PostgreSQL インサイド

pg_statsinfo のサイジング

技術を知る

FUJITSU

☐ 導入／環境設定　☐ 移行　☐ 性能　☐ チューニング　☐ バックアップ／リカバリー

☐ 冗長化／負荷分散　☑ 監視　☐ データ連携　☐ 災害対策　☑ 豆知識

PostgreSQL の周辺ツールの 1 つである pg_statsinfo は、サーバーの統計情報（以降、スナップショットと定義）を収集・蓄積・表示することで、PostgreSQL における日々の処理傾向についての把握、性能劣化などの兆候や問題発生の原因把握に役立ちます。ただし、大規模な業務システムにおいて取得したスナップショットを長期間に亘って蓄積し続けると、スナップショットの保存先となるデータベース（以降、リポジトリーDB と定義）の容量を圧迫します。そこで、運用を開始する前に業務要件からスナップショットのサイズを見積もり、運用開始後は不要になったスナップショットを定期的に削除することで、システムの資源を有効活用できます。
ここでは、pg_statsinfo で取得するスナップショットのサイズ、見積もり式、運用中のサイズについての確認方法を順に解説します。今回は、pg_statsinfo 12.1 をベースに説明します。

参考

- pg_statsinfo の仕様については、オープンソース・ソフトウェアに同梱されているマニュアルを参照してください。
- pg_statsinfo の使い方については「pg_statsinfo で統計情報を収集・蓄積する」で詳しく解説しています。

スナップショットのサイズ

スナップショットは、各インスタンスのデータベースクラスタ単位に取得されます。pg_statsinfo で取得するスナップショットのサイズには、固定値と可変値があります。固定値は、initdb コマンドでデータベースクラスタを作成した時に生成されるデータを保存するディレクトリーや postgres データベースなどのサイズです。可変値は、データベースクラスタ内に作成したデータベースやテーブルなどのオブジェクト数に応じたサイズです。
監視対象インスタンスが複数ある場合はその数だけスナップショット容量が必要になります。特にストリーミングレプリケーション機能を利用する場合、プライマリーサーバーとスタンバイサーバーの両方のスナップショットを 1 つのリポジトリーDB に格納することになり、レプリケーションを構成するインスタンス（サーバー）の数だけ必要な容量が増えるので注意してください。
以下の表に、スナップショット取得 1 回あたりのサイズを示します。

表 1. スナップショット取得 1 回あたりのサイズ

分類	構成要素	基本サイズ（バイト）（注 1）	オブジェクト数（注 2）	必要なサイズ（バイト）（注 3）	補足
固定値	snapshot	80	1	80	スナップショット情報を格納
	cpu	96	1	96	OS リソース（CPU）に関する情報を格納
	device	144	1	144	OS リソース（ディスク I/O）に関する情報を格納

分類	構成要素	基本サイズ（バイト）（注 1）	オブジェクト数（注 2）	必要なサイズ（バイト）（注 3）	補足
	loadavg	48	1	48	ロードアベレージに関する情報を格納
	memory	64	1	64	メモリーに関する情報を格納
	bgwriter	64	1	64	バックグラウンドライタに関する情報を格納
	xlog	96	1	96	WAL に関する情報を格納
	archive	64	1	64	WAL アーカイブに関する情報を格納
	database	256	1	256	データベースに関する情報を格納
	tablespace	208	2	416	データベーススペースに関する情報を格納
	schema	96	5	480	スキーマに関する情報を格納
	table	336	40	13,440	テーブルに関する情報を格納
	index	240	30	7,200	インデックスに関する情報を格納
	column	160	459	73,440	カラムに関する情報を格納
	role	64	9	576	ロールに関する情報を格納
	inherits	64	5	320	継承階層に関する情報を格納
	function	160	0	0	関数に関する情報を格納
	replication	352	0	0	レプリケーションに関する情報を格納

