

高速データベースマシンHDMにおける  
Dynamic SQLの実現

5Q-2

笑原辰夫 松橋正樹・ 中村俊一郎 石田喬也

三菱電機(株) 情報電子研究所

\*三菱東部コンピュータシステム(株)

1. はじめに

三菱電機 情報電子研究所で開発中の高速データベースマシンHDM(High speed Database Machine)[1-4]は、データベースのモデルとしてリレーショナル・データベースを採用しており、データベース操作言語としてはSQL[5]に基いた言語をサポートしている。SQLをアプリケーション・プログラムで用いるためのインターフェースとしては、埋め込み型SQL[5]やDynamic SQL[6]があるが、本報告では、Dynamic SQLのデータベースマシン上での実現方法について報告する。

2. HDMでのDynamic SQLの実現

HDMでは、Dynamic SQLおよび埋め込み型のSQLの両方をサポートする予定である。しかし、Dynamic SQLの方が、ホスト言語上の文字列型のデータとして、SQL文を用意し、実行できるため、データベースに対する処理をプログラム中で実行時に変更できるという特性を持っている。この特性は、実際にユーザ向けに作る非定型処理をするアプリケーション・プログラムとのインターフェースとして有効である。これに対し、埋め込み型SQLは、定型処理向きであるといえることができる。そのため、Dynamic SQLから先に実現することにした。

実現にあたっては、ホスト言語上に埋め込まれたSQL文を、モジュール言語に変換してから後、更にオブジェクトに落とすという方式を採った。そうしたのは、この方式がJISのSQLやSQL2[6]で規定されていることと、この方式の方が、複数のホスト言語をサポートするとき、モジュール言語からオブジェクトに落とす部分は共通に使えるので、作らなければならないツールの量が少なくなるという理由からである。また、実行時に必要となる航行マネージャやDynamic SQLコンパイラは、ライブラリとしてアプリケーション・プログラム内に含める形で実現した。

この埋め込み型SQLあるいはDynamic SQLは、SQLとSQL2に基いて仕様を決めている。

3. Dynamic SQLのプログラミング環境

Dynamic SQLが載るホスト言語としては、C言語を仮定している。図1に埋め込み型SQLも実現した場合のプログラミング環境を示した。プログラマは、C言語で記述されたアプリケーション・プログラムの中にSQLを埋め込むことができる。このソース・プログラムから、以下のような手順でロード・モジュールが自動的に作られてゆく。

- a) SQLプリ・プロセッサがSQLの部分関数呼び出しに変えたC言語から成るプログラムとSQLモジュール言語[5,6]に分離する。
- b) Cソース・プログラムの方は、Cコンパイラによってオブジェクトにする。

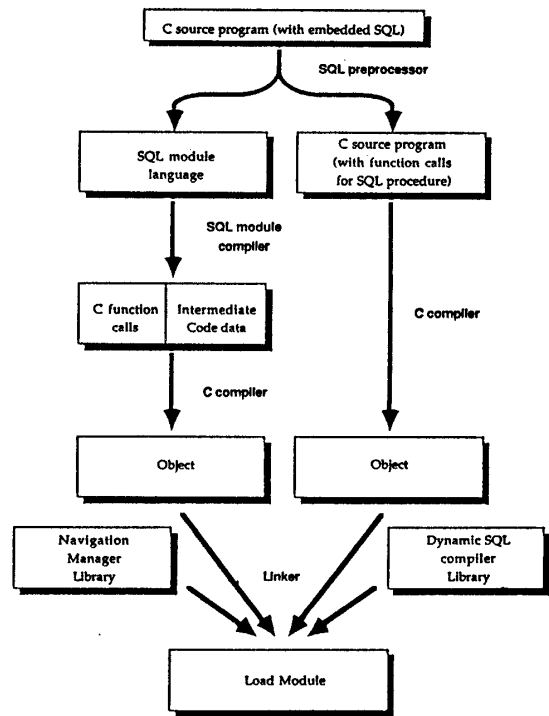


図1 Dynamic SQLのプログラム環境

An Implementation of Dynamic SQL on High Speed Database Machine, HDM

Tatsuo MINOHARA<sup>1</sup>, Masaki FUNAHASHI<sup>2</sup>, Shun-ichiro NAKAMURA<sup>1</sup>, Takaya ISHIDA<sup>1</sup>

<sup>1</sup> Mitsubishi Electric Corporation, <sup>2</sup> Mitsubishi Electric Computer System (Tokyo) Corporation

