

自動共有メモリでのSGAメモリ容量の管理設定方法

自動共有メモリ管理（SGA用）の動作モード種類

共有プールの構造

「共有プール」のメモリ使用量の監視方法

共有プールのサブプール毎のメモリ使用状況の調査方法

ORA-04031 共有メモリの ???バイトを割当てできません
への対応方法

現在使用中だった共有プールとバッファ・キャッシュ・サイズ
のポインタをSPFILEからリセットする方法

バッファ・キャッシュの使用メモリ容量を小さくする方法

SGAメモリ使用量の増加原因の調査方法

共有プール内のリテラル付きSQL文の一覧出力

共有プールのメモリの必要サイズの決定方法

SGAメモリ・サイズの決定のための指定メモリ・サイズの見積り計算
方法

自動共有メモリ管理（SGA用）の動作モード種類

DEFERRED モード

MMON プロセス（バックグラウンドプロセス）が収集するメモリ使用状況を基
に、MMAN プロセスが、コンポーネント間のメモリ割当てを自動調整する

IMMEDIATE モード（共有プール優先の強制モード）

ORA-004031（共有プール領域のメモリ確保エラー）が発生する場合には、強制
的他のコンポーネントのメモリを縮小して、共有プール領域にメモリを割り当てる
なお、共有プール領域へ割当てられたメモリは、増加は行われるが縮小されるこ
となく、増加の一方通行で管理される

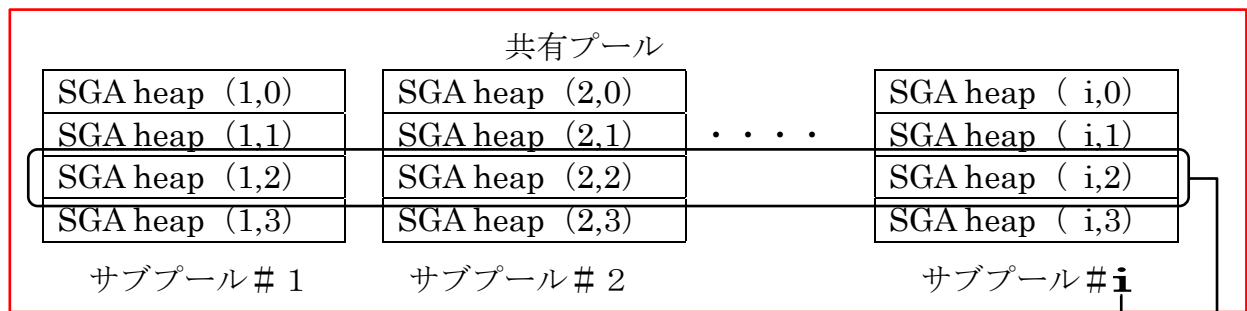
共有メモリ（SGA）と共有プールの論理構成



自動共有メモリ管理下で最低確保容量の確保の方法

SGA_TARGET が設定されて自動共有メモリ管理が行われている環境において、自動調整対象のコンポーネント（バッファ・キャッシュ、共有プール、Java プール、ラージ・プール）の初期化パラメータが明示的に指定された場合には、**そのコンポーネントの下限サイズ**となり、メモリが少なくともこれ以上に確保される

共有プールのメモリ構造



サブプールは、共有メモリ分割管理のための区分け ←

存続期間に関する分割管理区分 ←

※ () 内の右側の 0 ~ 3 の数字

ヒープ区分	存続期間	確保されるメモリ用途
SGA heap (i, 0)	D/B インスタンスの起動期間中	永続メモリ領域
SGA heap (i, 1)	セッションが接続されている期間中	ライブラリ・キャッシュ ディクショナリ・キャッシュ
SGA heap (i, 2)	カーソルが存在している期間中	親カーソル、個カーソル
SGA heap (i, 3)	SQL 文などの実行のコール中	SQL 領域

共有プールの新規メモリ割当て手順

- 1) 各サブプールのフリーリストから空きを探し、割当てる
フリーリストは、各ヒープ sga heap (x , y) 毎に存在
- 2) 未割当の領域 (Reserved Granule) から割当てる
- 3) 予約済みプールから割当てる (一定サイズ以上のものを割り当てる領域)
- 4) 使用済み領域を再利用して、空き領域を作成して割当てる
- 5) 新たなメモリを割り当てて共有プールの中の 1 つのサブプールを拡張して、その拡張されたサブプールの中から割当てる
- 6) 割当てが出来なかった場合は、ORA-04031 エラーが発生する

