



Ti	L1(A)	T ₂ :	L ₂ (B)
	L ₁ (B) read(A)	9273 92	read (B)
	read (B)	note:	read (A)
3	if A=0 then		U2 (B)
	write(B)		if $B=0$ then $A:=A+1$
	U1 (A)		Write (A)
通知王	U1 (B)	NATT	U2 (A)
1	7,	Tı	
2	Li(A)	L2(B)	
		read(B)	
		La(A) 相绝	1…等待
	L, (B) 1	三绝…等得	Good

4.

Ti	T ₂	P	9	(A) 1.1 (T
50	100	RT=0 WT=0	RT=0 WT=0	(A) p. (A)
	1/2(p)	RT=100	Train	(B) busy
W1(p)-	0-11		443	→507/00,T,13%
W ₁ (9)	12(9)			了2级联回落
	(8) 39		T1, T2	反复重启,导致了,万微元
T. Carlot			11	

5.

时间	T1	T2	T3
1		LOCK-S(A)	
2		READ(A)	
3		A:=A*3	
4		WRITE(A)	
5		UNLOCK(A)	
6			LOCK-S(A)
7			READ(A)

8		A:=A**2
9		WRITE(A)
10		UNLOCK(A)
11	LOCK-S(A)	
12	READ(A)	
13	A:=A+4	
14	WRITE(A)	
15	UNLOCK(A)	

最终 A=40

6.

定位到 checkpoint {T4,T5},考虑 checkpoint list 中的事务以及之后开始的事务,即 T4,T5,T6。T5 在 checkpoint 前开始,在 checkpoint 之后,故障之前提交,故需要 Redu。T4 在 checkpoint 前开始,故障时没有提交,故需要 Undo。T6 在 checkpoint 后开始,故障时没有提交,需要 Undo。所以 Undo-List:{T4,T6},Redo-List:{T5}。

回复后 A=700, B=1000, C=100。

7.

Undo-List: {T1,T3}, Redo-List: {T2} A=-20, B=-10, C=10, D=-20

8.

目的:在当系统故障发生时,检查日志,决定哪些事务需要重做,哪些事务需要撤销。为了减小搜索过程,使用检查点机制,只对有必要的事务进行处理。可以定期或不定期做一次检查点。如一个小时做一次,或者日志文件写满一半时做一次。

影响:

- (1) 无故障发生时, 频率越高, 检查时间越长, 越影响性能。
- (2)从系统崩溃中恢复,频率越高,恢复时需要应用的重做或撤销的日志就越少,一定程度内会缩短检查时间。
- (3)从介质故障中恢复,由于日志文件存储在介质中,所以频率越高,发现故障就越及时。

9.

(1) 采用嵌套循环连接算法执行 $R \bowtie S$ 分别需要进行多少次 I/O? 给出具体分析过程。

B(R)=20000/20=1000 块

B(S)=60000/30=2000 块

嵌套循环连接:

S作为外层, I/O 次数为 2000+1000*60000=6.0002e7

块嵌套循环连接:

使用 R 作为外层,38 个内存页作为 R 的缓冲。外层循环需要 1000/38=27 轮。对

于 R 需要的 I/O 次数为 1000, 对于 S 需要的 I/O 次数为 27*2000=54000, 共计 55000 次。

(2) 采用归并连接算法执行 R ⋈ S 分别需要进行多少次 I/O? 给出具体分析过程。

R 对外键 B 无序,故首先需要 2B(R)=2000的 I/O 次数进行块内排序。

R 和 S 对属性 B 分别有序后,按序扫描 R 和 S,I/O 次数为 B(R)+B(S)+ B(T)=3000 次.

总共需要 2000+3000=5000 次。

(3)设 R.B 是关系 R 的外键,参照 S.B。如果 R ⋈ S 的结果中元组的平均大小是 R 中元组平均大小的 2 倍,R ⋈ S 的结果中元组的平均大小是 S 中元组平均大小的 1.2 倍,那么在外存中存储 R ⋈ S 的结果需要占用多少个块(页)?给出具体分析过程。

由于 B 是 R 的外键,S 的主键,所以连接后元组个数为 20000。 由 R \bowtie S 的结果中元组的平均大小是 R 中元组平均大小的 2 倍,所以一块可以 容纳 10 个 R \bowtie S 元组。

故需要占用 20000/10=2000 个块。