

数据库系统和信息管理期末复习

并发控制专题

2025 年 6 月考试该题分值为 15 分

大概可能问到以下几个问题：

1. 调度之后的结果是什么？数据库崩溃恢复之后的结果是什么？数据日志文件是什么？
2. 该调度是否是冲突可串行化的？是否是可串行化的？如果是，请给出一个等价的串行调度（等价于哪个序列调度？）；如果不是，请简要说明理由。
3. 画出优先图（Precedence Graph）
4. 该调度是否是可恢复调度？是否是无级联调度？原因是？
5. 请在 T1 和 T2 中添加 X/S-Lock 和 Unlock，以符合两阶段锁定协议（2PL）
6. 添加了两阶段锁定协议之后，T1 和 T2 是否可能产生死锁？请给出判断理由

串行调度：最理想的状态，先执行完一个事务 A，之后再执行事务 B，之后再执行事务 C……所有事务完全不交叉，一个一个执行

数据库串行化如何理解：

串行化就是事务一个接着一个执行，先执行完事务 A 整体的操作，之后再执行事务 B 整体的操作，而不是执行 A 的部分操作，执行 B 的部分操作，再执行 A 的部分操作

两条相邻的但是“不相关”的语句的执行顺序是可以互换的，比如说 T1.read(A) 和 T2.read(A)，两者不冲突，可以互换；T1.read(A) 和 T2.write(B)，虽然一个是读一个是写，但是两者用的资源不同，也是可以互换的；或者是 T1.write(A) 和 T2.write(B)，对不同的资源进行操作，也是可以互换的。如果经过有限轮数的互换能够达到一种“串行化”的调度，那么这个调度称为是冲突可串行化

可串行化：一个调度是可串行化的，如果它的执行结果与某个串行调度的结果等价。虽然事务操作是交叉执行的，但最终对数据库的影响和某个按顺序执行的调度相同。

Schedule S1: T1.R(A), T2.R(C), T1.R(B), T2.R(B), T1.W(B:=6), T2.W(A:=8) 这个就是可串行化的。**如果一个调度的结果与某一串行调度执行的结果等价，则称该调度是可串行化调度，否则是不可串行化调度。**

1. 调度之后的结果是什么？数据库崩溃恢复之后的结果是什么？数据日志文件是什么？

(1) 建议对每一步标注当前的数值

一个事务 1 读取了 A 数据是 10，就相当于将这个数值存储到了这个事务 1 自己的“缓存”中（就是这么理解），既然读了之后，别的事务 2 再对 A 进行+5 操作，A 是不知道的，如果后续事务 1 对 A 进行+1 操作，那么 A 会对自己已经存储的数据进行一个修改，得到 $10+1=11$

另一个例子，原来 A 的数据是 50，事务 1 写了 A 的数据为 100，但是没有 commit，这是事务 2 在 A 写完之后读取 A，事务 2 读到的数值是 100（尽管 A 没有 commit），事务 write 了一个数值之后，别的事务再读取，读到的是新值

比如说下面的第 5 题

(2) 搞明白了，在这里还是区分一下 commit 和 write 的区别，**write 是将数据写入内存，**

其他的事务如果在此时 read，会读取到，是暂时保存下来，如果出现意外情况，数据会丢失。而 commit 是将数据写入磁盘，就是永久保存下来。比如说数据库崩溃，这时候没来得及 commit 到硬盘的数据就会丢失掉，数据还原为原来的初始数值，而 commit 了的数据保存在硬盘中，不会出现问题。

