

# PostgreSQL運用管理トレーニング

## 8章 チューニングの実施

SRA OSS, Inc.

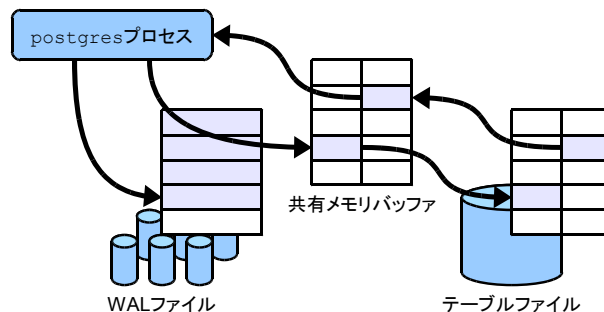
<http://www.sraoss.co.jp/>





# 読み書きのチューニング

- データの読み取り・書き込みはデータベースにおける最も基本的な処理
  - 検索 共有メモリバッファにキャッシュしてテーブルファイルから読み取る
  - 更新 WALファイルと共有メモリファイルに書き込み、チェックポイント時にテーブルファイルに書き込む
- 確実にチューニングによる効果がある
  - 10倍、100倍の性能改善は望めない



PostgreSQL運用管理トレーニング(8 チューニングの実施)  
Copyright © SRA OSS, Inc.

9

## 読み書きのチューニング

本節ではパラメータ設定によるデータベース読み書きのチューニングを解説します。ここで取り上げるパラメータは、PostgreSQLのデータ読み取り、データ書き込みに関わる基本的な動作を調整するものです。どの設定項目も正しくチューニングすれば確実に効果が出ますが、10～100倍の改善は望めません。パフォーマンスの改善は最大でも30%程度です。ただし、デフォルトの値から悪く変更すれば、大きく悪化します。

# 読み書きのチューニング(1) 共有メモリバッファの設定



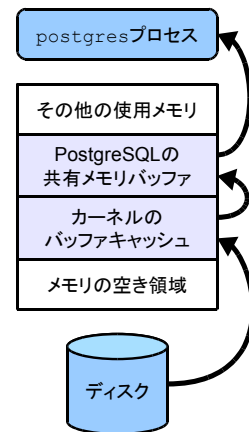
- 共有メモリバッファ(データの読み取り時に使用されるバッファ)を増やして読み取り性能を向上させる

- デフォルト値(32MB以下)は少なすぎる
- 物理メモリが1GB以上の場合には、共有メモリバッファに1/4くらいを割り当てる

`shared_buffers = 256MB` 共有メモリバッファのメモリ容量(要再起動)

- 共有メモリバッファを大きくしすぎると、メモリ領域を圧迫して逆に性能が低下

- Linuxでは余ったメモリ領域をディスクI/Oのバッファキャッシュとして使用してくれる



## 共有メモリバッファの設定

`shared_buffers`はテーブルやインデックスのデータ読み取り時に使用される共有メモリバッファのメモリ容量を設定するパラメータです。共有メモリバッファを介してデータを読み取ることによってディスクに直接アクセスする回数を減らして読み取り性能を向上させることができます。`shared_buffers`のデフォルト値は32MB以下です。

基本的に`shared_buffers`を増やせば増やすほど読み取り性能を向上させることができますが、メモリ領域を圧迫してディスクスワップの発生によって性能を低下させては意味がありません。また、Linuxでは余ったメモリ領域をディスクI/Oのバッファキャッシュとして使用してくれるため、`shared_buffers`には物理メモリの1/4くらいまでを設定するとよいでしょう。

ただし、古いバージョン(とくに7.4以前)では、共有バッファメモリの管理が洗練されておらず、`shared_buffers`に80MB(8kB × 10000)以上のメモリ容量を設定すると、読み取り性能に効果がないだけでなく逆に低下してしまいます。

`shared_buffers`に大きなメモリ容量を設定すると、カーネルの制限によって共有メモリを確保できずにサーバを起動できないことがあります。その場合には共有メモリの最大サイズ(`SHMMAX`)や総量(`SHMALL`)などのカーネルパラメータを変更する必要があります。Linuxでは`sysctl`コマンドによってカーネルパラメータを設定できます。例えば、共有メモリの最大サイズを128MB(134217728バイト)に一時的に変更する場合には以下のようにコマンドを実行します。

```
# /sbin/sysctl -w kernel.shmmax=134217728
```

OSの起動時にカーネルパラメータを変更する場合には`/etc/sysctl.conf`に設定を記述します。`sysctl.conf`に記述した設定をただちに反映させるには`-p`オプションを指定して`sysctl`コマンドを実行します。

## 読み書きのチューニング(2) WALバッファの設定



- WALバッファ(WALファイルの書き込み時に使用されるバッファ)を増やして書き込み性能を向上させる
    - デフォルト値(64kB)は少なめ
    - 少なくとも256~512kB、更新が多い場合には1MBくらいまで増やす
    - サーバ起動時に共有メモリから確保される
- ```
#wal_buffers = 64kB
```
- WALバッファのメモリ容量(要再起動)
- WALバッファが少なすぎると、コミット時以外にも書き込みが発生してしまう



### WALバッファの設定

`wal_buffers`はWALファイルの書き込み時に使用されるWALバッファのメモリ容量を設定するパラメータです。`wal_buffers`のデフォルト値は64kBです。

`wal_buffers`を増やすことによって1回にWALファイルに書き込むことのできるデータ量が増え、更新するデータ量の多いトランザクションを処理する場合や遅延コミットによって複数のトランザクションをまとめて処理する場合に効果があります。

# 読み書きのチューニング(3) 同期書き込み方式の設定



- 最適な同期書き込み方式を選択

```
#wal_sync_method = fsync
```

 同期書き込み方式(要再起動)

- 複数の同期書き込み方式が存在し、OSやファイルシステムなどによって最適な方式が異なる
  - open\_datsync、fdatsync、fsync\_writethrough、fsync、open\_sync

- ベンチマークを実施して書き込み時間が最も短くなる同期書き込み方式を検証

```
$ cd /usr/local/src/postgresql-8.3.6/src/tools/fsync  
$ make  
$ ./test_fsycn
```

```
Compare file sync methods with 2 8k writes:  
(o_dsync unavailable)  
write, fdatsync          0.647364  
write, fsync,            0.602060
```

PostgreSQL運用管理トレーニング(8 チューニングの実施)  
Copyright © SRA OSS, Inc.

12



## 同期書き込み方式の設定

PostgreSQLは、データ書き込みの際、WALと共有バッファに書き込みを行います。このとき、通常、WALに対しては同期書き込みを行います。wal\_sync\_methodはWALへの同期書き込みの方法を設定するパラメータです。スライドに挙げた候補の中から選びます。

どの値を指定するのがよいかは、OS、ファイルシステム、ディスク装置などによって異なります。OSによっては選択できない値もあります。

値を選ぶには実験をしてみるのが最善の方法です。PostgreSQLのソースセットには同期書き込み方法を比較するプログラムtest\_fsycnが含まれています。これを実行すると、各同期書き込み方法で同じサイズのデータを書き込み所要時間(秒単位)が表示されます。ここで最も所要時間の少ないものを選びましょう。

なお、極めて短時間で書き込みが終了している場合(1秒未満など)、ストレージデバイス内で書き込みキャッシュが働いている可能性があります。それが安全であるものが留意する必要があります。