

SGA メモリと PGA メモリの割当て量の確認とチューニング

1. OEM (EMDC) による【SGA メモリ】と【PGA メモリ】の割当て量の表示方法

[サーバ] タブ → データベース構成・セクション中の「メモリ・アドバイザー」

[SGA] タブと [PGA] タブ にて切り替えて表示させる

↓
次ページ

2. OEM (EMDC) による【SGA メモリ】設定量とアドバイス機能

【自動共有メモリ管理が無効の時】の表示状態

→ 自動共有メモリ管理 無効

有効化

| | | |
|-------------|-----|--------|
| 共有プール | 72 | MB ▽ |
| バッファ・キャッシュ | 68 | MB ▽ |
| ラージ・プール | 4 | MB ▽ |
| Java プール | 4 | MB ▽ |
| その他 | 6 | |
| SGA 合計 (MB) | 154 | 計算 |

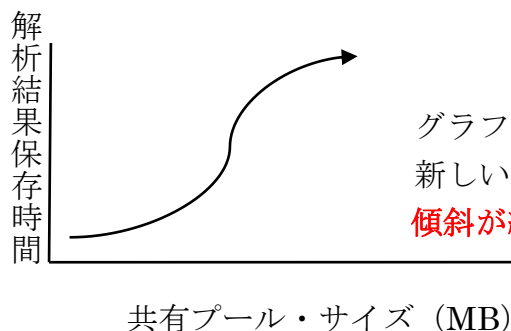
アドバイス
アドバイス

3. 共有プール・サイズのアドバイス

ライブラリ「SQL 実行計画」のキャッシュ

データ・ディクショナリ「オブジェクトの統計情報・定義」・キャッシュ

アドバイス ボタンを押すことにより、共有プールでのキャッシュの保存できる時間の相対グラフが表示されます



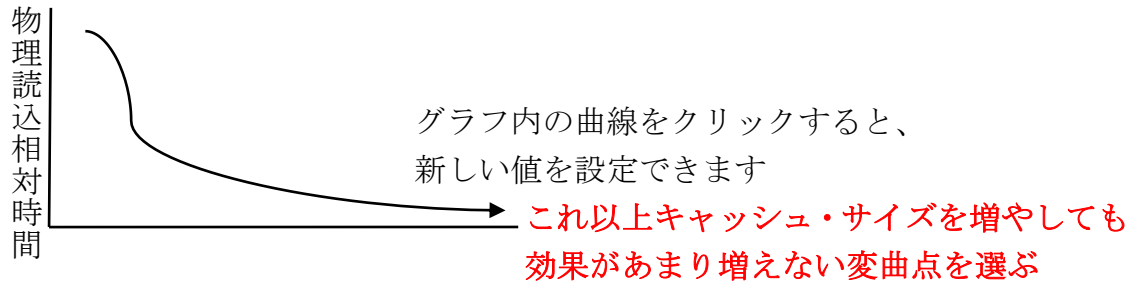
グラフ内の曲線をクリックすると、新しい値を設定できます

傾斜が緩くなる変曲点のメモリ・サイズを選ぶ

4. バッファ・キャッシュ・サイズのアドバイス

オブジェクト（テーブル、インデックス）・データと Disk I/O の間のバッファ

アドバイス ボタンを押すことにより、バッファ・キャッシュでの物理読込時間の
相対グラフが表示されます



キャッシュ・サイズ (MB)

