

なべ's blog

2015-12-18

SQLiteのジャーナルモード

データベース @SQLite /IT系

2つのSQLiteのデータベースファイルがあり、それぞれ更新するとは一方は「-journal」ファイルが作成されて、もう一方は「-shm」と「-wal」ファイルが作成されるという違うファイルが作成される理由がわからず調べたら、ジャーナルモードが違ったというお話。

ジャーナルモードの大体

SQLiteのジャーナルモードは大きく分けて「delete」と「wal」がある。
「delete」の時は「-journal」ファイルが作成され、「wal」の時は「-shm」と「-wal」ファイルが作成される。
例えば「foo」というデータベースファイルがあると、ジャーナルモードが「delete」の時は「foo」というファイルの他に「foo-journal」というファイルが作成され、ジャーナルモードが「wal」の時は「foo-shm」と「foo-wal」というファイルが作成される。どちらもジャーナルファイルなので、本体のデータベースにコミットするまでのデータが書き込まれるという意味では同じだが、動作は若干異なる。

ジャーナルモードの詳細は本家のサイトを参照。
http://www.sqlite.org/pragma.html#pragma_journal_mode

deleteモード

トランザクション開始から終了までの更新内容を「-journal」ファイルに書き込み、コミットしたときに本体のデータベースファイルへ更新内容を反映し、反映したら「-journal」ファイルを削除する。
最後の「-journal」ファイルを削除する動作に因んで「deleteモード」と呼んでいる。
コミットした時点でデータベースファイルを更新するため、大きなデータベースファイルの場合は書き込みに時間がかかるかもしれない。（未検証）

walモード

トランザクション開始から終了までの更新内容を「-shm」ファイルに書き込み、コミットした時に「-wal」ファイルへ更新内容を書き込む。
コミットした時点では本体データベースファイルには更新内容を書き込まないため、本体に比べて小さいファイルの更新でトランザクションを完了できるメリットがある。
なのでデータ更新が多いシステムではwalモードの方が有利と言える。（未検証）
ただしデータが本体データベースファイルとwalファイルに分散されるため、データの取得には時間がかかるかもしれない。（未検証）

walファイルはデータベース接続を閉じる時に、その内容を本体データベースファイルに反映する。
つまりクローズ処理はdeleteモードに比べて時間がかかる。

永続性の違い

先の通り、deleteモードはコミットした時点で本体データベースに更新を反映する一方、walモードではコミットしても本体データベースに更新を反映しない。
そのため、もしコミット後にシステムが異常終了等した場合、walモードでは本体データベースに更新を反映するためのリカバリが必要になる。

その他のジャーナルモード

上の2つ以外にも以下のようなジャーナルモードがある。

truncateモード

deleteモードの亜種で、コミットしても「-journal」ファイルを削除せず、0バイトのファイルを残しておく。
ファイル作成時の権限等のリスクを減らすには有利かもしれない。
データベースを閉じて再度接続するとdeleteモードになる。

persistモード

deleteモードの亜種で、コミットしても「-journal」ファイルの内容を削除せずに取っておく。
データベースを閉じて再度接続するとdeleteモードになる。

memoryモード

ジャーナルをファイルではなくメモリで管理する。
とにかく動作を速くしたい場合は選択する価値があるかもしれないが、メモリの容量に注意が必要かもしれない。
データベースを閉じて再度接続するとdeleteモードになる。

offモード

ジャーナリングしない・・・ということか？（未検証）

まとめ

deleteモードがwalモードはそれぞれメリット・デメリットがあるので、システムの特性に合わせて適切に使い分けましょう。

nave_kazu 4年前



8 B!ブックマーク 1 ツイート シェア



関連記事

- 2018-10-12

[Gitのコマンドを使ってみる（その1）](#)

Gitクライアントが重くて仕方がないので、そろそろちゃんとか？コ...
- 2017-10-26

[Djangoを使ってみる（準備）](#)

Djangoとは DjangoはPythonで実装されたWebアプリケーションフ...

	<div>2015-05-13</div> <div>GitBucket 3.2 を試す</div> <div>「Scalaで実装されたオープンソースのGitHubクローン」のGitBuc...</div>
	<div>2015-04-03</div> <div>Spring Boot/第四回 Spring Bootで REST</div> <div>Spring Bootを使うと、RESTの実装が簡単に出来るので試してみる...</div>
	<div>2015-03-26</div> <div>Spring Boot/第三回 Spring Bootでデータベース操作（JPA編）</div> <div>Spring Bootでデータベース操作をする。Spring Bootでのデータ...</div>

屋台ブルー (id:yataiblu)
この記事、めちゃ参考になりました！
ありがとうございました！

4年前

* * *

« Windows10にアップデートされないよう非表示...
DAG (Directed acyclic graph) - 有向非巡回... »

プロフィール



id:nave_kazu

読者になる 4

検索

記事を検索

リンク

- [はてなブログ](#)
- [ブログをはじめる](#)
- [週刊はてなブログ](#)
- [はてなブログPro](#)

最新記事

- [Gitのコマンドを使ってみる（その1）](#)
- [Djangoを使ってみる（アプリケーションへのルーティング）](#)
- [Djangoを使ってみる（準備）](#)
- [JJUG CGC 2017 Spring に参加してきた](#)
- [AnacondaでPython（データ解析の真似事）](#)

月別アーカイブ

- [2018 \(1\)](#)
- [2017 \(6\)](#)
- [2016 \(10\)](#)
- [2015 \(22\)](#)
 - [2015 / 12 \(1\)](#)
 - [2015 / 11 \(1\)](#)
 - [2015 / 7 \(2\)](#)
 - [2015 / 6 \(1\)](#)
 - [2015 / 5 \(1\)](#)
 - [2015 / 4 \(4\)](#)
 - [2015 / 3 \(4\)](#)
 - [2015 / 2 \(2\)](#)
 - [2015 / 1 \(6\)](#)

はてなブログをはじめよう！

nave_kazuさんは、はてなブログを使っています。あなたもはてなブログをはじめてみませんか？

はてなブログをはじめ（無料）

はてなブログとは