分類	構成要素	基本サイズ（バイト）（注 1）	オブジェクト数（注 2）	必要なサイズ（バイト）（注 3）	補足
	replication_slots	208	0	0	レプリケーションスロットに関する情報を格納
	setting	112	16	1,792	GUC に関する情報を格納
	固定値の合計			**98,576**	**固定値は約 100 キロバイト**
可変値	database	256	（可変）	256 ×（可変）	データベースに関する情報を格納
	tablespace	208	（可変）	208 ×（可変）	データベーススペースに関する情報を格納
	schema	96	（可変）	96 ×（可変）	スキーマに関する情報を格納
	table	336	（可変）	336 ×（可変）	テーブルに関する情報を格納
	index	240	（可変）	240 ×（可変）	インデックスに関する情報を格納
	column	160	（可変）	160 ×（可変）	カラムに関する情報を格納
	role	64	（可変）	64 ×（可変）	ロールに関する情報を格納
	inherits	64	（可変）	64 ×（可変）	継承階層に関する情報を格納
	function	160	（可変）	160 ×（可変）	関数に関する情報を格納
	replication	352	（可変）	352 ×（可変）	レプリケーションに関する情報を格納
	replication_slots	208	（可変）	208 ×（可変）	レプリケーションスロットに関する情報を格納
	setting	112	（可変）	112 ×（可変）	GUC に関する情報を格納（xxx.conf ファイルに設定されたものと環境変数で

分類	構成要素	基本サイズ（バイト）（注 1）	オブジェクト数（注 2）	必要なサイズ（バイト）（注 3）	補足
					設定されたものが統計対象）
	可変値の合計			**－**	**可変値のため合計も可変**

- 注 1　オブジェクト 1 個のサイズです。
- 注 2　postgresql.conf のパラメーター（pg_statsinfo.excluded_dbnames など）がデフォルト値の場合です。
- 注 3　基本サイズにオブジェクト数を掛けた値です。

ポイント

可変値の各オブジェクト数についての確認方法を以下に示します。データベースクラスタ単位に確認するものと、データベース単位に確認するものがあります。

確認単位	構成要素	確認方法（注 4）
データベースクラスタ単位	database	SELECT count(d.oid) FROM pg_database d WHERE datallowconn AND d.oid >= 16384;
	tablespace	SELECT count(oid) FROM pg_tablespace WHERE oid >= 16384;
	role	SELECT count(oid) FROM pg_roles where oid >= 16384;
	replication	SELECT count(pid) FROM pg_stat_replication;
	replication_slots	SELECT count(slot_name) FROM pg_replication_slots;
	setting	SELECT count(name) FROM pg_settings WHERE source NOT IN ('client', 'default', 'session') AND setting <> boot_val;
データベース単位	schema	SELECT count(oid) FROM pg_namespace WHERE oid >= 16384;
	table	SELECT count(c.oid) FROM pg_class c LEFT JOIN pg_index i ON c.oid = i.indrelid LEFT JOIN pg_class t ON c.reltoastrelid = t.oid LEFT JOIN pg_index x ON c.reltoastrelid = x.indrelid LEFT JOIN pg_namespace n ON c.relnamespace = n.oid WHERE c.relkind IN ('r', 't') AND c.oid >= 16384;
	index	SELECT count(x.indexrelid) FROM pg_class c JOIN pg_index x ON c.oid = x.indrelid JOIN pg_class i ON i.oid = x.indexrelid JOIN pg_namespace n ON n.oid = c.relnamespace WHERE c.relkind IN ('r', 't') AND x.indexrelid >= 16384;
	column	SELECT count(attname) FROM pg_attribute a LEFT JOIN pg_class c ON a.attrelid = c.oid LEFT JOIN pg_statistic s ON a.attnum = s.staattnum AND a.attrelid = s.starelid AND NOT s.stainherit LEFT JOIN pg_namespace n ON c.relnamespace = n.oid WHERE a.attnum > 0 AND c.relkind IN ('r', 't') AND c.oid >= 16384;

確認単位	構成要素	確認方法（注 4）
	inherits	SELECT count(i.inhrelid) FROM pg_inherits i JOIN pg_class c ON i.inhrelid = c.oid JOIN pg_namespace n ON c.relnamespace = n.oid WHERE i.inhrelid >= 16384;
	function	SELECT count(oid) FROM pg_proc WHERE oid >= 16384;

