

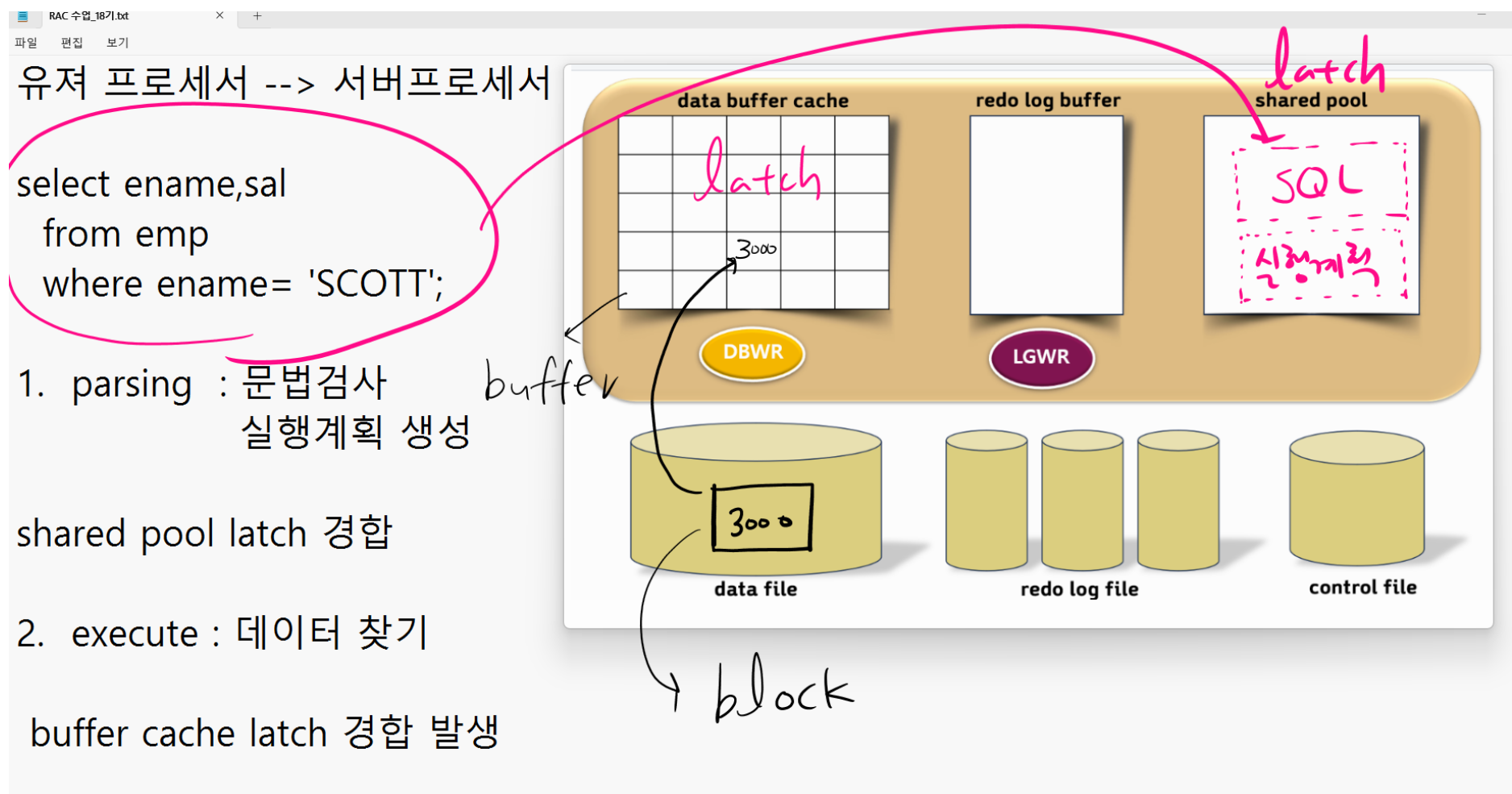
60. buffer cache latch 경합을 줄이기 위한 튜닝

buffer cache latch 경합이 발생하는 이유

=> 데이터를 검색하는 select 문을 수행하면 서버 프로세서는 먼저 그 데이터를 메모리에서 찾고 없으면 data file에서 찾아서 메모리에 올려둔다. 메모리에 올려두는 이유는 다음에 또 이 데이터를 검색하려 하면 바로 메모리에서 바로 가져다 줄려고 메모리에 올려놓는다.

이 메모리가 바로 db buffer cache 다.

=> (아래 그림) execute 단계에서 'SCOTT'의 데이터를 찾는데 메모리에 없으면 data file에서 찾아서 메모리에 올린다. 이 때 buffer cache latch를 확보해야 올릴 수 있다. 그런데 동시에 buffer cache를 사용하려는 세션들이 많게 되면 latch 경합이 발생하게 된다.



실습 1.

1. owi 유저로 접속한다.
connect owi/owi

2. 사진을 찍는다.
(sql 1 owi)
@snap

3. buffer cache latch 경합을 일으킨다.
@exec

cache_buffers_chains_latch

=> 새로운 창을 열어서 @event 실행

```

racdb1(SYS) > @event

SID EVENT
-----
136 gc buffer busy acquire
9 gc cr request
17 jobq slave wait
16 jobq slave wait
139 jobq slave wait
254 jobq slave wait
11 latch free
252 latch free
254 latch free
370 latch free
371 latch free

SID EVENT
-----
12 latch free
137 latch free
128 latch free

```

4. 사진을 찍는다.

@snap

5. awr report를 생성한다.