数据库崩溃恢复后结果：依据日志恢复规则，恢复中：

已提交事务会被重做（redo）；未提交事务会被撤销（undo）

(3) 数据日志文件：

记录的内容包括：

开始事务（<start-T1>）

写操作前后的值（<T1, X, old, new>）

提交记录（<commit-T1>）

2. 是否冲突可串行化？是否可串行化？如是，给出等价串行调度。

冲突可串行化：构造优先图，存在环就不是冲突可串行化，不存在环是冲突可串行化可串行化：

（1）执行一遍，得到数值

（2）按照 $T_1 \rightarrow T_2$ 或 $T_2 \rightarrow T_1$ 的顺序各自执行一遍，得到的结果如果和原来的结果一样的话，那么就是可串行化的，如果不是，那么就是不可串行化的

是冲突可串行化的话一定时可串行化的，如果不是冲突可串行化的话，也可能是可串行化的（不一定不是可串行化）

等价串行调度：根据优先图的拓扑排序给出事务执行顺序。

3. 画出优先图（Precedence Graph）

每个事务是一个节点。存在如下冲突操作，则画边：

T_1 的 $W(X)$ 在 T_2 的 $R(X)$ 之前，则： $T_1 \rightarrow T_2$

T_1 的 $R(X)$ 在 T_2 的 $W(X)$ 之前，则： $T_1 \rightarrow T_2$

T_1 的 $W(X)$ 在 T_2 的 $W(X)$ 之前，则： $T_1 \rightarrow T_2$

4. 该调度是否是可恢复调度？是否是无级联调度？原因是什么？

可恢复调度（Recoverable Schedule）：若一个事务 T_2 读取了 T_1 写的数据（也就是说 T_2 依赖 T_1 ），那么 T_2 必须在 T_1 之后才能提交，否则不是可恢复调度

无级联调度（Cascadeless Schedule）：若事务 T_2 读取的数据由 T_1 写（也就是说 T_2 依赖 T_1 ），则必须等到 T_1 提交之后， T_2 才能读取。

5. 添加两阶段锁（2PL）

两阶段锁协议要求：

所有加锁（Lock）必须在释放锁（Unlock）之前完成（增长阶段）

一旦释放锁，不能再加新锁（收缩阶段）

步骤：

对每个 $R(X)$ ，加 $S-Lock(X)$ （共享锁）

对每个 $W(X)$ ，加 $X-Lock(X)$ （排他锁）

两个进程可以同时加 $S-Lock(A)$ ，共享锁 $S-Lock$ 是不互斥的，可以同时申请；

但是 $X-Lock(A)$ 排他锁不行，只能有一个事务使用。

典型例子是下面的第 2 题（24 年考试题）

插入相应的 Unlock(X)操作，注意在恰当位置释放所有锁。

如果有提交，需要在解锁之前执行 commit（不要丢了 commit）下面的第 1、2 题

6. 是否可能死锁？

判断死锁：T1 等待 T2 释放锁，T2 也等待 T1 释放锁 \Rightarrow 形成等待环

分析 T1 和 T2 请求的锁是否有交叉依赖（如 T1 先锁 X，T2 锁 Y，随后 T1 要 Y 但被 T2 占用，T2 要 X 但被 T1 占用）

注意一下，X-Lock 可能发生死锁，如果某个资源仅仅被加了共享锁 S-Lock，他是不会发生死锁的。比如说下面的第 2 题，对 B 加的是共享锁，事务 T1 和 T2 可以同时对 B 加上共享锁，但是 X-Lock 排他锁就不行了

答题示例：

该调度不会产生死锁，因为 T1 加锁的是 A 和 B，而 T2 加锁的是 B 和 C，二者不存在循环等待条件。

不会产生死锁，因为 T1 只对数据项 A 申请排他锁，没有对数据 B 申请排他锁，T2 只对数据项 A，数据项 B 申请共享锁（S-Lock），不会造成互相等待的情况。

1. From 练习题

Consider the concurrent schedule of transactions T1 and T2.

