



EDB

Postgres Workload Report

リリース 1.1.0

EnterpriseDB Corporation.

2024 年 05 月 23 日

目次

第 1 章	Release notes	3
1.1	Postgres Workload Report 1.0.0 release notes	3
1.2	Postgres Workload Report 1.1.0 release notes	4
第 2 章	Installing Postgres Workload Report	5
第 3 章	Configuring Postgres Workload Report	7
3.1	構成ファイルの場所	7
3.2	構成ファイルオプション	8
3.2.1	input_dir	8
3.2.2	Output_dir	8
3.2.3	Report_name	8
3.2.4	log_file	8
3.2.5	log_level	8
3.2.6	log_format	9
3.2.7	attempts_dir	9
第 4 章	Using Postgres Workload Report	11
4.1	前提条件	11
4.2	レポートのソース情報	11
4.3	execute オプションの例	11
4.4	report オプションの例	12

Postgres Workload ReportPWR は、HTML、Markdown、DOCX、および PDF モードで PostgreSQL ワークロードレポートを構築するために使用される Python ベースのツールです。これらのレポートは、Oracle の自動ワークロードリポジトリ AWR レポートツールが提供するレポートを模倣しています。

Postgres 接続を使用すると、レポートが必要なサーバーにアクセスできる任意のマシンから Postgres ワークロードレポートを実行できます。Postgres ワークロードレポートはデータの収集に Lasso を使用するため、Lasso が前提条件です。edb-pwr パッケージは、インストールされる edb-lasso に明示的な依存関係があります。

Postgres サーバーでは、edb_wait_states バージョン 1.2 以降をインストールおよびロードする必要があります。また、Postgres ワークロードレポートで edb_wait_states ファンクションを使用して、保存されたスナップショットデータを収集できるように、できればメインデータベースに拡張機能を作成する必要があります。

第1章 Release notes

Postgres Workload ReportPWR ドキュメントでは、マイナーリリースとパッチを含む PWR 1 の最新バージョンについて説明しています。リリースノートは、各リリースの新機能に関する情報を提供します。マイナーまたはパッチリリースで導入された新機能の場合、コンテンツのインジケータは、機能を導入したリリースに関する情報を提供します。

Version	Release date
<i>Postgres Workload Report 1.1.0 release notes</i>	15 May 2024
<i>Postgres Workload Report 1.0.0 release notes</i>	15 Feb 2024

1.1 Postgres Workload Report 1.0.0 release notes

リリース日 2024 年 2 月 15 日

Postgres Workload Report 1.0.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Feature	<i>pwr execute</i> CLI コマンドを追加しました。このコマンドは、ワークロードレポートを生成するために使用されます。`pwr report` コマンドでは、Lasso を実行し、そのデータを入力として提供する必要がありますが、`pwr execute` はエンドツーエンドの実行、つまり Lasso と `pwr report` を暗黙的に実行します。`pwr execute` コマンドには、edb-lasso >=4.13.0 および edb_wait_states >=1.2.0 が必要です。
Feature	<i>pwr report</i> CLI コマンドを追加しました。`pwr report` コマンドは、ワークロードレポートを生成するために使用されます。PWR ワークロードレポートは、HTML、Markdown、PDF、および DOCX の 1 つ以上の形式で書くことができます。

1.2 Postgres Workload Report 1.1.0 release notes

リリース日 2024 年 5 月 15 日

Postgres Workload Report 1.1.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Feature	同時実行を回避するために PWR コマンドにロックを追加しました。
Change	パッケージがアセットを配置するフォルダー <code>/usr/share/pwr/assets</code> を使用するように <code>`assets_dir`</code> 構成パラメーターのデフォルト値を変更しました。
Change	<code>execute`および`report`コマンドに<code>--use-plain-report-name`</code> オプションを追加しました。このオプションは、データベース名とレポートタイムスタンプをレポート名に追加するデフォルトの動作をバイパスします。</code>
Change	PWR 出力とロギングを修正しました。
Bug Fix	<code>`edb-lasso`</code> リリース 4.13.0 以降に厳密に依存するように RPM パッケージを修正しました。
Bug Fix	PWR を修正して、 <code>`edb_wait_states`</code> の範囲内にサンプルがない場合に失敗しないようにしました。
Bug Fix	タイムゾーンなしでタイムスタンプを使用する場合の問題を修正しました。
Bug Fix	構成処理に関連する問題を修正しました。
Bug Fix	<code>`--output-directory`</code> が存在しない場合に <code>`pwr report`</code> が失敗する問題を修正しました。

