

通信システムのための ASN.1 データベースの評価

6 G - 4

小野 智弘

西山 智

堀内 浩規

小花 貞夫

国際電信電話株式会社 研究所

1. はじめに

抽象構文記法 1 (ASN.1) [1] は、OSI 応用層で扱うプロトコルやデータ要素の情報をコンピュータや端末の機種に依存することなく交換するための、「データ型の記法」と「符号化規則」を定めている。ディレクトリやネットワーク管理等の通信システムでは、ASN.1 で定義されたデータ型を持つ情報 (ASN.1 データ) をデータベースに格納する必要がある。筆者らは、これらのシステムの開発を容易にするために、汎用的に利用可能な、ASN.1 データを効率的に扱えるデータベース (ASN.1 データベースと呼ぶ) を実装した [2][3]。本稿では、実装した ASN.1 データベースの評価を述べる。

2. ASN.1 データベースの概要

ASN.1 データベースは、ASN.1 で定義された任意のデータ型を持つ情報を扱うデータベースであり、以下の特徴を持つ。

[特徴 1] ASN.1 データの符号化/復号処理にユニバーサル ASN.1 トランスレータ [4] を用いることにより、動的に型定義が変更が可能で、かつ高速な符号化/復号処理を実現

[特徴 2] データを通信路上で転送される符号化列のまま格納することにより、高速なデータ操作を実現
また、高速化のために以下の機能を有する。

- (1) ハッシュ関数を用いたインデックス機構
データ量が大きくなる SEQUENCE 等の ASN.1 構造形のデータについて、符号化列のハッシュ値をインデックス値として保持し、条件比較に使用する
- (2) 間接インデックスの利用
条件で指定された型に直接インデックスが存在しない場合でも、対象となるデータ型が構造形でその構成要素のいずれかの型にインデックスが存在する場合、それを間接的に利用する
- (3) 同一の検索操作に対するキャッシング
同一のパラメータで複数回実行される検索操作に対し、操作の解析コストを削減するため、操作の解析結果をキャッシングし、次回に利用する

ASN.1 データベースは C 言語で実装されており、図 1 に示すように、応用プロセスからデータ操作 (検索/格納/更新/削除) でアクセスされる。また、ソフトウェアは、ASN.1 データの符号化/復号を行なうユニバーサル ASN.1 トランスレータ [4] と、そのトランスレータを利

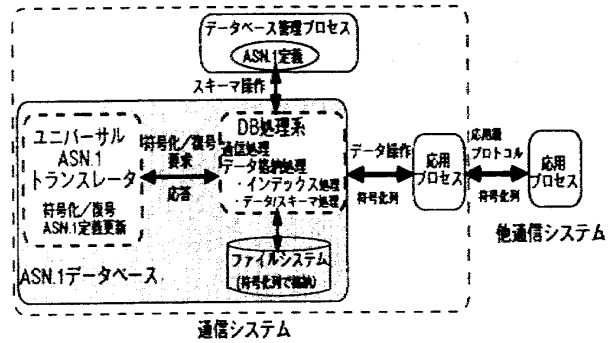


図 1: ASN.1 データベースのソフトウェア構成

用してデータベース機能を実現する DB 処理系からなる。DB 処理系は、プロトコル処理等を行なう「通信処理」部分と、ファイルシステムを利用しながらデータ/スキーマ管理やインデックス管理を行なう「データ格納処理」部分からなる。

3. ASN.1 データベースの評価と考察

図 2 のデータ構造を用いてデータ件数 (PI のインスタンス数) を変化させ、データベースの基本性能、および高速化機能の効果を SUN Ultra 1 (Solalis 2.5) 上で評価した。

```

PI ::= SEQUENCE{
    name    OCTET STRING, -- 'ONO'
    number  INTEGER, -- 各データに一意な値
    group   INTEGER, -- 1~10:各値の存在確率は等しい
    company OCTET STRING, -- 'KDD'
    subinfo OCTET STRING -- 'KDD'
}
    
```

図 2: 評価で使用したデータ構造

3.1 基本性能の評価

整数型の number にインデックスを付与し、10 件~1 万件のデータから任意の 1 件を検索・挿入した際の応答時間を図 3、図 4 に示す。応答時間の内訳は、各操作とも、通信処理、データ格納処理、ファイルシステム処理、の各部分からなる。

通信処理部分とデータ格納処理部分の応答時間は、検索操作、挿入操作のいずれの場合も、データ件数にかかわらずほぼ一定であった。一方、ファイルシステム処理部分の応答時間は、データ件数 n に対し、概ね log n に比例した。特にデータ挿入操作では書き込み処理に伴うディスクアクセスが生じるため、データ件数の影響を多く受けた。

また、データ検索操作では、通信処理部分とデータ格納処理部分のうちの約 84% が ASN.1 符号化/復号処理

Evaluation of ASN.1 Database for OSI Data Storing Applications
Chihiro ONO, Satoshi NISHIYAMA, Hiroki HORIUCHI and Sadao OBANA, KDD

