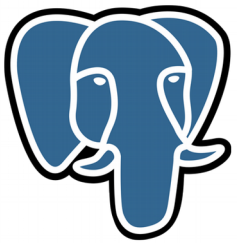
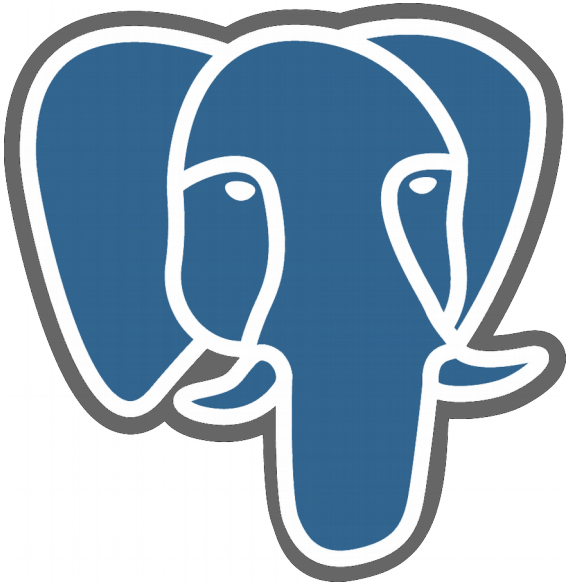


TodoPostgreSQL

Estadísticas y Monitorización

INSTRUCTOR:

José Segovia <info@todopostgresql.com>



Estadísticas

Parámetros en postgresql.conf:

- track_activities: Habilíta la recolección de información, "ON"
- track_counts:
- track_io_timing
- track_functions
- update_process_title
- Otros:
- log_min_duration_statement / log_connections / log_statement

Más información

Estadísticas

Algunas vistas y catálogos de interés:

- pg_stat_activity
- pg_stat_database
- pg_stat_all_tables
- pg_stat_user_functions
- pg_stat_all_indexes
- pg_locks

Más información

Recolector de estadísticas

- El logging es PostgreSQL es muy flexible y configurable
  - También permite registrar información de estadísticas
  - Los parámetros que lo controlan son:
  - **log_statement_stats** – Activa la demás estadísticas (no se deberá poner a "true" si cualquier otra estadística estuviera activada)
  - **log_parser_stats**
  - **log_planner_stats**
  - **log_executor_stats**

Recolector de estadísticas

- Dentro de la categoría de "Procesos de Utilidad" de PostgreSQL.
  - AUTOVACUUM (+Autovacuum launcher) y STATS COLLECTOR.

    - Mantienen las tablas "libres de filas muertas" (por MVCC).
    - Mantienen las estadísticas actualizadas para que el planificador pueda hacer un mejor trabajo.
    - Se describen juntos, porque colaboran:
      - Autovacuum actualiza estadísticas.
      - Stats collector puede priorizar autovacuum.

Recolector de estadísticas

- Cuando la recolección de estadísticas está habilitada, los datos generados son accesibles vía las tablas de sistema pg_stat:
  - **track_activities (boolean)** – Habilita la recolección de información sobre el comando que se esté ejecutando actualmente para cada sesión, además de la hora en la que comenzó dicha ejecución.

    El valor por defecto de este parámetro es ON.
  - **track_counts (boolean)** – Habilita la recolecta de estadísticas sobre la actividad de la base de datos.

    El valor por defecto de este parámetro es ON.

Monitorización

- La mayoría de los catálogos de sistema se copian de la plantilla de base de datos durante la creación de la base de datos y son, a partir de ahí, específicos a ella
  - Unos pocos de estos catálogos se comparten físicamente entre todas las bases de datos del cluster.

- Hay vistas en el catálogo global que dan información de monitorización
  - pg_stat_activity →listado de conexiones, usuarios, consultas...
  - pg_statio_*
  - pg_stat_bgwriter

Monitorización

- La actividad del servidor, se puede consultar mediante pg_stat_activity

- Con una consulta muestra:
  - los IDs de procesos y su hora de inicio
  - bases de datos y usuarios
  - consulta y hora de comienzo de la consulta que se esté ejecutando actualmente

- Ejecutar la siguiente consulta SQL:

```
SELECT * FROM pg_stat_activity;
```

Monitorización

- La vista pg_locks proporciona información sobre todos los bloqueos mantenidos por procesos activos del cluster (servidor de base de datos).
- Para revisar cuales son los procesos que están manteniendo o esperando los bloqueos, se utiliza la referencia en pg_stat_activity.

