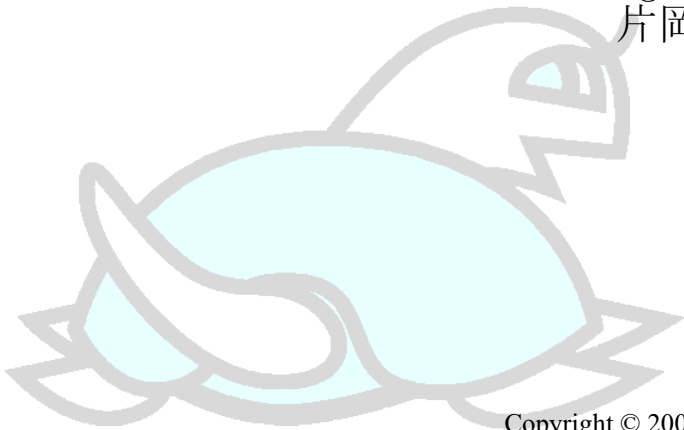


# PostgreSQL 8.0 の概要

日本PostgreSQLユーザ会  
片岡 裕生



Copyright © 2005, Hiroki Kataoka

1

## PostgreSQLの歴史

- INGRES** (1973年) UCBの研究室で開発。PostgreSQLの直系の祖先。商用DBMSのIngresを生む。
- POSTGRES** (1985年) UCBの研究室で開発。バージョン4で終了。商用DBMSのILLUSTRATEを生む。
- Postgres95** (1995年) UCBの学生が開発を引き継ぐ。問い合わせ言語をSQLに変更。
- PostgreSQL** ボランティアの手で開発を継続。Postgres95をバージョン5と見なし、PostgreSQLはバージョン6からスタート。
- 6.0 (1997年1月29日) UNIQUE INDEX
- 6.1 (1997年6月8日) GEQO、シーケンス、ANALYZE機能
- 6.2 (1997年10月2日) トリガ、制約
- 6.3 (1998年3月1日) 副問い合わせ、デッドロック検出、組み込みSQL、UNION句
- 6.4 (1998年10月30日) Windowsネイティブクライアント、ビュー、ルール、PL/pgSQL、マルチバイト拡張機能のオフィシャル化
- 6.5 (1999年6月9日) MVCC、トランザクション隔離レベル、ホットバックアップ、一時テーブル、NUMERIC型、CASE句、INTERSECT句、EXCEPT句
- 7.0 (2000年5月8日) 外部キー、JOIN構文
- 7.1 (2001年4月13日) WAL、TOAST、GUC、外部結合、サーバクライアント間のSSL暗号化
- 7.2 (2002年2月4日) 新しいVACUUM機能、ANALYZEの独立、最大トランザクション数(約40億)の撤廃、OIDのオプション化、アクセス統計機能、MD5パスワード
- 7.3 (2002年11月27日) マルチバイト標準機能化、スキーマ、テーブル型関数、依存関係の管理、アクセス権の大幅な拡張、カラムの削除、関数の引数が32個へ、識別子が63文字へ、ドメイン、
- 7.4 (2003年11月17日) インライン関数、IPv6、インフォメーションスキーマ、pg\_autovacuumコマンド
- 8.0 (2004年末) Windowsネイティブサーバ、アーカイブログ、PITR、セーブポイント、テーブルスペース、カラムのデータ型の変更、CSV対応

