オープンソースデータベース PostgreSQL最新動向のご紹介

PostgreSQL最新動向＆活用事例セミナー

2013-06-27　14:10～14:50

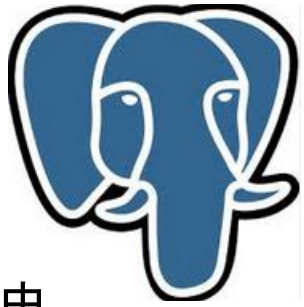
TOC

- PostgreSQL の概要
- 次期バージョン PostgreSQL 9.3 のご紹介

講演者
- SRA OSS, Inc. 日本支社マーケテイング部
  PostgreSQL技術グループ長　高塚 遙
- 主として PostgreSQL のヘルプデスク、導入構築、コンサルティング等の業務を担当

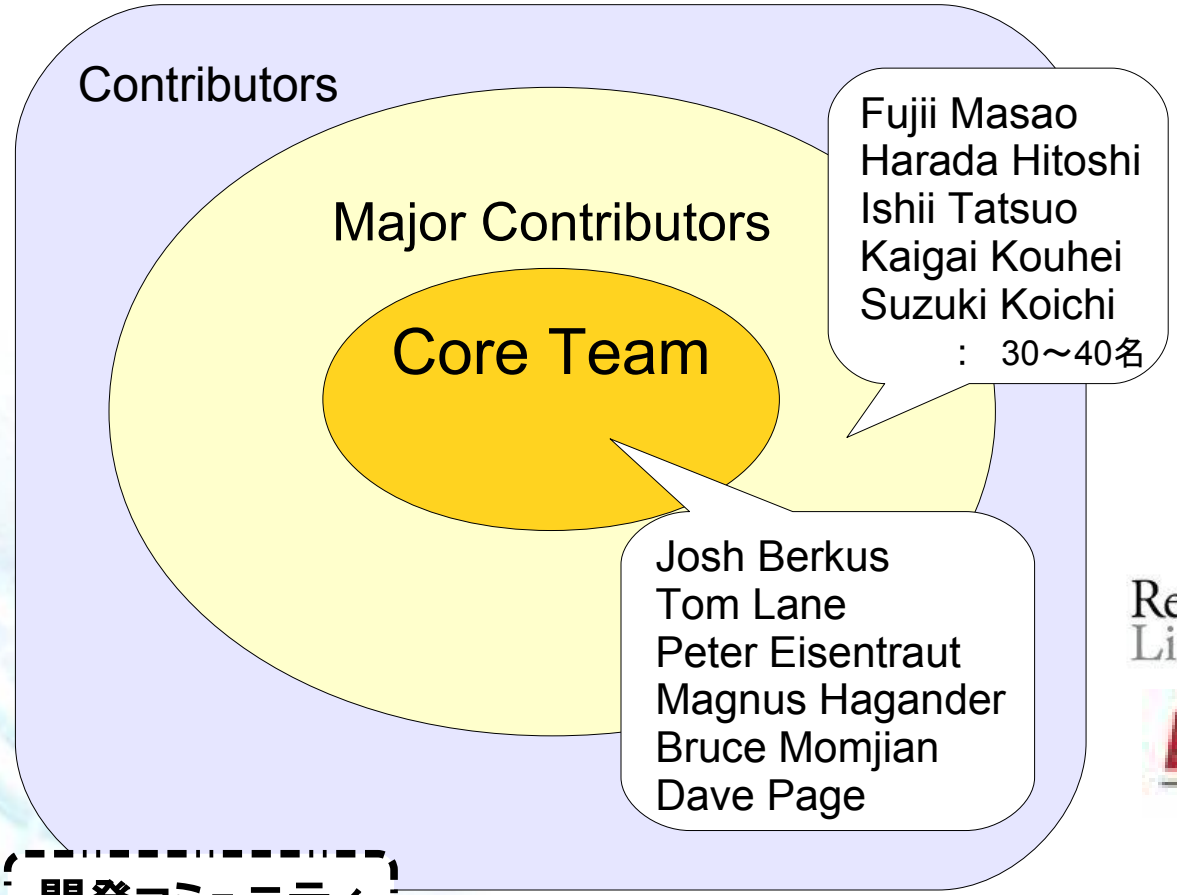
PostgreSQLの概要

改めて・・・ PostgreSQL とは



- 代表的なオープンソースRDBMS
  - Ingres(1970～ UCB) を先祖に持つ
    - PostgreSQL 6.0 (1996 ～) からでも 15年以上の歴史
  - BSDタイプのライセンスで配布
    - PostgreSQL Global Development Gruop と University of California が著作権を持つ
  - ひとつのオーナー企業、オーナー個人を持たない
    - PostgreSQL開発に時間を割く技術者を提供している企業がいくつかある／その企業群も少しずつ変遷している

