背景：如果出现系统界面卡慢，同时伴随着工单流转慢，app操作卡慢或频繁报错的现象，可考虑是数据库性能问题。

### 数据库主机

10.188.38.20-23

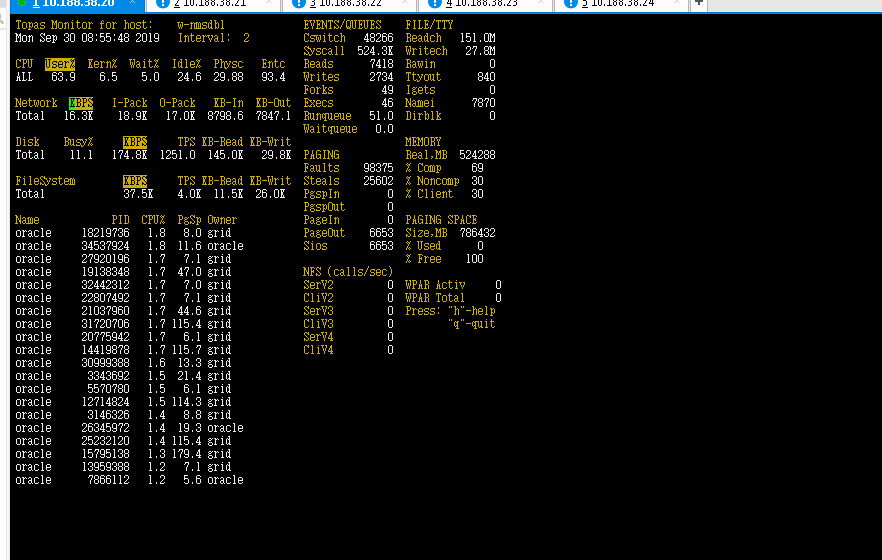
10.188.38.24（负载节点）

账号密码： oralce/OraN541GeZz

数据库主机使用的是AIX系统，许多linux下的命令无法使用。

查看cpu占比等指标：topas

一般情况下，数据库主机的cpu使用占比不会长时间超过70%以上。



### 查询数据库io情况

gv$session\_wait表中能查询到当前等待执行的sql所处于的事件，用以下sql查询出当前处于各事件的sql数量。

SELECT event,

SUM(DECODE(wait\_Time, 0, 0, 1)) "Prev",

SUM(DECODE(wait\_Time, 0, 1, 0)) "Curr",

COUNT(\*) "Tot"