Copyright © 2005, Hiroki Kataoka

2

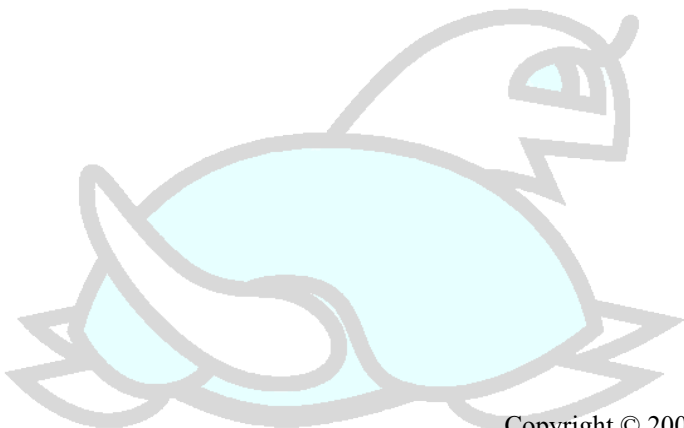
- ユーザ、グループ
- データベース、スキーマ
- 標準SQL
- 各国語対応
- ルール
- トリガ
- 制約、ドメイン
- トランザクション
- トランザクションログ (WAL)、アーカイブログ (※)
- ホットバックアップ
- プロシージャ記述言語
- プログラムインターフェース
- 拡張性



Copyright © 2005, Hiroki Kataoka

3

## PostgreSQL 8.0 の新機能



Copyright © 2005, Hiroki Kataoka

4

- Windows 2000以降に対応
- Windows Installer
- 付属ツール
  - クライアントアプリケーション
    - psql (SQLインタプリタ)
    - pgAdmin III
  - クライアントドライバ
    - ODBC Driver
    - JDBC Driver
    - OLE DB Provider
    - .NET Data Provider (Npgsql)
  - 開発ツール
    - libpq (C言語用クライアントライブラリ)
    - ecpg (埋め込みSQL)
  - ドキュメント
  - PostgreSQL 8.0オンラインマニュアル

## pgAdmin III

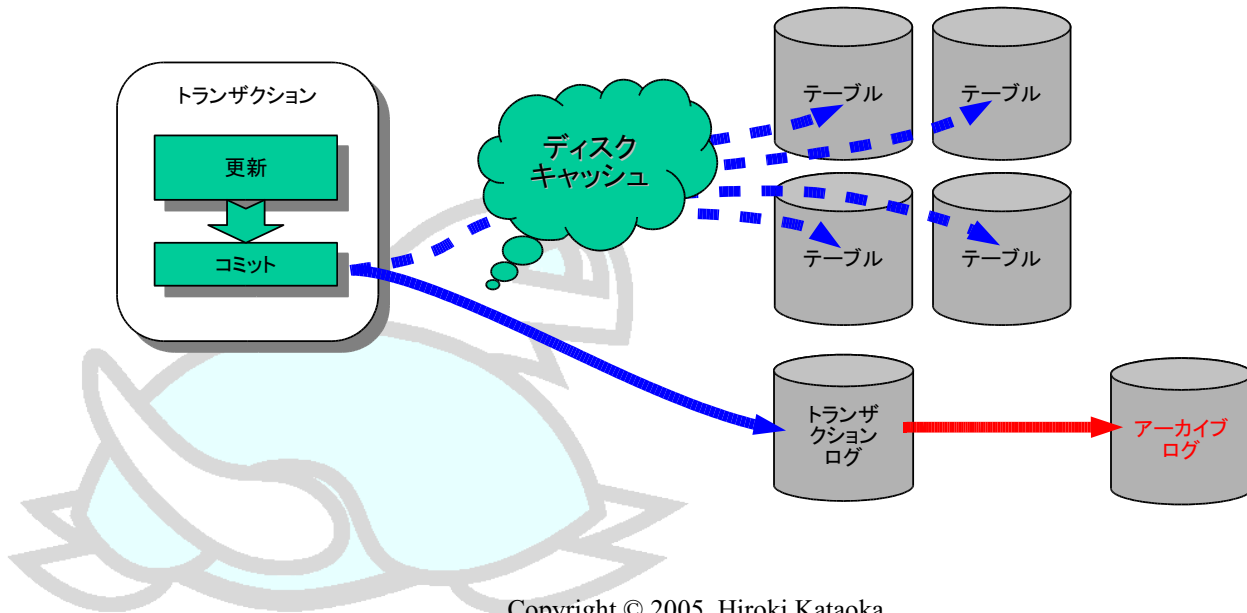
The screenshot shows the pgAdmin III application window. The left pane displays a tree view of the PostgreSQL Database Server 8.0-beta2-dev3 (localhost:5432). The tree is expanded to show the 'kataoka' database, which contains several objects including casts, languages, schemas (public, test), aggregates, comparisons, domains, functions, trigger functions, operators, operator classes, sequences (table1\_id\_seq, table2\_id\_seq), tables (table1, table2), types, and views. The 'table1' table is selected, and its properties are shown in the right pane.