T1	T2
read (A)	
$A=A+1$	
	read (B)
write(A)	
	$B=B-1$
read(B)	
	write(B)
$B=B+1$	
	read(C)
write(B)	
	$C=C+2$
	write(C)
	Commit
Commit	

1. Is the schedule conflict serializable? If so, give an equivalent serial schedule. If not, give an explain briefly. (5 points)
2. Add lock and unlock instructions to Transactions T1 and T2, only consider the 2PL concurrent schedule of transactions T1 and T2, observe the two-phase-locking protocol ,Can it results in a deadlock? (5 points)

1. 该调度是否是冲突可串行化的？如果是，请给出一个等价的串行调度；如果不是，请简要说明理由。

什么是冲突：属于不同事务；操作访问相同的数据项；至少一个操作是写操作
 什么是冲突可串行化（Conflict Serializable）：调度是多个事务交错执行的一种操作顺序，并发调度如果能够等价于某个串行调度（即事务一个接一个执行，先执行第一个T1事务，再执行第二个T2事务），就叫可串行化

这个题目中：

T1.Read(B)先于 T2.Write(B)，两语句冲突不可交换， $T1 \rightarrow T2$ ；

T2.Write(B)先于 T1.Write(B)，两语句冲突不可交换， $T2 \rightarrow T1$ 。

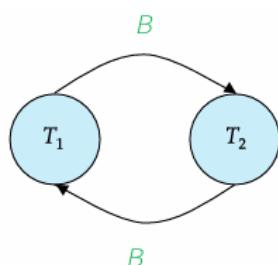
优先图中出现环，该调度不是冲突可串行化的调度

答案：不是冲突可串行化调度。

T1 的 read(B) 和 T2 的 write(B) 是冲突操作，

T2 的 write(B) 和 T1 的 write(B) 也是冲突操作，

在优先图（precedence graph）中存在环，所以该调度不是冲突可串行化的。



2. 在两阶段锁协议（2PL）下，为事务 T1 和 T2 添加加锁和解锁指令，以保证产生可串行化且无极联的调度。你设计的 2PL 调度是否可能导致死锁？请说明理由。

2PL (Two-Phase Locking Protocol) 是一种确保调度可串行化的方法：

生长阶段：事务可以申请锁 (lock)，但不能释放

收缩阶段：一旦事务释放了第一个锁，就不能再申请新的锁

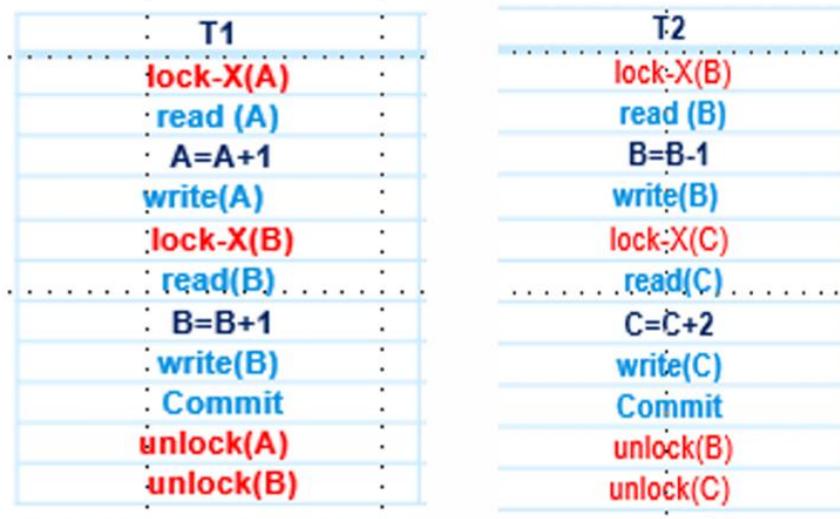
遵循 2PL 的调度一定是可串行化的（但不一定避免死锁）

原题目上的调度不是冲突可串行化调度，加锁满足 2PL 后的并发事务是可以保证产生冲突可串行化调度的

T1 :Lock-X(A), read(A), write(A), Lock-X(B), read(B), write(B), commit, unlock(A), unlock(B)