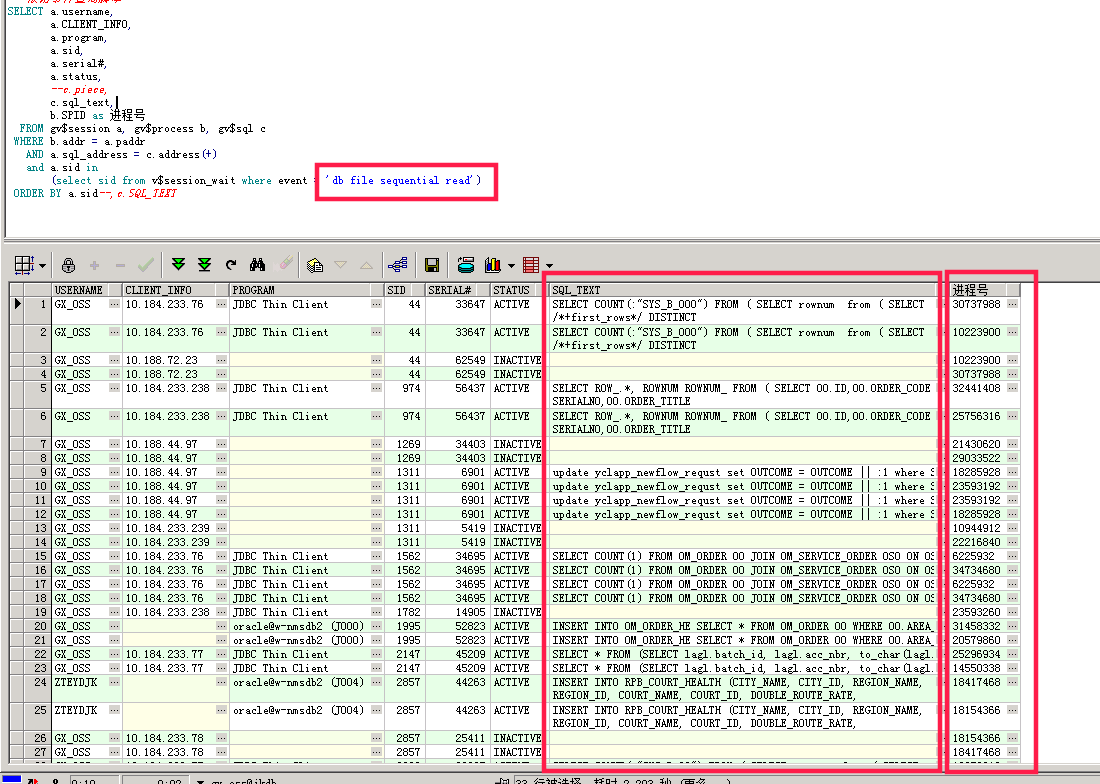
FROM gv$session\_wait

WHERE wait\_class != 'Idle'

GROUP BY event

ORDER BY 4 DESC

将等待数量多的事件名取出，用以下sql查出正处于该事件下的sql。根据查出的sql推断出导致卡慢的服务，后再分析是否需要重启服务或是杀死sql。



-- 根据事件查询脚本

SELECT a.username,

a.CLIENT\_INFO,

a.program,

a.sid,

a.serial#,

a.status,

--c.piece,

c.sql\_text,

b.SPID as 进程号,

'kill -9 '|| b.SPID

FROM gv$session a, gv$process b, gv$sql c

WHERE b.addr = a.paddr

AND a.sql\_address = c.address(+)