5. OEM (EMDC) による【PGA メモリ】のメモリ設定値と割当て量の表示

[サーバ] タブ → データベース構成・セクション中の「メモリ・アドバイザ」

PGA ターゲット総計 50 MB ▾ アドバイス ← ※ 全プロセス当りの設定値

割当済の現行 PGA(KB) 44561 ← 全プロセスの合計値

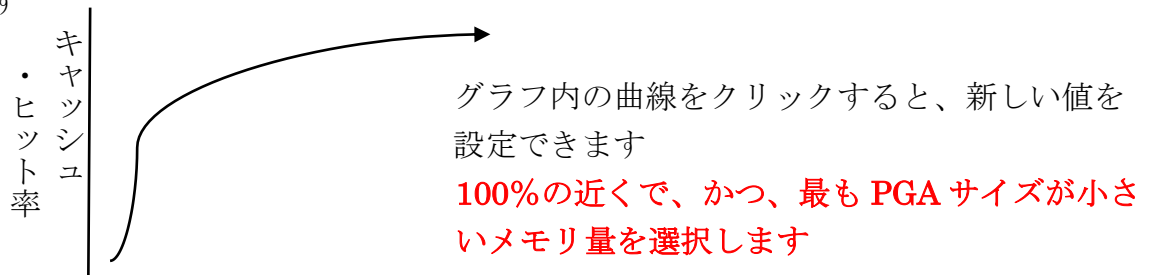
割当済の最大 PGA(KB) 57580 ← 全プロセスの合計値

キャッシュ・ヒット率(%) 99.96

→ PGA メモリの使用量詳細

6. 【PGA メモリ】のメモリ割当て量の最適化アドバイス

アドバイス ボタンを押すことにより、キャッシュ・ヒット率のグラフが表示されます



PGA ターゲット・サイズ (MB)

7. 【PGA メモリ】の使用量詳細の表示

PGA メモリの使用量詳細 ボタンを押すことにより、**SQL** の処理（ソートなど）において作業領域の容量別のソート実行方法の状況（メモリ上だけか、ディスクを使ったか）の割合が表示できます



作業領域サイズ別 (SQL 文の必要量)

PGA ターゲットのメモリ使用量の詳細を表示 MB **実行**

※ この数値を変更して、**実行** ボタンを押すと、アドバイザによりメモリ量を指定した値に仮定して、作業領域の容量別のソート実行方法の状況をシミュレートした予測値でのグラフを新たに表示します

SQL 作業領域の種類

| 種類 | SQL 作業領域の詳細 | 目安 |
|---------------------------|----------------------------------|--------|
| 最適実行 (optimal) | ソートなどの処理がすべてメモリ上で行われた | 90% 以上 |
| 単 一 パ ス 実 行 (onepass) | メモリ上だけでは行えず、最小限のディスク・アクセスが発生した | 10% 未満 |
| マ ル チ パ ス 実 行 (multipass) | 多くの処理において、メモリが不足してディスク・アクセスが発生した | 0 % |

メモリ（SGA、PGA）の割当て目安量

メモリ・チューニングが必要か、の判断基準

OS システム用メモリ： 20%程度は、OS のシステム利用のために残す

SGA

- ・ 共有プール： 「ライブラリ・キャッシュ」のヒット率を 99%以上
「データ・ディクショナリ・キャッシュ」のヒット率
を 95%以上に保つ
- ・ バッファ・キャッシュ： ヒット率を 95%以上に保つ
- ・ Redo バッファ： 待機率を 1%未満に保つ

PGA

「最適実行（メモリだけのソート実行）」の実行回数
の割合を 90%以上に保つ
ただし、PGA の作業領域が特に大量に必要な場合は、こ
の使用メモリで「マルチパスでの実行」の割合を除外して、
メモリが少量の時だけで「最適実行」を考え、PGA の容
量割当てを考えてよい

必要なメモリの概算量
(PGA)

自動 PGA 管理機能を
使用する場合の設定値
の算出根拠

=

セッションあたりのメモリ使用量

×

同時接続ユーザー数

+

バックグラウンド・プロセスのメモリ使用量

×

バックグラウンド・プロセスのプロセス数

—— SGA メモリに関する統計情報 ——

8. SGAメモリの現在の**割当て容量**調査

```
SELECT name, bytes from v$SGASTAT WHERE pool IS NULL
UNION ALL
SELECT pool, SUM(bytes) from v$SGASTAT
WHERE pool IS NOT NULL
GROUP BY pool ;
```

| NAME | BYTES | ← 割当て容量 | 9. に一致する |
|--------------|-------------|---------|------------------|
| fixed_sga | 2241373 | ... | その他 SGA エリア・サイズ |
| buffer_cache | 682580126 | ... | バッファ・キャッシュ・サイズ |
| log_buffer | 380069014 | ... | Redo ログ・バッファ・サイズ |
| java_pool | 18364589 | ... | Java プール・サイズ |
| streams pool | 18364589 | ... | ストリーム・プール・サイズ |
| shared pool | 12368970258 | ... | 共有プール・サイズ ← |
| large pool | 18364589 | ... | ラージ・プール・サイズ |