プロパティ	値
名前	table1
OID	17661
オーナー	kataoka
ACL	
主キー	id
行(概算)	1000
行(数)	8
継承テーブル	(ない)
継承されたテーブルカ...	0
OIDを持つ?	(はい)
システムテーブル?	(ない)
コメント	

```
-- Table: test.table1
-- DROP TABLE test.table1;

CREATE TABLE test.table1
(
  id serial NOT NULL,
  name text,
  addr text,
  tel text,
  CONSTRAINT table1_pkey PRIMARY KEY (id)
)
WITH OIDS;
```

- トランザクションログ (WAL) を自動的にバックアップ
  - ・ 物理バックアップとともに用いることでメディアリカバリが可能に。
- バックアップ手段は自由に設定可能
- 設定パラメータ  
`archive_command = 'cp %p /mnt/server/archivedir/%f'`

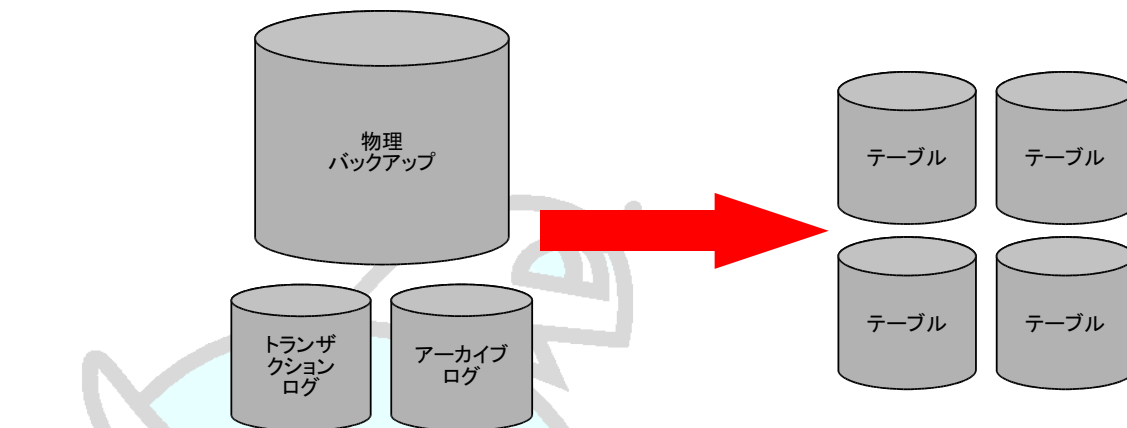


Copyright © 2005, Hiroki Kataoka

7

## Point In Time Recovery

- 物理バックアップとアーカイブログ、トランザクションログを用いてデータベースをリカバリ
- リカバリ処理は任意の時点で停止可能
  - ・ オペレーションミスを取り戻したい場合など、データベースを過去に戻すことが可能に！