- c) SQLモジュール言語の方は、SQLモジュールコンパイラによって、Cの関数呼出しから成るC言語のソース・プログラムに直されてる。このときに、SQLはHDMにアクセスするための中間言語[4]に直され、C言語のソース・プログラム中にデータとして埋め込まれる。これを、Cコンパイラによってオブジェクトにする。
- d) 各オブジェクトは、リンカによって航行マネージャ・ライブラリおよびDynamic SQLコンパイラ・ライブラリと共にリンクされ、最終的なロード・モジュールが作られる。

この過程において、a)のプリ・プロセッサを経ずに、SQLの部分に関数呼び出しに変換したものとモジュール言語をプログラマが直接作成してもよい。

#### 4. Dynamic SQLの実行環境

実行時の動作環境を示したのが、図2である。実行時には、SQLとして動作する部分に関数として、アプリケーション・プログラムから呼び出される。このときPREPARE文によって準備されるSQL文(prepared statement)は実行時にコンパイルされる必要があるので、Dynamic SQLコンパイラが関数という形で実行時に呼び出され、SQL文が中間コードに変換される。

実際にHDMにアクセスするときは、EXECUTE, OPENなどの文に対応する関数が呼ばれたときである。これらの関数が呼ばれたとき、PREPAREで用意されていた中間コードが通信サーバ[4]を介してHDMに送られる。その実行結果のステータスとデータが航行マネージャに対して送られる。カーソルを用いるような操作の場合は、航行マネージャがこのデータから中間的なテーブルをホストマシン上にファイルという形で作り、以降のカーソル操作はそのテーブルに対して行なわれる。

最終的にアプリケーション・プログラムに実行の結果のステータスやデータが渡されるのは、関数呼出しのときに、変数へのポインタとして渡された変数に値が代入される形で行なわれる。

通信で用いられる中間コードは、以下に列挙するようなコマンドから成る処理のシーケンス列である。これは、現在HDMが実現している機能の範囲を表わしている。ただし、ここには記していないが、中間コードには、ダンプ・ロード用のコマンドも定義されている。

session	project	insert
lock	aggregate	delete
unlock	join	update
create table	drop table	
create index	drop index	

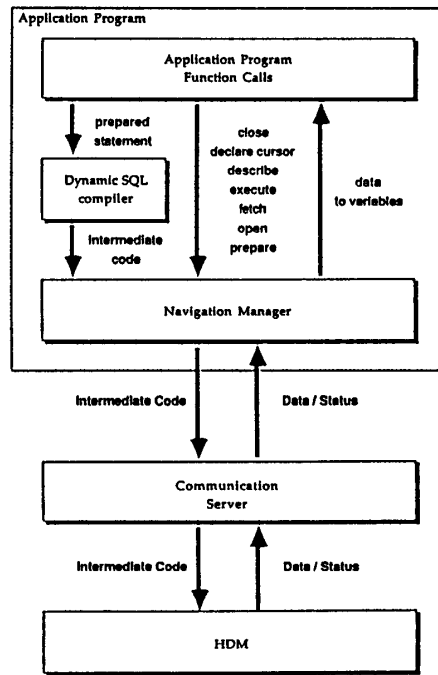


図2 Dynamic SQLの実行環境

#### 5. おわりに

本報告で述べたDynamic SQLでは、SQLプリ・プロセッサがなくてもよい。最小限のDynamic SQLの実現により、多くのアプリケーション・プログラムが開発できる。このDynamic SQL機能は、HDMをベースとした当研究所内の図書検索システムの開発において、その効果が実証されている。

#### 【参考文献】

- [1] Nakamura, S. et al., 「A High Speed Database Machine - HDM」, Proc. 5th Int. Workshop on Database Machines, pp.340-353, 1987.
- [2] 中村他, 「高速データベースマシンHDMのアーキテクチャ」, 情報処理学会第35回全国大会, 4 Cc-6, 1987.
- [3] 板倉他, 「高速データベースマシンHDMのソフトウェア」, 情報処理学会第35回全国大会, 4Cc-9, 1987.
- [4] 箕原他, 「高速データベースマシンHDMにおけるデータベースサーバ」, 情報処理学会第36回全国大会, 5E-8, 1988.
- [5] 「データベース言語SQL」, JIS, X3005-1987.
- [6] 「ISO Working Draft Database Language SQL2」, ISO, ISO/IEC JTC1/SC21/WG3 N580 and JTC1/SC21 N2663, March 1988.