

1 R-4

マルチメディアデータモデルを考慮した 同時実行制御の一考察

坂上 雄一[†][†] 広島大学大学院工学研究科宮尾 淳一[‡][‡] 広島大学総合科学部

1 はじめに

マルチメディアデータベースにおけるデータモデルは従来のデータモデルと異なり、動画像のような大規模で実時間性のあるデータを取り扱う必要がある。このようなデータモデルにおける同時実行制御を従来手法で行なうと並列性が低下するなどの問題点が生じる。

本研究では、マルチメディアデータモデル(画像)を考慮したスケジューリング手法を提案する。また、不整合(inconsistent)な状態を許容可能なものと不可能なものに分けることで、並列性の向上を目指す。

2 提案手法概要

提案手法では、動画像・音声などを含むマルチメディアデータの特徴として次のC1, C2に注目する。

- (C1) 非常に大規模なデータがある。
- (C2) 処理を開始すると中止(abort)・再実行が困難なものがある。

このような特徴をもつデータに従来の同時実行制御を適用すると、基本的に次のような問題点P1, P2が生じる。

- (P1) 悲観的手法において、大規模データにロックを掛けると並列性が著しく低下する。
- (P2) 楽観的手法では、処理途中で中止させられる可能性がある。

本手法では、悲観的手法と楽観的手法を組み合わせ、さらに、トランザクションを中止可能なもの(type2)と、中止不可能なもの(type1)に分けること、及び許容不整合の概念を導入することにより、上述の問題点P1, P2の解消を図る。ここで、許容不整合とは、トランザクション処理の交差実行により、データベースの処理が不整合(inconsistent)状態になつたとされる場合でも、その程度によりまったく影響を与えないか、あるいは影響が軽微であり不整合が無視できる場合である。例えば、動画像再生などの場合には連続した複数のフレームに誤りが混入しなければ無視できる場合が多いため、許容不整合となる可能性が高いと考えられる。

本手法の概要を説明する。type1トランザクションは拡張

した2相ロックで処理を行う。type2トランザクションは読み書きするデータセットにtype1とは異なるロックを行い、拡張した直列化可能グラフ(Serialization Graph)を構成しつつ処理を行う。但し、SGグラフでサイクルが生じる場合は、中止か不整合無視を適切なトランザクションに問い合わせせる。

許容不整合と同様な概念は文献[1]で述べられているが、[1]の条件判定はスケジューラが静的に行なうのに対し、本手法では、トランザクションがその状態において動的に判断する。これにより、正確な判断と並列性向上が期待できる。

3 システムモデル

3.1 マルチメディアデータモデル

本稿では、個々のマルチメディアシステムに捕われることなく同時実行制御を議論するため、抽象化したデータモデルを採用する。以下では、本稿の記述に関連したデータモデルを示す。

- (1) 単純データ x: 従来のデータベースで扱うデータで、データ項目 x は値 V(x)を持つ。
- (2) 時系列データ X: 動画像、音声などの一連のデータであり、X はさらにフレーム X(f₁), X(f₂), …, X(f_n) から構成され、各フレーム単位で読み込み、書き込みなどの操作を行うことができる。

3.2 データベースシステムモデル

ここでは、図1に示されるようなシステムモデルを考える。通常のモデルとは異なり、トランザクションマネージャ(TM)とスケジューラを統合したスケジューラ(Integrated Scheduler)を考える。トランザクション・スケジューラ間はメッセージ通信により処理を行なう。

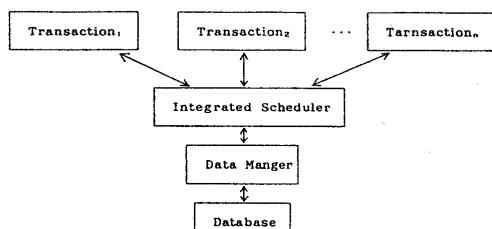


図1 データベースシステムモデル

3.3 トランザクションモデル

トランザクションは、中止 (abort) の可能・不可能、許容不整合の無視 (ignore) 可能・不可能の 2 つの概念により、4 種類のタイプのトランザクションに分類される。これらを本手法では、表 1 に示されるように、type1, type2 の 2 種類に分ける。

