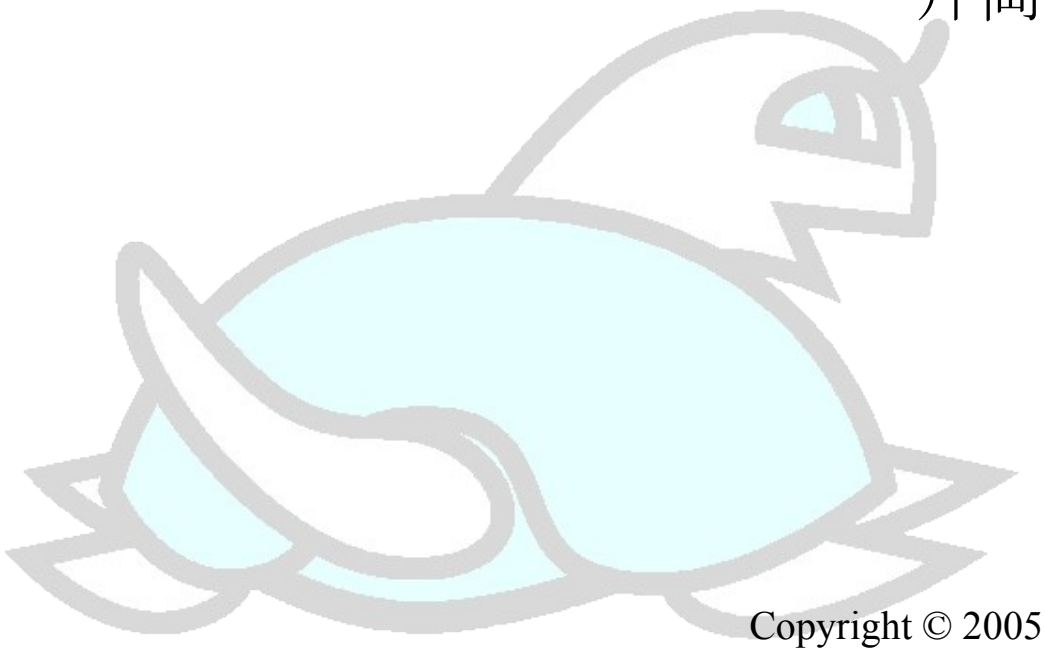
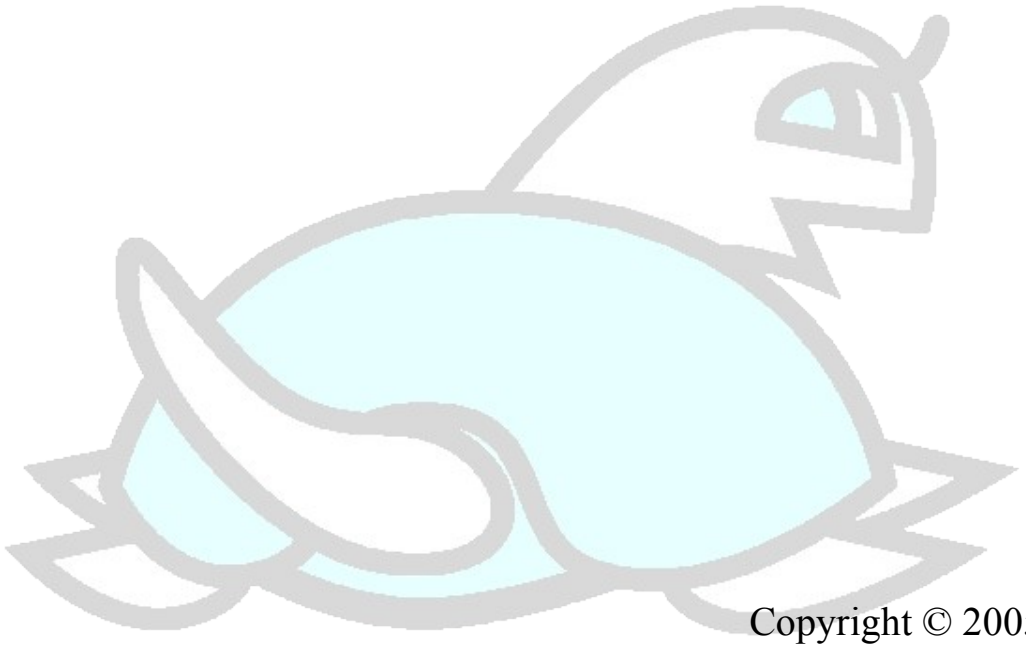
PostgreSQLをデフォルト設定のまま使っていませんか？

日本PostgreSQLユーザ会
片岡 裕生



PostgreSQLのデフォルト設定とは？

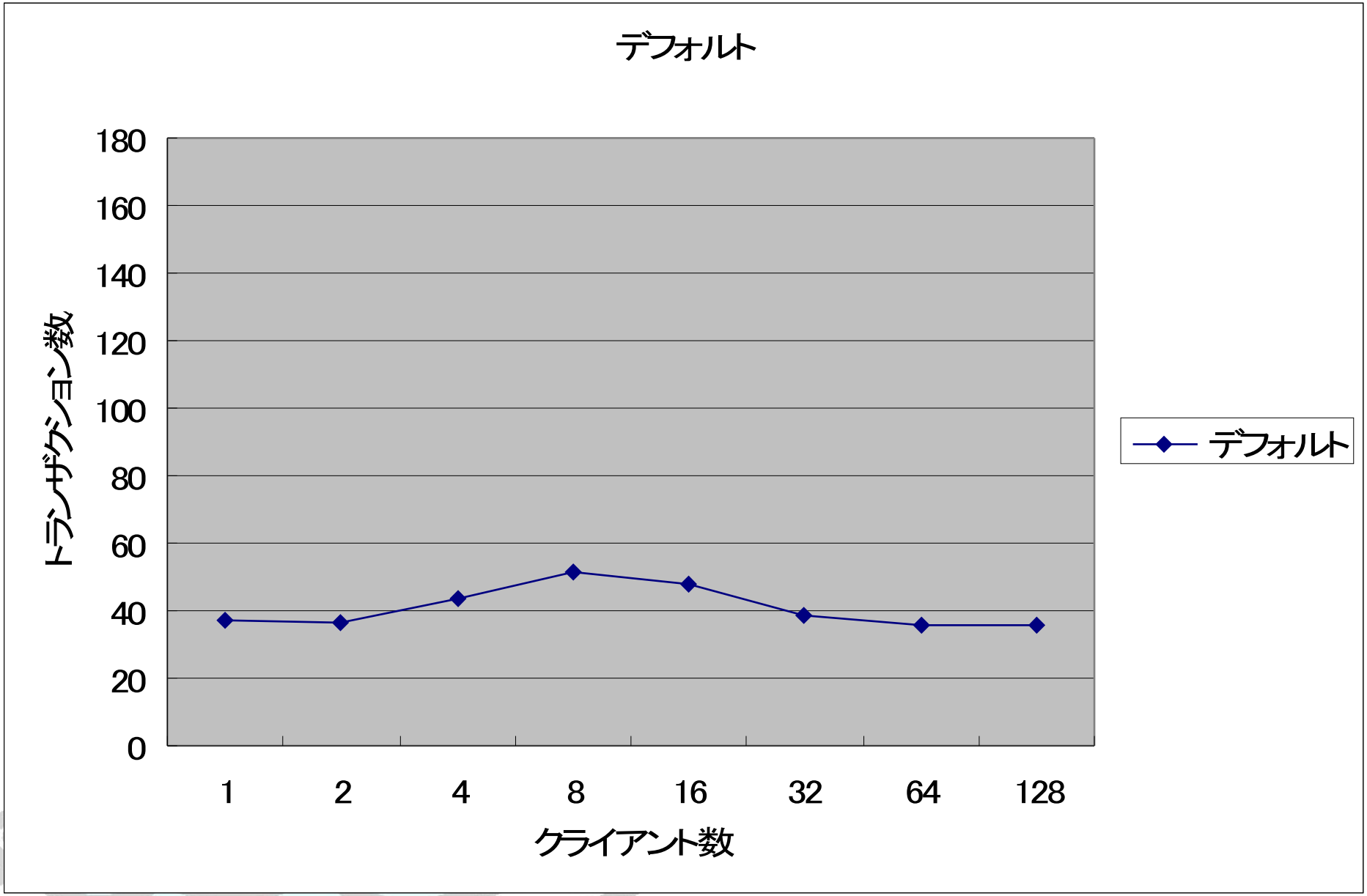
- PostgreSQLをインストールしただけの状態のこと
- 設定内容はミニマム
  - ロースペックのマシン環境でも動作するように。



