



PGECons
PostgreSQL Enterprise Consortium

2020年度WG3活動成果報告 クラウド検証編 Azure PostgreSQL PaaS検証 ～実機検証結果（性能）～

**PostgreSQL エンタープライズコンソーシアム
WG3 パブリッククラウド検証チーム**

Contents

- FlexibleServer性能検証概要
- FlexibleServer性能検証結果 (pgbench)
 - 参照系処理
 - 更新系処理
- FlexibleServer性能検証結果 (HammerDB)
- 検証結果総括

責任範囲

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

***FlexibleServer*性能検証概要**

FlexibleServer性能検証概要

- Azure Database for PostgreSQLの新サービスであるFlexibleServerに対し、昨年度同様、以下の性能検証を実施
 - 簡易なクエリでの参照/更新性能の確認 (pgbench:カスタムクエリ)
 - 実際のワークロードを想定したスループット性能の確認 (HammerDB)
- 従来のサービスであり、近いモデルのSingleServerとの比較を実施し、各サービスの特徴を確認

FlexibleServer性能検証結果 (pgbench)

pgbenchとは？

- PostgreSQLに同梱されているベンチマークツール
 - TPC-B (銀行の口座処理) のデータモデル・トランザクションをよりシンプルにしたベンチマークツール
 - **実行結果はTransaction Per Sec (TPS) で出力され、PostgreSQLの性能を数値で把握する事が可能**
 - ただし、負荷掛けのシナリオはシンプルな内容となっており、実際のシステムでの性能を測定するというよりは、例えばPostgreSQLバージョンアップ、サーバのスケールアップ等を行った際に、新旧で性能を比較する際に用いられることが多い
 - **あらかじめベンチマークシナリオ (参照・更新) が用意されているが、自前でスクリプトを書いて、実行するSQLを定義する事も可能**
 - **昨年度同様、カスタムスクリプトにより簡易なクエリによる参照/更新性能を確認**

FlexibleServer性能検証 (pgbench)

■ pgbench測定の大方針

- データスケールは固定 (ScaleFactor:2000)
- 同時接続数は1,4,16,32,64,96 と、順次上げていく
- SingleServerとスループット等の比較を行い、性質・特徴を調査

■ PostgreSQLバージョン

- FlexibleServer,SingleServerで比較ができるver11で実施
※昨年度検証でのSingleServerはver10のため、ver11で計測し直し

■ PostgreSQLパラメータ

- work_mem といったHWスペックに依存しやすいパラメータは、DBインスタンスのスペックに合わせて適宜変更する

FlexibleServer性能検証 (pgbench)

■ 性能測定 補足事項

- 各性能測定は3回ずつ計測し、その平均値を結果とする
- インスタンス作成リージョンは東日本とする
- バックアップは「GEO冗長」をOFFにする
- ストレージの自動拡張はOFFとする
- リードレプリカは未作成とする
- インスタンスタイプは、「メモリ最適化」モデルを採用する
- ストレージサイズはSingleServer, FlexibleServerのIOPSが1100になるように設定する
- CPUコア数は8,32で検証
- DBへの接続はプライベートネットワーク内での直接接続に設定
 - FlexibleServer … Private IP接続
 - SingleServer … Private Endpoint接続

postgresql.conf設定要素

※記載のないものはデフォルト値

手動で変更した設定値
計算式で自動設定される設定値
変更不可な設定値

	SS/FS	備考
listen_addresses	*	
max_connections	960 (8vCPU) 1987 (32vCPU)	SingleServerは変更不可 FlexibleServerは変更可能なため統一
shared_buffers	16 GB (8vCPU) 64 GB (32vCPU)	インスタンスのvCPU数によって変化
work_mem	64MB	
maintenance_work_mem	512MB (8vCPU) 1GB (32vCPU)	
autovacuum	ON	
autovacuum_vacuum_scale_factor	0.5	
max_wal_size	5GB	
checkpoint_timeout	5min	
log_min_duration_statement	10	
log_duration	OFF	
log_statement	NONE	
max_worker_processes	8	
max_parallel_workers	8	
max_parallel_workers_per_gather	2	
checkpoint_completion_target	0.9	

実機検証 使用インスタンス

項番	インスタンス	スペック
1	負荷掛け用マシン	E8as v4 (8 vCPU,64 GB RAM) OS: Windows
2	DBインスタンス:FlexibleServer メモリ最適化モデル	E8 v3 (8vCPU 64GB RAM) E32 v3 (32vCPU 256GB RAM) ストレージ256GB (1100 IOPS)
3	DBインスタンス:SingleServer メモリ最適化モデル	Gen5 8vCPU 80GB RAM Gen5 32vCPU 320GB RAM ストレージ377GB (1101 IOPS)

検証方法

■ pgbench (参照系) 検証

□ PGECconsで実施している検証方法を踏襲

■ 初期化

□ pgbench -i -s 2000 [dbname]

■ キャッシュへの読み込み

□ SELECT pg_prewarm ('pgbench_accounts');

■ カスタムスクリプト

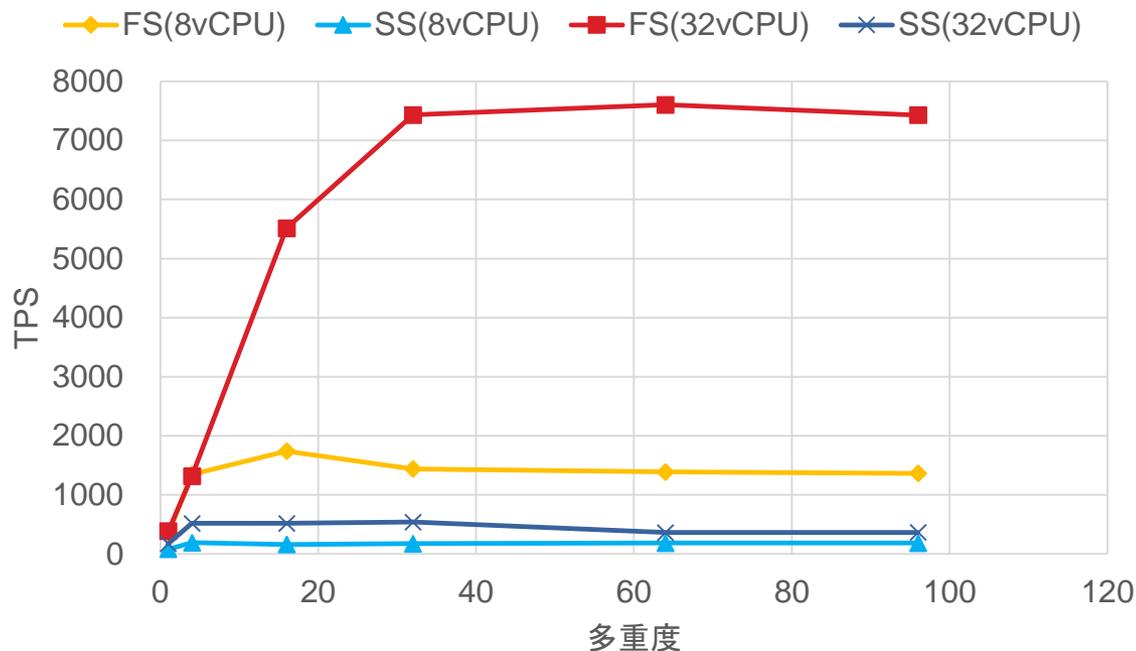
```
¥set naccounts 100000 * :scale
¥set row_count 10000
¥set aid_max :naccounts - :row_count
¥set aid random (1, :aid_max)
```

```
SELECT count (abalance) FROM pgbench_accounts WHERE aid BETWEEN :aid and :aid + :row_count;
```

■ 実行

```
pgbench -r -P 1 -n -c [clients] -j [threads] -f [カスタムクエリ] -T 300 -s 2000
-h [host] -p [port] [dbname]
```

検証結果:参照系 (TPS)

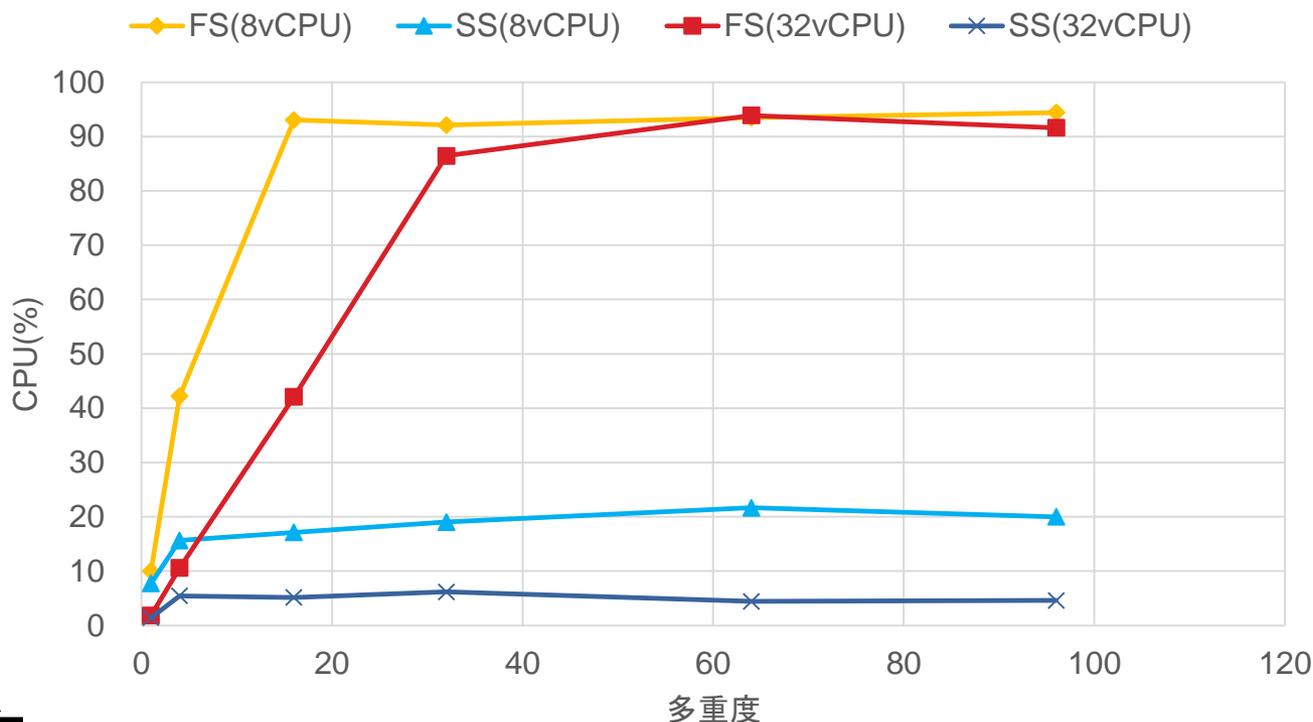


FlexibleServer:FS
SingleServer:SS

■ 傾向

- 8コア、32コアともに全ての多重度において、
FlexibleServerの方がSingleServerより高いスループット (TPS) を発揮
- FlexibleServerはコア数増加に伴いTPSが増加するが、
SingleServerはコア数を増やしてもTPSはほぼ変化なし

