

マルチメディアDBMSにおける長大データ管理について

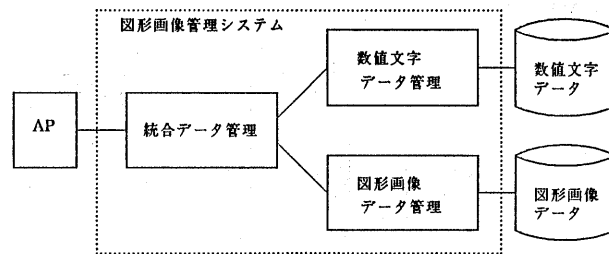
岸本 義一、長浜 芳寛、杉山 守

NTT 電気通信研究所

1. はじめに

近年、地図情報処理、図面管理、原文情報提供サービスなど図形画像情報を取り扱う情報処理分野が急速に進展してきている。これらの分野では、数値文字データと伴に図形画像データも統一的にデータベース化する必要が生じてきている。しかし、従来のデータベース管理システム(DBMS: Database Management System)は、数値文字データの効率の良い管理と操作を主目的に開発されてきているため、これらの分野では図形画像データの効率の良い管理と操作に向けたデータ管理とDBMSを併用した専用の図形画像管理システムを個別に構築しているのが現状である(DIMAP^[1], IMDS^[2], LANDSA T^{ソフトウエア}・システム^[3]等)。(図1) 応用分野の拡大に伴い、数値文字データと図形画像データを統一的に管理できる汎用のDBMSの必要性が高まっており、この要求に応えるため、Haskin & Lorie^[4]はリレーショナルDBMSを拡張して図形画像データ実体を長大データとして格納操作できるようにする提案をしている。

本稿では、図形画像データ実体を長大データの形態で扱うHaskin & Lorieの提案に基づき、DBMSによる効率の良い長大データ管理法について述べる。以下、2章では、汎用DBMSに対する長大データ管理の要求条件について、3章では、要求条件に基づく長大データの格納構造と格納制御方式について示す。



AP: アプリケーション・プログラム

- 数値文字データ管理: 数値文字データを管理する汎用DBMS (通常リレーショナルDBMS)
- 図形画像データ管理: 図形画像データを高速にアクセスできるようにした専用のデータ管理システム
- 統合データ管理: 数値文字データと図形画像データとの関連付け及び更新時の排他制御やリカバリ制御を矛盾なく行えるようにした専用の統合データ管理システム

図1 現状の専用化された図形画像管理システムの構成例

2. 長大データ操作と従来DBMSへの要求条件

2.1 長大データ操作

DBMSが管理する長大データの特徴を表1に示す。

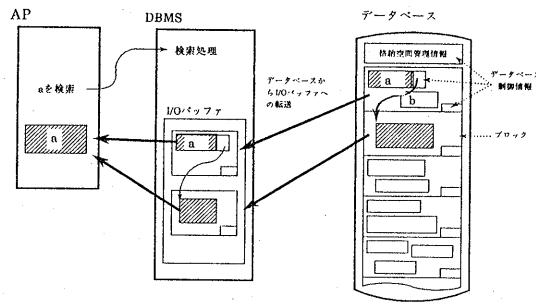
表1 長大データの特徴

項番	項目	特徴
1	データの内容	<ul style="list-style-type: none"> ・イメージ情報のような非フォーマット化データ ・図形要素の集合データ
2	データ長	<ul style="list-style-type: none"> ・数10K～数Mバイトの任意長の長大データ
3	データ・アクセス形態	<ul style="list-style-type: none"> ・APが利用できる主メモリ量の制約より、長大データは任意の長さに分割してアクセスされる。 ・長大データは入力装置から分割してAPに転送され、APはその都度データベースに格納処理を行うため、DBMS側では事前に長大データの全体長が分らない。

