

わかる PostgreSQL の設定とチューニング(後編) ～ チューニングの勘どころ ～

日本 PostgreSQL ユーザ会
片岡 裕生



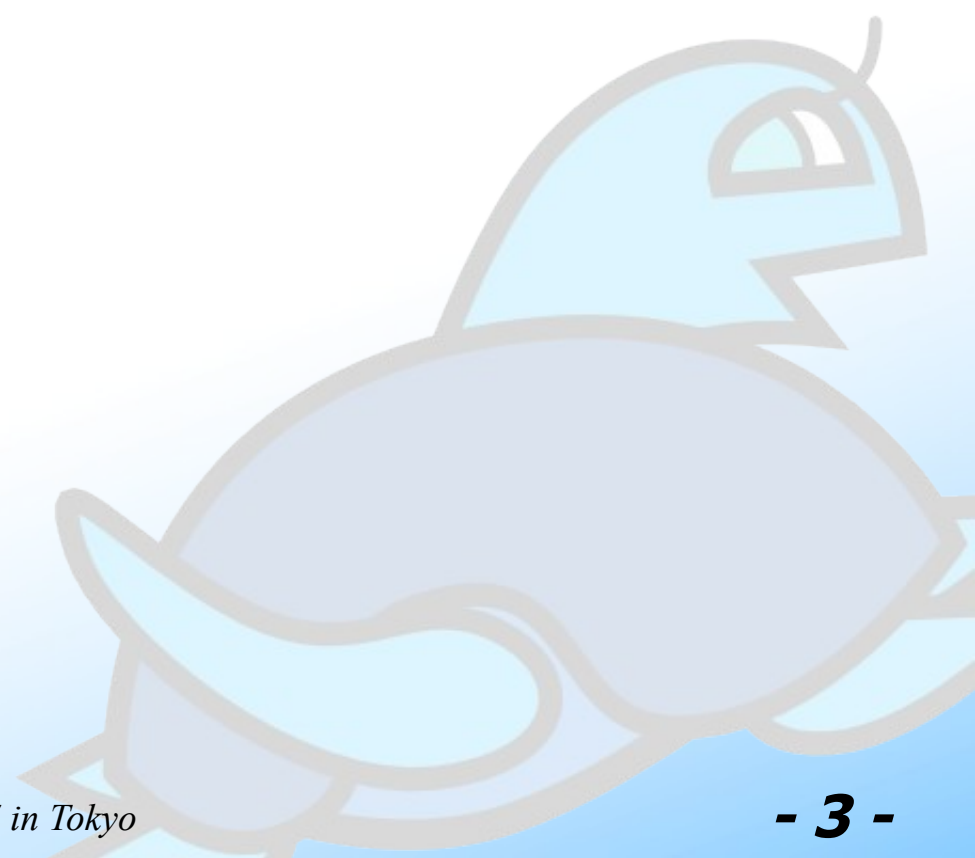
PostgreSQL をデフォルトのまま使っていませんか？

- PostgreSQL をインストールしたただけの状態とは…？
 - 設定内容はミニマム
 - メモリはできるだけ使わない
 - HDD もできるだけ使わない
 - ロースペックのマシン環境でも動作するように。

