トップエスイー修了制作

DBMSの開発における整合性検証のモデル検査適応検討

株式会社　日立製作所　　礒田 有哉　　yuuya.isoda.sj@hitachi.com

開発における問題点

ビッグデータ市場の拡大により，高性能なデータベース・マネージメントシステム(DBMS)の要求が高まっている。また，半導体技術の進歩により，より高性能なDBMSの開発が可能となりつつある。これらの技術を導入するには，DBMSを再設計する必要がある。このとき，DBMSで最も重要な整合性の検証に多大な時間が掛る課題がある。

手法・ツールの適用による解決

DBMSの整合性は，複数のロックを用いることによって実現できる。このため，整合性検証には，如何なる状況においても，ロックが正しく動作することを確認すればよい。また，DBMSでは，トランザクション処理(Tr)が成功又は，失敗するため，成功かつロックが正しく取得できていないときに不整合な現象が発生するといえる。この検証をモデル検査手法のSPINを用いて実装し，不整合な現象を正しく検出できることを確認した。

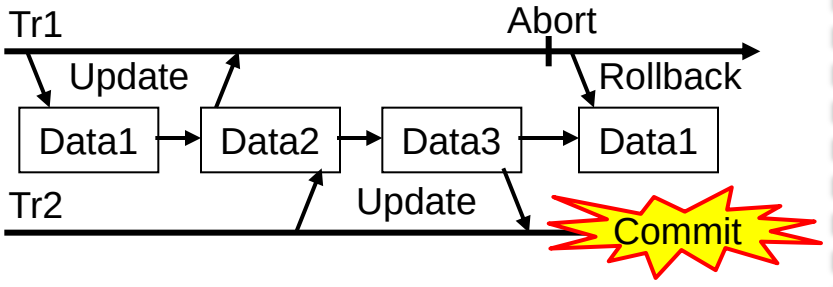
DBMSの整合性

■ 整合性

- DBMSは，不整合な現象を抑止する必要がある。
  例）Dirty Write, Dirty Read, Phantom, etc.

■ Dirty Write

- Tr2の更新がTr1によってロールバックされたにも関わらず，Tr2がコミットできてしまう現象



検証方法

■ 状態遷移と進行性の保存による検証

- 時相論理(LTL式)で複数のTrを表現することは困難
- データの状態遷移とTrの進行性を構造体で保存し，Trのコミット時に構造体をチェックすることで検証

● データの状態遷移の保存

```
inline DBMS_OP_INCREMENT(DATA, Tr_ID){
LOCK_MANAGER(Tr_ID, DATA);
ASSERT_MANAGER(Tr_ID, DATA);
INCREMENT(DATA);   }
```

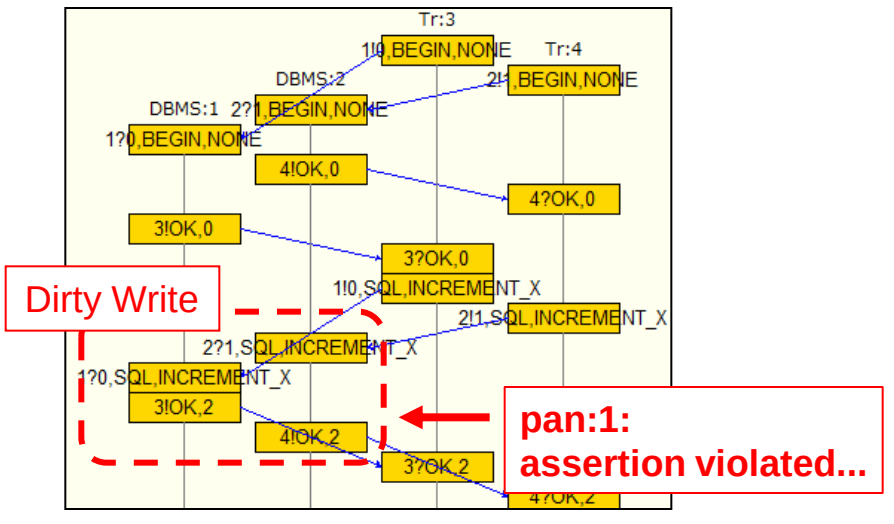
● コミット時の検証

```
inline DBMS_Commit(Tr_ID){
DBMS_Judge_Commit(Tr_ID);
DBMS_Judge_Phenomena(Tr_ID);
FREE_ASSERT_MANAGER(Tr_ID);
UNLOCK_MANAGER(Tr_ID);   }
```

検証結果

■ Lock Less Mode

- SPINを用いて，Dirty Writeを正しく検出できる。



まとめと今後の課題

■ まとめ

- ロックによる整合性検出方法を用いることによって，不整合な現象を正しく検出できた。また，ロックを用いる処理で汎用的に利用することが可能である。

■ 今後の課題

- Read Lockを用いないDBMSアルゴリズムにおいても正しく不整合な現象を検出できることを確認し，DBMS開発のアルゴリズム評価に活用していく。