

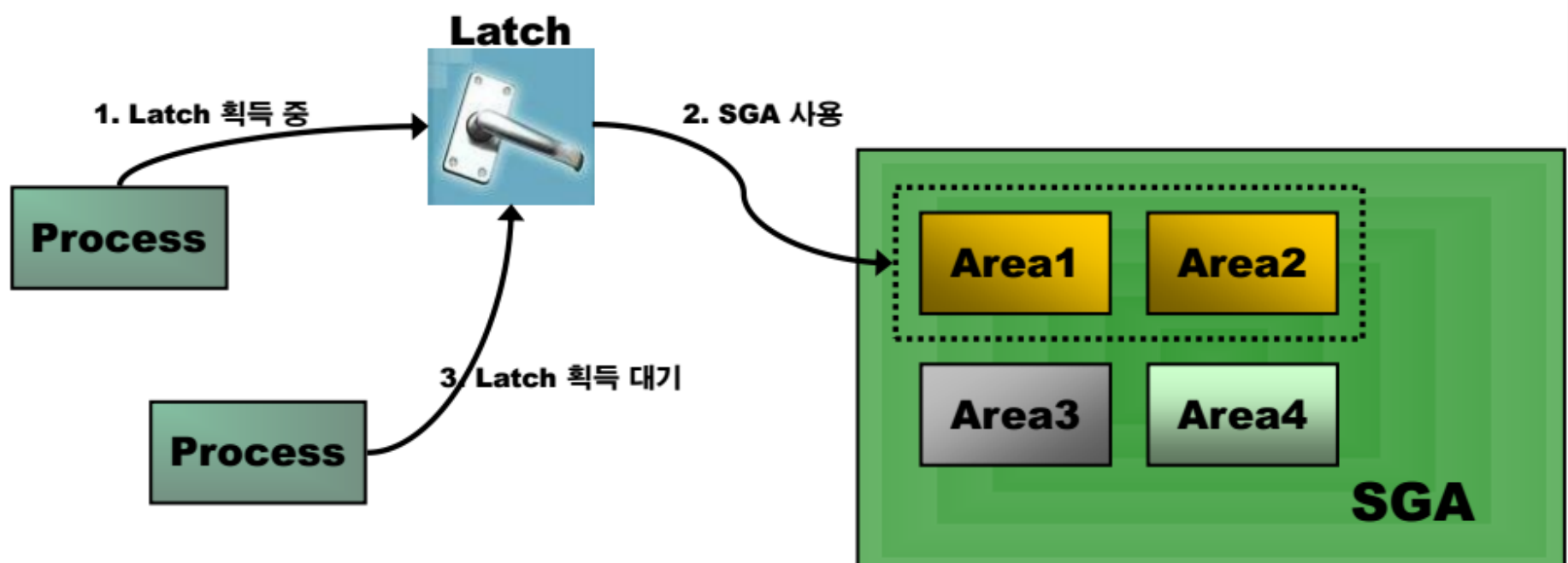
## 🌸 54. Shared Pool 관련한 latch

오라클 메모리 성능에 이슈 2가지

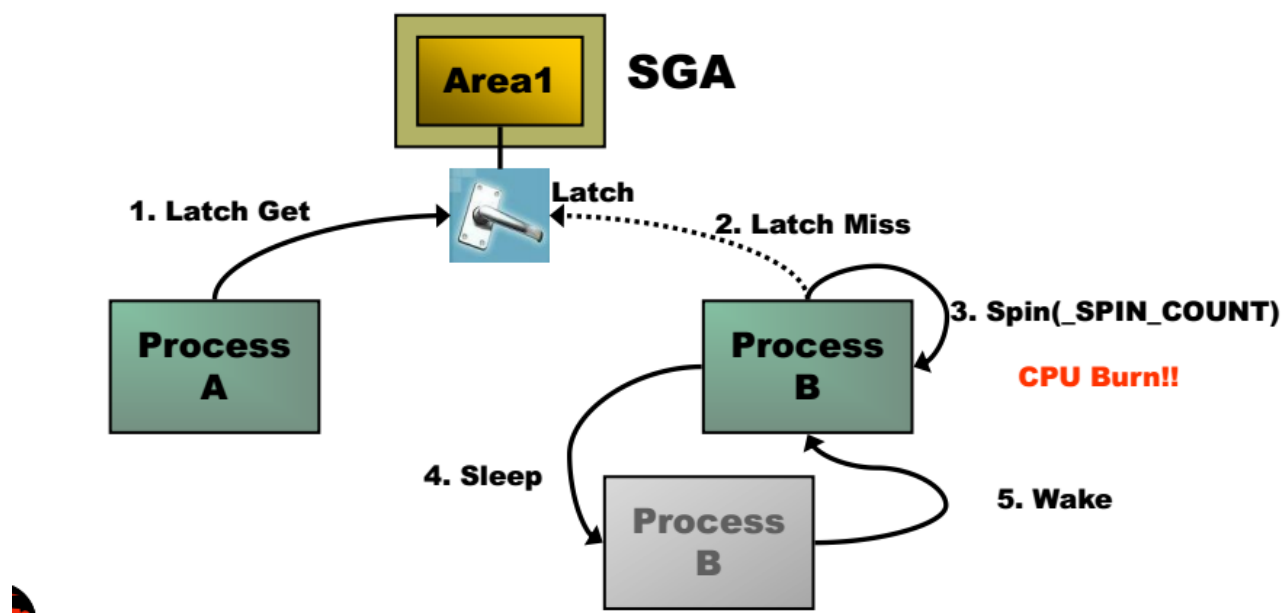
1. enqueue
2. latch

### Latch

- ✓ 가벼운 Lock(Light-weight Lock)
- ✓ 공유 Memory 영역(SGA)을 보호하기 위한 동기화 객체



### Spin에 의한 Latch 획득



=> latch를 획득해야지만 SGA 영역 메모리를 사용할 수 있게 되는 것이다.

Latch를 획득해야 하는 경우

1. shared pool의 경우 -> SQL을 parsing 하기 위해
2. db buffer cache의 경우 -> data file에서 data 블록을 메모리에 올리기 위해

※ 불필요한 parsing을 최소화 하는 방법 이론 설명

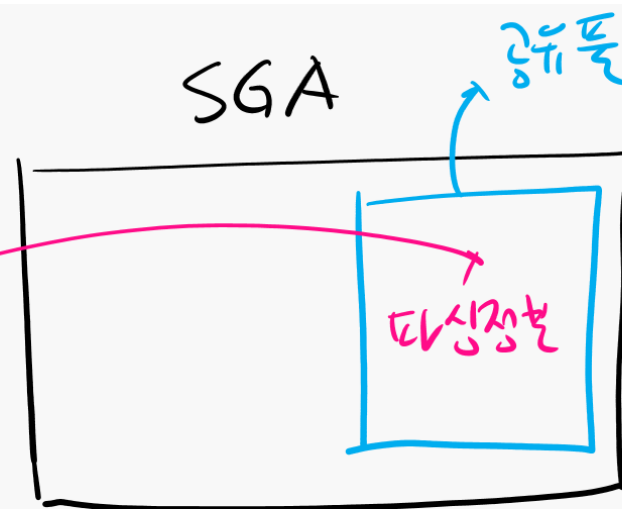
```
select ename, sal
from emp
where ename = 'SCOTT';
```

1. Parsing -> 문법 검사, 의미 검사
2. execute -> 데이터 찾는 과정
3. fetch -> 데이터 전달

\* 불필요한 파싱을 최소화하는 방법 이론 설명

```
select ename, sal
from emp
where ename='SCOTT';
```

파싱정보



1. Parsing ---> 문법검사, 의미검사
2. execute ---> 데이터 찾는 과정
3. fetch ---> 데이터 전달

❌ 다음이 똑같은 문장이  
들기엔 parsing 생략  
(soft parse)

- hard parse + 리시딩  
- soft parse + 파싱 x

실습- 불필요한 parsing을 최소화 하는 방법

1. 사진을 찍는다.

(sql 1)

@snap

2. hard parsing을 일으키는 부하를 일으킨다.