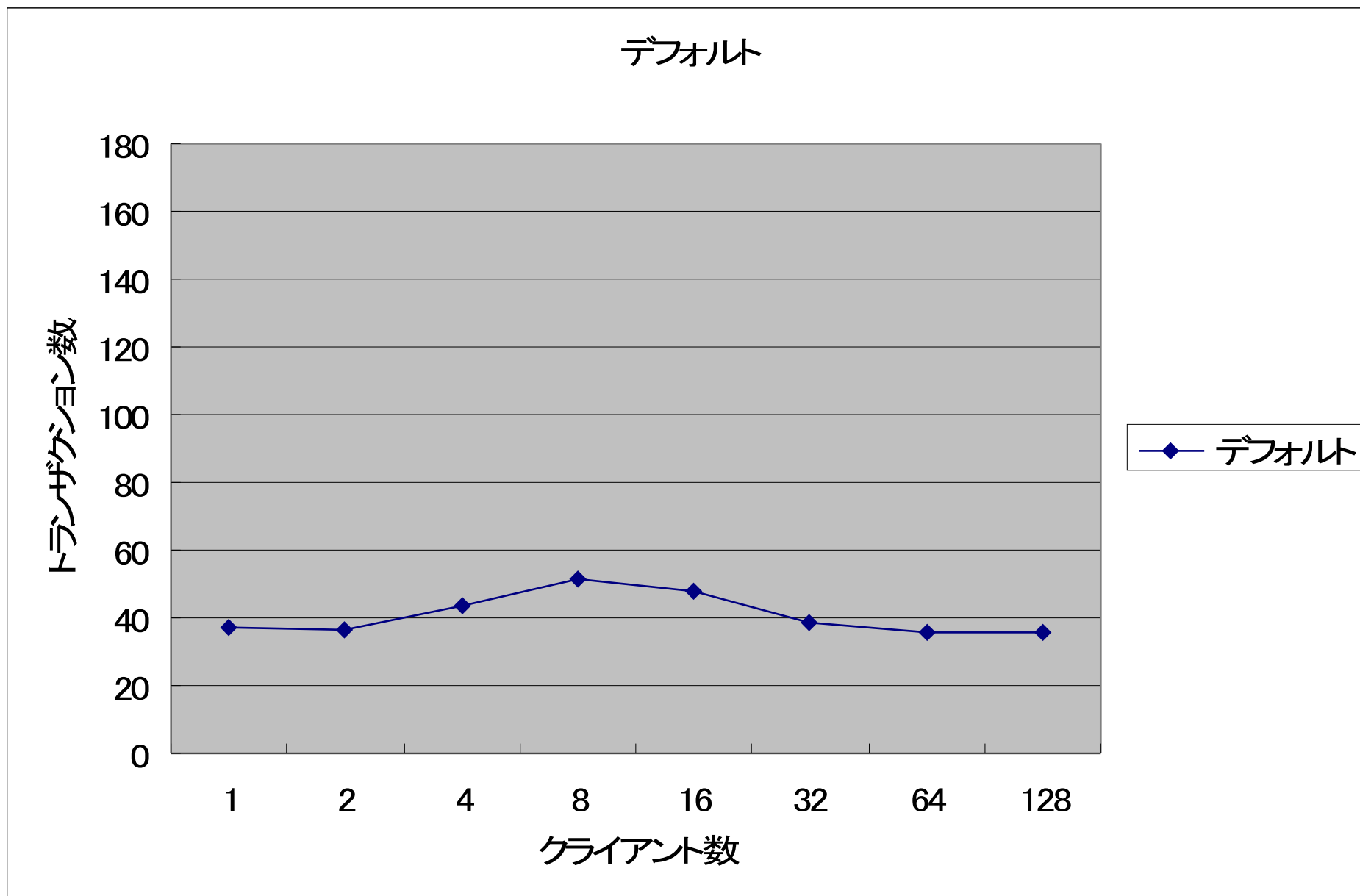
わずかな設定で PostgreSQL の性能が見違える！

スライド中のベンチマークについて

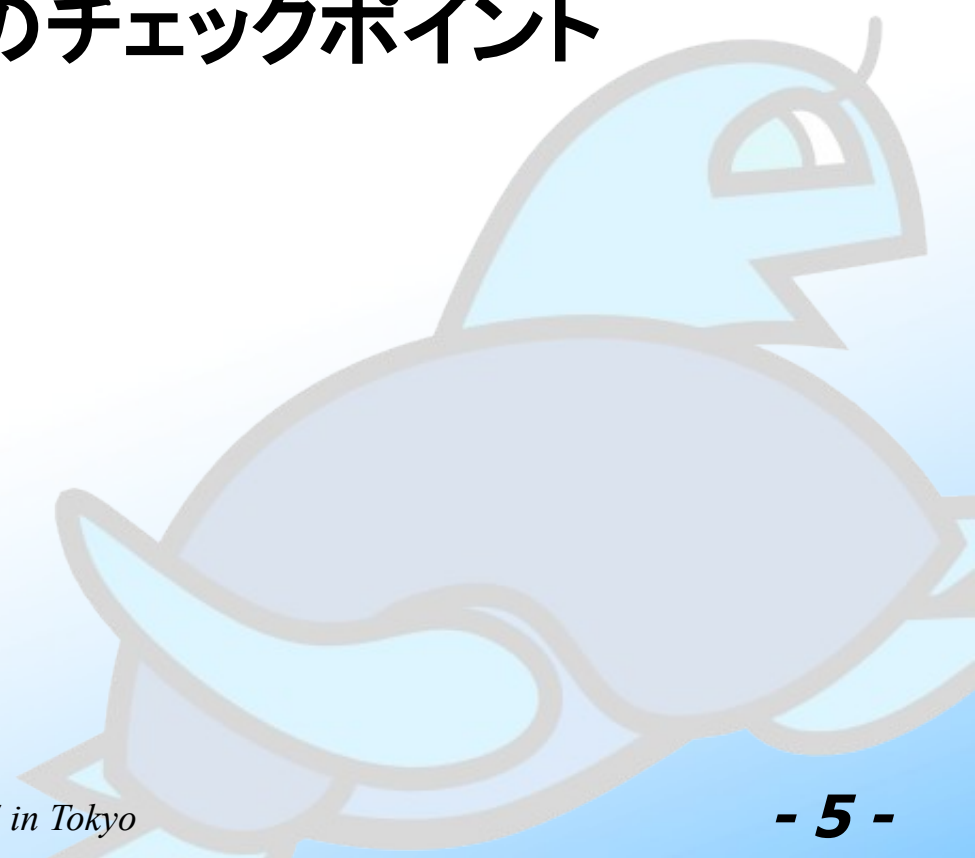
- `pgbench` によるベンチマークを実施
 - 更新割合の高いベンチマーク
 - `select×1 + update×3 + insert×1`



デフォルト設定でのパフォーマンス

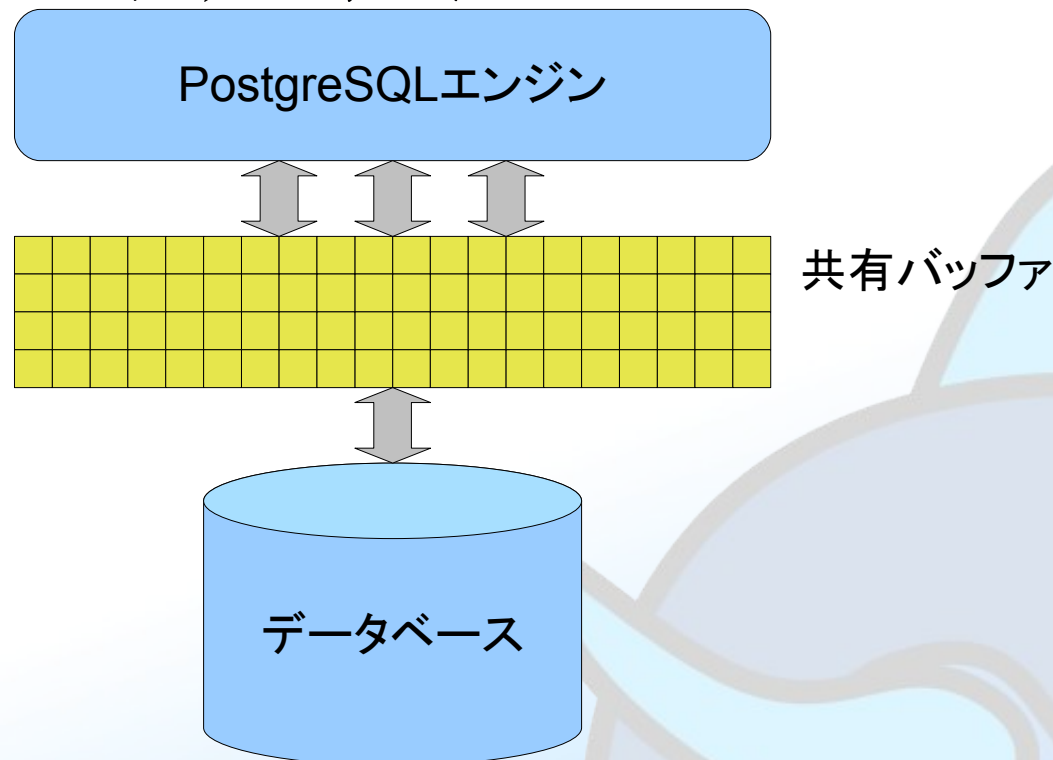


PostgreSQL 設定のチェックポイント



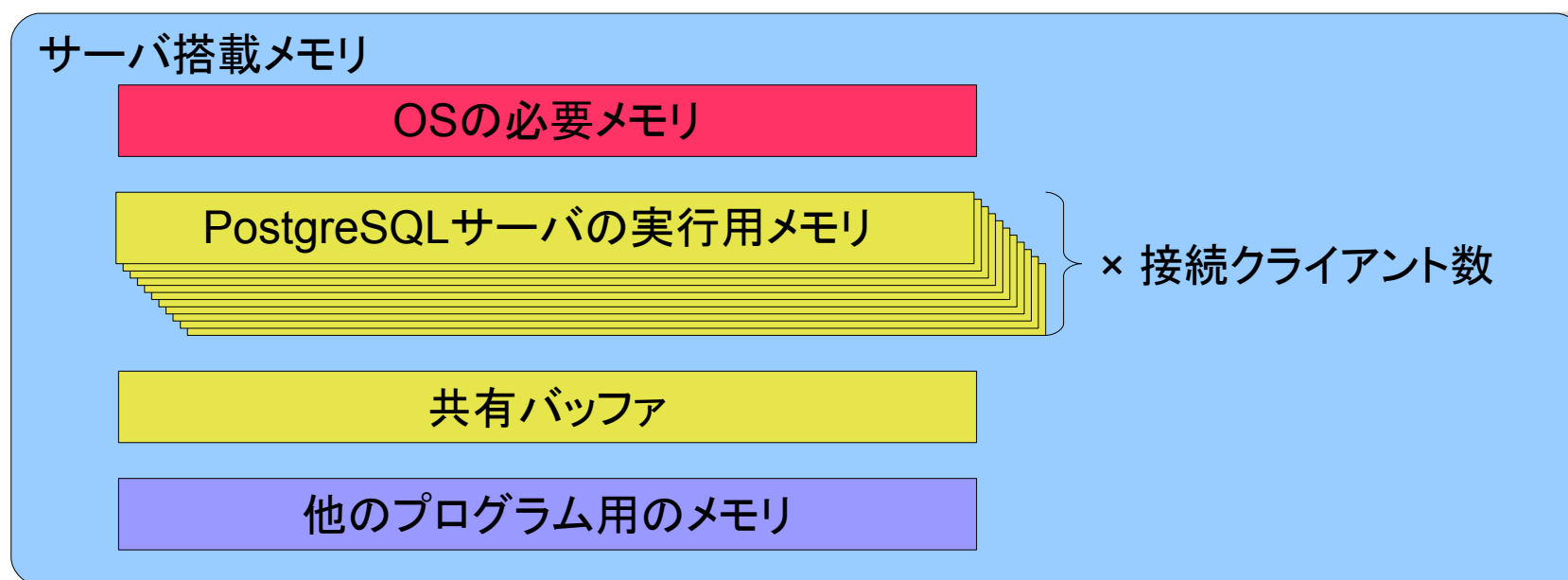
チェック1:共有バッファ

- PostgreSQLが独自に用意しているディスクキャッシュ
- デフォルトは32MB
 - ほとんどの場合、32MBでは少なすぎる。
 - 古いバージョンではデフォルトで256KBページしかない。