中止可能・不可能というのは、アボートを実行することができるかどうかを表す目安である。無視可能・不可能というのは、許容不整合を認めることができるか、全く認められないということを表している。

また、オペレーションの種類として無視 (Ignore) が増えた以外はトランザクションの定義としては文献 [2] に従う。また、ヒストリの定義も文献 [2] に従う。そして、トランザクションは意味的に正しく定義されており、メッセージに対して正しく応答すると仮定し、以下の R1-R3 を満たすとする。

- (R1) 各トランザクションは他のトランザクションを妨害せずに共有データにアクセスする。
- (R2) もしトランザクションが正常終了するならば、その影響は永続性を持ち、もしトランザクションが正常終了しなければ、全く影響を残さない。
- (R3) もしトランザクション自身がアボートしなければ、トランザクションの出力結果と実行結果としてのデータベース状態 (database state) は正しい。

表 1 トランザクションタイプモデル

abort 可能	ignore 可能	type
False	False	type1
True	False	
False	True	
True	True	type2

3.4 メッセージ構造

メッセージにはトランザクションからスケジューラへのメッセージ (TS-message) と、スケジューラからトランザクションへのメッセージ (ST-message) の二種類が存在する。TS-message は (1)start-up-message, (2)operation-message であり、ST-message は (3)conflict-message である。

(1)start-up-message はトランザクションが実行を開始する場合のメッセージであり、read-set (読み読むデータ集合), write-set (書き込むデータ集合) を宣言する。

(2)operation-message は各データ項目に対する操作、中止・無視などの判定を行う。

(3)conflict-message は不整合が発生したことを、トランザクションに通知し、無視あるいは中止のいずれかの operation message を要求する。

4 スケジューリングアルゴリズム

ここでのスケジューリング手法はロッキングを基本にした

ものを提案する。type1 トランザクションの実行方針はアクセスするすべてのデータ項目に対するロックを最初に一括して行なうことであり、type2 トランザクションの実行方針はアクセスするすべてのデータ項目に対するチェックを最後に一括して行なうことである。ロックとチェックの違いは、アクセス透過性の違いを表している。チェックは他の操作に対してブロックされにくい、つまり、アクセス透過性が高いと言える。Lock & Check 両立性行列は表 2 のようになる。また、アルゴリズムで用いる labeled-SG (ラベル付き直列化可能グラフ) について説明する。labeled-SG は従来の SG (Serialization Graph) の枝に処理内容 (r または w) を表すラベルの付いたものである。すなわち、枝の終点のトランザクションの競合オペレーションが読み込みの場合 r(r-labeled edge), 書き込みの場合 w(w-labeled edge) となる。

- S1: スケジューラはキューからメッセージを取り出す。メッセージが start-up-message の場合 S2 へ進む。メッセージが operation-message の場合 S3 へ進む。
- S2: 発信元のトランザクションを活性化 (active) トランザクションとして登録する。もし、そのトランザクションが type1 ならば、read-set, write-set の情報をもとにアクセスする全データをロックする。ロックできなければメッセージをキューに戻す。
- S3: 発信元のトランザクションが type1 の場合 S4 へ、type2 の場合 S5 へ進む。
- S4: labeled-SG を作り、オペレーションを実行する。オペレーションの実行後、これ以後アクセスされることのないデータ項目については解放する。オペレーションがコミットの場合は、終了するトランザクションに関する全情報をスケジューラ内から削除する。
- S5: オペレーションの実行前に labeled-SG にサイクルができるかどうか検査する。サイクルができなければ、データ項目をチェックし、オペレーションを実行する。サイクルができた場合、サイクル中の type2 トランザクションの中で実行開始時刻が最も遅いトランザクションに conflict-message を送る。

表 2 Lock & Check 両立性行列

		rc	wc	rl	wl
read	check	○	○	○	○
write	check	○	○	×	×
read	lock	○	×	○	×
write	lock	○	×	×	×

なお、本研究の一部は平成 4 年度文部省科学研究費補助金一般研究 (B) (課題番号 04452195) 分担課題による。

文献

- [1] Kun-Lung Wu, et al. : "Divergence control for epsilon-serializability", Proc. Data Engineering, pp.506-515(1992).
- [2] P. A. Bernstein, et al. : "Concurrency Control and Recovery in Database Systems", Addison-Wesley(1987).