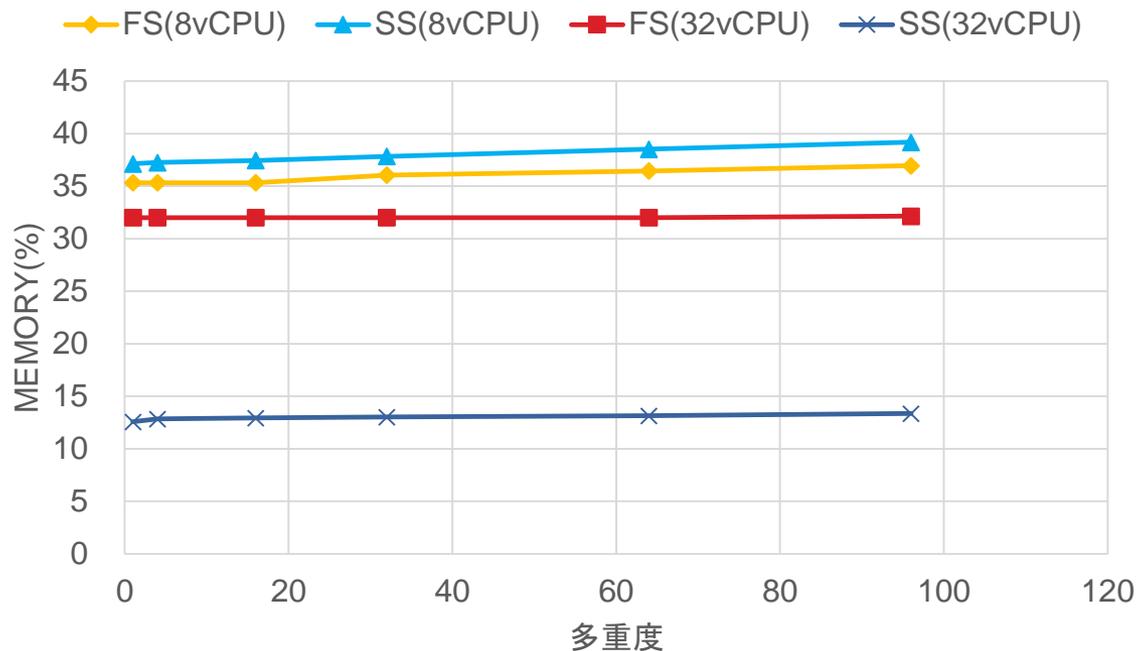
検証結果:参照系 (CPU利用率)



■ 傾向

- FlexibleServerはコア数×2の多重度でCPU利用率が100%近くに上昇
 - CPUコア数を増やすことによって多重度16においてCPUリソースに余裕が生じた
- SingleServerはCPUを使い切れていない (ネックにはなっていない)

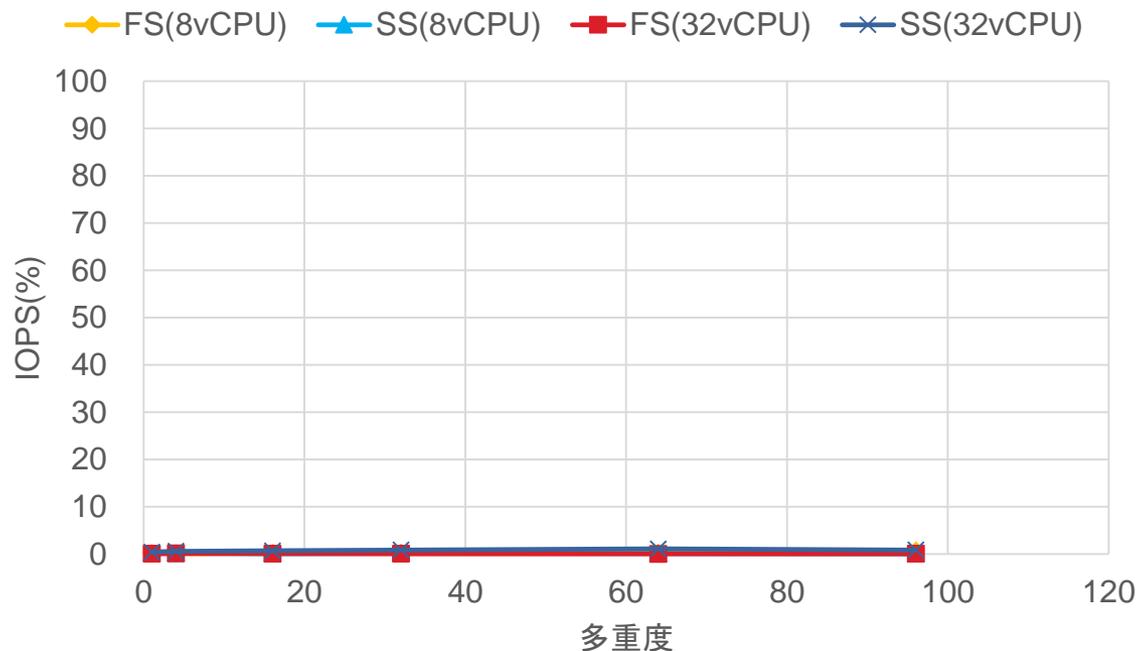
検証結果:参照系 (メモリ)



■ 傾向

- FlexibleServer, SingleServerともに簡易なクエリであるため、メモリの増減はほぼ見られず、ボトルネックになっていないと考えられる

検証結果:参照系 (IOPS)



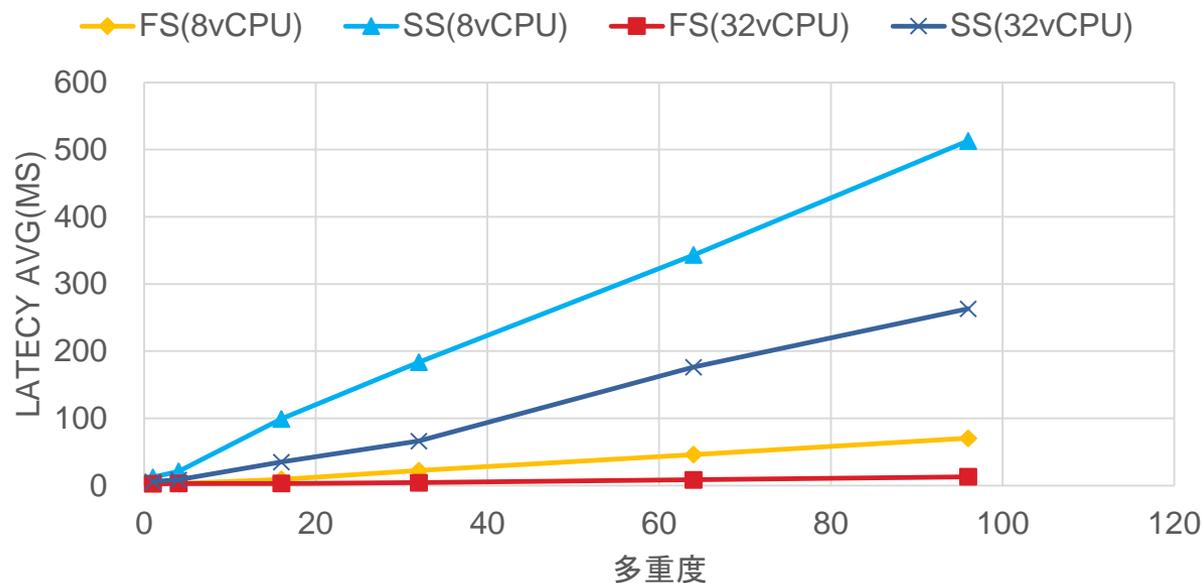
■ 傾向

- FlexibleServer, SingleServerともにIOPSはほぼ0%のため、ボトルネックになっていないと考えられる

検証結果:参照系 (応答時間)

■ pgbenchの結果出力:latency avg (応答時間)

□ NWレイテンシ、DB応答時間を含む



■ 傾向

- FlexibleServerは多重度を上げると値の差が大きくなる
- SingleServerはDBのCPU使用率が高くないのにも関わらず
応答時間が長くなっており、NW構成の仕様に起因していると考えられる

追加検証:DB内のクエリ実行時間

■ pg_stat_statementsを用いてクエリ実行時間を取得

※前スライドとは別計測。多重度:32に固定

- 平均クエリ実行時間 = total_time / calls
- 平均応答時間 … pgbenchのlatency avg

DB	tps	平均応答時間 単位:ms	平均クエリ実行時間 単位:ms	応答時間-クエリ実行時間 (NW到達時間) 単位:ms
FlexibleServer 32vCPU	6840	4.68	3.80	0.88
SingleServer 32vCPU	1714	18.70	1.77	16.93

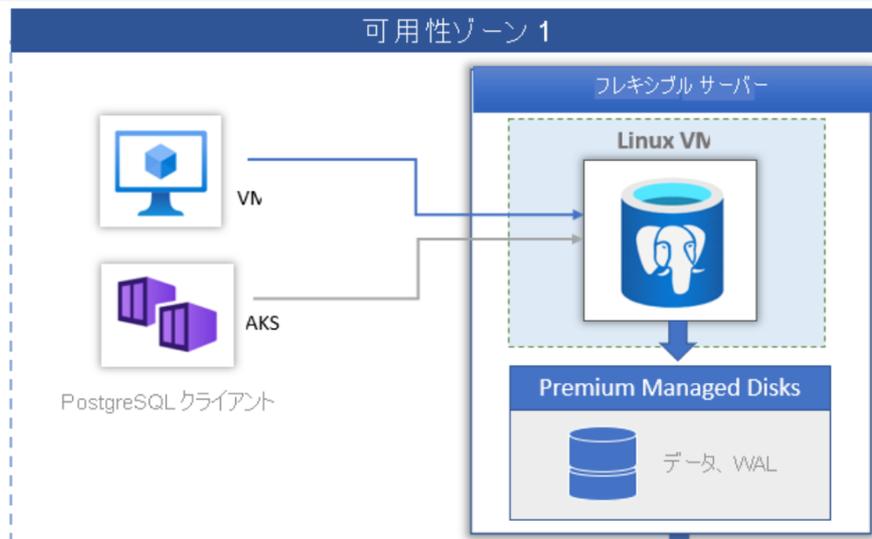
■ 傾向

- FlexibleServerはSingleServerよりクエリ実行時間が長い
 - 多重度:32ではCPU使用率が100%近くまで上昇するため
- SingleServerはNW到達時間が長く、tpsが伸びない原因となっていた
 - NW構成の仕様に起因すると考えられる