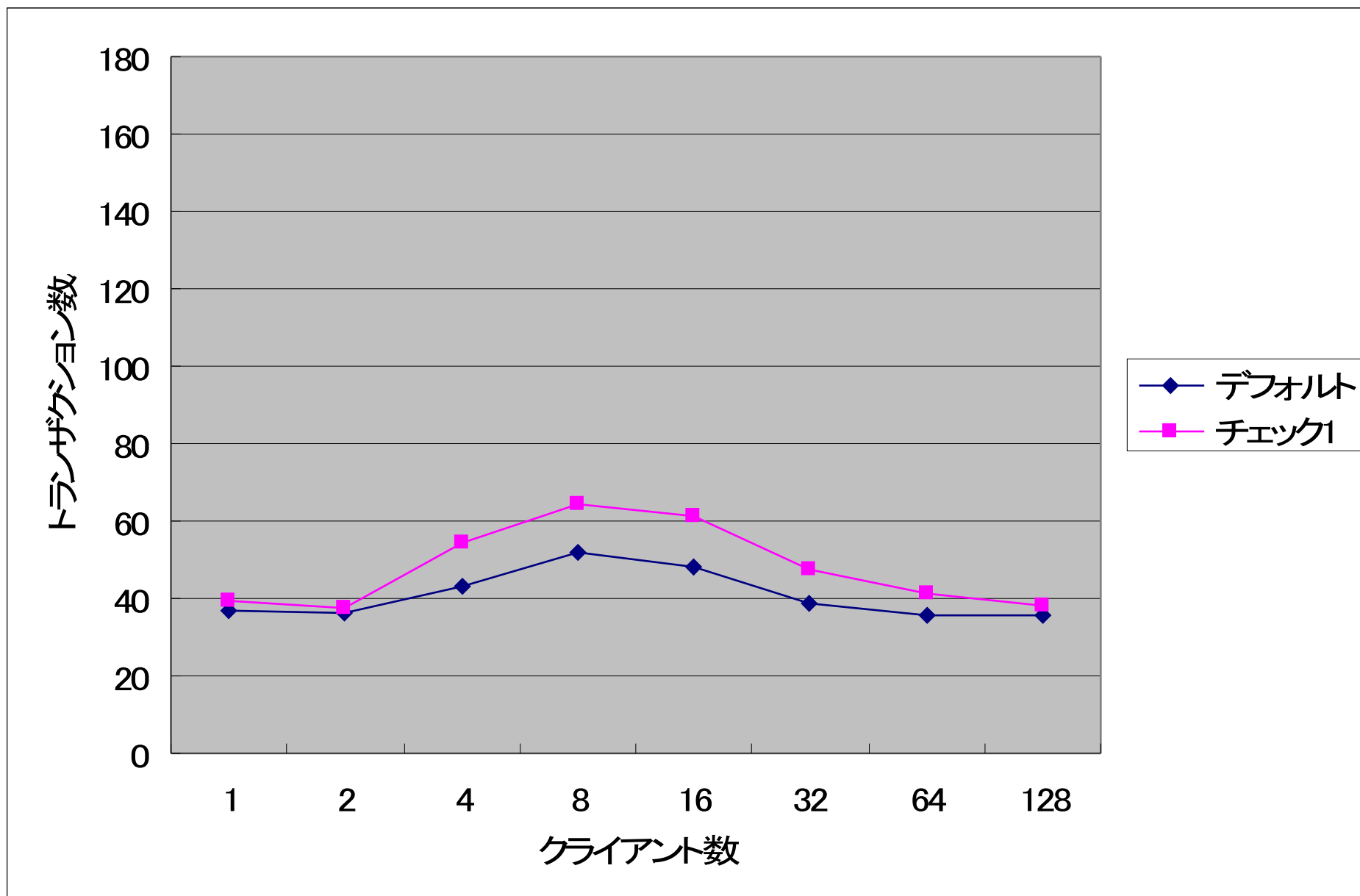


共有バッファの調整

- 調整のポイント
 - 共有バッファは共有メモリ上に作成される。
 - 基本的に大きいほど良い(誤解を恐れずに…)。
 - ただし、プログラム実行用のメモリは残しておく必要がある。

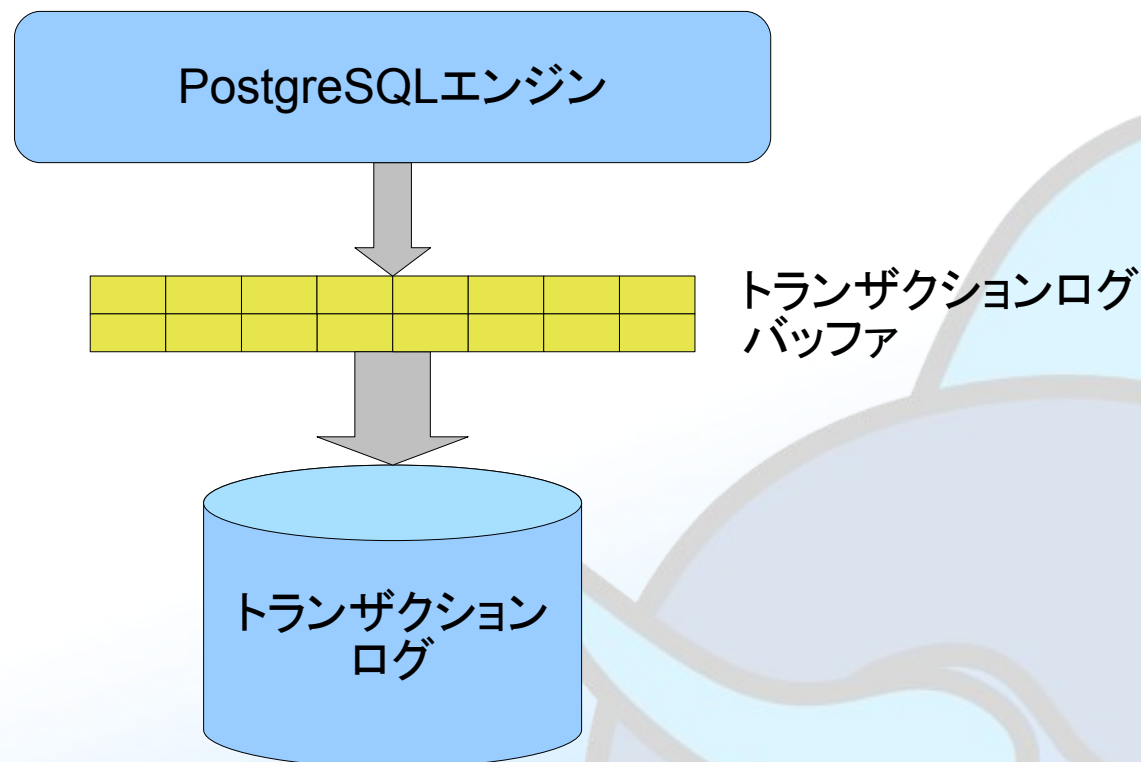


共有バッファを増やすと...