共有プールのメモリ不足の発生についての注意点

共有プールのメモリは、サブプールとヒープ区分によって分割管理されている
よって、必要なメモリ量が合計として残っていても、断片化していて必要な処理に割当てられない場合もある

メモリの残容量については、「使用済み領域を再利用」や「メモリの断片化」などの問題があるので、共有プール・メモリ残容量と個別の新たな処理へのメモリ割当ての可否を、管理者が計算で判断するのは困難 (≒不可能) である

「共有プール」のメモリ使用量の監視方法

・共有プール全体（全サブプールの合計）の空き領域のサイズ確認

現在割り当てられている共有プールに対しての残容量を示す

共有プールのサイズを固定で設定している場合には、この値が残容量を示す

但し、SGA の自動共有メモリ管理を使用して、動的に共有プールのメモリ量を管理している場合には、次の「拡張余力メモリ」についても残容量と合わせて考慮する必要がある

```
SELECT * FROM v$sgastat
WHERE pool = 'shared pool' AND name = 'free memory';
```

POOL	NAME	BYTES
shared pool	free memory	29654021

・共有プールの拡張余力メモリのサイズ確認

自動共有メモリ管理（SGA用）を使用している場合には、共有プールのメモリは、残量が無くなれば拡張されて確保されていく

よって、共有プールのメモリ残量として注意しておかなければならないのは、V\$SGA_DYNAMIC_COMPONENTS の拡張余力メモリがどれだけ残っているかについても考慮が必要である

```
SELECT USER_SPECIFIED_SIZE, CURRENT_SIZE,
       CURRENT_SIZE - USER_SPECIFIED_SIZE SIZE_DIFFERENCE
FROM   V$SGA_DYNAMIC_COMPONENTS
WHERE  COMPONENT = 'DEFAULT buffer cache';
```

USER_SPECIFIED_SIZE	CURRENT_SIZE	SIZE_DIFFERENCE
2,147,483,648	4,385,875,968	2,238,392,320
初期設定値	現在値	拡張余力(自由調整領域)

・ 共有プールのサブプール毎のメモリ使用状況の調査方法

共有プールのサブプール単位での残量の確認については、X\$KSMSS ビューに対して、次の SQL を発行して調査する

注意)

この起動操作については、**SYS ユーザー**を **SYSDBA 権限**で接続して行うこと
SYSTEM ユーザー等では、「ORA-00942: 表またはビューが存在しません。」となり実行できない

```
SELECT  KSMDSIDX  SUBPOOL#,
        DECODE( KSMSSNAM, 'free memory' , 'free memory' , 'used memory' )  TYPE ,
        SUM( KSMSSLEN )  BYTES
        FROM  X$KSMSS
        GROUP BY KSMDSIDX,
                decode( KSMSSNAM , 'free memory' , 'free memory' , 'used memory')
        ORDER BY KSMDSIDX,
                decode( KSMSSNAM , 'free memory' , 'free memory' , 'used memory')  ;
```

USER_SPECIFIED_SIZE SUBPOOL#	TYPE	CURRENT_SIZE BYTES	SIZE_DIFFERENCE
0	free memory	0	
0	used memory	0	
1	free memory	52,393,584	←
1	used memory	232,819,088	
2	free memory	53,940,240	←
2	used memory	231,272,432	
3	free memory	46,664,144	←
3	used memory	221,771,312	
4	free memory	60,816,984	←
4	used memory	190,841,256	

SUBPOOL# = 0 のサブプールは、インスタンス起動時だけに使用されるので無視すること

1 以降の番号の各サブプールの使用メモリの容量が均等かを確認する

「ORA-04031 共有メモリの ???バイトを割当てできません」への対応方法

発生原因

- ・ 共有プールの領域不足
- ・ 共有プールの断片化が発生

原因の判別方法

(自動共有メモリ管理を行っている場合も含め、)

