Programming Assignment 2
CS5280

Darpan Gaur
CO21BTECH11004

Overview

Implemented BTO, MVTO, MVTO-gc and k-MVTO using fine grained locking i.e., for operations read(), write() and tryCommmit() a lock on data-item is used, instead of a global lock. For all experiments, $numIters = rand()\%500$ is taken instead of $numIters = rand()\%m$ as it was taking too long to run.

Number of transactions

Parameters used to generate the data: $n = 16$, $m = 1000$, $constVal = 100$, $\lambda = 20$, $readRatio = 0.7$, $numTrans = [1000, 2000, 3000, 4000, 5000]$

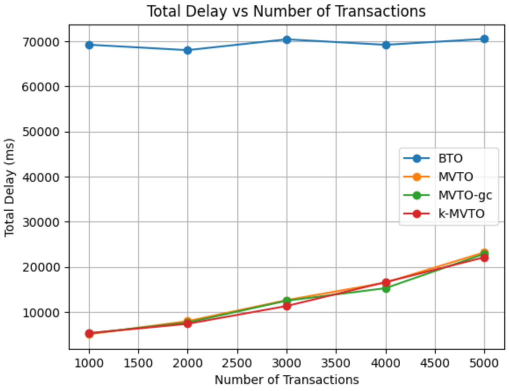

Figure 1: Commit Delay (BTO - MVTO)

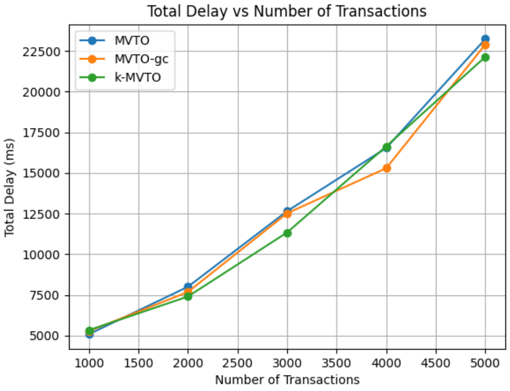

Figure 2: Commit Delay (MVTO)

Figure 1 shows the commit delay for BTO and MVTO varients, while figure 2 shows the commit delay for MVTO varients.

- Commit delay for BTO is more than all MVTO varients, as there are more number of aborts in BTO than MVTO.

- Commit delay remains almost constant for BTO with increase in number of transactions.

- For MVTO varients commit delay increases with increase in number of transactions, as it takes more time to check to abort or commit a transaction.

- For MVTO-gc and k-MVTO, commit delay is slightly less than MVTO, due to garblage collection and limited versions respectively, which reduces the time for tryCommmit.

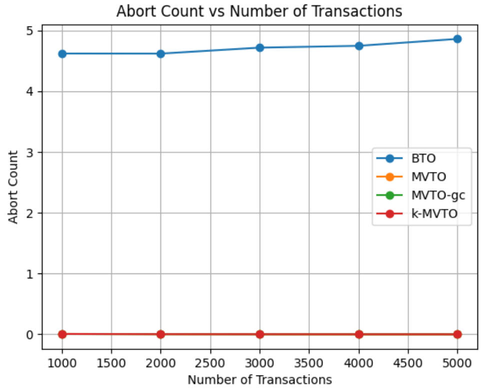

Figure 3: Abort Count (BTO - MVTO)

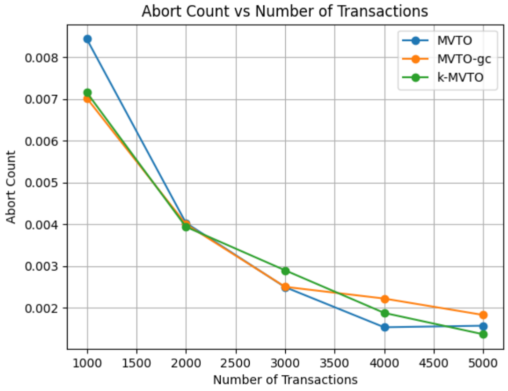

Figure 4: Abort Count (MVTO)

Figure 3 shows the abort count for BTO and MVTO varients, while figure 4 shows the abort count for MVTO varients.

- Abort count for BTO is more than all MVTO varients, as we check with latest version.

- Abort count remains almost constant for BTO with increase in number of transactions.