共有プールは、ライブラリ・キャッシュ（プロシージャなどのライブラリのキャッシュ・データとディクショナリ・キャッシュ（オブジェクトの統計情報や実行計画）とセッション・データ（ユーザーの PGA 使用部分との関連を制御する部分）から構成

```
select sum( sharable_mem ) "共有プール" from v$db_object_cache
union all
select sum( sharable_mem ) from v$sqlarea
union all
select sum( 250 * users_opening ) from v$sqlarea ;
```

| | |
|--------------|--------------------|
| 共有プール | ← これは、 実使用量 |
| ----- | |
| 93868331 | ... ライブラリ・キャッシュ |
| 57471077 | ... ディクショナリ・キャッシュ |
| 66000 | ... セッション・データ |
| 合計 151405408 | ← |

**この2つを比較してほとんど差が無ければ、
割り当てられた共有プール・メモリを使い切っていることになる
この場合は、共有プール・メモリの拡張を行う**

9. SQL コマンドで SGA メモリの割当て量（種類別各割当て量）の表示方法

```
SELECT * FROM v$sgainfo ;
```

| NAME | BYTES | RES |
|---------------------------------|------------|-----|
| Fixed SGA Size | 1334380 | No |
| Redo Buffers | 4644864 | No |
| Buffer Cache Size | 654311424 | Yes |
| Shared Pool Size | 385875968 | Yes |
| Large Pool Size | 8388608 | Yes |
| Java Pool Size | 16777216 | Yes |
| Streams Pool Size | 0 | Yes |
| Shared IO Pool Size | 0 | Yes |
| Granule Size | 8388608 | No |
| Maximum SGA Size | 1071333376 | No |
| Startup overhead in Shared Pool | 58720256 | No |
| Free SGA Memory Available | 0 | |

※ 出力される表示項目は 8. より多い、一致する項目の値は同じである

10. SQL コマンドでの共有プール・サイズのアドバイス表示

```
SELECT * FROM v$shared_pool_advice ;
```

| SHARED_POOL_SIZE_FOR_ESTIMATE | SHARED_POOL_SIZE_FACTOR | ESTD_LC_SIZE | ESTD_LC_MEMORY_OBJECTS | ESTD_LC_TIME_SAVED | ESTD_LC_TIME_SAVED_FACTOR | ESTD_LC_LOAD_TIME | ESTD_LC_LOAD_TIME_FACTOR | ESTD_LC_MEMORY_OBJECT_HITS |
|-------------------------------|-------------------------|--------------|------------------------|--------------------|---------------------------|-------------------|--------------------------|----------------------------|
| ↓ | | | | | | | | |
| 128 | .3478 | 5 | 560 | 37348 | .9978 | 174 | 1.9333 | 37373 |
| 168 | .4565 | 44 | 4821 | 37359 | 998 | 163 | 1.8111 | 282453 |
| 208 | .5652 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 248 | .6739 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 288 | .7826 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 328 | .8913 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 368 | 1 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 408 | 1.1087 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 448 | 1.2174 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 488 | 1.3261 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 528 | 1.4348 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 568 | 1.5435 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 608 | 1.6522 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |
| 648 | 1.7609 | 77 | 8726 | 37432 | 1 | 90 | 1 | 283262 |

1 1. SQL コマンドでの自動共有メモリ管理を使用時の SGA 全体量へのアドバイス表示

```
SELECT * FROM v$sga_target_advice ;
```

| SGA_SIZE | SGA_SIZE_FACTOR | ESTD_DB_TIME | ESTD_DB_TIME_FACTOR | ESTD_PHYSICAL_READS |
|----------|-----------------|--------------|---------------------|---------------------|
| ----- | ----- | ----- | ----- | ----- |
| 1024 | 1 | 375 | 1 | 23070 ←現状設定値 |
| 256 | .25 | 617 | 1.6453 | 33710 |
| 512 | .5 | 375 | 1 | 23070 |
| 768 | .75 | 375 | 1 | 23070 |
| 1280 | 1.25 | 375 | 1 | 23070 |
| 1536 | 1.5 | 375 | 1 | 23070 |
| 1792 | 1.75 | 375 | 1 | 23070 |
| 2048 | 2 | 375 | 1 | 23070 |

