

最新版PostgreSQL9.0.1の新機能

NPO法人 日本PostgreSQLユーザ会 九州支部



PostgreSQL

日本PostgreSQLユーザ会九州支部

レジメ

- PostgreSQLとは、最新版9.0までで出来る事。
- 9.1 alpha4 の事。
- レプリケーションデモ。
- 質疑応答。

- プラットフォーム

UNIX系列、Linux、Microsoft Windows、OS/2 など多くのOS

- 最新のバージョン

9.0.3 (2011-02-01) (9.0.0は 2010.9.20 release)

8.4は、2009年6月29日リリース。1年に1回メジャーバージョンアップが目標。

気の早い話ですが、9.1はalpha4です。9.0の頃から、commit festaという開発手法がとりいれられています。



PostgreSQLの機能 を実装順に振り返る

てたら、時間がいくらあっても足りないので、ダイジェストでいきます。

POSTGRES時代

- 1986年 - POSTGRESプロジェクトを発足
- 1987年 - プロトタイプが完成
- 1989年 - POSTGRES 1 を限定的にリリース
- 1990年 - POSTGRES 2 のリリース。
- 1991年 - POSTGRES 3 のリリース。
- 1993年 - POSTGRES 4.2 POSTGRESプロジェクトが終了

あくまで、学術研究用でしたので、用途がすめばおさらばです。

Postgres95時代

- 1995年5月1日 – Postgres95 0.01 がリリース
- 1995年9月5日 - Postgres95 1.0

カリフォルニア大学バークリー校 (UCB) は、Postgres をオープンソースライセンスであるBSDライセンスのもとで公開したのでそれをもとに、SQL準拠等へ改良。

PostgreSQL時代

Postgres95 プロジェクトは、プロジェクトの名称を、SQL のサポートをしているという意味をこめて PostgreSQL に変更。1997年1月に PostgreSQL プロジェクトとしての最初のバージョンである、PostgreSQL バージョン 6.0 が公開されました。

- 1997年1月29日 - PostgreSQL 6.0
- 1998年10月30日 - PostgreSQL 6.4
- 1999年6月9日 - PostgreSQL 6.5
- 2000年5月8日 - PostgreSQL 7.0
- 2001年4月13日 - PostgreSQL 7.1
- 2002年2月4日 - PostgreSQL 7.2
- 2002年11月27日 - PostgreSQL 7.3
- 2003年11月17日 - PostgreSQL 7.4
- 2005年1月19日 - PostgreSQL 8.0
- 2005年11月8日 - PostgreSQL 8.1
- 2006年12月5日 - PostgreSQL 8.2
- 2008年2月4日 - PostgreSQL 8.3
- 2009年6月29日 - PostgreSQL 8.4



メンテナンス終了

- PostgreSQL 6.4 : PL/pgSQL, マルチバイト文字列サポート の追加
- PostgreSQL 6.5 : MVCC の追加
- PostgreSQL 7.0 : 外部キー制約の追加
- PostgreSQL 7.1 : WAL, TOAST, OUTER JOIN の追加
- PostgreSQL 7.2 : コンカレントVACUUM の追加
- PostgreSQL 7.3 : スキーマ, ドメイン, PREPARE の追加
- PostgreSQL 7.4 : IPv6 の追加
- PostgreSQL 8.0 : **Microsoft Windows**対応, サブランザクション, PITR, 表領域の追加



メンテナンス中バージョン

PostgreSQL 8.1 : 2相コミット, ROLE, 行共有ロック, テーブル・パーティショニング の追加

PostgreSQL 8.2 : ウォームスタンバイ, GIN の追加

PostgreSQL 8.3 : 更新処理性能の向上, XMLデータ型, 全文検索, JIS X 0213サポート, ENUM型, UUID型

PostgreSQL 8.4 : 再帰SQL, ウィンドウ関数, 列単位のアクセス制御, SQLと関数の性能解析機能

特徴

- 標準SQLへの準拠 (SQL:2003にほぼ対応)
- 並列実行(マルチCPU)性能に重点
- 日本語(マルチバイト)対応
- 地理情報システム (GIS)
- 簡単に試せる手軽さ
- PostgreSQL のレプリケーション(9.0から)
-pgpool-II , Slony-I , PGCluster

8. 1あたりから対応

- 8. 1では、8cpuあたりまで性能がでる
- 8. 2では、16cpuあたりまで性能がでる。
(ロック周りのボトルネック解消により,そろそろ考え時か?)
- IPAのサイトの性能評価あたりを参考にしてください。