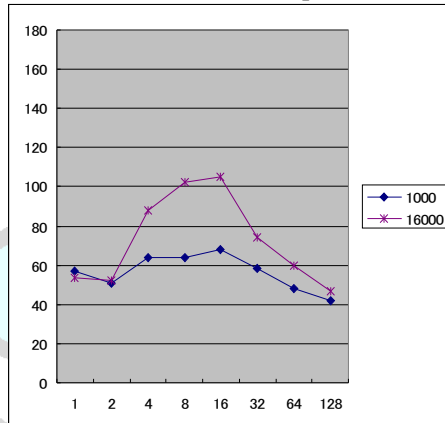
Copyright © 2005, Hiroki Kataoka

8

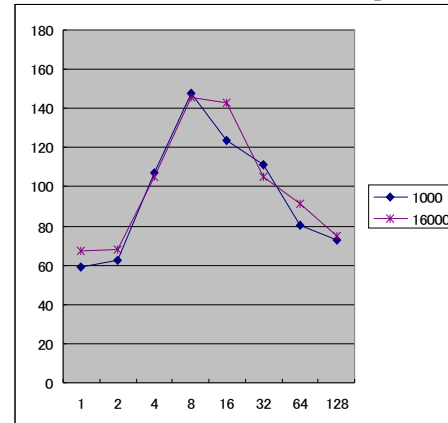
- オブジェクトを任意のディレクトリに配置可能！
  - ・ DATABASE、SCHEMA、TABLE、INDEX
- ディスクI/Oの負荷分散など
- 使用例  

```
CREATE TABLESPACE myspace LOCATION '/db1/myspace';  
CREATE TABLE ~ TABLESPACE myspace;  
ALTER INDEX ~ SET TABLESPACE myspace;
```
- initlocationは廃止

全て1つのディスクの場合 (tps)



2つのディスクに振り分けた場合 (tps)

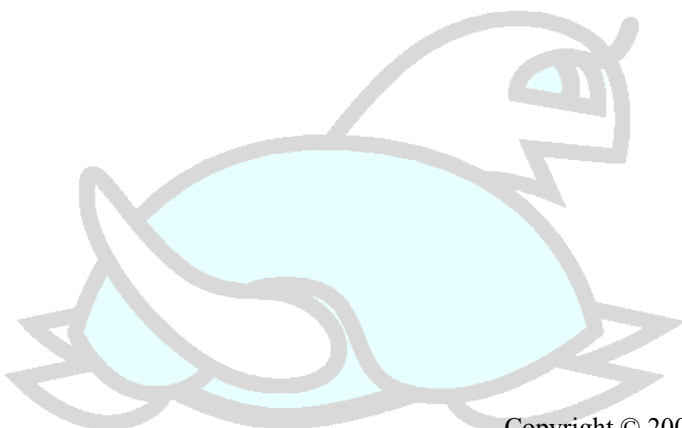


Copyright © 2005, Hiroki Kataoka

9

- トランザクションを部分的にロールバック
- 使用例  

```
BEGIN;  
:  
SAVEPOINT point1;  
:  
何らかのエラーが発生!  
ROLLBACK TO point1;  
:  
COMMIT;
```

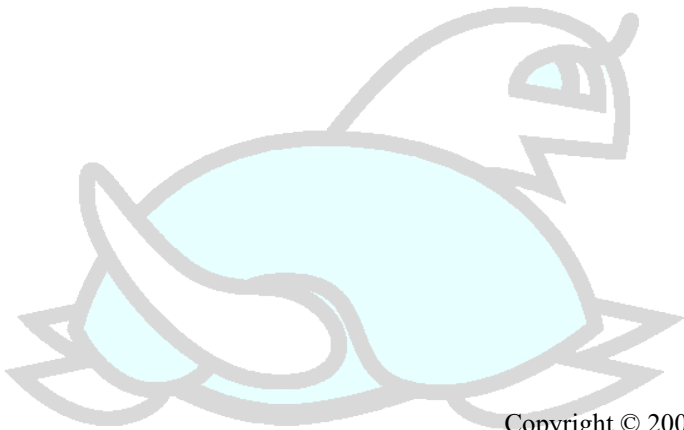


Copyright © 2005, Hiroki Kataoka

10

- 各プロシージャ記述言語で例外処理をサポート
- 使用例 (PL/pgSQL)

```
BEGIN
  INSERT INTO tbl1 ~;
  RETURN TRUE;
EXCEPTION
  WHEN UNIQUE_VIOLATION THEN
    UPDATE tbl1 SET ~;
END;
```



