

数据库理论作业week17

曹广杰

15352015 数据科学与计算机

授课教师：刘玉葆

Content

数据库理论作业week17

Content

15.1

15.2

15.21

15.1

证明两阶段封锁协议保证冲突可串行化，并且事务可以根据其封锁点串行化。

由于可串行化表示在有锁的协议的前提下，事务是可以有特定的排序方式的，事务会根据其增长阶段的结束点——封锁点，进行排序。存在这样一个顺序使得这一系列事务可以执行完成。

首先假设两阶段封锁协议不能保证可串行化，即这一系列事务的执行顺序会形成一个环。

假设一系列事务： $T_0, T_1, T_2, \dots, T_{n-1}$ ，这n个事务都遵循两阶段封锁协议，但是非可串行化。那么它们是可以形成一个环的。此时笔者使用 α 作为该事务的封锁点，则若有以下顺序：

$$\alpha_1 < \alpha_2 < \alpha_3 < \dots < \alpha_{n-1}$$

此时如果执行顺序成环状，则在某一个位置某事务会重复出现，假设该事务是x。那么无论x为多少，都会出现 $\alpha_x < \alpha_x$ 的情况，这显然是不合理的。

- 综上，得出结论，两阶段封锁协议是可以根据其封锁点可串行化的。

15.2

考虑以下两个事务：

```
T34 : read(A);  
      read(B);  
      if(A = 0) then B := B + 1;  
      write(B);
```

```
T35 : read(B);  
      read(A);  
      if(B = 0) then A := A + 1;  
      write(A);
```

给事务 T_{34} 和 T_{35} 增加加锁、解锁指令，使其遵守两阶段封锁协议，则会引起死锁吗？

如果按照在只有读取操作的时候使用S锁的原则，则会发生死锁：

T_{31}	T_{32}
lock-S(A)	
	lock-S(B)
	read(B)
read(A)	
lock-X(B)	
	lock-X(A)

此时已经发生了死锁。

15.21

大部分数据库系统实现采用严格的两阶段封锁协议，说明该协议流行的三点理由。

严格的两阶段封锁协议，要求：

- 封锁是两个阶段；
 - 事务的排他锁必须在事务提交之后才可以释放；
-
1. 实现较为简单；
 2. 避免了级联回滚导致已完成工作的丢失；
 3. 由于使用的数据在使用之前都加了排它锁，允许了大量事务的并发；