

分散処理対応 OSI ディレクトリシステムエージェント (DSA) の実装と評価

西山 智 横田 英俊 小花 貞夫 鈴木 健二

国際電信電話株式会社 研究所

本稿では、シャドウを含む OSI ディレクトリの全ての分散処理機能を提供する OSI ディレクトリエージェント (DSA) の実装と性能評価について報告する。既に開発済みのディレクトリ情報ベース (DIB) 専用 DBMS である ASSIST/D のソフトウェアをベースとしてモジュールの置換により実装し、また、ハッシュによる高速な名前解析処理方式を採用して高速化を図った。シャドウの実現では、シャドウ元がシャドウ範囲の境界を下位参照等の分散処理知識に変換し転送するシャドウ情報に加えることで、シャドウ先でシャドウされた部分木がローカルな部分木と区別なく名前解析処理できるようにした。評価の結果、UNIX ワークステーション上で 18,000 件のエンTRIES を格納した DSA で TCP/IP を下位プロトコルとして用いた場合、ローカルな Read 操作が 12.5 ミリ秒、他のワークステーション上の DSA への Read 操作の中継が 12 ミリ秒で実行でき、この DSA が UPT 等の高度交換通信サービスの呼処理用ネームサーバとしても適用できることを示唆している。

Implementation of OSI Directory System Agent (DSA) Supporting Distributed Operations and Its Evaluation

Satoshi NISHIYAMA Hidetoshi YOKOTA
Sadao OBANA Kenji SUZUKI

KDD R & D Laboratories
2-1-15, Ohara, Kamifukuoka-shi, Saitama 356, JAPAN

This paper presents the implementation and the performance of an OSI Directory System Agent (DSA) which supports all kinds of distributed operations of OSI Directory including shadow operations. We use ASSIST/D: a dedicated DBMS for Directory Information Base (DIB), which has been designed using the extensible DBMS techniques, as the database function needed for storing entries, and implement the DSA software on top of ASSIST/D. We also apply the hash-based name resolution method, which we have proposed as the name resolution method for the DSA software. When a portion of DIB is shadowed to another DSA, the shadow supplier DSA adds the knowledge (e.g. subordinate references) for the boundary of the portion to the shadow information, so that the shadow consumer DSA does not need to distinguish the shadowed portion from the locally held portions on the name resolution phase. The evaluation result on UNIX workstations connected by LAN shows that a DSA which stores about 18,000 entries can execute a Read operation for a locally held entry and chain a Read operation to another DSA in a different workstation within 12.5 milliseconds and 12 milliseconds, respectively.

1. はじめに

OSIディレクトリ^[1]は、これまでOSI通信システムや通信利用者にアドレス等のディレクトリ情報を提供することを主目的としたネームサーバとして標準化が進められてきた。近年、OSIディレクトリをUPT(ユニバーサルパーソナル通信)等の高度交換通信サービスでの呼処理に必要な通信情報を提供するためのネームサーバとして適用することが検討されている^[2]。このような適用例では、呼処理によって発生する大量の検索等の操作要求を処理するために、高速なディレクトリシステムエージェント(DSA)にディレクトリ情報ベース(DIB)を分散格納し、高い処理能力を持つOSIディレクトリを実現することが重要となる。筆者らはこれまでに、OSIディレクトリの高速度の観点から、ディレクトリ情報ベース(DIB)専用DBMS(データベース管理システム): ASSIST/D^[3]を開発し、またハッシュによる高速名前解析処理方式^[4, 5]を提案してきた。そこで今回、高速かつ高い処理能力を持つOSIディレクトリを実現するために、ASSIST/Dをベースに分散処理機能を持つDSAの実装を行なったので、本稿ではその実装概要と性能評価について報告する。