チェック 2 : トランザクションログバッファ

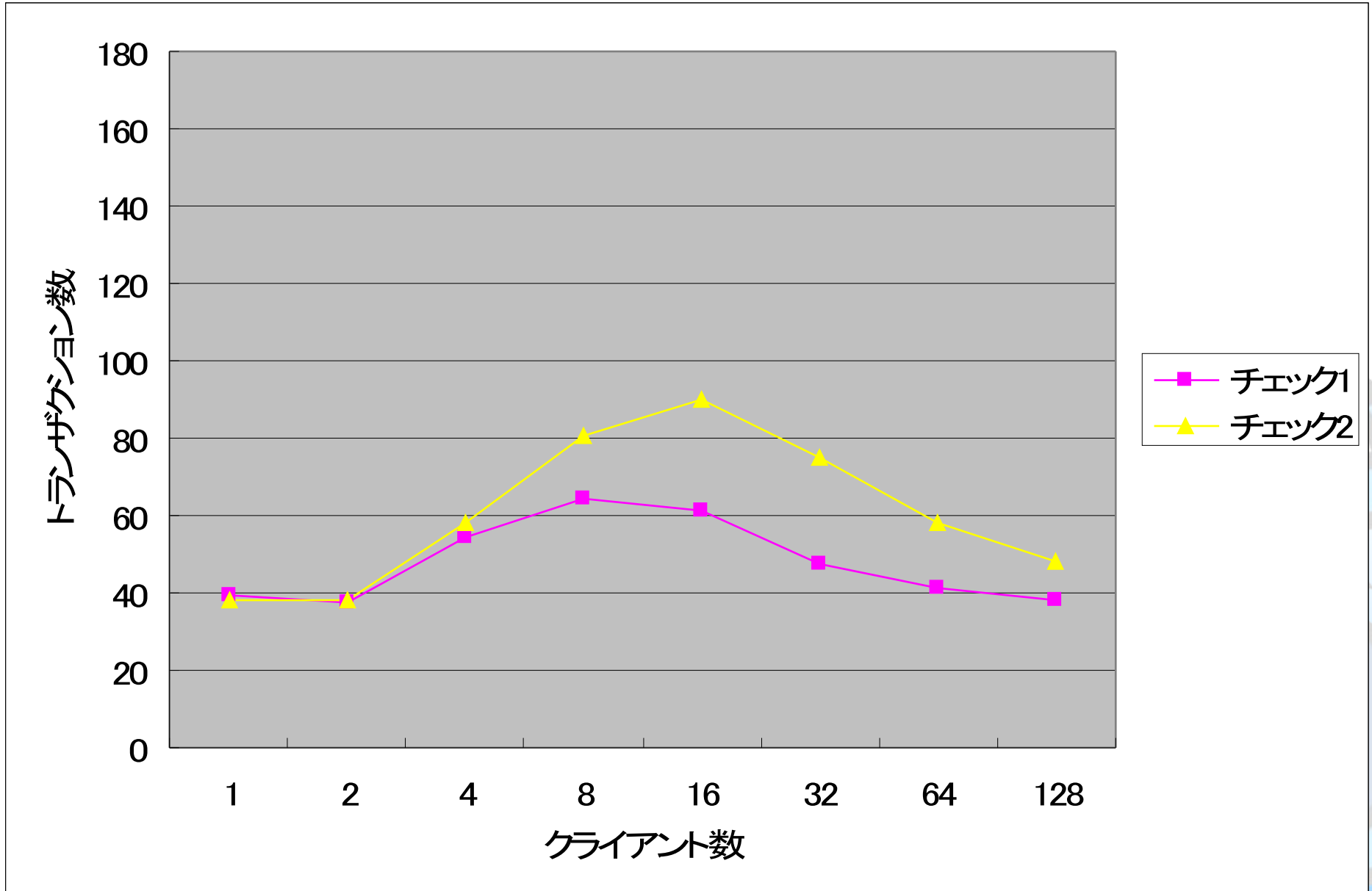
- トランザクションログ (WAL) の書き込みに用いられるバッファ
- デフォルトは 8 ページ (64KB)
 - トランザクションの規模に対してこれが小さすぎると、不必要なタイミングでトランザクションログの書き出しが行われてしまう。



トランザクションログバッファの調整

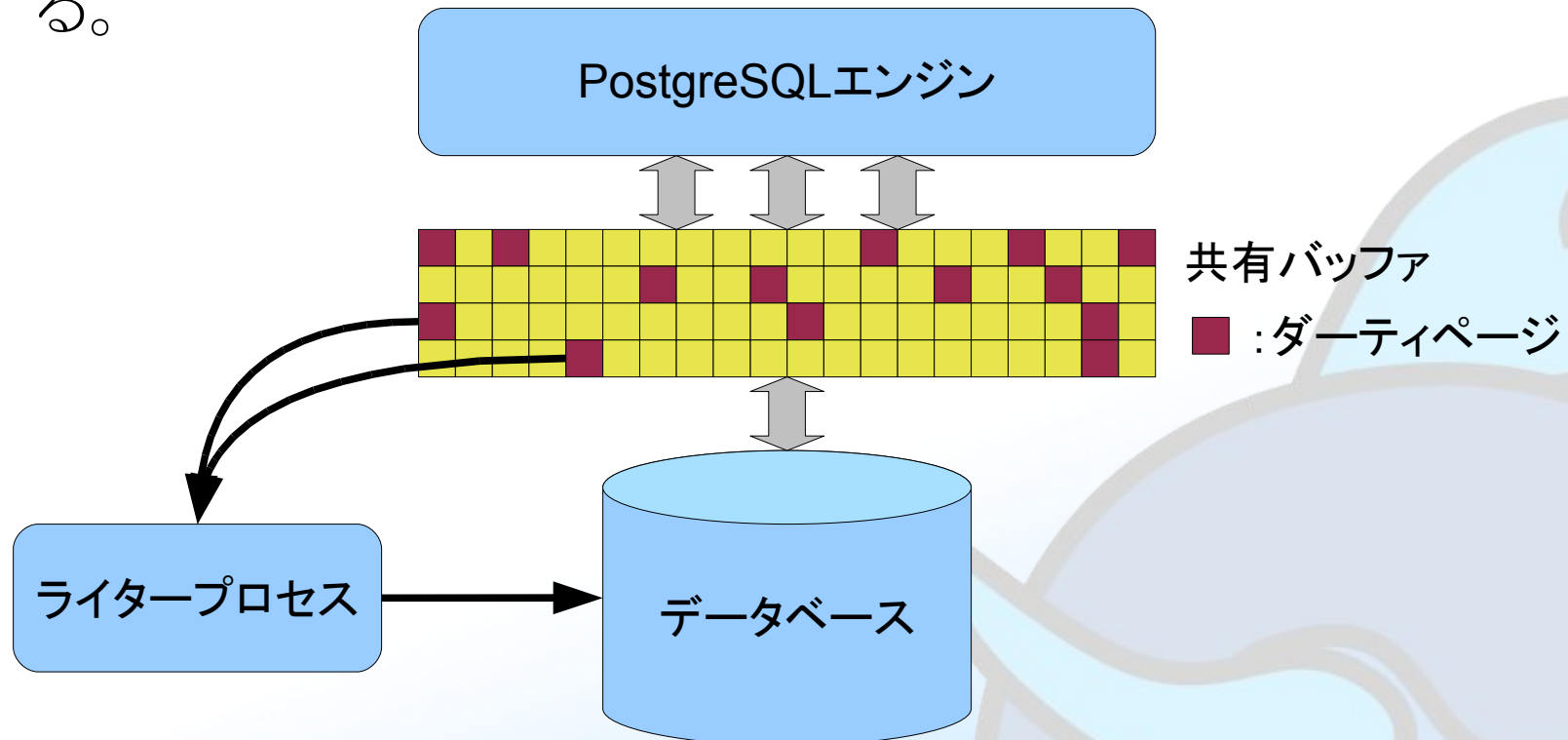
- 調整のポイント
 - トランザクションログバッファは共有メモリ上に作成される。
 - 1 ページは 8K バイト。
 - コミット→コミット間のトランザクションログが格納できる大きさが理想。
 - 複数のクライアントが同時接続していると、コミットは不定期かつ頻繁に発生するので、正確な見積もりは難しい。
 - ベンチマークテストで最適値を見つける。
 - とりあえず適当な大きさ(16 ~ 32 くらい)に設定してもよい。
- 例として、これを 32 ページ(256KB)に増やしてみると…
 - postgresql.conf の編集
 - wal_buffers = 32

トランザクションログバッファを増やすと...