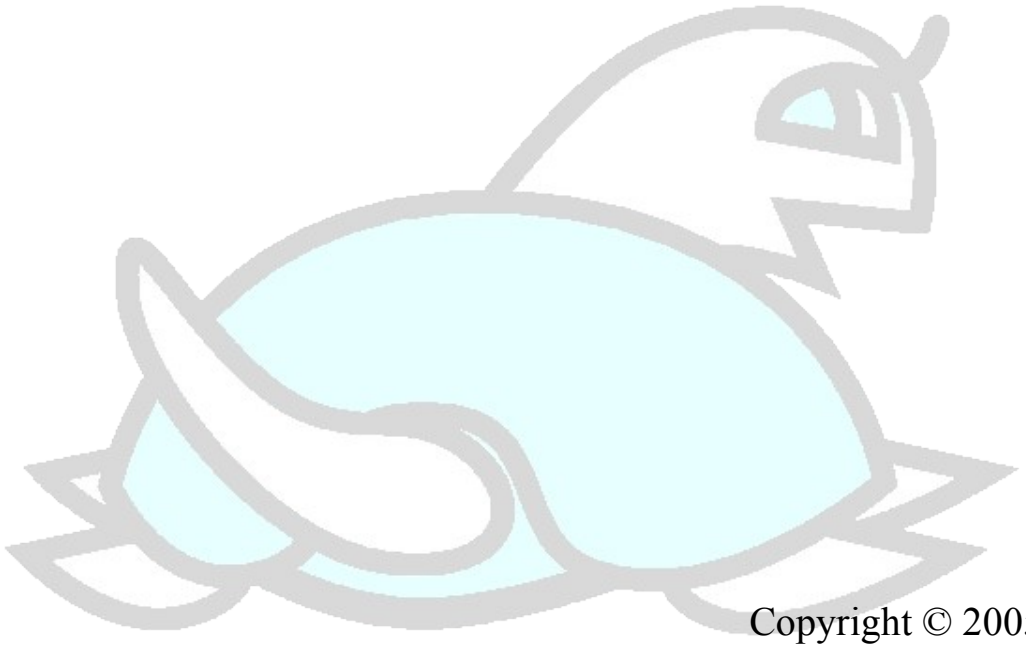
ベンチマークについて

- pgbenchによるベンチマークを実施
  - select×1＋update×3という、更新割合の高いベンチマーク。
- データ規模
  - 500万行
  - 800MBのデータサイズ
- ハードウェア
  - Pentium III 1.4G×2
  - メモリ512MB
  - HDD
    - ATA 20GB（OS用）
    - U160 SCSI 18GB 10000rpm×2（データベース用）
- ソフトウェア
  - Red Hat EL 3.0 ES
  - PostgreSQL 8.0.3

デフォルト設定でのパフォーマンス

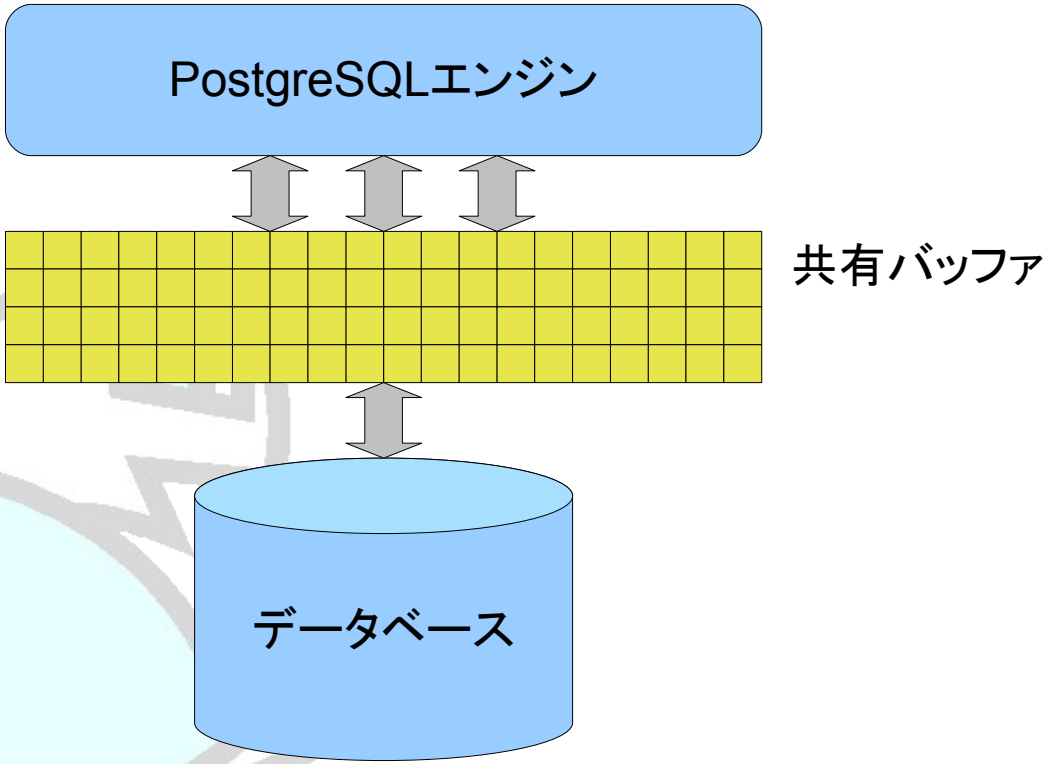


PostgreSQL設定のチェックポイント



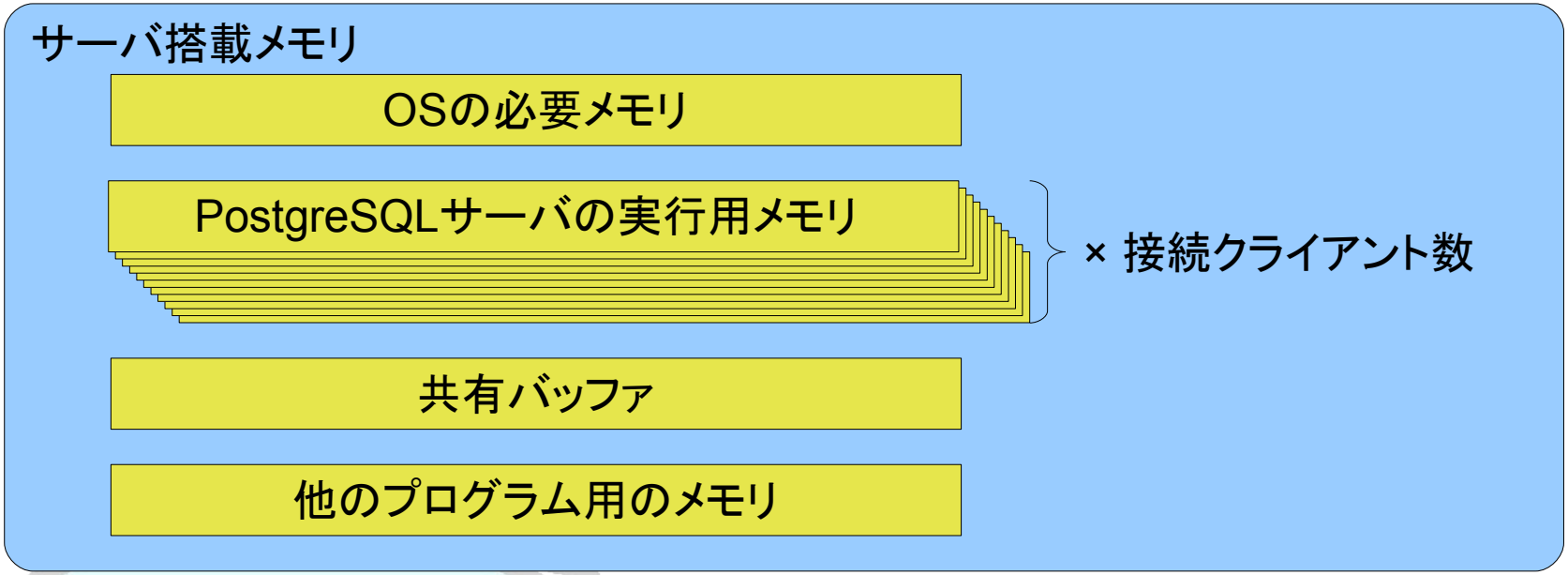
チェック1：共有バッファ

- 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1000ページ（8MB）
  - ほとんどの場合、1000ページでは少なすぎる。
  - 古いPostgreSQLのバージョンではデフォルトで32ページしかない。



