SRA OSS, INC.

PostgreSQL Update
9.1リリース&9.2の展望

INSIGHT OUT 2011

2011-10-19 15:00～15:50

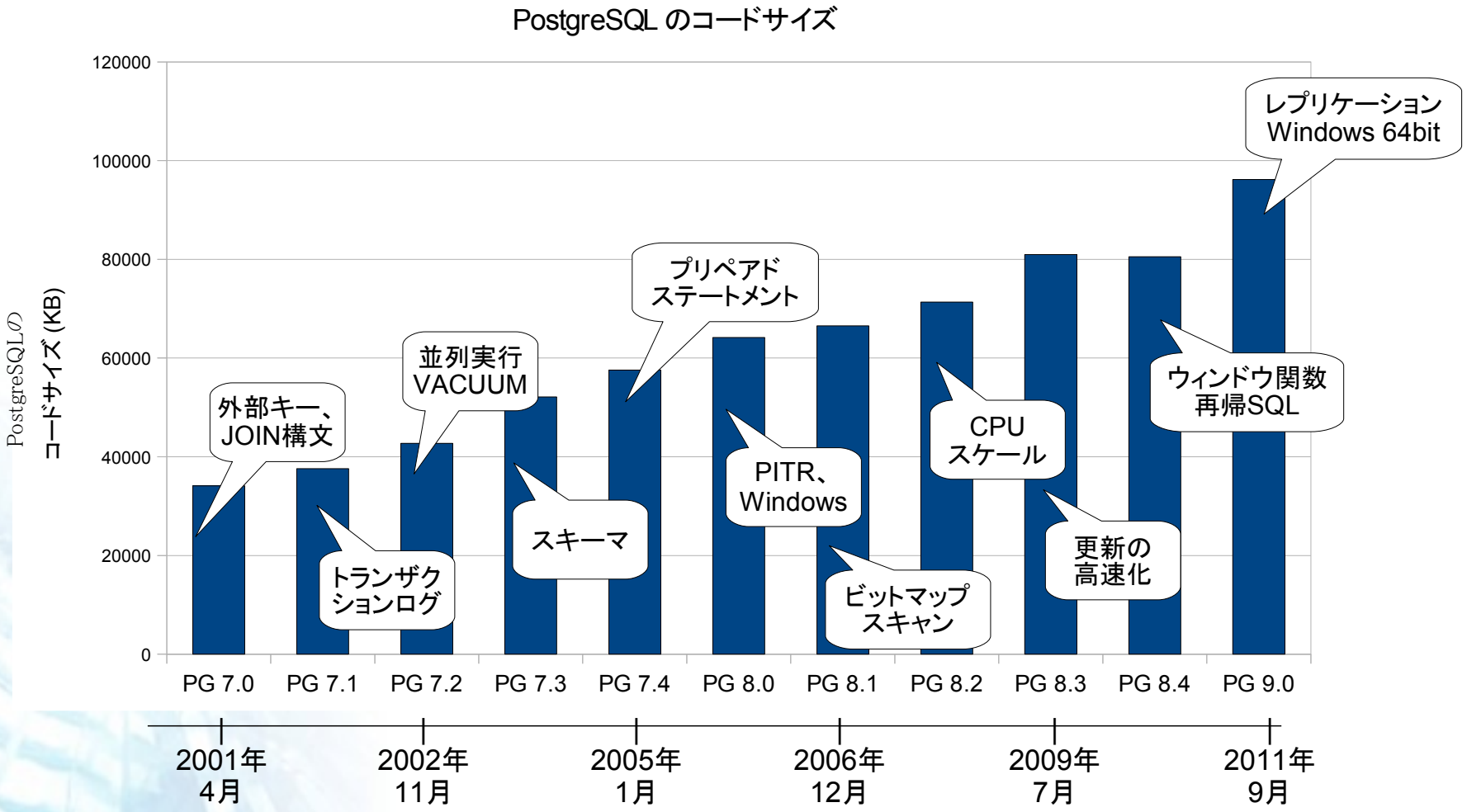
SRA OSS, Inc. 日本支社

高塚 遥 harukat@sraoss.co.jp

PostgreSQLとは何ぞや

- 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 10年以上の歴史
- BSDタイプのライセンスで配布
  - PostgreSQL Global Development Gruop と UCB 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の歩み