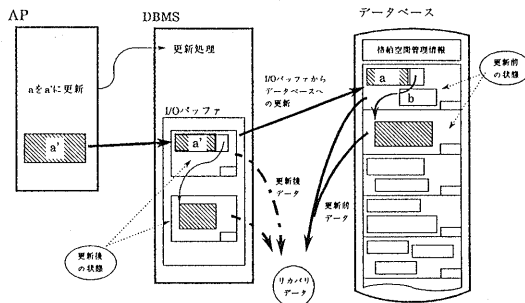
2.2 問題点と要求条件

従来のDBMSでのデータ管理・操作方法を以下に示す。(図2)

- ① データベースの格納空間は一定長のブロックで分割され、各ブロックには数値文字データから構成される複数のレコードが格納される。また各レコードの管理をDBMSが行えるようにデータベース制御情報が各ブロック中に存在する。
- ② データの検索時はファイルから1ブロック単位にデータベースのI/Oバッファに転送した後、APに該当レコード中の数値文字データが転送される。
- ③ データベース更新時にリカバリ・データとして更新前データ(BI:Before Image)/更新後データ(AI:After Image)を取得する。



(a) 検索処理の例



(a) 更新処理の例

図2 従来のDBMSでのデータの管理と操作方法

以上の方式を長大データに適用すると以下の問題点が生じる。

- [問題点1] 1つの長大データが非常に多くのブロック(数10~数100)に分割して格納されるためブロック単位のI/OではI/O回数が非常に多くなり処理時間がかかる。
- [問題点2] データのアクセス時にデータベースのI/Oバッファを介してデータ実体の抽出(検索時)、データベース制御情報の付加(格納時)を行うため、長大データでは特にDS増が問題となる。
- [問題点3] 数値文字データと同様に長大データのBI/AIを取得していたのでは、リカバリ・データの取得量が膨大化(数10~数100倍)する。

以上より、長大データを効率良く管理・操作するための汎用DBMSに対する要求条件は以下の3点に絞られる。

- (要求条件1) ブロック単位にデータのアクセスを分割しないこと
- (要求条件2) データの編集加工操作を不要とするため、データ実体を分離すること
- (要求条件3) リカバリ・データの取得量を削減すること

3. 長大データの格納構造と格納制御方式

本章では、上記要求条件を満足する長大データの格納構造と格納制御方式の実現法について示す。

3.1 長大データの格納構造

格納構造の検討にあたり、先ず要求条件2の実現を図る。このためには、長大データのデータ実体とデータベース制御情報を同一ブロック内に混在させないことが必要となる。これには、以下の2案が考えられる。

- (案1) 格納空間を1つとし、ブロック単位でデータ実体とデータベース制御情報を分離する。
- (案2) データ実体とデータベース制御情報の格納空間を完全に分離する。

(案1)の場合、以下に示す2つの問題点がある。

[問題点1] ブロック種別判定処理の追加によるデータ操作の複雑化

(案1)の場合、長大データ格納ブロックとそれ以外のデータの格納ブロックが混在することから、各ブロックが長大データ格納ブロックかそうでないかを表示する情報(例えば、ビットマップ)が必要となる。ブロック種別の判定のために、長大データだけでなく数値文字データにあってもビットマップを参照する必要があるため、データ操作が複雑となり、DSが増大する。

【問題点2】 データベース構成の柔軟性の阻害

格納コストの低減化のために、長大データの格納媒体に追記型光ディスクや光ディスク・ライブ러리等を使用する場合があります。しかし、1つの格納空間中でのデータ分離では、長大データと数値文字データの各々に適した格納媒体の選択ができない。

(案1)の場合上記問題があることから、(案2)を採用する。但し、(案2)の採用に当っては、APが格納空間の分離を意識する必要がないように、論理構造レベルでは1つの格納空間(これをDBSPACEと呼ぶ)とし、格納構造レベルでは主DBSPACEと補助DBSPACEの2つに分離する(図3)。