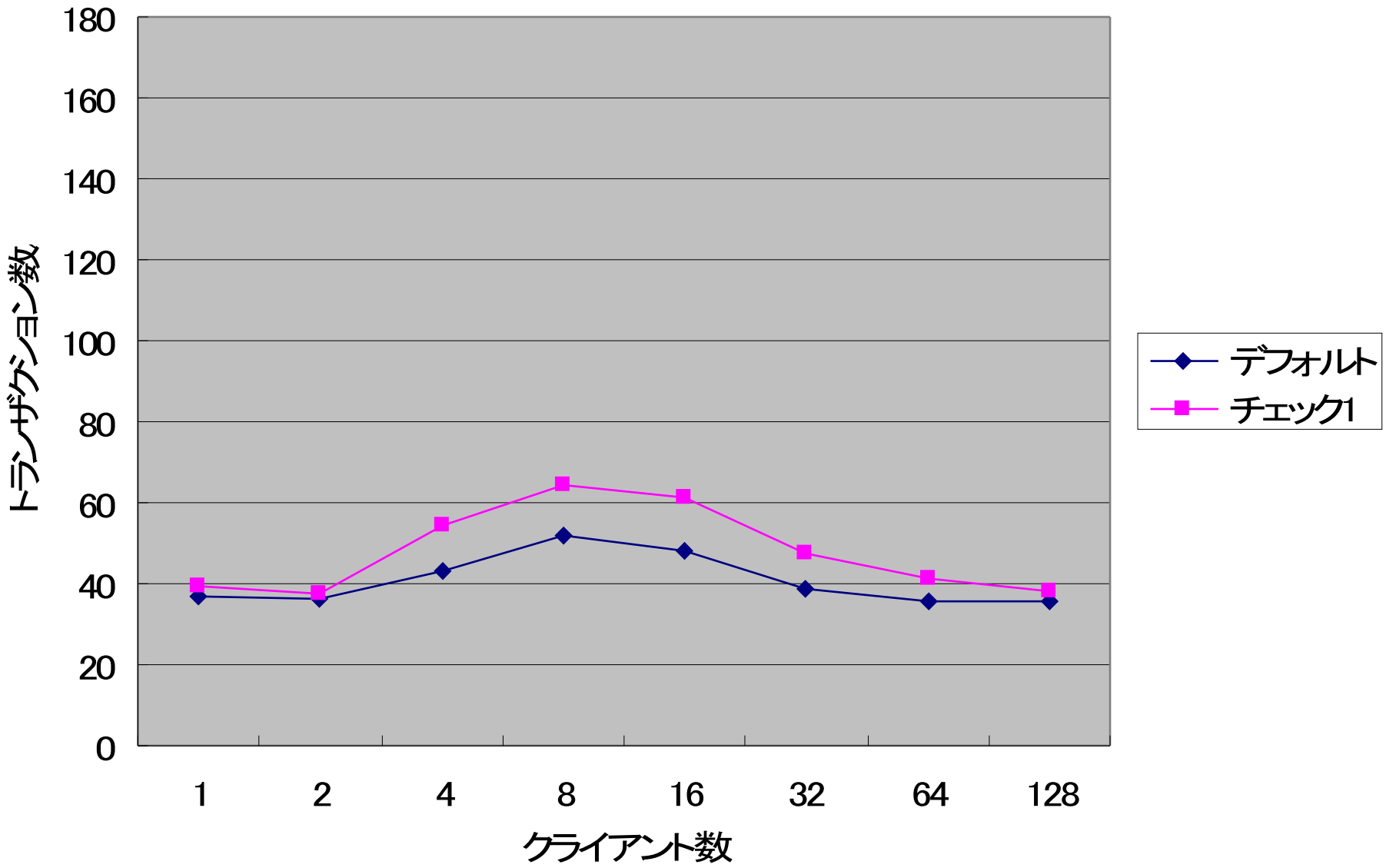
共有バッファの調整

- 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。
  - 1ページは8Kバイト。
  - 基本的に大きいほど良い。
  - 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。



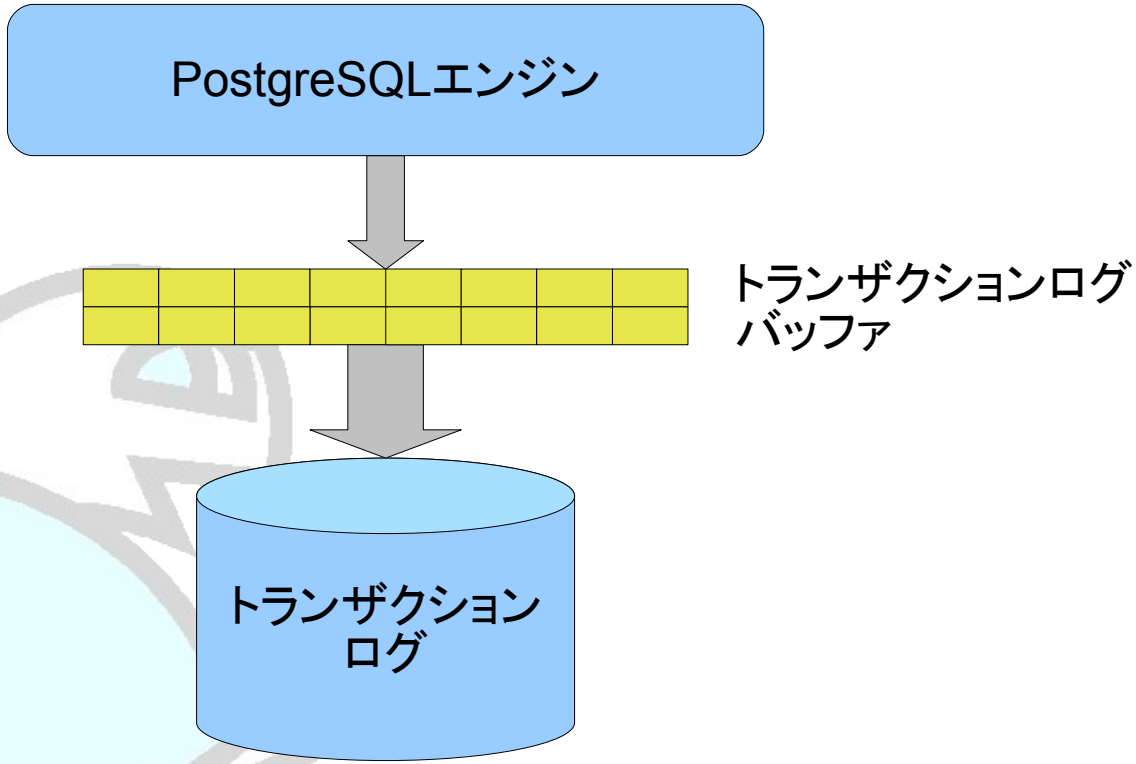
- 例として、これを16000ページ(128MB)に増やしてみると…
  - postgresql.confの編集
    - ➔ shared_buffers = 16000

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

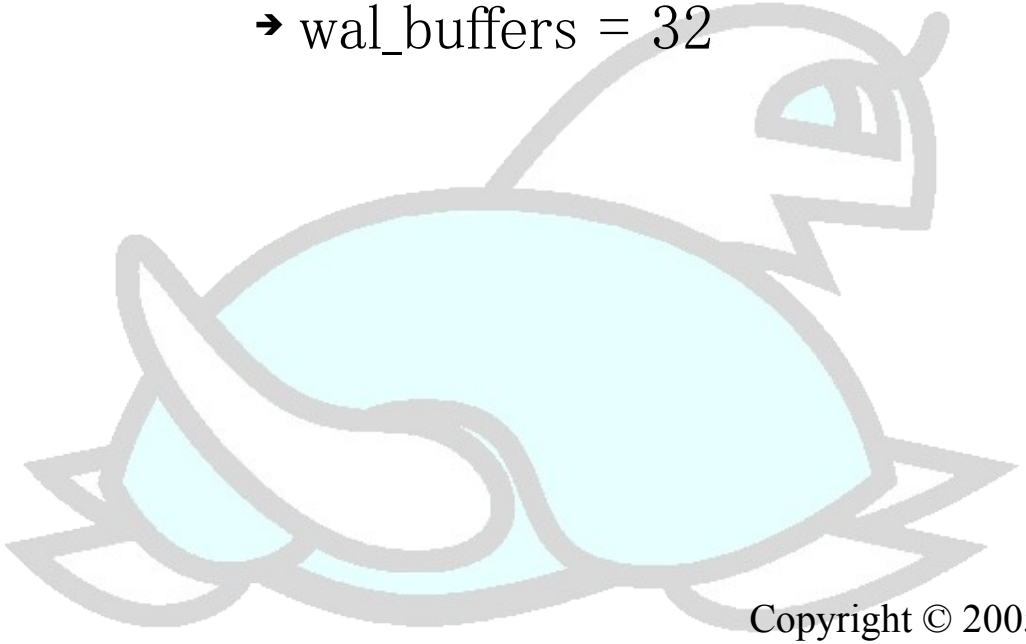
- トランザクションログ(WAL)を書き込む際に用いられるバッファ
- デフォルトは8ページ(64KB)
  - トランザクションの規模に対してこれが小さすぎると、不必要なタイミングでトランザクションログの書き出しが行われてしまう。