チェック3 : ライタープロセス (Ver.8.0)

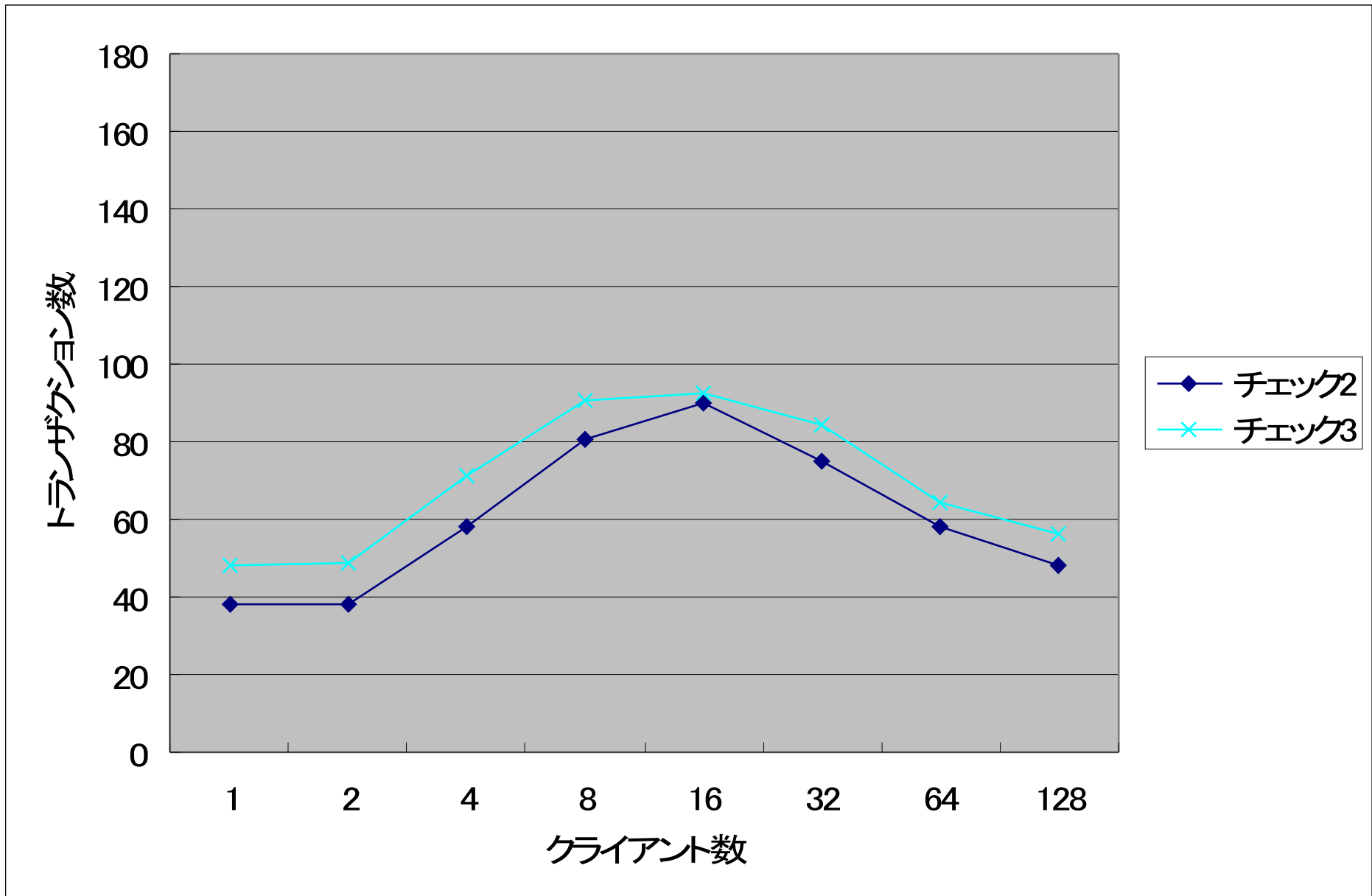
- 共有バッファ内のダーティページを HDD へ保存し続けるバックグラウンドプロセス
- デフォルトでは 200ms ごとに最大 100 ページを書き込む
 - 更新が多いデータベースではライタープロセス自体が重くなる。



チェック3 :ライタープロセス (Ver.8.0)

- 調整のポイント
 - 負荷の高いサーバの場合、ライタープロセスの仕事を軽くすれば表面的な性能は向上する。
 - 周期を長くする
 - 最大書き込みページ数を少なくする
 - ライタープロセスの仕事を軽くしすぎると副作用もある。
 - CHECKPOINT が重くなる
- 例として 200ms ごとに最大 4 ページに設定してみると…
 - postgresql.conf
 - bgwriter_delay = 200
 - bgwriter_maxpages = 4

ライタープロセス (Ver.8.0) を調整すると...



ライタープロセス (Ver.8.1 以降)

- バージョン 8.1 ~ 8.2
 - デフォルト値が見直され、最大 5 ページになった
- バージョン 8.3
 - 共有バッファ全体を対象としたダーティページの書き込みが行われなくなった
 - CHECKPOINT の改良により不要となった
 - 再利用されるページを対処としたダーティページの書き込みのみ行われる
 - 負荷に応じて自動調整が行われるようになった！

チェック4：テーブルスペース

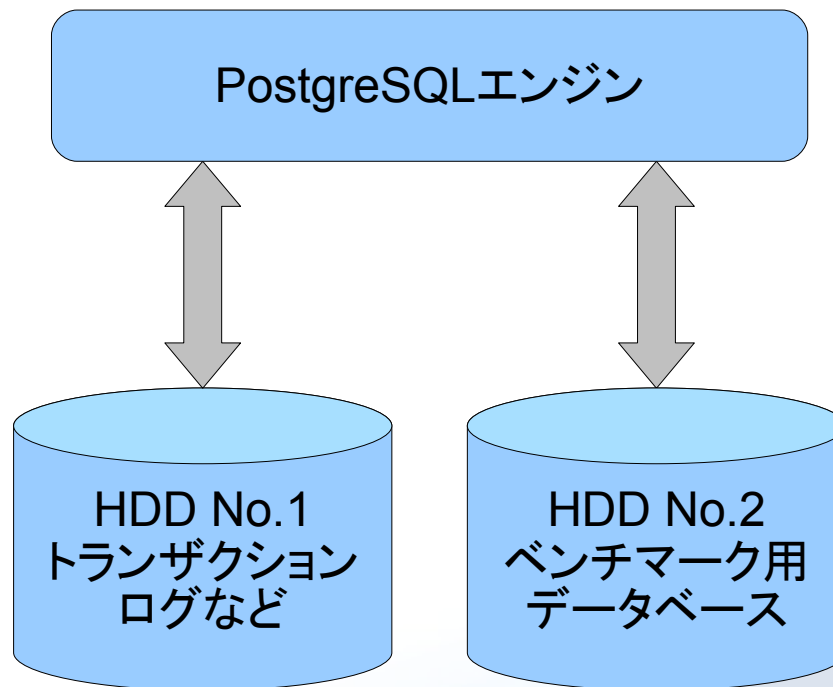
- データベースオブジェクトを任意のディレクトリに配置するための機能
 - 配置可能なオブジェクト
 - DATABASE、TABLE、INDEX、SEQUENCE
- デフォルトテーブルスペース“pg_default”
 - データベースオブジェクトのデフォルトの格納先。
 - トランザクションログなどもここに格納されると考えてよい。
 - I/O の負荷分散の余地が残されている。

(例)