T2: Lock-X(B), read(B), write(B), Lock-X(C), read(C), write(C), commit, unlock(B), unlock(C)

这里给出的是强二阶段锁 (Rigorous 2PL)，可以保证 T1、T2 的调度总是可串行化且可恢复无极联。并发事务不会发生死锁，因只有 B 是两个事务都要访问的数据项，所以不会构成两个事务相互等待对方对不同数据项的解锁。



T1: lock-X(A), read(A), write(A), lock-X(B), read(B), write(B), unlock(A), unlock(B)

T2: lock-X(B), read(B), write(B), lock-X(C), read(C), write(C), unlock(B), unlock(C)

该调度不会产生死锁，因为 T1 加锁的是 A 和 B，而 T2 加锁的是 B 和 C，二者不存在循环等待条件。

2. From 2023-2024 学年第二学期考试题

Consider the following concurrent schedule **S1** of transactions T1 and T2:

T1	T2
read(A)	
write(A)	
	read(B)
	read(A)
read(B)	
	commit
commit	

Answer the following questions based on the schedule S1:

1. Is S1 a conflict serializable schedule? Please give the reason for your judgment. If so, S1 conflict equivalent to which serial schedule? (3 points)
2. Determine whether schedule S1 is a recoverable schedule, is a cascadeless schedule, and give the reasons. (2 points)
3. Please add lock and unlock instructions to T1 and T2 to meet the two-phase locking protocol (2PL). (3 points)
4. Whether T1 and T2 may produce deadlock after adding the two-phase locking protocol(2PL)? Please give the reason for your judgement. (2 points)

调度序列：

T1.read(A), T1.write(A), T2.read(B), T2.read(A), T1.read(B), T2.commit, T1.commit

- (1) T1.write(A)与 T2.read(A)冲突，优先图箭头从 T1 指向 T2，优先图中不存在环，调度 S1 是冲突可串行化的，等效调度序列为 T1→T2
- (2) S1 是否是可恢复?
S1 是否是?
- (3) T1: X-Lock(A), read(A), write(A), S-Lock(B), read(B), **commit**, Unlock(A), Unlock(B)
T2: S-Lock(B), read(B), S-Lock(A), read(A), **commit**, Unlock(A), Unlock(B)
加锁后，满足两阶段锁协议，保证了可得到可串行化调度
- (4) 不会产生死锁，因为 T1 只对数据项 A 申请排他锁，没有对数据 B 申请排他锁，T2 只对数据项 A，数据项 B 申请读锁，不会造成互相等待的情况。

3. From 陈鹏练习题

There are two transactions:

T1: R(A), R(B), W(B:=6)

T2: R(C), R(B), W(A:=8)

R is read, W is write.

Consider the following schedule:

Schedule S1: T1.R(A), T2.R(C), T1.R(B), T2.R(B), T1.W(B:=6), T2.W(A:=8)

Schedule S2: T2.R(C), T1.R(A), T1.R(B), T1.W(B:=6), T2.R(B), T2.W(A:=8)

1. Are S1 and S2 conflict serializable? Are they serializable? Give the reason for your judgment.

2. Draw precedence graph for S1 and S2.

3. Please add SLock(X), XLock(X) and Unlock(X) instructions for T1 to meet the two-phase locking protocol (2PL)

3. From 陈鹏练习题

There are two transactions:

T1: R(A), R(B), W(B:=6)

T2: R(C), R(B), W(A:=8)

R is read, W is write.