PostgreSQL開発体制



開発コミュニティ

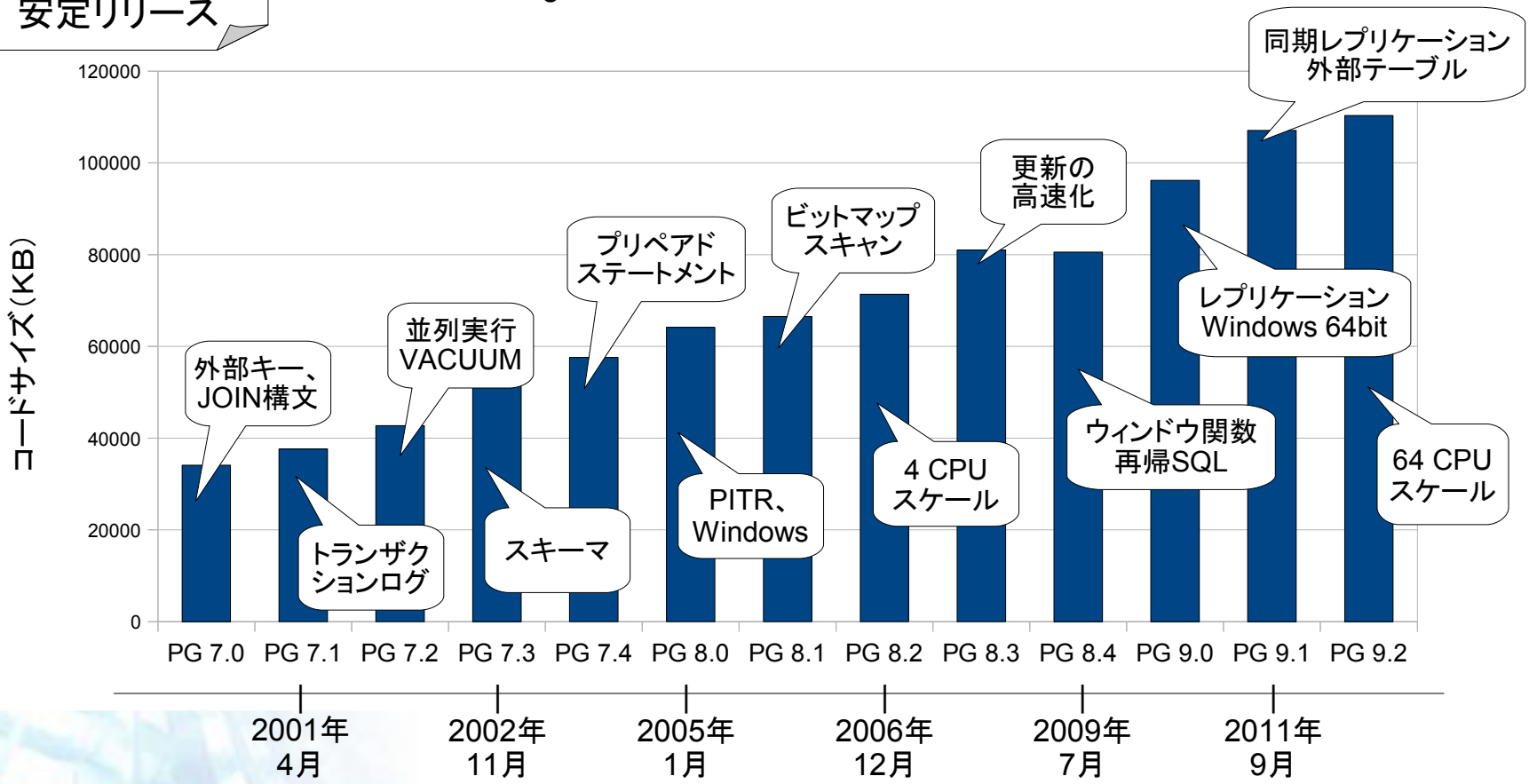
支援企業

NTT
NEC
EMC² where information lives
SRA OSS, INC.
salesforce
PGX POSTGRESQL EXPERTS, INC.
Redpill Linpro
meetme where new friends meet
EnterpriseDB
etc...