位置づけと使われ方の変貌

2001年

- おもちゃ、社内システム
- Accessからの移行先
- Webバックエンド
- 「小規模なら」公共・エンタープライズ
- 「大規模」「ミッションクリティカル」でも頑張れば公共・エンタープライズ
- パッケージ製品内部の基盤部分
- Oracleからの移行先
- OSS Webアプリの基盤部分

2011年

携帯DLサイト

自治体パッケージ

ゲームSNS

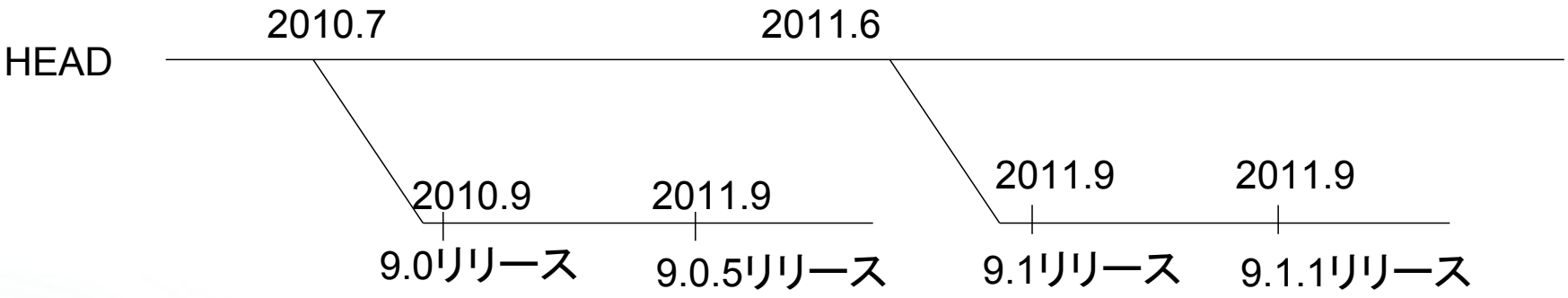
マンモス校教務

地図/ナビ

オンライン証券

銀行基幹

PostgreSQL 9.1 リリース

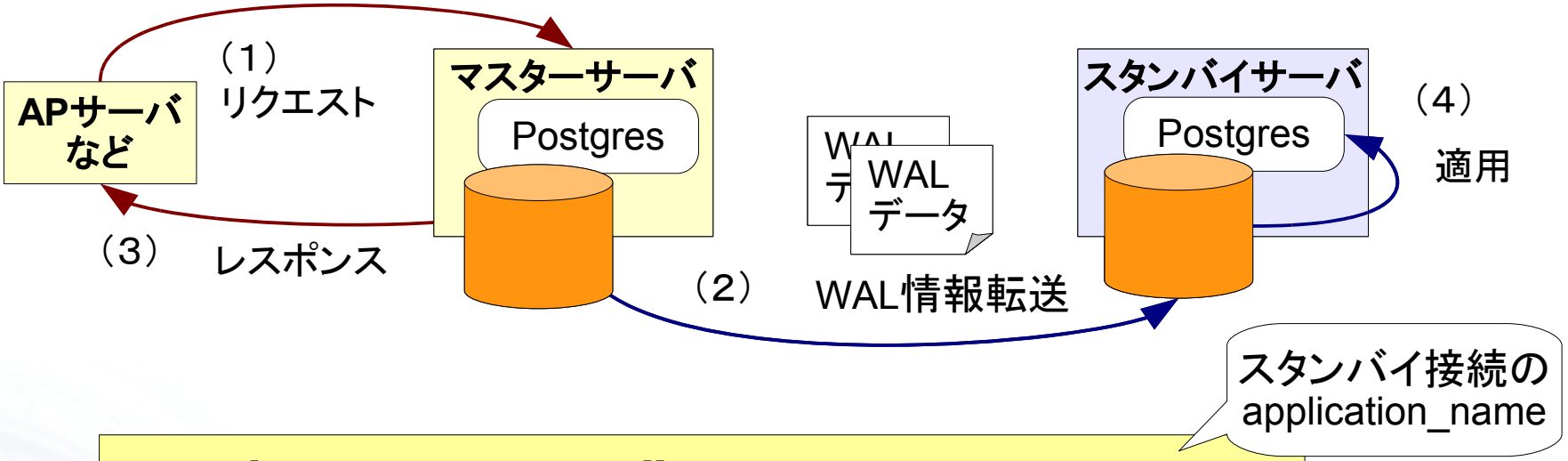


例によって1年毎メジャーバージョンリリース

- 9.0に入りそびれた大物機能
- 運用、開発を便利にする機能

レプリケーション関連

同期レプリケーションが選択可能

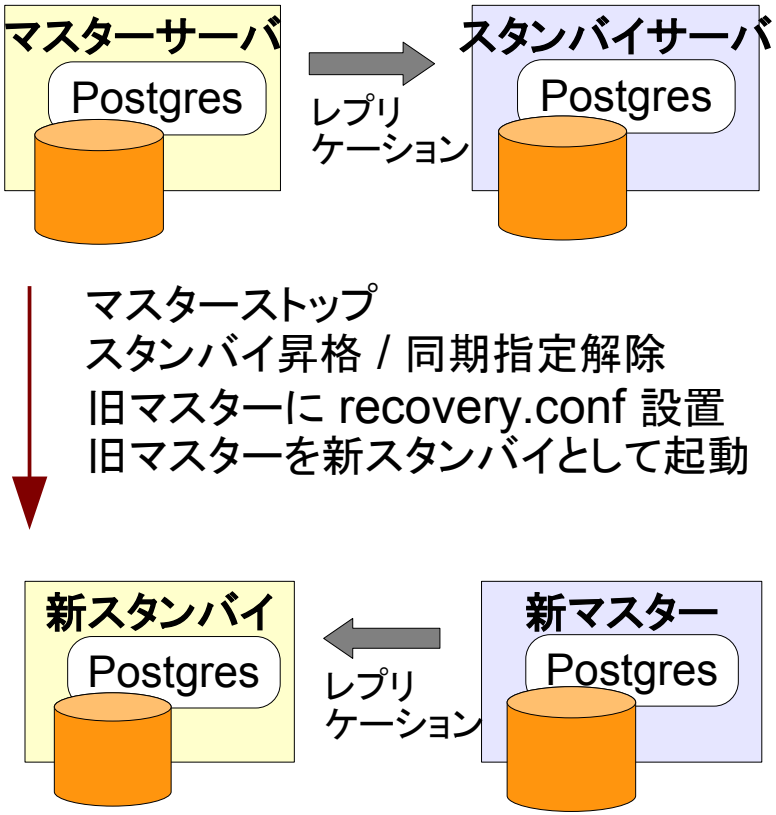



```
synchronous_standby_names = 'syncrep'
```

- 9.0 では非同期のみ（＝障害直近データ損失あり）
- WALデータを転送完了後にコミット完了とできる
- 「スタンバイに問い合わせた内容がマスタと一致」を保障するわけではない
- 逆にスタンバイが止まっているとマスタがブロック

再全同期なしのスイッチ可能

- 9.0はスイッチ困難
  - ベースバックアップの取り直しが必要になることが多い
- 正常停止した旧マスタのデータを利用可能
- PaceMaker の Master-Slave型リソースにも適用可能
  - 試作RA有



recovery.conf

```
recovery_target_timeline = leastest
```

pg_stat_replicationビュー

- マスターでレプリケーション状況を一覧できる

```
=# SELECT * FROM pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
procpid          | 2604
usesysid         | 10
username         | postgres
application_name | syncrep
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 56573
backend_start    | 2011-10-12 13:13:12.121657+09
state            | streaming
sent_location    | 0/12C5E9B4
write_location   | 0/12C5E9B4
flush_location   | 0/12C5E7D0
replay_location  | 0/12C5E7D0
sync_priority    | 1
sync_state       | sync
```

ただし進度の単位は
WALファイル位置

pg_basebackupコマンド
pg_ctl promoteコマンド

- ベースバックアップがコマンドひとつで完了
  - SELECT pg_start_backup('hoge'); ⇒ tarとか ⇒ SELECT pg_stop_backup(); の手順が不要に
  - テーブルスペースにも対応
  - PostgreSQLの通常接続を使うので scp や NFS を用意しなくてよい
- スタンバイのレプリケーションを終了させて通常起動(＝マスタ昇格)させるコマンド
  - これまではトリガーファイルを置く方式のみ