ORA-4031 エラー・メッセージの要求サイズと `v$sgastat` の空きバイトを比較して、要求サイズの 1000 倍未満あれば共有プールの残量不足と考える

それ以上の場合は、共有プール内での空メモリの断片化が発生していると考え

共有プールの残容量不足に対する対応策

- ・ 固定で共有プールのサイズを指定している場合は、初期化パラメータ `SHARED_POOL_SIZE` のサイズを大きな値に設定し、共有プールの拡張を行う
- ・ 自動共有メモリ管理 (SGA用) を使用している場合にも、共有プール・最下限サイズとして初期化パラメータ: `SHARED_POOL_SIZE` を大きな値に設定し、共有プールの最低限の容量を確保する
また、断片化を解消できない場合にもこの方法をとる
- ・ バッファ・キャッシュの使用メモリ容量を小さくする (動的対応が可能)

共有プールの断片化に対する対応策

- ・ インスタンスを再起動する (次ページ参照)

※ 自動共有メモリ管理を使用している場合、前回起動時のメモリ割当てを記憶していて、これを使用して SGA 内のメモリ割り当てが行われるので、割当て記憶をリセットする必要がある

- ・ 定期的に `ALTER SYSTEM FLUSH SHARED_POOL` を実行して、共有プールの未使用領域を解放し、断片化を無くす

※ この方法は、データベース負荷が高い処理なので、インスタンスの負荷が高い時には行わないこと

- ・ `DBMS_SHARED_POOL.KEEP` プロシージャを使用して、使用頻度の高いパッケージを共有プールに固定し、再ロードを防止する
- ・ 次回再発防止のためには、初期化パラメータ (`SHARED_POOL_SIZE`、`SGA_TARGET`) を調整する
- ・ 予防対策して、共有プールのしきい値を監視し、残容量が少なくなったら対応を行う

現在使用中だった共有プールとバッファ・キャッシュ・サイズのポインタを SPFILE からリセットして Oracle インスタンスを再起動する方法

- Oracle インスタンスの再起動を行う場合、現在使用中の共有プールとバッファ・キャッシュの現在使用中のメモリポインタは、保持されたままである。

このため、再起動前に共有プールとバッファ・キャッシュのメモリポインタを以下の手順でリセットして、再起動を行う

```
shutdown immediate
```

```
startup mount
```

```
alter system reset "__shared_pool_size" SCOPE=SPFILE SID= '<Oracle  
インスタンス ID>';
```

```
alter system reset "__db_cache_size" SCOPE=SPFILE SID= '<Oracle イン  
スタンス ID>';
```

```
alter system reset "__java_pool_size" SCOPE=SPFILE SID= '<Oracle イン  
スタンス ID>';
```

```
alter system reset "__large_pool_size" SCOPE=SPFILE SID= '<Oracle イン  
スタンス ID>';
```

```
alter system reset "__streams_pool_size" SCOPE=SPFILE SID= '<Oracle  
インスタンス ID>';
```

```
alter database open ;
```

バッファ・キャッシュの使用メモリ容量を小さくする方法

```
alter system set db_cache_size = <確保容量サイズ>G SCOPE=BOTH  
SID='<Oracle インスタンス SID>';
```

SGAメモリ使用量の増加原因の調査方法

- 1) どのコンポーネントのメモリ使用量が増えているかを調査する

AWR レポートには、V\$SGASTAT の情報が出力されるので、どのコンポーネントが増えていったかを時系列で確認する

【新・門外不出の Oracle 現場ワザ】 ADDM レポートの全体構成

(= AWR の出力結果) の中の

「メモリ統計」セクションの『SGA メモリ情報』、『SGA ブレイクダウン差異』

「アドバイザ統計」セクションの『SGA アドバイザリ』、『メモリターゲットアドバイザ』、『バッファ・プールアドバイザリ』

を確認する

- 2) 共有プールのサブプールの使用に偏りがいないか確認する

3 ページ前の「共有プールのサブプール毎のメモリ使用状況の調査方法」を参照のこと

- 3) ハード解析回数が増えていないか確認する

ハード解析回数が増えている場合には、リテラル SQL のアプリケーションが使用された可能性がある。SQL 文を確認して、バインド変数に変更できないか検討する

共有プール内のリテラル付き SQL 文の一覧出力

sql> select SQL_TEXT, SQL_FULLTEXT from V\$SQL ;