※www.postgresql.org 記載より

PostgreSQLの歩み

1年1バージョン
10年以上
安定リリース



PostgreSQLはどこで使われているか？

- 業務基幹システムの商用データベース製品置き換え
  - EnterpriseDB「Postgres Plus Advanced Server」など、PostgreSQLベースの商用DB互換製品も
- Blog、SNS、ゲーム、各種の新しいオンラインサービス
  - PostgreSQL も使われているが、MySQL も強い
    - オープンソースWebアプリケーションの標準データベース
    - 廉価クラウドで標準提供される DBMS
- BI分野の独自データベース製品のベースとして
  - Netezza、GreenPlum、Yahoo自社内むけデータベース
- 地理情報システムで大きな存在感
  - PostGIS というオープンソースの追加モジュールが強力

PostgreSQL 9.3 のご紹介

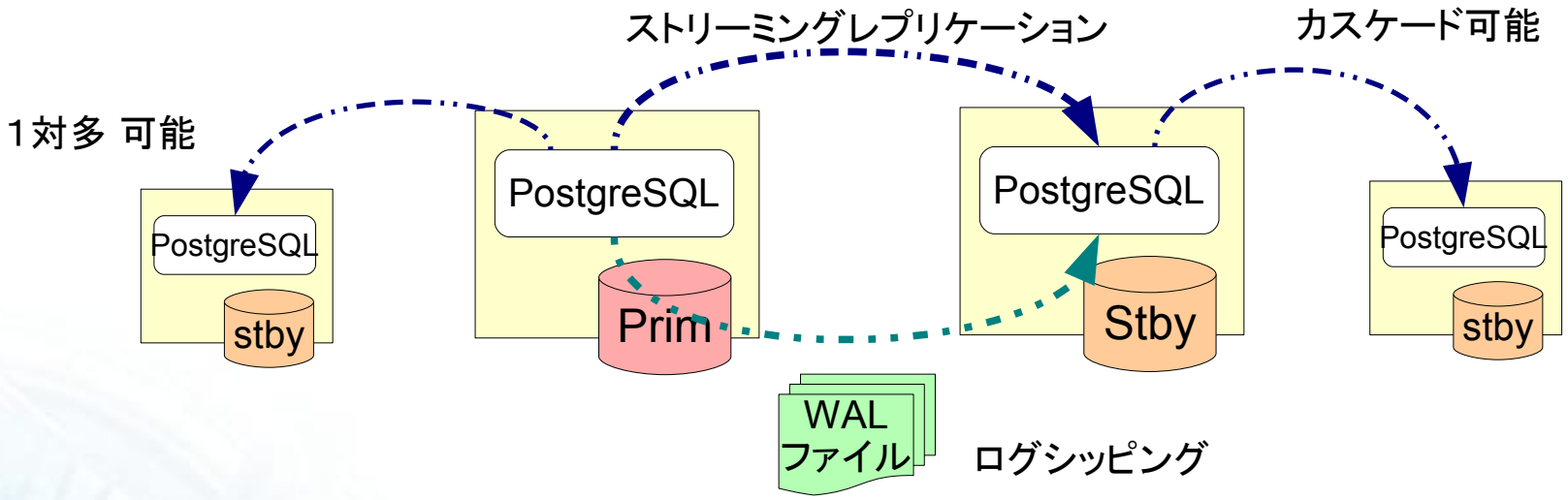
PostgreSQL 9.3 は盛りだくさん

マテリアライズドビュー
pg_isready
DDLトリガ
LATERAL結合
外部キー制約
postgres_fdwと
書き込みFDW
ロック競合軽減
並列pg_dump
JSON関数
ストリーミングレプリケーション
worker_spi
暗黙の
タイムライン追随
更新可能ビュー
再帰ビュー
LOBサイズ拡張
ページ
pg_xlogdump
チェックサム
COPY FREEZE
ロックタイムアウト