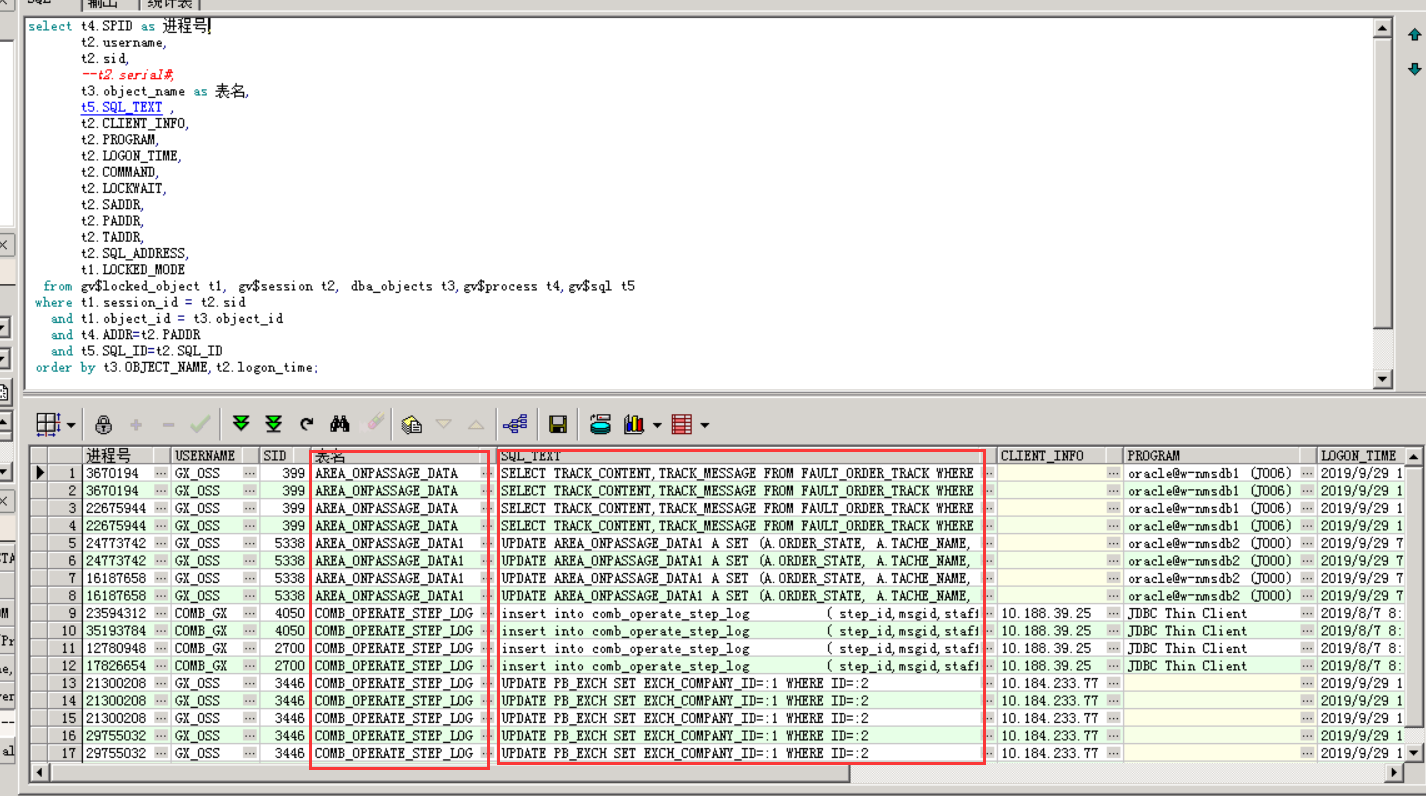
and a.sid in

(select sid from v$session\_wait where event = 'db file sequential read')

ORDER BY a.sid--,c.SQL\_TEXT

### 查询锁表情况

（1）如果出现大量长时间的锁表，如某个表出现某个表同时有几十上百个锁，就可能是某个应用将频繁操作某个表，或着是表本身出现问题。



--------查询所有的锁表情况

select t4.SPID as 进程号,

t2.username,

t2.sid,

--t2.serial#,

t3.object\_name ,

t5.SQL\_TEXT ,

t2.CLIENT\_INFO,

t2.PROGRAM,

t2.LOGON\_TIME,

t2.COMMAND,

t2.LOCKWAIT,

t2.SADDR,

t2.PADDR,

t2.TADDR,

t2.SQL\_ADDRESS,

t1.LOCKED\_MODE

from gv$locked\_object t1, gv$session t2, dba\_objects t3,gv$process t4,gv$sql t5