Consider the following schedule:

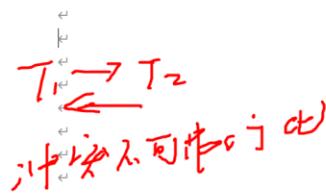
Schedule S1: T1.R(A), T2.R(C), T1.R(B), T2.R(B), T1.W(B:=6), T2.W(A:=8)

Schedule S2: T2.R(C), T1.R(A), T1.R(B), T1.W(B:=6), T2.R(B), T2.W(A:=8)

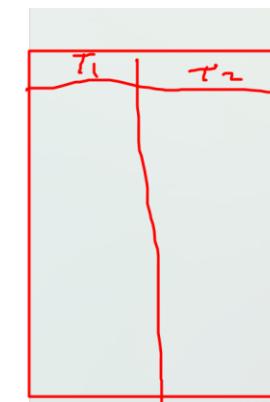
1. Are S1 and S2 conflict serializable? Are they serializable? Give the reason for your judgment.

2. Draw precedence graph for S1 and S2.

3. Please add SLock(X), XLock(X) and Unlock(X) instructions for T1 to meet the two-phase locking protocol (2PL)



T_1	T_2
R.A 1	
	R.C
R.B 1	
	R.B 1
W.B B=6	
	W.A A=8



可串行化
Serializable

4. From 2021-2022 学年第二学期考试题

有如下关于事务 T1 和 T2 的一个调度.

step	T1	T2
1	Read(C)	
2	$C = C - 5$	
3		Read(B)
4		$B = B - 10$
5	Write(C)	
6		Write(B)
7		Read(A)
8		$A = A - 2$
9		Write(A)
10	Read(A)	
11		Commit
12	$A = A / 2$	
13	Write(A)	
14	Commit	

1. 调度是冲突可串行化的吗? 如果是, 它等效于 T1,T2 还是 T2,T1 的串行顺序? 如果不是, 给出简要理由. (5 分)
2. 假定在步骤 1 时, A, B, C 的值分别是 100, 100, 100, 请按“<Ti-Start>,<Ti, 数据名, 旧值, 新值>,<Ti-Commit>”

这样的格式为这个调度写出日志(不记录读操作), 如 “<T1-Start>,<T2,B,45,50>……” (2 分)

3. 如果在步骤 12 后 13 前发生了系统崩溃, 请给出系统重启数据库完成恢复后 A,B,C 的值. (2 分)

调度的序列: (AI 用)

T1.read(C), T1.C=C-5, T2.read(B), T2.B=B-10, T1.wirte(C), T2.write(B), T2.read(A),
T2.A=A-2, T2.write(A), T1.read(A), T2.commit, T1.A=A/2, T1.write(A), T1.commit

5. From 2018-2019 学年第二学期考试题

Consider two transactions T1 and T2.

T1: read(B); A=B+1; write(A)

T2: read(A); B=A+1; write(B)

Consider the concurrent schedule of transactions T1 and T2:

T1	T2
read (B)	
A=B+1	
	read (A)
write(A)	
	B=A+1
	write (B)
Commit	
	commit

1. Give the result of above schedule. (initial value A=2,B=2) . (3points)

2. Is the schedule correct or not? Why? (3points)

3. Add lock and unlock instructions satisfying 2PL to Transactions T1 and T2, to implement a serializable schedule.(4points)

调度序列:

T1.read(B), T1.A=B+1, T2.read(A), T1.write(A), T2.B=A+1, T2.write(B), T1.commit, T2.commit

6. From 2017-2018 学年第二学期考试题

Consider the concurrent schedule of transactions T1 and T2.

T1	T2
read (A)	
	'read (B)
	read (A)
	write (A)
	write (B)
write (A)	
	commit
commit	

1. Is the schedule conflict serializable? If so, give an equivalent serial schedule. If not, give an explain briefly. (5 points)

No. it is not conflict serializable. 3 分 T1's read(A) and write(A) is conflict instruction to T2's write(A). 2 分

2. Add lock and unlock instructions to Transactions T1 and T2, so that they observe the two-phase-locking protocol. Can it results in a deadlock? (5 points)

T1: lock-X(A), read(A), write(A), unlock(A) 2 分

T2:lock-X(B),lock-X(A), read(B),read(A),write(A),write(B),unlock(A),unlock(B). 2 分

It cannot results in a deadlock, for T1 just lock only one item. 1 分

调度序列：

T1.read(A), T2.read(B), T2.read(A), T2.write(A), T2.write(B), T1.write(A), T2.commit,
T1.commit

7. From 2017-2018 学年第二学期考试题

Consider two concurrent schedules **S1** and **S2** of transactions T1 and T2.