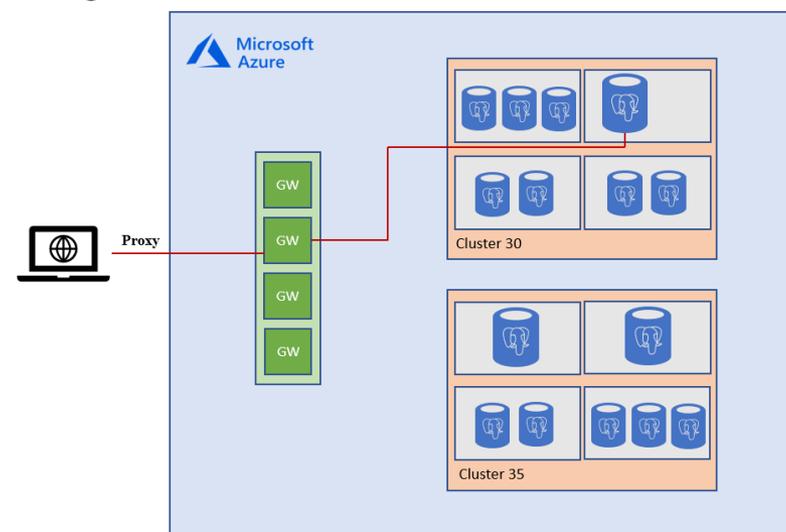
参照系:レイテンシ原因考察

- アーキテクチャ上、SingleServerはDB接続の際にGatewayを経由するが、FlexibleServerは直接接続であるために、NWレイテンシの差が大きいと考えられる

FlexibleServer



SingleServer



ここを経由するため、
DB到達に時間がかかる

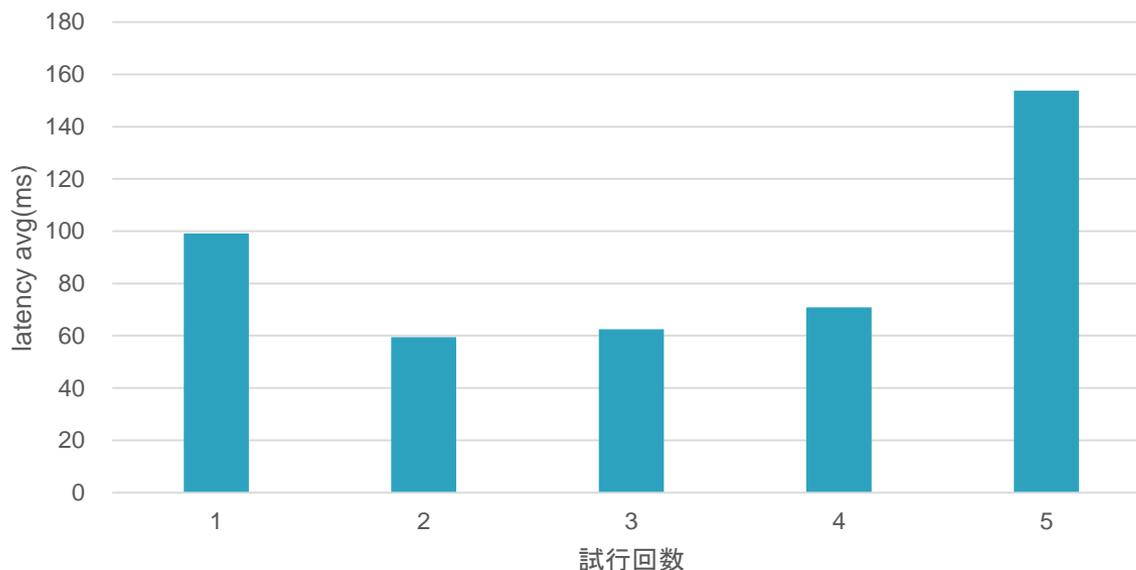
出典: <https://docs.microsoft.com/ja-jp/azure/postgresql/flexible-server/overview>
<https://docs.microsoft.com/ja-jp/azure/postgresql/concepts-connectivity-architecture>

補足検証: SingleServerのNWレイテンシ調査

- SingleServerは可用性ゾーンを指定してインスタンスを作成することができず、作成されたゾーンによってNWレイテンシがばらつく仕様もあるため、どの程度値がばらけるかの追加検証を実施

補足検証: SingleServerのNWレイテンシ調査

- SingleServerにおいて、多重度を16に固定し、インスタンスを5回繰り返し作成し、計測を実施



■ 傾向

- 最大2.5倍程度の応答時間のばらつきを確認
 - インスタンスの可用性ゾーンによるNWのレイテンシの差異だと考えられる

性能テスト結果 (*pgbench*) 更新系処理

検証方法

■ pgbench (更新系) 検証

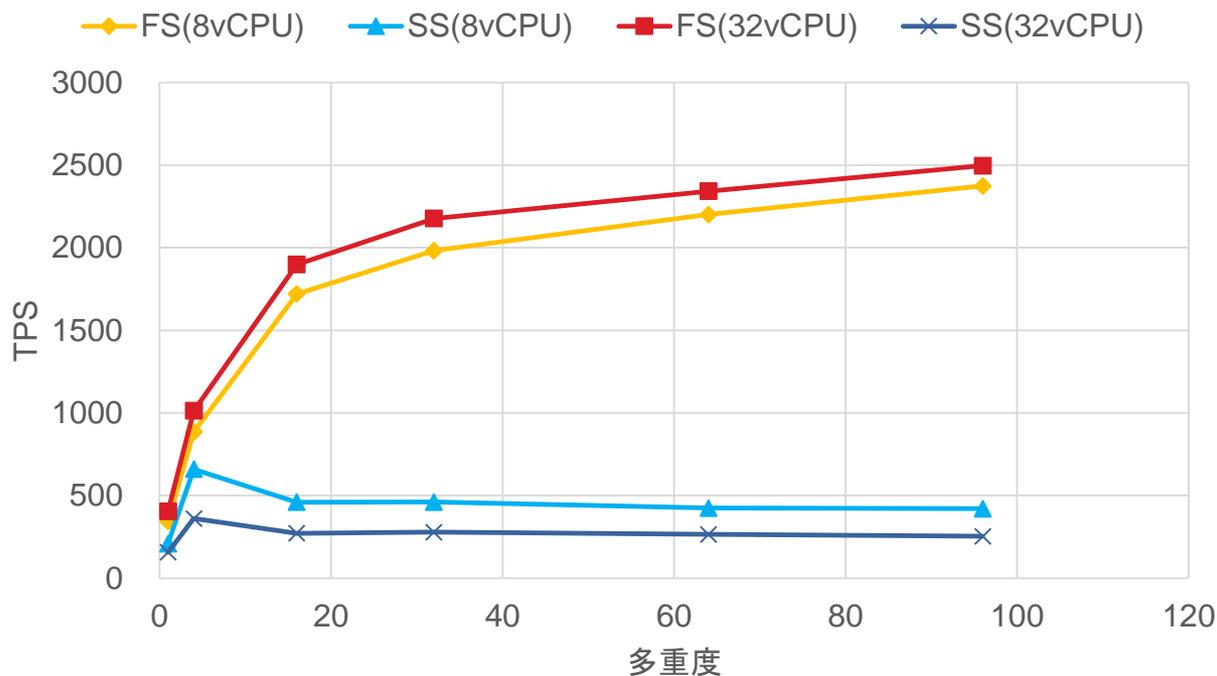
- PGEConsで実施している検証方法を踏襲
- 初期化
 - `pgbench -i -s 2000 [dbname] -F 80`
- キャッシュへの読み込み
 - `SELECT pg_prewarm ('pgbench_accounts');`
- カスタムスクリプト

```
¥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;
```

■ 実行

```
pgbench -r -P 1 -n -c [clients] -j [threads] -f [カスタムクエリ] -T 300 -s 2000
-h [host] -p [port] [dbname]
```

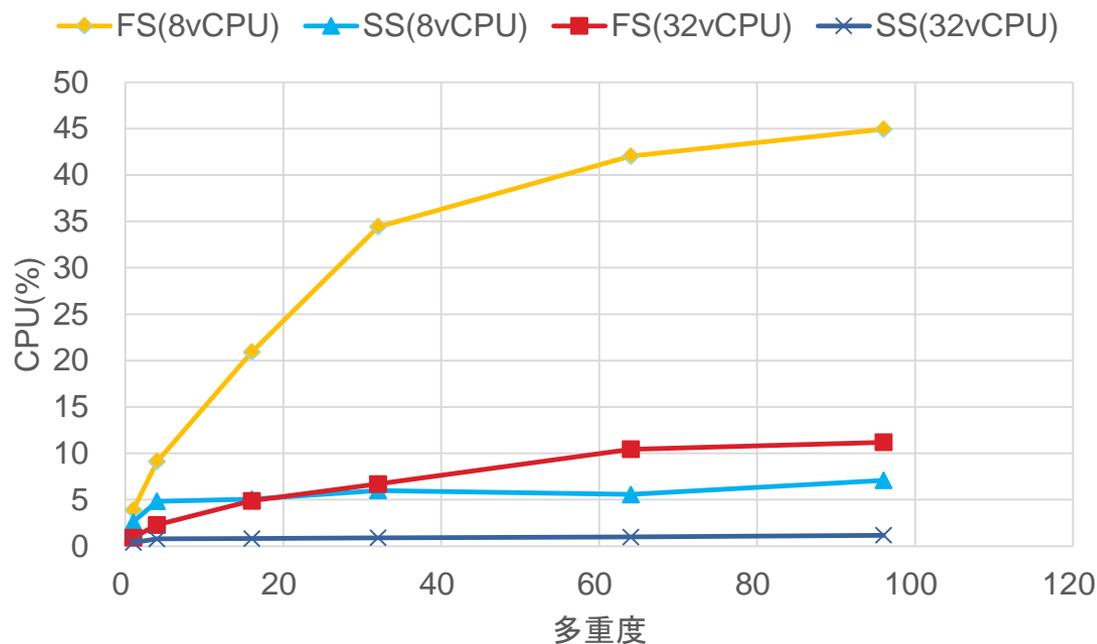
検証結果:更新系 (TPS)