<http://ossipedia.ipa.go.jp/capacity/index.php>

- 日本語, 中国語などの各国の EUC, Unicode, mule internal code, ISO-8859-* がデータベース作成時に選択可能。JIS2004対応。
- テーブル名・カラム名・データにマルチバイト文字が使用可能
- 正規表現検索・LIKE 検索が使用可能
- substring() などの文字列関数でのサポート
- フロントエンドとバックエンド側での自動的にエンコーディング変換。
- ユーザ定義のエンコーディング変換を作成可能。



local = c(no_locale)

- `initdb -encoding=UTF8 --no-locale`
- DB毎にロケールの設定ができるなった by8.4。
- 8.4からは、localeの仕様変更により、`createdb`に注意が必要
(`initdb -e EUC_JP` としてUTF8のDBを作りたい場合)
`#createdb -E UTF8 -T template0 table_name`

取り上げる項目

- 追記型という誤解。
- バキュームが面倒という誤解。
- レプリケーションが実装されるまで。

autovacuum (8.1 8.3)

- バキュームを自動でしてくれる。
- 8.3から、postgresql.confのデフォルトがONになる。
- reindex、vacuum full等手動でする必要はあります。
- 運用後、値が適切か確認する必要がある。
select pg_database_size('databasename');
free_spaceの確認(ただし、8.3まで)

autovacuum (8.4)

■ Free Space Mapのメモリ管理の自動化

VACUUM によってガベージが回収された空き領域を次回以降、再利用するため追跡する機構が **Free Space Map** です。

8.3 までは起動時に固定量を割り当てていましたが、8.4 では自動的に確保／調整。

設定ミスでメモリが不足 すると空き領域が再利用されず DB 肥大化の原因になっていた。

HOT (Heap Only Tuples) (8.3)

- 追記型アーキテクチャの弱点を補完。
削除領域を利用。
- VACUUMの必要性を減らす。
- 更新対象列にインデックスカラムが含まれてない場合に有効。
→ indexの更新頻度を減らします。

- 8. 2からは、
 - > archive_timeout(postgresql.conf)
 - > pg_switch_xlog(システム管理関数)
新しいトランザクションログファイルへの強制変更で、任意にアーカイブのタイミングを指定できる。
- pg_start_backup, pg_stop_backup コマンド
オンラインバックアップの実行準備と停止。
- で、バックアップの完了。

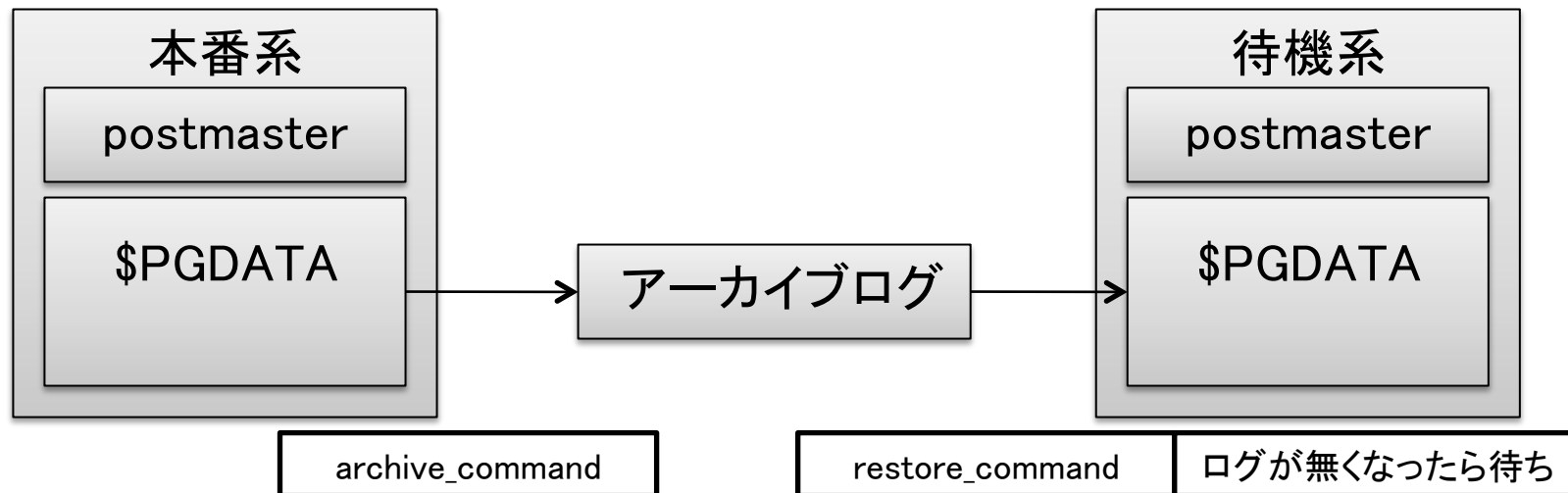
PITRリカバリ

- バックアップを展開(パーミッションに注意)
 - \$PGDATA/pg_xlogは不要
- \$PGDATAにrecovery.confを配置
 - share/recovery.conf.sampleを参照
- recovery.confの、restore_commandを設定
- postmasterを起動

