# Oracle中查询被锁的表和杀掉session

1.如何查看session级的等待事件?

当我们对数据库的性能进行调整时，一个最重要的参考指标就是系统等待事 件。

$system_event,v$session_event,v$session_wait这三个视图里记录的就是系统级和session级的等待 事件，通过查询这些视图你可以发现数据库的一些操作到底在等待什么，是磁盘I/O，缓冲区忙，还是插锁等等。

查询你的每个应用程序到底在等待什么，从而针对这些信息对数据库的性能进行调整 ：

```
select s.username,s.program,s.status,se.event,se.total_waits,se.total_timeouts,se.time_waited,se.average_wait from v$session s,
v$session_event se Where s.sid=se.sid And se.event not like 'SQl*Net%' And s.status ='ACTIVE' And s.username is not null;
```

2.oracle中查询被锁的表并释放session

```
SELECT A.OWNER,
A.OBJECT_NAME,
B.XIDUSN,
B.XIDSLOT,
B.XIDSQN,
B.SESSION_ID,
B.ORACLE_USERNAME,
B.OS_USER_NAME,
B.PROCESS,
B.LOCKED_MODE,
C.MACHINE,
C.STATUS,
C.SERVER,
C.SID,
C.SERIAL#,
C.PROGRAM
FROM ALL_OBJECTS A, V$LOCKED_OBJECT B, SYS.GV_$SESSION C
WHERE (A.OBJECT_ID = B.OBJECT_ID)
AND (B.PROCESS = C.PROCESS)
ORDER BY 1, 2;
```

释放session ：

```
alter system kill session 'sid, serial#'
eg : alter system kill session '30, 2412';
```

3.查看占用系统io较大的session ：

```
SELECT se.sid,
se.serial#,
pr.SPID,
se.username,
se.status,
se.terminal,
se.program,
se.MODULE,
se.sql_address,
st.event,
st.p1text,
si.physical_reads,
si.block_changes
FROM v$session se,    v$session_wait st, v$sess_io si, v$process pr
WHERE st.sid = se.sid   AND st.sid = si.sid
AND se.PADDR = pr.ADDR
AND se.sid > 6  AND st.wait_time = 0
AND st.event NOT LIKE '%SQL%'
ORDER BY physical_reads DESC;
```

4.找出耗cpu较多的session :

```
select a.sid,spid,status,substr(a.program,1,40) prog,a.terminal,osuser,value/60/100 value
from v$session a,v$process b,v$sesstat c
where c.statistic#=12 and c.sid=a.sid and a.paddr=b.addr order by value desc;
```

5. 查询session被锁的 :

```
select sys.v_$session.osuser,sys.v_$session.machine,v$lock.sid,
    sys.v_$session.serial#,
    decode(v$lock.type,
    'MR', 'Media Recovery',
    'RT','Redo Thread',
    'UN','User Name',
    'TX', 'Transaction',
    'TM', 'DML',
    'UL', 'PL/SQL User Lock',
    'DX', 'Distributed Xaction',
    'CF', 'Control File',
    'IS', 'Instance State',
    'FS', 'File Set',
    'IR', 'Instance Recovery',
    'ST', 'Disk Space Transaction',
    'TS', 'Temp Segment',
    'IV', 'Library Cache Invalida-tion',
    'LS', 'Log Start or Switch',
    'RW', 'Row Wait',
    'SQ', 'Sequence Number',
    'TE', 'Extend Table',
    'TT', 'Temp Table',
    'Unknown') LockType,
    rtrim(object_type) || ' ' || rtrim(owner) || '.' || object_name object_name,
    decode(lmode, 0, 'None',
    1, 'Null',
    2, 'Row-S',
    3, 'Row-X',
    4, 'Share',
    5, 'S/Row-X',
    6, 'Exclusive', 'Unknown') LockMode,
    decode(request, 0, 'None',
    1, 'Null',
    2, 'Row-S',
    3, 'Row-X',
    4, 'Share',
    5, 'S/Row-X',
    6, 'Exclusive', 'Unknown') RequestMode,
    ctime, block b
    from v$lock, all_objects, sys.v_$session
    where v$Lock.sid > 6 and sys.v_$session.sid = v$lock.sid
    and v$lock.id1 = all_objects.object_id;
```

6. OS一级for kill 处理Oracle中杀不掉的锁

如果利用上面的命令杀死一个进程后，进程状态被置为"killed"，但是锁定的资源很长时间没有被释放，

那么可以在os一级再杀死相应

```
--1 查询session被锁的sql,简要查询,得到SID
select object_name,machine,s.sid,s.serial#
from v$locked_object l,dba_objects o ,v$session s
where l.object_id  =  o.object_id and l.session_id=s.sid;
```

--2 使用alter system kill session '24,111'; (其中24,111分别是上面查询出的sid,serial#)进行释放

```
alter system kill session '30, 2412'

--3 执行下面的语句获得进程（线程）号,sid为第一步查询出的sid号：
select spid, osuser, s.program
from v$session s,v$process p
where s.paddr=p.addr and s.sid=30;
/*==========================================================================
```

4.在OS上杀死这个进程（线程）：

1)在unix上，用root身份执行命令:

#kill -9 12345（即第3步查询出的spid）

2)在windows（unix也适用）用orakill杀死线程，orakill是oracle提供的一个可执行命令，语法为：

orakill sid thread

其中：

sid：表示要杀死的进程属于的实例名

thread：是要杀掉的线程号，即第3步查询出的spid。

例：c:>orakill orcl 12345

参考 ： https://www.cnblogs.com/tracy/archive/2011/08/04/2127461.html