■ 傾向

- 8コア、32コアともに全ての多重度において、
FlexibleServerの方がSingleServerより高いスループット (TPS) を発揮
- FlexibleServer, SingleServerのどちらもvCPUコア数を増加させてもTPSはほぼ変化せず、特にSingleServerはvCPUコア数を増やすと低いTPSとなった
 - 参照系で確認したNWレイテンシのばらつきが原因だと考えられる

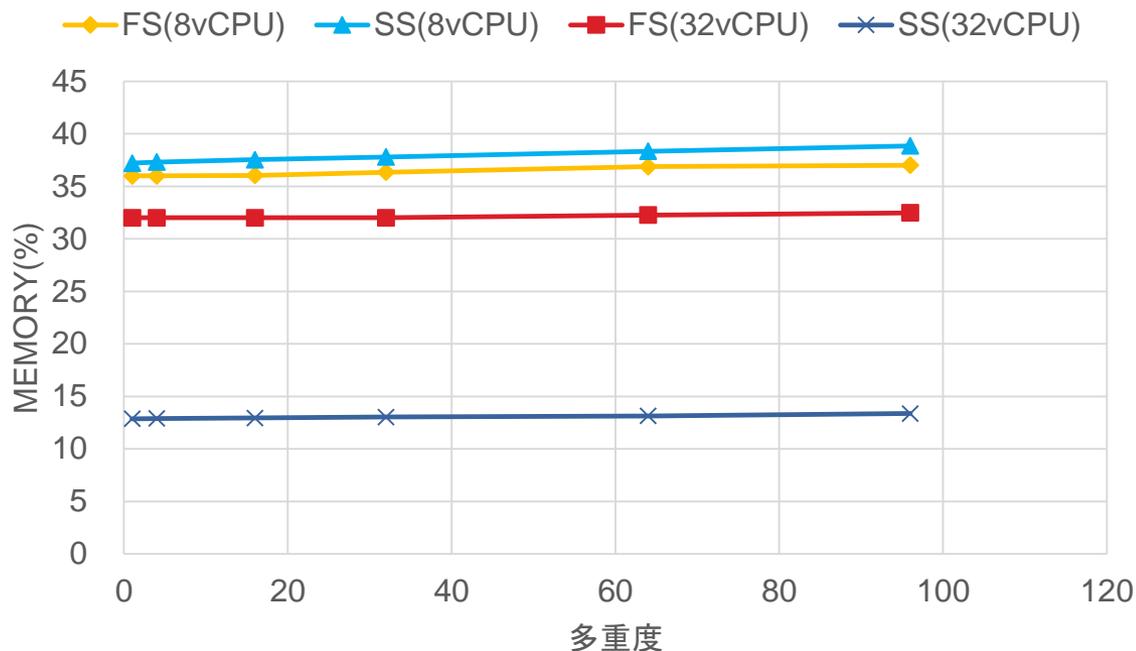
検証結果:更新系 (CPU利用率)



■ 傾向

- FlexibleServer, SingleServerのどちらもCPUには余裕がある状態で、ボトルネックは別の要因であると考えられる

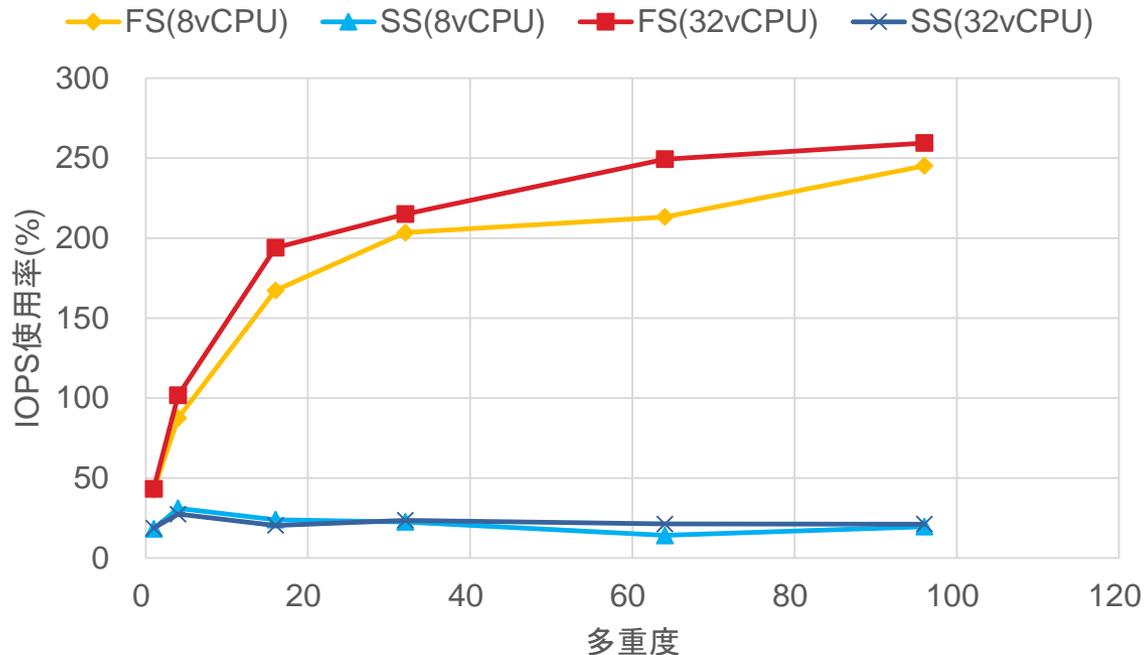
検証結果:更新系 (メモリ)



■ 傾向

- FlexibleServer, SingleServerともに参照系同様、メモリの増減はほぼ見られず、ボトルネックになっていないと考えられる

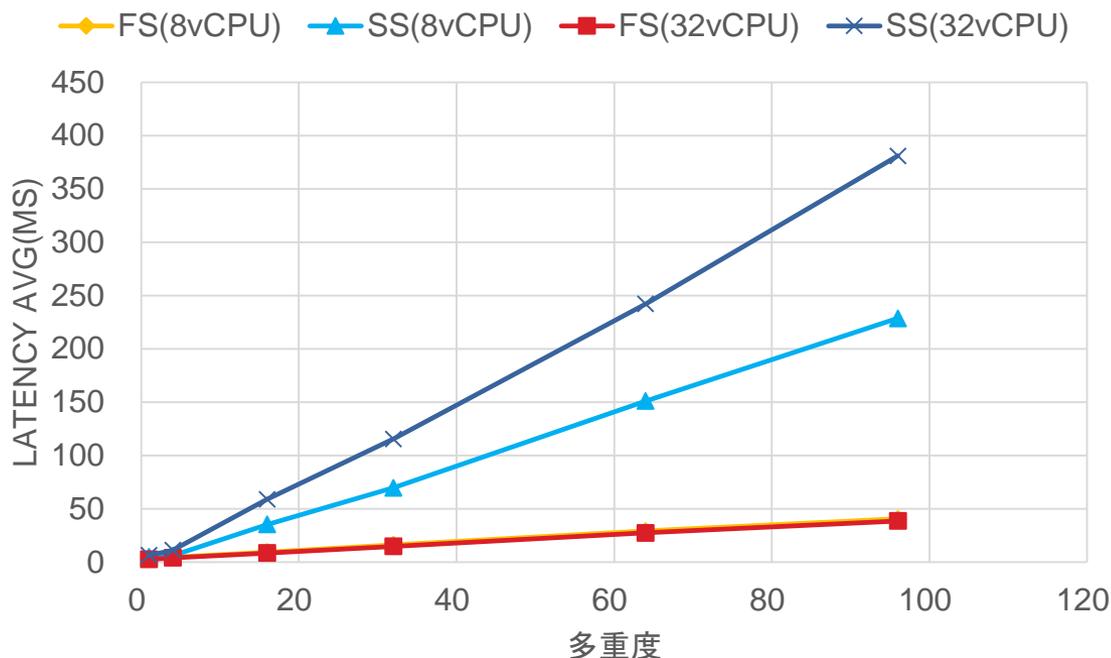
検証結果:更新系 (IOPS利用率)



■ 傾向

- FlexibleServerは多重度16以降、IOPS利用率が100%を超え、最大250%付近まで上昇
 - FlexibleServerのストレージ (IOPS:1100) はIOPS:3500までバースト可能
- SingleServerはIOPSにも余裕が存在

検証結果:更新系 (応答時間)



■ 傾向

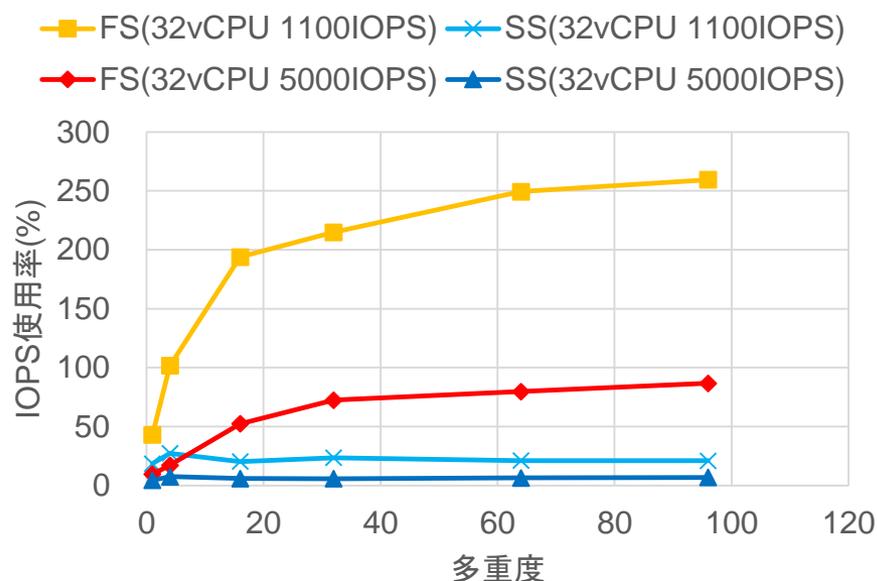
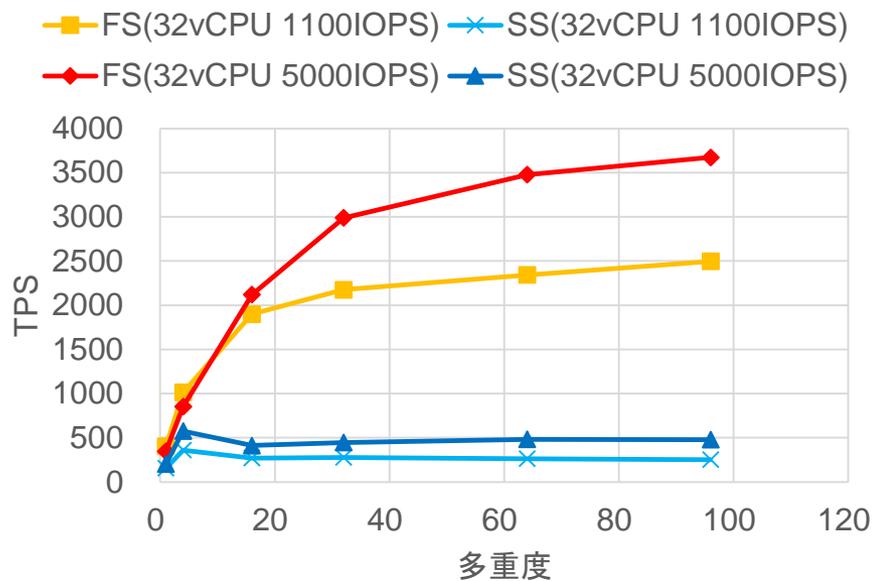
- FlexibleServerは低い応答時間の値となっており、vCPU数を変化させてもほぼ同じ値となった
- SingleServerは参照系同様、NW構成の仕様に起因するレイテンシの増加、ばらつきが確認され、32vCPUの方が悪い性能となった