- 通常のPITRのリカバリは、アーカイブログの最後までを再生する。
- アーカイブログの途中の任意の時点までリカバリするのがRPIT。
 - 間違ってデータを消した時点までとか。
- recovery.confの
 - recovery_target_time(時間で指定)
 - recovery_target_xid(トランザクションIDで指定)で任意の時点を設定

WARM STANDBY

- アーカイブされたものを待機系が次々読み込む。
- 8.2ではシェルスクリプト等を自作する必要がある。
- 8.3からはpg_standbyがcontribに追加された。



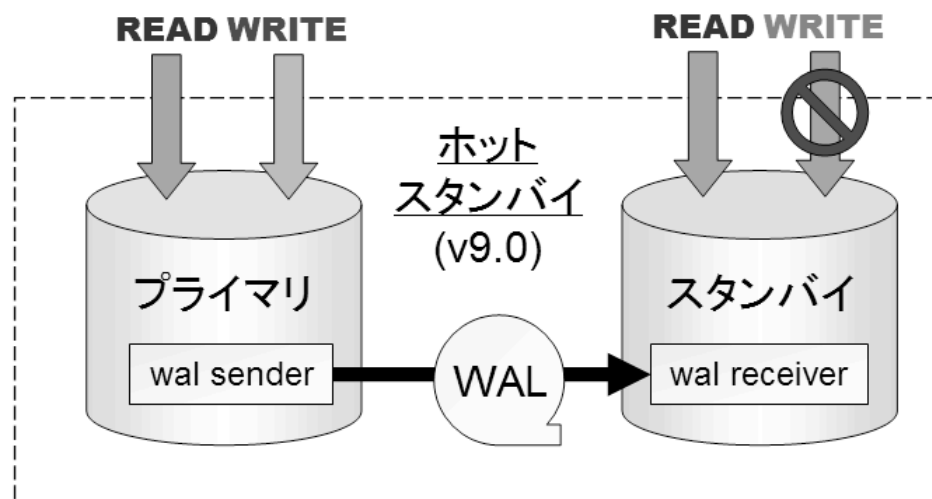
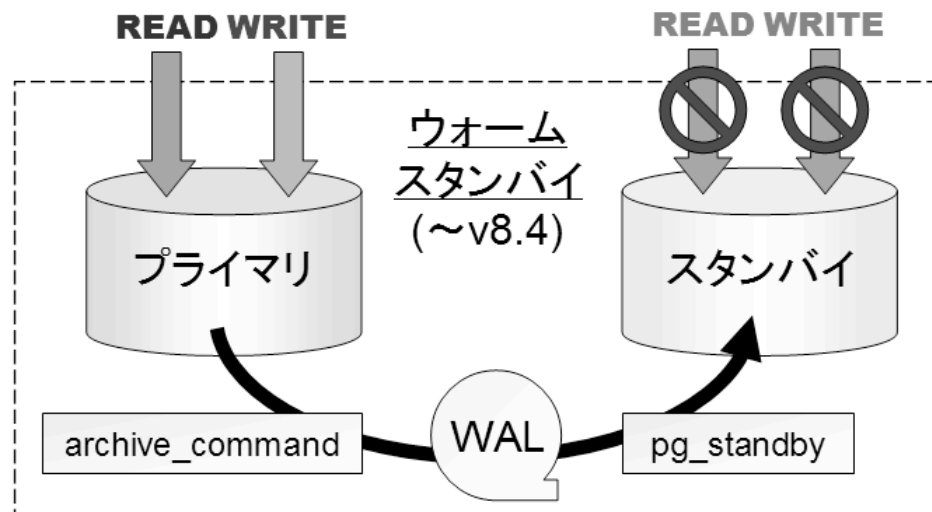
もう、追記型と呼ばないで！！

- 追記型に相当する英語がみあたらない。
- Fillfactor, hot,などを考慮すると、もう追記型という言葉では説明できない。
- 追記型=updateが遅いというのは昔のはなし。
- 昔は、データを追加で書き込んでいたという事で。

