



模拟1 log buffer space等待事件!需要有一个大事务！

create table test\_lbs\_wait (id int,name varchar2(22));

insert into test\_lbs\_wait values (1,'aaa');

commit;

做一个大事务，先找出rowid，目的最快的找到这条数据！！

SQL> select rowid from test\_lbs\_wait where id=1;

ROWID

------------------

AAATvlAAGAAAAC8AAA

begin

    for i in 1..2000000

    loop

    update test\_lbs\_wait set id=id+0,name='BBBBB' where rowid='AAATvlAAGAAAAC8AAA';

        if mod(i,50)=0 then

        commit;

        end if;

    end loop;

end;

/

每200次提交一次，批量提交！

select sid from v$mystat where rownum=1;

or

select userenv('sid') from dual;

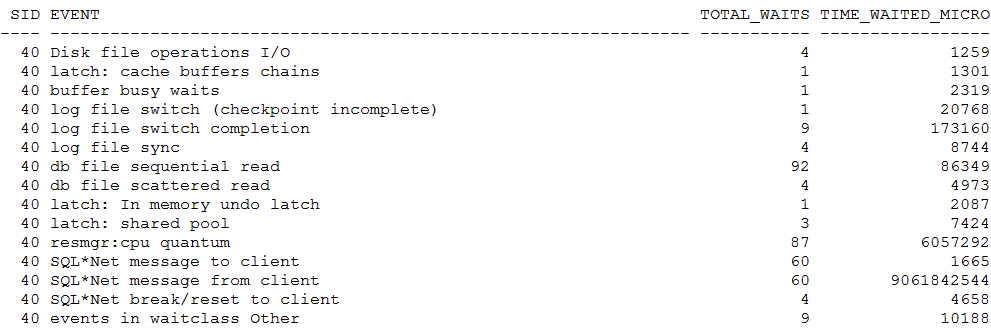
SID:40

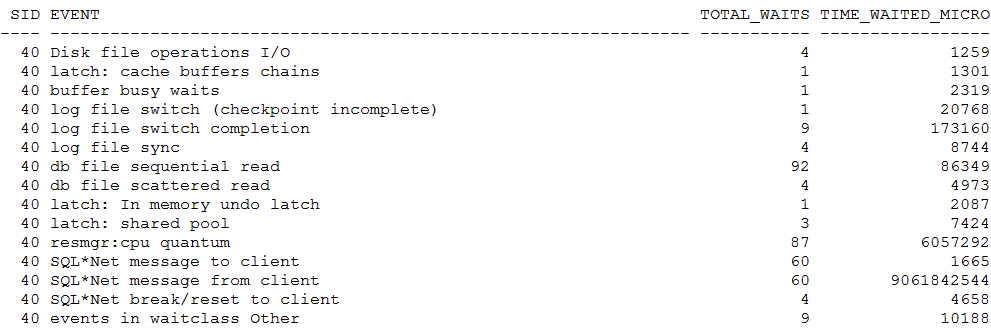
用40执行以上的大事务

然后查询等待事件

v$session\_event有对会话等待事件的统计！！！

select sid,event,total\_waits,time\_waited\_micro from v$session\_event where sid in (40,42);





为什么没有log buffer space等待事件！！！？？？？？？？其他经典等待事件都出现了！！！

毫秒，符号ms（英语：millisecond ）.

1毫秒等于一千分之一秒（10-3秒）

微秒，符号μs（英语：microsecond ）.

1微秒等于一百万分之一秒（10-6秒）

纳秒，符号ns（英语：nanosecond ）.

1纳秒等于十亿分之一秒（10-9秒）

皮秒，符号ps（英语：picosecond ）.

1皮秒等于一万亿分之一秒（10-12秒）

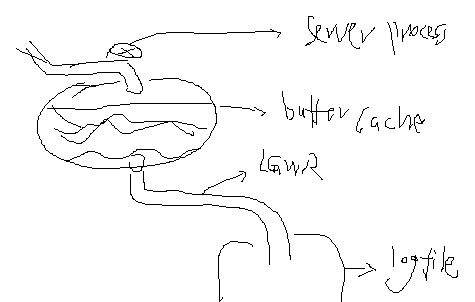
1小时=60分钟=3600秒=3600000毫秒=3600000000微秒

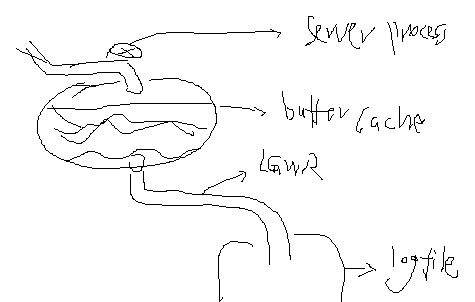
以上事件单位，oracle等待事件的级别是什么？（敏感度）：毫秒级别ms！！！！！！！

