FUJITSU
Enterprise Postgres 14 SP1

リファレンス

Linux

J2UL-2754-02ZJZ0(00)
2022年9月

まえがき

本書の目的

本書は、コマンドリファレンスです。
本書は、PostgreSQLから機能を拡張したFUJITSU Enterprise Postgresのコマンドおよびオプションについて説明しています。

本書の読者

本書は、FUJITSU Enterprise Postgresを管理、運用される方を対象としています。

なお、本書は、以下についての一般的な知識があることを前提に書かれています。

- PostgreSQL
- SQL
- Linuxに関する知識

本書の構成

本書の構成と内容は以下のとおりです。

第1章 コマンド一覧と記述形式

コマンドの一覧と、コマンドの記述形式について説明しています。

第2章 クライアントコマンド

クライアントコマンドについて説明しています。

第3章 サーバコマンド

サーバコマンドについて説明しています。

第4章 Mirroring Controllerコマンド

Mirroring Controllerコマンドについて説明しています。

第5章 Connection Managerコマンド

Connection Managerコマンドについて説明しています。

本書の読み方

本書では主にUNIX/Linuxでの例を使用しています。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

出版年月および版数

2022年　9月　第2版
2022年　2月　初版

著作権

Copyright 2021-2022 FUJITSU LIMITED

目　次

第1章 コマンド一覧と記述形式 1
1.1 コマンド一覧 1
1.1.1 クライアントコマンド 1
1.1.2 サーバコマンド 1
1.1.3 Mirroring Controllerコマンド 1
1.1.4 Connection Managerコマンド 2
1.2 コマンドの記述形式 2

第2章 クライアントコマンド 3
2.1 pg_dumpall 3
2.2 pgx_loader 3

第3章 サーバコマンド 6
3.1 pg_ctl 6
3.2 pgx_dmpall 6
3.3 pgx_fjqssinf 9
3.4 pgx_keystore 10
3.5 pgx_rcvall 11
3.6 postgres 14

第4章 Mirroring Controllerコマンド 15
4.1 mc_arb 15
4.2 mc_ctl 16

第5章 Connection Managerコマンド 21
5.1 cm_ctl 21

第1章 コマンド一覧と記述形式

コマンドの一覧とコマンド記述形式について説明します。

1.1 コマンド一覧

本章では、PostgreSQL Documentationの“PostgreSQL Client Applications”、“PostgreSQL Server Applications”で説明していないコマンドや、コマンドのオプションについて説明します。

1.1.1 クライアントコマンド

以下のコマンドについて、PostgreSQL Documentationの“PostgreSQL Client Applications”で説明していないオプションがあります。

コマンド	機能概要
pg_dumpall	データベースクラスタをスクリプトファイルへ抽出する

PostgreSQL Documentationの“PostgreSQL Client Applications”で説明していない以下のコマンドがあります。

コマンド	機能概要
pgx_loader	外部ファイルのデータをPostgreSQLのテーブルへ投入

1.1.2 サーバコマンド

以下のコマンドについて、PostgreSQL Documentationの“PostgreSQL Server Applications”で説明していないオプションがあります。

コマンド	機能概要
pg_ctl	データベースサーバの初期化、起動、停止、制御
postgres	データベースサーバ

PostgreSQL Documentationの“PostgreSQL Server Applications”で説明していない以下のコマンドがあります。

コマンド	機能概要
pgx_dmpall	データディレクトリ、テーブル空間および設定ファイルのバックアップ
pgx_fjqssinf	障害調査データの収集
pgx_keystore	キーストアを管理する
pgx_rcvall	データディレクトリ、テーブル空間および設定ファイルのリカバリ

1.1.3 Mirroring Controllerコマンド

Mirroring Controllerには、以下のコマンドがあります。

コマンド	機能概要
mc_arb	Mirroring Controller裁定プロセスの起動、停止および状態表示
mc_ctl	Mirroring Controllerの起動、停止、サーバの状態表示、切り替え、および切り離し

1.1.4 Connection Managerコマンド

Connection Managerには、以下のコマンドがあります。

コマンド	機能概要
cm_ctl	conmgrプロセスの起動、停止および状態表示

1.2 コマンドの記述形式

コマンドの記述形式について以下に示します。

項目	説明
[ ]	省略できるオプションであることを示します。
…	繰り返して指定できることを示します。

第2章 クライアントコマンド

PostgreSQL Documentationの“PostgreSQL Client Applications”で説明していないオプションについて説明します。

2.1 pg_dumpall

名前

pg_dumpall --データベースクラスタをスクリプトファイルへ抽出する

概要

pg_dumpall [*connection-option*...] [*option*...]

オプション

--no-tablespace-encryption

テーブル空間を暗号化するコマンドを出力しません。生成されたSQLスクリプトを実行すると、もともと暗号化されていたデータが暗号化せずにリストアされます。

参照

その他の説明については、“PostgreSQL Documentation”の“Reference”の“pg_dumpall”を参照してください。

2.2 pgx_loader

名前

pgx_loader --外部ファイルのデータをPostgreSQLのテーブルへ投入

概要

pgx_loader load -c *command* [options...]

pgx_loader recovery -t *table*

説明

pgx_loaderは、PostgreSQLのテーブルへの外部ファイルのデータ投入処理、および投入処理中に準備されたトランザクションのコミットまたはロールバックを行います。

loadモードは、*command*で与えられたCOPY FROMコマンドを任意の並列度で実行することで、高速にデータ投入を行います。正常に完了した場合、以下の形式の標準出力を返します。

LOAD *count*
*count*はロードされた行数です。

ロード処理の進捗を確認する場合はpgx_stat_progress_loaderビューを参照してください。

recoveryモードは、データ投入中に準備されたトランザクションをコミットまたはロールバックします。

参照

pgx_stat_progress_loaderビューの詳細は、“運用ガイド”の“pgx_stat_progress_loader”を参照してください。

オプション

-a
--echo-sql

実行されたコマンドを標準出力に表示します。

-c *command*
--copy-command=*command*

実行されるCOPY FROMコマンドです。COPY FROMコマンドのFROM句にSTDINが指定された場合、入力データはpgx_loaderコマンドの標準入力から読み込まれます。この場合、外部ファイルまたは外部プログラムへのアクセス権限はデータベースサーバではなくローカルユーザーのものを使用するので、SQLのスーパーユーザー権限(またはpg_read_server_filesかpg_execute_server_programのいずれかのロールを持つこと)は必要ありません。

