

ORDBにおける多重定義関数の呼び出し処理の高速化

2 T - 3

岩田 守弘* 金子英司* 中野幸生* 八高克志**

*(株)日立製作所システム開発本部 **同ソフトウェア事業部

1. はじめに

オブジェクトリレーショナルデータベース(ORDB)では、種々のメディアデータをユーザ定義のデータ型によって表現し、それらに対する検索や格納などの操作を、ユーザ定義の関数を用いて提供する。

ORDBでは、同じ名称でパラメタの型や数が異なる関数の多重定義を許しており、関数呼び出しの型にあう関数を選択する必要がある。しかし、継承関係を持つユーザ定義のデータ型(UDT)をパラメタとする多重定義関数の場合、以下で説明するように、関数選択に対して、パラメタのUDTの組合せの増大がネックとなる可能性があり、効率的な選択方法が必要となる。

以下において、2. でユーザ定義型と多重定義関数について説明し、3. で例を用いて問題点を説明し、4. で提案方式について説明を行う。

2. ユーザ定義型と多重定義関数

UDTは、一般的に複数の属性と関数(もしくはメソッド)から構成される。属性には、数値型、文字型やUDTのデータ型を使用できる。

以下の説明では、通信手段、通信媒体などの複数の属性や、通信に関する関数で定義される通信機器型を仮定する。

UDTは継承などオブジェクト指向の機能を利用することができる。例えば、電話型、FAX型などを通信機器型から継承して作成することができる。

一方、関数は、引数の型や数が変われば同名の関数を幾つでも定義できる。これを多重定義と呼ぶ。以下では、多重定義関数の例としてFAX接続という2つの通信機器の間でFAX機能を利用して接続できるかどうか、真偽値を返す関数を使用する。

- (a) FAX接続(FAX型, FAX型)→真偽値型
- (b) FAX接続(FAX型, FAXモデム型)→真偽値型
- などは、常に真を返す関数。
- (c) FAX接続(FAX型, 電話型)→真偽値型
- (d) FAX接続(固定電話型, 移動電話型)→真偽値型
- などは、常に偽を返す関数

3. 多重定義関数の呼出し処理と問題点

例えば、図3のように、住所録表において、ある人とFAXで連絡とれる人を検索する関数呼び出し有りの問合せを考える。

電話型など継承した型のデータは、継承元(この場合通信機器型)のデータとして使用できる。そのため、検索文を実行してみないと、X, Yそれぞれの通信手段が何型のデータであるかわからない。

しかし、多大なデータに対して実行時にそれぞれ、型に適切な関数を選択していたのでは、検索時間が遅くなってしまいます。

実行時の負荷を軽くするには、コンパイル時に、予めパラメタの型と対応する関数定義の組合せを作成しておくことが考えられるが、組合せが多くなると、コンパイルに要する時間、および、オブジェクトの大きさが巨大なものとなってしまいます。

```
CREATE TYPE 住所型 {
    郵便番号 char(8)
    都道府県 nchar(4)
    市町村 char(20)
    ...
}
FUNCTION 郵便番号検索(住所型) ...
FUNCTION 周辺地図表示(住所型) ...
...
```

図 2.1 UDT の例

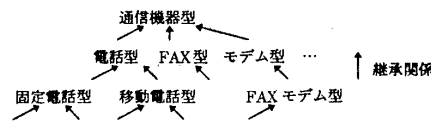


図 2.2 継承関係の例

住所録表	住所(住所型)	通信手段(通信機器型)	FAX接続 (X,Y)
田中	住所型データ 東京...	電話型データ 番号:03XXXX...	→
山田	住所型データ 神奈川...	FAX型データ 番号:04XXXX...	
鈴木	住所型データ 静岡...	モデム型データ 番号:05XXXX...	
...	

