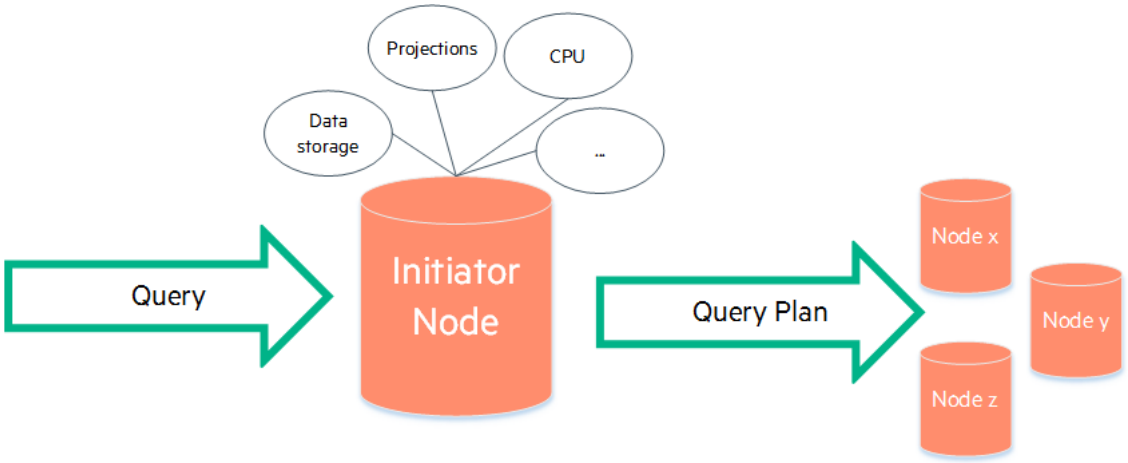
クエリバジェットのチューニング手法

原文はこちら

「バジェット」という言葉で怖がらせてしまう前に、このブログを読んだ後に、好きな活動をあきらめたり、車を売ったりしなければならないということはないでしょう。本書の内容により、Vertica のリソースプールパラメーターがクエリのバジェットにどのように影響するかを理解できるようになります。クエリのバジェットが不十分な場合、クエリの処理速度が低下する可能性があります。しかしながら、恐れる必要はありません。バランスブックと同じように、Vertica はクエリのリソースバジェットを見直し、変更する方法を提供します。

Vertica のクエリバジェット

Vertica でクエリを実行すると、イニシエーターノードは、Vertica がデータを格納する場所、使用可能なプロジェクション、使用可能な CPU 帯域幅、およびその他の要素を決定します。この情報から、イニシエーターノードはクエリプランを作成し、それをクエリを実行するその他のノードに送信します。



各ノードは、クエリの一部を実行するために必要なメモリと同時実行性の量を見積もります。これがクエリのバジェットとなります。

通常、計算されたクエリバジェットはクエリのパフォーマンスを最大限に引き出します。しかし、バランスブックを無視するのと同じように、リソースプールのパラメーターを無視すると、クエリのパフォーマンスに悪影響を与える可能性があります。

クエリバジェットの要素

Vertica は、クエリが実行されるリソースプールに関連付けられた次のパラメーターに基づいて、クエリのバジェットを見積もります。

- MEMORYSIZE
- MAXMEMORYSIZE
- PLANNEDCONCURRENCY

さらに、各リソースプールにはキューイングの閾値があります。各リソースプールについては下記の通りとなります。

- キューイングの閾値 = 各プールの MAXMEMORYSIZE の 95%

リソースプールに対するすべての要求によって使用されるメモリの量がキューイングの閾値を超えると、プールに対する新しい要求はメモリが使用可能になるまでキューに入れられます。このプロセスを「保留中」と考えてください。あなたのアカウントでもっと多くのお金が利用できるようになるまで、手数料は処理されません。

リソースプールの MAXMEMORYSIZE を指定しない場合、MAXMEMORYSIZE はプールの MEMORYSIZE に GENERAL プールの MAXMEMORYSIZE を加えたものに等しくなります。したがって、プールの MEMORYSIZE が設定されていない場合（0 がデフォルト）、MAXMEMORYSIZE は単純に GENERAL プールの MAXMEMORYSIZE になります。