9.2 開発での機能候補がシフトした

ストリーミングレプリケーション

タイムライン追随



- PostgreSQL のレプリケーションはどんなもの？
- タイムライン（時系列）とは？
  - リカバリするごとに +1 されるメタ情報

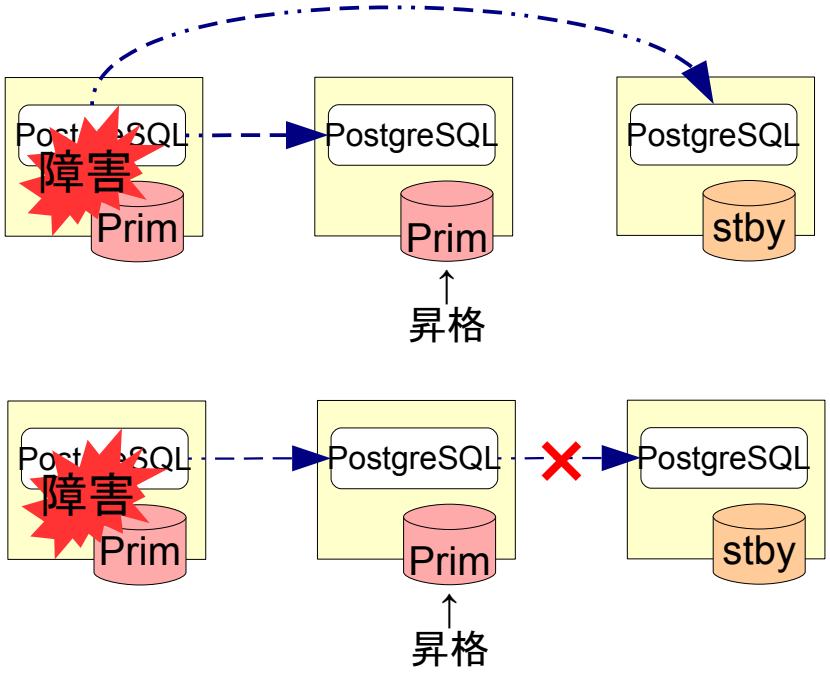
ストリーミングレプリケーション

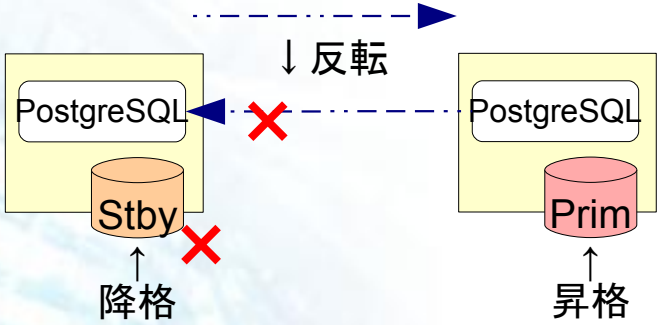
タイムライン追随

従来は
ログシッピングを使うか、
ベースバックアップ再取得

- これまでのストリーミングレプリケーションで、できそうでできなかったことが可能に

（1）昇格ノードに参照元を切り替え
（2）カスケードで昇格後に継続処理
（3）ノードのスイッチオーバー



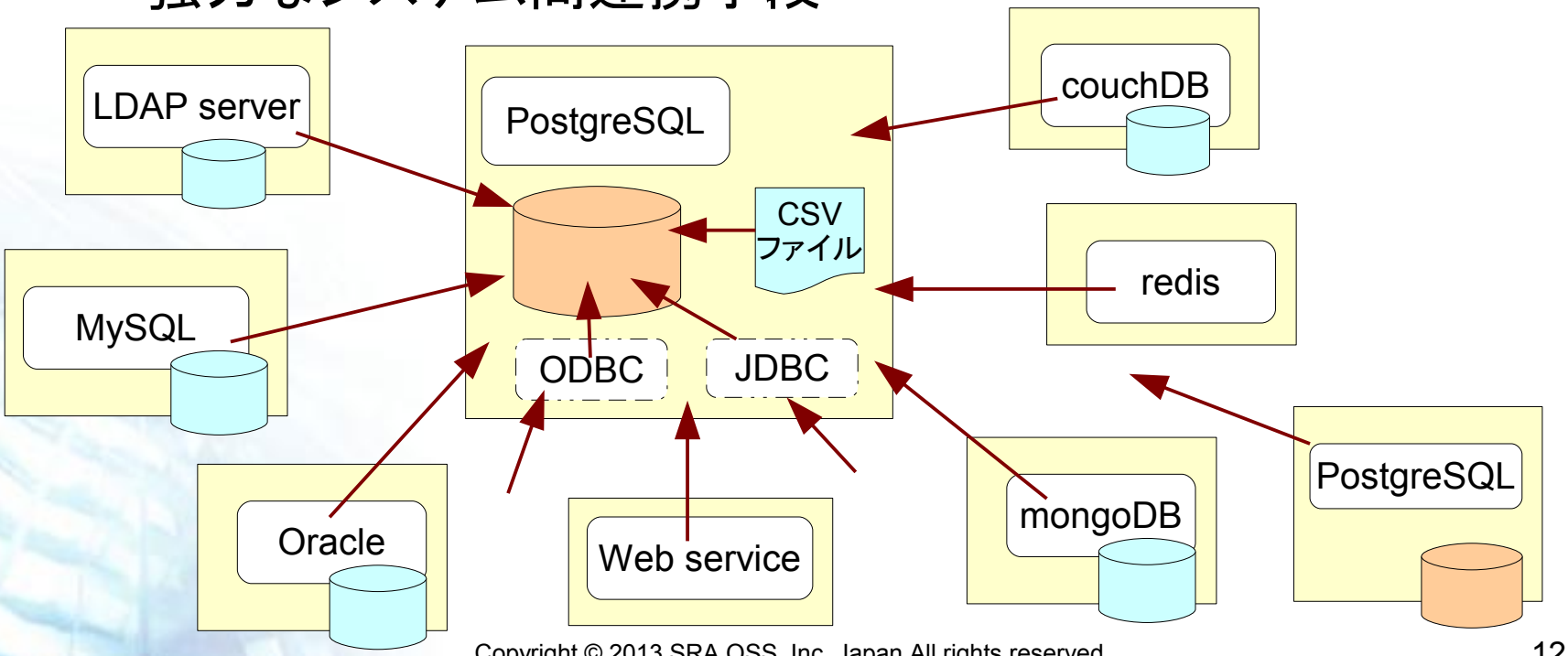


他にも改善項目：
「昇格の高速化」
「タイムアウト設定拡充」

postgres_fdw と書き込み FDW

9.3 から
書き込みに対応
postgres_fdw が本体付属

- FDW = foreign data wrapper（外部データラッパ）
  - 外部にあるデータをテーブルのようにアクセスする枠組み