- For MVTO varients abort count decreases with increase in number of transactions.

Number of variables

Parameters used to generate the data: $n = 16$, $numTrans = 1000$, $constVal = 100$, $\lambda = 20$, $readRatio = 0.7$, $m = [1000, 2000, 3000, 4000, 5000]$.
Figure 5 shows the commit delay for BTO and MVTO varients, while figure 6 shows the commit delay for MVTO varients.

- BTO has high commit delay than MVTO varients, as there are more number of aborts in BTO.

- For both BTO and MVTO varients, commit delay decreases with increase in number of variables, as less more variables less conflicts.

- For k-MVTO, commit delay is least as it has limited versions, which reduces the time for tryCommmit.

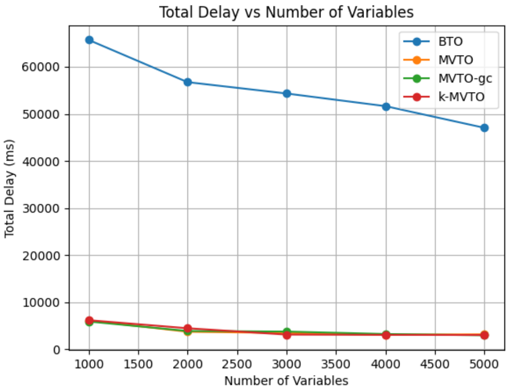

Figure 5: Commit Delay (BTO - MVTO)

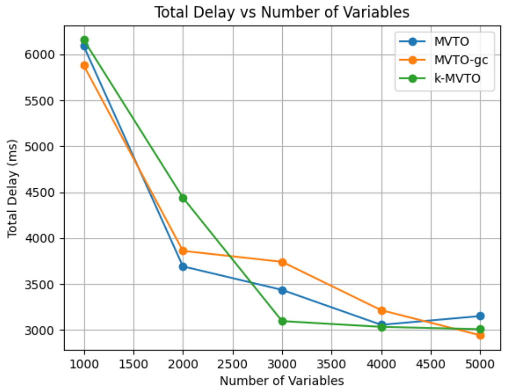

Figure 6: Commit Delay (MVTO)

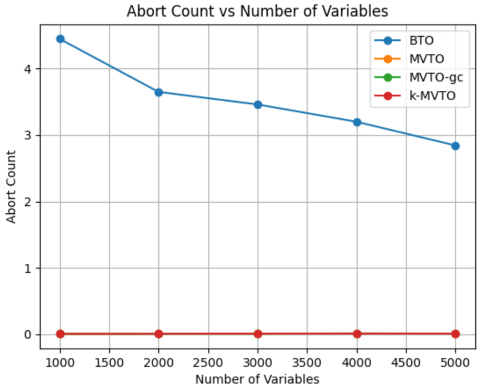

Figure 7: Abort Count (BTO - MVTO)

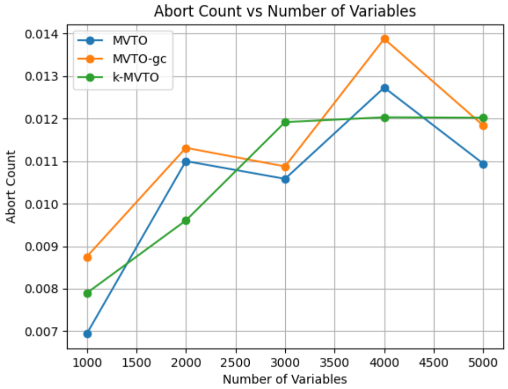

Figure 8: Abort Count (MVTO)

Figure 7 shows the abort count for BTO and MVTO varients, while figure 8 shows the abort count for MVTO varients.

- Abort count for BTO is more than all MVTO varients, as we check with latest version.
- Abort count decreases with increase in number of variables for BTO, as less more variables less conflicts.
- For MVTO varients, abort count remains almost constant (slight increase) with increase in number of variables.

Number of threads

Parameters used to generate the data: $n = 16$, $m = 1000$, $numTrans = 1000$, $constVal = 100$, $\lambda = 20$, $readRatio = 0.7$, $numThreads = [2, 4, 8, 16, 32]$
Figure 9 shows the commit delay for BTO and MVTO varients, while figure 10 shows the commit delay for MVTO varients.