選択するのは、ESTD_DB_TIME_FACTOR の値が一定に近づくときの、最も小さい SGA_SIZE の値

1 2. バッファ・キャッシュ・サイズのアドバイスを利用設定するための初期化パラメータ

```
DB_CACHE_ADVICE : ON
show parameter db_cache_advice
```

1 3. SQL コマンドでのバッファ・キャッシュ・サイズのアドバイス表示

```
SELECT size_for_estimate , estd_physical_reads , estd_physical_read_factor
FROM v$db_cache_advice ORDER BY size_for_estimate ;
```

| SIZE_FOR_ESTIMATE | ESTD_PHYSICAL_READS | ESTD_PHYSICAL_READ_FACTOR |
|-------------------|---------------------|---------------------------|
| ----- | ----- | ----- |
| 56 | 33725 | 1.4618 |
| 112 | 24995 | 1.0834 |
| 168 | 23115 | 1.0019 |
| 224 | 23070 | 1 |
| 280 | 23070 | 1 |
| 336 | 23070 | 1 |

選択するのは、ESTD_PHYSICAL_READ_FACTOR の値が一定に近づくときの、最も小さい SIZE_FOR_ESTIMATE の値

1 4 . shared pool、large pool、java pool の空きメモリの確認

```
SELECT pool , name , bytes FROM v$sgastat WHERE name = 'free memory'
AND pool IN ( 'shared pool', 'large pool', 'java pool' ) ;
```

| POOL | NAME | BYTES |
|-------------|-------------|-----------|
| shared pool | free memory | 185706764 |
| large pool | free memory | 7314608 |
| java pool | free memory | 16777216 |

この値が、0に近い場合は、メモリを使い切っていることになる
各プールの拡張を考えること

1 5 . データベース・バッファ（・キャッシュ）ごとのヒット率の表示

```
SELECT name , physical_reads , db_block_gets , consistent_gets,
1 - (physical_reads / (db_block_gets + consistent_gets)) "Hit Rate"
FROM v$buffer_pool_statistics ;
```

| NAME | PHYSICAL_READS | DB_BLOCK_GETS | CONSISTENT_GETS | Hit Rate |
|---------|----------------|---------------|-----------------|------------|
| DEFAULT | 23070 | 318608 | 1494618 | .98727682 |
| KEEP | 4061 | ... | ... | .967532456 |

↑

キープ・バッファ・プールのヒット率

バッファ・キャッシュのヒット率の目安は、95%以上である

下回るのであれば、拡張を考える

キープ・バッファ・プールのヒット率の確認方法

```
select name,
( 1 - physical_reads / ( db_block_gets + consistent_gets ) ) hit_ratio
from v$buffer_pool_statistics where name = 'KEEP' ;
```

| NAME | HIT_RATIO |
|------|------------|
| KEEP | .967532456 |

← キープ・バッファ・プールのヒット率

16. ライブラリ・キャッシュのヒット率の表示

```
SELECT namespace , sum_pinhits / sum_pins "Hit Rate"
FROM ( SELECT namespace , SUM ( pins ) AS sum_pins ,
      SUM ( pinhits ) AS sum_pinhits
      FROM v$librarycache GROUP BY namespace )
WHERE sum_pins > 0 ;
```

| NAMESPACE | Hit Rate |
|-----------------|------------|
| BODY | .991062695 |
| SQL AREA | .982209373 |
| TABLE/PROCEDURE | .97676927 |
| TRIGGER | .989047697 |
| INDEX | .701694915 |
| CLUSTER | .981092437 |

ライブラリ・キャッシュのヒット率の目安は、99%以上である
下回るのであれば、拡張を考える

ライブラリ・キャッシュ全体でのキャッシュ・ヒット率の表示

```
SELECT SUM( pinhits ) / sum ( pins ) * 100 AS "ヒット率"
FROM v$librarycache ;
```

