Exercise 9

Released: June 3, 2019 · Discussion: June 17, 2019

1 Serializability

T_1	`read`	`(x)`	
	`write`	`(b)`	
	`read`	`(c)`	
	`write`	`(x)`	

T_2	`read`	`(d)`	
	`read`	`(a)`	
	`write`	`(y)`	

T_3	`read`	`(a)`	
	`read`	`(b)`	
	`write`	`(a)`	
	`write`	`(b)`	
	`write`	`(x)`	

The transactions T_1, T_2 and T_3 shall be executed concurrently. For this purpose a database management system utilizing the two-phase locking protocol is used. The transactions are processed using a round-robin strategy ($T_1, T_2, T_3, T_1, \ldots$), which executes one transaction step for a transaction T_i at a time.

Transaction step

1. Retrieve the next `read/write` operation `op(X)` of T_i.
2. If T_i does not hold the lock for `X`: `lock(X)`.
3. Execute `op(X)`.
4. Enter the release-phase as soon as possible and perform for each object `Y`, not used by T_i anymore, `unlock(Y)`.

If a lock can not be granted for a transaction, the transaction step will be aborted and the transaction acquires the lock in the next regular step where the lock is free.

Assignments

1. Determine the schedule S the DBMS is going to use in order to execute the transactions.
2. Determine all conflicts in the conflict relation of S.
3. To which serial plan is S conflict-equivalent?

2 Compression Implementation

Compression techniques can be used to save a lot of space especially in DBMS using bitmap indices. The Word-Aligned Hyprid (WAH) algorithm uses a kind of run-length encoding but with respect to a specific word size. In this exercise you have to implement functions for **compressing** and **decompressing** a bitstream with the WAH algorithm.
Download the file `05_compression.zip` from the course website[1] and extract it. If you extract it into the folder of the previous project(s) you may have to merge the `CMakeLists.txt` files. You have to complete the functions `compress()` and `decompress()` in the file `src/compression/wah_compression.cpp`.
To test your compression function you can use the files `bitvector_1.bit` and `bitvector_2.bit`.

- Which file benefits more from the WAH compression?
- How much space can be saved by using the WAH compression? Explain your observations.

Hint:
The function to decompress a bitstream might be a bit easier to implement. Therefore the file `compressed_bitvector.bit` can be used to test just the decompression function. The file is located in the workloads folder.

Build instructions:

1. You may have to modify the path to the workloads (e.g. to `../workloads/bitstream_1.bit`)
2. Extract the archive and navigate into the extracted folder.
3. Run `cmake` to create a makefile for your system: `cmake .`
4. Run `make` to create an executable binary file: `make`
5. Execute the created binary file (e.g. `./05_compression` on linux)

[1] `http://dbis.cs.tu-dortmund.de/cms/en/teaching/ss19/arch-dbms/exercises/05_compression.zip`