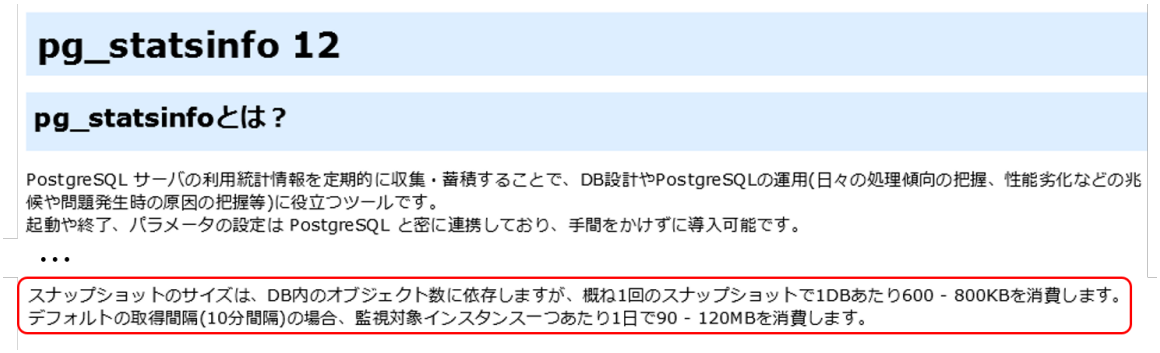
- 注 4　確認方法の「16384」は、OID（オブジェクト識別子）で PostgreSQL が内部で管理する値です。ユーザーが作成するオブジェクトの OID は「16384」以降が採番されます。詳細については、PostgreSQL 12.0 文書の「69.2.2 OID の割当」を参照してください。

見積もり式

見積もり方法は 2 つあります。運用前に必要な容量をおおまかに見積もりたい場合は「概算見積もり式」をご利用ください。業務要件やデータベース、テーブル数などが明確で可変要素の変動が少ない場合は「詳細見積もり式」をご利用ください。

概算見積もり式

pg_statsinfo に同梱されているマニュアルに、スナップショットのサイズに関する記載があります。そこには「スナップショットのサイズは、DB 内のオブジェクト数に依存しますが、概ね 1 回のスナップショットで 1DB あたり 600 - 800KB を消費します。デフォルトの取得間隔（10 分間隔）の場合、監視対象インスタンス一つあたり 1 日で 90 - 120MB を消費します。」と書かれています。


pg_statsinfo 12

pg_statsinfoとは？

PostgreSQL サーバの利用統計情報を定期的に収集・蓄積することで、DB設計やPostgreSQLの運用(日々の処理傾向の把握、性能劣化などの兆候や問題発生時の原因の把握等)に役立つツールです。
起動や終了、パラメータの設定は PostgreSQL と密に連携しており、手間をかけずに導入可能です。
・・・
スナップショットのサイズは、DB内のオブジェクト数に依存しますが、概ね1回のスナップショットで1DBあたり600 - 800KBを消費します。
デフォルトの取得間隔(10分間隔)の場合、監視対象インスタンス一つあたり1日で90 - 120MBを消費します。

1 回のスナップショットで 1 データベースあたりのサイズを最大の「約 800 キロバイト」と見積もる場合は、以下の式になります。

800 キロバイト × データベース数 ×（スナップショットの保持時間 ÷ スナップショットの取得間隔）

【例】

「概算見積もり式」を使って見積もってみます。pg_statsinfo の設定値は、以下とします。

設定項目	値	postgresql.conf のパラメーター名
データベース数	2	
スナップショットの取得間隔（分）	10min（デフォルト値）	pg_statsinfo.snapshot_interval
スナップショットの保持期間（日）	7（デフォルト値）	pg_statsinfo.repository_keepday
自動メンテナンス機能	on（デフォルト値）	pg_statsinfo.enable_maintenance

```
800 キロバイト × 2 × （7 × 24 × 60 ÷ 10） ÷ 1024 ＝ 約 1,575 メガバイト
```