17. ディクショナリ・キャッシュのヒット率の表示

```
SELECT parameter , SUM ( gets ) AS "gets" ,
      SUM ( getmisses ) AS "misses" ,
      100 * SUM ( gets - getmisses ) / SUM ( gets ) AS "Hit Rate"
FROM v$rowcache WHERE gets > 0 GROUP BY parameter ;
```

| PARAMETER | gets | misses | Hit Rate |
|------------------------|--------|--------|------------|
| dc_tablespace | 93276 | 8 | 99.9914233 |
| dc_tablespace_quotas | 4 | 1 | 75 |
| dc_awr_control | 587 | 1 | 99.8296422 |
| dc_object_grants | 6465 | 212 | 96.7208043 |
| dc_histogram_data | 29381 | 2796 | 90.4836459 |
| dc_rollback_segments | 6872 | 21 | 99.6944121 |
| dc_sequences | 60 | 13 | 78.3333333 |
| dc_segments | 14348 | 1827 | 87.266518 |
| dc_objects | 204232 | 5115 | 97.4954953 |
| dc_database_links | 420 | 1 | 99.7619048 |
| dc_histogram_defs | 45355 | 7003 | 84.5595855 |
| dc_users | 309788 | 163 | 99.9473834 |
| qmtmrcin_cache_entries | 1 | 1 | 0 |
| outstanding_alerts | 339 | 8 | 97.640118 |
| dc_files | 65 | 5 | 92.3076923 |
| dc_global_oids | 138689 | 241 | 99.8262299 |
| dc_profiles | 942 | 2 | 99.7876858 |
| global database name | 20215 | 1 | 99.9950532 |
| qmtmrctn_cache_entries | 6 | 6 | 0 |
| qmtmrciq_cache_entries | 1 | 1 | 0 |
| qmtmrctq_cache_entries | 309 | 309 | 0 |
| qmrc_cache_entries | 7 | 7 | 0 |

ディクショナリ・キャッシュのヒット率の目安は、**95%以上**である
下回るのであれば、拡張を考える

1 8．再解析頻度の多いSQLの抽出

```
SELECT  sql_text ,    parse_calls ,    executions
FROM    v$sqlarea
WHERE   executions < ( parse_calls * 2 )
ORDER  BY parse_calls DESC ;
```

| 列名 | 出力内容の説明 |
|--------------|-----------------------|
| sql_text | SQL 文の最初の 1000 文字 |
| sharable_mem | メモリ上に保持されている SQL のサイズ |
| disk_reads | SQL 文がディスクから取得したデータ量 |
| buffer_gets | SQL 文が取得したバッファの合計量 |
| parse_calls | SQL 文の解析コール数 |
| executions | SQL 文の実行コール数 |

実行回数が多くて、かつ、再解析の回数が多い SQL 文については、ライブラリ・キャッシュに常駐化させるようにする

1 9．処理時間の多いSQLの抽出

```
SELECT  sql_text FROM  v$sqlarea WHERE  address IN
( SELECT  sql_address FROM  v$sqlsession_longops ) ;
```

実行時間の長い SQL 文については、実行時間を短縮するような対策をとる
例えば、インデックス作成、SQL 文にヒント句を付けるなど

20. バッファ・キャッシュにキャッシュされているオブジェクトの名前と使用メモリ・ブロック数の表示

```
SELECT  o.object_name ,   count ( * )   number_of_blocks
FROM    dba_objects o ,   v$bh bh
WHERE   o.data_object_id   =   bh.objd
        AND  o.owner   != 'SYS'
GROUP   BY   o.object_name
ORDER   BY   COUNT ( * );
```

| OBJECT_NAME | NUMBER_OF_BLOCKS |
|---------------------|------------------|
| ----- | ----- |
| MGMT_LOADER_QTABLE | 380 |
| I_RLS | 278 |
| EMP_KEY | 128 |
| I_RULE_EC | 108 |
| I_PARTCOL\$ | 46 |
| I_KOPM1 | 29 |
| I_TS1 | 23 |
| WWV_FLOW_SESSIONS\$ | 12 |