T1	T2
read (A)	
write (A)	
	read (A)
	write (A)
read (B)	
write (B)	
	read (B)
	write (B)
commit	
	commit

S1

T1	T2
read (A)	
	read (A)
	write (A)
	read (B)
	write (A)
	read (B)
	write (B)
	commit
	write (B)
	commit

S2

1. Is the schedule **S1** conflict serializable? If so, give an equivalent serial schedule. If not, give an explain briefly. **(5 points)**
2. Is the schedule **S2** conflict serializable? If so, give an equivalent serial schedule. If not, give an explain briefly. **(5 points)**

// ----- //

Consider the following concurrent schedule **S1** of transactions T1 and T2:

T1	T2
read(A)	
write(A)	
	read(B)
	read(A)
read(B)	
	commit
commit	

Answer the following questions based on the schedule S1:

1. Is S1 a conflict serializable schedule? Please give the reason for your judgment. If so, S1 conflict equivalent to which serial schedule? (3 points)
2. Determine whether schedule S1 is a recoverable schedule, is a cascadeless schedule, and give the reasons. (2 points)
3. Please add lock and unlock instructions to T1 and T2 to meet the two- phase locking protocol (2PL). (3 points)
4. Whether T1 and T2 may produce deadlock after adding the two- phase locking protocol(2PL)? Please give the reason for your judgement. (2 points)

- (1) Is S1 a conflict serializable schedule? Please give the reason for your judgment. If so, S1 conflict equivalent to which serial schedule? (3 points)

S1 是冲突可串行化调度，T1 的 read(B) 指令与 T2 的 read(B) 可交换顺序，冲突等价于串行调度 T1, T2。

- (2) Determine whether schedule S1 is a recoverable schedule, is a cascadeless schedule, and give the reasons. (2 points)

S1 是可恢复调度，当 T2 读了 T1 所写的数据 A，有 T2 的 commit 在 T1 的 commit 之后。S1 是无级联调度，当 T2 读了 T1 所写的数据 A，有 T2 的 read(A) 在 T1 的 commit 之后。

- (3) Please add lock and unlock instructions to T1 and T2 to meet the two- phase locking protocol (2PL). (3 points)

T1: lock- X(A) read(A) write(A) lock- S(B) read(B) commit unlock(A) unlock(B)

T2: lock- S(B) read(B) lock- S(A) read(A) commit unlock(A) unlock(B)

加锁后，保证了可得到可串行化调度。

- (4) Whether T1 and T2 may produce deadlock after adding the two- phase locking protocol(2PL)? Please give the reason for your judgement. (2 points)

加锁后，不会产生死锁，因为 T1 只对数据项 A 申请排他锁，没有对数据 B 申请排他锁，T2 只对数据项 A，数据项 B 申请读锁，不会造成互相等待的情况。

There are two transactions:

T1: R(A), R(B), W(B:=6)

T2: R(C), R(B), W(A:=8)

R is read, W is write.

Consider the following schedule:

Schedule S1: T1.R(A), T2.R(C), T1.R(B), T2.R(B), T1.W(B:=6), T2.W(A:=8)

Schedule S2: T2.R(C), T1.R(A), T1.R(B), T1.W(B:=6), T2.R(B), T2.W(A:=8)

1. Are S1 and S2 conflict serializable? Are they serializable? Give the reason for your judgment.
2. Draw precedence graph for S1 and S2.
3. Please add SLock(X), XLock(X) and Unlock(X) instructions for T1 to meet the two-phase locking protocol (2PL)