詳細見積もり式

表 1.におけるスナップショットのサイズについて固定値と可変値を使って見積もる場合は、以下の式になります。

```
（固定値 ＋ 可変値） × （スナップショットの保持時間 ÷ スナップショットの取得間隔）
```

【例】

「詳細見積もり式」を使って見積もってみます。pg_statsinfo の設定値は、以下とします。

設定項目		値	postgresql.conf のパラメーター名
固定値		約 100 キロバイト	－
可変値	データベースクラスタ単位	約 5 キロバイト	－
	データベース単位	約 828 キロバイト	－
スナップショットの取得間隔（分）		10min（デフォルト値）	pg_statsinfo.snapshot_interval
スナップショットの保持期間（日）		7（デフォルト値）	pg_statsinfo.repository_keepday
自動メンテナンス機能		on（デフォルト値）	pg_statsinfo.enable_maintenance

```
（100 ＋ 5 ＋ 828） × （7 × 24 × 60 ÷ 10） ÷ 1024 ＝ 約 918 メガバイト
```

補足

pg_statsinfo の自動メンテナンス機能を使用することで、保持期間を経過した古いスナップショットは自動的に削除されます。上記例では、自動メンテナンス機能が作動しているため、最大 7 日分のスナップショットを見積もっています。

運用中のサイズ調整

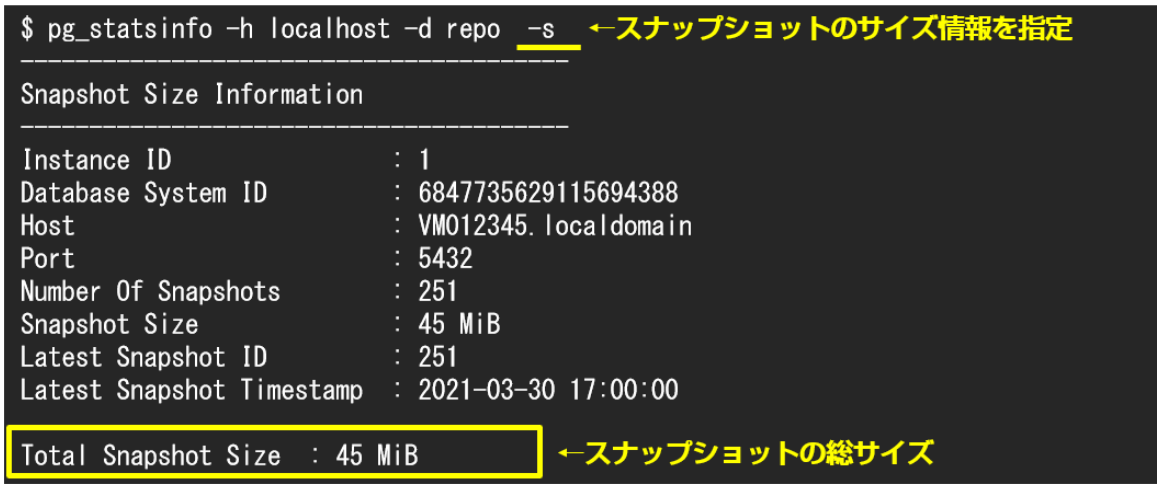
運用中にスナップショットのサイズを確認したり、手動で古いスナップショットを削除したりすることもできます。

運用中のスナップショットのサイズ確認

スナップショットの情報は、pg_statsinfo のコマンドを使うことにより、リポジトリーDB に保存されたスナップショットの情報をテキスト形式で標準出力に表示します。以下を実行すると、スナップショットの総サイズが確認できます。

```
$ pg_statsinfo -h localhost -d repo -s
```

実行結果は以下の画面キャプチャー例のような形で表示されます。



取得サイズの変更

運用前の見積もりよりもスナップショットのサイズが大きい傾向が見られた場合は、以下の対処を実施することで、容量を削減できます。postgresql.conf のパラメーター値を変更して、PostgreSQL を再起動してください。