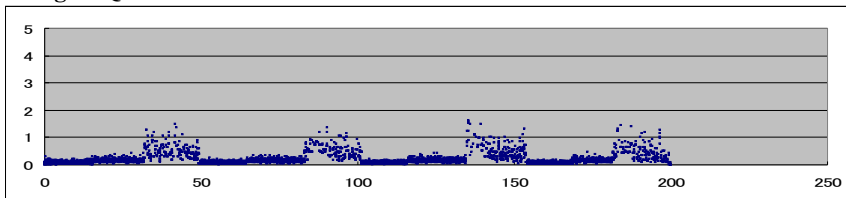
Copyright © 2005, Hiroki Kataoka

11

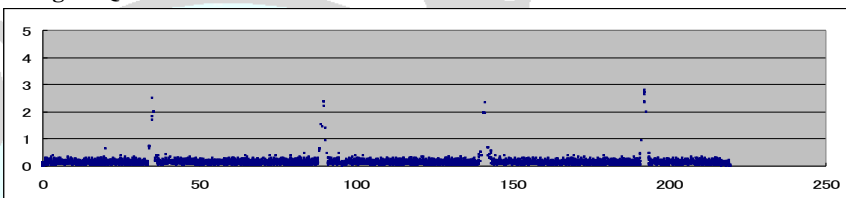
## ライタープロセス

- キャッシュ内のダーティページを少しずつ書き出すバックグラウンドプロセス
- レスポンスタイムの安定化
- 設定パラメータ
  - ◆ bgwriter\_delay = 200 (ミリ秒)
  - ◆ bgwriter\_percent = 1
  - ◆ bgwriter\_maxpages = 100

PostgreSQL 7.4



PostgreSQL 8.0



### pgbenchによるベンチマーク

縦軸: レスポンスタイム (秒)  
横軸: 経過時間 (秒)

DB規模: 500万レコード  
約800Mバイト  
接続クライアント数: 8  
総トランザクション数: 12800

### PostgreSQLの設定

shared\_buffer: 16000  
checkpoint\_segments: 3

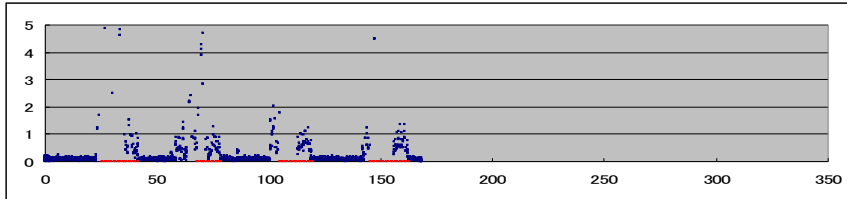
Copyright © 2005, Hiroki Kataoka

12

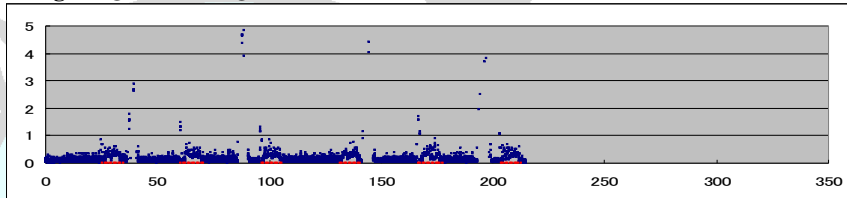
# バッファマネージャの改良

- 今までは…
  - 大きいテーブルのシーケンシャルスキャンなどがあると、使用頻度の高いキャッシュページでもフラッシュアウトされていた。
  - VACUUMやANALYZEでも問題
- 8.0では…
  - キャッシュページの使用頻度も考慮されるようになった。

PostgreSQL 7.4 (LRU)



PostgreSQL 8.0.2 (2Q)



pgbenchによるベンチマーク

縦軸: レスポンスタイム(秒)  
横軸: 経過時間(秒)  
赤線: シーケンシャルスキャン  
(COUNT(\*)の実行)