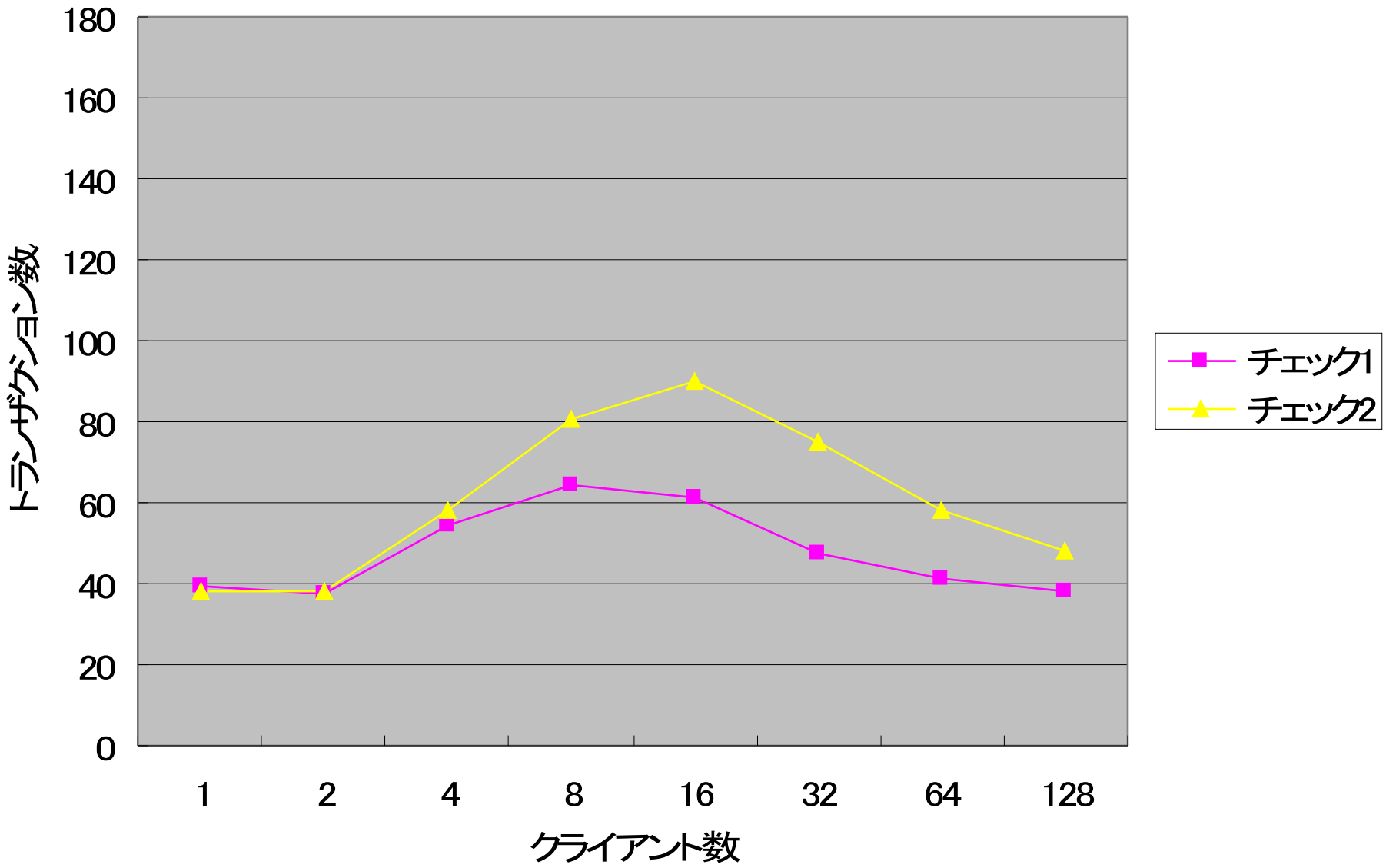
トランザクションログバッファの調整

- 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

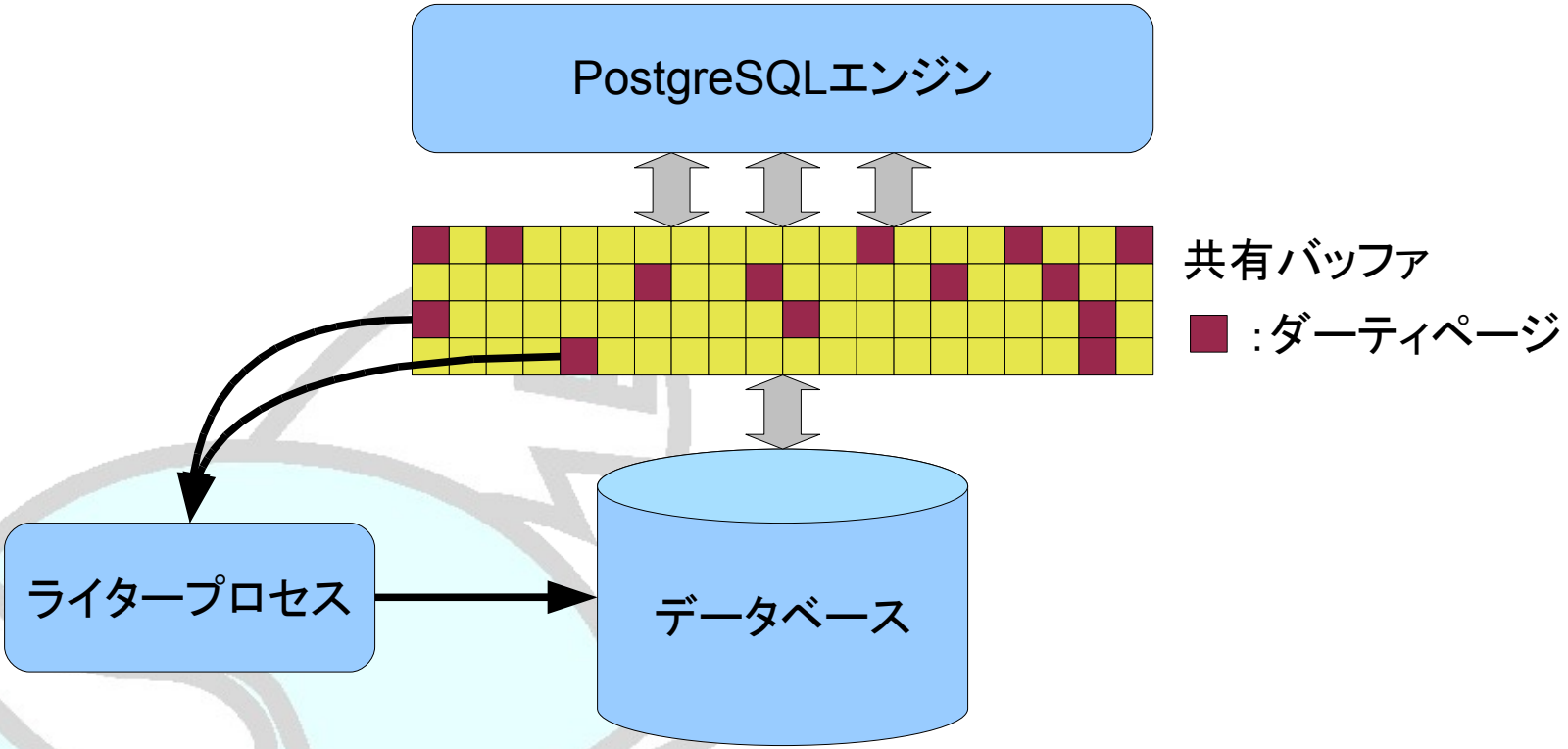


トランザクションログバッファを増やすと…



チェック3:ライタープロセス

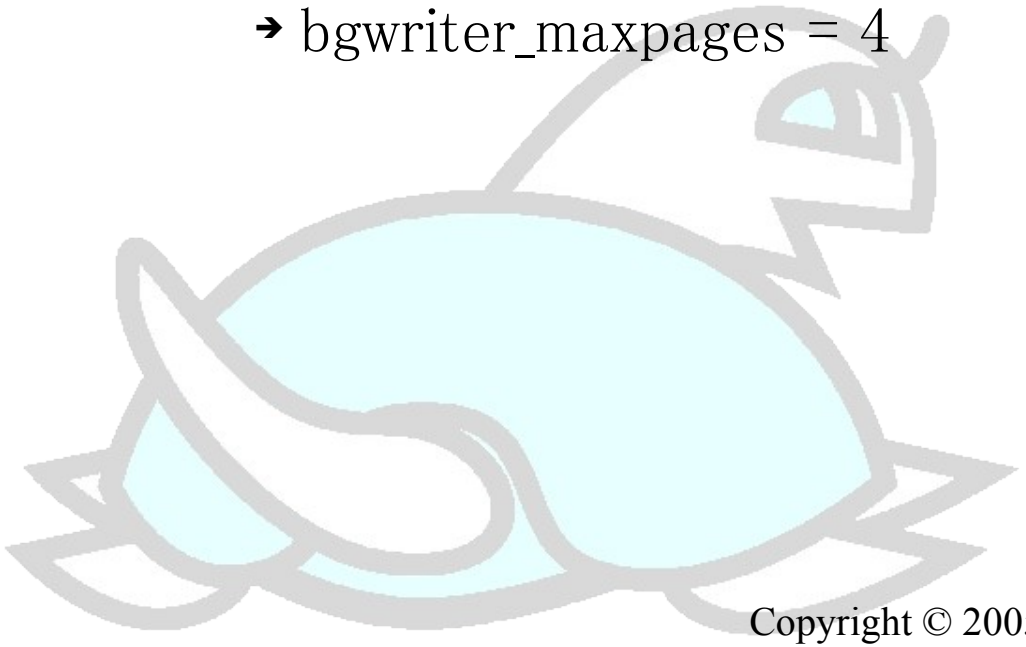
- 共有バッファ内のダーティページ(HDDにまだ保存されていないページ)をHDDへ保存し続けるバックグラウンドプロセス
- デフォルトでは、200msごとに最大100ページを書き込む
  - 更新が多いデータベースではライタープロセス自体が重くなる。