- 更新 (UPDATE) も遅くない。
- VACUUME も問題無い。(AUTO)
- 性能もそこそこでる。
- まあ、レプリケーションとか、運用ツールとかたりないところもあるにはある。

- ホット・スタンバイ(Hot Standby)
- ストリーミング・レプリケーション
(Streaming Replication)
- window関数の強化
- 64bit版 Windowsサポート
- O/R map 対応(不必要JOINの削除)
- VACUUM FULLの高速化
ただし、可能なら使わない
- EXPLAINの出力の多様化
(text | xml | json | yaml)

- ラージオブジェクトのアクセス制御
- Pgbench -j スレッド数 (マルチスレッド化)



- 待機サーバに対して参照ができる。
- ログシッピングなので、物理的に完全に同じ状態。(vacuumの結果も)
- 遅延はゼロではない。

<http://lets.postgresql.jp/documents/technical/9.0/1>
板垣 貴裕氏の資料から

- 別紙、ハンズオン資料を参照

<http://www.itm.ne.jp/postgresql/20101127/201011-jpug-kyusyu-seminar.zip>
に、UPLしています。あと一週間ぐらいかな。

- 操作単位での転送なので、遅延がすくない。(以前はファイル単位)
- 複数のスタンバイ・サーバへの転送可(時間とはトレードオフ)
スタンバイサーバの追加・切り離しはいつでも。



- カスケーディングは、出来ない。
スタンバイ1から、スタンバイ2へのストリーミング。
- フェイルオーバーは無い。
Heartbeatなどと併用すること。
- 参照の自動分散はできない。
Pgpool-II や、PL/Proxyなどを利用。
- メジャーバージョンの統一が必要
- 全体の複製で一部の複製ができない。
Slony-I を利用。

デモ

- MERGE
(行が存在すればupdate なければinsert)
- SE-PgSQL 外部セキュリティ管理連携
- SQL/MED 外部データの直接読み取り、遠隔DB
データ連動
- 範囲型, JSONデータ型
- wal_buffers の自動チューニング

- Let'sPostgres に詳しくかいてます。

- `standard_conforming_strings = on`
エスケープに注意が必要。(offにした方がいいかも)

- レプリケーション周りの機能が增加。

`pg_stat_replication`:

マスタサーバでのレプリケーション状態確認

`pg_last_xact_replay_timestamp()`:

スタンバイサーバでの最終変更日時

`pg_basebackup`

ベースバックアップ用ツール

`restart_after_crash` パラメータ 自動リカバリをするか



JPUGについて

- 関東(東京)を本部として、全国に支部を形成
北海道、東北、新潟、名古屋、関西、四国、中国、九州、
沖縄と全国に9支部(2009年10月現在)

ほぼ全国です。

- 海外に行ったり、海外から講師をお招きしたりとワールドワイドな活動をしています。
- 定例行事(本部)
 - 事例紹介セミナー
 - PostgreSQLカンファレンス(活動報告)
 - 合宿(2月23.24日に嬉野温泉にて)
 - その他、OSC(オープンソースカンファレンス)及び、DevelopersSummit 等各種セミナーへの参加
- 各支部ベースでの、勉強会やセミナー
ほとんどの支部で、講師お招きセミナーや勉強会を行っています。

- 当支部は、JPUGの九州支部としてPostgreSQLの九州における普及を目的としたユーザ・グループです。
- 2002年6月15日に日本ポストgresユーザー会により正式承認されました。(現在、7年半運営)
- 現在九州支部の会員数(ML登録数)は68名です。(2010年12月7日現在) 随時募集中です。

事務局

〒810-0042 福岡市中央区薬院3-13-11 4F

TEL 092-525-0081 FAX 092-525-0082

支部長 清末 直 (株式会社アイティマネジメント)

副支部長 江藤 博文 (佐賀大学学術情報処理センター)



- 年に一回のお招きセミナー(60～100名程度)
- 毎月の定例会(……というなの飲み会)
毎月第二金曜日19:00～、定例会を開いています。たとえば、勉強会や、セミナーの下準備や東京カンファレンスのフィードバック等々
- 支部勉強会(年2～3回、30名程度)
postgresql.confファイルを読む、等。

活動してるよ！！

**DBの会話とかをしましょう、
MySQLもFirebirdも、Oracleも
DB2,SQLServerも
もちろんPostgreSQLの話を**





ユーザ会でまっています。

僕はJPUGのマスコットなのだ