  - 強力なシステム間連携手段



postgres_fdw と書き込み FDW

- サーバ、 ユーザマッピング、 外部テーブル、のモデル
- 外部テーブルは通常テーブルと同様に読み書き可能
- カスケード可能
- COMMIT、ROLLBACK が可能

振る舞いは
各FDW の実装次第

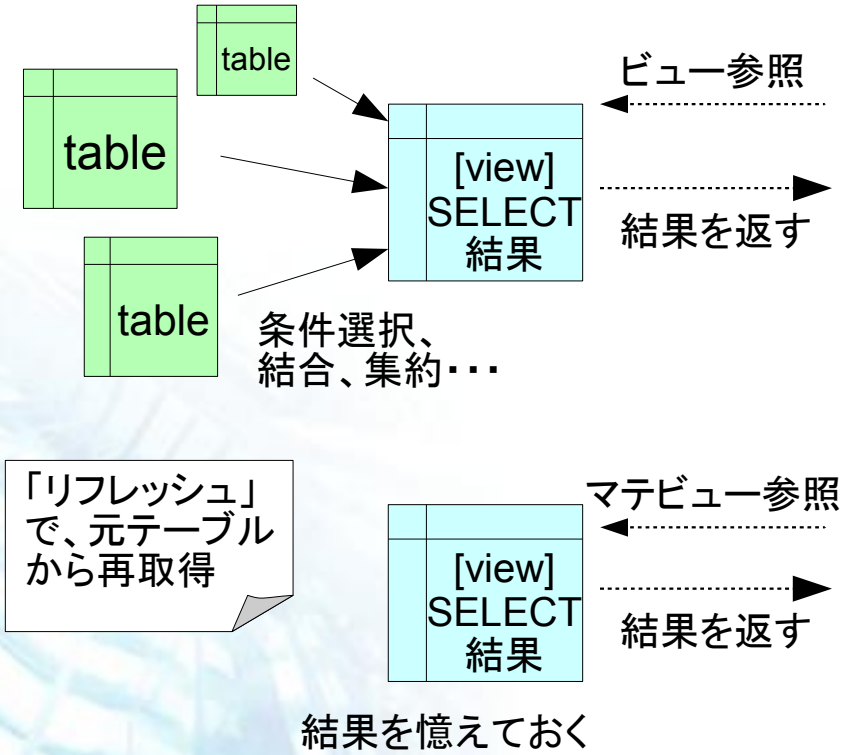
(postgres_fdw の例)

```
db1=# CREATE SERVER db02srv FOREIGN DATA WRAPPER postgres_fdw
      OPTIONS (port '5432', host 'dbhost02', dbname 'db02');

db1=# CREATE USER MAPPING FOR user1 SERVER db02srv
      OPTIONS (user 'user1', password 'secret');

db1=# CREATE FOREIGN TABLE remote_t1 (id int, v text)
      SERVER db02srv
      OPTIONS (schema_name 'public', table_name 't1');
```