このオプションに指定するCOPY FROMコマンドのFORMATオプションにbinaryを選択することはできません。そのため、バイナリ形式の入力ファイルは指定できません。

このオプションに指定するCOPY FROMコマンドにFREEZEオプションを指定することはできません。

参照

COPY FROMコマンドの詳細は、“PostgreSQL Documentation”の“COPY”を参照してください。

-j number-of-jobs
--jobs=number-of-jobs

COPY FROMコマンドで実行されるデータの変換、テーブルの作成およびインデックスの作成処理を、このオプションで指定した数のバックグラウンドワーカ(並列ワーカ)が同時実行します。このオプションは、複数プロセッサマシンで稼働するインスタンスに大量データを投入する時間を劇的に減らすことができます。

このオプションの最適値はサーバ、クライアント、ネットワークの構成に依存します。要素にはCPUコア数やディスク構成も含まれます。試行する最初の値としてサーバのCPUコア数を勧めます。当然ながらあまりに大きな値を使用すると、スラッシングやコンテキストスイッチのために性能が劣化することになります。

このオプションに指定可能な値は2から128です。デフォルトは2です。

-t table
--table=table

指定されたテーブルのみについて、準備されたトランザクションを完了します。

-?
--help

pgx_loaderコマンドライン引数の使用方法を表示し、終了します。

以下のコマンドラインオプションは、データベース接続パラメータを制御します。

-d connstr
--dbname=connstr

接続するデータベースの名前を指定します。指定されていない場合はPGDATABASE環境変数が使われます。この変数も設定されていない場合は、オペレーティングシステムのユーザー名が使用されます。

-h host
--host=host

データベースサーバが稼働しているマシンのホスト名を指定します。この値がスラッシュから始まる場合、Unixドメインソケット用のディレクトリとして使用されます。デフォルトは、設定されていれば環境変数PGHOSTから取得されます。設定されていなければ、Unixドメインソケット接続とみなされます。

-p port
--port=port

サーバが接続を監視するTCPポートもしくはローカルUnixドメインソケットファイルの拡張子を指定します。デフォルトは、設定されている場合、環境変数PGPORTの値になります。設定されていなければ、デフォルト値は27500となります。

-U username
--username=username

接続ユーザー名です。

-w
--no-password

パスワードの入力を促しません。サーバがパスワード認証を必要とし、かつ、.pgpassファイルなどの他の方法が利用できない場合、接続試行は失敗します。バッチジョブやスクリプトなどパスワードを入力するユーザーが存在しない場合にこのオプションは有用かもしれません。

-W
--password

データベースに接続する前に、pgx_loaderは強制的にパスワード入力を促します。サーバがパスワード認証を要求する場合 pgx_loaderは自動的にパスワード入力を促しますので、これが重要になることはありません。しかし、pgx_loaderは、サーバにパスワードが必要かどうかを判断するための接続試行を無駄に行います。こうした余計な接続試行を防ぐために -W の入力が有意となる場合もあります。

診断

loadモード

0: 正常終了

0以外: 異常終了

recoveryモード

0: 完了すべき準備されたトランザクションはありません

3: 準備されたトランザクションをコミットしました

4: 準備されたトランザクションをロールバックしました

上記以外: 異常終了

注意

pgx_loaderによって読み込まれるテーブルの行の順序は、ファイルの行の順序に一致しないことがあります。これは、ファイルの行が複数の並列ワーカによってテーブルに並列に挿入されているためです。

例

以下は、外部ファイル“/path/to/data.csv”(2000レコード)をテーブル“tbl”に並列度:3でロードする例です。

```
$ pgx_loader load -j 3 -c "COPY tbl FROM '/path/to/data.csv' WITH CSV"
LOAD 2000
```

以下は、外部ファイル“/path/to/data.csv”(2000レコード)をpgx_loaderコマンドの標準入力から読み取り、テーブル“tbl”に並列度:3でロードする例です。

```
$ pgx_loader load -j 3 -c "COPY tbl FROM STDIN WITH CSV" < /path/to/data.csv
LOAD 2000
```

以下は、テーブル“tbl”に関連する準備されたトランザクションを完了する例です。

```
$ pgx_loader recovery -t tbl
```

第3章 サーバコマンド

PostgreSQL Documentationの“PostgreSQL Server Applications”で説明していないコマンドおよびオプションについて説明します。

3.1 pg_ctl

名前

pg_ctl --データベースサーバの初期化、起動、停止、制御

概要

```
pg_ctl start [-D datadir] [-l filename] [-W] [-t seconds] [-s]
          [-o options] [-p path] [-c] [--user-pin | --keystore-passphrase | --kms-secret]

pg_ctl restart [-D datadir] [-m s[mart] | f[ast] | i[mmediate] ]
            [-W] [-t seconds] [-s] [-o options] [-c]
            [--user-pin | --keystore-passphrase | --kms-secret]
```

オプション

ハードウェアベースのキーストアオプション

--user-pin

キーストアをオープンするためのユーザPINの入力を要求します。

ファイルベースのキーストアオプション

--keystore-passphrase

キーストアをオープンするためのパスフレーズの入力を要求します。

鍵管理システムをキーストアとして使用する場合のキーストアオプション

--kms-secret

キーストアをオープンするための秘密情報の入力を要求します。

参照

その他の説明については、“PostgreSQL Documentation”の“Reference”の“pg_ctl”を参照してください。

3.2 pgx_dmpall

名前

pgx_dmpall -- データディレクトリ、テーブル空間および設定ファイルのバックアップ

概要

pgx_dmpall [*option*…]

説明

データディレクトリ、テーブル空間、および設定ファイルをバックアップします。バックアップデータの格納先は、postgresql.confのbackup_destinationパラメータで指定したバックアップ格納ディレクトリです。また、バックアップの正常終了時に不要なアーカイブ済みWALを削除します。