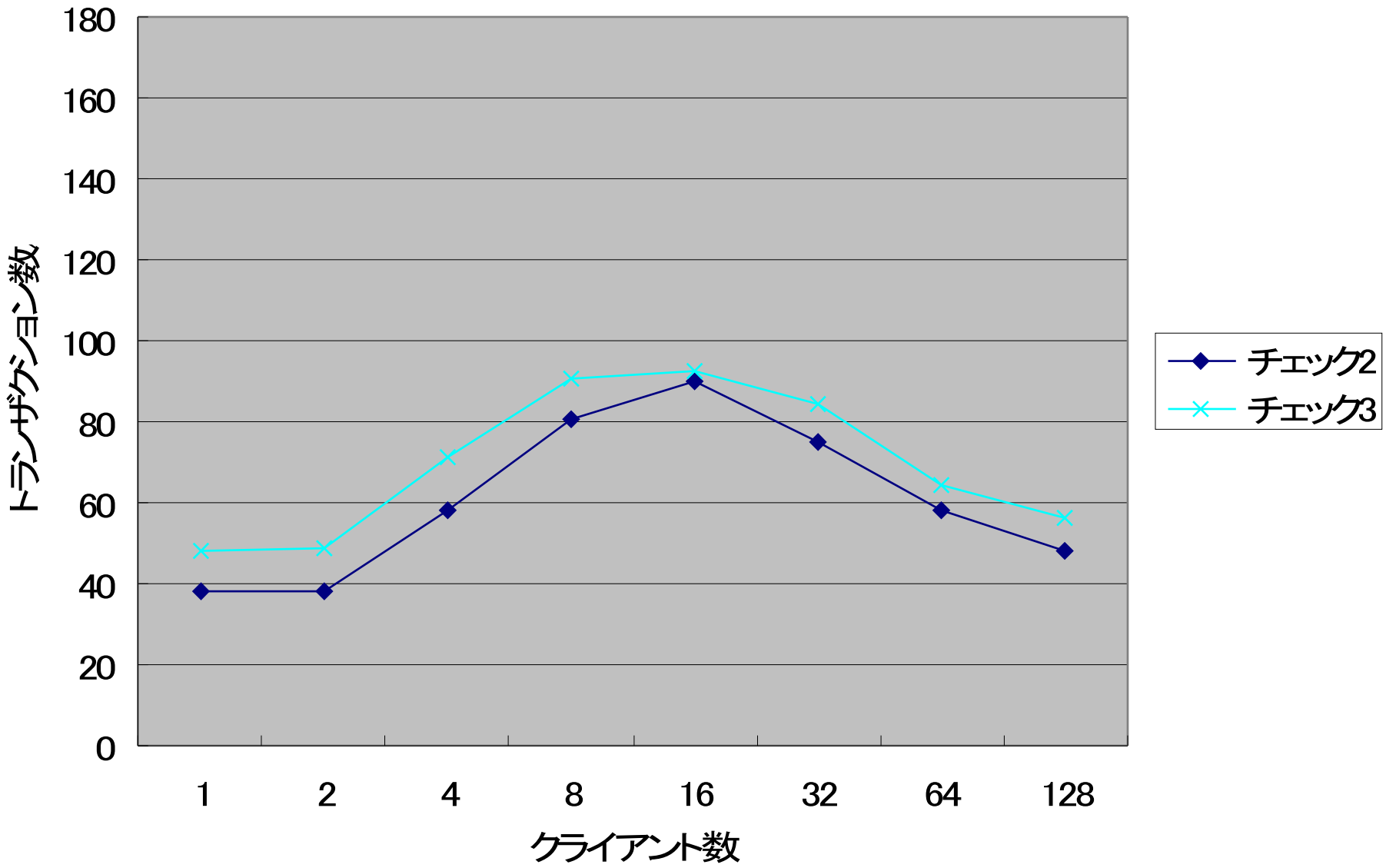
チェック3:ライタープロセス

- 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、これを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



ライタープロセスを調整すると…



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

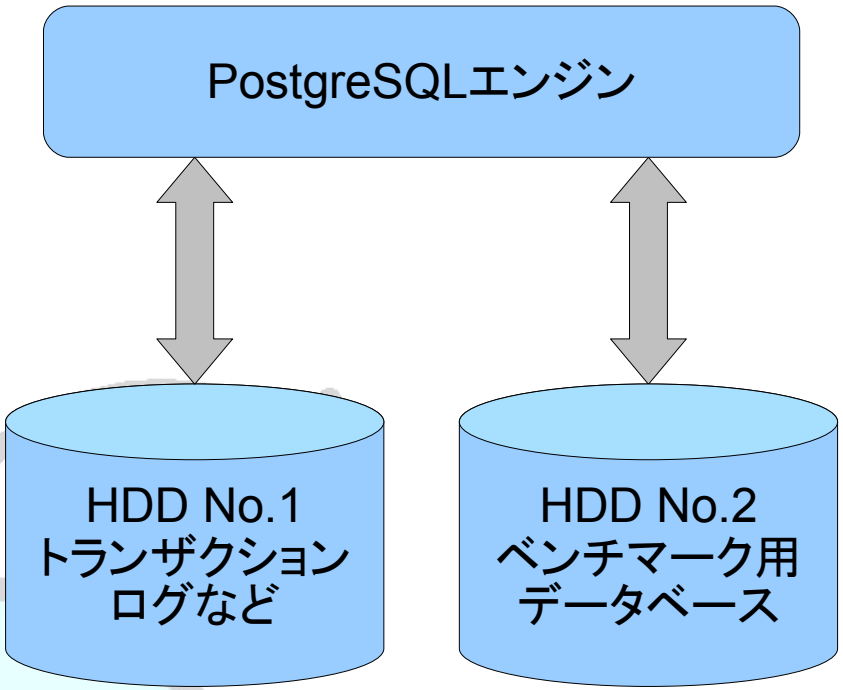
- デフォルトテーブルスペース“pg_default”
  - データベースオブジェクトのデフォルトの格納先。
  - トランザクションログなどもここに格納されると考えてよい。
    - → I/Oの負荷分散の余地が残されている。