- 別バージョン対応
- リモートで条件絞り込み
- トランザクション処理に制限

マテリアライズドビュー

- 結果を保持するビュー



- 基本機能のみ提供

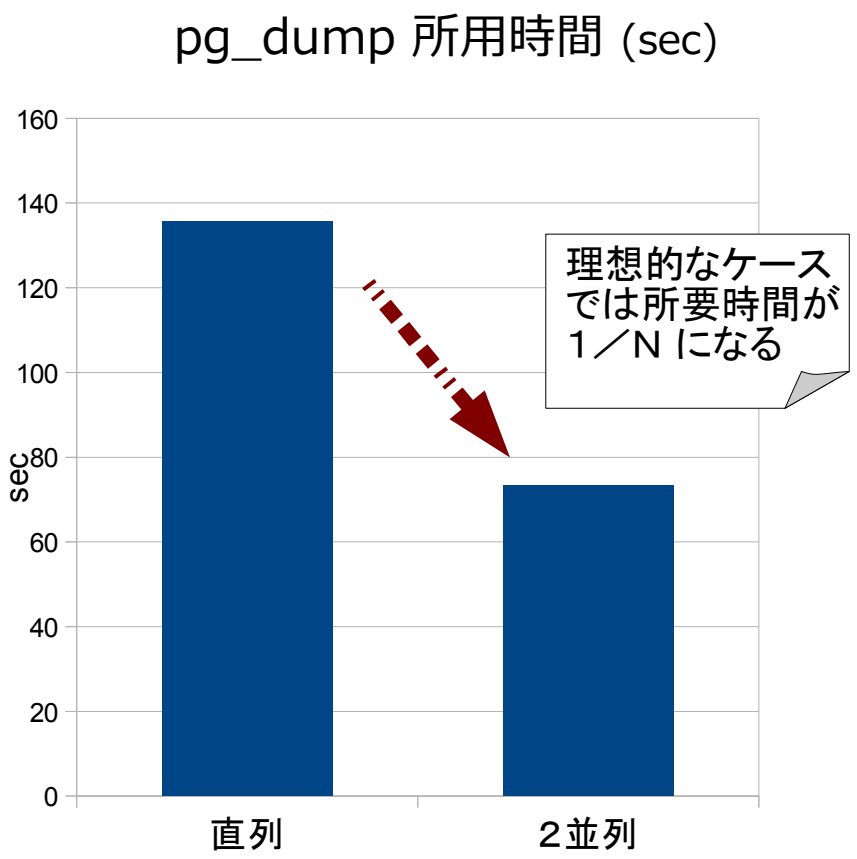
  提供されない機能：

  - × 差分更新
  - × 自動リフレッシュ
  - × プランナでのクエリリライト

```
db1=# CREATE MATERIALIZED VIEW
      mv_abalance AS
        SELECT aid, abalance
          FROM pgbench_accounts
          ORDER BY abalance
          LIMIT 10;
db1=# REFRESH MATERIALIZED VIEW
      mv_abalance;
```

並列pg_dump

- ダンプを並列実行する
  - 以下のとき高速化できる
    - 複数テーブルがある
    - 複数CPUコアがある
    - ディスクI/Oに余裕がある
  - -j で並列数を指定
  - 並列でも単一スナップショットが保証される
  - ディレクトリ形式
  - スタンバイでは不可