DECLARE

TYPE rc IS REF cursor;

l\_rc rc;

l\_dummy all\_objects.object\_name%TYPE;

l\_start NUMBER DEFAULT dbms\_utility.get\_time;

BEGIN

FOR i IN 1 .. 10000

loop

OPEN l\_rc FOR

'select object\_name from all\_objects where object\_id = ' || i;

fetch l\_rc INTO l\_dummy;

close l\_rc;

END loop;

dbms\_output.put\_line ( round( (dbms\_utility.get\_time - l\_start)/100, 2) || 'seconds');

END;

/

(새로운 1번 노트 창)

top

=> CPU를 과도하게 사용하는 프로세서가 하나 있다.

```
[oracle@racdb1 ~]$ top
top - 11:14:47 up 1:40, 3 users, load average: 1.19, 0.88, 0.78
Tasks: 294 total, 3 running, 291 sleeping, 0 stopped, 0 zombie
Cpu(s): 22.9%us, 0.9%sy, 0.0%ni, 75.3%id, 0.7%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 4045608k total, 3074480k used, 971128k free, 121084k buffers
Swap: 4194300k total, 0k used, 4194300k free, 1436080k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6445	oracle	20	0	1851m	268m	255m	R	97.6	6.8	1:26.58	oracle
3793	grid	-2	0	1306m	43m	41m	S	1.7	1.1	3:07.27	oracle

(sql 1)  
@event

```
racdb1(SYS) > @event
SID EVENT
-----
364 latch: shared pool
```

(sql 1)  
select sql\_text  
from v\$sql  
where sql\_text like 'select object\_name%';

=> 공유 풀의 SQL 확인

=> 공유 풀 안에 공유 되지 않은 SQL이 가득 차 있다. 공유가 안 되어 있다는 것은 다시 parsing 했다는 뜻이다. parsing을 하기 위해서 CPU를 많이 소모했다는 뜻이다. CPU를 100% 가까이 사용하는 세션이 서버에 있으면 엄청 느려진다. 그 원인을 찾아서 해결해야 한다.

```
racdb1(SYS) > select sql_text
from v$sql
where sql_text like 'select object_name%'; 2 3
SQL_TEXT
-----
select object_name from all_objects where object_id = 7765
select object_name from all_objects where object_id = 7770
select object_name from all_objects where object_id = 7795
select object_name from all_objects where object_id = 7783
select object_name from all_objects where object_id = 7752
select object_name from all_objects where object_id = 7744
select object_name from all_objects where object_id = 7798
select object_name from all_objects where object_id = 7787
select object_name from all_objects where object_id = 7756
select object_name from all_objects where object_id = 7785
select object_name from all_objects where object_id = 7759
SQL_TEXT
-----
select object_name from all_objects where object_id = 7766
select object_name from all_objects where object_id = 7800
select object_name from all_objects where object_id = 7760
select object_name from all_objects where object_id = 7790
select object_name from all_objects where object_id = 7779
select object_name from all_objects where object_id = 7769
select object_name from all_objects where object_id = 7742
select object_name from all_objects where object_id = 7738
select object_name from all_objects where object_id = 7761
select object_name from all_objects where object_id = 7751
select object_name from all_objects where object_id = 7780
```

3. 사진을 찍는다.

@snap

4. AWR report를 생성한다.

@?/rdbms/admin/awrrpt.sql

-----  
html

79

80

```

77 01 Apr 2024 09:50      1
78 01 Apr 2024 11:00      1
79 01 Apr 2024 11:11      1
80 01 Apr 2024 11:22      1

Specify the Begin and End Snapshot Ids
~~~~~
Enter value for begin_snap: 79
Begin Snapshot Id specified: 79

Enter value for end_snap: 80
End Snapshot Id specified: 80

Specify the Report Name
~~~~~
The default report file name is awrrpt_1_79_80.html. To use this name,
press <return> to continue, otherwise enter an alternative.