共有プール不足を防止するための残容量しきい値の目安

- ・共有プールのサイズを固定で設定している場合

通常時から共有プールの残容量を記録して、下限許容値を下回ったら拡張を行う

- ・自動共有メモリ管理（SGA用）を使用している場合

拡張余力が、2 G を切った場合には、業務状況を確認して、Oracle を停止するタイミングを調整し、インスタンスの再起動の検討を行う

※ 自動共有メモリ管理を使用している場合、前回起動時のメモリ割当てを記憶していて、これを使用して SGA メモリの割り当てが行われるので、割当て記憶をリセットする必要がある

自動共有メモリ管理（SGA用）を使用している時の 最低共有プール・サイズと最低バッファ・キャッシュ・サイズの指定方法

SGA メモリ容量の管理イメージ と 初期化パラメータ

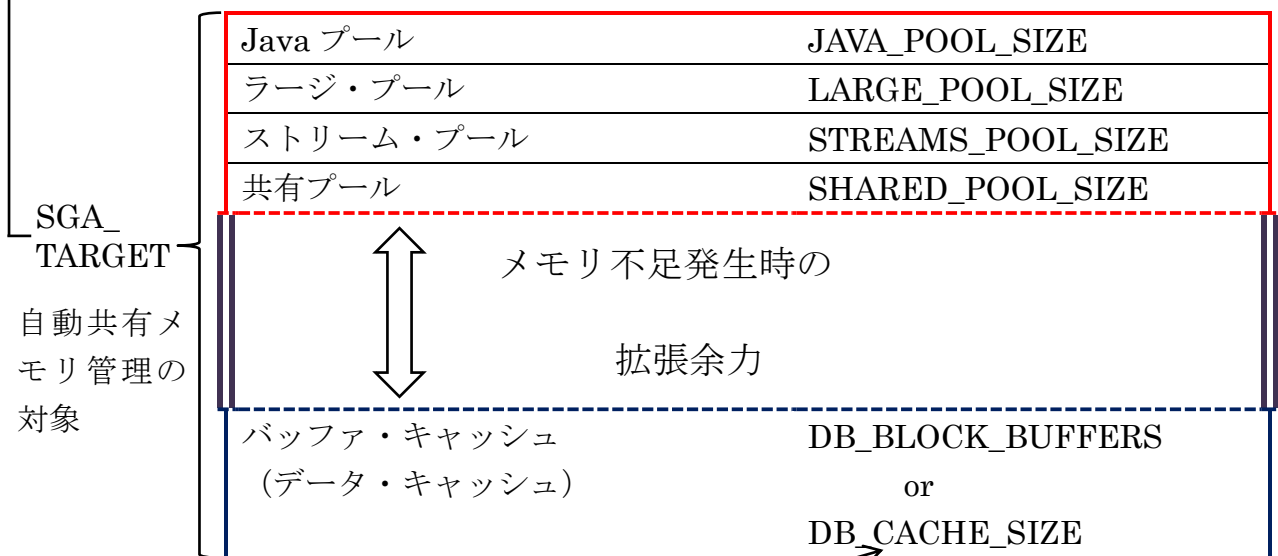
SGA 自動共有メモリ管理対象外メモリ

Redo ログ・バッファ（ LOG_BUFFER ）

KEEP バッファ・キャッシュ（ DB_KEEP_CACHE_SIZE ）

RECYCLE バッファ・キャッシュ（ DB_RECYCLE_CACHE_SIZE ）

非標準ブロックサイズ・バッファ・キャッシュ（ DB_nK_CACHE_SIZE ）



SGA_TARGET を指定して、SGA 自動共有メモリ管理を行ってメモリ配分の自動管理を行っている状況で、自動調整対象のコンポーネントに対して初期化パラメータを明示的に指定した場合には、対象コンポーネントの下限値を指定したことになり、メモリ不足時には、必要な分のメモリが拡張余力メモリから与えられる

※ 実運用では、SHARED_POOL_SIZE（共有プール）と DB_BLOCK_BUFFERS or DB_CACHE_SIZE（バッファ・キャッシュ）の最低値の2つのみの指定が、多い