更新系:IOPS考察

- FlexibleServerはIOPSがバーストした値となっていたため、本来の想定である1100IOPSでの計測が出来ていない
- また、FlexibleServerはIOPSがボトルネックとなりTPSが頭打ちとなっていると考えられるため、IOPSを1100→5000に増加させて計測を実施
 - FlexibleServer : 256GB → 1TB
 - SingleServer : 377GB → 1667GB

追加検証:ストレージサイズ (IOPS) を増加

■ コア数は32vCPUに固定し、IOPSを増加させ計測



■ 傾向

- FlexibleServerはIOPSボトルネックが若干解消され、TPSは増加
 - ただし、IOPS:1100の値はバーストしているため、増加率はIOPSの数値比にはならない
- SingleServerはTPSに変化がないことからボトルネックはIOPSとは無関係
 - DBリソースに余裕があったことから参照系同様、NWLレイテンシが原因だと考えれる

計測結果まとめ (pgbench 参照系)

DB vCPU数	計測項目	多重度毎のTPS					
		1	4	16	32	64	96
8コア	Flexible Server	348	1349	1742	1441	1388	1364
	Single Server	81	191	161	175	186	187
32コア	Flexible Server	387	1317	5514	7432	7603	7428
	Single Server	175	520	520	540	363	365

計測結果まとめ (pgbench 更新系)

DB vCPU数	計測項目	多重度毎のTPS					
		1	4	16	32	64	96
8コア	Flexible Server	342	885	1719	1982	2201	2373
	Single Server	208	659	460	460	424	420
32コア	Flexible Server 1100 IOPS	404	1013	1898	2176	2341	2496
	Flexible Server 5000 IOPS	347	851	2120	2989	3477	3672
	Single Server 1100 IOPS	156	360	271	277	264	253
	Single Server 5000 IOPS	204	576	412	446	483	479

FlexibleServer性能検証結果 (HammerDB)

HammerDBとは

- PostgreSQLを含む複数のDBMSに対応したベンチマーク実行ツール
- TPC-C/TPC-HといったOLTP/OLAPを模したトランザクションを用いたベンチマークが可能
 - 今回はOLTPを想定したTPC-Cを使用
- TPC-Cの検証結果としてNOPMとTPMを採用している
 - NOPM: New Order Per Minutes
 - 1分間に作成できた新規Orderの数
 - TPM: Transaction Per Minutes
 - 1分間のトランザクション発行数

HammerDBでの検証:環境

- **性能測定の大方針 / PostgreSQLバージョン / 負荷掛けマシンスペック はpgbenchに準じる**
 - PostgreSQLパラメータはリソース作成時から変更なしとする
 - リソーススペックは以下とする
 - インスタンスタイプ: メモリ最適化
 - CPU: 8、32
 - ディスクIOPS (サイズ):
 - FlexibleServer: 1,100 (256GB)
 - SingleServer: 768 (256GB)
 - **実施する多重度は以下とする**
 - 1、4、8、16、32、48、64、80、96、112、128、144
 - 8vCPU は性能の頭打ちが見えたため 96までとした

HammerDBでの検証:設定

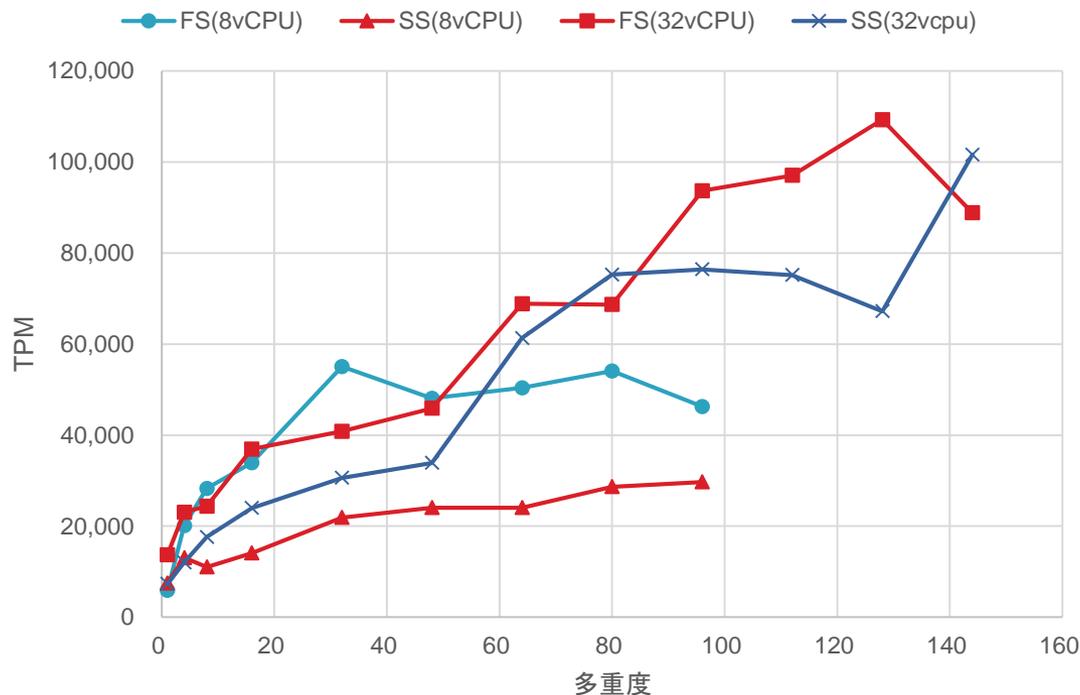
■ HammerDBの主な設定箇所

- **ベンチマーク方式は TPC-C とする**
 - **設定コマンド: dbset bm TPC-C**
- **データサイズは昨年実施を踏まえた設定とする**
 - **設定コマンド: diset tpcc pg_count_ware 200**
 - **データベースサイズ: 40GB**
- **処理対象を特定データのみとしない**
 - **設定コマンド: diset tpcc pg_allwarehouse true**
※デフォルトはfalse(特定データのみを処理対象とする)
- **検証開始直後のデータ読込によるIO影響を避けるため
開始3分は計測対象としない**
 - **設定コマンド: diset tpcc pg_rampup 3**

HammerDBでの検証:手順

- **ベンチマーク処理には更新が含まれるため
試行ごとにデータを初期化する**
- **構築した直後の環境を退避した後
以下の手順を3セット繰り返す**
 1. **データ投入、バキューム処理**
 2. **ベンチマーク実施/測定**
 3. **データ削除**

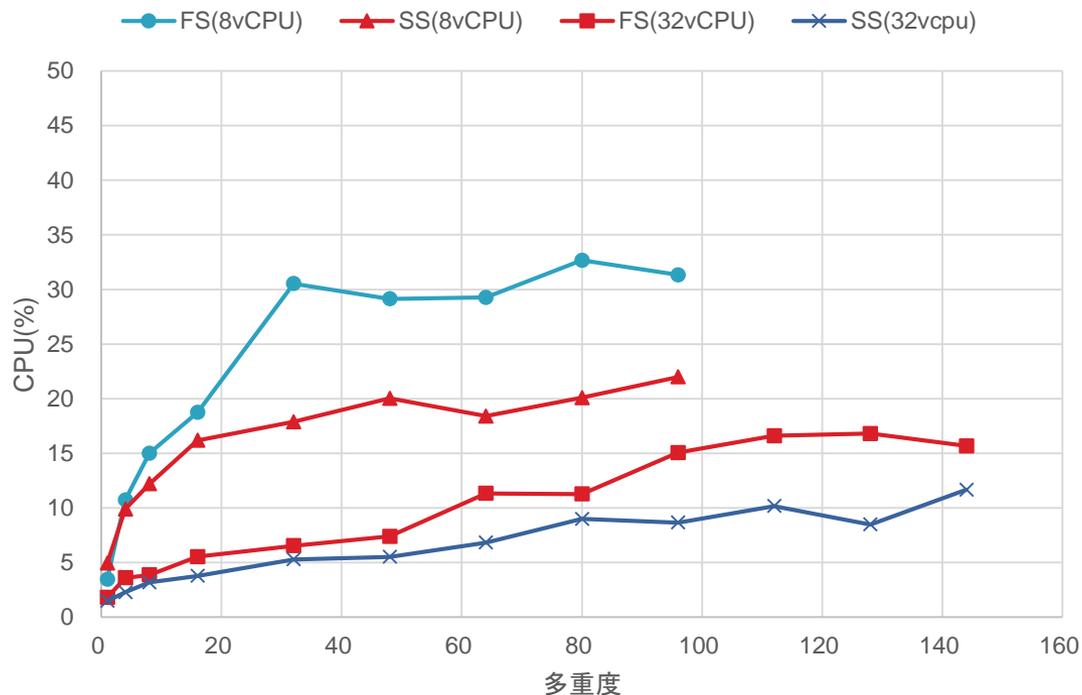
検証結果:TPM



■ 傾向

- 概ねFlexibleServerの方がSingleServerよりも良い性能となった
- FlexibleServerとSingleServerの同一CPU間でpgbenchのような性能差 (参照系: 最大20倍、更新系: 最大5倍) は見られない

