

・ **オプティマイザの実行計画の作成に影響する初期化パラメータ**

オプティマイザの実行計画の作成に影響する「初期化パラメータ」

初期化パラメータ名	説明
PARALLEL_EXECUTION_ENABLED	
OPTIMIZER_FEATURES_ENABLE	
CPU_COUNT	
ACTIVE_INSTANCE_COUNT	
PARALLEL_THREADS_PER_CPU	
HASH_AREA_SIZE	
BITMAP_MERGE_AREA_SIZE	
SORT_AREA_SIZE	
SORT_AREA_RETAINED_SIZE	
DB_FILE_MULTIBLOCK_READ_COUNT	マルチブロックリードにおける 1 度の I/O 当たりの DB ブロックの読み込み個数
PGA_AGGREGATE_TARGET	
OPTIMIZER_MODE	最適化の目標方針 <ul style="list-style-type: none"> ・レスポンス重視 ・スループット重視 (このドキュメントに詳細説明)
CURSOR_SHARING	類似した SQL 文の Where 条件を、同一と認識させ、実行計画を共有させるかの基準
STAR_TRANSFORMATION_ENABLED	
OPTIMIZER_INDEX_CACHING	バッファキャッシュ効率の加味 (このドキュメントに詳細説明)
OPTIMIZER_INDEX_COST_ADJ	インデックス・アクセスのコスト加味 (このドキュメントに詳細説明)
QUERY_REWRITE_ENABLED	
QUERY_REWRITE_INTEGRITY	
WORKAREA_SIZE_POLICY	
OPTIMIZER_DYNAMIC_SAMPLING	動的サンプリング実施によるオブジェクト統計の取得
STATISTICS_LEVEL	
SKIP_UNUSABLE_INDEXES	
_OPTIM_PEEK_USER_BINDS	バインドピーク機能の使用の可否

オブティマイザが参照する実行計画の作成のための 「初期化パラメータ」の表示

確認できる値は、上記表（↑）のパラメータ値

ただし、これらの値は、システムレベル、セッションごと、SQL ごとに異なる値を取っている場合があるので、以下のように確認先を選択する

- ・システムレベルでの設定値の確認先： V\$**SYS**_OPTIMIZER_ENV
- ・セッションで使用される設定値の確認先： V\$**SES**_OPTIMIZER_ENV
- ・SQL で使用された設定値の確認先： V\$**SQL**_OPTIMIZER_ENV

```
Select NAME, ISDEFAULT, VALUE From V$SYS_OPTIMIZER_ENV;
```

```
Select NAME, ISDEFAULT, VALUE From V$SES_OPTIMIZER_ENV;
```

```
Select sql_id, sql_fulltext
from v$sqlarea
where sql_fulltext like '%Plan_Search%';
```

SQL_ID	SQL_FULLTEXT
6njqx8z5umwmr	select /* Plan_Search */ * from emp where empno=1

```
Select NAME, ISDEFAULT, VALUE
From V$SQL_OPTIMIZER_ENV
Where SQL_ID = '6njqx8z5umwmr';
```

NAME	ISD	VALUE
cpu_count	YES	4
hash_area_size	YES	131072
sort_area_size	YES	65536
optimizer_mode	YES	all_rows
cursor_sharing	YES	exact
optimizer_index_cost_adj	YES	100
optimizer_index_caching	YES	0
workarea_size_policy	YES	auto
optimizer_dynamic_sampling	YES	2
statistics_level	YES	typical
optimizer_secure_view_merging	YES	true
result_cache_mode	YES	MANUAL
optimizer_capture_sql_plan_baselines	YES	false
optimizer_use_sql_plan_baselines	YES	true
optimizer_use_invisible_indexes	YES	false

CPU 能力と I/O 処理能力を考慮した実行計画の作成方法

Oracle のオプティマイザが実行計画を作成するときの評価項目は、ディスクへの I/O アクセスの回数だけをコスト計算し最も少ない実行計画を採用されてきた

Oracle10 以降は、これに加えて CPU の処理速度も加味してコスト計算して、最適な実行計画が作成されている（デフォルトとして採用）

この新たに加わったコスト計算決定モデルは、『CPU+I/O コストモデル』と呼ばれています

『CPU+I/O コストモデル』について

『CPU+I/O コストモデル』を使用するには、CPU 計算と I/O アクセスに関しての実際の処理速度が実行計画を比較判断するに必要となります

このため、CPU 計算と I/O アクセスに関しての情報を、実際に Oracle インスタンスが稼働しているサーバーから収集する必要があります

CPU 計算と I/O アクセスに関しての情報が収集されていない場合でも、実行計画作成においての最適計画の判断は『CPU+I/O コストモデル』のロジックで行われるが、計算に使用される統計情報がセットされていないので、I/O コストモデルの結果と同一となってしまう

必要となるシステム統計情報の採取方法

採取手順)

1. Oracle インスタンスに**ワークロード（通常使用時の負荷）**をかける
2. `exec dbms_stats.gather_system_stats('start');`
3. `exec dbms_lock.sleep(300);`
4. `exec dbms_stats.gather_system_stats('stop');`