对sql语句慢等等待事件的时间单位

等待事件是性能优化的入口！！

举例





service process==》 log buffer ===》lgwr==》redo log file

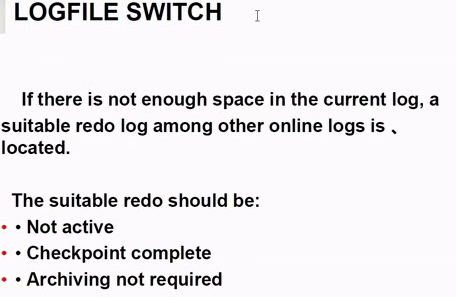
如果事务特别多==》log buffer 一下满了，lgwr来不及写完，就造成了log buffer space等待事件

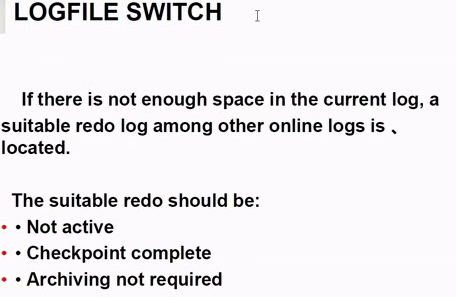
解决方法！

1 可以加大log buffer大小，减少问题的出现

2 优化i/o

3 增加加cpu





以上的脚本，也可以模式此等待事件

可以看到 checkpoint incomplete

1 log file switch (checkpoint incomplete)，添加日志组,替换日志组（每次切换日志组）这个错误会导致数据库hang住！

日志覆盖之前其对应的data dirty buffer必须写入磁盘。否则在alert报无法分配新的日志！

alter database drop logfile group 1;

alter database add logfile group 1 'D:\APP\ORACLE\ORADATA\TEST\REDO01.LOG'  size 100m reuse;

注意：inavtive

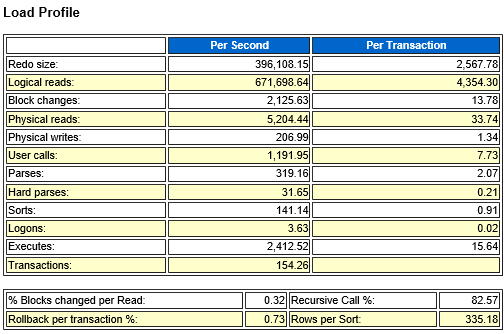
alter system checkpoint;

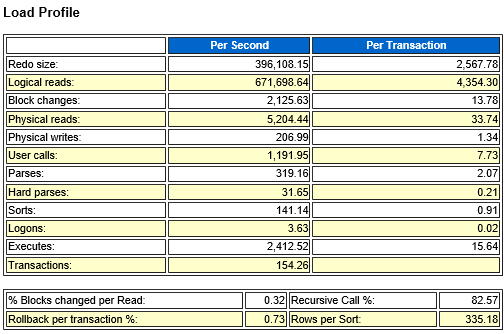
alter system switch logfile;