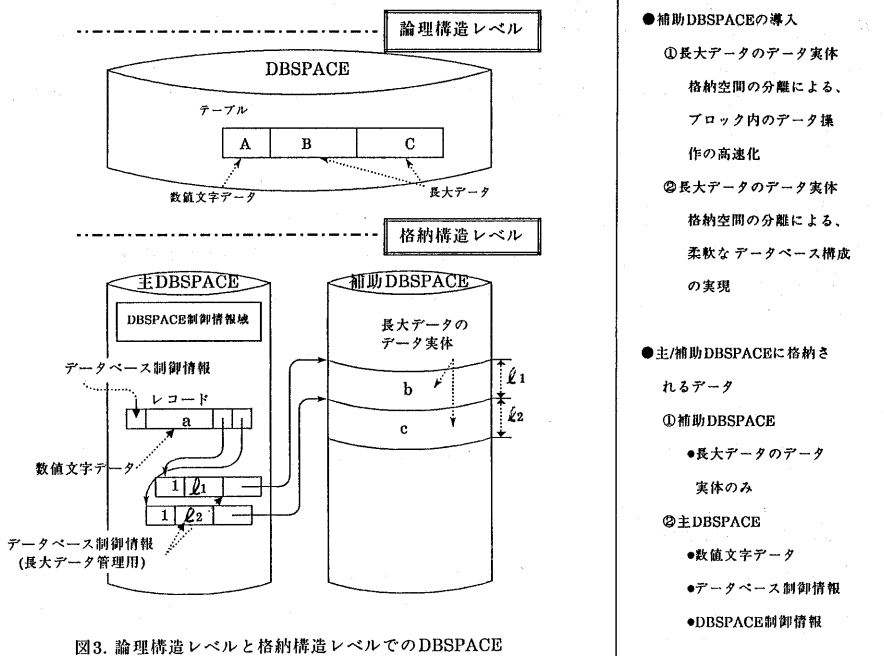


図3. 論理構造レベルと格納構造レベルでのDBSPACE

3.2 補助DBSPACE空間に対する格納制御方式

次に要求条件1と3を実現する補助DBSPACE空間に対する格納制御方式について示す。

(1) 連続空間への格納制御

長大データのアクセスにおいてI/O回数を削減するためには、OSのデータ管理が提供するマルチブロック一括アクセス機能が利用できる。そのためには、予め連続するブロックに長大データを格納しておく必要がある。この際、2章で述べた長大データのアクセス特性として、個々のAPが利用できるメモリ量の制約より一般に数ブロックから数10ブロック単位で分割して格納せざるを得ないことから、以下の点を考慮する必要が生じる。

(i) 未使用空間への優先的格納

補助DBSPACE内には一度もデータが格納されていない未使用ブロックと長大データの

更新/削除処理により生じた再使用可能ブロックが混在している。長大データの全体長をDBMSが予測できないことから、連続域格納が保証される『未使用ブロックへ優先的に格納する』制御が必要となる。ビットマップのみによる使用済空間の管理では、未使用空間の先頭を容易に識別できないことから、これを管理するための制御情報が必要となる。これをFSP (Free Space Pointer) とよび、主DBSPACE中に設ける。

(ii)長大データの並列格納の抑止

長大データの全体長を予測できないことから、複数のトランザクションで並列に長大データを格納する場合、各長大データを連続ブロックに格納することが困難である。検索時の処理時間を短縮するためには、『1つの補助DBSPACEに対しては、長大データの並列格納を抑止する』制御が必要となる。また、この制御を行うと更新トランザクションのコンカレンシの低下が問題となるため、1つの主DBSPACEに対して複数の補助DBSPACEを用意して、DBMSで長大データ対応に補助DBSPACEを振り分ける制御が新規に必要となる。