Enter value for report_name: report0401.html

```

	Begin	End
Memory Usage %:	85.16	83.30
% SQL with executions>1:	78.81	63.10
% Memory for SQL w/exec>1:	64.80	21.38

## Main Report

- [Report Summary](#)
- [Wait Events Statistics](#)
- [SQL Statistics](#)
- [Instance Activity Statistics](#)
- [IO Stats](#)
- [Buffer Pool Statistics](#)
- [Advisory Statistics](#)
- [Wait Statistics](#)
- [Undo Statistics](#)
- [Latch Statistics](#)
- [Segment Statistics](#)
- [Dictionary Cache Statistics](#)
- [Library Cache Statistics](#)
- [Memory Statistics](#)
- [Streams Statistics](#)
- [Resource Limit Statistics](#)
- [Shared Server Statistics](#)
- [init.ora Parameters](#)

## Wait Events Statistics

- [Time Model Statistics](#)
- [Operating System Statistics](#)
- [Operating System Statistics - Detail](#)
- [Foreground Wait Class](#)
- [Foreground Wait Events](#)
- [Background Wait Events](#)
- [Wait Event Histogram](#)
- [Wait Event Histogram Detail \(64 msec to 2 sec\)](#)
- [Wait Event Histogram Detail \(4 sec to 2 min\)](#)
- [Wait Event Histogram Detail \(4 min to 1 hr\)](#)
- [Service Statistics](#)
- [Service Wait Class Stats](#)

[Back to Top](#)

## Time Model Statistics

## Time Model Statistics

- Total time in database user-calls (DB Time): 380.6s
- Statistics including the word "background" measure background process time, and so do not contribute to the DB time statistic
- Ordered by % of DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	380.32	99.92
parse time elapsed	372.56	97.88
hard parse elapsed time	369.12	96.97
DB CPU	366.63	96.32
PL/SQL compilation elapsed time	10.60	2.79
PL/SQL execution elapsed time	0.51	0.13
connection management call elapsed time	0.17	0.05
hard parse (sharing criteria) elapsed time	0.13	0.03
sequence load elapsed time	0.01	0.00
repeated bind elapsed time	0.01	0.00
hard parse (bind mismatch) elapsed time	0.00	0.00
DB time	380.64	
background elapsed time	28.12	
background cpu time	15.03	

=> 위의 그림을 통해 hard parsing이 문제라는 것을 알 수 있다. shared pool 문제.

5. ADDM report를 생성한다.

```
@?/rdbms/admin/addmrpt.sql
```

```
-----
```

```
79
```

```
80
```

```
report0401.txt
```

```
-----
```

```

Enter value for begin_snap: 79
Begin Snapshot Id specified: 79

Enter value for end_snap: 80
End Snapshot Id specified: 80

Specify the Report Name
The default report file name is addmrpt_1_79_80.txt. To use this name,
press <return> to continue, otherwise enter an alternative.

Enter value for report_name: report0401.txt
Using the report name report0401.txt

Running the ADDM analysis on the specified pair of snapshots ...

Generating the ADDM report for this analysis ...

      ADDM Report for Task 'TASK_329'
      -----

Analysis Period
-----
AWR snapshot range from 79 to 80.
Time period starts at 01-APR-24 11.11.24 AM
Time period ends at 01-APR-24 11.22.01 AM

Analysis Target
-----
Database 'RACDB' with DB ID 1148479398.
Database version 11.2.0.4.0.
ADDM performed an analysis of instance racdb1, numbered 1 and hosted at rac1.

Activity During the Analysis Period
-----
Total database time was 381 seconds.
The average number of active sessions was .6.

Summary of Findings
-----

```

Description	Active Sessions Percent of Activity	Recommendations
1 Hard Parse Due to Literal Usage	.58   96.94	1
2 CPU Usage	.02   3.5	1
3 PL/SQL Compilation	.02   2.79	1

ADDM report의 추천 해결 방법은 SQL이 공유가 될 수 있도록 Literal SQL을 Bind Variable로 변경 하라는 것이다.

- literal SQL의 예

```
select empno, ename, sal from emp where empno=7788;
```

```
select empno, ename, sal from emp where empno=7902;
```

```
select empno, ename, sal from emp where empno=7566;
```

=> 뒤의 사원 번호만 틀려도 오라클은 다른 SQL로 인식해서 다시 parsing 한다. 그래서 튜닝 방법이 바인드 변수로 변경 해줘야 한다.

```
select empno, ename, sal from emp where empno =: v_empno;
```

6. 사진을 찍는다.

```
@snap
```

7. 공유가 되는 튜닝된 부하를 일으킨다.