where t1.session\_id = t2.sid

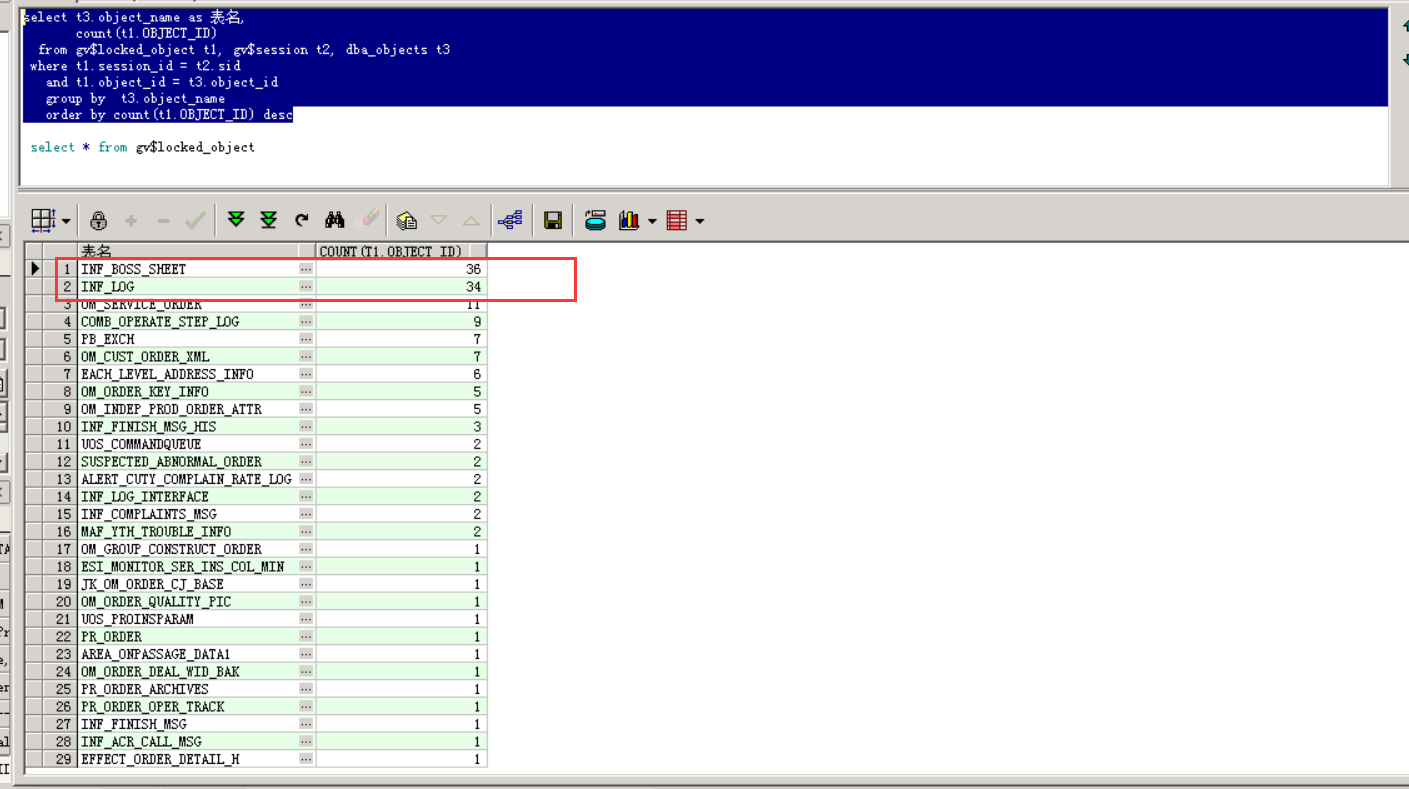
and t1.object\_id = t3.object\_id

and t4.ADDR=t2.PADDR

and t5.SQL\_ID=t2.SQL\_ID

order by t3.OBJECT\_NAME,t2.logon\_time;

1. 如果出现大量的锁表，用以下sql可以直观地查询是否存在所有表的锁表数量



--查看各个表的锁表数量

select t3.object\_name as 表名,

count(t1.OBJECT\_ID)