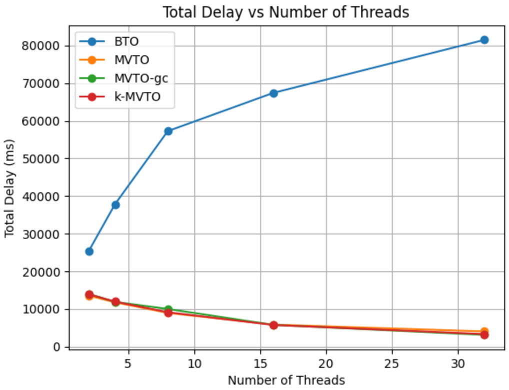

Figure 9: Commit Delay (BTO - MVTO)

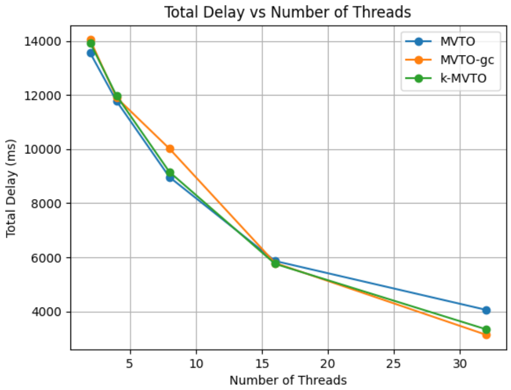

Figure 10: Commit Delay (MVTO)

- BTO has high commit delay than MVTO varients.
- For BTO commit delay increases with increase in number of threads, as more threads more conflicts.
- For MVTO varients, commit delay decreases with increase in number of threads, as more versions less conflicts.

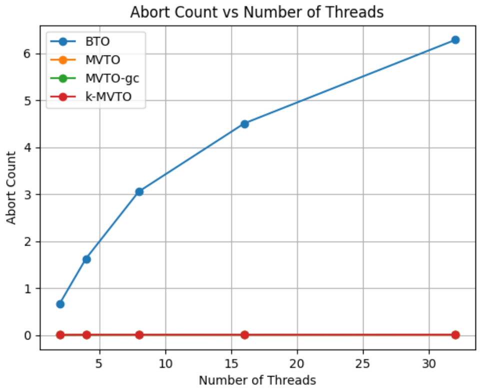

Figure 11: Abort Count (BTO - MVTO)

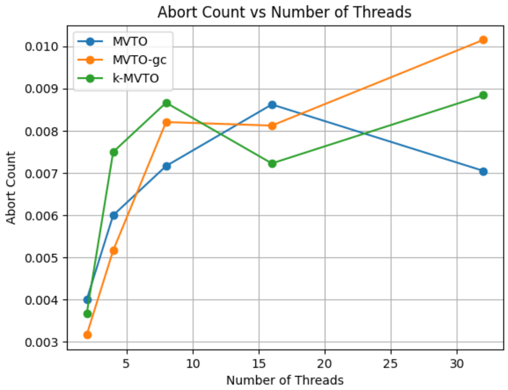

Figure 12: Abort Count (MVTO)

- Abort count for BTO is more than all MVTO varients.
- Abort count increases with increase in number of threads for BTO, as more threads more conflicts.
- For MVTO varients, abort count remains almost constant (slight increase) with increase in number of threads.

K for K-MVTO

Parameters used to generate the data: $n = 16$, $m = 1000$, $numTrans = 1000$, $constVal = 100$, $\lambda = 20$, $readRatio$, $K = [5, 10, 15, 20, 25]$

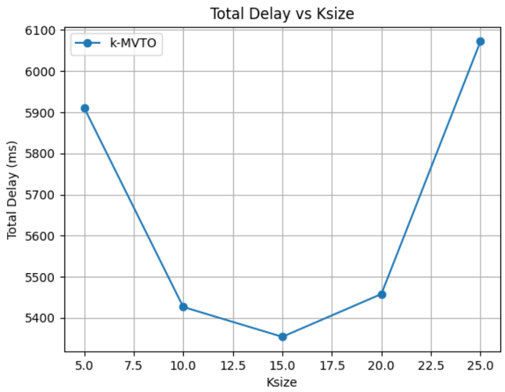

Figure 13: Commit Delay (BTO-MVTO)