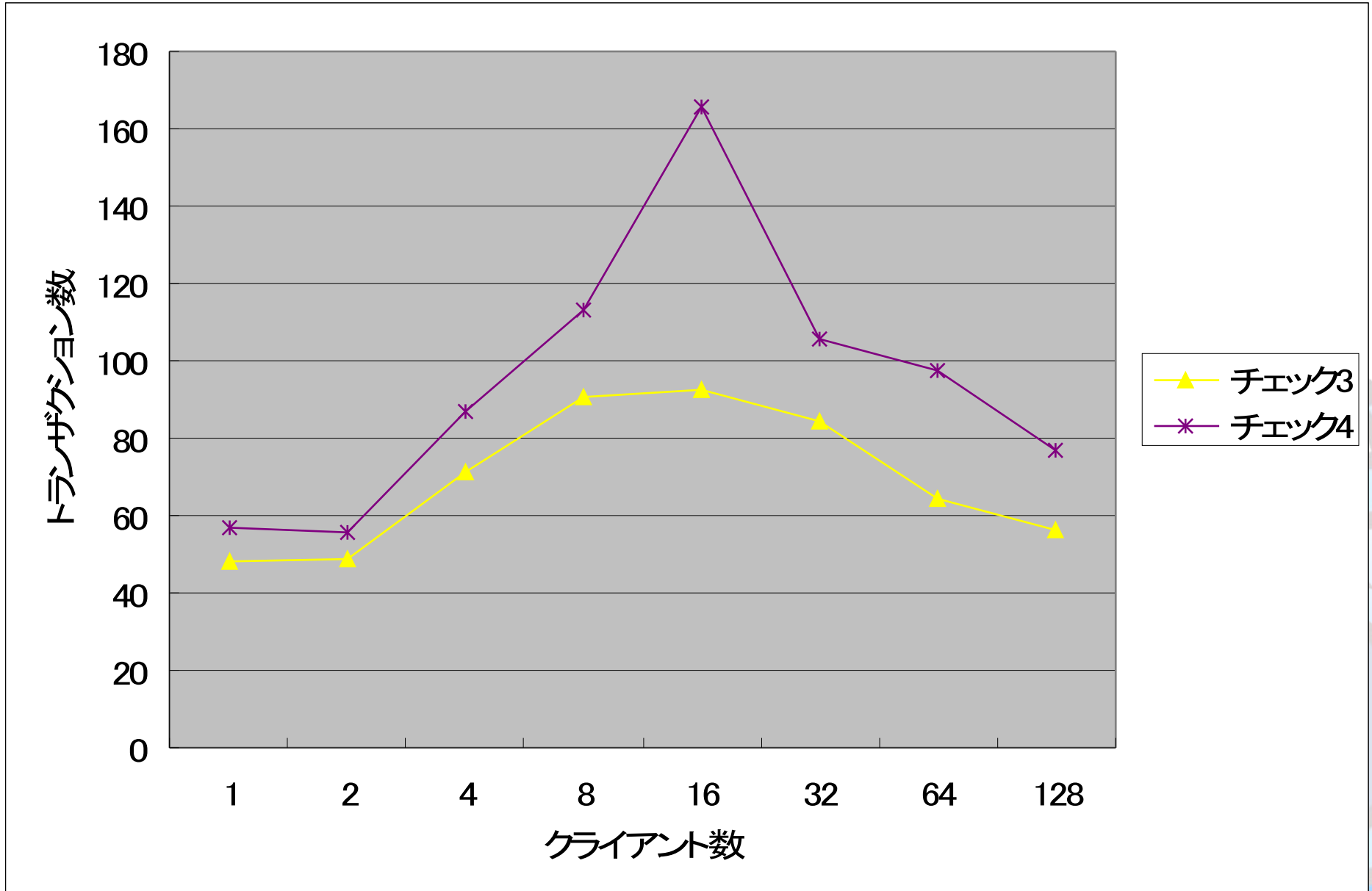
```
CREATE TABLESPACE pgdata1 LOCATION '/disk1/pgdata1';  
CREATE DATABASE ~ TABLESPACE pgdata1;  
CREATE TABLE ~ TABLESPACE pgdata1;  
ALTER TABLE ~ SET TABLESPACE pgdata1;
```