(例)

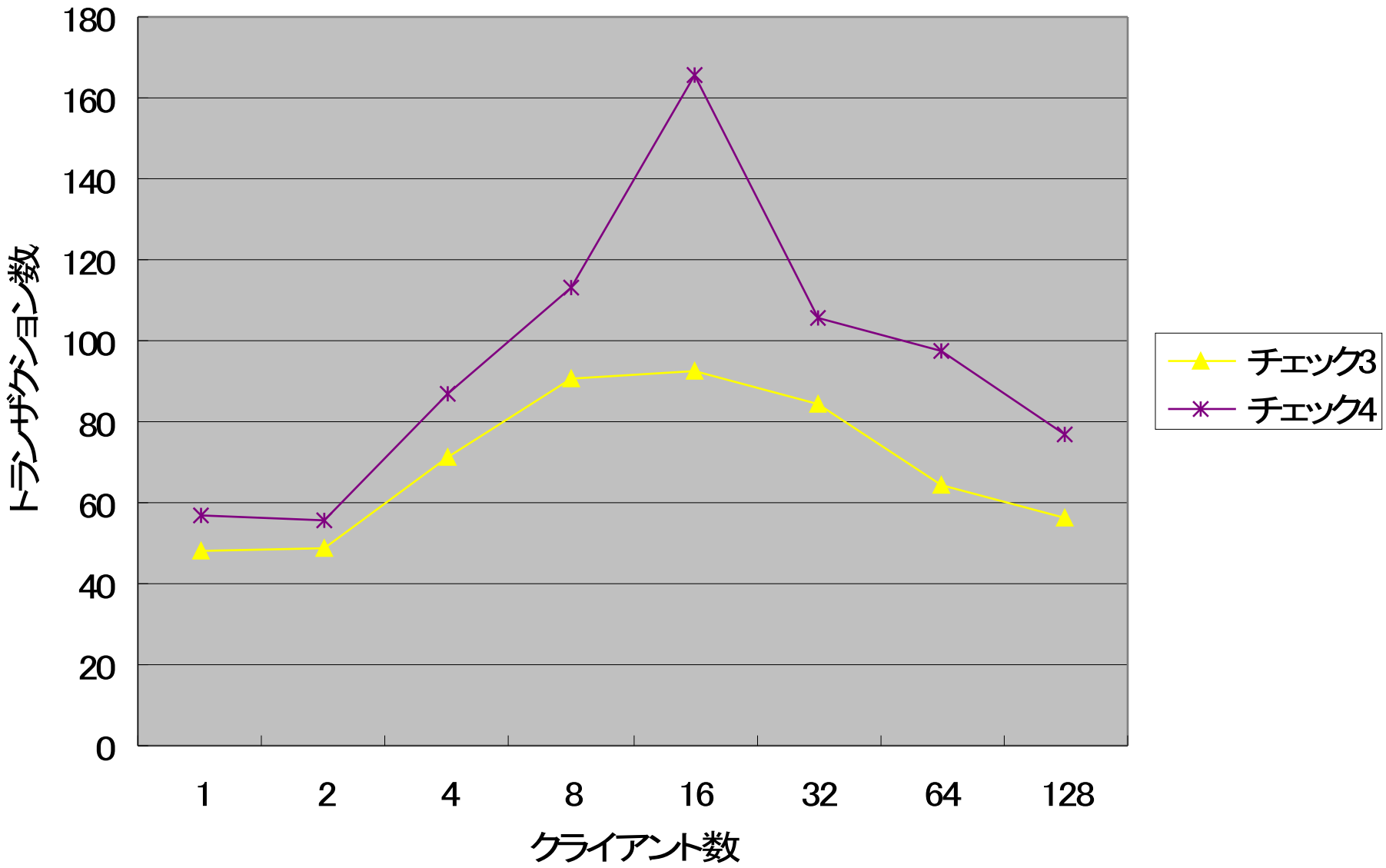
```
CREATE TABLESPACE pgdata1 LOCATION '/disk1/pgdata1';
CREATE DATABASE ～ TABLESPACE pgdata1;
CREATE TABLE ～ TABLESPACE pgdata1;
ALTER TABLE ～ SET TABLESPACE pgdata1;
```