拡張

SQL/MED FOREIGN TABLE

- 外部データをテーブルとして読み込む枠組み
  - contrib/file_fdw はCSV形式などのファイルをテーブルとして読み込む外部データラッパーの実装例
  - さまざまな外部データ連携のモジュールが作れる
- 9.1 では参照だけ、更新は未対応
- mysql_fdw や twitter_fdw などいろいろ開発されている

CREATE EXTENSION

- 拡張モジュールをインストールする命令
- 所定の作法を守って書かれた拡張モジュールを、SQLコマンドで管理できる
  - ビルドと.dll / .soのインストールは予め必要です
- 拡張モジュールのバージョンアップに対応

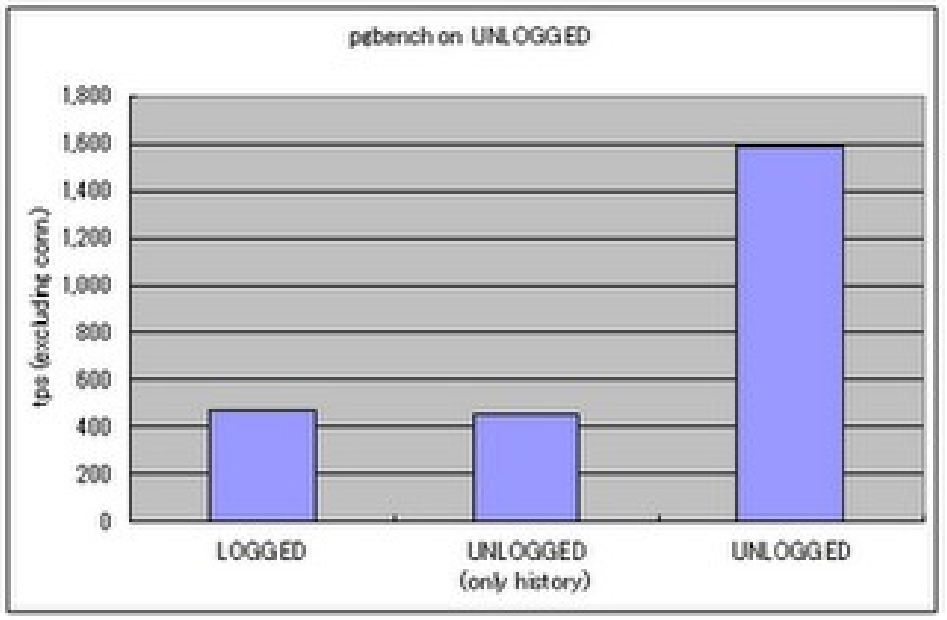
```
=# CREATE EXTENTION hstore;
=# DROP EXTENSION hstore;
```

性能

UNLOGGEDテーブル

- トランザクションログ（WAL）に記録しないテーブルを作れる
  - 書き込みが速い
  - クラッシュしたらデータ消えてしまうかも
  - レプリケーションされない

```
=# CREATE UNLOGGED TABLE
 tbl01 ( … )
```



pgsnaga.blogspot.comより

ORDER BY pushdown

- プランナが改善
- Merge Append プランタイプ
- テーブルパーティショニングされたテーブルでソートを行うとき各々の子テーブルにあるインデックスが使われる

```
=> SELECT id, v1 FROM t_partitioned_oya
     ORDER BY v1 LIMIT 10;
```

賢いCLUSTERコマンド

- テーブル統計情報をみて、最適な再編成の方式を自動的に選択
  - 新方式の「Seq Scan + Sort」は、
    - 速い、ただし、ほぼクラスタ済みデータなら旧方式有利
    - 一時ファイルを大きく使う

テーブル状態	処理方式
断片化が少ない	Index Full Scan（従来方式）
断片化が多い	Seq Scan + Sort（新方式）

性能…大事な話

- PCサーバ上、単純なpgbench 標準テスト、デフォルト設定では PostgreSQL 9.1 は速くない
  - そもそも PostgreSQL 8.4 ～9.0 も速くない
  - 単純テストはPostgreSQL 8.3 が速い
- 「状況がハマるとき」「しかるべく設定したとき」初めて性能アップの成果が得られる

機能

KNN GiST インデックス

- k-NN(k Nearest Neighbor) 検索：空間上に指定された地点に近接するオブジェクトを，空間データベースの中からk 個求める
- GiSTインデックスを距離が近いものトップリストを出す検索に使える
  - POINT型、Geometry型など

```
SELECT coordinates,
 (coordinates <-> '5.0,5.0'::point) AS dist
FROM spots ORDER BY dist ASC LIMIT 10;
```