オプション

-c

設定ファイルのみをバックアップします。設定ファイルは以下の3ファイルです。

- postgresql.confファイル（postgresql.conf）
- ホストベース認証用ファイル（pg_hba.conf）
- ident認証用設定ファイル（pg_ident.conf）

なお、postgresql.confのincludeなど、外部参照が設定されている場合は、参照先のファイルもバックアップします。

-C fast|spread
--checkpoint=fast|spread

チェックポイントをfastまたはspread（デフォルト）に設定します。fastを指定すると、バックアップ開始時のチェックポイント処理は高速になりますが、集中したI/Oのために動作中のアプリケーションへの性能の影響が大きくなります。spreadではチェックポイントはゆっくり実行されるためアプリケーションへの影響は小さいですが、バックアップに時間がかかります。

-D *datadir*

データディレクトリを指定します。省略時は、環境変数 PGDATA が有効となります。

-f *config_file*

設定ファイルpostgresql.confを指定します。postgresql.confファイルのdata_directoryパラメータに設定したデータディレクトリと設定ファイルを、異なるディレクトリで運用している場合に設定します。

-P *tablespaces_list_file*
--tablespaces-list-file=*tablespaces_list_file*

コピーコマンドでバックアップ対象とするテーブル空間の名前を記述したファイルパス名を1024バイト未満の長さで指定します。

以下に、ファイル内の形式を説明します。

テーブル空間の名前＜改行＞
テーブル空間の名前＜改行＞
・・・

ファイル内に記述されなかったテーブル空間は、バックアップ格納ディレクトリにバックアップします。また、本オプションの指定がなければ、すべてのテーブル空間をコピーコマンドによってバックアップします。
本オプションは、-Yオプションが指定されている場合に指定可能であり、コピーコマンドでバックアップするテーブル空間を限定する場合に指定します。

-U *username*
--username=*username*

データベースのスーパーユーザーのユーザー名を指定します。このコマンドを実行している実効ユーザーの名前がデフォルトです。

-Y *copy_command_file*
--copy-command=*copy_command_file*

バックアップ用コピーコマンドのファイルパス名を1024バイト未満の長さで指定します。本プションは-cオプションと同時に指定することはできません。

-w
--no-password

パスワードの入力を促しません。サーバがパスワード認証を必要とし、かつ、.pgpassファイルなどの他の方法が利用できない場合、接続試行は失敗します。バッチジョブやパスワードを入力するユーザーが存在しない場合にこのオプションは有用かもしれません。

-W
--password

データベースに接続する前に、pgx_dmpallは強制的にパスワード入力を促します。

サーバがパスワード認証を要求する場合pgx_dmpallは自動的にパスワード入力を促しますので、これが重要になることはありません。しかし、pgx_dmpallは、サーバにパスワードが必要かどうかを判断するための接続試行を無駄に行います。こうした余計な接続試行を防ぐために-Wの入力が有意となる場合もあります。

-z

デフォルトの圧縮レベルでバックアップファイルのzlibによる圧縮を有効にします。

本オプションは、-cオプションと一緒に指定することはできません。

-Z level
--compress=level

圧縮レベルを指定することで、バックアップファイルのzlibによるdeflate形式の圧縮を有効にします。(0～9まで、0は圧縮なし、9が最大圧縮)

本オプションは、-cオプションと一緒に指定することはできません。

--maintenance-db=*dbname*

接続先となるデータベースの名前を指定します。 指定がなければ、postgresデータベースが使用されます。もし存在しなければtemplate1が使用されます。

接続可能であれば、どのデータベースを指定しても構いません。

--exclude-copy-cluster

データベースクラスタをコピーコマンドのバックアップ対象から除外します。本オプションは-Yオプションが指定されている場合に指定可能です。本オプションの指定がなければ、データベースクラスタをコピーコマンドによってバックアップします。

環境

PGDATA

データディレクトリを指定します。-Dオプションを使用して上書きすることができます。

診断

0: 正常終了

0以外: 異常終了

注釈

このコマンドは、データベースサーバの稼働中にのみ実行できます。

このコマンドは、PostgreSQLユーザーアカウントで実行してください。

バックアップ格納ディレクトリ内のファイルを更新、および削除しないでください。データディレクトリを復旧できなくなる可能性があります。

バックアップ格納ディレクトリには、ほかのファイルを格納しないでください。

このコマンドはデータベースに接続するため、接続を1つ使用します。接続を確立するために、UNIXドメイン・ソケットを使用します。そのため、pg_hba.confファイルでこの接続を許可してください。

本コマンドは、スタンバイサーバでは実行できません。

例

以下は、データディレクトリ、テーブル空間および設定ファイルをバックアップする例です。このとき、バックアップにより不要になったアーカイブ済みWALも破棄されます。

```
$ pgx_dmpall
```

関連項目

pgx_rcvall

3.3 pgx_fjqssinf

名前

pgx_fjqssinf - 障害調査データの収集

概要

```
pgx_fjqssinf  -i {1|2|3} [-w directory]
```

説明

環境の構築または運用中に発生したトラブルの原因が特定できない場合、初期調査のための情報を収集します。収集された調査情報は、指定されたディレクトリにpgx_fjqssinf_YYYYMMDD_HHMMSS/として作成されます。

オプション

-i {1|2|3}

発生した問題のインシデントを指定します。プロセスエラーの場合は1、プロセス結果エラーの場合は2、応答なしの場合は3を指定します。データベースサーバが情報を収集するためには、このオプションを指定する必要があります。裁定サーバで情報を収集する場合は、1を指定します。

-w directory

収集データの格納ディレクトリを指定します。デフォルトは/tmpです。

環境 (データベースサーバによって情報が収集される場合)

PGDATA

データディレクトリを指定します。.

