

DB2和Oracle的并发控制（锁）比较五 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/269/2021_2022_DB2_E5_92_8COrac_c67_269292.htm

5 Oracle 多粒度封锁机制的监控为了监控Oracle系统中锁的状况，我们需要对几个系统视图有所了解：5.1 v\$sqllock视图v\$sqllock视图列出当前系统持有的或正在申请的所有锁的情况，其主要字段说明如下：表七：v\$sqllock视图主要字段说明其中在TYPE字段的取值中，本文只关心TM、TX两种DML锁类型；5.2 v\$sqllocked_object视图v\$sqllocked_object视图列出当前系统中哪些对象正被锁定，其主要字段说明如下：表八：v\$sqllocked_object视图字段说明5.3 Oracle锁监控脚本根据上述系统视图，可以编制脚本来监控数据库中锁的状况

5.3.1 showlock.sql 第一个脚本showlock.sql，该脚本通过连接v\$sqllocked_object与all_objects两视图，显示哪些对象被哪些会话锁住：/* showlock.sql */column o_name format a10column lock_type format a20column object_name format a150select

```
rpad(oracle_username,10) o_name,session_id  
sid,decode(locked_mode,0,None,1,Null,2,Row share,3,Row  
Exclusive,4,Share,5,Share Row Exclusive,6,Exclusive)
```

```
lock_type,object_name ,xidusn,xidslot,xidsqnf  
from  
v$sqllocked_object,all_objectswhere
```

```
v$sqllocked_object.object_id=all_objects.object_id.5.3.2
```

showalllock.sql第二个脚本showalllock.sql，该脚本主要显示当前所有TM、TX锁的信息；/* showalllock.sql */0select

```
sid,type,id1,id2,decode(lmode,0,None,1,Null,2,Row share,3,Row  
Exclusive,4,Share,5,Share Row
```

Exclusive,6,Exclusive)lock_type,request,ctime,blockfrom
v\$lockwhere TYPE IN(TX,TM).6 DB2 多粒度封锁机制示例 以下
示例均运行在DB2 UDB中，适用所有数据库版本。首先打开
三个命令行窗口(DB2 CLP)，其中两个（以下用SESS#1
、SESS#2表示）以db2admin用户连入数据库，以操作SAMPLE
库中提供的示例表（employee）；另一个（以下用SESS#3表
示）以db2admin用户连入数据库，对执行的每一种类型
的SQL语句监控加锁的情况；希望读者通过这种方式对每一
种类型的SQL语句监控加锁的情况。（因为示例篇幅很大，笔
者在此就不做了，建议读者用类似方法验证加锁情
况）/home/db2inst1>db2 c 0update employee set
comm=9999(SESS#1)/home/db2inst1>db2 c 0select * from
employee(SESS#2处于lock wait)/home/db2inst1>db2 c get
snapshot for locks on sample(SESS#3监控加锁情况)注：db2 c为
不自动提交（commit）SQL语句，也可以通过 db2 0update
command options using c off关闭自动提交（autocommit，缺省
是自动提交）7 总结总的来说，DB2的锁和Oracle的锁主要有
以下大的区别：1．Oracle通过具有意向锁的多粒度封锁机制
进行并发控制，保证数据的一致性。其DML锁（数据锁）分
为两个层次（粒度）：即表级和行级。通常的DML操作在表
级获得的只是意向锁（RS或RX），其真正的封锁粒度还是在
行级；DB2也是通过具有意向锁的多粒度封锁机制进行并发
控制，保证数据的一致性。其DML锁（数据锁）分为两个层
次（粒度）：即表级和行级。通常的DML操作在表级获得的
只是意向锁（IS，SIX或IX），其真正的封锁粒度也是在行级
；另外，在Oracle数据库中，单纯地读数据（SELECT）并不

加锁，这些都提高了系统的并发程度，Oracle强调的是能够"读"到数据，并且能够快速地进行数据读取。而DB2的锁强调的是"读一致性"，进行读数据(SELECT)时会根据不同的隔离级别（RR,RS,CS）而分别加S,IS,IS锁，只有在使用UR隔离级别时才不加锁。从而保证不同应用程序和用户读取的数据是一致的。

2. 在支持高并发度的同时，DB2和Oracle对锁的操纵机制有所不同：Oracle利用意向锁及数据行上加锁标志位等设计技巧，减小了Oracle维护行级锁的开销，使其在数据库并发控制方面有着一定的优势。而DB2中对每个锁会在锁的内存（locklist）中申请分配一定字节的内存空间，具体是X锁64字节内存，S锁32字节内存（注：DB2 V8之前是X锁72字节内存而S锁36字节内存）。

3. Oracle数据库中不存在锁升级，而DB2数据库中当数据库表中行级锁的使用超过 $locklist * maxlocks$ 会发生锁升级。

4. 在Oracle中当一个session对表进行insert，update，delete时候，另外一个session仍然可以从Oracle回滚段或者还原表空间中读取该表的前映象（before image）。而在DB2中当一个session对表进行insert，update，delete时候，另外一个session仍然在读取该表数据时候会处于lock wait状态，除非使用UR隔离级别可以读取第一个session的未提交的值；所以Oracle同一时刻不同的session有读不一致的现象，而DB2在同一时刻所有的session都是"读一致"的。

100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com