2. ASSIST/Dの概要とOSIディレクトリの分散処理

2.1 ASSIST/D

ASSIST/D^[3]は筆者らが開発したDIB専用DBMSで、以下の特徴を持つ。

- DIBのデータモデルとRead, Listなどのディレクトリ操作(以下単に操作と呼ぶ)をDBMSのデータモデル並びに操作として提供する。ASSIST/Dは分散処理を提供しないDSAとして使用できる。
- DIBを高速に格納、検索するための専用アクセス手法やインデックス機構を持つ。
- 拡張可能DBMS構築技法に基づき、9階層11モジュールのソフトウェア構成をとる。モジュールの交換により、内部での格納方式やトランザクション等の処理方式の変更が行なえるのみならず、他の応用(例えばOSI管理情報ベース(MIB))のための専用DBMSを実現できる^[6]。
- 1つのエントリを処理する毎に実行する操作を切替えることで、操作要求の多重処理を1プロセスで実現している。

2.2 OSIディレクトリの分散処理

OSIディレクトリの標準^[1]では、DSAの分散処理機能を以下のように規定している。

• 分散処理の形態

DSAが操作で要求されたエントリをローカルに格納しない場合、DSA間の分散処理の形態として、チェーン、マルチキャスト、リフェラルの3種類がある。チェーン(図1(a))では、操作要求を実行できる(或は実行できるDSAを知っている可能性のある)特定の1つのDSAに操作を転送し、転送先DSAから得られる結果を要求元に返送する。マルチキャスト(図1(b))では、最適な転送先DSAが特定でき

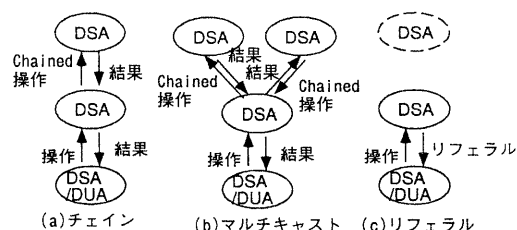


図1: OSIディレクトリの分散処理の形態

ない場合に、同一の操作要求を実行できる可能性のある複数のDSAに同時に転送する。この場合、いずれかのDSAから得られる結果を要求元に返送する。リフェラル(図1(c))では、DSAが自らは操作転送を行わず、操作要求元に操作実行できるであろうDSAの情報(アドレス等)を返送する。

• 分散処理知識

分散処理を行なうために、DSAは表1に示す知識(分散処理知識)を持つ。

表1: DSAが保持する分散処理知識

内部参照 (INTR)	自DSAに格納するエントリへのポインタ
上位参照 (SUPR)	自DSAに持つ他の知識では処理できない場合の転送先DSAに関する情報。DSA内に1つのみ存在
下位参照 (SUBR)	自DSA内に格納されているエントリの特定の直接下位エントリを格納するDSAに関する情報
不特定下位参照 (NSSR)	自DSA内に格納されているエントリの不特定の直接下位エントリを格納するDSAに関する情報
クロス参照 (CR)	自DSAが格納するエントリとは直接上位・下位関係にないエントリを格納するDSAに関する情報

• 操作分割と結果合成

複数エントリに対する操作(List, Search)では操作対象となるエントリが複数のDSAにまたがって存在する場合がある。この場合、操作を要求されたDSAは、要求された操作を分割して、操作範囲のエントリを格納する各々のDSAに分割した操作を転送する(操作分割)。これらのDSAから得られた結果と、ローカルなエントリに対する操作結果を合成して要求元に返送する(結果合成)。

• シャドウ

DSAは他のDSAが格納するエントリのコピーを持つことができる(シャドウ)。そのエントリに更新が発生した場合は、まずオリジナルが更新され、さらにオリジナルを持つDSAが更新内容を更新時或は定期的にコピーを持つDSAに通知することでコピーへの更新の反映がなされる。シャドウにより、コピーを持つDSAでもそのエントリに対する検索操作を実行でき処理の負荷分散を実現できる。

3. DSA実装の基本方針

- ASSIST/Dをベースに実装する。

表 2: 機能仕様

ディレクトリ操作	DSA/DUA 間	DAP 知識更新操作*1 シャドウ制御操作*1
	DSA/DSA 間	DSP DOP DISP
分散処理	分散処理形態の全て (チェーン, マルチキャスト, リフェラル), シャドウ, 操作分割・結果合成	
分散処理知識	全ての知識タイプ (上位参照, 下位参照, 不特定下位参照, クロス参照) を使用可能	
認証	DSA/DUA 間	パスワードによる弱認証
	DSA/DSA 間	認証なし
通信方式	下位プロトコル	TCP/IP, OSI スタック*2
	符号化方式	ASN.1 基本符号化規則