第2章 Installing Postgres Workload Report

Postgres ワークロードレポートは、サポートされているすべてのオペレーティングシステムの Python ソース配布パッケージとして提供されます。これは、**enterprise** および **standard** リポジトリにあります。

次のコマンドは、Postgres Workload Report をインストールし、必要に応じて、リポジトリから **edb-lasso** パッケージを取得してインストールします。

```
sudo <package-manager> -y install edb-pwr
```

<package-manager> は、オペレーティングシステムで使用されるパッケージマネージャーです。

Package manager	Operating system
apt	Debian / Ubuntu
dnf	RHEL 8/9 and derivatives
yum	RHEL 7 and derivatives, CentOS 7
zypper	SLES

第3章 Configuring Postgres Workload Report

`pwr` の実行時に必要なコマンドライン引数の数を減らすには、構成ファイルを使用して、常に同じ値を持ち、値がデフォルトとは異なるオプションを指定できます。

注釈: `edb-lasso` が正常に動作するには、構成ファイルが存在する必要があります。正しく構成されていない場合、`pwr` は完了できません。構成方法の詳細については、[Lasso documentation](#) を参照してください。

3.1 構成ファイルの場所

Postgres ワークロードレポートは、次の場所で構成ファイルを検索し、最初に見つかったものを使用します。

1. `--config` コマンドラインオプションで名前付けたファイルが指定されている場合。
2. `PWR_CONFIG_FILE` 環境変数で名前付けられたファイル設定されている場合。
3. `~/.pwr.conf`
4. `/etc/pwr.conf`

インストールパッケージは、`/etc/pwr.conf.template` に構成ファイルのテンプレートを作成します。このファイルを Postgres ワークロードレポートがデフォルトで構成ファイルを検索する2つの場所 `~/.pwr.conf` および `/etc/pwr.conf` のいずれかにコピーし、必要に応じてテンプレートのオプションを編集することをお勧めします。

注釈: 構成ファイルが見つからない場合、Postgres ワークロードレポートはすべてのオプションのデフォルト値を想定していますが、対応するコマンドラインオプションを使用してオーバーライドできます。コマンドラインオプションの使用の詳細については、[Using Postgres Workload Report](#) を参照してください。

注釈: 構成ファイルの拡張子は `.conf` ですが、そのコンテンツは **valid** YAML 形式である必要があります。

3.2 構成ファイルオプション

3.2.1 input_dir

Lasso レポートの `edb_wait_states` 部分が存在する既存のディレクトリ。このオプションは、主に `pwr` report 実行に使用されます *Using Postgres Workload Report* を参照してください。

3.2.2 Output_dir

Postgres Workload Report がレポートファイルを書き込むディレクトリの場所。`pwr` を実行すると、このディレクトリが存在しない場合は作成されます。

3.2.3 Report_name

生成されたレポートファイルの名前。通常、レポートごとに名前が異なるため、コマンドラインでこのオプションを指定します。

ファイル拡張子を含めないでください。コマンドラインで指定された出力形式に基づいて、適切な拡張子が追加されます。つまり、`--pdf` は `.pdf` を追加し、`--html` は `.html` を追加します。

3.2.4 log_file

Postgres ワークロードレポートが `stdout` および `stderr` ログを書き込むファイルへのフルパス。

3.2.5 log_level

Postgres ワークロードレポートを実行するときに使用するログレベル。以下は有効な値であり、より冗長なものからより低いものまでリストされています。

- DEBUG
- INFO 指定しない場合のデフォルト
- WARNING
- ERROR
- CRITICAL

ログレベルの詳細については、

[the Python logging](#) ドキュメントを参照してください。

3.2.6 log_format

ログファイルに書き込まれるログメッセージの形式。ログのフォーマットの詳細については、
[the Python logging](#) ドキュメントを参照してください。

