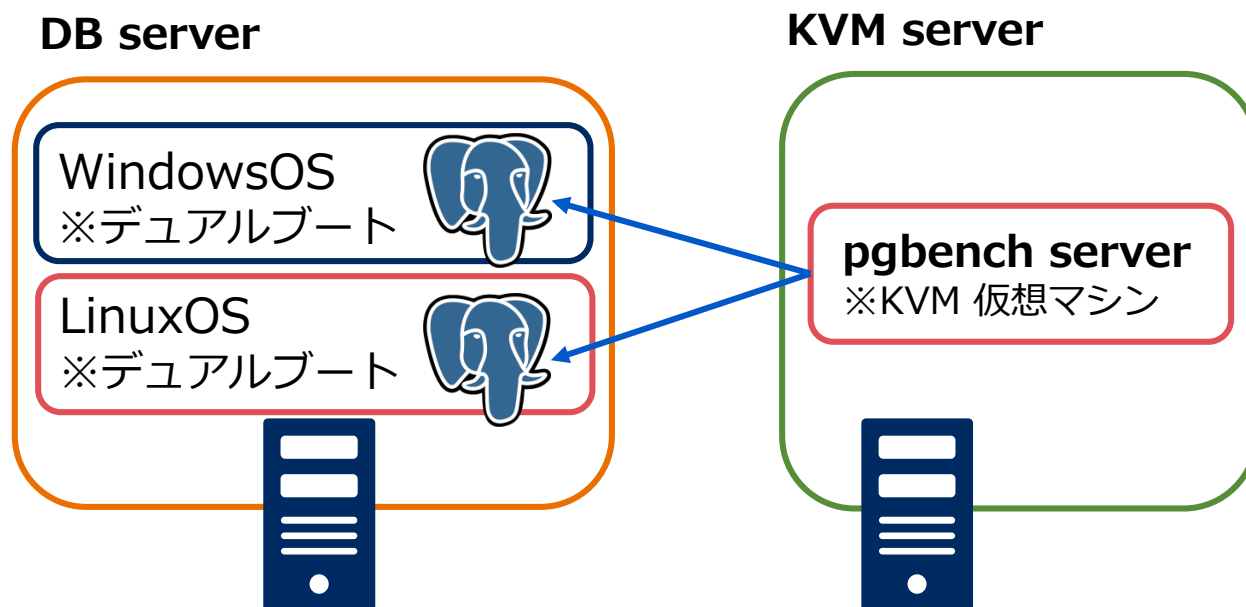
- 192接続までtpsのピークは迎えず、概ね線形に増加。リソース状況ではCPUは全体的に10%未満しか使いきれておらず、I/Oネックとなっていた
- 11,12,13の各Ver間の差については多少の揺らぎはありつつも明らかな差はなく、Linux定点観測と同じ傾向と言える

WindowsOSにおける定点観測の結果について

- V12とV13における参照性能、更新性能はバージョン間で性能傾向の差はなし。Linuxの結果も踏まえ、性能に影響する改修はなかったことが裏付けられる
- V11とV12における比較は参照性能のみV12が劣化しており、これは2019年度検証で確認されたTable Access Methodの導入に伴うオーバヘッドがWindowsOSでも発生していると考えられる
- 更新性能は192接続時点まではほぼ線形にTPSが上昇
 - ・ WAL保存領域のAvg. Disk sec/Writeは64接続で頭打ちとなるが、CPU使用率は僅かながらあがっていく傾向が見られた
 - ・ 2018年度検証の結果ではHDD環境で収束しており、ストレージがSSDになったことで収束しづらくなった傾向が見られる

