

過去に実行された SQL 文に対する実行計画の調べ方  
SQL 文が実行された事後に、使用された実行計画と実行統計の調査方法  
SQL 文に割当てられた SQL\_ID の調べ方

過去に（指定した時刻）実行されていた SQL 文の洗い出し  
過去に（指定した時刻）実行されていた SQL 文の実行統計情報  
過去に（指定した時刻）実行されていたロック状態の調査  
過去に（指定した時刻）発生していた待機イベントの件数

## 調査対象となる元データ

- A) SGA メモリ内のバッファキャッシュ
- B) システム表領域に残っている記録（Active Session History 情報）
- C) Statspack スナップショットによって記録されている実行実績統計
- D) AWR スナップショットによって記録されている実行実績統計

## 調査できる内容

- A) SGA メモリ内のバッファキャッシュ
  - ・実行計画の出力
  - ・実行実績統計情報の出力
- B) システム表領域に残っている記録（Active Session History 情報）
  - ・実行計画の出力
  - ・実行実績統計情報の出力
  - ・指定時刻における待機イベント発生件数の把握
  - ・指定時刻におけるロック状態の表示
- C) Statspack スナップショットによって記録されている実行実績統計
  - ・実行計画の出力
  - ・実行実績統計情報の出力
  - ・スナップショット間の実行実績

#### D) AWR スナップショットによって記録されている実行実績統計

- 実行計画の出力
- 実行実績統計情報の出力
- スナップショット間の実行実績

使用する動的ビューの名前については、  
実行するスクリプトのソースコードを  
確認のこと

## 履歴保存ポイント（時刻）の一覧

```
alter session set NLS_DATE_FORMAT = 'yyyy/mm/dd hh24:mi:ss';
alter session set NLS_TIMESTAMP_FORMAT = 'yyyy/mm/dd hh24:mi:ss';
```

### A) ASH (Active Session History) 記録

```
select distinct sample_time
from v$active_session_history
order by sample_time;
```

SAMPLE_TIME
-----
2018/07/30 09:20:10
2018/07/30 09:20:20
2018/07/30 09:30:40

### B) DBA\_HIST 記録

```
select distinct sample_time
from dba_hist_active_sess_history
order by sample_time;
```

実行結果は、A) と同じ

### C) Statspack スナップショット

```
select snap_id, snap_time
from stats$snapshot
order by snap_id;
```

SNAP_ID	SNAP_TIME
-----	-----
1	2018/01/22 10:00:00
2	2018/01/22 11:00:00
3	2018/01/22 12:00:00

### D) AWR スナップショット

```
col begin_interval_time format a20
col end_interval_time format a20
select dbid, instance_number, snap_id, begin_interval_time,
end_interval_time
from dba_hist_snapshot
order by dbid, instance_number, snap_id;
```

SNAP_ID	BEGIN_INTERVAL_TIME	END_INTERVAL_TIME
-----	-----	-----
668	2018/07/30 09:14:46	2018/07/30 09:25:53
669	2018/07/30 09:25:53	2018/07/30 10:00:04
670	2018/07/30 10:00:04	2018/07/30 11:00:35

## 実行計画の表示

- A) キャッシュ・メモリ上にある実行計画
- B) DBA\_HIST 履歴テーブルに記録されている実行計画
- C) Statspack スナップショットにて記録されている実行計画
- D) AWR スナップショットにて記録されている実行計画

## 実行統計情報の表示

- A) キャッシュ・メモリ上にある実行統計情報
- B) DBA\_HIST 履歴テーブルに記録されている実行統計情報
- C) Statspack スナップショットにて記録されている実行統計情報
- D) AWR スナップショットにて記録されている実行統計情報

## 発生イベント件数の状況

### A) ASH (Active Session History) 記録

```
BREAK ON serach_time
VARIABLE search_date1 varchar2(20);
VARIABLE search_date2 varchar2(20);
EXECUTE :search_date1 := '2018/08/08 10:00:00';
EXECUTE :search_date2 := '2018/08/08 10:20:00';

select TO_CHAR( sample_time, 'yyyy/mm/dd hh24:mi:ss') serach_time ,
       event, count(event)
from v$active_session_history
where TO_CHAR( sample_time, 'yyyy/mm/dd hh24:mi:ss')
      between :search_date1 and :search_date2
group by sample_time , event
order by sample_time ASC , count(event) DESC ;
```

SERACH_TIME	EVENT	COUNT(EVENT)
2018/08/08 10:16:12	db file sequential read	20
	db file scattered read	13
	control file sequential read	4
2018/08/08 10:16:13	db file sequential read	11
2018/08/08 10:16:14	db file sequential read	15
	db file scattered read	8