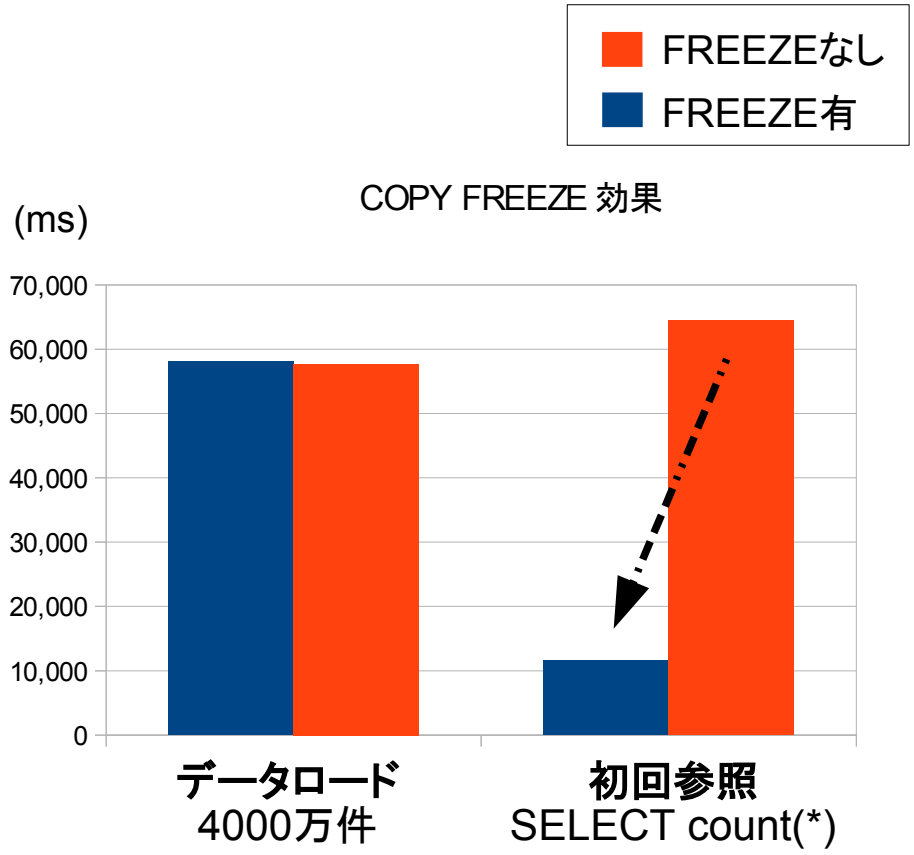
2テーブルのデータベースを2CPUマシンでダンプ

直列: pg_dump -j 1 -Fd -f out.d db1

2並列: pg_dump -j 2 -Fd -f out.d db1

COPY FREEZE

- 高速データローディングを実現するオプション
  - COPY後の初回参照時の処理が不要になる
  - トランザクション隔離動作としては例外的な振る舞いになる
    - COMMIT前に見えてしまう




```
db1=# BEGIN;
db1=# TRUNCATE t_huge;
db1=# COPY t_large FROM '/tmp/t_huge.copy' FREEZE;
db1=# COMMIT;
```

外部キー制約

ロック競合軽減

- 競合の少ない行ロックが追加された
  - SELECT … FROM … FOR KEY SHARE
    - 主キー以外のカラムに対する UPDATE とロック競合しない
    - 外部キー制約で使われる
  - SELECT … FROM … FOR NO KEY UPDATE
    - 「FOR KEY SHARE」と競合しない「FOR UPDATE」

ロックタイムアウト

- タイムアウト設定が可能に
  - lock_timeout 設定
  - 従来は、WAIT か NOWAIT の二択

（例：t に対するロック待ちの場合）

```
db1=# BEGIN;
db1=# SET LOCAL lock_timeout
       TO '10s';
db1=# UPDATE t SET c = 'xx'
       WHERE id = 12345;
ERROR:  canceling statement
due to lock timeout
```

暗黙の
更新可能ビュー

- シンプルなビューは、作っただけで更新できる

シンプルなビューとは？：

```
SELECT ≪カラムリスト≫
 FROM ≪単一テーブル≫
 WHERE ≪検索条件≫
 ORDER BY ≪ソートカラム≫
```

- INSERT、UPDATE、DELETE は参照元テーブルに適用される
- 従来は手動でRULE や TRIGGER を定義して上げる必要があった

再帰ビュー

- ビューで再帰SQLを記述

```
CREATE RECURSIVE VIEW v (n)
AS
 SELECT 1
 UNION ALL
 SELECT n+1 FROM v WHERE n < 3;
```

```
db=# SELECT * FROM v;
 n
---
 1
 2
 3
(3 rows)
```

WITH RECURSIVE 構文と同様のことが記述できる

DDLトリガ (EVENT TRIGGER)

- CREATE、DROP、ALTERにトリガ設定できる
  - DDL操作の一部制限に
  - 連動した自動操作に
    - テーブル単位レプリケーションツールでの応用が有力

pg_isready

- PostgreSQL死活確認をするクライアントツール
  - 「接続できるか？」を確認
  - 実際には接続しない
    - ホスト、ポートを指定
    - ユーザ名、パスワード不要
    - 最大接続数が埋まっていても大丈夫
    - プロセスが生きているだけではNG扱い