DB_BLOCK_BUFFER と DB_CACHE_SIZE の違い

DB_BLOCK_BUFFER と DB_CACHE_SIZE は、両者ともデータベースのオブジェクト（テーブル or インデックス）の Disk 読込みのときに、メモリへの取り込みを一括で行うバッファ・キャッシュ（データ・キャッシュ）の単位サイズを指定するものです
全く同じ対象のメモリを確保するための初期化パラメータです

両者の違いは、指定時に確保するメモリのサイズ（バイト）で指定するか、データベース・ブロックの個数で指定するかの違いです

このバッファ・キャッシュが使用される対象は、Oracle インストール時に規定した DB_BLOCK_SIZE（標準データベース・ブロックサイズ）の大きさ単位で、ディスク・ブロックが管理（確保）されているオブジェクトです

なお、非標準データベース・ブロックサイズでメモリの確保が管理されているオブジェクトのバッファ・キャッシュ確保のためには、DB_<n>K_CACHE_SIZE を指定します

DB_CACHE_SIZE

バッファ・キャッシュ（データ・キャッシュ）に確保するメモリを容量で指定する
DB_CACHE_SIZE = 整数値 [K | M | G] (バイト)

DB_BLOCK_BUFFER

バッファ・キャッシュ（データ・キャッシュ）に確保するメモリをデータベース・ブロックの個数で指定する

DB_BLOCK_BUFFER = 整数

実際に確保されるメモリ量は、DB_BLOCK_SIZE × DB_BLOCK_BUFFER

※ 初期化パラメータ・ファイルで指定できるのは、どちらか一方の方法だけです
両者を2つとも記述した場合には、エラーが発生します

DB_BLOCK_SIZE

DB_BLOCK_SIZE は、インストール時に規定した Oracle データベースの標準データベース・ブロックサイズです。

デフォルト値は、8192 バイトです

SGA メモリ・サイズの決定のための指定メモリ・サイズの見積り計算方法

自動共有メモリ管理使ってメモリ割当ての自動制御を行うにあたって、指定しなければならないコンポーネント

- ・ SGA_TARGET 全体のメモリ容量サイズ
- ・ 以下のコンポーネントは、自動共有メモリ管理で自動に配分されるものなので、SGA_TARGET パラメータを使って、自動メモリ管理の全体容量を設定し、必要なメモリ量を適宜配分しておく方式をとる

ただし、パフォーマンスの観点から**必要最低限值**を各パラメータで指定しておく

対象コンポーネント	初期化パラメータ	指定低限值
Java プール	JAVA_POOL_SIZE	設定しないこと
ラージ・プール	LARGE_POOL_SIZE	設定しないこと
ストリーム・プール	STREAMS_POOL_SIZE	設定しないこと
共有プール	SHARED_POOL_SIZE	※ 別途見積り
バッファ・キャッシュ (データ・キャッシュ)	DB_BLOCK_BUFFERS or DB_CACHE_SIZE	※ 別途見積り

①. 構築するデータベースの構成を決める

- ・ シングル構成 or RAC 構成
RAC の場合には、RAC を構成する Oracle インスタンスの数も決める
- ・ 最大プロセス発生想定数
(初期化パラメータ : PROCESSES での指定も必要である)

②. 自動共有メモリ管理対象外コンポーネントへのメモリ容量サイズの指定

以下のコンポーネントは、自動共有メモリ管理で自動に配分されるものではないので、個別に必要なメモリ容量を見積もって、初期化パラメータ・ファイルにメモリ・サイズの設定を行う

- ・ Redo ログ・バッファ (LOG_BUFFER)
- ・ KEEP バッファ・キャッシュ (DB_KEEP_CACHE_SIZE)
- ・ RECYCLE バッファ・キャッシュ (DB_RECYCLE_CACHE_SIZE)
- ・ 非標準ブロックサイズ・バッファ・キャッシュ (DB_nK_CACHE_SIZE)

必要メモリ・サイズは、インターネット等で調査して見積りを行う

③. 共有プールの中に作成されるサブプールの個数を求める

※ 共有プールのサブプールの数は、CPU 数、共有プール・サイズの大きさにより、異なった数で分割される

詳しくは、KROWN : 147122 Doc ID 17523151.1 参照のこと

もしくは、共有プールのサブプールの使用状態の偏り調査

```
SELECT distinct KSMDSIDX SUBPOOL# from X$KSMSS;
```