### B) DBA\_HIST 記録

### C) Statspack スナップショット

Statspack レポートに指定期間のスナップショット全体での上位発生件数イベントが出力される

### D) AWR スナップショット

AWR レポートに指定期間のスナップショット全体での上位発生件数イベントが出力される

## 実行されていたロック状態の調査

### A) ASH (Active Session History) 記録

### B) DBA\_HIST 記録

### ~~C) Statspack スナップショット~~

### ~~D) AWR スナップショット~~

## SQL 文に割当てられた SQL\_ID の調べ方

### 方法 A.

対象の SQL 文は、AWR レポートから負荷の高い SQL 文を選択したのであれば、そこに SQL\_ID が記述されている

### 方法 B.

DBA\_HIST\_SQLTEXT ビューに SQL テキストを条件指定して問い合わせる  
最近実行の SQL 文ならば、V\$ACTIVE\_SESSION\_HISTORY と V\$SQL を使う

```
select sql_id from DBA_HIST_SQLTEXT
       where sql_text like '文字列%' ;
```

v\$SQL ビューの中から、sql\_text 列が SQL 文の文字列と一致するレコードを検索する  
(続・門外不出の Oracle 現場ワザ P. 12)

```
select sql_id from DBA_HIST_SQLTEXT
       where UPPER(sql_text) like '文字列%' ;
```

例) 

```
select sql_id from DBA_HIST_SQLTEXT
       where UPPER(sql_text) like 'SELECT /* LIST07_1 */%' ;
       'select /* list07_1 */%' ;
```

SQL\_ID

-----  
4cqayrarbsfd

### 方法 C.

DBA\_HIST\_ACTIVE\_SESS\_HISTORY もしくは、V\$ACTIVE\_SESSION\_HISTORY  
ビューへ実行時刻をを条件指定して問い合わせる P. 422

```
select TO_CHAR(ASH.SAMPLE_TIME, 'YYYY/MM/DD HH24:MI:SS') SDATE ,
       ASH.SESSION_ID , US.USERNAME , ASH.SQL_ID ,
       ASH.EVENT , ASH.TIME_WAITED , ASH.BLOCKING_SESSION
from DBA_HIST_ACTIVE_SESS_HISTORY ASH , DBA_USERS US
where ASH.SAMPLE_TIME
      between TO_DATE('2018/08/06 12:00:00', 'YYYY/MM/DD HH24:MI:SS')
      and TO_DATE('2018/08/06 14:01:00', 'YYYY/MM/DD HH24:MI:SS')
      and ASH.USER_ID = US.USER_ID
order by ASH.SAMPLE_TIME asc ;
```

## 過去に実行された SQL 文の実行計画と実行統計の調べ方

方法 A)

DBMS\_XPLAN パッケージを使って問合せを行う

```
sql> select * from TABLE( DBMS_XPLAN.DISPLAY_AWR(  
    '確認したい SQL_ID', NULL, NULL, 'ALL'));
```

方法 B)

AWR SQL レポートを出力する

( \$ORACLE\_HOME/rdbms/admin/awrsqrpt.sql スクリプトを使う )

```
sql> conn / as sysdba
```

```
sql> @?/ rdbms/admin/awrsqrpt.sql
```

AWR の SNAP\_ID > SNAP\_ID 番号の入力

↑

確認したい SQL が実行されている時間の SNAP\_ID

SQL\_ID > SQL\_ID の入力

← 上記で調べた SQL\_ID 番号

前ページのスクリプト実行により、『過去の SQL の実行計画やパフォーマンス統計、待機イベントのサマリ情報』について書かれたファイルが、カレントディレクトリに出力される

過去（指定した時間に）実行されていた SQL 文の洗い出し  
 過去（指定した時間に）発生していた待機イベントの調査

DBA\_HIST\_ACTIVE\_SESS\_HISTORY ビュー

もしくは、V\$ACTIVE\_SESSION\_HISTORY からの確認 P. 422

```
select TO_CHAR(ASH.SAMPLE_TIME, 'YYYY/MM/DD HH24:MI:SS') SDATE ,
       ASH.SESSION_ID , US.USERNAME , ASH.SQL_ID ,
       ASH.EVENT , ASH.TIME_WAITED , ASH.BLOCKING_SESSION
from   DBA_HIST_ACTIVE_SESS_HISTORY ASH , DBA_USERS US
where  ASH.SAMPLE_TIME
       between TO_DATE('2018/08/06 12:00:00', 'YYYY/MM/DD HH24:MI:SS')
       and    TO_DATE('2018/08/06 14:01:00', 'YYYY/MM/DD HH24:MI:SS')
       and ASH.USER_ID = US.USER_ID
order by ASH.SAMPLE_TIME asc ;
```

SDATE	SESSION_ID	USER	実行中の SQL ↓ SQL_ID	発生していた待機イベント ↓ EVENT
11/12/20 12:00:10	121	KOZUE	ft52k1ie5a	enq:TX – row lock contention
11/12/20 12:00:10	37	IKURA	hj34k112pe	SQL*Net message from client

  

TIME_WAITED	BLOCKING_SESSION
11776 0	37



## USER\_ID からユーザー名の調査方法

```
select A.USERNAME, B.NAME, A.user_ID, b.user#  
from dba_users A, sys.user$ B  
where A.USERNAME(+) = B.NAME  
order by A.user_ID;
```

USERNAME	NAME	USER_ID	USER#
SYS	SYS	0	0
SYSTEM	SYSTEM	5	5
OUTLN	OUTLN	9	9
KOZUE	KOZUE	12	12
ITOKUTORA	ITOKUTORA	13	13
TSMSYS	TSMSYS	14	14