述語ロック対応SERIALIZABLE

- Serializable Snapshot Isolation
- 9.0 までのSERIALIZABLEは、不完全
  - マニュアルにも但し書きがついている

順に片方ずつ行った場合

以下を 旧SERIALIZABLE で同時実行すると直列実行と異なる結果になる／9.1 SERIALIZABLEでは 直列化失敗エラー

```
INSERT INTO tbl (val, grp)
  SELECT SUM(val), 'b' FROM tbl WHERE grp = 'a';
```

```
INSERT INTO tbl (val, grp)
  SELECT SUM(val), 'a' FROM tbl WHERE grp = 'b';
```

旧SERIALIZABLE ⇒ (9.1) REPEATABLE READ

新SERIALIZABLE ⇒ (9.1) SERIALIZABLE

互換性注意

contrib/sepgsql

- データベースオブジェクトに強制アクセス制御
  - アクセス可否のチェックをSE-Linuxと統合
    - 各SQLについてSE Linux に内部的に問い合わせるので SELinux Enabled が大前提
    - SELinuxのポリシーとして設定を行う
  - SE-Linux同様のラベルベース強制アクセス制御

```
SECURITY LABEL FOR selinux
 ON TABLE mytable
 IS 'system_u:object_r:sepgsql_table_t:s0';
```

さまざまな文字列関数が追加

- format('%s and %s', 'abc', '123')
  - printf タイプのフォーマッタ
  - SQLリテラル、SQL識別子のエスケープに対応
- concat(a, b, c, ...)、concat_ws()
  - || 演算子と同じだけど、某DBとの互換性的に
- reverse()
- left()、right()
  - 右から何文字、左から何文字

更新を行えるWITH句

- Writable CTE (Common Table Expressions)
  - 処理行データ返す「RETURNING 」と組み合わせ、WITHクエリでINSERT、UPDATE、DELETE 記述可
  - 例：DELETEした内容を別テーブルにINSERT

```
WITH t_tmp (id, v) AS (DELETE FROM t RETURNING *)
  INSERT INTO t_new SELECT * FROM t_tmp;
```

  - 例：MERGE代替(UPDATE で該当なければINSERT)

```
WITH val AS (SELECT 100 as id, 'AAA' as v),
     upd AS (UPDATE t SET v = val.v FROM val
              WHERE t.id = val.id RETURNING t.id)
INSERT INTO t
   SELECT * FROM val WHERE id NOT IN (SELECT id FROM upd);
```

ビューに対するトリガー

- 更新可能ビューを作るのに RULE でなくトリガーも使えるようになった
  - RULEは PostgreSQL独自概念だし、書き方も独特
  - トリガーの方が複雑なことも書ける

```
CREATE TRIGGER mytrig
   INSTEAD OF UPDATE ON myview
   FOR EACH ROW
   EXECUTE PROCEDURE myupdate();
```

カラム単位ロケール指定

- 検索毎、カラム毎で COLLATE（言語を考慮した文字比較）設定ができる
  - 8.3 まで　データベースクラスタ単位
  - 8.4 データベース単位
- glibc の日本語ロケールは例によって使えない

```
CREATE COLLATION c_en (LOCALE = 'en_US.UTF-8');

SELECT t FROM t_coll ORDER BY t COLLATE c_en;
SELECT t FROM t_coll ORDER BY t;
```

```
ABC
XYZ
abc
xyz
```

```
abc
ABC
xyz
XYZ
```

contrib/pg_tramの拡張

- LIKE検索で中間一致にもインデックス検索
  - もともとは類似度一致の演算子やインデックス機能を提供するもの
  - 9.1 で LIKE、ILIKE に対応
  - 実はマルチバイト対応にはソース上のフラグ変更必要

gist も可

```
CREATE INDEX ON docs
  USING gin (doc gin_trgm_ops);

SELECT * FROM docs WHERE doc LIKE '%foo%';
```