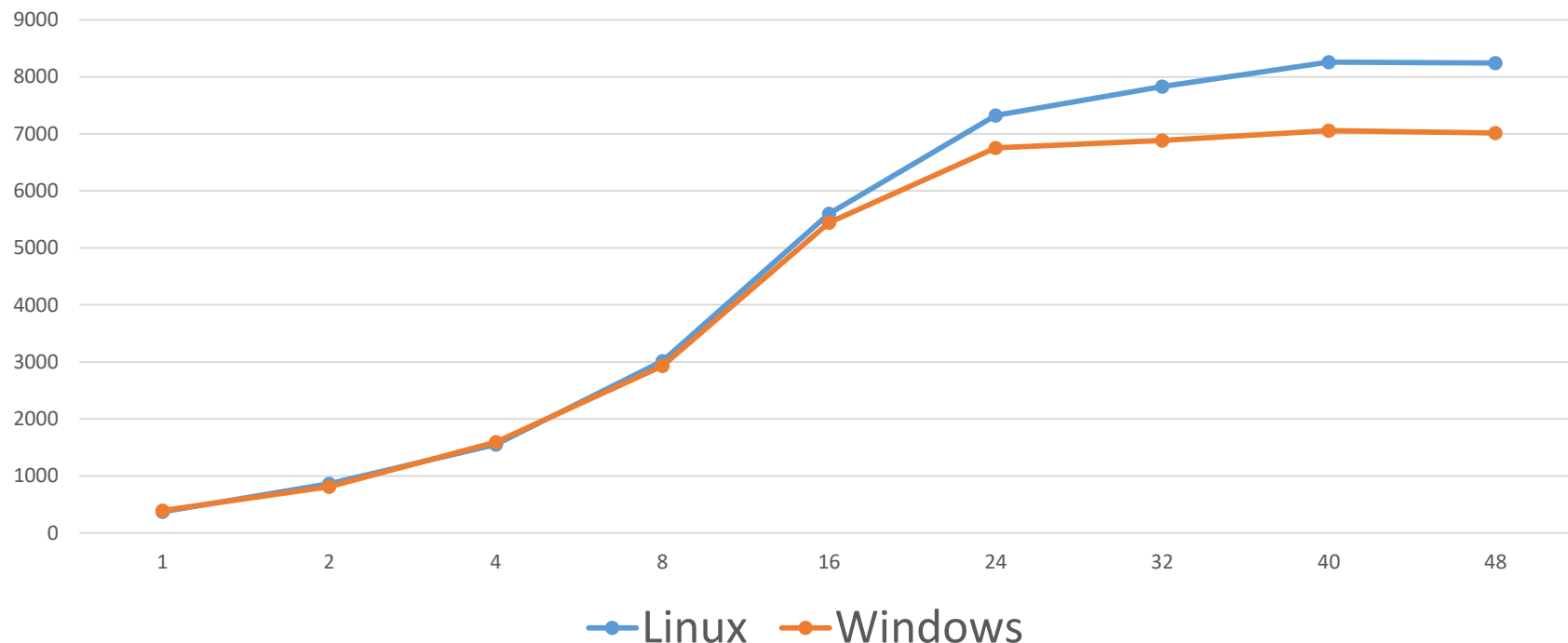
Linux WindowsOS 比較検証

- 同一サーバ上でのLinux版PostgreSQLとWindows版PostgreSQLの比較検証を行い性能差を確認する
- DB serverにデュアルブートでインストールしたRHEL 8.3にOSを切り替えて、構成や検証モデルと測定フロー、conf設定などすべてWindowsOS版測定と同条件としたうえでPostgreSQL 13での比較検証を実施



検証結果 (参照系)