DB規模: 500万レコード  
(約800Mバイト)  
接続クライアント数: 8  
総トランザクション数: 12800

PostgreSQLの設定

shared\_buffer: 16000  
checkpoint\_segments: 3

Copyright © 2005, Hiroki Kataoka

13

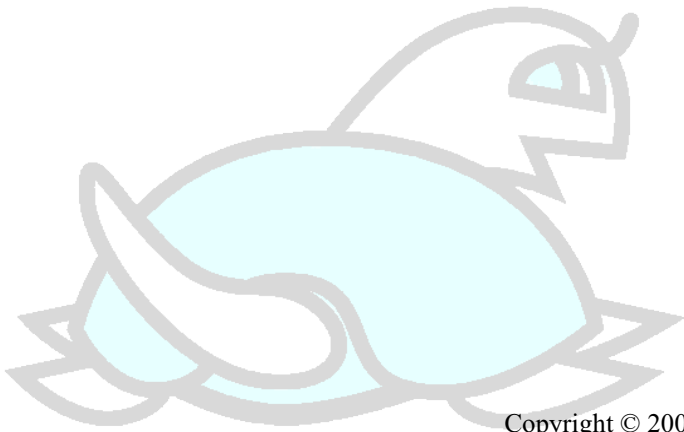
# その他の拡張

- CSVフォーマットのサポート
  - COPY文、¥copyコマンドがCSVフォーマットに対応
  - 使用例  
COPY ~ TO '~.csv' WITH CSV;
- カラムのデータ型の変更が可能
  - テーブル作成後にカラムのデータ型が変更可能
  - ALTER TABLE ~ ALTER COLUMN ~ TYPE ~ USING ~;
- VACUUMの拡張
  - 一定量のVACUUM処理ごとにデレイを入れることが可能になった
  - 高負荷システムにおけるVACUUM処理のパフォーマンスインパクトを軽減
  - 設定パラメータ
    - vacuum\_cost\_delay = 0 (ミリ秒)
    - vacuum\_cost\_page\_hit = 1
    - vacuum\_cost\_page\_miss = 10
    - vacuum\_cost\_page\_dirty = 20
    - vacuum\_cost\_limit = 200

Copyright © 2005, Hiroki Kataoka

14

- 設定パラメータの変更
  - ◆ virtual\_host, tcpip\_socket → listen\_addresses
  - ◆ sort\_mem → work\_mem
  - ◆ vacuum\_mem → maintenance\_work\_mem
  - ◆ log\_pid, log\_timestamp, log\_source\_port → log\_line\_prefix
  - ◆ syslog, その他 → log\_destination
  - ◆ max\_expr\_depth → max\_stack\_depth
- その他
  - ◆ length()関数はCHAR(n)型のパディングスペースを数えなくなった



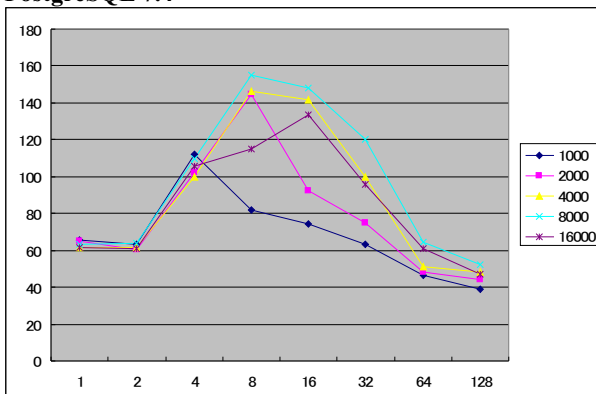
Copyright © 2005, Hiroki Kataoka

15

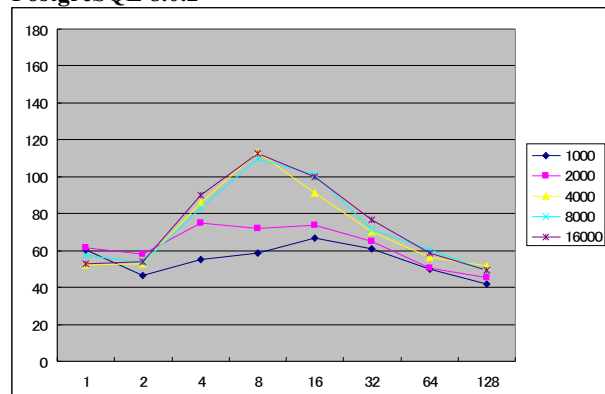
## PostgreSQL 7.4との性能比較(1)