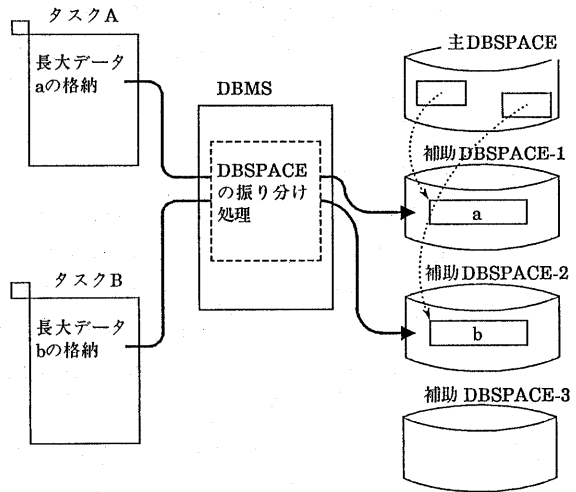


図4. 複数補助DBSPACEによる並列格納処理

(図4)

(2) リカバリ・データ量の削減

長大データ以外の更新では更新データ量が少ないためにBIやAIをデータベース更新時に取得したとしても問題ないが、長大データでは更新データ量が多くなるために問題が生じる。(2章)長大データに対するBIおよびAIの量を削減することを考える上で、まず、従来のリカバリ処理の概要を表2に示す。

表2. リカバリ処理の概要

障害種別	リカバリ処理
AP論理矛盾	BIによるデータベース復元 通常、DBMSのクイック・リカバリ機能を用いる
CPUダウン	BIによるデータベース復元
媒体障害	全ダンプ情報およびAIによるデータベース復元、 もしくはファイルの二重化によるデータベース復元

長大データに対するリカバリ処理においても、上記3つの障害に対処可能であることが必要である。この点を考慮した上で、長大データに対するA IおよびB Iの量を削減可能であるかを考える。

(i) A P論理矛盾とC P Uダウン対策

トランザクション内で更新前データを破壊しないような格納制御を行うと、長大データのB Iを取得する必要がない。すなわち、リカバリ・データの削減の観点から補助DBSPACEに対する格納制御として、『トランザクション内で更新前データの存在するブロックへの再格納を抑止する』制御が必要である。このためには、DBMSが各トランザクション中で再使用可能化したブロック群をトランザクション対応に管理する機能が新たに必要となる。この管理を行うために、再利用可能化したブロック群を示すスタック・エリアを主メモリ上に持つ。

(ii) 媒体障害対策

補助DBSPACEに対応する媒体が破壊されると、別媒体に退避した更新情報により復元する必要がある。従って、媒体障害対策を行うためには、リカバリ・データとして全ダンブ・A I方式かファイル二重化方式による対処が必須である。

長大データ更新時の処理時間を短くするためには、並列にデータの書き込み処理が可能であるファイル二重化方式による媒体障害対策をとるほうがよいといえる。

(3) 長大データに対する格納制御アルゴリズム

格納制御に必要な上記3つの項目、①『未使用ブロックへの優先的格納』、②『1補助DBSPACEに対する複数長大データの並列格納の抑止』と③『トランザクション内での更新前データの存在するブロックへのデータ再格納の抑止』を実現するための格納制御アルゴリズムを以下に示す。(図5)

- ① 長大データの分割格納の開始時点にすべてのトランザクションで使用されていない補助DBSPACEを探し②の処理に進む。未使用のDBSPACEがない場合は該トランザクションをウェイトさせる。
- ② 未使用ブロックの存在有無をFSPにより判定する。
 - ②-1) 未使用ブロックが存在するとき
FSPの指すブロックより必要ブロック数を割り当て、該当ブロックに対応するビットマップを使用状態に変更する。
 - ②-2) 未使用ブロックが存在しないとき
ビットマップおよびスタック・エリアにより、同一トランザクション内で再使用可能空間となったブロック以外の再利用可能ブロックを割り当て、該当ブロックに対応するビットマップを使用状態に変更する。
再使用可能ブロックの空間不足時は該トランザクションをエラーリタンさせる。

