

❁ 49. RAC 환경에서 발생하는 SQ enqueue 성능 문제 해결하는 방법 1 (nocache 일 때)

실습 1. 환경 구성

```
(sql 1)
drop user owi cascade;
drop tablespace owi_tbs including contents and datafiles;
alter system set db_recovery_file_dest_size=30G;
```

```
(1번 노드 rman)
delete backup;
delete copy;
backup database;
```

```
(sql 1)
@install
```

```
-----
+data
on
-----
```

실습 2. sequence 속성 nocache 성능 테스트

1. owi 사용자가 가지고 있는 sequence를 확인하고 속성을 조회하시오.

```
(sql 1 owi)
select sequence_name, cache_size, order_flag
from user_sequences;
```

```
racdb1(OWI) > select sequence_name, cache_size, order_flag
from user_sequences;
 2
SEQUENCE_NAME                CACHE_SIZE 0
-----
SCENARIO_NO_SEQ                20 N
SEQ_SQ_ENQUEUE                20 N
```

```
alter sequence seq_sq_enqueue nocache;
```

```
select sequence_name, cache_size, order_flag
from user_sequences;
```

```
racdb1(OWI) > select sequence_name, cache_size, order_flag
from user_sequences;  2
SEQUENCE_NAME                CACHE_SIZE 0
-----
SCENARIO_NO_SEQ                20 N
SEQ_SQ_ENQUEUE                0 N
```

2. lock 관련한 대기 이벤트로 대기하는 세션을 조회한다.

```
(sql 1 owi)
select sid, event
from gv$session_wait
where event like '%lock%'
or event like '%enqueue%';
```

```
racdb1(OWI) > select sid, event
from gv$session_wait
where event like '%lock%'
or event like '%enqueue%'; 2    3    4
no rows selected
```

3. 터미널 창을 하나 더 열어서 sequence 관련 성능 부하를 일으킨다. 실행 후, 다른 터미널 창에서 lock 관련한 대기 이벤트로 대기하는 세션을 조회한다.

```
(sql 1 owi)
```

```
@exec
```

```
-----
sq_enqueue
-----
```

```
(sql 1 owi)
```

```
select sid, event
from gv$session_wait
where event like '%lock%'
or event like '%enqueue%';
```

=> 아래의 그림에 row cache lock은 db에 있는 sequence의 번호를 획득하지 못하여 waiting하는 것

```
Scenario List
-----
dfs_lock_handle
generic

13 rows selected.

Elapsed: 00:00:00.10
Event name to simulate: sq_enqueue
Session count [10]:
Expired by time(1) or looping count(2) [1]:
Execution interval(sec or count) [30]:
Enable_trace (1=TRUE, 0=FALSE) [0]:
Exec method(0=Oracle Job, 1=Unix Shell) [0]:
Init data(1=TRUE, 0=FALSE) [1]:

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.07

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.63
racdb1(OWI) > racdb1(OWI) >
racdb1(OWI) > █
```

4. 192.168.56.111 (oracle)

```
from gv$session_wait
where event like '%lock%'
or event like '%enqueue%'; 2    3    4
no rows selected
```

```
racdb1(OWI) > select sid, event
from gv$session_wait
where event like '%lock%'
or event like '%enqueue%'; 2    3    4
```

```

SID EVENT
-----
15 row cache lock
16 row cache lock
129 row cache lock
137 row cache lock
141 row cache lock
257 row cache lock
259 row cache lock
371 row cache lock
373 row cache lock
375 row cache lock
10 rows selected.
```

```
create sequence seq_sq_enqueue  
start with 1  
maxvalue 10000  
increment by 1  
nocache  
order;
```

=> 위와 같이 sequence를 생성하면 은행에 갔는데 많은 사람들이 번호표 기계 앞에서 다투고 있는 상황이 된다. 그러지 않은 상황이 되려면 nocache가 아니라 cache로 해줘야 한다.