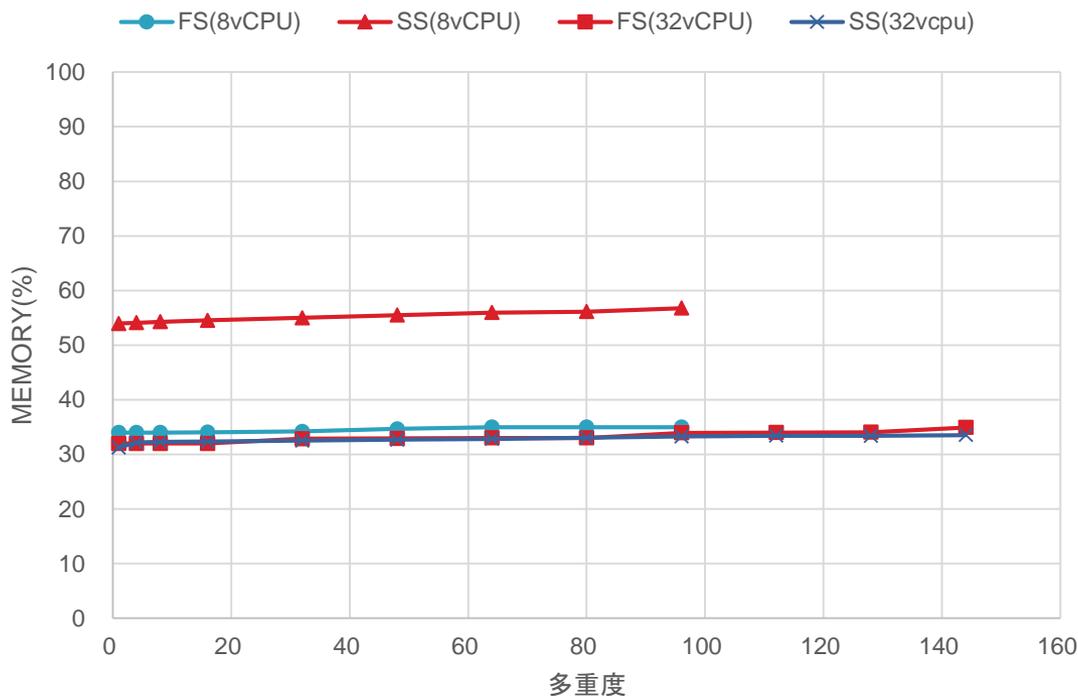
検証結果:CPU利用率



■ 傾向

- FlexibleServer、SingleServer、CPU数の違いに関わらず多重度と比較してCPUの利用率は低い
- 性能の推移とほぼ連動している

検証結果:メモリ利用率



■ 傾向

□ FlexibleServer、SingleServer、vCPUの違いに関わらず
メモリ利用率の増減はほぼ見られない

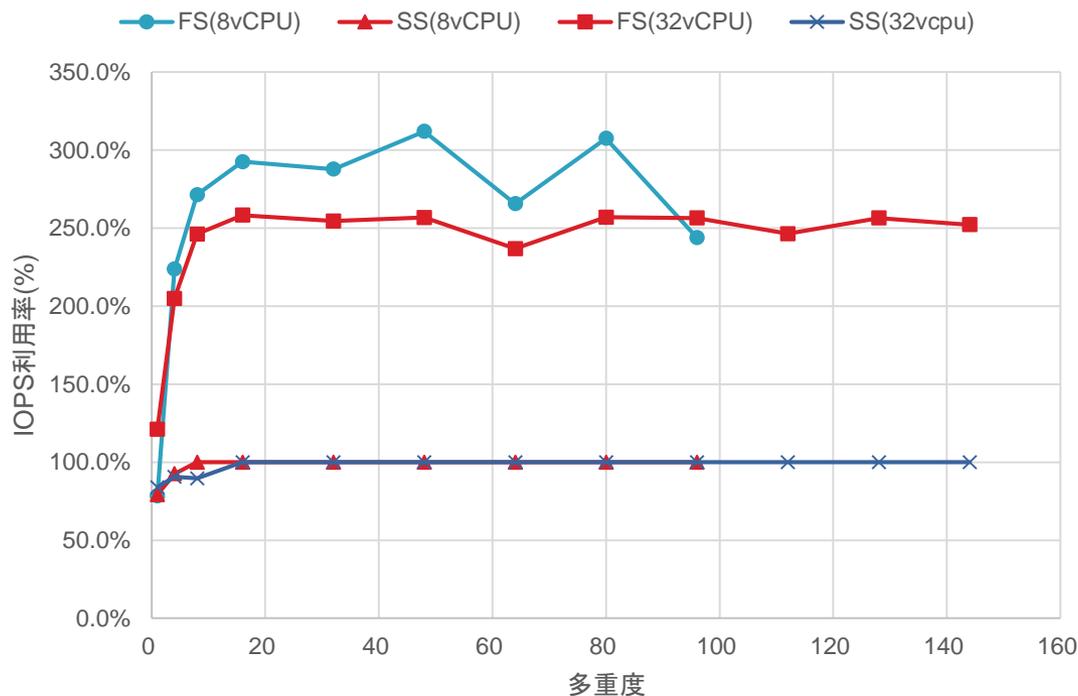
■ 接続によるwork_memの消費分が増加している

□ work_mem: 4MB、92接続: 368MB、144接続: 576MB

□ インスタンスのメモリ:

FS (8vCPU) : 64GB、SS (8vCPU) : 80GB、FS (32vCPU) : 256GB、SS (32vCPU) : 320GB

検証結果:IOPS利用率



FS 1,100IOPS (256GB)
SS 768IOPS (256GB)

■ 傾向

- FlexibleServerは100%を超え最大300%付近まで上昇している
 - マネージドディスクのバースト機能が働いている
- SingleServerは100%になるが設定値を超えることはない

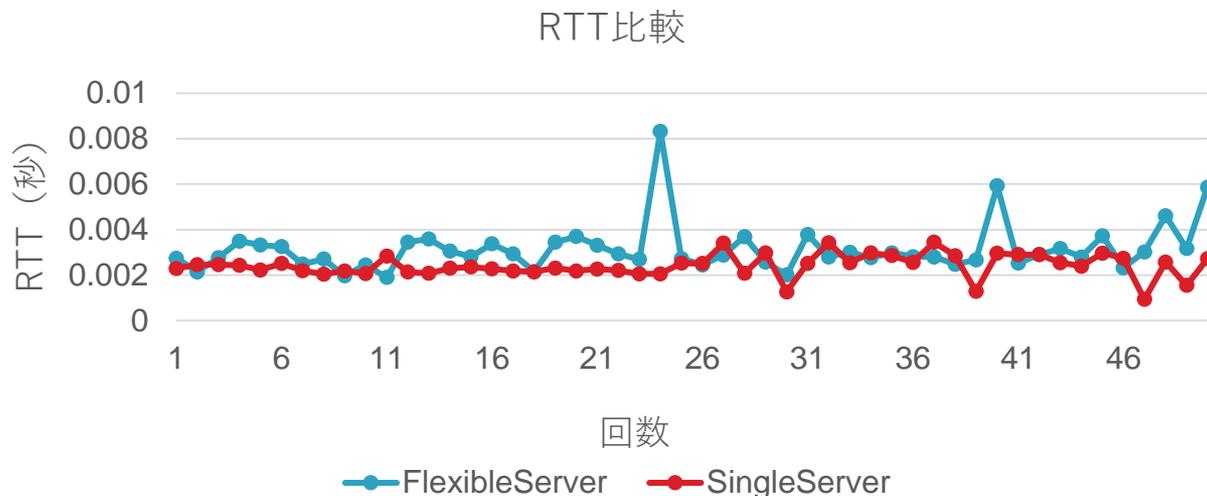
HammerDB結果考察

- FlexibleServerの方がSingleServerより性能が良いがIOを使い切るワークロードの場合pgbenchほどの差がない
 - pgbenchで発行するカスタムクエリと比べHammerDBは複雑なSQLを発行しており処理時間が長くなる※ことでRTTの影響が少なくなると推測(本環境でのRTTの計測結果は次ページを参照)
 - 性能のボトルネックはIOと推測される
 - CPUの利用率が低く、またメモリの利用率は一定であるがIOの利用率が100%以上となっている

※: HammerDBではSQL処理時間を計測する機能を持たないため推測となる

補足: HammerDB検証環境でのRTT

- HammerDB検証環境にて、SQLを発行してから最初のACKパッケージが返ってくるまでの時間を計測
 - 50回実施しての平均値がFlexibleServerは約3.1ms、SingleServerは約2.4msとFlexibleServerがやや高い結果となった
 - 前述の通りFlexibleServerはRTTが小さくなると公表されており、本来なら実施結果はFlexibleServerがSingleServerよりもRTTが小さくなるはずであるが、このような結果となった原因については不明