- 8.0はやや強めの安定志向
  - ◆ ピーク性能劣化の原因はライタープロセス

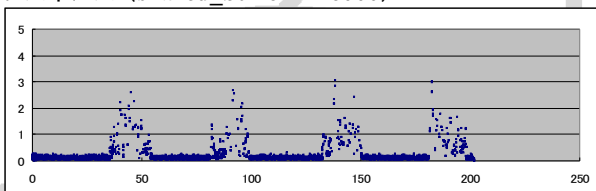
PostgreSQL 7.4



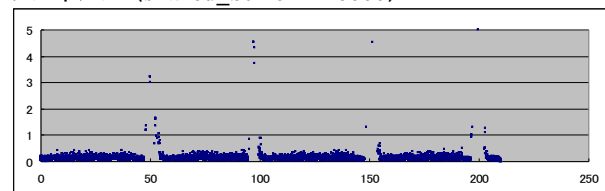
PostgreSQL 8.0.2



レスポンス (shared\_buffer = 16000)



レスポンス (shared\_buffer = 16000)



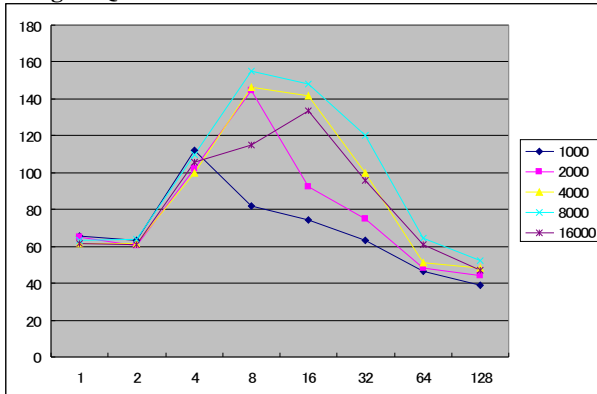
Copyright © 2005, Hiroki Kataoka

16

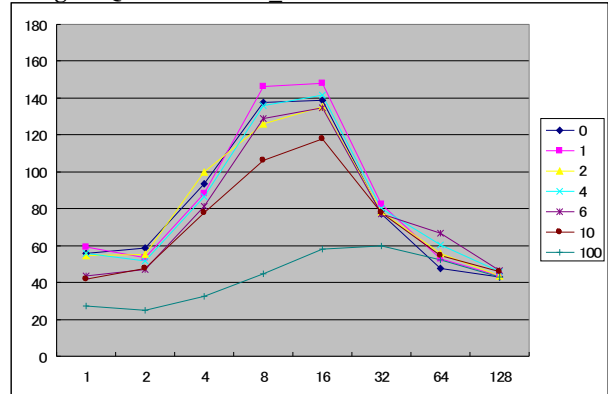
# PostgreSQL 7.4との性能比較(2)

- ライタープロセスを調整すると…
  - ♦ 7.4に近いピーク性能を出すことは可能

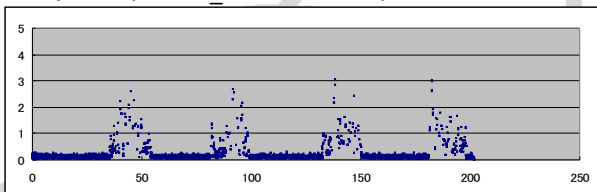
PostgreSQL 7.4



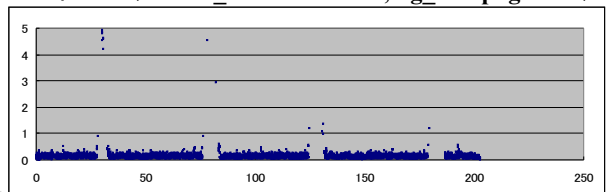
PostgreSQL 8.0 (shared\_buffer = 16000)



レスポンス (shared\_buffer = 16000)



レスポンス (shared\_buffer = 16000, bg\_maxpages = 4)

