John Lee-jol59

Miguel Dickson-lmd90

CS 1555 HW 7

1)

a)

With respect to x:

H1: W2(x)->R1(x)->R3(x)->W1(x)

H2: W2(x)->W1(x)->R3(x)->R1(x)

So H1 and H2 cannot be conflict equivalent because they differ for R3(x)->W1(x) versus W1(x)->R3(x).

H3: W2(x)->W1(x)->R1(x)->R3(x)

H2 and H3 are conflict equivalent so far… (both the W2 and W1 writes will happen with respect to x before either of the R1(x) or R3(x); R1(x) and R3(x) do not conflict and can be safely swapped, leading to identical histories with respect to x).

H4: W1(x)->R3(x)->W2(x)->R1(x).

H4 cannot be conflict equivalent to any of H1-H3 (because the leading W1(x) breaks conflict equivalence with any of the three others)

With respect to y:

H1: W2(y)->R3(y)

H2: W2(y)->R3(y)

H3: W2(y)->R3(y)

H4: R3(y)->W2(y)

So H2-H3 remain conflict equivalent with respect to y; H4 further cannot be conflict equivalent to the others.

With respect to z:

H1: R3(z)->R2(z)

H2: R3(z)->R2(z)

H3: R2(z)->R3(z)

H4: R2(z)->R3(z)

Read-only produce no conflicts, so H2-H3 remain conflict equivalent with respect to z.

**Only H2 and H3 are conflict equivalent.**

b)

SG(H1):

T2->T1 (due to W2(x), R1(x))

T3->T1 (due to R3(x), W1(x))

T2->T3 (due to W2(y), R3(y) and W2(x), R3(x))

**So T2->T3->T1 is the equivalent serial history:**

**W2(x)R2(z)W2(y)R3(z)R3(x)R3(y)R1(x)W1(x)**

SG(H2):

T2->T1 (due to W2(x), W1(x))

T2->T3 (due to W2(x), R3(x))

T1->T3 (due to W1(x), R3(x))

So **T2->T1->T3** **is the equivalent serial history:**

**W2(x)W2(y)R2(z)W1(x)R1(x)R3(x)R3(z)R3(y)**

SG(H3):

T2->T1 (due to W2(x), W1(x))

T2->T3 (due to W2(x), R3(x))

T1->T3 (due to W1(x), R3(x))

So **T2->T1->T3** **is the equivalent serial history:**

**W2(y)W2(x)R2(z)W1(x)R1(x)R3(y)R3(x)R3(z)**

SG(H4):

T3->T2 (due to R3(y), W2(y))

T1->T3 (due to W1(x), R3(x))

T2->T1 (due to W2(x), R1(x))

T3->T2->T1->T3 is a cycle, so **NOT SERIALIZABLE**

2)

1) Copy sectors 4-35 to buffer while rotating. (6ms, ½ disk rotation)

2) Spin back to sector 4 (6ms) *simultaneously* move tracks to track 14.

3) Write out sectors 4-35 from buffer to track 14 (6ms, ½ disk rotation)

4) While rotating from 35-51 (3ms) *simultaneously* move tracks to track 12.

5) Copy 52-3 while rotating (3ms, ¼ disk rotation)

6) Skip past 4-35 while rotating (6 ms, ½ disk rotation)

7) Copy 36-51 (3 ms, ¼ disk rotation)

8) While rotating from 52 to 3 *simultaneously* move tracks to track 14 (3 ms, ¼ disk rotation)

9) Skip past sectors 4-35 while rotating (6 ms, ½ disk rotation)

10) Write sectors 36-3 (6 ms, ½ disk rotation).

Total elapsed time: 6+6+6+3+3+6+3+3+6+6 = 48ms = 4 disk rotations.

3)

5)

NOTE: [ ] = nodes. All arrows -> pointers. ( #) ->Node number (in order of creation.)

After inserting 17:

[5] (3)

[2 | 3] ---> [5 | 7 | 8 | 17]

(1) (2)

After inserting 25:

[15] (9)

(3) [5 | 8] [19 | 25] (8)

|

[2 | 3]->[5|7]->[8 | 11]->[15 | 17]->[19|20]->[25| 29 |31]

(1) (4) (5) (6) (7) (2)

After inserting all keys:

[15] (9)

(3) [5 | 8] [19 | 25 | 29 ] (8)

|

[2 | 3]->[5|7]->[8 | 11]->[15 | 17]->[19|20|23]->[25 | 27] [29 |31 | 35 | 38]

(1) (4) (5) (6) (7) (10) (2)