追加検証

■ IOPSを増加した場合の性能向上を検証する

- IOPSがバースト後含め上限に達しており性能のボトルネックになっていると推測
- IOPSを以下の数値に増加することで性能が向上することを確認する

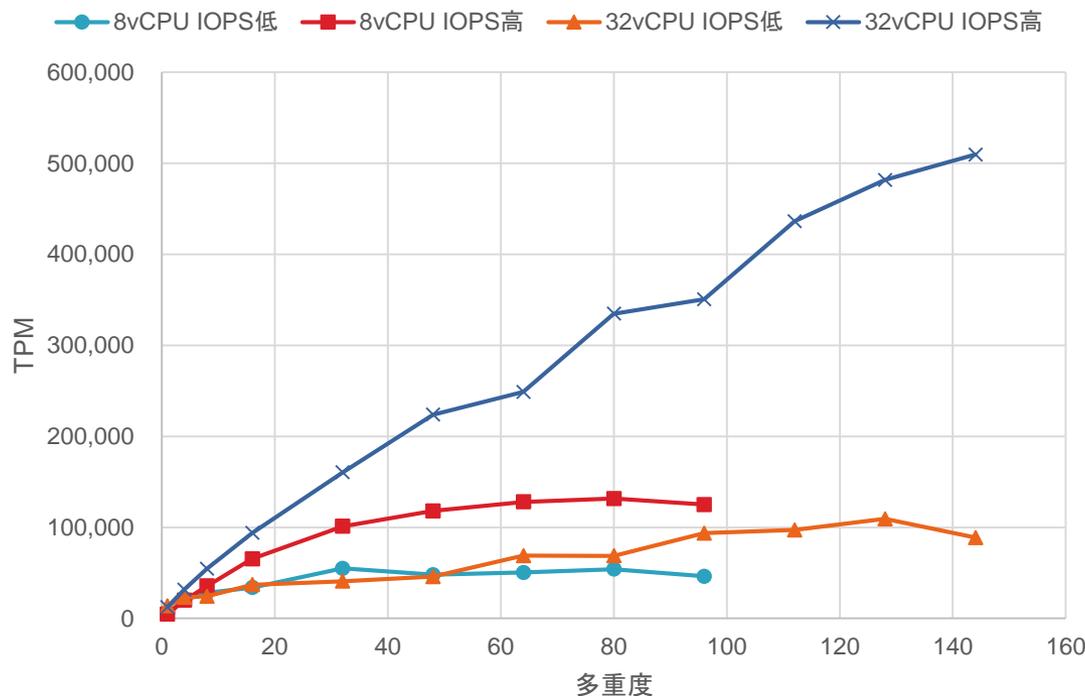
■ FlexibleServer

- 8vCPU: 1,100 ⇒ 12,800
- 32vCPU: 1,100 ⇒ 18,000

■ SingleServer

- 8vCPU: 768 ⇒ 5,133
- 32vCPU: 768 ⇒ 10,080

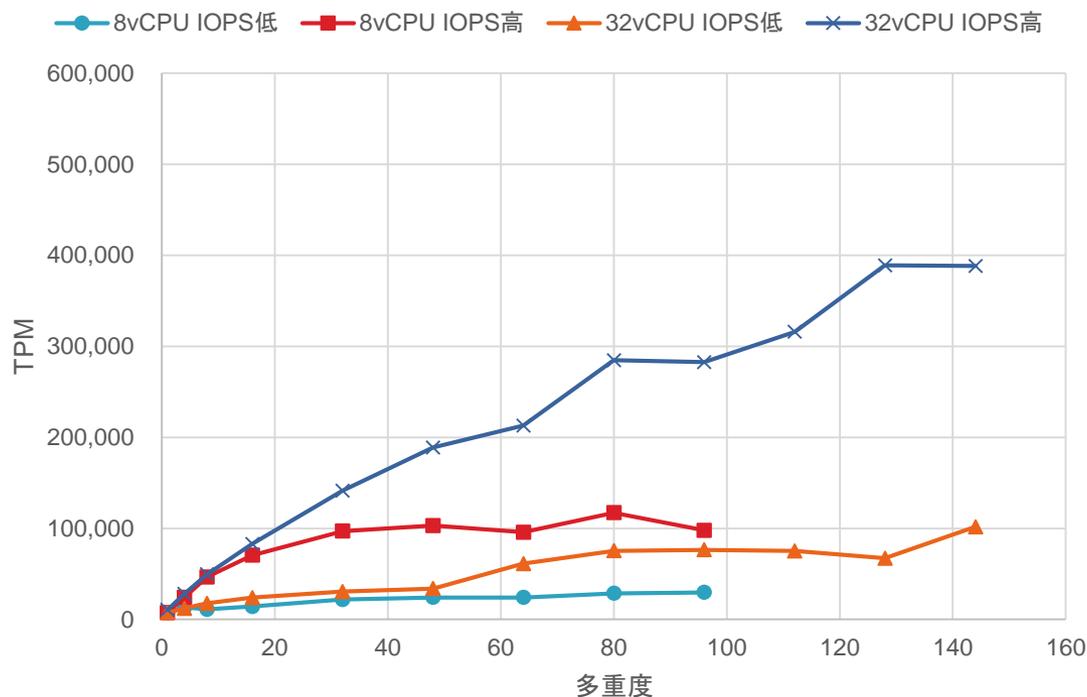
追加検証結果:TPM (FlexibleServer IOPS増速前後)



■ 傾向

- IOPS増速後の性能が増加前と比べて向上している
- 8cpuはIOPS増速前後でともに性能が頭打ちしている
- 32cpuは
 - IOPS増速後は多重度144でもまだ頭打ちしていない
 - IOPS増速前後では多重度に比例して性能差が大きくなっている

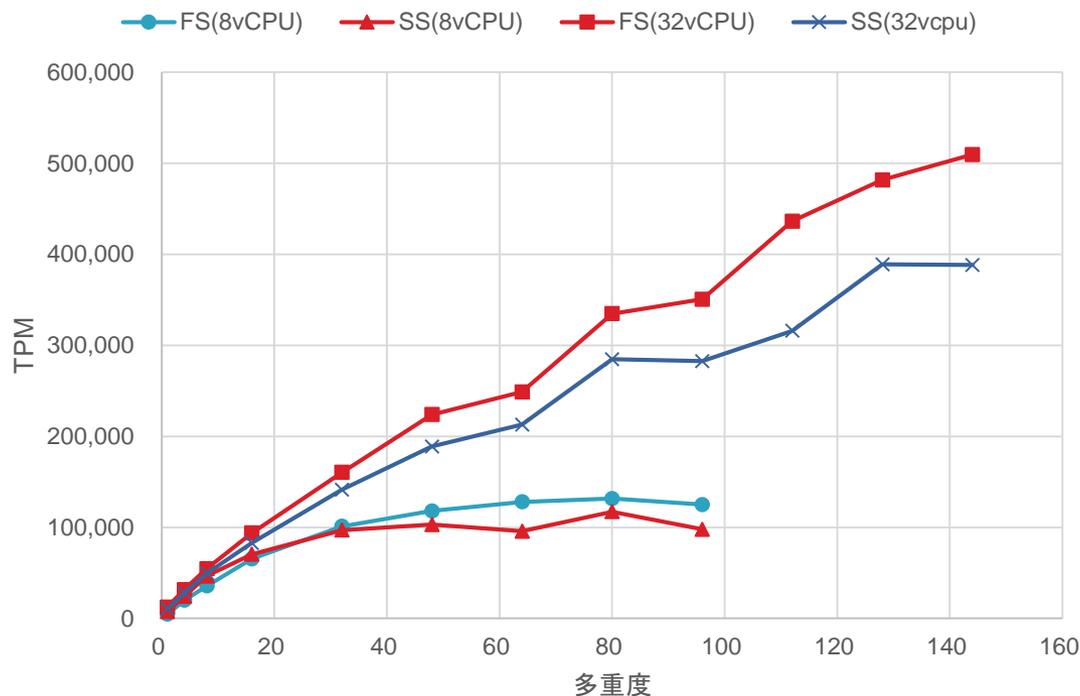
追加検証結果:TPM (SingleServer IOPS増速前後)



■ 傾向

- FlexibleServerと同じ傾向がみられる

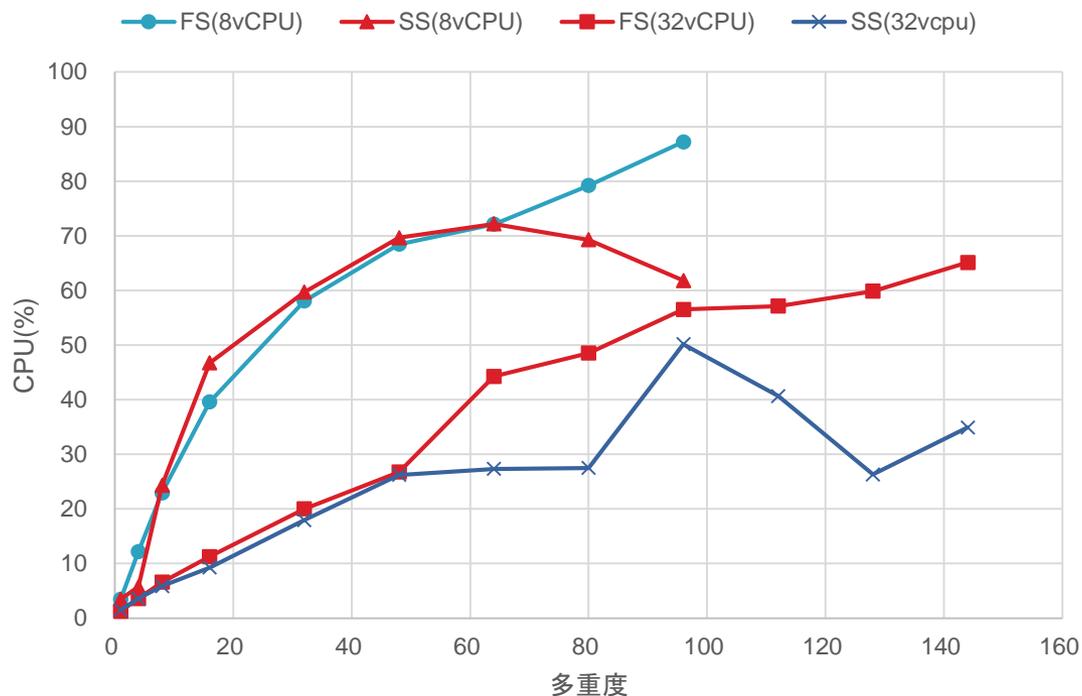
追加検証結果:TPM (IOPS増速後リソース間比較)



■ 傾向

- 処理性能が向上した点を除けば、IOPS増速前と同じ傾向である
 - 同一CPUで比較するとFlexibleServerの方がSingleServerより性能が良い
 - pgbenchのような差は見られなかった

