

オブジェクト指向型管理情報ベース

5F-1

今井伊知郎 桐葉佳明
NEC C & C 研究所

1. はじめに

OSI管理により規定される管理情報の特徴として、オブジェクト指向を導入したモデル化と階層化、ASN.1を利用して定義される複雑なデータ構造の保持などが挙げられる。そのため、管理情報ベース(MIB)の実現方式として、複雑なデータ構造によるデータの蓄積/管理が可能であるオブジェクト指向データベース(OODB)は、非常に優れた資質を備えている。しかし、研究段階でのOODBは、処理速度の面で劣っていたため、MIBの実現方式については、現在に至るまで、リレーショナルデータベース(RDB)を利用した方式[1][2]や独自開発のDBMSを利用した方式[3][4]に関する研究が行われてきた。

現在、NM/forumのOMNIPoint等の動向に見られるように、オープン指向のネットワーク管理プラットフォーム構築の観点から本稿では、近年本格的な実用化が進んでいる汎用OODBのMIBへの実装方式を提案し、その評価結果について述べる。

2. 実装方式

OODBのMIBへの実装方式における管理情報の格納方式と管理情報間の参照関係の設計方針を述べ、それらに基づき検索処理の高速化を実現する2つスキーマモデルを提案する。

管理情報の格納方式：

OODBにおけるDBへのデータの格納は、C++のインスタンス毎に、仮想記憶空間上の複数のページから構成されるセグメントを指定することにより行われる。よって、MIBの高速化を実現するためには、データの読み出し効率を上げるために、同時にアクセスされる頻度の高いデータを同一セグメントに格納(クラスタリング)する必要がある。特に管理対象(MO)の名前を管理するディレクトリ情報は、ディレクトリ処理の高速化を実現するため、他の管理情報と分離して、ディレクトリ情報インスタンスとして、全て同一セグメントに格納すべきである。

管理情報間の参照関係：

属性インスタンスは、個々の属性値を保持すると共に属性に対する管理操作を提供する。MOインスタンスは、後述する2つのスキーマモデルに対応可能とするため、管理対象で定義された属性のインスタンスへのポインタを保持する。ディレクトリ情報インスタンスは、管理対象間の包含関係を表現するポインタと対応するMOインスタンスへのポインタを保持する。これにより、識別名からMOインスタンスを検索する名前読解処理や、包含木の部分木で範囲指定された複数の管理対象を検索するスコープ処理におけるデータアクセスの回数が削減され、ディレクトリ処理の高速化が図れる。

スキーマモデル：

ネットワーク管理における種々のアクセスパターンを考慮し、MOと属性に対する2つのスキーマモデルを提案する。

モデル1：管理対象の複数の属性に対する管理操作を高速に行うため、管理対象クラス毎に属性インスタンスとMOインスタンスを同一のセグメントに格納する。

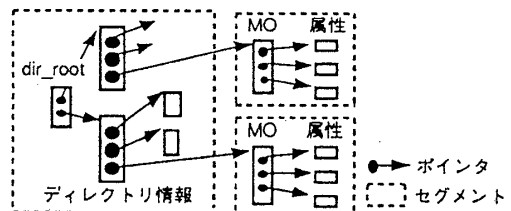


図1. モデル1

モデル2：単一属性の検索およびスコープ/フィルタ処理を高速に行うため、属性クラス毎に、属性インスタンスを同一のセグメントに格納し、管理対象クラス毎に、MOインスタンスを同一セグメントに格納する。

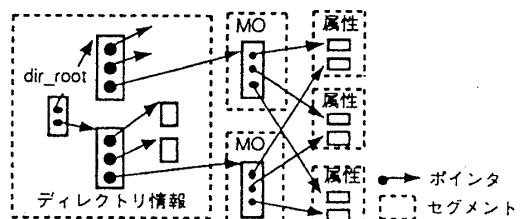


図2. モデル2

3. 性能評価

以下に示す測定環境、評価モデルで、それぞれ、2章のモデル1、モデル2のMIBを作成し、以下の測定項目に関するディレクトリ処理、属性操作、DBのコミット処理の総和時間を報告する。

測定環境：CPU=RISC(50MHz)、DB SIZE=47MB

DBキャッシュサイズ=8MB

評価モデル：性能評価を行うために、多重化装置により構成される伝送ネットワークを想定した図3の包含木に従って、MIBを作成した。図中の括弧の中の値は、包含関係において、親インスタンス1個に対する子インスタンスの数である。