from gv$locked\_object t1, gv$session t2, dba\_objects t3

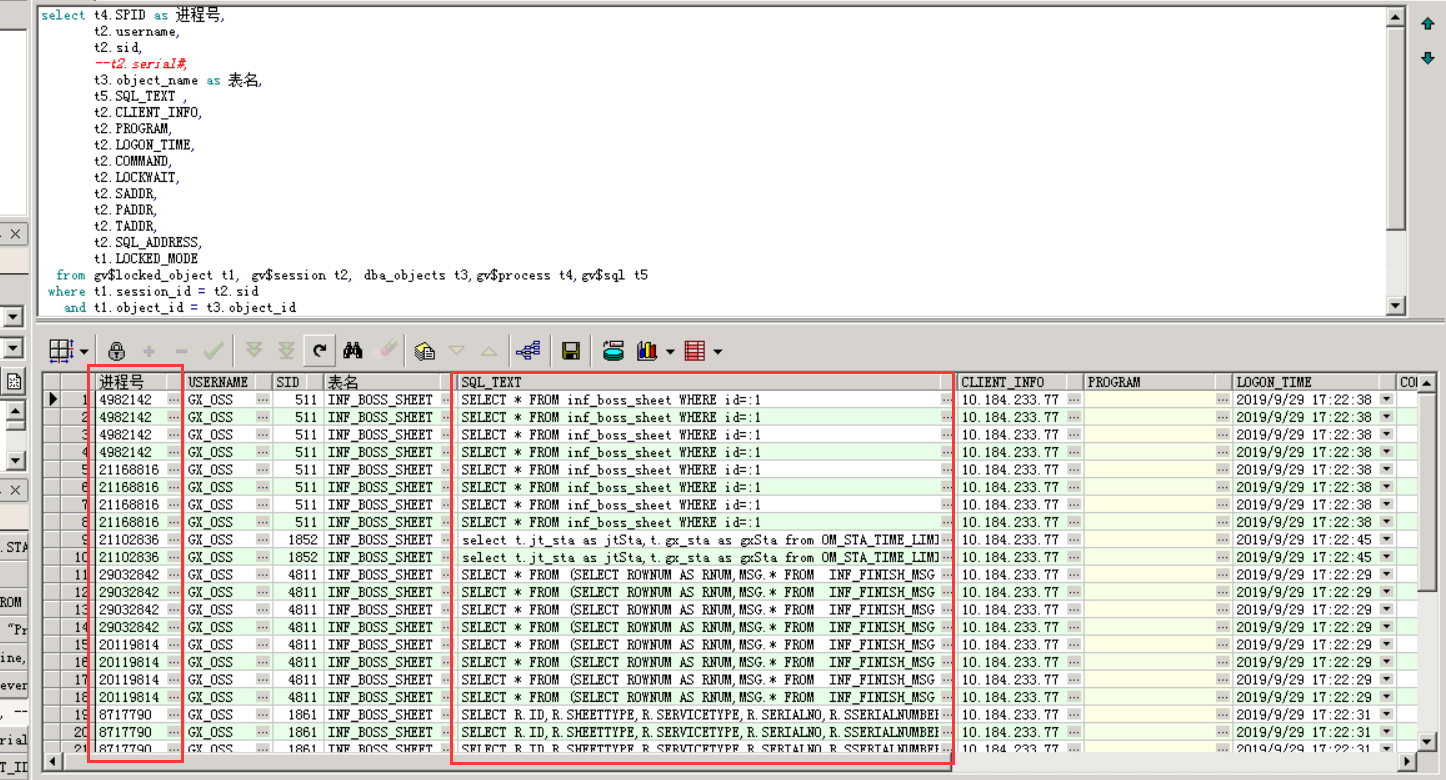
where t1.session\_id = t2.sid

and t1.object\_id = t3.object\_id

group by t3.object\_name

order by count(t1.OBJECT\_ID) desc

1. 取出锁表数量大的表名，用以下sql查询造成锁表的sql，进而可以根据sql分析造成锁表的服务。进而考虑是否需要杀掉缩表的sql或重启相应服务。



select t4.SPID as 进程号,

t2.username,

t2.sid,

--t2.serial#,

t3.object\_name as 表名,

t5.SQL\_TEXT ,

t2.CLIENT\_INFO,

t2.PROGRAM,

t2.LOGON\_TIME,

t2.COMMAND,

t2.LOCKWAIT,

t2.SADDR,

t2.PADDR,

t2.TADDR,

t2.SQL\_ADDRESS,

t1.LOCKED\_MODE

from gv$locked\_object t1, gv$session t2, dba\_objects t3,gv$process t4,gv$sql t5