PGDATABASE

接続するデータベースの名前を指定します。接続可能な任意のデータベースを指定できます。

PGPORT

インスタンスのポート番号を指定します。デフォルトのポート番号(27500)が変更されていない場合は、指定しないでください。

PGUSER

データベーススーパーユーザーのユーザー名を指定します。データベースのスーパーユーザーは、クライアント認証を許可するように設定する必要があります。

MCCONTROLDIR

Mirroring Controller管理ディレクトリを指定します。データベース多重化運用をしている場合にのみ指定してください。

環境 (裁定サーバによって情報が収集される場合)

ARBCONTROLDIR

Mirroring Controller裁定プロセスの管理ディレクトリを指定します。

ARBUSER

Mirroring Controller裁定プロセスを初期化したOSユーザーを指定します。

診断

0: 正常終了

0以外: 異常終了

注釈

本コマンドは、スーパーユーザー(root)で実行してください。

例

以下は、プロセスが失敗した場合の初期調査のための情報収集の例です。

```
$ /opt/fsepv14server64/bin/pgx_fjqssinf -i 1
```

以下は、裁定サーバによる初期調査のための情報収集の例です。

```
$ /opt/fsepv14assistant/bin/pgx_fjqssinf -i 1
```

3.4 pgx_keystore

名前

pgx_keystore -- キーストアを管理する

概要

pgx_keystore [-a|--enable-auto-open] [*option*...] *keystore_location*

pgx_keystore [-s|--obfuscate-secret] [*option*...]

説明

pgx_keystoreはキーストアの自動オープンの有効化を行います。

オプション

ファイルベースのキーストアを使用して自動オープンを有効化する場合

-a
--enable-auto-open

キーストアの自動オープンを有効にします。これにより、データベースサーバの起動時にパスフレーズを入力しなくても、自動的にキーストアがオープンします。

自動オープンを有効にすると、キーストアのファイルkeystore.ksと同じディレクトリに、難読化されたコピーであるkeystore.aksが作成されます。自動オープンを無効にするにはkeystore.aksを削除してください。

-P *passphrase*
--passphrase=*passphrase*

キーストアをオープンするために使用しているパスフレーズを指定します。このオプションを省略すると、パスフレーズを入力するためのプロンプトが表示されます。

keystore_location

キーストアのファイルの場所を絶対パスまたは相対パスで指定します。

鍵管理システムをキーストアとして使用して自動オープンを有効化する場合

-s
--obfuscate-secret

鍵管理システムに接続するために必要な秘密情報を難読化します。難読化した秘密情報を鍵管理システム接続情報ファイルの認証オプションとして指定することにより、データベースサーバの起動時に鍵管理システムの資格情報を入力しなくても、自動的にキーストアがオープンします。

--secret=*secret*

鍵管理システムに接続するために必要な秘密情報を指定します。このオプションを省略すると、秘密情報を入力するためのプロンプトが表示されます。

-o *obfuscated-secret-file*

難読化した秘密情報を格納するファイルを指定します。ファイルが既に存在する場合は上書きせず異常終了します。

診断

0: 正常終了

0以外: 異常終了

注釈

このコマンドは、データベースサーバが稼働中でも停止中でも実行できます。データベースサーバには接続しません。

このコマンドは、鍵管理システムに接続しません。

例

ファイルベースのキーストアを使用する場合に、キーストアの自動オープンを有効化します。

```
$ pgx_keystore -a /key/store/location/keystore.ks
```

鍵管理システムをキーストアとして使用する場合に、秘密にすべき資格情報を難読化してキーストアの自動オープンを有効化します。

```
$ pgx_keystore -s -o /example/keypassphrase.ksc
Enter secret:
```

3.5 pgx_rcvall

名前

pgx_rcvall -- データディレクトリ、テーブル空間および設定ファイルのリカバリ

概要

pgx_rcvall [*option*…]

説明

pgx_dmpallコマンドでバックアップしたデータとアーカイブ済みWALを使用して、データディレクトリ、テーブル空間および設定ファイルをリカバリします。復旧時点を示すオプションのいずれも指定しない場合、すべてのアーカイブ済みWALを適用して、最新状態にデータをリカバリします。

オプション

-B *backupdir*

バックアップ格納ディレクトリを指定します。データディレクトリが破壊されている場合、本オプションを省略することはできません。

-D *datadir*

データディレクトリを指定します。省略時は、環境変数 PGDATA が有効となります。

-e *target_time*

指定した日時の時点にデータをリカバリする場合に指定します。

target_time

データを復旧する日時を指定します。指定形式は以下のとおりです。

"YYYY-MM-DD HH:MM:SS"

-l

pgx_dmpallコマンドで取得したバックアップ格納ディレクトリ内のバックアップデータの情報を一覧で表示します。pgx_dmpallコマンドにバックアップ用コピーコマンド(-Yオプション)を使用して実行した場合、コピーコマンドでバックアップした対象資源の一覧が追加で表示されます。-p、-eまたは-nオプションと同時に指定することはできません。

-n *restore_point*

指定したリストアポイントの時点にデータをリカバリする場合に指定します。リストアポイントはSQL関数pg_create_restore_pointにより作成します。同じ名前を持つ複数のリストアポイントを作成した場合、バックアップ取得以降の最初のリストアポイントがリカバリに使用されます。指定したリストアポイントが存在しない場合、リカバリが失敗します。-eおよび-pオプションと同時に指定することはできません。

-p

最後のバックアップが完了した時点にデータをリカバリする場合に指定します。-eおよび-nオプションと同時に指定することはできません。

-x

-eオプションに指定した時刻にコミットしたトランザクションをリカバリに含めない場合に指定します。

-Y *copy_command_file*
--copy-command=*copy_command_file*

リカバリ用コピーコマンドのファイルパス名を1024バイト未満の長さで指定します。本オプションは-lオプションと同時に指定することはできません。

--user-pin

キーストアをオープンするためのユーザPINの入力を要求します。本オプションは、ハードウェアベースのキーストアでのみ指定できます。

--keystore-passphrase

ファイルベースのキーストアを使用している場合に、キーストアをオープンするためのパスフレーズの入力を要求します。本オプションは、ファイルベースのキーストアでのみ指定できます。

--kms-secret

鍵管理システムをキーストアとして利用する場合に、キーストアをオープンするための秘密情報の入力を要求します。

--view-results-of-copying

pgx_dmpallコマンドで実行したコピーコマンドによって書き出されたバックアップ情報ファイルの内容を表示します。-l、-p、-e、-nまたは-Yオプションと同時に指定することはできません。