@?/rdbms/admin/awrrpt.sql

html

89

90

=> latch free 대기 이벤트가 나오고 있다. 이 대기 이벤트는 buffer cache 쪽에 관련한 대기 이벤트이다. 이 경우의 해결 방법은 latch free 경합을 일으키는 SQL을 찾아서 튜닝하는 방법으로 해결을 하면 된다.

```

88 02 Apr 2024 09:38 1
89 02 Apr 2024 10:23 1
90 02 Apr 2024 10:29 1

Specify the Begin and End Snapshot Ids
~~~~~
Enter value for begin_snap: 89
Begin Snapshot Id specified: 89

Enter value for end_snap: 90
End Snapshot Id specified: 90

```

Top 10 Foreground Events by Total Wait Time

| Event | Waits | Total Wait Time (sec) | Wait Avg(ms) | % DB time | Wait Class |
|-------------------------------|-----------|-----------------------|--------------|-----------|-------------|
| DB CPU | | 118.1 | | 31.4 | |
| db file sequential read | 11,219 | 6.5 | 1 | 1.7 | User I/O |
| db file scattered read | 202 | 1.5 | 8 | .4 | User I/O |
| enq: TX - row lock contention | 46 | 1.5 | 33 | .4 | Application |
| control file sequential read | 246 | 1.5 | 6 | .4 | System I/O |
| cursor: pin S wait on X | 24 | 1.5 | 62 | .4 | Concurrency |
| gc cr grant 2-way | 5,532 | 1.2 | 0 | .3 | Cluster |
| gc current block busy | 30 | .7 | 24 | .2 | Cluster |
| row cache lock | 429 | .3 | 1 | .1 | Concurrency |
| <u>latch free</u> | <u>76</u> | <u>.3</u> | 4 | .1 | Other |

6. addm report를 생성한다.

@?/rdbms/admin/addmrpt.sql

89

90

report0402.txt

```
88 02 Apr 2024 09:38 1
89 02 Apr 2024 10:23 1
90 02 Apr 2024 10:29 1

Specify the Begin and End Snapshot Ids
~~~~~
Enter value for begin_snap: 89
Begin Snapshot Id specified: 89

Enter value for end_snap: 90
End Snapshot Id specified: 90

Specify the Report Name
~~~~~
The default report file name is addmrpt_1_89_90.txt. To use this name,
press <return> to continue, otherwise enter an alternative.

Enter value for report_name: report0402.txt
```

```
Finding 1: Top SQL Statements
Impact is .97 active sessions, 96% of total activity.
-----
SQL statements consuming significant database time were found. These
statements offer a good opportunity for performance improvement.

Recommendation 1: SQL Tuning
Estimated benefit is .97 active sessions, 96% of total activity.
-----
Action
Run SQL Tuning Advisor on the SELECT statement with SQL_ID
"q2vdqw3b66177".
Related Object
SQL statement with SQL_ID q2vdqw3b66177.
SELECT /*+ index(t idx_cache_buffers_chains) */ COUNT(*) FROM
T_CACHE_BUFFERS_CHAINS_LATCH T WHERE T.TYPE = '4' AND T.NAME = 'KIM'
AND T.ID = '4'
Rationale
The SQL spent 100% of its database time on CPU, I/O and Cluster waits.
This part of database time may be improved by the SQL Tuning Advisor.
Rationale
Database time for this SQL was divided as follows: 100% for SQL
execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java
execution.
Rationale
SQL statement with SQL_ID "q2vdqw3b66177" was executed 3078 times and
had an average elapsed time of 0.079 seconds.

Finding 2: "User I/O" wait Class
Impact is .02 active sessions, 2.14% of total activity.
-----
Wait class "User I/O" was consuming significant database time.
The throughput of the I/O subsystem was not significantly lower than expected.
The Oracle instance memory (SGA and PGA) was adequately sized.

No recommendations are available.
```

=> ADDM report가 알려준 sql 튜닝이 필요한 sql을 위의 sql id를 가지고 찾아낸다.

```
select sql_text
from v$sql
where sql_id='q2vdqw3b66177';
```

=> 메모리에서 사라져서 찾을 수 없지만 찾았으면 sql 튜닝을 하면 해결 된다.

SQL 튜닝 외에 또 다른 해결 방법

1. db_cache_size 파라미터를 조정해서 buffer cache를 더 크게 늘린다.
2. db_block_lru_latches 파라미터를 조정해서 latch의 개수를 늘린다.

실습 2. db_block_lru_latches파라미터 조정하기

1. 현재 db_block_lru_latches파라미터의 개수를 확인한다.

@hidden

```
-----  
latch  
-----
```

| Parameter | Session Value | Instance Value |
|--|---------------|----------------|
| ----- | ----- | ----- |
| _disable_latch_free_SCN_writes_via_32cas | FALSE | FALSE |
| _disable_latch_free_SCN_writes_via_64cas | FALSE | FALSE |
| db_block_lru_latches | 32 | 32 |
| db_block_hash_latches | 8192 | 8192 |
| db_blocks_per_hash_latch | | |

2. 양쪽 인스턴스의 db_block_lru_latches파라미터의 개수를 늘린다.

```
alter system set "_db_block_lru_latches"=40 scope=spfile sid='*';
```

3. 양쪽 인스턴스를 내렸다가 올린다.

(sql 1, sql 2)

```
shutdown immediate
```

```
startup
```

| Parameter | Session Value | Instance Value |
|--|---------------|----------------|
| ----- | ----- | ----- |
| _disable_latch_free_SCN_writes_via_32cas | FALSE | FALSE |
| _disable_latch_free_SCN_writes_via_64cas | FALSE | FALSE |
| db_block_lru_latches | 40 | 40 |
| db_block_hash_latches | 8192 | 8192 |
| db_blocks_per_hash_latch | | |