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…

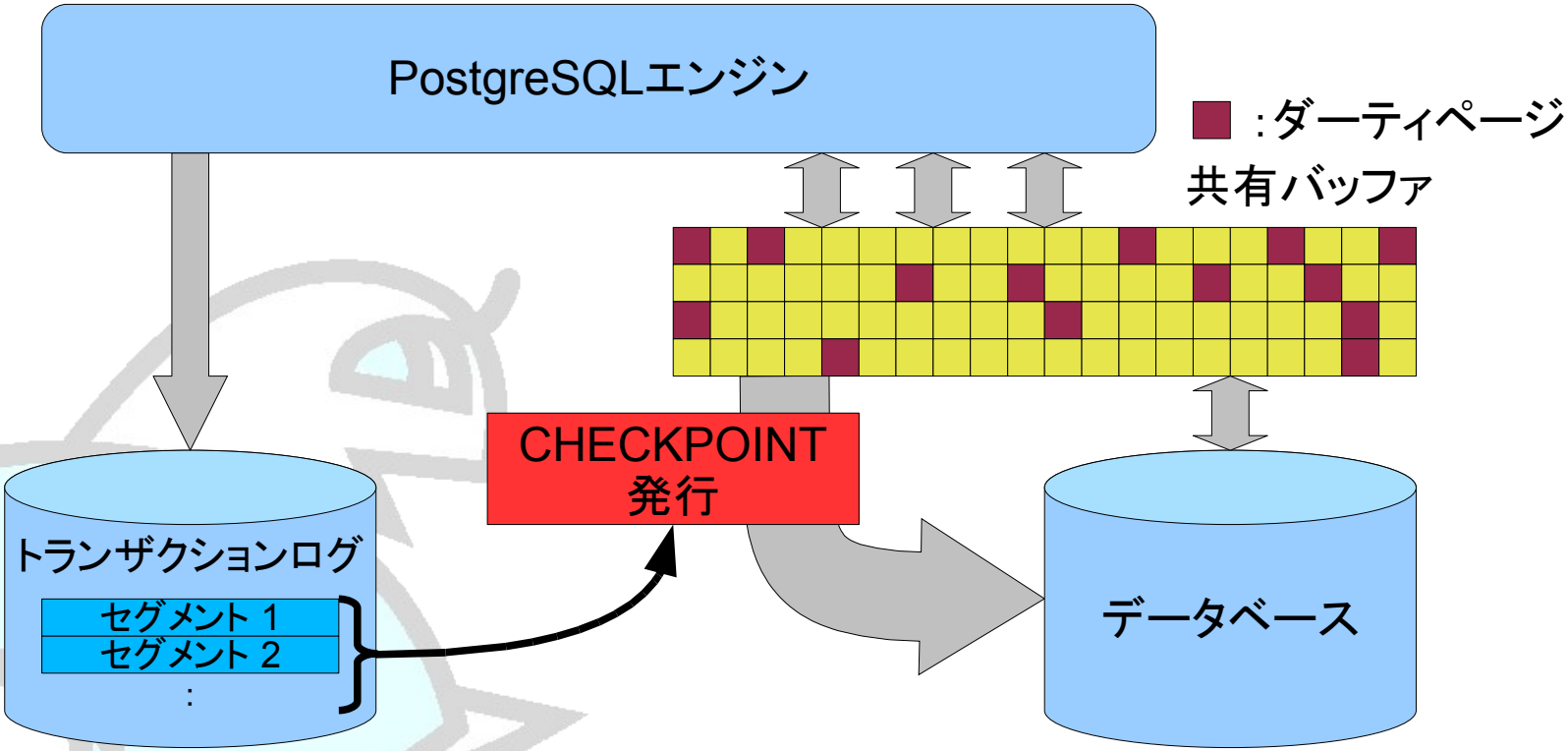


テーブルスペースを利用すると…



チェック5：チェックポイントセグメント数

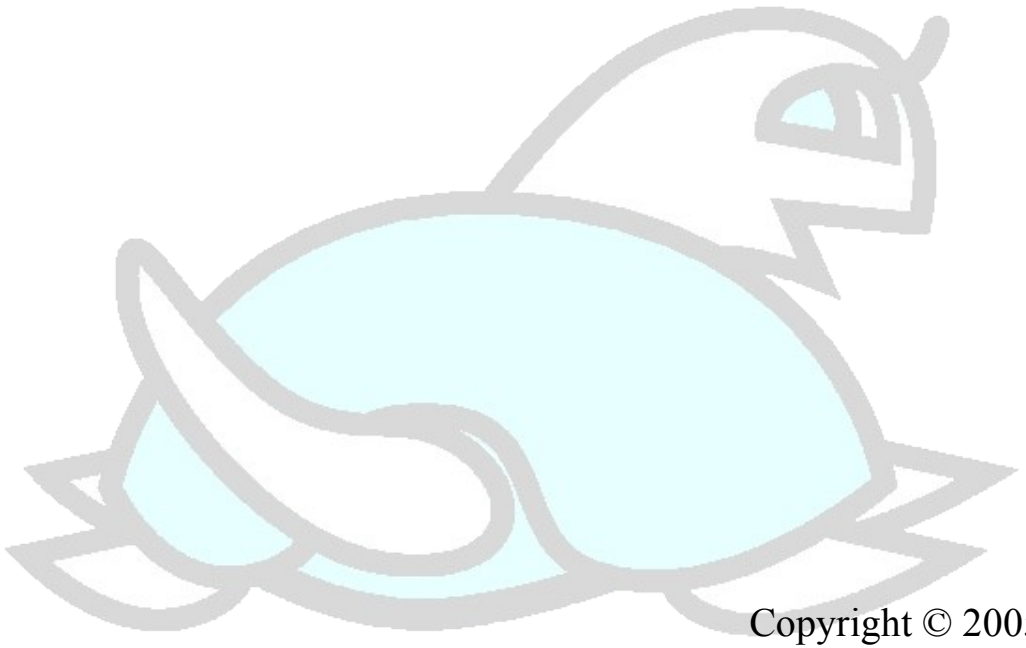
- トランザクションログのセグメント数がこの値に達するとチェックポイントが発生する
- デフォルトでは3セグメント（112MB）
  - 更新の多い環境ではやや少ない。



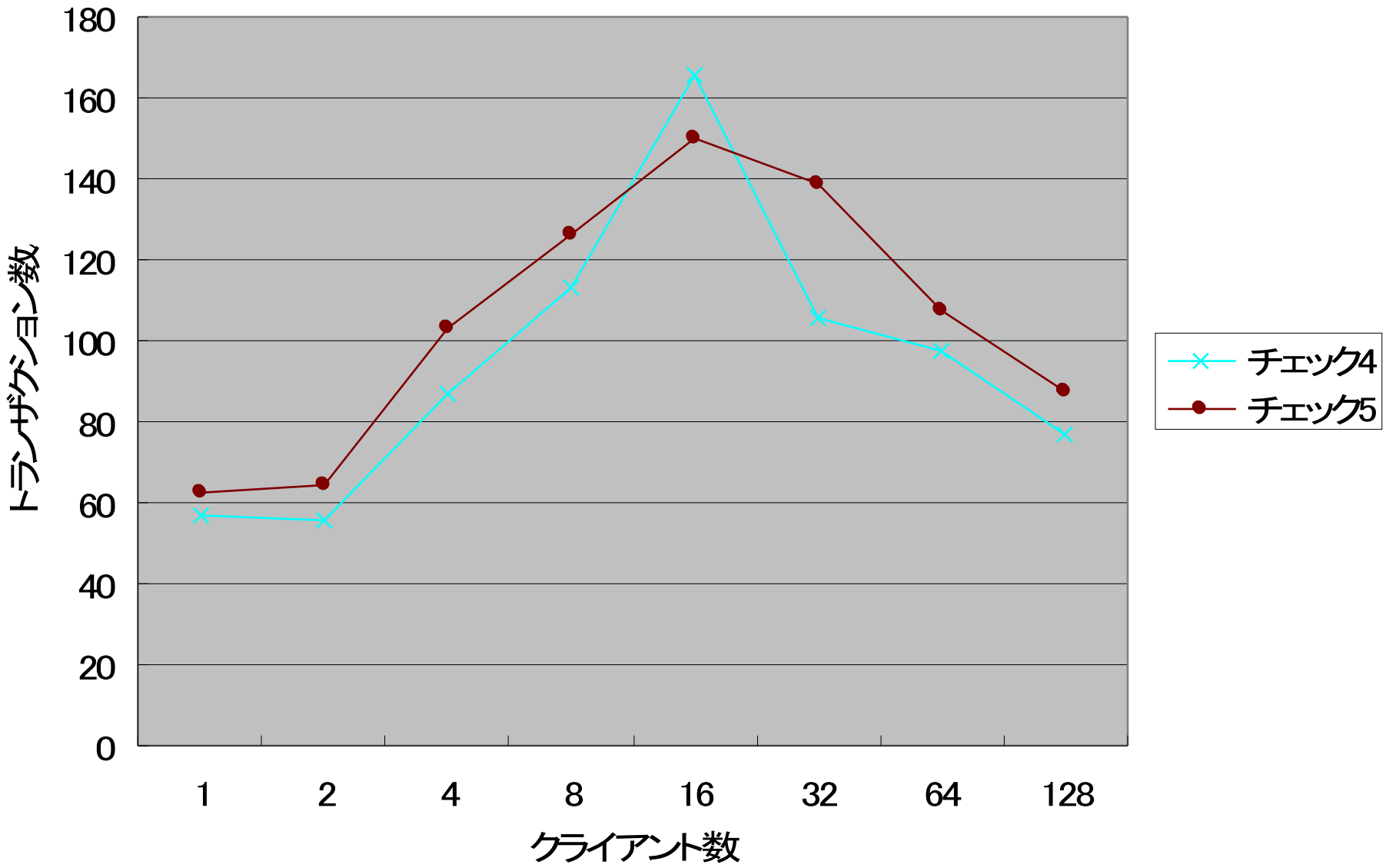
チェックポイントセグメント数の調整

- 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。
  - 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(ディスク使用量は最大で264MB)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…

