

Implementing FOCC and BOCC (optimistic) Concurrency Control Algorithms

CS16BTECH11038 Supreet Singh

Introduction

Have implemented the following algorithms:

- BOCC (CTA)
- FOCC CTA: Abort the current transaction if its Validation fails
- FOCC OTA: Abort all other conflicting transactions if current transaction's validation fails.

These have been implemented as per the book **Transactional Informational Systems by Vossen (Chapter 4)**

Some Implementation Details

I have used **Fine Grained Locking!**

Each class has been implemented in a separate file (*.h/*.cpp).

The testing code has been written in opt-test.cpp

The data structures are self-explanatory (comments have been added in the .h files) and are as discussed in the class.

A read-write lock (**shared_mutex** from C++17) has been used for the scheduler, to allow concurrent access to a map storing each transaction.

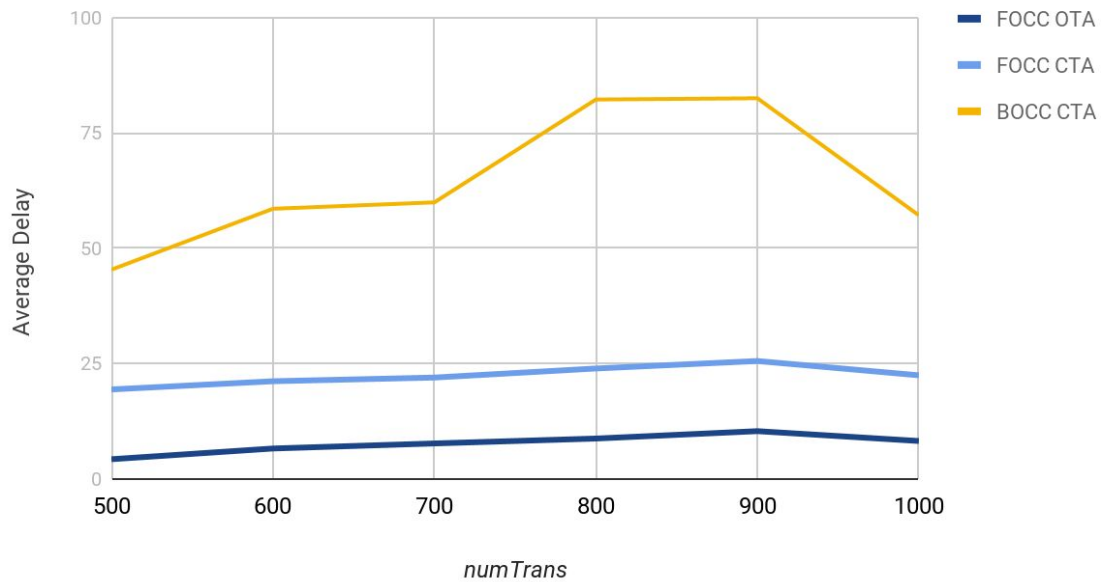
Comparison

For all the comparisons, number of threads was **N = 6**

And the number of data items was **M = 10**.

Comparison of average delay to commit vs numTrans

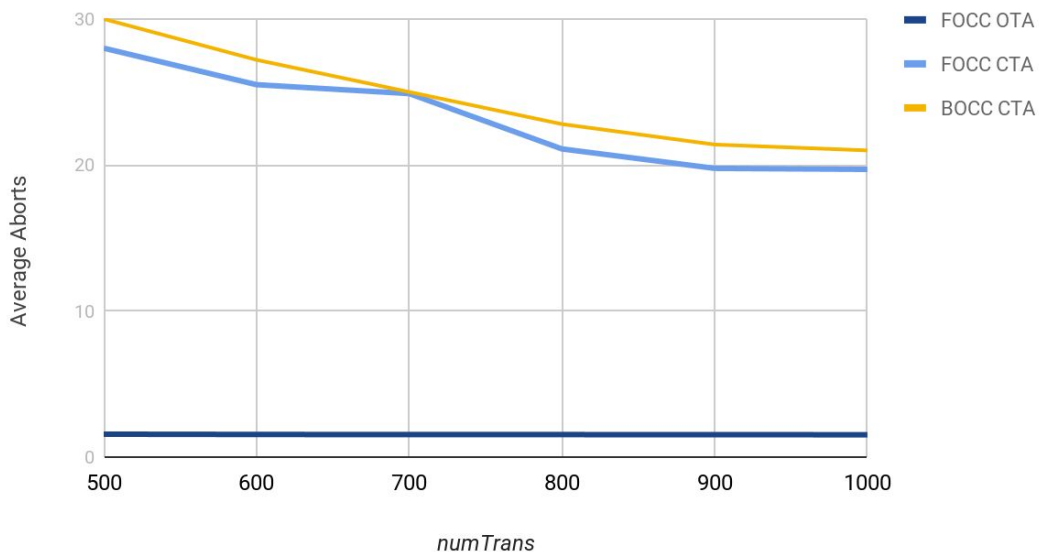
Average Delay



This graph plots average delay to commit vs total number of committed transactions per thread.

Comparison of average aborts to commit vs numTrans

Average aborts vs numTrans



Observations

The average delay to commit increases slightly with increase in number of transactions.

This slight increase could be due to increased time taken in book-keeping operations due to larger number of transactions. Also, there is a lot of variation in the average delay in BOCC algorithm. BOCC performs more book-keeping; it also has to keep track of transaction start/abort time.

FOCC - OTA performs the best of the three algorithms, with very low average delays. BOCC is the worst performing, with average delays high in both magnitude and variation. This might be because it performs additional timestamp checking in validation phase (it checks the transactions that have committed after the start of the current transaction).

Average number of aborts remains roughly constant, decreasing slightly for all the algorithms. This might be due to the fact that the total number of aborts are not changing as fast as the number of transactions are increasing.

FOCC - OTA is again the clear winner here, with average aborts roughly = 1.54. That is, a transaction on average is aborted only 1.5 times before being committed. This number is substantially higher for FOCC CTA and BOCC - it is in the range 20-30 for these algorithms. This also explains their high average delays.