

DB2和Oracle的并发控制（锁）比较三 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_269295.htm 3.2.2 意向锁的类型 由两种基本的锁类型（S锁、X锁），可以自然地派生出两种意向锁：意向共享锁（Intent Share Lock，简称IS锁）：如果要对一个数据库对象加S锁，首先要对其上级结点加IS锁，表示它的后裔结点拟（意向）加S锁；意向排它锁（Intent Exclusive Lock，简称IX锁）：如果要对一个数据库对象加X锁，首先要对其上级结点加IX锁，表示它的后裔结点拟（意向）加X锁。另外，基本的锁类型（S、X）与意向锁类型（IS、IX）之间还可以组合出新的锁类型，理论上可以组合出4种，即：S IS，S IX，X IS，X IX，但稍加分析不难看出，实际上只有S IX有新的意义，其它三种组合都没有使锁的强度得到提高（即：S IS=S，X IS=X，X IX=X，这里的"="指锁的强度相同）。所谓锁的强度是指对其它锁的排斥程度。这样我们又可以引入一种新的锁的类型：共享意向排它锁（Shared Intent Exclusive Lock,简称SIX锁）：如果对一个数据库对象加SIX锁，表示对它加S锁，再加IX锁，即SIX=S IX。例如：事务对某个表加SIX锁，则表示该事务要读整个表（所以要对该表加S锁），同时会更新个别行（所以要对该表加IX锁）。这样数据库对象上所加的锁类型就可能有5种：即S、X、IS、IX、SIX。具有意向锁的多粒度封锁方法中任意事务T要对一个数据库对象加锁，必须先对它的上层结点加意向锁。申请封锁时应按自上而下的次序进行；释放封锁时则应按自下而上的次序进行；具有意向锁的多粒度封锁方法提高了系统的并发度，减少了加锁和解

锁的开销。3.3 Oracle的TM锁（表级锁） Oracle的DML锁（数据锁）正是采用了上面提到的多粒度封锁方法，其行级锁虽然只有一种（即X锁），但其TM锁（表级锁）类型共有5种，分别称为共享锁（S锁）、排它锁（X锁）、行级共享锁（RS锁）、行级排它锁（RX锁）、共享行级排它锁（SRX锁），与上面提到的S、X、IS、IX、SIX相对应。需要注意的是，由于Oracle在行级只提供X锁，所以与RS锁（通过SELECT ... FOR UPDATE语句获得）对应的行级锁也是X锁（但是该行数据实际上还没有被修改），这与理论上的IS锁是有区别的。锁的兼容性是指当一个应用程序在表（行）上加上某种锁后，其他应用程序是否能够在表（行）上加上相应的锁，如果能够加上，说明这两种锁是兼容的，否则说明这两种锁不兼容，不能对同一数据对象并发存取。下表为Oracle数据库TM锁的兼容矩阵（Y=Yes，表示兼容的请求；N=No，表示不兼容的请求；-表示没有加锁请求）：表五：Oracle数据库TM锁的相容矩阵

一方面，当Oracle执行SELECT...FOR UPDATE、INSERT、UPDATE、DELETE等DML语句时，系统自动在所要操作的表上申请表级RS锁（SELECT...FOR UPDATE）或RX锁（INSERT、UPDATE、DELETE），当表级锁获得后，系统再自动申请TX锁，并将实际锁定的数据行的锁标志位置位（指向该TX锁）；另一方面，程序或操作人员也可以通过LOCK TABLE语句来指定获得某种类型的TM锁。下表是笔者总结了Oracle中各SQL语句产生TM锁的情况：表六：Oracle数据库TM锁小结

我们可以看到，通常的DML操作（SELECT...FOR UPDATE、INSERT、UPDATE、DELETE），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；另外，Oracle数据库的一个显著特点是，在缺省情况下，单纯地读数据（SELECT）并不加锁，Oracle通过回滚段（Rollback segment）来保证用户不读“脏”数据。这些都提高了系统的并发程度。由于意向锁及数据行上锁标志位的引入，减小了Oracle维护行级锁的开销，这些技术的应用使Oracle能够高效地处理高度并发的事务请求。

4 DB2多粒度封锁机制的监控

在DB2中对锁进行监控主要有两种方式，第一种方式是快照监控，第二种是事件监控方式。

4.1 快照监控方式

当使用快照方式进行锁的监控前，必须把监控锁的开关打开，可以从实例级别和会话级别打开，具体命令如下：

```
db2 update dbm cfg using dft_mon_lock on(实例级别)
db2 update monitor switches using lock on(会话级别, 推荐使用)
```

当开关打开后，可以执行下列命令来进行锁的监控

```
db2 get snapshot for locks on ebankdb
```

（可以得到当前数据库中具体锁的详细信息）

```
db2 get snapshot for locks on ebankdb
Fri Aug 15 15:26:00 JiNan 2004
Database Lock Snapshot
Database name = DEV
Database path = /db2/DEV/db2dev/NODE0000/SQL00001/
Input database alias = DEV
Locks held = 49
Applications currently connected = 38
Agents currently waiting on locks = 6
Snapshot timestamp = 08-15-2003 15:26:00.951134
Application handle = 6
Application ID = *LOCAL.db2dev.030815021007
Sequence number = 0001
Application name = disp work
Authorization ID = SAPR3
Application status = UOW
WaitingStatus change time =
Application code page = 819
Locks held = 0
Total wait time (ms) = 0
Application handle = 97
Application ID =
```

*LOCAL.db2dev.030815060819Sequence number =
0001Application name = tpAuthorization ID = SAPR3Application
status = Lock-waitStatus change time = 08-15-2003
15:08:20.302352Application code page = 819Locks held = 6Total
wait time (ms) = 1060648 Subsection waiting for lock = 0ID of agent
holding lock = 100 Application ID holding lock =
*LOCAL.db2dev.030815061638 Node lock wait occurred on = 0
Lock object type = Row Lock mode = Exclusive Lock (X) Lock
mode requested = Exclusive Lock (X) Name of tablespace holding
lock = PSAPBTABD Schema of table holding lock = SAPR3 Name
of table holding lock = TPLOGNAMES Lock wait start timestamp =
08-15-2003 15:08:20.302356 Lock is a result of escalation = NOList
Of Locks Lock Object Name = 29204 Node number lock is held at =
0 Object Type = Table Tablespace Name = PSAPBTABD Table
Schema = SAPR3 Table Name = TPLOGNAMES Mode = IX Status
= Granted Lock Escalation = NO 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