3.2.7 attempts_dir

HTML 出力のフォーマットに使用される Jinja テンプレートと、PDF 出力に使用される CSS が含まれるディレクトリ。デフォルト値は `/usr/share/pwr/assets` で、これには、`edb-pwr` パッケージで提供されるアセットが含まれています。

第4章 Using Postgres Workload Report

4.1 前提条件

Postgres ワークロードレポートは、`edb_wait_states` 拡張機能バージョン 1.2 以降がロードされている Postgres サーバーのレポートを提供できます。さらに、PWR は、`edb_wait_states` 拡張機能がサーバーにロードされた時間間隔のみのクエリ待機レポートを提供できます。

詳細については、`edb_wait_states` を参照してください。

4.2 レポートのソース情報

`edb-pwr` および `edb-lasso` パッケージがマシンにインストールされ、`edb_wait_states` 拡張機能が十分な期間ロードされた状態でサーバーが実行された後、Postgres Workload Report を使用して、実行中に実行されていたクエリの待機状態のレポートを抽出できます。指定された時間間隔。

あるいは、Postgres ワークロードレポートは、`edb_wait_states` がロードされたサーバーでレポートが実行されたと仮定して、既存の Lasso レポートからレポートを生成できます。このため、Postgres ワークロードレポートには必須の第 1 引数があり、次のいずれかです。

- **execute** —エンドツーエンドの実行を実行します。適切なオプションを使用して `lasso` を呼び出し、生成された tarball レポートを入力として使用して、HTML、Markdown、DOCX、または PDF レポートを生成します。
- **report** —保存された `lasso` レポートコンテンツのディレクトリを、待機状態レポートを処理および生成するための入力として使用します。**report** 引数を使用する前に、解凍および解凍された `lasso` レポートが既に存在する必要があります。

4.3 execute オプションの例

この例では、1 月 10 日 9:00 から 1 月 10 日 13:00 までの間隔での Postgres サーバー `myserver` のクエリ待機に関するレポートを生成します。その頃、根本的な原因を提供するために調査する必要があるインシデントが発生しました。`edb_wait_states` がインストールされるサーバーのメインデータベースは `my-oltp` です。

HTML 形式でレポートを取得するには、次のコマンドを使用します。

```
pwr execute --host-name myserver --sampling-start 2024-01-10 09:00:00+00:00 \  
--sampling-end 2024-01-10 13:00:00+00:00 --html \  
--report-name Jan10_incident my-oltp postgres
```

注釈: `--sampling-start` および `--sampling-end` は、タイムゾーンの有無にかかわらずタイムスタンプを受け入れます。タイムスタンプでタイムゾーンが明示的に設定されていない場合、PWR はシステムのタイムゾーンを使用します。

`pwr execute -h` を実行して、使用可能なオプションの完全なリストを取得します。

4.4 report オプションの例

注釈: 以前のレポートを上書きするのを避けるため、`pwr execute` コマンドは、`%Y%m%d%H%M%S` 形式で提供されたレポート名にデータベース名とタイムスタンプを追加します `March_report-mydatabase-20240321-133727.html` など。この動作を回避するには、`pwr execute` を実行するときに `--use-plain-report-name` オプションを使用します。

場合によっては、既に Lasso レポートがあり、PWR で Lasso レポートをソースとして使用し、それに基づいてレポートを構築したい場合があります。これらの場合、`pwr report` オプションは役立ちます。

この例では、既に説明したのと同じシナリオを使用しますが、前の例の時間境界を使用して実行されたような Lasso レポートを使用します。Lasso レポートの名前が `edb-lasso-Jan10-incident.tar.bz2` で、`pwr report` が実行されるマシンのホームディレクトリにあるとします。

次のコマンドは、`~/pwr_output/Jan10_incident.html` に保存される HTML レポートを生成します。

```
cd ~/
mkdir -p pwr_tmp/
tar jxf ../edb-lasso-Jan10-incident.tar.bz2 -C ~/pwr_tmp/ --strip-components=1
pwr report \
  --input-dir ~/pwr_tmp/postgresql/dbs/my-oltp/edb_wait_states/ \
  --html --output-dir ~/pwr_output/ --report-name Jan10_incident
```

`pwr report -h` を実行して、使用可能なオプションの完全なリストを取得します。