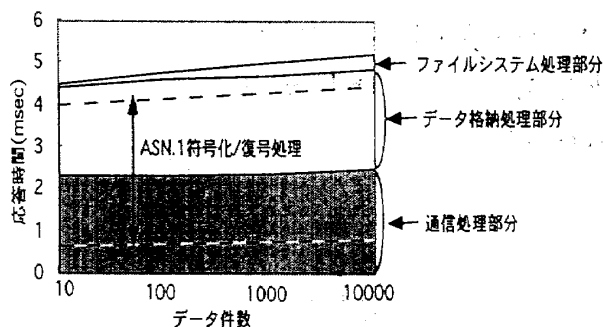


図 3: 検索操作の性能

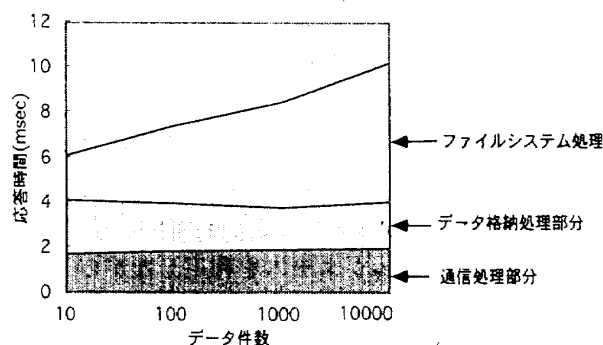


図 4: 挿入操作の性能

であった。データを格納する応用では、従来の ASN.1 インタプリタを用いる場合は、ユニバーサル ASN.1 トランスレータを用いる場合に比べて ASN.1 符号化/復号部分が 5 倍程度遅くなる^[4]。このことから、従来のインタプリタを用いたデータベースの応答時間は ASN.1 データベースに比べて約 4.5 ($\approx 0.84 \times 5 + 0.16$) 倍となると推定できる。

3.2 高速化機能の評価

構造形の値全体を検索条件とした際の応答時間の変化を図 5 に示す。ここでは、PI の値全体を検索条件とし、(a)PI にインデックスを付与した場合、(b)group にインデックスを付与した場合、(c)インデックスがない場合の応答時間をそれぞれ、10~1 万件のデータに対して測定した。

(1) ハッシュ関数の利用の効果

PI 全体に対する検索 (a) では、インデックスはハッシュ値を用いている。ここでは 1 バイトの簡易なハッシュ関数を使用しており、応答時間は、データが 2^8 件位までは $O(\log n)$ で推移し、キー値の短縮によるディスク領域の削減効果を受けた。データが 2^8 件を超えると、ハッシュ値に重複が生じるため、応答時間が $O(n \log n)$ で推移する。1 万件のデータに対するハッシュ値での比較を行なった場合では、ハッシュ値に 40 件 (0.4%) 程度の重複が生じ、この候補から絞り込む処理に 4msec 程度要した。CRC(巡回冗長検査)16 アルゴリズムのようなハッシュ範囲の広い関数を用いて重複を防ぐことによ

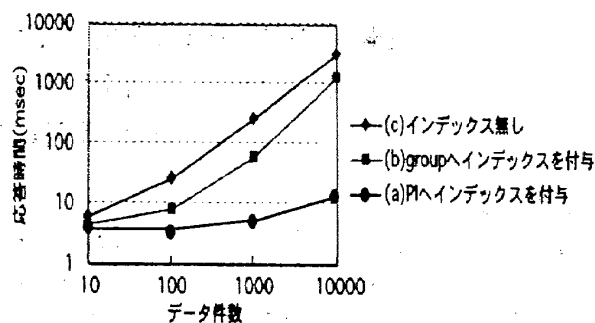


図 5: PI に対する条件を設定した検索操作の性能

り、応答時間を $O(\log n)$ までに短縮できる。

(2) 間接インデックスの効果

PI に対する直接インデックスが存在せず、group に対するインデックスが存在する場合 (b)、ASN.1 データベースは group に対するインデックスを利用して検索対象を絞り込み (group の 1~10 の各値の存在確率は等しいため、10%まで絞り込まれる)、絞り込まれたデータに対して PI 全体との一致を検査する。データが 1 万件の場合、(b) の応答時間は 1,365msec であり、(c) のインデックスなしの場合の 3,239msec と比べて約 42% に削減できた。この応答時間が絞り込んだ割合 (10%) より大きくなったのは、(c) の場合が全てのデータを順に B^+ 木から取得して検査するのに対し、(b) の場合では絞り込んだデータが B^+ 木上で連続しているとは限らないため、逐一 B^+ 木の根から検索することが主要な理由である。このオーバーヘッドを考慮すると、約 23.8 ($\approx 10/42$)% 以下までデータ候補を絞り込むことができる場合に、間接インデックスを用いることが有効となる。

(3) キャッシングの効果

同一パラメータの検索操作に対するキャッシング機能を (a) の場合に使用した。この結果、全てのデータ件数において応答時間が 0.5~0.8msec 程度短縮できた。これは、応答時間の 10% 強の削減に相当する。

4. おわりに

本稿では、開発した ASN.1 データベースを用いて検索・挿入操作の基本性能を評価し、高速化のためのハッシュを用いたインデックス機構、間接インデックス、キャッシングの効果を示した。最後に日頃御指導頂く国際電信電話(株)研究所 村上仁己取締役 所長 および 鈴木健二 副所長 に感謝します。

参考文献

- [1] ITU-T 勧告 X680 シリーズ "ASN.1" (1992)
- [2] 西山 他: "ASN.1 データベースの実現方式に関する一考察", 第 49 回情処全大 4W-11, (1994)
- [3] 小野 他: "ASN.1 データベースのためのデータ格納方式", 第 55 回情処全大 5R-04, (1997)
- [4] C. Ono et al: "Universal Translator for Data Communication and Data Storage", Proc. of 13th International Conference on Computer Communication (ICCC) (1997)