直列化可能性

直列化可能性 (Serializability)

同時実行を行った結果が正しいのかどうかを判断する基準が必要

- 複数のトランザクションを同時実行　(非直列スケジュール)した結果とそれらを何らかの順序で逐次的に処理　(直列スケジュール)した結果が一致すれば正しいと考えられる

  (この条件は十分条件である)
- このとき，非直列スケジュールは直列化可能であるという

直列化可能であることの判定は

- 非直列スケジュールと等価な直列スケジュールを見つけることができるか？
  - n個のトランザクションの直列スケジュールの数は $n!$
    → nが大きくなると天文学的数字に
- 等価性をどこまで求めるかで計算時間が変わる

等価性のレベル

下に行くほど条件が厳しく，直列化可能の判定時間は短くなる

- 最終状態直列化可能
  - データベースの最終状態が等価である
- ビュー直列化可能
  - 各*read* が同じ値を読み，最終状態が等価である
- 相反直列化可能
  - 非直列スケジュール内の競合する操作の順序が等価な直列スケジュール内でも同じ順序になる

競合する操作とは

順序を入れ替えると結果が異なる可能性がある操作の組

→ 同じデータに対する操作の少なくとも一方が書込み

相反直列化可能性の判定

- 相反グラフ(conflict graph) を用いて判定
  1. トランザクションを節点とする
  2. 非直列スケジュール S の中に競合する操作 (op, op') が存在し，その順序が op, op' であるとき，op を含むトランザクション T から op' を含むトランザクション T' に対して有向辺 ($T \rightarrow T'$) を設ける
- S の相反グラフに**巡回閉路を含まなければ S は直列化可能**である
  - 競合しない操作を入れ替えることで，等価な直列スケジュールを得ることができる

相反グラフの例

系列1

$r_1(A)\ r_2(C)\ w_1(A)\ r_1(B)\ w_2(C)\ w_1(B)\ r_2(B)\ w_2(B)$

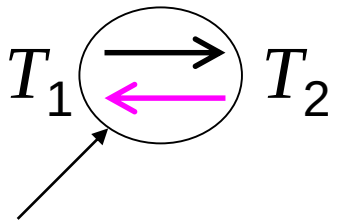
$T_1 \longrightarrow T_2$

$w_1(B)$ と $r_2(B)$

$w_1(B)$ と $w_2(B)$

系列2

$r_1(A)\ r_2(C)\ w_1(A)\ r_1(B)\ w_2(C)\ r_2(B)\ w_2(B)\ w_1(B)$



巡回閉路が発生

$r_1(B)$ と $w_2(B)$

$r_2(B)$ と $w_1(B)$

$w_2(B)$ と $w_1(B)$