※ ~~使用頻度が高くて、サイズが大きい~~ オブジェクトについては、メモリ（バッファ・キャッシュ）に常に確保されているように、KEEP バッファ・キャッシュでの使用に変更する

2 1. ライブラリ・キャッシュ内にキャッシュされているオブジェクトの統計情報とメモリ使用サイズとロード回数を表示

```
SELECT *  
FROM ( SELECT * FROM v$db_object_cache  
      WHERE owner NOT IN ('SYS','SYSTEM','PUBLIC')  
            AND type <> 'NOT LOADED'  
            AND sharable_mem <> 0  
      ORDER BY sharable_mem desc , executions desc )  
WHERE rownum <= 10 ;
```

OWNER : 所有者
NAME : オブジェクト名
DB_LINK :
NAMESPACE :
TYPE : オブジェクト・タイプ
 TABLE、PUB_SUB (待機中セッション)
SHARABLE_MEM : 使用メモリ・サイズ
LOADS : ロード回数
EXECUTIONS :
LOCKS : ロック回数
PINS :
KEPT : YES/NO
CHILD_LATCH :
INVALIDATIONS : 異常現象発生回数

2 2. **現在実行中の SGA サイズ変更** (拡張、縮小) 操作に関する情報を表示する

```
SELECT * FROM v$sga_current_resize_ops ;
```

| 名前 | NULL? | 型 |
|------------------|-------|--------------|
| COMPONENT | | VARCHAR2(64) |
| OPER_TYPE | | VARCHAR2(13) |
| OPER_MODE | | VARCHAR2(9) |
| PARAMETER | | VARCHAR2(80) |
| INITIAL_SIZE | | NUMBER |
| TARGET_SIZE | | NUMBER |
| CURRENT_SIZE | | NUMBER |
| START_TIME | | DATE |
| LAST_UPDATE_TIME | | DATE |

2 3 . SGA サイズ変更操作に関する過去記録の情報（最新 800 件）を表示する

```
SELECT * FROM v$sga_resize_ops ;
```

| COMPONENT | OPER_TYPE | OPER_MODE | PARAMETER |
|--------------------------|--------------|-----------|-----------------------|
| shared pool | STATIC | | shared_pool_size |
| Shared IO Pool | STATIC | | _shared_io_pool_size |
| java pool | STATIC | | java_pool_size |
| streams pool | STATIC | | streams_pool_size |
| DEFAULT buffer cache | INITIALIZING | | db_cache_size |
| ASM Buffer Cache | STATIC | | db_cache_size |
| DEFAULT buffer cache | STATIC | | db_cache_size |
| DEFAULT 2K buffer cache | STATIC | | db_2k_cache_size |
| DEFAULT 4K buffer cache | STATIC | | db_4k_cache_size |
| DEFAULT 8K buffer cache | STATIC | | db_8k_cache_size |
| DEFAULT 16K buffer cache | STATIC | | db_16k_cache_size |
| DEFAULT 32K buffer cache | STATIC | | db_32k_cache_size |
| KEEP buffer cache | STATIC | | db_keep_cache_size |
| RECYCLE buffer cache | STATIC | | db_recycle_cache_size |
| large pool | STATIC | | large_pool_size |

レコード項目のつづき

| INITIAL_SIZE | TARGET_SIZE | FINAL_SIZE | STATUS | START_TIME | END_TIME |
|--------------|-------------|------------|----------|------------|----------|
| 0 | 385875968 | 385875968 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 0 | 0 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 16777216 | 16777216 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 0 | 0 | COMPLETE | 17-10-26 | 17-10-26 |
| 654311424 | 654311424 | 654311424 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 0 | 0 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 654311424 | 654311424 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 0 | 0 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 0 | 0 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 0 | 0 | COMPLETE | 17-10-26 | 17-10-26 |
| 0 | 8388608 | 8388608 | COMPLETE | 17-10-26 | 17-10-26 |