テーブルスペースの活用

- ベンチマーク用のデータベースを別 HDD に配置して I/O の負荷分散をしてみると…

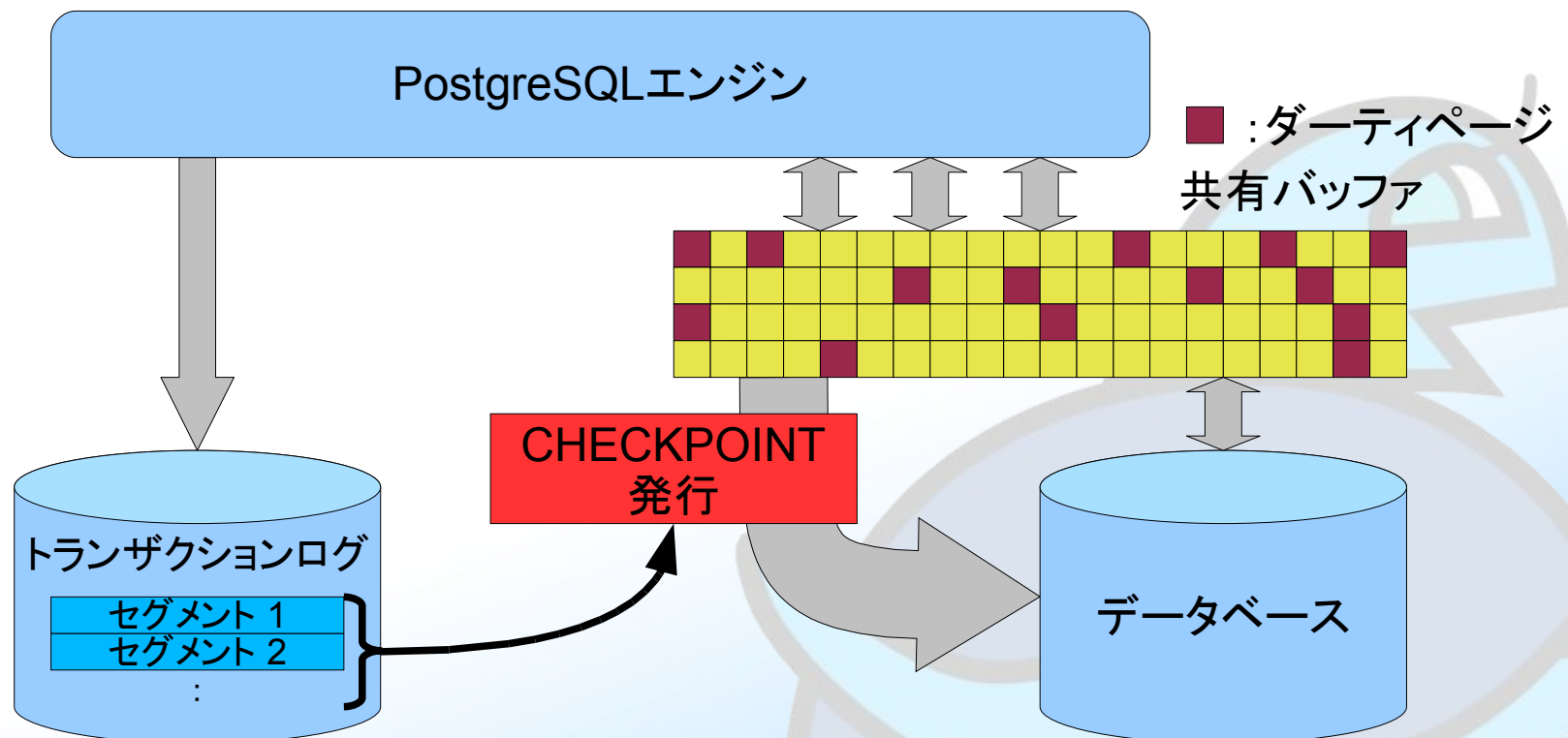


テーブルスペースを利用すると…



チェック5:チェックポイントセグメント数

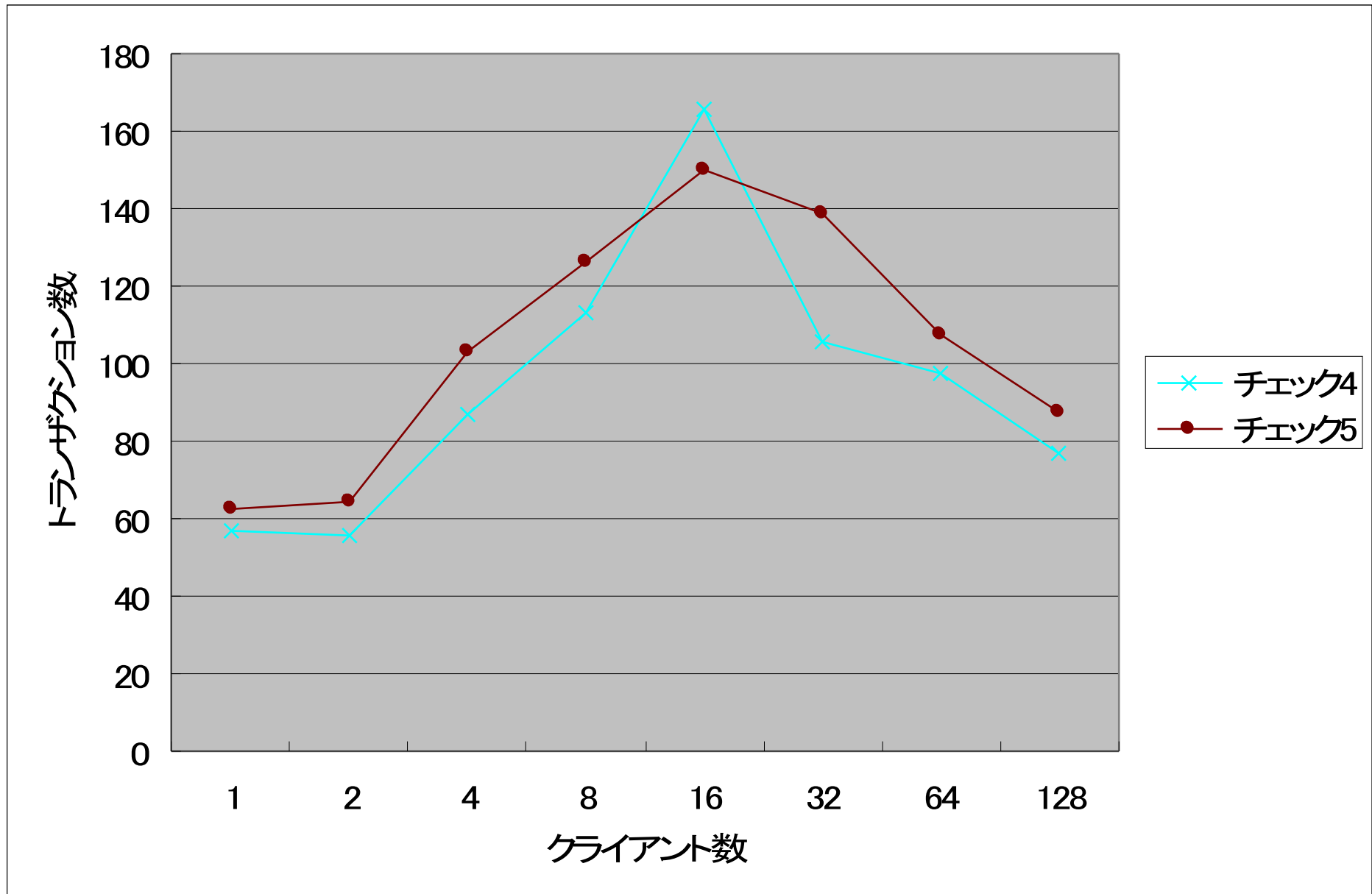
- トランザクションログのセグメント数がこの値に達するとチェックポイントが発生する
- デフォルトでは3セグメント
 - 更新の多い環境ではやや少ない。



チェックポイントセグメント数の調整

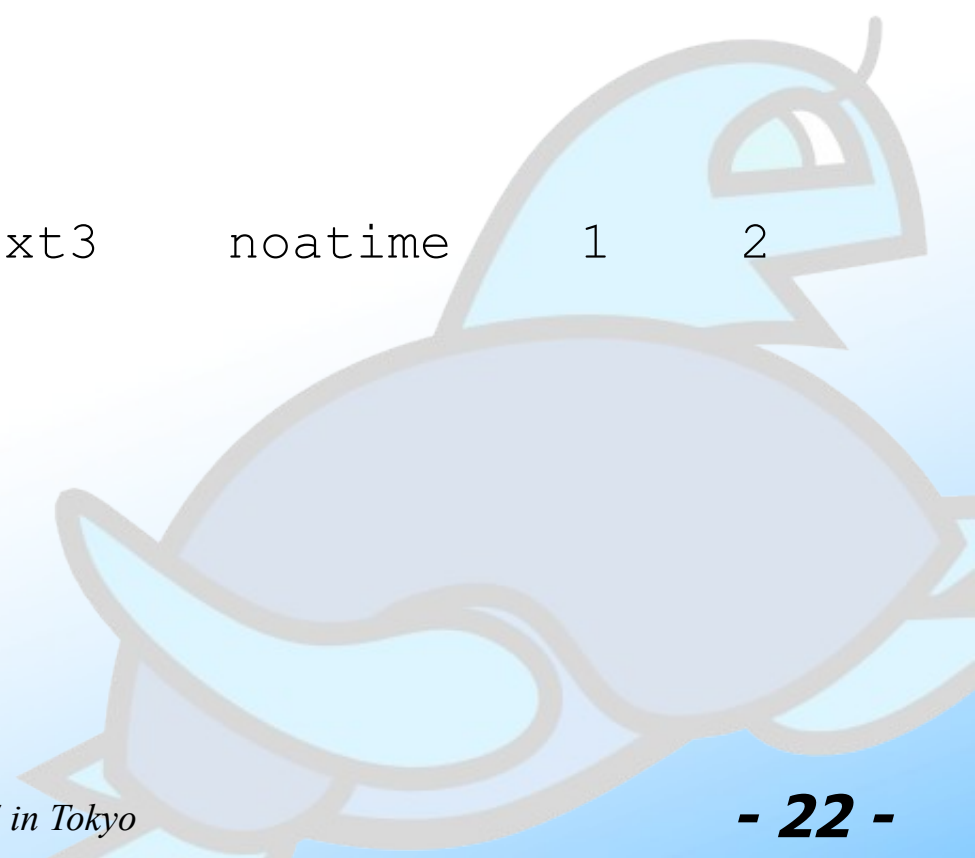
- 調整のポイント
 - トランザクションログはディスク上に作成される。
 - 1 セグメントは 16MB
 - 最大ディスク使用量は、その約 2 倍 (8.2 まで)、約 3 倍 (8.3 以降)。
 - 基本的に大きいほど良い。
 - ディスク空き容量と相談。
- 例として、これを 16 セグメントに増やしてみると…
 - postgresql.conf
 - `checkpoint_segments = 16`

チェックポイントセグメント数を増やすと...