CPU 計算と I/O アクセスに関して、必要となるシステム統計情報 とその値の表示方法

Select PNAME, PVAL1, PVAL2 From sys.aux_stats\$;

PNAME	PVAL1	PVAL2
CPUSPEEDNW	1845.90945	
IOSEEKTIM	10	
IOTFRSPEED	4096	
SREADTIM		
MREADTIM		
CPUSPEED		
MBRC		
MAXTHR		
SLAVETHR		

システム統計情報項目	内 容
CPUSPEEDNW	1 秒あたりの平均サイクル数 (単位 : MHz) Oracle からみた相対的な CPU 速度 ※ CPU クロック数とは、異なる
IOSEEKTIM	ディスクリード当たりのオーバーヘッド時間 (ミリ秒) ディスクシーク時間 + ディスク回転遅延時間 + OS オーバーヘッド時間
IOTFRSPEED	I/O 転送速度 (単位 : バイト / ミリ秒) Read 時間 = IOSEEKTIME + (read サイズ ÷ IOTFRSPEED)
SREADTIM	平均シングルブロックリード時間 (単位 : ミリ秒) 単位ブロックのランダムリードの平均レスポンス
MREADTIM	平均マルチブロックリード時間 (単位 : ミリ秒) 複数ブロックのシーケンシャルな読み込みの平均レスポ ンス
CPUSPEED	1 秒当たりの平均サイクル数
MBRC	マルチブロックリード時の 1 回の平均読み込みブロック数
MAXTHR	I/O サブシステムが提供可能な最大 I/O スループット (単位 : バイト / ミリ秒)
SLAVETHR	パラレルスレーブごとの平均 I/O スループット (単位 : バイト / ミリ秒)

実行計画全体のコストと CPU コスト、I/O コストとの関係

$$\text{COST} = (\text{IO_COST} \times \text{SREADTIM} + \text{CPU_CUST} \div (\text{CPUSPEED} \times 1000)) \div \text{SREADTIM}$$

実行計画の表示

```
explain plan for
select * from emp;
@?/rdbms/admin/utlxpls
```

PLAN_TABLE_OUTPUT

Plan hash value: 3956160932

Id	Operation	Name	Rows	Bytes	Cost (CPU%)
0	SELECT STATEMENT		50	1400	120 (22)
1	TABLE ACCESS FULL	EMP	50	1400	120 (22)

【CPU コストを考慮しない場合の実行計画】

```
explain plan for
select /*+ NO_CPU_COSTING */ * from emp;
@?/rdbms/admin/utlxpls
```

PLAN_TABLE_OUTPUT

Plan hash value: 3956160932

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT		50	1400	2
1	TABLE ACCESS FULL	EMP	50	1400	2

Note

- cpu costing is off (consider enabling it)

・実行計画の作成計算でのバッファキャッシュ効率の加味のための
のキャッシュ・ヒットに関する補正パラメータ

実行計画の作成計算でのバッファキャッシュ効率の加味のためのキャッシュ・ヒットに関する補正パラメータ

これらのパラメータの設定は、計算で求めるのではなく、
セットする値を適宜変更して、実際の業務のレスポンス状態をみて**最適値を試行錯誤で見つける**ようにする

※ 指定する項目は、下記のどちらかの初期化パラメータだけをセットする

OPTIMIZER_INDEX_CACHING

索引ブロックが何パーセントくらいキャッシュ・ヒットすると想定するかの指定値
デフォルト 0
OLTP 環境下での指定値（目安） 90

OPTIMIZER_INDEX_COST_ADJ

インデックス・アクセスのコストを通常の何パーセントと想定するかの指定値

（ 上のようにインデックスが、ある程度キャッシュ・ヒットすることを想定して、そのコストを通常の読み込みに対してどの位になるかを示した%値での指定 ）

デフォルト 100
OLTP 環境下での指定値（目安） 25

※ 推奨値を使用すれば、索引インデックス読みの実施確率が高くなる

- ・ データベースが重視する最適化方針
(レスポンス or スループット) の指定

データベースが重視する最適化方針 **OPTIMIZER_MODE** の指定
(レスポンス or スループット)

スループット重視型: **ALL_ROWS** (デフォルト設定値)

レスポンス重視型: **FIRST_ROWS**

FIRST_ROWS_n (n=1, 10, 100 を指定)

最初の何行を返すのを優先するか行数指定

※ **FIRST_ROWS** を指定した場合は、レスポンス重視となるため全体効率はテーブル・フルスキャンの方が効率が良くても、インデックス・スキャンが選択され実行し、全体効率が落ちてしまうので、使用には注意すること

OPTIMIZER_MODE は、**ALTER SESSION** 文でセッション内での変更や **SQL** ヒント句での **SQL** 文内での変更を行うことが可能です

使用例)

```
ALTER SESSION set OPTIMIZER_MODE = 'FIRST_ROWS_100';
```

```
SHOW PARAMETER OPTIMIZER_MODE
```