環境

PGDATA

データディレクトリを指定します。-Dオプションを使用して上書きすることができます。

PGPORT

データベースへ接続するためのポート番号を指定します。

PGUSER

データベースのスーパーユーザーのユーザー名を指定します。このコマンドを実行している実効ユーザーの名前がデフォルトです。

診断

0: 正常終了

0以外: 異常終了

バックアップデータの情報

Date

pgx_dmpallコマンドによりバックアップデータを作成した年月日

Dir

バックアップデータが格納されている、バックアップ格納ディレクトリ内のディレクトリ名

ディレクトリの命名形式: 日時形式（YYYY-MM-DD_HH-MM-SS）

Status

pgx_dmpallコマンドで取得したバックアップデータの状態

COMPLETE : 完了

INCOMPLETE : 未完了

Compression

pgx_dmpallで取得したバックアップデータの圧縮状態

USED: 圧縮されている

この情報は、圧縮されていない場合は表示されません。

Resources backed up by the copy command

pgx_dmpallコマンドでコピーコマンドによりバックアップした資源の一覧

コピーコマンドでバックアップした資源が存在する場合に、見出しおよび半角カンマ区切りでデータベースクラスタ('pg_data')またはテーブル空間名の一覧が表示されます。

注釈

このコマンドは、-lオプション指定での実行以外は、データベースサーバの停止時にのみ実行できます。

このコマンドは、PostgreSQLユーザーアカウントで実行してください。

リカバリに使用するバックアップデータは、リカバリの対象となるデータディレクトリから取得したものを使用してください。

このコマンドを実行する前に、データベースへ接続を行うすべてのアプリケーションを切断してください。また、リカバリ実行中はデータベースへの接続は行わないようにしてください。

設定ファイルは、pgx_dmpallコマンド(-cオプションを含む)で、最後に取得した設定ファイルの状態に復旧されます。

このコマンドでは、リカバリの終了判定の際に、データベースへ接続します。このため、複数のインスタンスが存在する環境では、リカバリ対象のインスタンスに接続するよう、環境変数PGPORTでポート番号を設定してください

pgx_rcvallとpgx_dmpallを実行するときのOSの時間帯設定を、postgresql.confのtimezoneパラメータで指定されている時間帯に一致させるようにしてください。さもないと、-eまたは-pオプションを指定した場合に、データが意図しない時刻に復旧されてしまうことがあります。

過去の時点に復旧した場合、その復旧時点を起点とする新たな時系列(データベース更新の歴史)が始まります。リカバリが完了したときには、その復旧時点が新たな時系列における最新地点です。以後、最新状態にリカバリする場合には、この新たな時系列上のデータベース更新が再実行されます。

有効なリストアポイントは、バックアップを取得した時系列上で作成したものです。つまり、過去の時点に復旧した場合、以降に設定したリストアポイントは利用できません。したがって、望みの過去データを復元できたら、バックアップを取得してください。

pgx_dmpallコマンドにバックアップ用コピーコマンド(-Yオプション)を使用して実行した場合、本コマンドは、リカバリ用コピーコマンド(-Yオプション)を使用して実行する必要があります。ただし、コピーコマンドでバックアップした対象資源(データベースクラスタまたはテーブル空間)はバックアップディレクトリで記録されているため、本コマンド実行時には対象資源を指定する必要はありません。コピーコマンドでバックアップを取得した対象資源については、-lオプションを使用することで確認できます。

例

以下は、データディレクトリ、テーブル空間、および設定ファイルを復旧する例です。

```
$ pgx_rcvall -B /home/pgsql/Backupdir
```

以下は、“2022年2月1日 10時0分0秒”の時点にデータディレクトリ、およびテーブル空間を復旧する例です。設定ファイルは、最後に取得した時点に復旧されます。

```
$ pgx_rcvall -B /home/pgsql/Backupdir -e "2022-02-01 10:00:00"
```

以下は、リストアポイント”before_match_20220210_1”の時点にデータディレクトリ、およびテーブル空間を復旧する例です。設定ファイルは、最後に取得した時点に復旧されます。

```
$ pgx_rcvall -B /home/pgsql/Backupdir -n before_match_20220210_1
```

以下は、バックアップ格納ディレクトリに取得されているバックアップデータの情報を一覧で表示させる例です。

```
$ pgx_rcvall -l
```

関連項目

pgx_dmpall

3.6 postgres

名前

postgres --データベースサーバ

概要

postgres [*option*...]

オプション

-K

キーストアをオープンするためのパスフレーズの入力を促します。

本オプションはシングルユーザーモードでのみ動作します。したがって、以下のように“--single”オプションも指定してください。

```
postgres --single -K
```

参照

その他の説明については、“PostgreSQL Documentation”の“Reference”の“postgres”を参照してください。

第4章 Mirroring Controllerコマンド

Mirroring Controllerコマンドについて説明します。

4.1 mc_arb

名前

mc_arb - Mirroring Controller裁定プロセスの起動、停止、および状態表示

概要

mc_arb start [-M *mcdir*] [-w| -W]

mc_arb stop [-M *mcdir*] [-e]

mc_arb status [-M *mcdir*]

説明

mc_arbはMirroring Controller裁定プロセスの起動、停止、および状態表示を行うコマンドです。

startモードは、新たにMirroring Controller裁定プロセスを起動します。

stopモードは、Mirroring Controller裁定プロセスを停止します。

statusモードは、Mirroring Controller裁定プロセスとプライマリサーバおよびスタンバイサーバのMirroring Controllerプロセスとの接続状態を表示します。

コマンドを実行するサーバでMirroring Controller裁定プロセスが起動していない場合、stopモード、statusモードはエラーで終了します。

また、データベースサーバでMirroring Controllerを強制停止した場合、statusモードのサーバの接続状態の表示が、offlineになるまでに時間がかかる場合があります。

