統計情報

統計情報の種類

●統計情報は主に以下の２つがある。

①stats collector（統計情報コレクタ）プロセスが一定間隔で収集（更新）する DB の活動状況に関する稼働統計情報であり、アクセス統計情報ともいう。

②一部の DDL コマンド（ANALYZE、VACUUM、CREATE INDEX 等）実行時に収集（更新）する統計情報であり、システムカタログとして保存される。例えば、ANALYZE コマンドのパラメータにテーブル名がない時は DB の全てのテーブル。解析対象列を指定できるが、デフォルトは全ての列

●上記の他、統計情報までとは言えないが、各 DB の情報スキーマ（information schema）には、各 DB 内のテーブル、列、ユーザ等の定義情報を保持するビュー（table ビュー、column ビュー、enabled_role ビュー等）がある。これらのビューはポスグレ特有ではなく標準 SQL で定義されており、移植可能である。ポスグレ特有の情報はシステムカタログを参照するという点からは統計情報といえるかもしれない。

稼働統計情報

稼働統計情報の概要

●主要な稼働統計情報を収集するためには、track_counts 及び track_activities の両パラメータを on にしておく必要がある。両方ともデフォルトは on なので確認しておけばよい。

●収集した情報の結果は、一時フォルダである pg_stat_temp ディレクトリ（クラスタ直下）に格納され、他のプロセスに送られる。永続的なコピーが pg_stat ディレクトリ（クラスタ直下）に格納される。

●サーバ起動時にリカバリが実施される場合（即時シャットダウン、サーバクラッシュ、PITR 等）は、情報は全てリセットされる。意図的な統計情報リセット関数によるリセットは別項参照。

●収集結果の表示は全てビューで定義される。

●ユーザ定義関数の統計情報は、pg_stat_user_functions ビューで確認できるが、track_functions（デフォルトは none）を all 又は pl にする。

主な稼働統計情報その１：pg_stat_database ビュー

●クラスタ内で１つ。DB 毎の情報は WHERE 句で datnam='DB 名'と指定すればいい。

●詳細は日本 PostgreSQL ユーザ会の HP の該当項目を参照。その中で xact_commit、xact_rollback、blks_read、blks_hit、tup_fetched、deadlocks 等は押さえておいた方がいい。

主な稼働統計情報その２：pg_stat_all_tables ビュー等

●DB 内で１つ。テーブル毎の情報は WHERE 句で relnam='テーブル名'と指定すればいい。

●詳細は日本 PostgreSQL ユーザ会の HP の該当項目を参照。その中で seq_scan、seq_tup_read、idx_scan、idx_tup_fetch、n_tup_ins、n_tup_upd、n_tup_del、n_tup_hot_upd、n_live_tup、n_dead_tup、last_autovacuum 等は押さえておいた方がいい。

●pg_stat_all_tables ビューは、pg_stat_user_tables ビュー（ユーザテーブルのみ）と pg_stat_sys_tables ビュー（システムテーブルのみ）が合体したものであり、ユーザテーブル情報を参照したい場合は pg_stat_all_tables ビュー又は pg_stat_user_tables ビューのどちらを使用してもいい。

主な稼働統計情報その３：pg_stat_activity ビュー

●動作中（リアルタイム）のバックエンドプロセスの情報を提供する。

●詳細は日本 PostgreSQL ユーザ会の HP の該当項目を参照するのであるが、pg_stat_activity ビューは統計情報コレクタの中で最大のページを誇るため、全てを理解するのは不可能。その中で敢えていうならば、backend_start、xact_start、query_start、state_change、wait_event_type、wait_event、state、query_id、query、backend_type 等は押さえておいた方がいい。

主な稼働統計情報その４：pg_stat_bgwriter ビュー

●バックグラウンドライタプロセスの活動状況に関する統計情報である。クラスタ内で１つなので、WHERE 句で DB 等を指定する必要なし。以下に詳細を示す。

①共有メモリ内のダーティバッファをファイルに書き出すのは、チェックポインタプロセス（書き出し数は buffers_checkpoint）、バックグラウンドライタプロセス（書き出し数は buffers_clean）、そしてバックエンドプロセス（書き出し数は buffers_backend）があり、括弧内の書き出し数が pg_stat_bgwriter ビューで表示される。

②通常、ダーティバッファはチェックポインタプロセスが実行するが、チェックポイント時の書き込み量を削減するためにバックグラウンドライタプロセスが働く。つまり、チェックポインタプロセスの補助プロセスである。