日志文件可以做多路复用，注意磁盘i/o！！

那么日志组多大合适呢！！建议切换的的间隔15-30min之间

awr》profile》redo size





以上可以看到redo size=396108.15byte/s=396k/s

那么如果以20min间隔为例则需要500m大小的日志

396k/s=x/20\*60

x=20\*60\*396k=475200k=500m

2 log file switch(archiving needed)，添加日志组（每次所有日志组循环）

写前协议对应的日志没有被归档。

3 log file sync （0-7ms） 在5ms就可以

commit==》log file sync，commit就会发生logfile sync

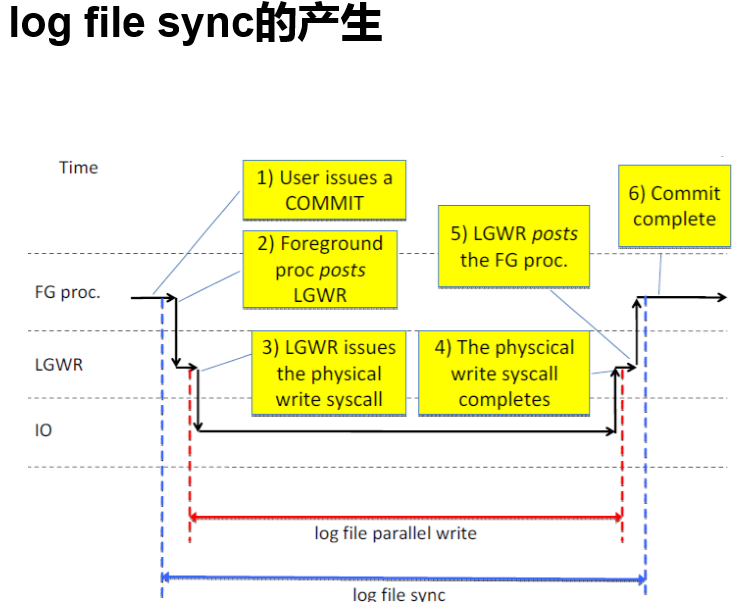
用户发出的commit：commit命令

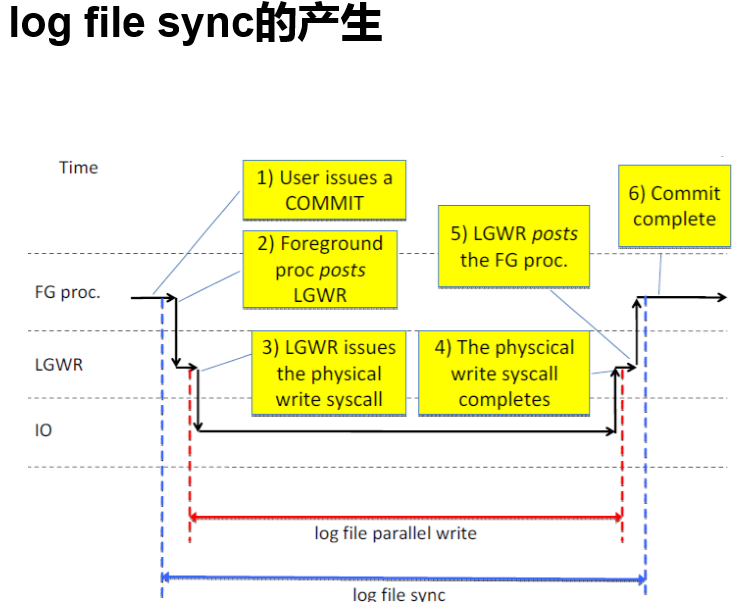
程序发出的commit：commit步骤

ddl发生的commit： 前commit，后commit

sequence引起的commit： rowcache 20  ==》修改数据字典==》commit

rollback： commit





1-2 commit ==》消耗cpu ，post信号量通知（ipc）lgwr==》lgwr   ==》开始log file sync （主要是cpu时间）  一般时间比较短

2-3 lgwr获取cpu，因为需要定位log buffer位置，进程需要获取资源

3-4 lgwr调用操作系统函数去写日志，latch锁保护，此时lgwr sleep  ==》 log file parallel write（主要是i/o时间）时间比较长！

4-5 唤醒lgwr，gwr通知fg proc写完，释放latch，可以被覆盖了！

6   commit complete==》通知用户

通过v$session\_event  ,total\_waits,time\_waited\_micro,两者时间对比一下

log file sync 时间 对比log file parallel write 时间 后者占用时间大===》i/o问题  红线

log file sync 时间 对比log file parallel write 时间 后者占用时间小===》cpu问题  红线和蓝线之间

解决方法经验

1 提升i/o  raid 5==>raid 1+0 写的块

2 提升cpu  lgwr优先级，隐含参数

3 减少commit（了解事务的logical work unit）

4 减少日志产生（nolloging）

5 增加log buffer

6 commit\_write='immediate nowait'  ===>异步提交

7 存储电池等硬件

《深入理解计算机》！！！！！！！！！！！！！！！！！！！！！！！！

desc v$event\_histogram;   ===>等待事件时间比较长的放在这个地方

From <<http://www.zybang.com/question/a7be3140d09dded7da0e06d718ecb8ba.html>>