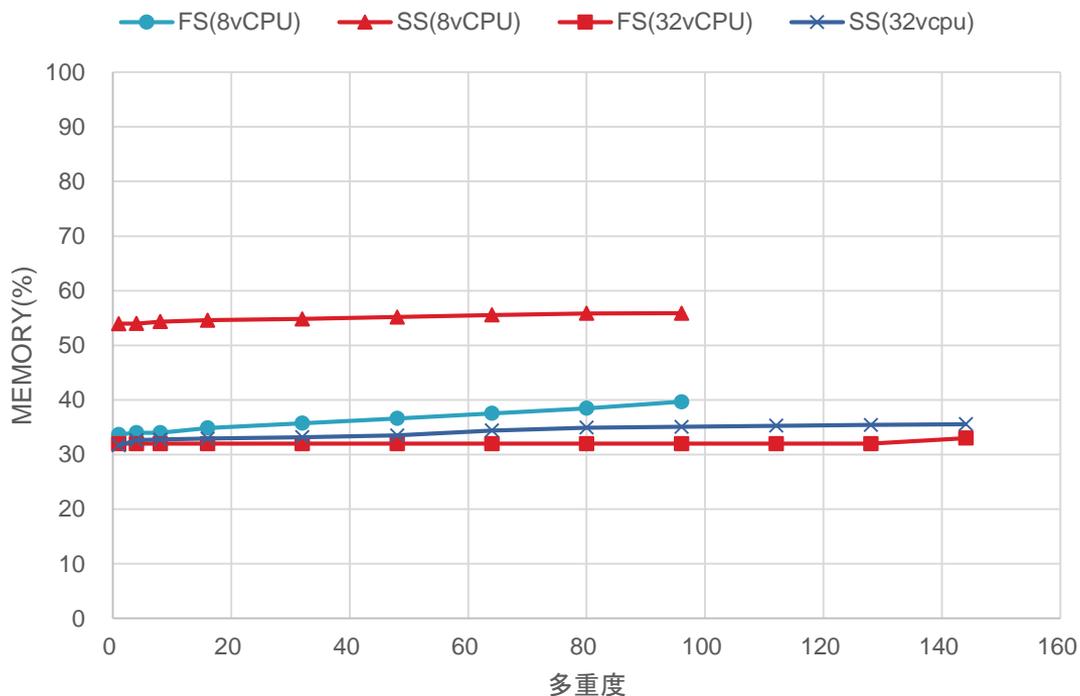
追加検証結果:CPU使用率



■ 傾向

- 多重度が増えるに従い使用率が上がるが
検証した多重度の範囲では上限まで達していない

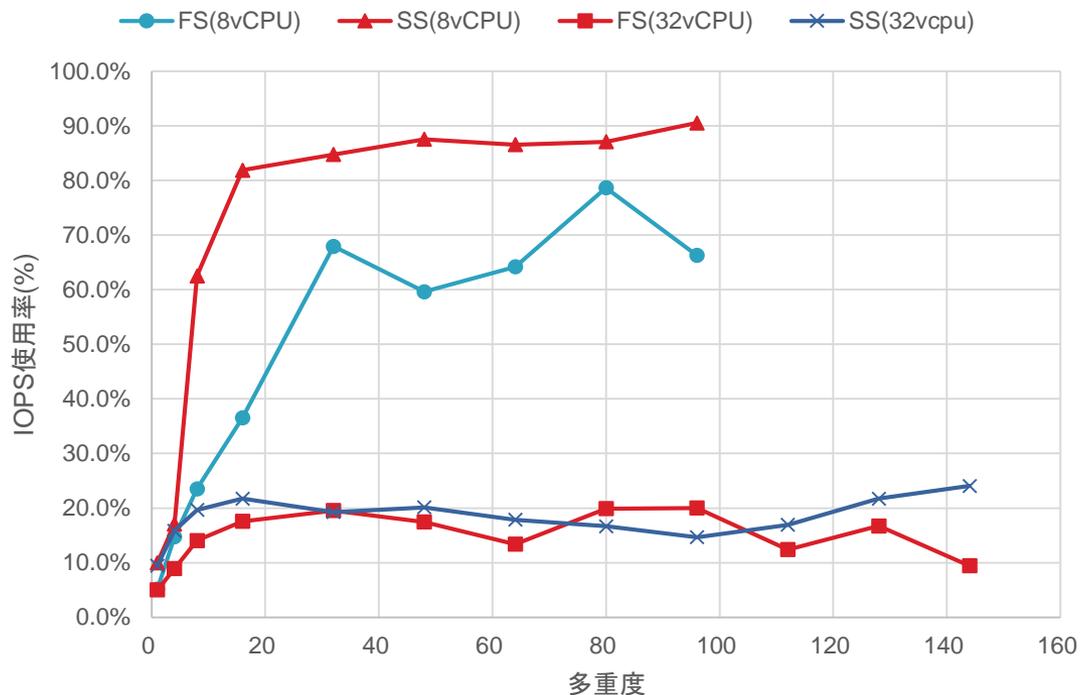
追加検証結果:メモリ使用率



■ 傾向

- どのリソースも、多重度によらずほぼ一定となっている
 - 接続によるwork_memの消費分が増加している

追加検証結果:IOPS使用率



■ 傾向

- どのリソースも、
 - 上限に達していない
 - 多重度が上がると使用率がほぼ頭打ちとなる

HammerDB追加検証結果考察 1/2

- **どのパターンでもIOPS増速により性能向上を確認できた**
 - FlexibleServerの方がSingleServerより性能が良い点はIOPS増速前と変わらず
 - **IOPS増速後もFlexibleServerとSingleServerの間でpgbenchほどの性能差が見られなかった**
 - pgbenchの性能差はRTTの差（SingleServerの方がFlexibleServerよりも長い）によるものと推測されるがHammerDB検証環境でRTTを実測すると差が見られずこれが大きな性能差が生じない原因と推測される
- (次ページへ続く)

HammerDB追加検証結果考察 2/2

- IOPS増速後もFlexibleServerとSingleServerの間でpgbenchほどの性能差が見られなかった
 - Microsoft社のRTT測定ではFlexibleServerがSingleServerの1/5程度であったが、なぜHammerDB検証環境ではRTT測定結果が異なる（FlexibleServerとSingleServerが同程度となった）のか原因の解明までは至らなかった
 - 追加検証で見られたFlexibleServerとSingleServerの性能差はPostgreSQLが動作する基盤の性能差と考えられる

検証結果総括

総括

- 全ての測定ケースにおいてFlexibleServerの方がSingleServerより良い性能 (スループット) となった。
 - FlexibleServerはCPUやIOPSがボトルネックとなるが、スケールアップさせることによって性能も向上することを確認
 - ただし、ストレージがバーストするので想定より高いスループットを発揮することはあるが、実際の伸び幅については注意が必要
- SingleServerはDB接続周りの仕様上、FlexibleServerと比べるとNWレイテンシが比較的高くなる
 - 今回のケースのような簡易なクエリにおいてはボトルネックとなった
- 小規模クエリ×多数の同時アクセスのワークロードにおいて、FlexibleServerはSingleServerより性能のスケールアップをさせやすいと考えられる

補足：RTT (Round-Trip Time) の影響による FlexibleServerとSingleServerの性能差について 1/2

■ RTTは“FlexibleServer < SingleServer”となる

□ サービスによりネットワーク構成に差がある

- クライアントからの接続時、SingleServerはGWを経由するがFlexibleServerは直接接続する
- FlexibleServerが可用性ゾーンを指定できるのに対しSingleServerは可用性ゾーンを指定できない(ランダムとなる)ためクライアントとSingleServerが異なる可用性ゾーンとなる可能性がある

□ Microsoftにて公開されているRTT※

- SingleServer: 2.76ms
- FlexibleServer: 0.54ms

※ Modernize your data on Azure Database for Postgres & MySQL | Data Platform Summit 2020 | Sunil Agarwal & Parikshit Savjani
(<https://speakerdeck.com/azuredbpostgres/modernize-your-data-on-azure-database-for-postgres-and-mysql-data-platform-summit-2020-sunil-agarwal-and-parikshit-savjani?slide=15>)

補足：RTT (Round-Trip Time) の影響による FlexibleServerとSingleServerの性能差について 2/2

■ SQL実行に対するRTTの影響

- SQL実行時のネットワークパケットの流れは右図の通り
そのため、SQL実行のレイテンシは最低でも以下の時間を要する

RTT + サーバ側のSQL処理時間 (右図③)

- 1接続における単位時間あたりのSQL実行回数の上限※はRTTの影響を受ける
 - SingleServer (RTT=2.76ms): 362回/秒 ($1/2.76 \times 1,000$)
 - FlexibleServer (RTT=0.54ms): 1,851回/秒 ($1/0.54 \times 1,000$)
- 接続を多重化すれば性能は向上するが、多重化には限界があり特定の多重度以上は逆にスループットが下がってしまう事が知られている

※ “サーバ上でのSQL処理時間がほとんど0である”と仮定した場合

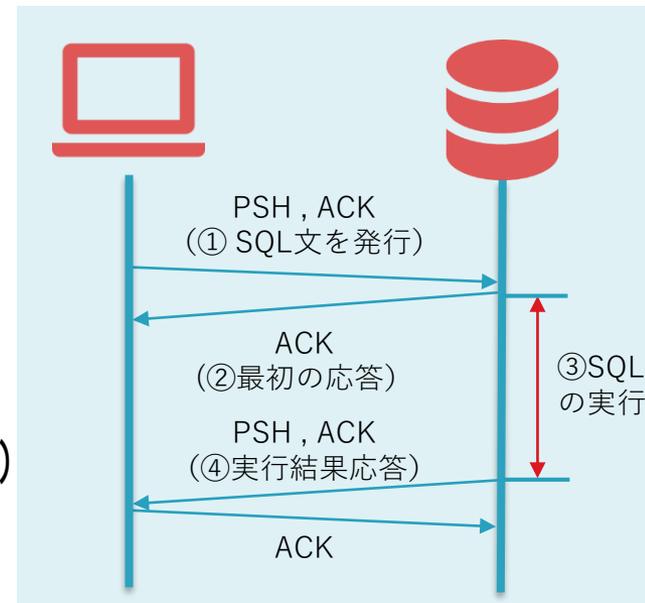


図: ネットワークパケットの流れ

ライセンス

本作品はCC-BYライセンスによって許諾されています。ライセンスの内容を知りたい方は[こちら](#)でご確認ください。文書の内容、表記に関する誤り、ご要望、感想等につきましては、[PGEConsのサイト](#)を通じてお寄せいただきますようお願いいたします。

- Amazon Web Services、“Powered by Amazon Web Services”ロゴ、Amazon EC2、Amazon S3、Amazon Relational Database Service (Amazon RDS) およびAmazon Auroraは、米国その他の諸国における、Amazon.com, Inc.またはその関連会社の商標です。
- IBMおよびDb2は、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。
- Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。
- Red HatおよびShadowman logoは、米国およびその他の国におけるRed Hat,Inc.の商標または登録商標です。
- Microsoft、Microsoft Azure、Windows Server、SQL Server、米国 Microsoft Corporationの米国及びその他の国における登録商標または商標です。
- MySQLは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
- Oracleは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
- PostgreSQLは、PostgreSQL Community Association of Canadaのカナダにおける登録商標およびその他の国における商標です。
- TPC, TPC Benchmark, TPC-B, TPC-C, TPC-E, tpmC, TPC-H, TPC-DS, QphHは米国Transaction Processing Performance Councilの商標です。
- その他、本資料に記載されている社名及び商品名はそれぞれ各社が 商標または登録商標として使用している場合があります。

著者

(企業・団体名順)

版	所属企業・団体名	部署名	氏名	
第1.0版 (2021年度 WG3)	日鉄ソリューションズ株式会社	流通・サービスソリューション事業本部 アドバンステクノロジー部	伊藤 春	
			永井 光	
			小野 晋也	
	ヤマトシステム開発株式会社	システム本部	鳥居 英明	
			システム本部 技術開発部	毛呂 良寛
			IT基盤本部 運用技術部	藤井 大和