- スナップショットの取得間隔（pg_statsinfo.snapshot_interval）の値を長く変更する。
- スナップショットの保持期間（pg_statsinfo.repository_keepday）の値を短く変更する。

古いスナップショットの強制削除

古いスナップショットを強制的に削除したい場合は、pg_statsinfo の関数を使って手動で削除してください。

【例】

取得日時が 2021-04-01 17:00:00 より古いスナップショットを削除する場合は以下を実行します。

```
$ psql -d mydb -c "SELECT statsinfo.maintenance('2021-04-01 17:00:00'::timestamptz);"
```


特定のスナップショットの強制削除

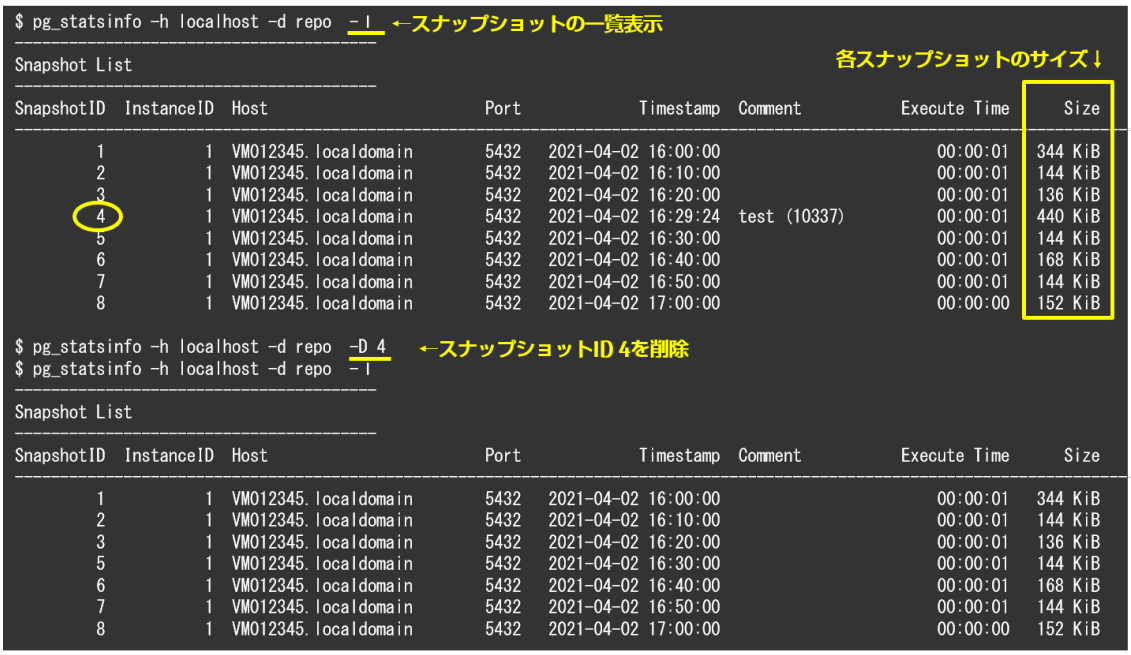
特定のスナップショットを強制的に削除したい場合は、pg_statsinfo のコマンドを使って手動で削除してください。

【例】

スナップショット一覧で各スナップショットのサイズを確認し、SnapshotID 4 を削除する場合は以下を実行します。

```
$ pg_statsinfo -h localhost -d repo -l
$ pg_statsinfo -h localhost -d repo -D 4
```

実行結果は以下の画面キャプチャー例のような形で表示されます。



pg_statsinfo を使ってシステムの監視を行う場合は、事前にスナップショットのサイズを正しく見積もって、業務を運用することをお勧めします。

参考

FUJITSU Software Enterprise Postgres では、バージョン 10 から pg_statsinfo を同梱しています。本記事の見積もり式は、FUJITSU Software Enterprise Postgres に同梱されている pg_statsinfo にも適用できます。

2021 年 6 月 25 日