- ③ ②で割り当てられたブロック群に対してAPからマルチブロック一括アクセス機能を利用して、APから補助DBSPACEに長大データの部分データを直接格納する。
- ④ 主DBSPACE中の更新対象ブロックのBIを取得する。
- ⑤ DBMSのI/Oバッファ中の更新済ブロックをデータベースに書き出す。

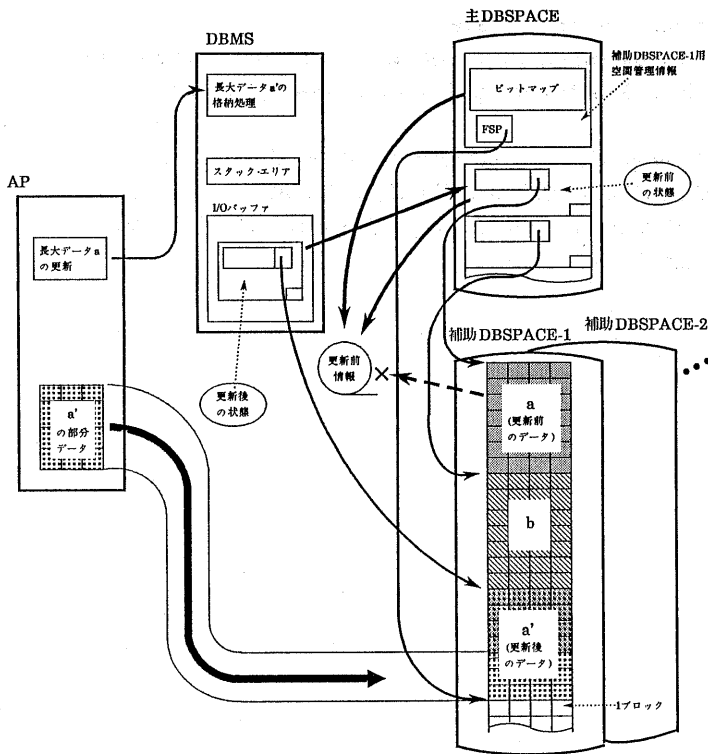


図5 長大データの格納制御の概要

- マルチブロック一括アクセスを行うための長大データの未使用ブロックへの優先的格納
- マルチブロック一括アクセスを行うための1つの補助DBSPACE内での長大データ並列格納の抑止
- 長大データ格納時のコンカレンシ向上のための複数補助DBSPACEによる振り分け処理
- リカバリデータ量の削減のためのトランザクション内の更新前データの存在するブロックへのデータ再格納の抑止

4. 終りに

DBMSにおける長大データの格納構造および格納制御方式について示した。すなわち、DBMSが管理する上での2つのポイント、①論理構造上は1つの格納空間と見せながら、格納構造として補助DBSPACEの導入による格納空間の分離（長大データのデータ実体とデータベース制御情報の分離）、②連続ブロックへの長大データの格納を可能とする補助DBSPACEの格納制御方式について示した。このような格納構造および格納制御方式を採用することにより、図形画像処理におけるデータ管理の一元化が専用システムの性能と比較しても遜色のない汎用DBMSが実現できる。

参考文献

- [1] Chang, S.K., Lin, B.S. and Waiser, R. : "A Generalized Zooming Technique for Pictorial Database," National Computer Conference, 1979.
- [2] Tong, G.Y. : "A Logical Data Organization for the integrated Database of Pictures and Alphanumerical Data," Proc. IEEE Workshop Picture Data Description and Management, 1980.
- [3] Lohman, G.H. and other : "Remotely-Sensed Geophysical Database : Experience and Implications for Generalized DBMS," SIGMOD, Vol 13, No 4, 1983.
- [4] Haskin, R.L. and Lorie, R.A. : "On extending the Function of a Relational Database System," proc. ACM SIGMOD, 1982.