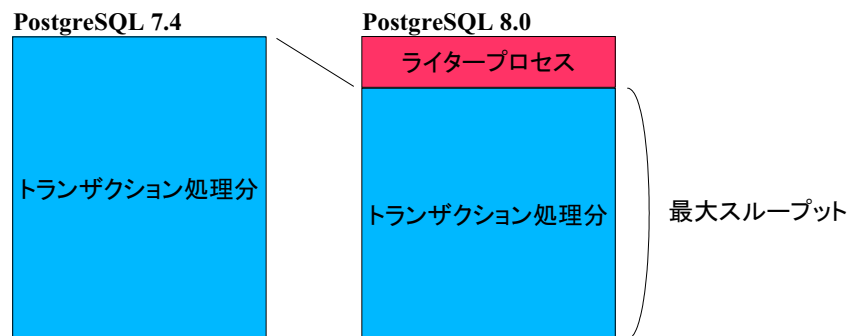


Copyright © 2005, Hiroki Kataoka

17

# PostgreSQL 8.0の性能

- PostgreSQL 8.0が本来の性能を発揮するためには…
  - ♦ ライタープロセスのための余裕を残した運用が必要

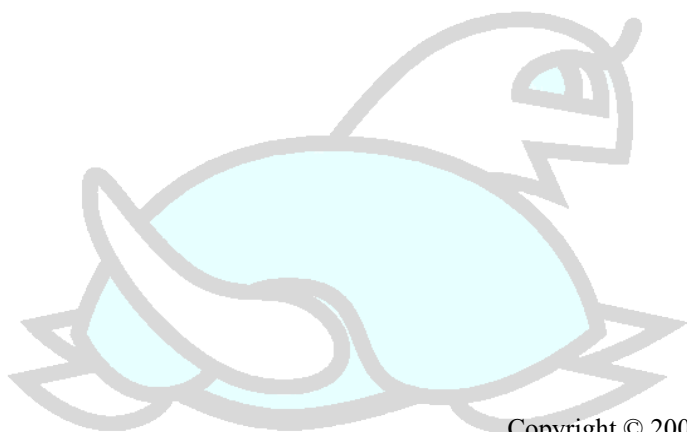


- どうしても7.4と同じハードウェア&負荷で同じ性能が必要なら…
  - ♦ ライタープロセスを止めるか、調整する必要あり
  - ♦ (前ページのレスポンス試験の調整例)
    - bgwriter\_delay = 200 (ミリ秒)
    - bgwriter\_percent = 100
    - bgwriter\_maxpages = 4

Copyright © 2005, Hiroki Kataoka

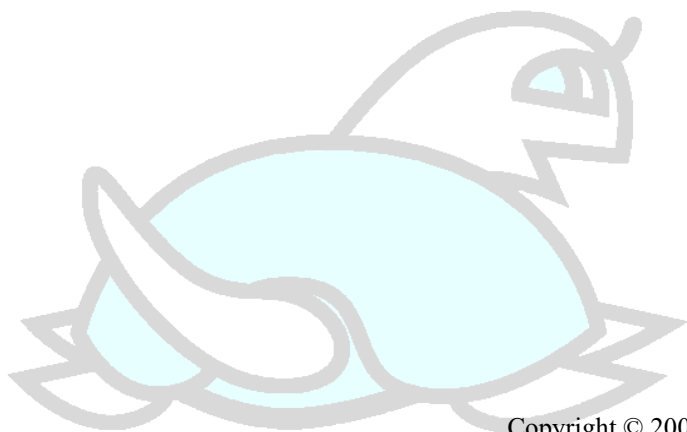
18

- PostgreSQL 8.0の進化の方向性
  - 安定性の向上(レスポンスタイムなど)
  - 安全性の向上(PITRなど)
  - 開発をより容易に(CSV対応やカラムのデータ型変更など)
  - 適用範囲の増大(ネイティブWindows版サーバなど)
- PostgreSQLは熟成期に入ったと考えられる



Copyright © 2005, Hiroki Kataoka

ご清聴ありがとうございました



Copyright © 2005, Hiroki Kataoka