Home

Synchronous Replication

Replication Tools

Per-Column Collations

Other Features

Extensions

SE-Postgres

K-Nearest Neighbor

wCTEs

Unlogged Tables

SQL/MED

Other Features

Extensions

SE-Postgres

K-Nearest Neighbor

wCTEs

Unlogged Tables
Synchronous Replication
Synchronous Replication
Synchronous Replication

standby Paul:
primary_conninfo = 'host=john
    port=5432 user=replication
  application_name=paul'

master John:
synchronous_commit = on
synchronous_standby_names = 'paul,george,ringo'
Synchronous Replication

postgres=# select application_name as appname, state, sync_state, sync_priority as prty, replay_location from pg_stat_replication;

<table>
<thead>
<tr>
<th>appname</th>
<th>state</th>
<th>sync_state</th>
<th>prty</th>
<th>replay_location</th>
</tr>
</thead>
<tbody>
<tr>
<td>paul</td>
<td>STREAMING</td>
<td>SYNC</td>
<td>1</td>
<td>0/29154EB8</td>
</tr>
<tr>
<td>george</td>
<td>STREAMING</td>
<td>POTENTIAL</td>
<td>2</td>
<td>0/291452F0</td>
</tr>
<tr>
<td>ringo</td>
<td>STREAMING</td>
<td>POTENTIAL</td>
<td>3</td>
<td>0/29154EB8</td>
</tr>
</tbody>
</table>
# set synchronous_commit = on;
# begin;

# update user_balance
# set balance = balance - 300
# where user = 14301;

# update user_balance
# set balance = balance + 300
# where user = 13221;

# commit;
# set synchronous_commit = 'local';
# insert into user_messages
# values ( 14301, 13221,
# 'hello a/s/l?';
New Replication Tools

- `pg_stat_replication` view
  - shows all replicas and their statuses
- `pg_basebackup`
  - single-command cloning over port 5432
- re-mastering
  - promote a replica to be the new master
- new functions
  - `pg_last_xact_replay_timestamp()`
  - `pg_xlog_replay_pause()`
<table>
<thead>
<tr>
<th>ID</th>
<th>English</th>
<th>Japanese</th>
</tr>
</thead>
<tbody>
<tr>
<td>787</td>
<td>error</td>
<td>エラー</td>
</tr>
<tr>
<td>12</td>
<td>unique</td>
<td>ユニークな</td>
</tr>
<tr>
<td>456</td>
<td>index</td>
<td>インデックス</td>
</tr>
<tr>
<td>333</td>
<td>documentation</td>
<td>ドキュメンテーション</td>
</tr>
<tr>
<td>217</td>
<td>open source</td>
<td>オープンソース</td>
</tr>
<tr>
<td>2134</td>
<td>penguin</td>
<td>ペンギン</td>
</tr>
</tbody>
</table>
$ create table collated_polish (some_text text collate "pl_PL.utf8");
$ copy collated_polish from '/tmp/polish';

$ select * from collated_polish order by some_text;
some_text
-----------
alfa
car
coś
ćma
lama
łódka
Psycadelic Queries
WITH deleted_posts AS (  
    DELETE FROM posts  
    WHERE created < now()  
    - '6 months'::INTERVAL  
    RETURNING *  
  )  
SELECT user_id, count(*) FROM deleted_posts group BY 1;
WITH deleted_posts AS (  
    DELETE FROM posts  
    WHERE created < now()  
        - '6 months'::INTERVAL  
    RETURNING *  
),  
deleted_per_user as (  
    SELECT user_id, count(*)  
    FROM deleted_posts  
    GROUP BY 1  
)  
UPDATE counts  
SET posts_count = posts_count - d.count  
FROM deleted_per_user d  
WHERE d.user_id = counts.user_id;
Serialized Snapshot Isolation

- Allows true serialization of concurrent transactions
- No more explicit locking!
- No more deadlocks!
- As long as each transaction works, all will work.
- Including limited predicate locking.
update othello set pieces = 'white' where pieces = 'black';

commit;

ERROR: could not serialize access due to read/write dependencies among transactions
CREATE UNLOGGED TABLE cleaned_log_import
AS SELECT hit_time, page
FROM raw_hits, hit_watermark
WHERE hit_time > last_watermark
   AND is_valid(page);
SE-Postgres

- Now a contrib module!
  - uses “hooks” enabled at build time
- Integrates with SE-Linux
  - SE-Linux “security labels” become object permissions
  - covers all database object permissions
- row-level access control not implemented yet
SE-Postgres
K-Nearest Neighbor

SELECT *,
    position <-> $my_current_location
as distance
FROM ben_n_jerry
ORDER BY position <-> $my_current_location
LIMIT 3;
CREATE FOREIGN TABLE raw_hits
  ( hit_time TIMESTAMP,
    page TEXT )
SERVER file_fdw
OPTIONS (format 'csv', delimiter ';;', filename '/var/log/hits.log');
SELECT page, count(*)
FROM raw_hits
WHERE hit_time > '2011-04-16 16:00:00'
AND hit_time <= '2011-04-16 17:00:00'
GROUP BY page;
Extensions
CREATE EXTENSION IF NOT EXISTS citext
WITH SCHEMA ext;

\dx

<table>
<thead>
<tr>
<th>Schema</th>
<th>Name</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ext</td>
<td>citext</td>
<td>9.1devel</td>
<td>case-insensitive character string type</td>
</tr>
</tbody>
</table>
Extensions

ALTER EXTENSION citext UPDATE TO 9.2;

DROP EXTENSION citext;
Extensions

analyze api arithmetic data types datatype dictionary explain explain analyze fdw france full-text search gmp hash integer internet ispell italian key value key value pair md5 node ordered pair pair plan rational semantic version semver sha sha1 statistics table tap tdd test driven database development testing tinyint twitter unit testing variadic function version version
Other Features

- PL/Python overhaul
- Valid-on-creation Foreign Keys
- Extensible ENUMs
- Triggers on Views
- New Trigram implementation
- Reduced NUMERIC size
- ALTER TYPE without rewrite
- pg_dump directory format
9.1 Sessions at pgCon

- Real Federation Database System leveraging PostgreSQL FDW - Thursday 13:30
- Label-based Mandatory Access Control: Friday 10:00
- Writeable CTEs: Friday 11:00
- Distributing Extensions on PGXN: Friday 15:00
- Extensions Development: Friday 16:00
- Serializable Snapshot Isolation: Friday 16:00
Contact

• Josh Berkus: josh@pgexperts.com
  • blog: blogs.ittoolbox.com/database/soup
• PostgreSQL: www.postgresql.org
  • pgexperts: www.pgexperts.com
• Upcoming Events
  • pgOpen: September 14-16, Chicago
  • OpenSourceBridge: Portland, June

This talk is copyright 2010 Josh Berkus and is licensed under the creative commons attribution license. Special thanks for materials to: Hubert Lubaczewski (wCTE, collation examples), Andrew Dunstan (file_FDW)