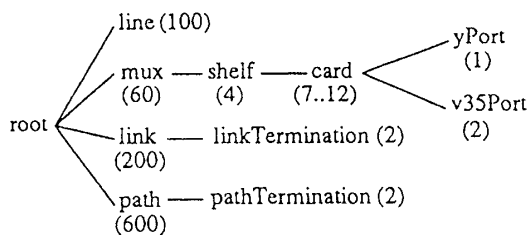


図3. 評価モデル

測定項目/測定結果：

・単一MOに対するGET操作（単一/複数属性）

ランダムに選択した管理対象に対して(1)モデル1の単一属性、(2)モデル2の単一属性、(3)モデル1の複数属性、(4)モデル2の複数属性のGETの操作を行う。ここで、複数属性は、全ての管理対象クラスに共通する4つの属性に対して操作を行う。以上の操作を3000回実行し、その平均処理時間を図4に示す。

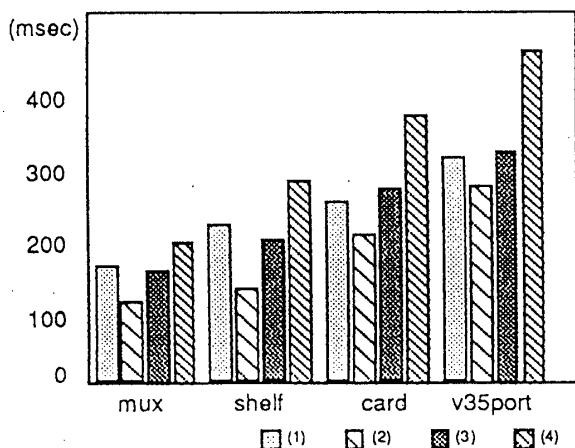


図4 単一MOに対するGET操作（単一/複数属性）

上記図4から解かるように、包含木の上位MOが低コストで、下位MOほどコストが増大することがわかる。これは、下位MOのディレクトリ情報を検索する際に、まずその上位MOのディレクトリ情報を読み出すため、上位MOのディレクトリ情報ほど

DBのキャッシュに存在する確率が高いためである。また、モデル1 [(1),(3)]が操作属性数とは独立に処理時間がほぼ一定であるのに対して、モデル2の(4)は、操作属性数に比例して読み出しデータ量が増加するため、DBキャッシュのヒット率が低下し、処理時間が著しく増大している。

・複数MOに対するGET操作（単一属性）

基点をmuxクラスからランダムに選択し、基点からNレベル下位までの範囲の管理対象の単一属性に対するGET操作を行う。以上の操作を300回実行し、その平均処理時間を図5に示す。

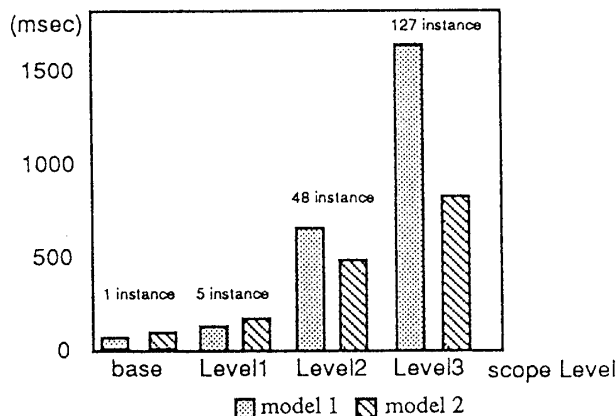


図5. 複数MOに対するGET操作（単一属性）

上記図5から解かるように、モデル1とモデル2を比較すると、スコープレベル2で処理時間が逆転している。これは、モデル1が操作対象以外の属性インスタンスも読み出すため、DBキャッシュの内容が頻繁に書き変わり、キャッシュヒット率が低下するためである。

4. おわりに

OODBのMIBへの実装方式を提案し、提案したモデル1、モデル2のMIBに対する性能評価を行った。この評価結果等からモデル1は、単一MOに対する管理操作の頻度が高いシステムに適しており、モデル2は、スコープ/フィルタ処理の高速化が要求されるシステムに適していることが判明した。

参考文献：

- [1] 桐葉他 "管理情報ベース(MIB)の開発支援環境"情処SIG-DB86-9,1991.11
- [2] 依田他 "伝送網オペレーションにおける管理情報ベース(MIB)の構成法"信学論文誌,VOL.J75-B-I NO.8,pp517-527
- [3] 宮内他 "OSI管理における管理情報ベース(MIB)の構成法"信学論文誌VOL.J74-B-I,NO.11,pp971-982,1991.11
- [4] 西山他 "高速OSIディレクトリ用DBMSの評価"情処全大 3N-3,平5前期