これらの値はすべて、RESOURCE_POOL_STATUS システムテーブルで表示できます。

クエリバジェットの計算

上記のリソースプールパラメーターを使用して、Vertica は次の方法で各リソースプールのクエリバジェットを計算します。

GENERAL プール：

- クエリバジェット = GENERAL プールのキューイングの閾値 / GENERAL プールの PLANNEDCONCURRENCY

MEMORYSIZE が 0 で MAXMEMORYSIZE が設定されていない場合：

- クエリバジェット = GENERAL プールのキューイングの閾値 / 該当プールの PLANNEDCONCURRENCY

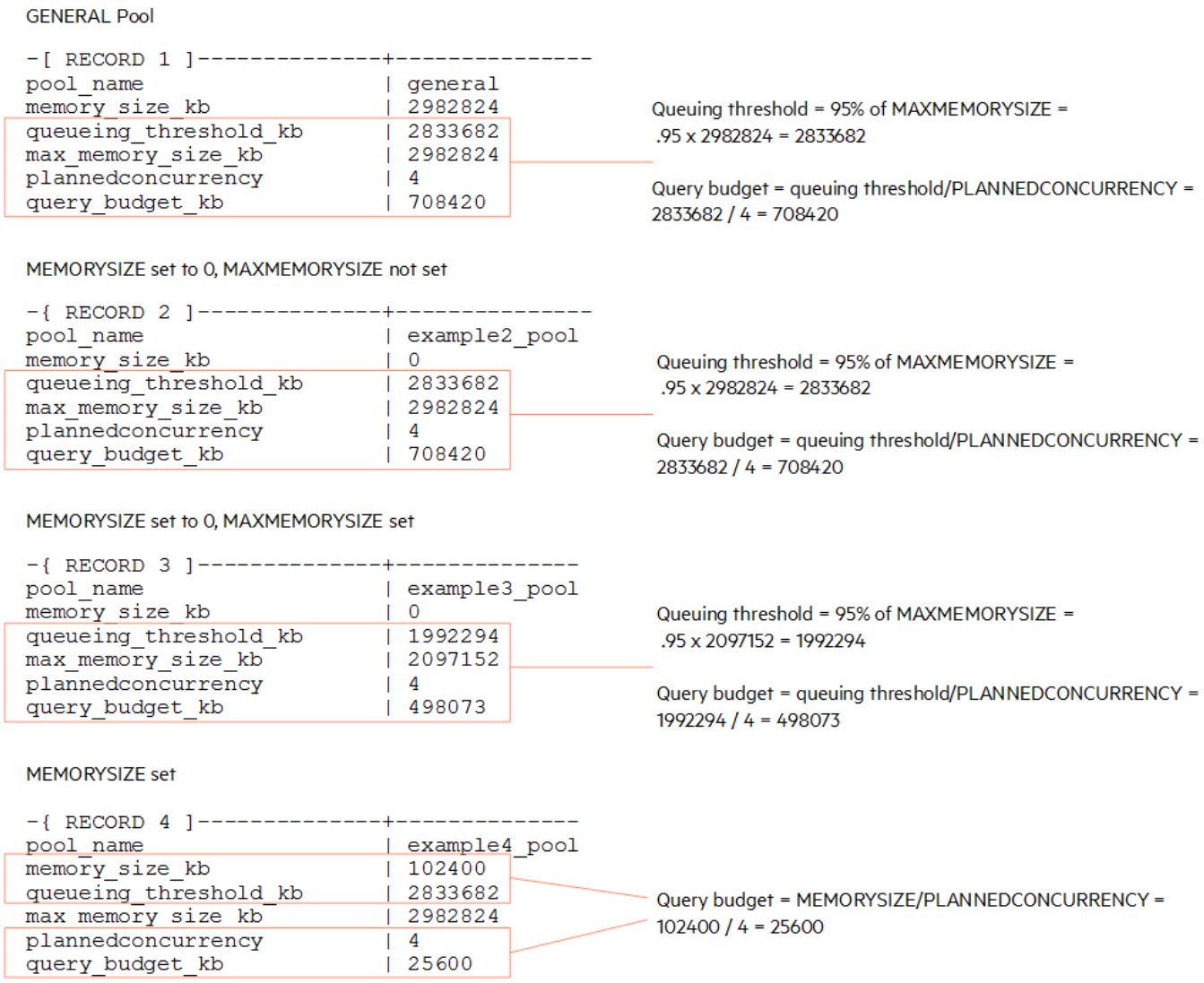
MEMORYSIZE が 0 で MAXMEMORYSIZE がデフォルトではない値に設定されている場合：

- クエリバジェット = 該当プールのキューイングの閾値 / 該当プールの PLANNEDCONCURRENCY

MEMORYSIZE がデフォルトではない値に設定されている場合：

- クエリバジェット = 該当プールの MEMORYSIZE / 該当プールの PLANNEDCONCURRENCY

次の図は、上記の計算の例を示しています。



クエリバジェットのチューニング

MEMORYSIZE パラメーターと PLANNEDCONCURRENCY パラメーターを調整することにより、クエリで使用されるメモリ量を制限することができます。つまり、クエリのバジェットを設定できます。