9.1 その他・・・

- 非互換！
  - standard_conforming_strings = on がデフォルト
    エスケープ利用で E'xxx' を使っていないなら要注意
  - 複合型に対する、関数スタイルのキャストが禁止
  - PL/pgSQL の RAISE命令と例外処理の振る舞い変更
    - 例外補足のされ方が（Oracle PL/SQL に似せるように）変わ
  - SERIALIZABLEの定義変更
  - その他、細かな非互換はリリースノートを参照
- 修正点は以上紹介したものの他にも多数

9.2 以降

レンジデータ型

- 範囲を表現するデータ型
  - 重なり検出する演算子 &&
  - 8.4 で導入された排他制約と 組み合わせ
    - 重なりがあったら制約違反
    - これまでは BOX型くらい　しか使える対象がなかった

データ型	要素データ型
int4range	int
int8range	bigint
numrange	numeric
tsrange	timestamp without timezone
tstzrange	timestamp with timezone
daterange	date

```
SELECT range(11.1, 22.2) && range(20.0, 30.0);

ALTER TABLE reservation
  ADD EXCLUDE USING gist (during WITH &&);
```

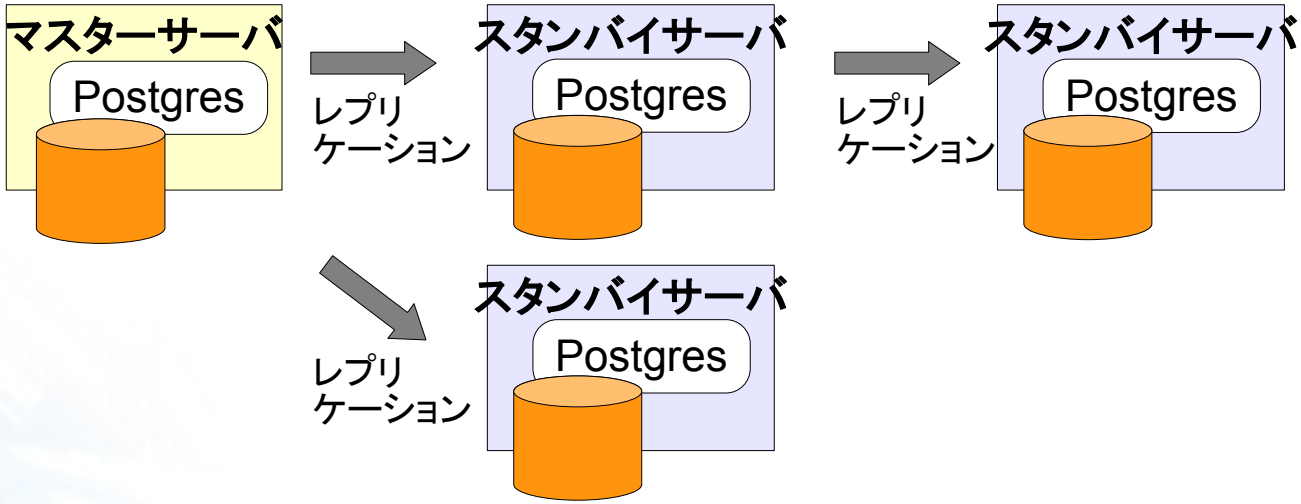
Index Only Scan

- min/max、Sort/Limit、Hash-Join、Merge-Join などで使える

```
=# EXPLAIN select min(unique1) from tenk1;

                              QUERY PLAN
-----------------------------------------------------------------
 Result
   InitPlan 1 (returns $0)
     ->  Limit
       ->  Index Only Scan using tenk1_unique1 on tenk1
             Index Cond: (unique1 IS NOT NULL)
```

レプリケーションのカスケード対応



- カスケード構成の障害となっている細かな問題を修正
- ホットスタンバイサーバからのベースバックアップ取得対応

commitfest 2011-09 より

- Inserting heap tuples in bulk in COPY
- バックグラウンドジョブの改善
  - Separating bgwriter and checkpointer
  - Autovacuum polling loop elimination
- pg_last_xact_insert_timestamp()
  - トランザクション状態を時刻で知りたい
- recovery.conf が postgresql.confに統合
- ロック～排他制御まわりの改善多数

未確定のものも含めて紹介しています。

他にも多数挙がっています。

新バージョン情報は

- コミットフェスト管理Webサイト
  https://commitfest.postgresql.org/
- アルファ版が数ヶ月おきに
  http://www.postgresql.org/developer/alpha
- HEADのマニュアルとか
- 各種イベントの講演など