SELECT relname, pg_locks.* FROM pg_class, pg_locks WHERE relfilenode=relation AND NOT GRANTED;

Monitorización

- Para ver las conexiones actuales

```
SELECT datname, usename, query FROM pg_stat_activity ;
```

- Funciones de Emergencia:
  - Cancelar la consulta actual.

```
SELECT * FROM pg_terminate_backend(pid);
```

  - Terminar el proceso backend.

```
SELECT * FROM pg_cancel_backend(pid);
```

Te devuelven un true ("t") si se cancela el proceso o un false ("f") si no es posible.

Más información

Monitorización

- Para ver las consultas activas

```
SELECT datname, usename, query FROM pg_stat_activity
WHERE query != '<IDLE>' ;
```

- Para ver las consultas que llevan largo tiempo de ejecución

```
SELECT current_timestamp – query_start as runtime,
datname, usename, query FROM pg_stat_activity WHERE
query !='<IDLE>' ORDER BY datname desc;
```

Monitorización

- Para ver las consultas en espera

```
SELECT datname, usename, query FROM pg_stat_activity
WHERE wait_event = 'true' AND wait_event_type = 'true';
```

- Para Terminar las transacciones que quedaron pendientes por mucho tiempo

```
SELECT pg_terminate_backend(pid) FROM pg_stat_activity
WHERE query = '<IDLE> in transaction' AND
current_timestamp - query_start > '10 min';
```

Monitorización

- Para obtener detalles de bloqueos que mantienen consultas en espera

```
SELECT w.query AS waiting_query, w.pid AS w_pid, w.usename AS
w_user, l.query AS locking_query, l.pid AS l_pid, l.usename AS l_user,
t.schemaname || '.' || t.relname AS tablename FROM
pg_stat_activity w JOIN pg_locks l1 ON w.pid = l1.pid
AND NOT l1.granted JOIN pg_locks l2 ON l1.relation = l2.relation
AND l2.granted JOIN pg_stat_activity l ON l2.pid = l.pid JOIN
pg_stat_user_tables t ON l1.relation =
t.relid WHERE w.wait_event = 'f' AND w.wait_event_type = 'f' ;
```

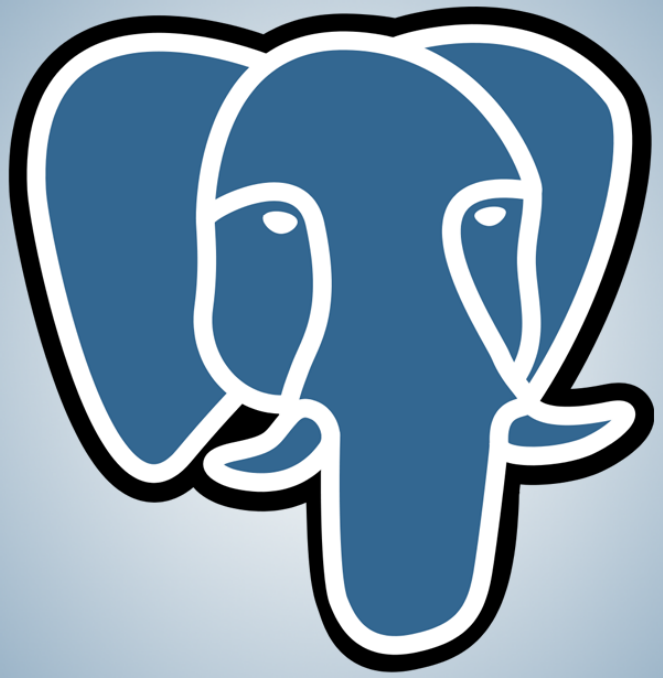
Monitorización

- Para averiguar si se está usando una tabla

```
CREATE TEMP TABLE tmp_stat_user_tablas AS SELECT * FROM
pg_stat_user_tables ;
```

- Después de pasar un rato

```
SELECT * FROM pg_stat_user_tables n JOIN
tmp_stat_user_tables t ON n.relid = t.relid AND
(n.seq_scan, n.idx_scan, n.n_tup_upd, n.n_tup_del) <>
(t.seq_scan, t.idx_scan, t.n_tup_upd, t.n_tup_del);
```



todopostgresql.com