

2J-8

データディクショナリ/ディレクトリを有する
構造型データベースシステムにおけるDBアクセスの一手法

田中 和明, 米田 茂 (日立製作所 システム開発研究所)
横山 一郎, 今城 哲二 (日立製作所 ソフトウェア工場)

1. はじめに

銀行の第3次オンライン、高度情報サービスなどの大規模DBへ適用するDB/DCシステムには、従来以上に操作性が良く、高性能であることが必要である。

我々は、この分野への適用を目的とするデータマネジメントシステムXDM(eXtensible Data Manager)について、既に報告した。[1],[2],[3]

XDMは、構造型DBに対するDB定義言語やデータ操作言語として、ISOで標準化作業が進められているANSIのNDL(Network Database Language)を基調としている。[4]

本稿では、NDLに基づく構造型DBにおいて、従来、ユーザプログラムの実行時に行なわれていたDBアクセス処理の一部を、コンパイル時に行なうことにより、高性能化を図るDBアクセス方式について説明する。

2. DD/Dを有する構造型DB管理システム

DD/D(Data Dictionary/Directory system)では、DB/DCシステムの定義情報が、登録・更新・参照に適した単位・形態でディクショナリに、DB管理プログラムが参照する単位・形態でディレクトリにおいて、整合性をとって管理される。DBに関する定義は、論理構造、仮想構造、格納構造、物理構造を規定する、それぞれ<schema>,<subschema>,<storage schema>,<physical schema>の4階層に分けてディクショナリで管理される。

3. ユーザインタフェース

ユーザプログラムからのDBアクセスは、操作性の向上を目的としてホスト言語仕様化された、データ操作言語(DML)に依る。

COBOL言語の場合を例にとると、ユーザプログラムで使用する<subschema>は、DATA DIVISION 中のSUBSCHEMA SECTIONで指定する。各DMLは、従来のREAD/WRITE文やMOVE/PERFORM文などと同様に記述する。

例えば、ある検索条件で<set view>内の<record view>を検索する場合、次のように記述する。

```
FIND FIRST <record view name>
      WITHIN <set view name>
      WHERE <condition>
```

ここで、<condition>は、<simple condition>を、AND, OR, NOT, カッコで結び付けて指定する。

(<simple condition> ::= <operend><relation><operand>)

<operand>には、<component view name>、プログラム内の作業領域名、定数を指定する。

4. 高速DBアクセス方式

従来の言語プロセサでは、DBアクセス要求に対して構文上のチェックのみ可能であった。それゆえ、DB管理プログラムはアクセス処理に入る前に、ユーザプログラムからの論理的なDBアクセス要求内容に対して、ディレクトリテーブル中のDB定義情報に基づき、要求内容を解析・診断する処理、実行時形式に加工する処理、および、格納構造や物理構造をも加味した内部のアクセス要求に展開する処理を行っていた。

提案方式は、DD/Dを有するDB管理システムにおいて、ディクショナリ内のDB定義情報と、それに基づき生成されるディレクトリテーブルとが、整合性をもって管理されていることに着目し、DB管理プログラムによる実行時処理の一部を、ユーザプログラムのコンパイル時に行なう方式である。

提案方式では、ディレクトリテーブルが生成される時期とコンパイル時期とに運用上、順序性を設けることは望ましくないため、ディクショナリ内のDB定義情報を参照することとした。

又、本方式によって、コンパイル時に参照したDB定義情報が変更されると、そのユーザプログラムは、リコンパイルを必要とする。ユーザプログラムは、論理的なDB定義、即ち、<schema>,<subschema>の定義内容に基づき作成される。それゆえ、これらの定義内容の変更によりユーザプログラムも変更あるいはリコンパイルが必要であり運用上許容範囲とした。

一方、<storage schema>や<physical schema>の定義

内容は、ユーザプログラムとは、直接関係しないことから事前処理に起因するリコンパイルは運用上問題であり参照しない方式とした。

ユーザプログラム実行までの過程と、実行時処理の概要を次に示す。(図1参照)

- (1) DD/DによりDB定義情報をディクショナリに登録する。その後、ディレクトリテーブルを生成する。
- (2) 言語プロセサはDMLの事前評価と実行時テーブルの生成を行なう。DMLの事前評価の例を次に示す。
 - (a) DMLとDB定義情報との整合性の診断
 - 例えば、3.の例では、次のような診断項目がある。
 - (i) FIND文中の<record view name>,<set view name>などが SUBSCHEMA SECTION で指定<subschema>で定義されているか。
 - (ii) <record view name>が示す<record type>が,<set view name>が示す<set type>の<member record>か。