—— PGA メモリに関する統計情報 ——

2 4. OEM (EMDC) による【PGA メモリ】のメモリ割当て量の最適化
アドバイス

2 ページ目の「6. 【PGA メモリ】のメモリ割当て量の最適化アドバイス」を参照

2 5. OEM (EMDC) による PGA メモリの作業領域サイズ別の実行割合
(最適実行、単一パス実行、マルチパス実行) の表示

・ 2 ページ目の「【PGA メモリ】の使用量詳細の表示」を参照

2 6. 起動してからのソート処理において、使用 SQL 作業領域別 (最適実行、
単一パス実行、マルチパス実行) の総実行回数

```
SELECT * FROM v$sql_workarea_histogram ;
```

| LOW_OPTIMAL_SIZE | | ONEPASS_EXECUTIONS | | MULTIPASSES_EXECUTIONS | |
|-------------------|--------|--------------------|-------|------------------------|-------|
| HIGH_OPTIMAL_SIZE | | OPTIMAL_EXECUTIONS | | TOTAL_EXECUTIONS | |
| ----- | ----- | ----- | ----- | ----- | ----- |
| 0 | 1023 | 0 | 0 | 0 | 0 |
| 1024 | 2047 | 0 | 0 | 0 | 0 |
| 2048 | 4095 | 10360 | 0 | 0 | 10360 |
| 4096 | 8191 | 2 | 0 | 0 | 2 |
| 8192 | 16383 | 0 | 0 | 0 | 0 |
| 16384 | 32767 | 0 | 0 | 0 | 0 |
| 32768 | 65535 | 0 | 0 | 0 | 0 |
| 65536 | 131071 | 12 | 0 | 0 | 12 |

この実行結果は、ソートで使用した作業領域の分布幅で表示される

| | | |
|--------------------------|---|------------------------|
| LOW_OPTIMAL_SIZE : | } | この 2 つの列で、ソートで使用した PGA |
| HIGH_OPTIMAL_SIZE : | | |
| OPTIMAL_EXECUTIONS : | | すべての処理メモリ上で行われたソート回数 |
| ONEPASS_EXECUTIONS : | | メモリ上だけでは行えず、最小限のディスク・ |
| | | アクセスが発生したソート回数 |
| MULTIPASSES_EXECUTIONS : | | 多くの処理において、メモリが不足して |
| | | ディスク・アクセスが発生したソート回数 |
| TOTAL_EXECUTIONS : | | 合計実行回数 |

2 7. PGA メモリでメモリ不足が発生したかの情報表示

【ソート実行でのメモリ・ソートとディスク・ソートの実行回数】

```
SELECT name, value FROM v$sysstat
       WHERE name LIKE 'workarea exec%' ;
```

| NAME | VALUE (回数) |
|---------------------------------|------------|
| workarea executions – optimal | 7238 |
| workarea executions – onepass | 23 |
| workarea executions – multipass | 0 |

↑

| neme 列の値 | この値の意味 | 目安 |
|------------------------------------|----------------------------------|-----------|
| workarea executions - optimal | ソートなどの処理がすべてメモリ上で行われた | 90% 以上 |
| workarea executions - onepass | メモリ上だけでは行えず、最小限のディスク・アクセスが発生した | 10% 未満 |
| workarea executions - multipass | 多くの処理において、メモリが不足してディスク・アクセスが発生した | 0 % |

上記の **SQL 文の実行結果が、この表の割合を満たしているか確認する**

ただし、特にデータが多数存在するソートを行う場合には、PGA の作業領域だけで実行すると他のメモリが不足してしまうことになるので、「マルチパスでの実行」も認めるような PGA の容量割当てを考えてよい

【ソート実行でのメモリとディスクの使用回数】

```
SELECT name, value FROM v$sysstat
       WHERE name IN ( 'sorts (memory)', 'sorts (disk)' ) ;
```

| NAME | VALUE (回数) |
|----------------|------------|
| sorts (memory) | 14009 |
| sorts (disk) | 43 |

※ 1 回のソート実行では、いろいろな領域が複数回使われて実行が行われる
これは、ソート実行回数ではなく、領域の使用回数から見た統計情報

28. PGA メモリのキャッシュ・ヒット率、設定メモリ量を超えた回数、推定メモリ必要量（すべてメモリ上で操作させた場合の想定量）の表示

（データベース起動時から現在までの統計）

```
SELECT name, value FROM v$pgastat;
```

| NAME | VALUE |
|--|-----------|
| aggregate PGA target parameter | 713031680 |
| aggregate PGA auto target | 590681088 |
| estimated PGA memory for one-pass | 999999999 |