where t1.session\_id = t2.sid

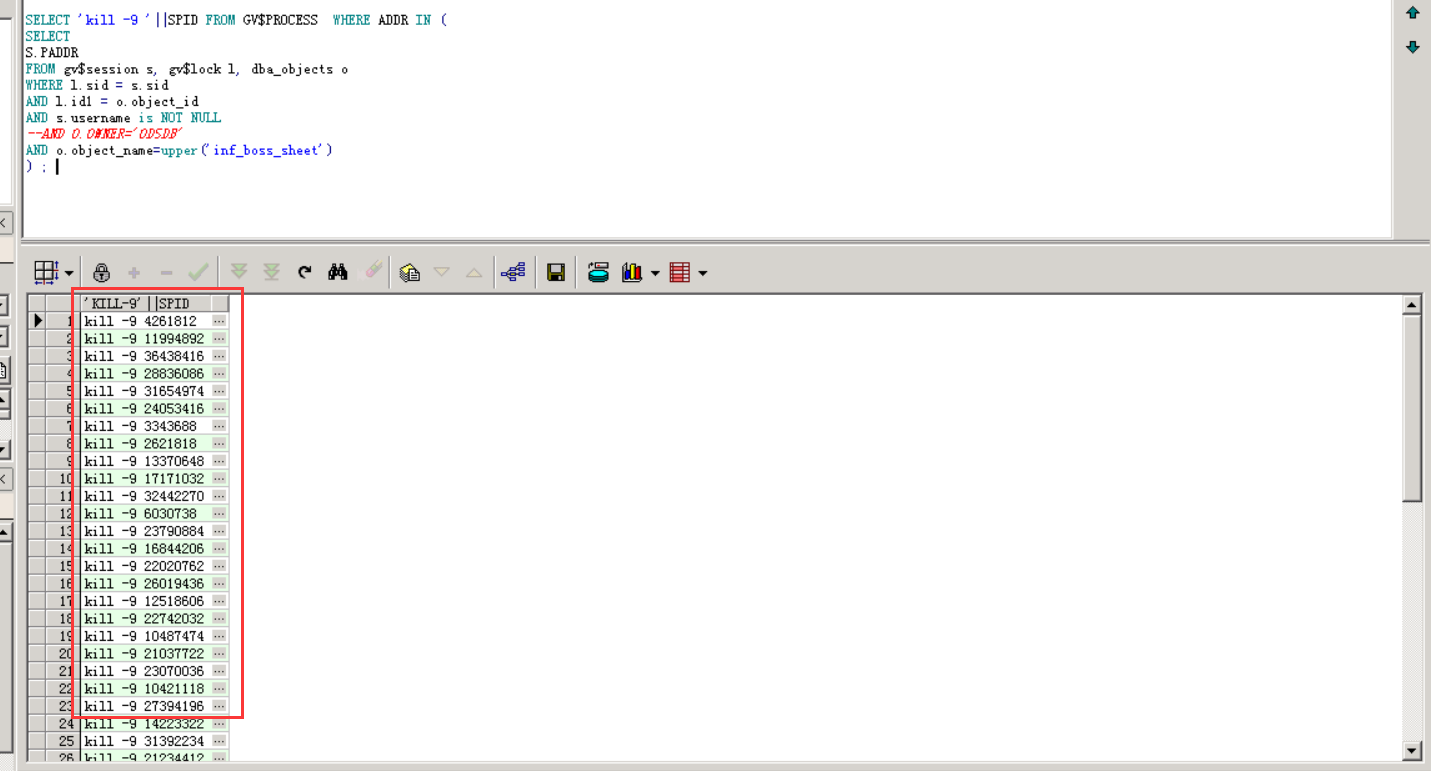
and t1.object\_id = t3.object\_id

and t4.ADDR=t2.PADDR

and t5.SQL\_ID=t2.SQL\_ID

and t3.OBJECT\_NAME=upper('\*\*\*\*表名\*\*\*')

1. 如果需要杀死锁表的sql进程，用以下sql得到kill语句，到数据库主机执行（四台数据库主机）。



SELECT 'kill -9 '||SPID FROM GV$PROCESS WHERE ADDR IN (

SELECT

S.PADDR

FROM gv$session s, gv$lock l, dba\_objects o

WHERE l.sid = s.sid

AND l.id1 = o.object\_id

AND s.username is NOT NULL

--AND O.OWNER='ODSDB'

AND o.object\_name=upper('\*\*\*\*表名\*\*\*')

);

### 查询消耗大的sql

--查看总消耗时间最多的前10条SQL语句

select \*

from (select v.sql\_id,

v.child\_number,

v.sql\_text,

v.elapsed\_time,

v.cpu\_time,

v.disk\_reads,

v.executions,

rank() over(order by v.elapsed\_time desc) elapsed\_rank

from gv$sql v) a

where elapsed\_rank <= 10;

--查看CPU消耗时间最多的前10条SQL语句

select \*

from (select v.sql\_id,

v.child\_number,

v.sql\_text,

v.elapsed\_time,

v.cpu\_time,

v.disk\_reads,

v.executions,

rank() over(order by v.cpu\_time desc) elapsed\_rank

from gv$sql v) a

where elapsed\_rank <= 10;

--查看消耗磁盘读取最多的前10条SQL语句

select \*

from (select v.sql\_id,

v.child\_number,

v.sql\_text,

v.elapsed\_time,

v.cpu\_time,

v.disk\_reads,

v.executions,

rank() over(order by v.disk\_reads desc) elapsed\_rank

from gv$sql v) a

where elapsed\_rank <= 10;

### 根据进程号查sql

可以到数据库主机使用topas命令查看cpu占比较大的oracle进程，取出进程号，用以下sql查询该oracle进程正在执行的sql。

select vs.SQL\_TEXT ,

vp.spid,

vse.SQL\_ID,

vse.STATUS as 会话状态, --ACTIVE-正在执行 INACTIVE-等待执行 killed-已被杀死

vse.CLIENT\_INFO as 主机,

vse.USERNAME 用户,

vse.SERVICE\_NAME 服务名

from V$PROCESS vp,v$session vse,v$sqlarea vs

where vp.ADDR=vse.PADDR

and vse.SQL\_ID=vs.SQL\_ID

--and vp.SPID='\*\*进程号\*\*'

### 其他脚本

-- 根据脚本查询sql\_id

SELECT a.SQL\_ID, a.CHILD\_NUMBER, a.SQL\_FULLTEXT, a.SQL\_TEXT，

FROM gv$sql a

WHERE a.SQL\_TEXT LIKE '%AND wo.work\_order\_type != %'