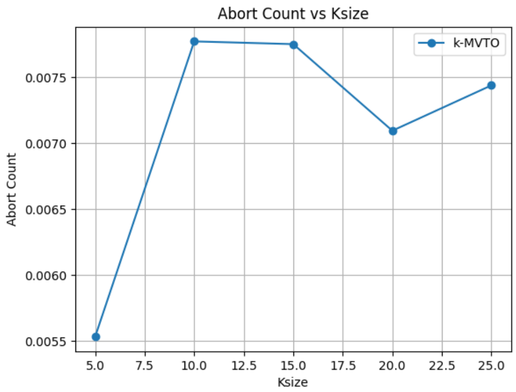

Figure 14: Abort Count (BTO-MVTO)

- Total delay first decreases and then increases with increase in K.
- Abort count reamins constant (slight increase) with increase in K.

Read Ratio

Parameters used to generate the data: $n = 16$, $m = 1000$, $numTrans = 1000$, $constVal = 100$, $\lambda = 20$, $readRatio = [0.5, 0.6, 0.7, 0.8, 0.9]$

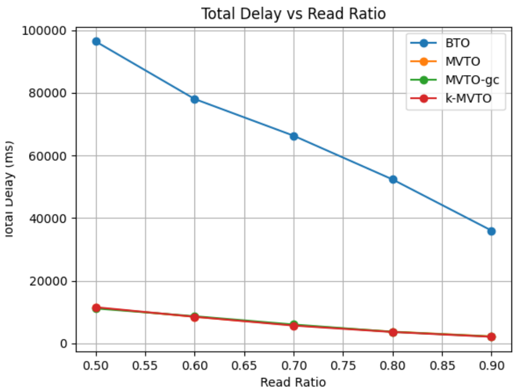

Figure 15: Commit Delay (BTO-MVTO)

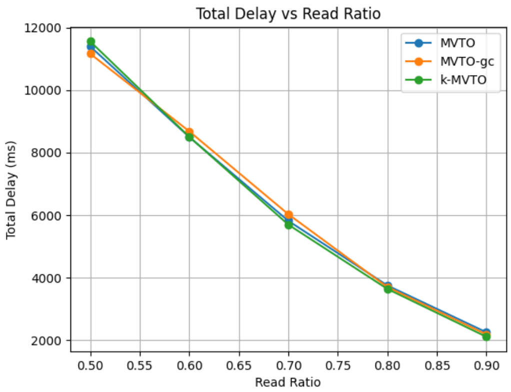

Figure 16: Commit Delay (MVTO)

- Commit delay for BTO is more than all MVTO varients, as there are more number of aborts in BTO.
- Commit delay decreases with increase in read ratio for both BTO and MVTO varients, as more reads less conflicts.

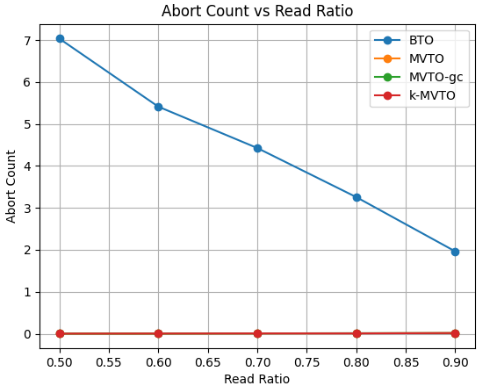

Figure 17: Abort Count (BTO-MVTO)

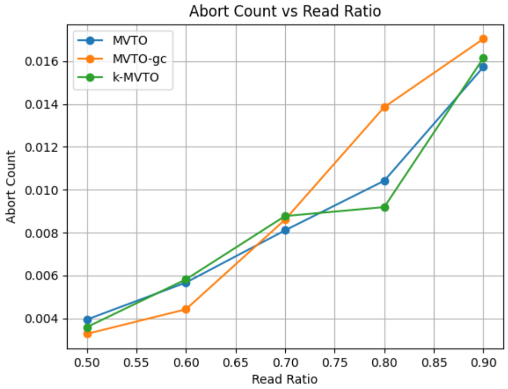

Figure 18: Abort Count (MVTO)

- Abort count for BTO is more than all MVTO varients, as we check with latest version.
- Abort count decreases with increase in read ratio for BTO, as more reads less conflicts.
- For MVTO varients, abort count remains almost constant (slight increase) with increase in read ratio.