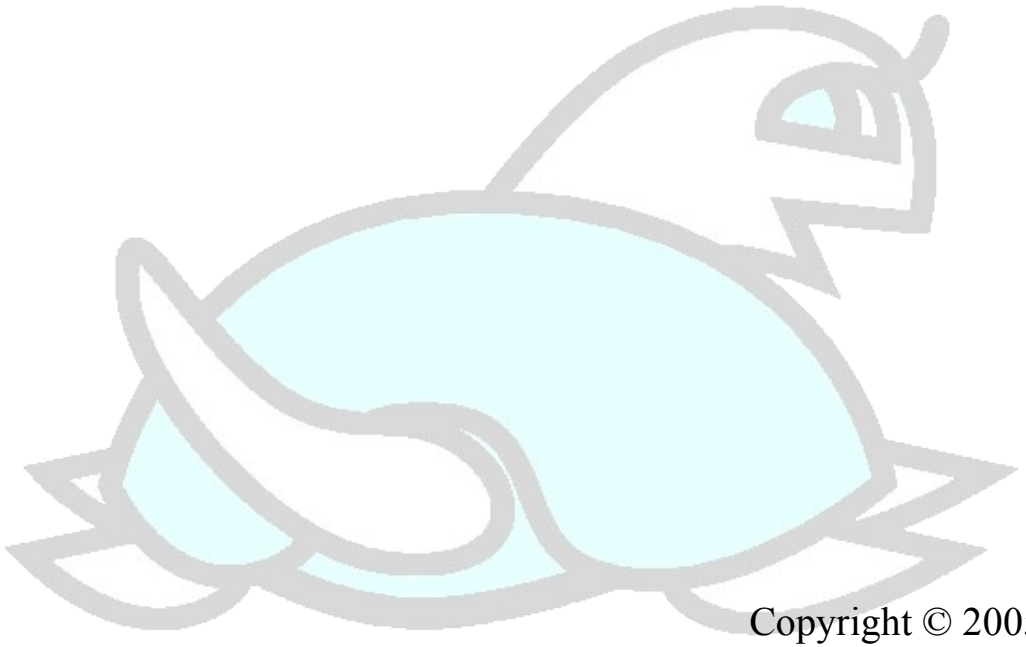


チェック6:ファイルシステムの調整

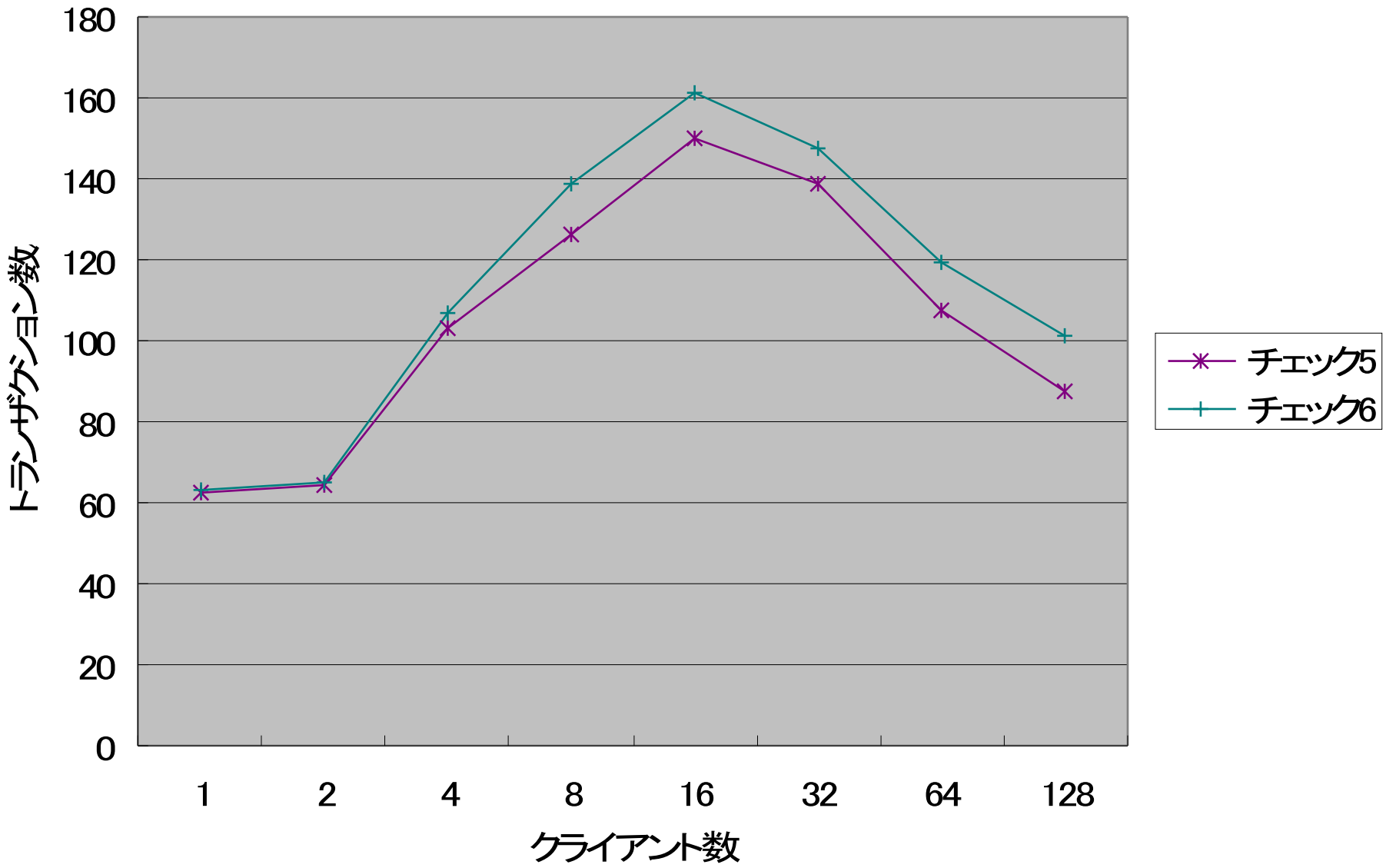
- ファイルシステムがデータベース専用であれば、atime(アクセス時間の記録)を「off」に設定することでパフォーマンスが向上

- 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)。
  - 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

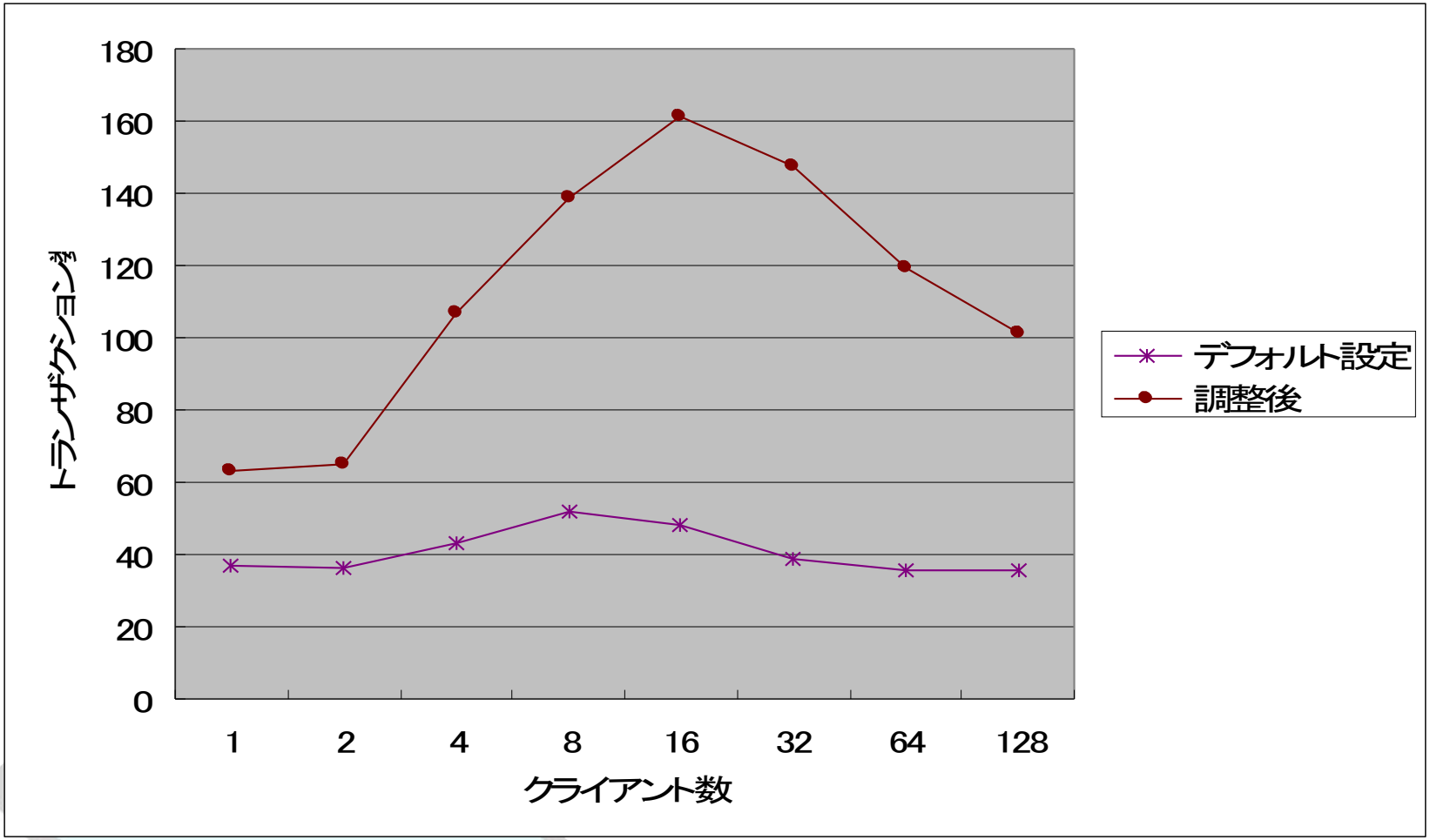
その他のチェック項目：VACUUM

- 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
  - VACUUM
    - ➔ ゴミ掃除を行うためのPostgreSQL特有のSQL命令
  - vacuumdb
    - ➔ VACUUMを行うラッパコマンド
  - pg_autovacuum
    - ➔ データベースを監視して自動的にVACUUMを行うコマンド。

- 定期的にVACUUMを行う必要
  - cronなどで定期的にデータベースのゴミ掃除を行う。
    - ➔ 常に最適なパフォーマンスでPostgreSQLを使おう！

PostgreSQLを最大限に使い切りましょう！



ご清聴ありがとうございました