- (b) DML中のビュー名称をディレクトリテーブルのエントリ番号に変換後、実行時テーブルに設定する。
 - 実行時にディレクトリテーブルのサーチが不要となる。
 - (c) 検索条件中の、定数を比較対象の<component view>の<data type>に合せて内部データ形式に変換する。
- (3) ユーザプログラムからのDBアクセス要求によって、該当DMLの実行時テーブルを伴い、DB管理プログラムに制御が渡る。DB管理プログラムは、該当<subschema>のディレクトリテーブルなどをloading後、DML実行時テーブルの内容に合せた、DBアクセス処理を実施する。
- このとき、ユーザプログラムのコンパイル時に参照した<subschema>と、loadingしたディレクトリテーブルを生成したときのそれとが同一であることを確認する。

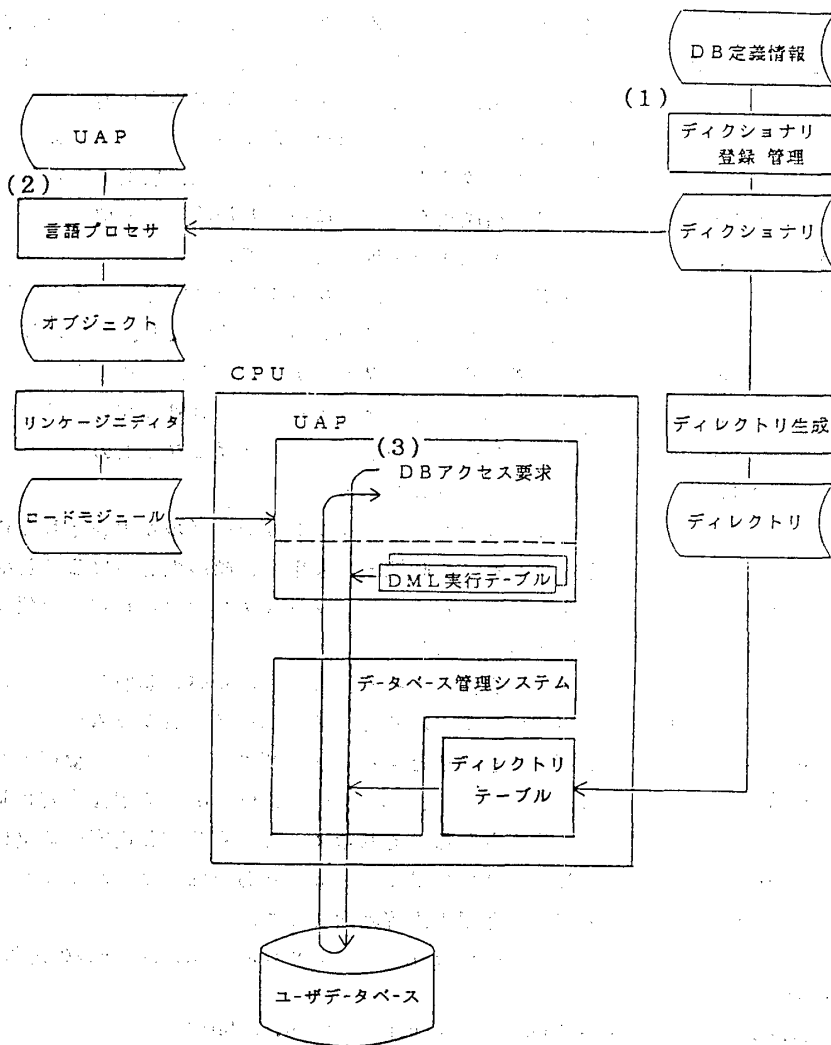


図1 DBアクセス処理

5. おわりに

本稿では、DD/Dを有する構造型DB管理プログラムにおいて、DBアクセス処理の一部をユーザプログラムのコンパイル時に行なうことにより、1 DML単位の実行時性能を向上させる方式を示した。

6. 参考文献

- [1] 斎藤、他：データマネジメントシステム XDM(1)-総論-、情報処理学会第32回全国大会,1986
- [2] 横山、他：XDM(2) -DB定義言語と操作言語-、情報処理学会第32回全国大会,1986
- [3] 和歌山、他：XDM(4)-データディクショナリ/ディレクトリ-、情報処理学会第32回全国大会,1986
- [4] ISO/TC97/SC21/WG3 and ANSI X3H2-86-25: "Database Language NDL",1986