③buffers_alloc は共有メモリに割り当てられたバッファ数であり、そのバッファが枯渇すると、バックエンドプロセスが自分で書き出しを行う。（←どこでデータを展開していたかは不明）その数が buffers_backend である。

④よって、buffers_backend が buffers_alloc より大きい場合は、shared_buffers が不足しているため、チューニングが必要となる。（←buffers_backend が存在する時点で不足しているような気がするが・・・）

hit or read

●共有バッファのヒット回数や共有バッファ以外から読み込んだ回数について、DB 全体における情報は、以下の①のとおり pg_stat_database ビューから取得できる。しかし、テーブル単位及びインデックス単位の情報は、pg_stat_all_tables ビューや pg_stat_all_indexes ビューではなく、以下の②及び③のビューから取得できる。

①pg_stat_database（blks_hit と blks_read）：DB 全体

②pg_statio_all_tables（heap_blks_hit と heap_blks_read）：テーブル単位。all が user でも同じ

③pg_statio_all_indexes（idx_blks_hit と idx_blks_read）：インデックス単位。all が user でも同じ

※②にも③と同じ idx_blks_hit と idx_blks_read があるが、同じものだと思われる。

●キャッシュヒット率は直接表示されないため、hit と read の値から SELECT 文を用いて計算することになる。

統計情報リセット関数によるリセット

●リセットは情報をゼロにすることであり、スーパーユーザ権限が必要である。

●pg_stat_reset_shared（'text'）とすると、クラスタ内に１つ（1 行のみ）存在する統計情報、例えば、引数が bgwriter の場合、pg_stat_bgwriter ビューの値が全てゼロになる。引数が archiver の場合、pg_stat_archiver ビューの値が全てゼロになる。

●pg_stat_reset は、現在の DB に関する全ての統計情報をゼロにする。

●pg_stat_reset_single_table_counters（oid）は、現在の DB 内にある１つのテーブル又はインデックスの統計情報をゼロにする。

fetch 関連

●pg_stat_database ビューの tup_fetched は DB 内の問い合わせで取り出された行数。シーケンシャルスキャンとかインデックススキャンとかは関係ない。

●pg_stat_all_tables ビュー等の idx_tup_fetch はインデックススキャン（idx_scan）で取り出された有効行の個数。シーケンススキャン（seq_scan）で取り出された有効行の個数は seq_tup_read であり、fetch の文言なし。

●pg_stat_all_indexes ビュー等の idx_tup_fetch は上記に同じ。ビュー名から当然シーケンシャルスキャンの情報はあるはずがない。また、当該ビューには idx_tup_read があるが、インデックススキャン（idx_scan）によって返されたインデックスの行数

システムカタログ

システムカタログ

●pg_statistic や pg_class 等のシステムカタログ（実体はテーブル）は、主にプランナ（オプティマイザ）が実行計画を作成する時に使用される。

●pg_statistic は、カラムに関する統計情報（列長、最頻値、分布等。ここで列長の総和は行長になる）を扱っており、default_statistics_target（デフォルトは 100）によってサンプリング数を決めている。デフォルトの 100 はヒストグラムを収集する統計情報（最頻値や分布等）の数であり、100 に定数である 300 を乗じた数、つまり 30,000 がデフォルトのサンプリング数となる。

●pg_class は、テーブル、ビュー、インデックスの格納行数（reltuples）や格納ページ数（relpages）等を扱う。

●DB 固有のものである pg_statistic や pg_class 等は、base ディレクトリ内の各 DB ディレクトリ内の pg_catalog スキーマ内に格納

●クラスタ内共通（クラスタ内で 1 つのみ）である pg_database や共有システムカタログのインデックス等は、global ディレクトリに格納

システムビュー

●システムカタログ以外にシステムビューがある。例えば、pg_statistic は、セキュリティの関係からスーパーユーザ専用であり、一般ユーザは pg_stats ビューを使う。

●pg_locks ビューは、ロックの取得待ちをしているプロセスを表示するが、当該ビューだけでは情報不足のため、pg_stat_activity ビューや pg_class システムカタログ等との結合処理を必要とする。それでも、ロックをブロックしている犯人は不明のままである。その時は、pg_blocking_pids 関数でロック待ちの PID を引数とすることで犯人を確認できる。別の方法として log_lock_waits パラメータ（デフォルトは off）を on にすると、deadlock_timeout（デッドロック状態か否かを調査する前にロックを待つ時間。デフォルトは 1s、スーパーユーザ権限）で設定された時間後にログに犯人が表示される。