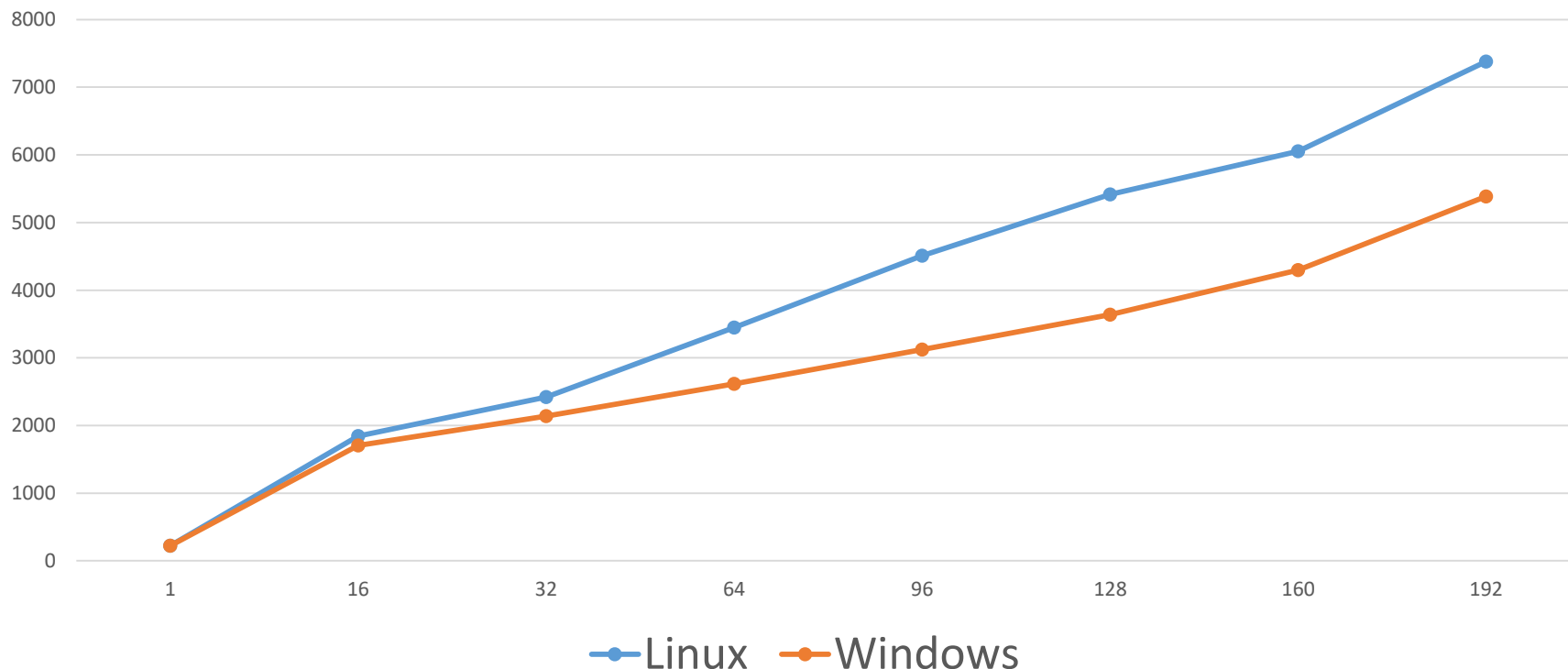
PostgreSQL 13 OS比較 参照系 TPS



- 同時接続数16までは同等の傾向/性能があると言えるが、24接続を超えたところでLinux版が優位に。CPU使用率はLinux版も同等の傾向であった
- 事前にprewarmでオンメモリにした上で測定を実施しているため、メモリの参照や管理になんらかの性能差が出ている可能性が考えられる

検証結果 (更新系)

PostgreSQL 13 OS比較 更新系 TPS



- 32接続から差が出始め、64接続以降は明確にLinuxの優位を確認。Linuxもsarによる%utilは100%に張り付いておりI/Oネックが確認された
- OSによるファイルシステムの違いが原因となった可能性が考えられる。本測定ではWindows版はNTFS、Linux版はxfs (inode64) で設定している

Linux WindowsOS 比較検証の結果について

- 同ースペック、同一条件でのLinux版との比較についてはTPSの増加傾向や参照系における頭打ちのタイミングなど性能特性は似ているが、スループットの絶対値については参照/更新ともにLinuxが優位という結果が確認できた
- Linuxが優位となる原因については、参照系についてはオンメモリでも差が出るためメモリ管理について差があること、更新系はファイルシステムによる性能の違いと見込んでいる
その仮説を裏付けるためのOSリソース値の取得であったり、メモリとファイルシステムそれぞれに特化したベンチマーク検証の考案が今後の課題である



PGECons

PostgreSQL Enterprise Consortium