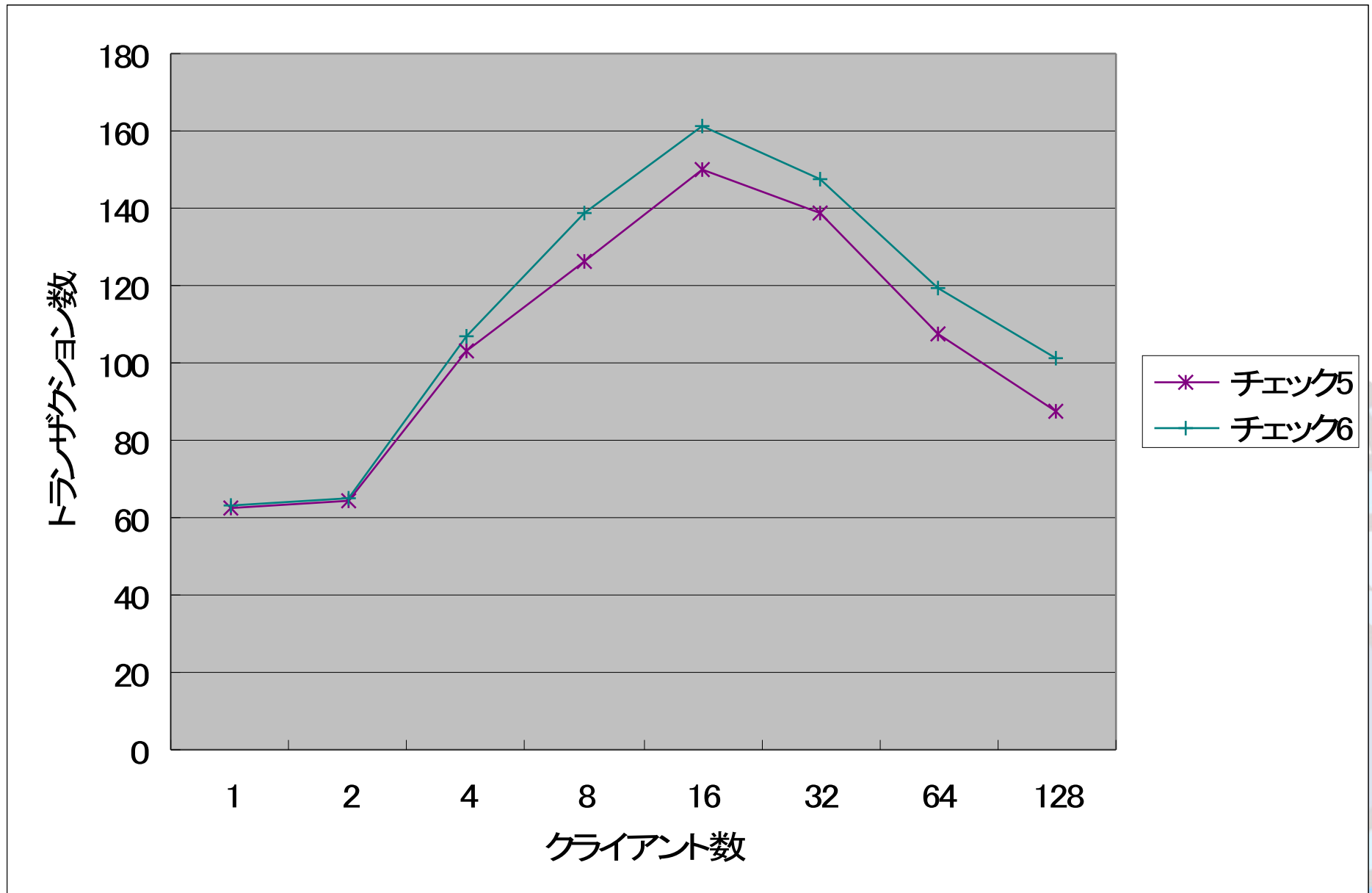


チェック 6 : ファイルシステムの調整

- ファイルシステムがデータベース専用であれば、**atime** (アクセス時間の記録) を「**off**」に設定することでパフォーマンスが向上
- 設定例
 - **mount** コマンドでは
 - `mount -o noatime ...`
 - **/etc/fstab** では
 - `/dev/sdc1 /disk1 ext3 noatime 1 2`
- では、どうなるかというと...



ファイルシステムを調整すると…



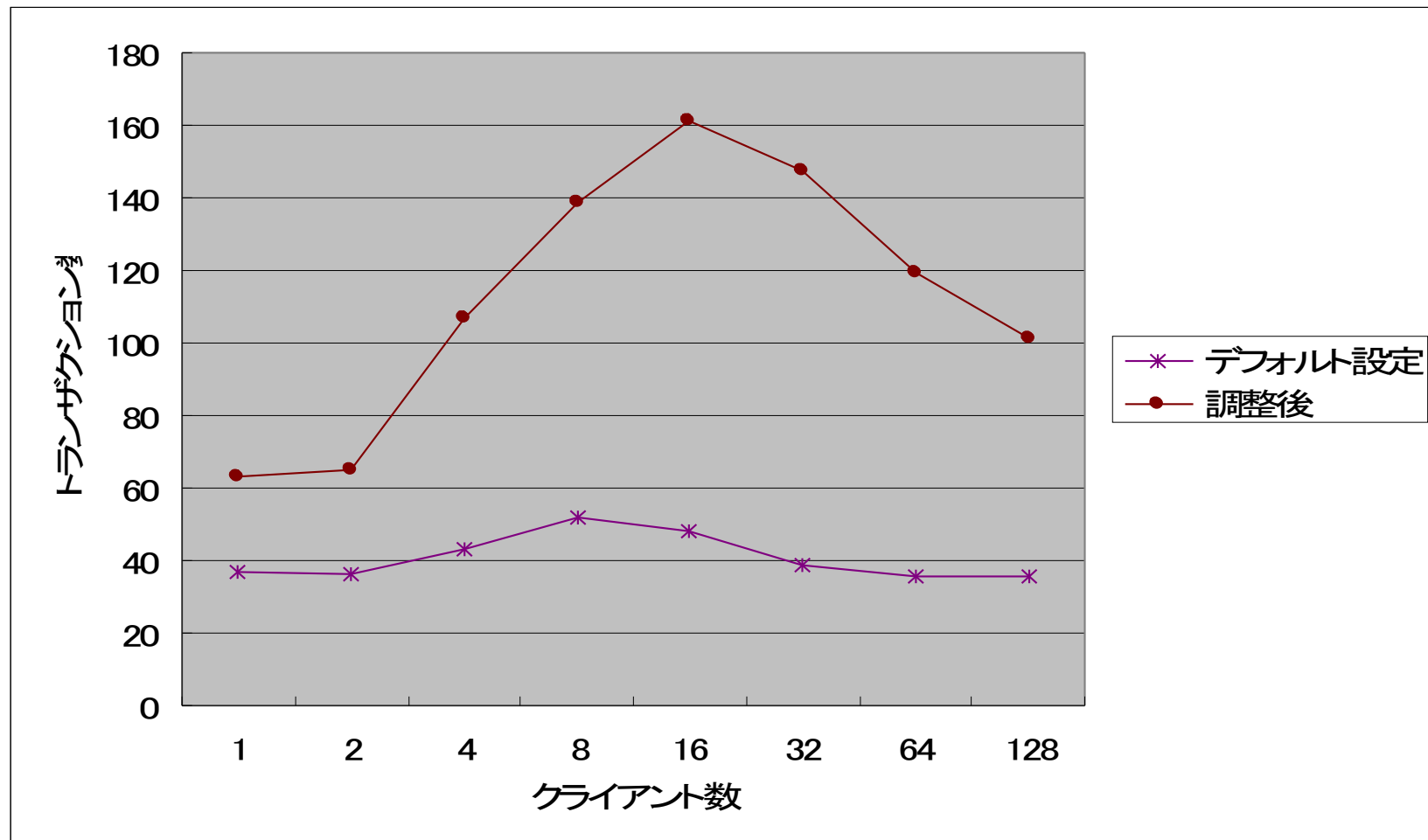
その他のチェック項目: effective_cache_size

- OS のディスクキャッシュサイズの想定値
 - 正しく設定していない場合、効率の悪い問い合わせプランを選択することがある。
 - デフォルトでは 1000 ページ (8K バイト)。ul> - ほとんどの場合で少なすぎる。
- 調整のポイント
 - サーバの平均的な空きメモリ量 (サーバ搭載メモリ量の 1/3 くらい?) を設定すると良い。
 - 主な OS では空きメモリ量のほとんどをディスクキャッシュに利用するため。
- 設定例
 - postgresql.conf
 - effective_cache_size = 16000

その他のチェック項目：バキューム

- PostgreSQL は追記型アーキテクチャ
 - レコードの更新と削除では不要領域が増えてしまう。
 - 時間と共に不要領域が増え、徐々に性能が低下。
 - ゴミ掃除が必要！
- バキュームの方法
 - VACUUM
 - ゴミ掃除を行うための PostgreSQL 特有の SQL 命令
 - vacuumdb
 - VACUUM を行うラップコマンド
- 定期的な VACUUM の実行
 - cron
 - 自動バキューム (Ver.8.3 からはデフォルトで ON)

PostgreSQL を最大限に使い切ろう！



— ご清聴ありがとうございました —