このコマンドは、任意のOSユーザーで実行可能です。

オプション

-e

実行しているサーバ上のMirroring Controller裁定プロセスを強制停止する場合に指定します。

Mirroring Controller裁定プロセスを停止させる場合、事前にデータベースサーバのMirroring Controllerを停止しておく必要がありますが、Mirroring Controllerが起動した状態でMirroring Controller裁定プロセスを停止させる場合に、本オプションを指定します。また、裁定処理で動作するフェンシングコマンドが復帰しないなどの原因で裁定処理が長期化する場合でも、本オプションによりMirroring Controller裁定プロセスを停止することで、裁定処理を中断させることが可能です。

-M *mcdir*

Mirroring Controller裁定プロセスの管理ディレクトリを指定します。

ディレクトリパスにはASCII文字が指定可能です。

これが省略された場合、ARBCONTROLDIR環境変数が使われます。

-w

操作の完了を待ち合わせします。

本オプションはstartモードのデフォルトです。

-W

操作の完了を待ち合わせません。

環境

ARBCONTROLDIR

Mirroring Controller裁定プロセスの管理ディレクトリを指定します。

ディレクトリパスにはASCII文字が指定可能です。

-Mオプションを指定して上書きすることができます。

診断

0: 正常終了

0以外: 異常終了

注釈

Mirroring Controller裁定プロセスの停止処理中に、コマンドプロセスを強制停止した場合やコマンドプロセスとMirroring Controller裁定プロセス間の通信が途絶した場合に、他にmc_arbコマンドを実行中でないにもかかわらず、実行中のメッセージを出力して停止処理が異常終了する場合があります。
この問題を解決するには、他にmc_arbコマンドが実行されていないことを確認したあと、Mirroring Controller裁定プロセスを強制停止してください。

例

Mirroring Controller裁定プロセスを起動します。

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

mc_arb statusの表示内容

```
server_id   host            status
(1)         (2)             (3)
```

(1) サーバ識別子
(2) ホスト名またはIPアドレス
(3) サーバの接続状態
online　：接続状態
offline　：切断状態

4.2 mc_ctl

名前

mc_ctl - Mirroring Controllerの起動、停止、サーバの状態表示、切り替え、および切り離し

概要

mc_ctl start [-M *mcdir*] [-w | -W] [-f | -F] [--mc-only] [--async-connect-arbiter] [--local-server *server_id*]

mc_ctl stop [-M *mcdir*] [[-a] [--mc-only] | [-e]] [--local-server *server_id*]

mc_ctl status [-M *mcdir*] [--arbiter] [--local-server *server_id*]

mc_ctl switch [-M *mcdir*] [--force [--no-fencing]] [--local-server *server_id*]

mc_ctl detach [-M *mcdir*] [--no-fencing] [--local-server *server_id*]

mc_ctl enable-failover [-M *mcdir*] [--local-server *server_id*]

mc_ctl disable-failover [-M *mcdir*] [--local-server *server_id*]

説明

mc_ctlはMirroring Controllerの起動、停止、サーバの状態表示、切り替え、および切り離しを行うコマンドです。

startモードは、新たにMirroring Controllerを起動します。--mc-onlyオプションを指定しない場合はデータベースのインスタンスを起動します。

stopモードは、Mirroring Controllerを停止します。--mc-onlyオプションを指定しない場合はデータベースのインスタンスを停止します。スタンバイサーバ上で--mc-onlyオプションを指定せずに実行した場合、プライマリサーバからスタンバイサーバを切り離します。

statusモードは、Mirroring Controllerが監視するサーバ、データベースのインスタンスのプロセスおよびディスクの状態を表示します。また、--arbiterオプションを指定している場合には、Mirroring Controller裁定プロセスとの接続状態を表示します。

switchモードは、プライマリサーバの切り替えを行います。切り替えを行うと、プライマリサーバのデータベースのインスタンスが停止し、スタンバイサーバのデータベースのインスタンスがプライマリサーバに昇格して縮退運転になります。本モードは、Mirroring Controllerプロセスが相手サーバのMirroring Controllerプロセスと通信ができる環境において、プライマリサーバ、スタンバイサーバで実行可能です。

detachモードは、スタンバイサーバの強制切り離しを行います。本モードは、スタンバイサーバへのログインが必要なstopモードによるMirroring Controllerの停止が実行できない場合に、相手サーバを強制的に切り離す場合に使用します。プライマリサーバでのみ実行可能です。

enable-failoverモードは、自動切り替え/切り離しの有効化を行います。

disable-failoverモードは、自動切り替え/切り離しの無効化を行います。

コマンドを実行するサーバでMirroring Controllerが起動していない場合、startモード、およびstatusモード以外のコマンドはエラーで終了します。

このコマンドは、インスタンス管理者ユーザーで実行してください。

プライマリサーバのMirroring Controllerを起動後、スタンバイサーバのMirroring Controllerが起動されるまでの間、プライマリサーバのみでも運用を開始できますが、スタンバイサーバのMirroring Controllerを起動するとスタンバイサーバが組み込まれ、多重化構成での運用が行えるようになります。

オプション

-a

全サーバのMirroring Controllerを停止する場合に指定します。

-e

実行しているサーバ上のMirroring Controllerを強制停止する場合に指定します。

-f

Mirroring Controller起動直後に自動切り替え/切り離しの有効化を行う場合に指定します。

本オプションはstartモードのデフォルトです。

-F

Mirroring Controller起動直後に自動切り替え/切り離しの有効化を行わない場合に指定します。

--async-connect-arbiter

Mirroring Controllerの起動処理とは非同期でMirroring Controller裁定プロセスに接続する場合に指定します。Mirroring Controller裁定プロセスが起動していない場合に、Mirroring Controllerを強制的に起動したいときに指定してください。

このオプションはMirroring Controllerの裁定サーバを利用する場合に指定します。

--arbiter

Mirroring Controller裁定プロセスとの接続状態を表示する場合に指定します。statusモードの場合に指定可能です。

このオプションはMirroring Controllerの裁定サーバを利用する場合に指定します。

--local-server *server_id*

システムテストなどの目的のため、プライマリサーバとスタンバイサーバを擬似的に単一のサーバ内に構築した場合に、操作対象のサーバを特定するために指定します。

server_idには、network.confファイルに指定したサーバ識別子を指定します。半角空白以外のASCII文字が指定可能です。指定したサーバ識別子のサーバに対してログインして操作をしているものと見なして動作します。