```
DECLARE
```

```
TYPE rc IS REF cursor;
```

```
l_rc rc;
```

```
l_dummy all_objects.object_name%TYPE;
```

```
l_start NUMBER DEFAULT dbms_utility.get_time;
```

```
BEGIN
```



```

FOR i IN 1 .. 10000
loop
OPEN l_rc FOR
'select object_name from all_objects where object_id =:x' USING i;
fetch l_rc INTO l_dummy;
close l_rc;
END loop;
dbms_output.put_line ( round( (dbms_utility.get_time - l_start)/100, 2) || 'seconds');
END;
/

```

8. 사진을 찍는다.

@snap

9. AWR 비교 report를 생성한다.

@?/rdbms/admin/awrddrpt.sql

-----

79

80

81

82

awr\_report0401.html

-----

=> AWR 비교 report를 보니 CPU time이 튜닝 전과 후가 차이나게 줄어들었습니다.

ADDM report가 권장하는 해결 방법 2가지

1. PL/SQL logic에서 literal SQL을 바인드 변수로 변경하기

2. cursor\_sharing을 force로 변경하면 알아서 모든 literal SQL을 오라클이 바인드 변수로 자동으로 변경 해준다.

```

74 28 Mar 2024 17:19 1
75 28 Mar 2024 17:22 1
76 28 Mar 2024 17:23 1

77 01 Apr 2024 09:50 1
78 01 Apr 2024 11:00 1
79 01 Apr 2024 11:11 1
80 01 Apr 2024 11:22 1
81 01 Apr 2024 11:56 1
82 01 Apr 2024 11:59 1

Specify the Second Pair of Begin and End Snapshot Ids
~~~~~
Enter value for begin_snap2: 81
Second Begin Snapshot Id specified: 81

Enter value for end_snap2: 82

```

## Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time(ms)	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time(ms)	%DB time
CPU time			366.63		96.32	control file sequential read	System I/O	594	3.22	5.42	36.33
control file sequential read	System I/O	2,013	8.20	4.07	2.15	CPU time			1.77		19.99
db file parallel write	System I/O	442	6.63	14.99	1.74	log file parallel write	System I/O	68	1.44	21.23	16.30
db file sequential read	User I/O	103	2.15	20.85	0.56	gc current block busy	Cluster	20	1.40	70.09	15.83
log file parallel write	System I/O	93	0.89	9.56	0.23	reliable message	Other	10	1.28	127.95	14.45
control file parallel write	System I/O	223	0.72	3.23	0.19	db file sequential read	User I/O	48	1.07	22.23	12.05
gc current block 2-way	Cluster	775	0.51	0.66	0.13	gcs log flush sync	Other	60	0.84	13.94	9.44
row cache lock	Concurrency	2,659	0.46	0.17	0.12	gc cr block busy	Cluster	4	0.48	119.28	5.39
gc current block busy	Cluster	33	0.38	11.37	0.10	direct path write	User I/O	11	0.14	12.36	1.54
gcs log flush sync	Other	42	0.28	6.66	0.07	row cache lock	Concurrency	762	0.12	0.16	1.36
-gc cr block busy	Cluster	6	0.14	23.06	0.04	-control file parallel write	System I/O	50	0.11	2.19	1.23
-reliable message	Other	33	0.04	1.32	0.01	-gc current block 2-way	Cluster	131	0.08	0.61	0.90
-						-db file parallel write	System I/O	1	0.00	0.53	0.01

## Report Details

문제 1. 다음의 Literal SQL을 scott 유저에서 수행하고 공유풀에 올라가 있는지 확인하시오.

(sql 1 scott)

```
select empno, ename, sal from emp where empno=7788;
```

```
select empno, ename, sal from emp where empno=7902;
```

```
select sql_text
```

```
from v$sql
```

```
where sql_text like 'select empno%';
```

```
racdb1(SCOTT) > select empno, ename, sal from emp where empno=7788;
```

EMPNO	ENAME	SAL
7788	SCOTT	3000

```
racdb1(SCOTT) > select empno, ename, sal from emp where empno=7902;
```

EMPNO	ENAME	SAL
7902	FORD	3000

```
racdb1(SCOTT) > select sql_text  
from v$sql  
where sql_text like 'select empno%'; 2 3
```

```
SQL_TEXT
```

```
select empno, ename, sal from emp where empno=7902  
select empno, ename, sal from emp where empno=7788
```