ページ

チェックサム

- テーブル、インデックスデータのバイナリ破損を検知できる
  - 従来は、ヘッダ部分の破損しか検知できなかった

(エラーメッセージ例)

```
WARNING: page verification failed, calculated checksum 61554 but expected 3960
ERROR:  invalid page in block 23 of relation base/12896/16466
```

LOBサイズ拡張

- ラージオブジェクトの上限2GB を拡張
  - 従来は、(圧縮後)2GBを超えるとおかしな動作

↓

  - 最大 4TB に拡張

弊社(SRA OSS,Inc.)による開発項目

LATERAL結合

- ISO SQL:1999 にあるLATERAL構文に対応

```
SELECT * FROM t1,
 LATERAL
  (SELECT * FROM t2
   WHERE c2 = t1.c2) v2,
 LATERAL
  (SELECT * FROM t3
   WHERE c3 = v2.c3) v3;
```

  - FROM句内のサブクエリで左側にあらわれた要素を参照できる

JSON関数 追加

- JSON型データむけの組み込み関数が追加
  - 行データをJSON型として出力する関数のみ

↓

  - JSON配列の各種の操作
  - 表への逆変換
  - 要素取り出し

多機能な1カラムデータとして応用できる

pg_xlogdump

読み解くには、
それなりに技術が
必要となる

- WALファイル内容を参照するツールが標準付属

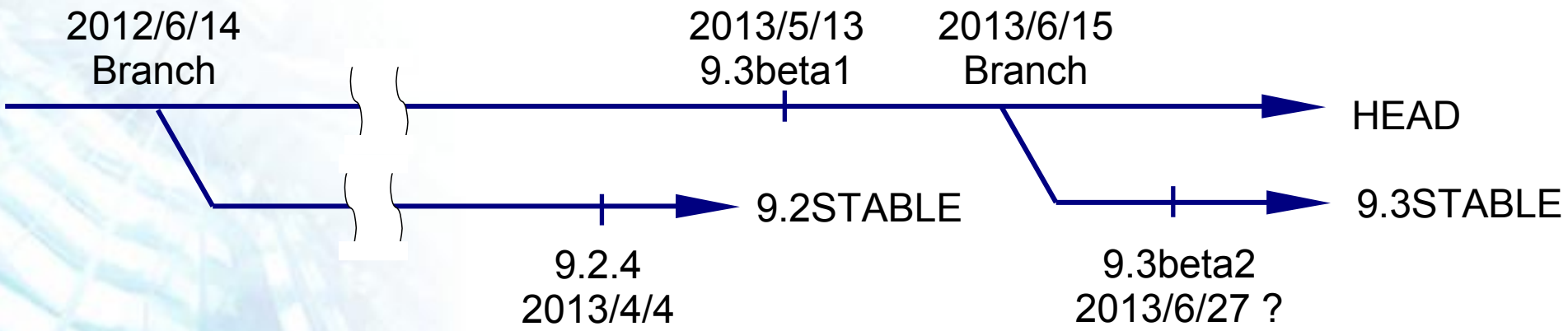
  用途としては：

  - WALファイルが破損していないかを確認
  - 新たな更新処理が発生したかどうかをデータベース接続することなく確認
  - トランザクションIDを使った、PITRリカバリ位置決め

```
$ pg_xlogdump \
  000000130000000200000F5
rmgr: Heap        len (rec/tot):
   21/   237, tx:      13238,
lsn: 2/F502A4A8, prev 2/F502A440,
bkp: 1000, desc: insert: rel
1663/12896/16573; tid 0/1

rmgr: Btree       len (rec/tot):
   18/   174, tx:      13238,
lsn: 2/F502A598, prev 2/F502A4A8,
bkp: 1000, desc: insert: rel
1663/12896/16579; tid 1/3

rmgr: Transaction len (rec/tot):
   12/    44, tx:      13238,
lsn: 2/F502A648, prev 2/F502A598,
bkp: 0000, desc: commit: 2013-06-
06 17:46:32.281513 JST
  :
（後略）
```

PostgreSQL 9.3 リリース

- beta1 時点で 9.3 の大きな機能追加は完結
  - 6/15 に 9.3 系 STABLEブランチ
  - まもなく 9.3beta2 がリリース予定
- 2013年秋にリリースされる見通し（例年通りなら･･･）



まとめ

- PostgreSQL はオープンソースソフトウェアとして、安定した開発体制を維持しています
- PostgreSQL 9.3 には、多数の機能拡張が含まれています

ご清聴ありがとうございました。
ご質問を承ります。