

03/02/2012 13:50

### Late write - Late conflict Insert

- Generate Row ID

FEC:

- Write to Transactor (Txactor.Apply)

~~~~ other trx - read from buff, apply txactor ~~~~

- Pre detect to optimize time in critical section

- Prepare (TM.Prepare → rev)

FEC:

- Detect conflicts (revisor.detect(revisor.changes) → [error])

FEC: - Start persist (store.log(revisor.changes, affected colids))

FEC:

- Begin Commit (Buff.Begin)

- Write change (Buff.Apply)

- End commit (Buff.End(Rev))

- Commit Transaction (TM.Commit)

- Join Persist

- ...

### Column Locker

- Holds pending {TID, RowID}

### Column Transactor

- Stores Trx changes

- Commits

- Redoes

### Insert 1 row

- Immediate write → late conflict

- Generate Row ID (Generator.NewRow → RowID)

FEC:

- Write to Transactor (Txactor.Apply → brev)

- Write to buffer (Buff.Apply(brev))

~~~~ other trx ops ~~~~

- Prepare (TM.Prepare → rev)

FEC:

- Detect conflicts (Txactor.changes → revisor → [error])

- Commit (TM.Commit)

FEC: - Storage commit (store.commit(affected colids))

FEC:

- Commit (Buff.Commit(row))

- Return if picky

Part of engine

could be distributed (against current log)

could optimize to apply if small

T!