--mc-only

Mirroring Controllerのプロセスのみを起動、停止する場合に指定します。startモードでは、データベースのインスタンスが起動中の場合にのみ指定可能です。本オプションを省略した場合、データベースのインスタンスも同時に起動、停止します。

-M *mcdir*

Mirroring Controller管理ディレクトリを指定します。

ディレクトリパスにはASCII文字が指定可能です。

これが省略された場合、MCCONTROLDIR環境変数が使われます。

--force

本オプションを指定した切り替えは、スタンバイサーバでのみ指定可能です。データベースサーバ間のネットワーク異常やサーバの無応答などが原因で、相手サーバのMirroring Controllerプロセスとの通信ができず、通常の切り替えができない場合に、プライマリサーバをフェンシングしてから強制的に切り替えを行います。ただし、--no-fencingオプションを指定した場合、プライマリサーバをフェンシングしません。

--no-fencing

--forceオプションを指定した切り替え、または切り離しの実行時に、縮退させるサーバのフェンシングを迂回します。そのため、縮退させるサーバを利用者が事前にクラスタシステムから隔離しておく必要があります。

-w

操作の完了を待ち合わせします。

本オプションはstartモードのデフォルトです。

-W

操作の完了を待ち合わせません。

環境

MCCONTROLDIR

Mirroring Controller管理ディレクトリを指定します。

ディレクトリパスにはASCII文字が指定可能です。

-Mオプションを指定して上書きすることができます。

診断

0: 正常終了

0以外: 異常終了

注釈

このコマンドの実行、自動切り替え、および自動切り離しの処理中にサーバがダウンした場合や、プライマリサーバとスタンバイサーバの間の通信が途絶した場合に、他にmc_ctlコマンドを実行中でないにもかかわらず、実行中のメッセージを出力して異常終了する場合があります。この問題を解決するには、他にmc_ctlコマンドが実行されていないことを確認したあと、Mirroring Controllerを再起動してください。その後、必要な操作を実施してください。

このコマンドの実行中に通信のタイムアウトエラーが発生し、メッセージの表示内容とMirroring Controllerのプロセスまたはインスタンスのプロセスの状態が異なる場合があります。必要に応じてメッセージに記載されている“利用者の処置”にしたがって対処を実施してください。

enable-failoverモード、disable-failoverモード、および、startモードによる自動切り替え/切り離しの有効化/無効化は、Mirroring Controller起動中にのみ有効となります。このため、起動時点で自動切り替え/切り離しを有効化/無効化する場合は、Mirroring Controllerを起動する度に、-fオプション、または、-Fオプションを指定してください。

このコマンドを実行した時点で、postgresql.confファイルに記述誤りがある場合、コマンドが異常終了する場合があります。その場合には、postgresql.confファイルを修正してから、コマンドを再実行してください。

Mirroring Controllerの裁定サーバを利用する場合には、startモードによる起動を-Fオプションを指定して実行、または--local-serverを指定して実行しても、起動時には裁定サーバとの接続を行います。

裁定サーバを利用する運用の場合、裁定サーバとの接続を試行する時間が加算されるため、Mirroring Controllerの起動時間が長くかかる場合があります。

例

Mirroring Controllerを起動します。

```
$ mc_ctl start -M /mcdir/inst1
```

mc_ctl statusの表示内容

```
mirroring status
----------------
(1)

server_id   host_role          host            host_status  db_proc_status         disk_status
(2)         (3)                (4)             (5)          (6)                    (7)
```

(1) 多重化状態
- switchable : 切替可能
- switching : 切替途中
- switched : 切替済（切り替えが終わって縮退運用の状態になった場合に表示されます）
- not-switchable : 切替不可（切り離しが行われて切り替えができない場合に表示されます）
- unknown : 不明(注1)
- failover-disabled : フェイルオーバが無効化されている状態

(2) サーバ識別子

(3) サーバの役割
- primary : プライマリ
- standby : スタンバイ
- none(inactivated primary): 役割なし
  (停止中のプライマリ、またはプライマリとして定義中)
- none(inactivated standby): 役割なし
  (停止中のスタンバイ、またはプライマリとして定義中)

(4) ホスト名またはIPアドレス

(5) サーバの生死状態
- normal : 正常稼動
- abnormal : 異常
- unknown : 不明(注1)

(6) DBMSプロセスの状態
- normal : 正常
- abnormal(異常プロセス名(注2)) : 異常
- unknown : 不明(注1)

(7) ディスクの状態
- normal : 正常
- abnormal(異常ディスクの種類(注3)) : 異常
- unknown : 不明(注1)

注1) Mirroring Controllerが停止状態，管理用ネットワーク異常，Mirroring Controllerのダウン/無応答が発生している場合に表示されます。

注2) 異常を検知したDBMSプロセスが出力されます。意味は以下のとおりです。ただし、複数のDBMSプロセスの異常を検知した場合は、最初に異常を検出したDBMSプロセスのみを表示します。
- postmaster
  アプリケーション接続を受け付けるプロセス(postmaster)の異常を検知したことを意味します。
- wal_senderまたはwal_receiver
  トランザクションログを送受信するプロセス(wal sender、またはwal receiver)の異常を検知したことを意味します。

注3) 異常を検知したディスクの種類がコンマ区切りで出力されます。意味は以下のとおりです。
- data
  データ格納ディスクの異常を検知したことを意味します。
- tran_log
  トランザクションログ格納ディスクの異常を検知したことを意味します。

・tablespace
テーブル空間格納ディスクの異常を検知したことを意味します。

mc_ctl status(--arbiterオプション指定時)の表示内容

```
arbiter_id   host            status
(1)          (2)              (3)
```

(1) 裁定サーバ識別子
(2) ホスト名またはIPアドレス
(3) Mirroring Controller裁定プロセスとの接続状態
online ： 接続状態
offline ： 切断状態(注)

注) Mirroring Controller停止時は、Mirroring Controller裁定プロセスとの接続が行われていないため、offlineが表示されます。