図 3 関数呼び出しの例

4. パラメタ型継承レベルソートによる多重定義関数の呼び出し処理の高速化

上記の問題点などを解決し効率良い多重定義関数の選択を行うために、コンパイル時、実行時の関数選択フェーズを図4.1のように分ける。

(1) コンパイル時

(a) 型適合候補関数選択フェーズ

関数呼び出しと同一名称でパラメタの数が一致する関数から実行時に呼び出される可能性がある関数を選択するフェーズ。

継承関係のあるUDT以外の引数に関して一意に候補を選択する。継承関係のあるUDTの引数に関しては、その引数のUDTと継承関係にある各UDTをパラメタとする関数を候補関数として残す。

(b) 継承レベルソートフェーズ

継承関係のあるUDTのパラメタに関して、継承のレベル(継承の深さ、例えば、通信機器型を1として、電話型やモデム型は2、移動電話型やFAXモデム型は3)で各候補関数をソート順に並べる。レベルが同じ場合は、型のID(名称など)でソートを行う。継承関係のあるUDTのパラメタが複数存在する場合には、あらかじめ決められた優先順位(一般的にはパラメタ記述順で左優先)でソートを行う。

(c) 基本絞り込みフェーズ

関数呼び出しの引数の型に適合するUDTよりもソート順で後ろにくるものは呼ばれる可能性が無いので、コンパイル時に候補から外す(図4.2)。

(c) 特殊絞り込みフェーズ

任意の2関数を比較し、呼び出しの引数のUDTに対し、パラメタがカバーする範囲を比較し、ソート順で前にくる関数が後ろにくる関数を完全にカバーする場合、後ろにくる関数は呼ばれる可能性が無いので、コンパイル時に候補から外す(図4.3)。

(2) 実行時

(d) ソート順関数選択フェーズ

実行時のデータの型に対して、呼び出せる候補関数をソート順にサーチし、最初に見つかるものを選択する(複数呼び出せる候補があっても、ソート順で前の関数の方が呼び出しの型より適合する候補である)。

5. まとめ

本稿では、多重定義関数の呼び出し処理を効率良く行う方式の提案を行った。本方式により以下の効果が見込まれる。

(1) 実行時候補関数のソート順データのみを持つ。パラメタの組合せ数に依存したデータを持たないため、パラメタの組合せの増大に対し、実行用のオブジェクトが肥大化しない。

(2) 実行時ソート順にサーチすることで実行時選択の負荷が少ない。

(3) コンパイル時、パラメタの組合せ数に依存しないため、パラメタの組合せの増大に対し、実行時間が増大しない。

(4) ソート順を利用した候補関数絞り込みにより、実行時まで候補として残る関数を効率良く削減することができる。

備考

本研究は通商産業省、情報処理振興事業協会(IPA)による「次世代電子図書館システム研究開発事業」の一部として実施されたものである。

参考文献

- [1] ISO/IEC FCD3 9075-1: "ISO Final Committee Draft(FCD) Database Language SQL", 1998

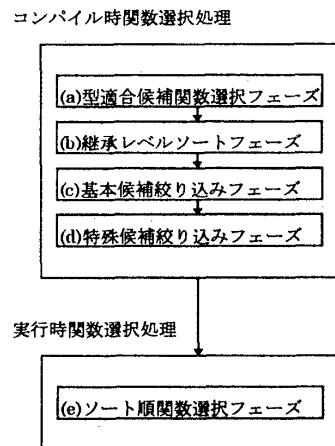
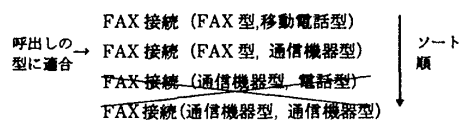


図4.1 多重定義関数の呼び出し処理

基本的な絞り込みパターン

呼び出し: FAX 接続 (FAX 型, 電話型)

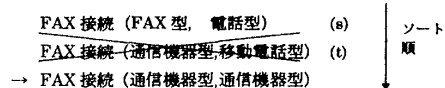


絶対呼ばれないのでコンパイル時に絞り込める

図4.2 基本絞り込みの例

特別な絞り込みパターン

呼び出し: FAX 接続 (FAX 型, 通信機器型)



呼び出しの引数に入る全てのデータに対して、(t)より(e)が優先される

図4.3 特殊絞り込みの例