-- 根据sql\_id查询执行计划

SELECT \* FROM gv$sql\_plan WHERE SQL\_ID = 'bqrnap8n9b8zy';

-- 根据sql\_id, child number查询执行计划

SELECT \* FROM table(dbms\_xplan.display\_cursor('bqrnap8n9b8zy', 3));

--根据

select \* from gv$session vs where vs.SQL\_ID='bkr81h4cxyqkk'

select vp.SPID,vp.\* from gv$process vp where vp.ADDR='0700012FE23FF358'

根据sql\_id查进程号

--------查询当前运行的sql语句

select sql\_text, COUNT(1)

from gv$session, gv$process, gv$sqlarea

where paddr = addr

and sql\_hash\_value = hash\_value

and status = 'ACTIVE'

and gv$session.username is not null

GROUP BY sql\_text

ORDER BY COUNT(1) DESC;

平时也可以看一下这个语句，然后如果故障时没有哪个进程CPU特别高，也可以看看有没有哪个语句执行数量暴增

### awr报告

使用具有DBA权限的数据库用户登录到数据库中；

2、使用以下SQL查询一下，具体时间参数视需要监控的时间而定，如果每天中

午12点监控，建议时间从前一天的中午12点到当天的12点。

select\*

From dba\_hist\_snapshot

Where end\_interval\_time>=

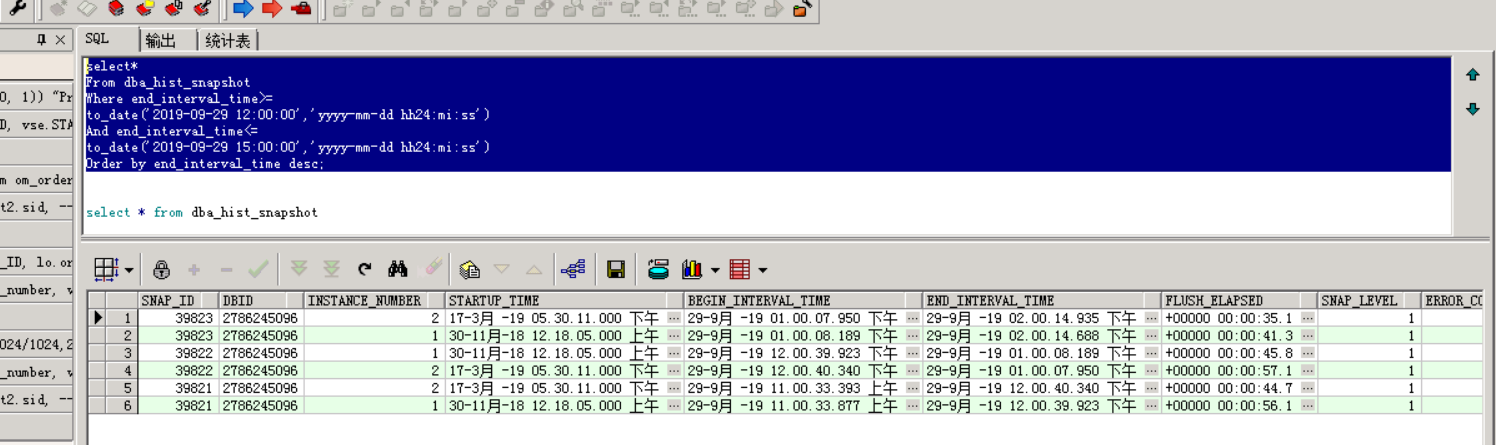
to\_date('2019-09-29 12:00:00','yyyy-mm-dd hh24:mi:ss')

And end\_interval\_time<=

to\_date('2019-09-29 15:00:00','yyyy-mm-dd hh24:mi:ss')

Order by end\_interval\_time desc;

查询结果如下，从查询结果里面获取以下信息，应用到下一步的SQL



3、使用以下SQL查询一下

Select \* from

table(dbms\_workload\_repository.awr\_report\_html(DBID,1,开始的

SNAP\_ID,终止的SNAP\_ID,8));

组合后如下：

select\*from

table(dbms\_workload\_repository.awr\_report\_html(4263475968,1,12753,127

62,8));

4、把查出来的信息整体复制保存到某个文件中，文件必须为HTML格式的文件。