すべての処理を最小限のディスク・アクセス(one-pass)だけで処理を行わせるための PGA_AGGREGATE_TARGET の推定のメモリ必要量設定値

この値が、PGA_AGGREGATE_TARGET を超えている場合は、設定値を大きくする

```
show parameter PGA_AGGREGATE_TARGET
```

| NAME | TYPE | VALUE |
|---------------------------------------|-------------|-----------|
| pga_aggregate_target | big integer | 0 |
| global memory bound | | 104857600 |
| total PGA inuse | | 6845312 |
| total PGA allocated | | 97417216 |
| maximum PGA allocated | | 131322880 |
| total freeable PGA memory | | 0 |
| process count | | 39 |
| max processes count | | 41 |
| PGA memory freed back to OS | | 0 |
| total PGA used for auto workareas | | 0 |
| maximum PGA used for auto workareas | 1411072 | |
| total PGA used for manual workareas | | 0 |
| maximum PGA used for manual workareas | | 0 |
| over allocation count | | 0 |

起動してから、PGA_AGGREGATE_TARGET の設定値を超えてメモリ割当てが行われた回数

この値が増加しているようであれば、PGA_AGGREGATE_TARGET を大きく設定する

| | |
|-----------------------------|-----------|
| bytes processed | 486083584 |
| extra bytes read/written | 0 |
| cache hit percentage | 100 |

起動してからの PGA メモリのキャッシュ・ヒット率

| | |
|-------------------------|------|
| recompute count (total) | 2042 |
|-------------------------|------|

29. 現在のセッションが、使用している PGA メモリの割当て状況の表示

```
SELECT  pga_used_mem ,  pga_freeable_mem ,  pga_alloc_mem ,  pga_max_mem
        FROM  v$process  ;
```

| PGA 使用メモリ | PGA 空きメモリ | PGA 確保メモリ量 | PGA 最大使用時メモリ量 |
|--------------|------------------|---------------|---------------|
| PGA_USED_MEM | PGA_FREEABLE_MEM | PGA_ALLOC_MEM | PGA_MAX_MEM |
| 437890 | 0 | 716366 | 716366 |
| 437082 | 0 | 716366 | 716366 |
| 433902 | 0 | 716366 | 716366 |
| 510246 | 0 | 781902 | 781902 |
| 437082 | 0 | 716366 | 716366 |
| 2309106 | 0 | 3673826 | 3673826 |
| . | . | . | . |
| . | . | . | . |

合計使用量の調査

```
SELECT  sum( pga_used_mem ) ,  sum( pga_freeable_mem ) ,
        sum( pga_alloc_mem ) ,  sum( pga_max_mem )
        FROM  v$process  ;
```

| SUM(PGA_USED_MEM) | SUM(PGA_FREEABLE_MEM) | SUM(PGA_ALLOC_MEM) | SUM(PGA_MAX_MEM) |
|-------------------|-----------------------|--------------------|------------------|
| 56155097 | 0 | 96339746 | 165340846 |

30. 現在の SQL カーソルに割当てられている PGA 作業領域の状況の表示

```
SELECT  *
        FROM  (  SELECT  workarea_address ,  operation_type ,  policy ,
                        estimated_optimal_size
                  FROM  v$sql_workarea  ORDER  BY  estimated_optimal_size )
        WHERE  rownum  <=  10  ;
```

| WORKAREA | OPERATION_TYPE | POLICY | ESTIMATED_OPTIMAL_SIZE |
|----------|-----------------|--------|------------------------|
| BAE3CF40 | SORT (v2) | AUTO | 1024 |
| BAF4D75C | SORT (v2) | AUTO | 1024 |
| BAD9CFD8 | SORT (v2) | AUTO | 1024 |
| B75A3C08 | SORT (v2) | AUTO | 1024 |
| B726CB90 | GROUP BY (SORT) | AUTO | 1024 |
| BAF0B15C | SORT (v2) | AUTO | 1024 |