第5章 Connection Managerコマンド

Connection Managerコマンドについて説明します。

5.1 cm_ctl

名前

cm_ctl - conmgrプロセスの起動、停止および状態表示

概要

```
cm_ctl start [-D diretory] [-W] [--complete] [-t seconds]

cm_ctl stop [-D diretory] [-W] [-m {smart | fast | immediate}] [-t seconds]

cm_ctl status [-D directory] [-t seconds] [-i {all | instance | application }]
```

説明

startモードは、conmgrプロセスを起動します。少なくともプライマリのインスタンスとの間で生死監視の接続が完了するまで待ってからコマンドが復帰します。--completeを指定した場合には、conmgr.confに定義されたすべてのインスタンスとの間に生死監視の接続が完了するまで待ち合わせます。これらの待ち合わせにはタイムアウトを設定できます。タイムアウトのデフォルトは60秒です。-tオプションで変更することができます。タイムアウトした場合は、単に待ち合わせを諦めるだけであり、conmgrプロセスは起動状態のままです。

注意

conmgr.confに設定したインスタンスの中に、プライマリが含まれない場合には、cm_ctlコマンドでConnection Managerを起動するときに、-Wオプションを付けてください。-Wオプションを付けないと、プライマリとの接続が完了できるまでcm_ctlコマンドが復帰しないからです。

stopモードは、conmgrプロセスにシグナルを送信してシャットダウンし、プロセスが消滅するまで待ち合わせます。待機時間のデフォルトは60秒です。-tオプションで変更することができます。タイムアウトした場合は、単に待ち合わせを諦めるだけです。
シャットダウンの方法には、smart、fastおよびimmediateの3通りがあり、-mオプションで指定します。smartは、conmgrプロセスを使用しているすべてアプリケーションのSQL接続がなくなるまで待ち合わせてからシャットダウンします。fastは、conmgrプロセスを使用しているすべてのアプリケーションとconmgrプロセスとの接続を強制的に切断してからシャットダウンします。結果としてアプリケーションのSQL接続はエラーを受け取ります。immediateは、ただちにconmgrプロセスを終了します。何も指定しなければ、fastモードで停止します。

statusモードは、conmgrプロセスが存在していれば、conmgrプロセスにconmgrプロセスが認識しているインスタンスとアプリケーションの情報を問い合わせて、conmgrプロセス自身の状態とともにそれらを標準出力に出力します。-iオプションによってどのような情報を問い合わせるかを指定できます。
instanceはインスタンスの情報を問い合わせます。applicationは、アプリケーションの情報を問い合わせます。

allは、インスタンスとアプリケーションの情報を問い合わせます。

conmgrプロセス自身の情報は常に出力されます。問い合わせの応答が返るまでの待機時間のデフォルトは60秒です。-tオプションで変更することができます。

オプション

--complete

conmgr.confに設定されたすべてのインスタンスとの間に生死監視の接続が完了するまで待ちます。-Wオプションと同時に使用した場合には、-Wオプションが優先されます。

-D *directory*

--directory=*directory*

conmgr.confが配置されたディレクトリを指定します。省略した場合には環境変数CMDATAに指定されたディレクトリを参照します。両方を省略することはできません。

-i {all | instance | application }

状態を表示する情報を指定します。

-m

--mode={smart | fast | immediate}

シャットダウンのモードを指定します。デフォルトはfastです。

-t *seconds*

--timeout=*seconds*

操作が完了するまでの待機時間を指定します。単位は秒です。デフォルトは60秒です。

-W

--no-wait

startモードで指定すると、cm_ctlコマンドは、conmgrプロセスをforkした直後に復帰します。stopモードで指定すると、プロセスの消滅を待たずにcm_ctlコマンドは復帰します。

診断

startモード/stopモードの場合

0: 正常終了

2: タイムアウトが発生

3: 指定したディレクトリへのアクセス不可

上記以外: 上記のいずれでもない

statusモードの場合

0: 正常終了

3: 指定したディレクトリへのアクセス不可

4: conmgrプロセスが存在しない

上記以外: 上記のいずれでもない

権限

管理者(例えばLinuxのスーパーユーザー(root))はconmgrプロセスを起動することはできません。

例

statusモードの出力例

-iオプションに指定できる情報ブロックを1つの単位とします。
ブロック間には空行が1行あり、ブロック内には空行がありません。
カラム間や、データ間には1つ以上の空白を含みます。

```
$ cm_ctl status -i all
conmgr_status:
status pid
(1)    (2)
ready  3456
```

```
instance_information:
addr          port  database_attr
(3)           (4)   (5)
192.0.2.100   27500 standby
192.0.2.110   27500 primary
192.0.2.120   27500 standby
192.0.2.130   27500 unknown

application_information:
addr          port  pid   connected_time
(6)           (7)   (8)        (9)
127.0.0.1     12345 5678      2022-02-15 02:03:04
```

(1) conmgrプロセスの状態
　starting : conmgrプロセスは起動シーケンスを開始していますが、
　　クライアントからの接続を受け付ける準備ができていません。
　ready　 : クライアントからの接続を受け付ける準備ができています。
　stopping : 終了指示を受け取り、停止シーケンスを開始しています。
　inactive : conmgrプロセスが存在しません。
(2) conmgrプロセスのPID
(3) インスタンスのホスト名またはIPアドレス
(4) postmasterが待ち受けるポート番号
(5) インスタンスの状態（primary|primary(read-only)|standby|unknown）
　primary　　: プライマリサーバ
　primary(read-only) : プライマリサーバ（デフォルトのトランザクションモードが読み取り専用）
　standby　　: スタンバイサーバ
　unknown　　: 不明(注)

(6) conmgrプロセスに対する接続元のIPアドレス
(7) conmgrプロセスに対する接続元の（エフェメラル）ポート番号
(8) 接続元のPID
(9) conmgrプロセスに接続した日付と時刻
　ISO8601にも準拠した年月日の後に空白が1つあり、その後にISO8601に準拠した秒精度の時刻が続きます。
　この表現は、PostgreSQLのtimestamp型の文字列表現の1つです。
注）インスタンスに接続できない場合に表示されます。