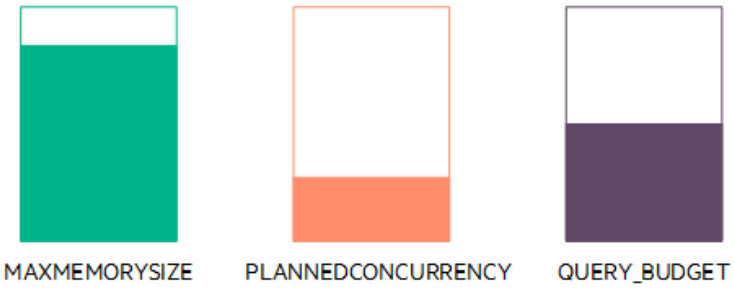
他の目的でメモリが必要であるため MAXMEMORYSIZE を減らす場合、クエリのバジェットも縮小されることに注意してください。クエリのバジェットを減らすことは、特にクエリが複雑な場合、クエリのパフォーマンスに悪影響を及ぼします。

リソースプールの元のクエリバジェットを維持するには、MAXMEMORYSIZE を減らす際に PLANNEDCONCURRENCY の値も減らしてください。

上記の RECORD 1 の GENERAL プールの例を使用して、これがどのように機能するかを見ていきましょう。

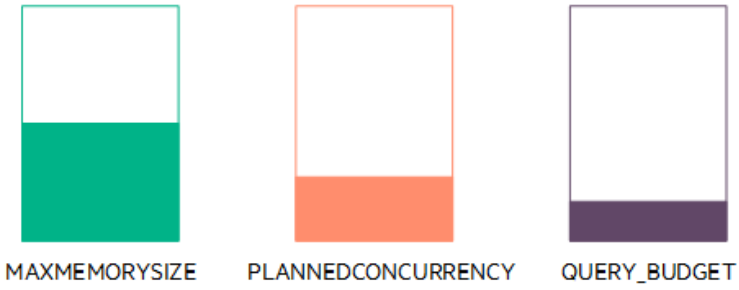
GENERAL Pool

```
-[ RECORD 1 ]---------------+------------------
pool_name                   | general
memory_size_kb              | 2982824
queueing_threshold_kb       | 2833682
max_memory_size_kb          | 2982824
planned_concurrency         | 4
query_budget_kb             | 708420
```



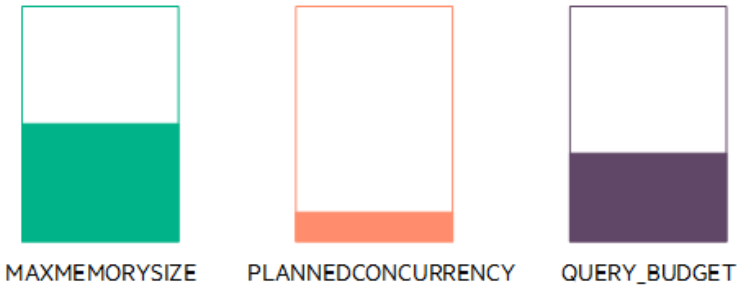
MAXMEMORYSIZE を三分の一減らしたとします。新しいクエリバジェットは同じ量だけ削減されます。

```
=> ALTER RESOURCE POOL GENERAL maxmemorysize '2000M';
=> SELECT * FROM RESOURCE_POOL_STATUS
     WHERE pool_name = GENERAL;
-[ RECORD 1 ]---------------+------------------
memory_size_kb              | 2982824
queueing_threshold_kb       | 1945600
max_memory_size_kb          | 2048000
planned_concurrency         | 4
query_budget_kb             | 486400
```



クエリが通常どおり実行されるように元のクエリのバジェットに戻すには、PLANNEDCONCURRENCY を 3 に減らします。クエリのバジェットが元の 708420 に近づきました。

```
=> ALTER RESOURCE POOL general maxmemorysize '2000M';
=> SELECT * FROM RESOURCE_POOL_STATUS
     WHERE pool_name = GENERAL;
-[ RECORD 1 ]--------------+------------------
memory_size_kb             | 2982824
queueing_threshold_kb      | 1945600
max_memory_size_kb         | 2048000
planned_concurrency        | 3
query_budget_kb            | 648533
```



結論

リソースプールのパラメーターがクエリのバジェットにどのように影響するかについて分かりました。注意して調整するようにください。最も重要なことは、リソースプールの MAXMEMORYSIZE を低くする必要がある場合、クエリバジェットのサイズを減らしてクエリのパフォーマンスに不注意で悪影響を与えないように、PLANNEDCONCURRENCY を調整します。

詳細情報

Vertica ドキュメントの Resource Pool Architecture を参照してください。