④. SGA_TARGET 全体のサイズを決定する

テスト段階では SGA+PGA のメモリ割り当ての自動化を使用して、それぞれのメモリ割り当て量の監視を行う

本番段階では、テスト時に割り当てられていたメモリ量の実績から、SGA メモリの総量と PGA メモリの総量を決定し、**それぞれに分けてメモリ割り当ての自動化**を行う

⑤. 共有プールの設定下限のメモリ・サイズを算出する

共有プール (SHARED_POOL_SIZE) のメモリの下限サイズの決定方法

- 1) バッファ・キャッシュ依存分の共有プール必要量
- 2) PROSESS 数依存分の共有プール必要量
- + 3) その他の共有プール必要量

合計メモリ量 が、共有プールの
必要最低値

- 1) バッファ・キャッシュ依存分の共有プール必要量

＜バッファ・キャッシュ 1 ブロック当りに必要な共有プール・メモリサイズ＞

×

(SGA_TARGET — JAVA_POOL_SIZE — LARGE_POOL_SIZE
— STREAMS_POOL_SIZE) ÷ DB_BLOCK_SIZE

※ 通常の設定では、Java プール等は、0 である

バッファ・キャッシュ 1 ブロック当りに必要な共有プール・メモリサイズ

Oracle 分散環境の有無	1 ブロック当りに必要メモリ
RAC 環境 (Real Application Clusters)	500 バイト
シングルインスタンス	80 バイト

2) PROSESS 数依存分の共有プール必要量

$$0.121\text{M バイト} \times \text{最大プロセス発生想定数 (①. で決めた値)} \\ + 316\text{M バイト}$$

3) その他の共有プール必要量

$$1\text{ GM バイト} \\ + 1\text{ GM バイト} \times \text{＜共有プールのサブプールの数＞} \\ ※ ③. で求めた値$$

⑥. 拡張余力サイズの決定

$$\text{拡張余力サイズ} = \text{共有プール・サイズ}$$

⑦. バッファ・キャッシュの設定下限のメモリ・サイズを算出する

$$\text{バッファ・キャッシュ・サイズ} \\ = \text{SGA_TARGET サイズ} - \text{共有プール・サイズ} \\ - \text{拡張余力サイズ}$$

SGA メモリ・サイズの設定例（見積り方法）

①．構築するデータベースの構成を決める

RAC 環境

RAC 構成インスタンス数： 2

最大プロセス発生想定数（PROCESSES）： 4000 個

②．自動共有メモリ管理対象外コンポーネントへのメモリ容量サイズの指定 別途に調査して、決定する

③．共有プールの中に作成されるサブプールの個数を求める テスト段階のサブプールの実数を調査する

この例では、

共有プールのサブプール数： 3 個 と想定する

④．SGA_TARGET 全体のサイズを決定する

テスト段階のメモリ割当て状況を調査して、判断する

この例では、

SGA_TARGET サイズ： 45G バイト と想定する

⑤．共有プールの設定下限のメモリ・サイズを算出する

DB_BLOCK_SIZE： 8192 バイト

JAVA_POOL_SIZE： 設定無し

LARGE_POOL_SIZE： 設定無し

STREAMS_POOL_SIZE： 設定無し

1) バッファ・キャッシュ依存分の共有プール必要量

RAC 環境のブロック単位必要量 × ブロック数

$500 \times ((45 \times 1024 \times 1024) - 0 - 0 - 0) / 8192$ 2.8GB

2) PROSESS 数依存分の共有プール必要量

0.121M バイト × PROCESSES 数 + 316M バイト 800MB

3) その他の共有プール必要量

1GB + 1GB × サブプール数 4.0GB

合計

7.6GB

⑥．拡張余力サイズの決定

拡張余力 = SHARED_POOL_SIZE = 7.6GB

⑦. バッファ・キャッシュの設定下限のメモリ・サイズを算出する

【DB_CACHE_SIZE】

SGA_TARGET - SHARED_POOL_SIZE - 拡張余力 = 29.8GB