DAP: ディレクトリアクセスプロトコル

DSP: ディレクトリシステムプロトコル

DOP: ディレクトリ操作結合プロトコル (DSA 間のシャドウ制御用)

DISP: ディレクトリ情報シャドウプロトコル

*1 DSA 管理用の特権操作。

*2 現在実装中

- 高速化を図るために, 名前解析方式としてハッシュを用いた高速名前解析処理方式^[4, 5]を採用する。
- シャドウを含む OSI ディレクトリの全ての分散処理を実現することとし, 表 2 に示す機能仕様とする。なお, DSA では分散処理知識の保守やシャドウの開始・終了等の制御のためにローカルな DIB 管理機能も必要となる。これらについても, DSA の管理者が DUA を介して行う形態とし, DSA/DUA 間に対応する操作を設けることとする。

4. ソフトウェア構成

ASSIST/D の最上位層の DSA モジュールを分散処理対応のモジュールに置換して実現する。ハッシュによる高速名前解析処理方式は DIT 情報 (エン트리間の上下関係) を参照しない^[4, 5]。DIT 情報は複数エントリに対する List, Search 操作等では必要であるが参照頻度が少ないため, 格納効率を考慮して DIT 情報も識別名のハッシュ値をキーとして B-木に格納することとし, アクセス手法から木構造モジュールを削除した。また, この高速名前解析処理方式では操作対象となるエントリを確定するために, この DIT 情報を検索した後に, エントリに含まれる識別名を確認する必要があるが, ASSIST/D では, 最上位層を除く全ての下位層は格納符号化方式に依存しない構造であるため識別名を確認できない。従って, 従来最上位層の直下のインデックスレイヤで実施していた名前解析処理を DSA モジュールで実行することとし, インデックスレイヤを使用していない。さらに, シャドウ起動時の大量の更新処理に対応可能とするため, トランザクション機能を実現するトランザクションレイヤを, 主記憶上で更新後のイメージを保持する AI モジュールから, 更新前イメージファイルを使用する BIF モジュールに置換した。最上位層を除く下位レイヤのモジュール

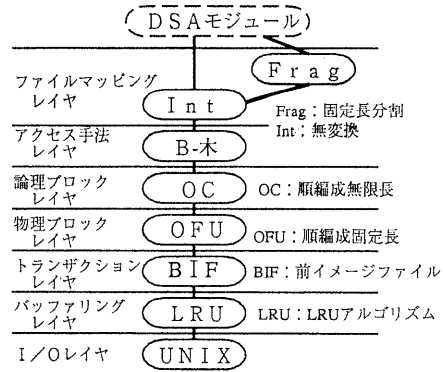


図 2: 下位レイヤのモジュール構成

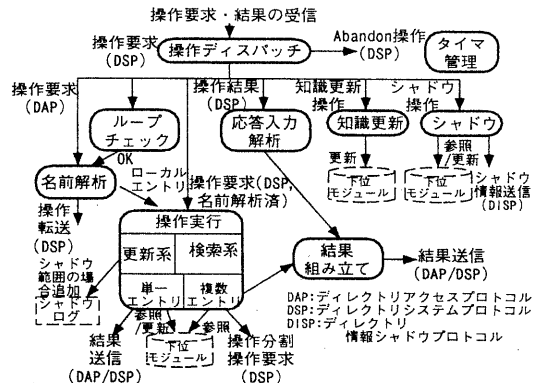


図 3: DSA モジュールのソフトウェア構成

構成を図 2 に示す。また新たに作成した DSA モジュールの構成を図 3 に示す。次節では, DSA モジュールでの分散処理機能の実現方式の詳細について述べる。

5. 分散処理機能の実現方式

5.1 分散操作の管理

各々のディレクトリ操作は図 4 に示す操作管理テーブルで表現される。操作要求が発生すると対応する操作管理テーブルが作成され, 操作完了時点で廃棄される。チェーン, マルチキャストや操作分割が発生した場合, 他の DSA に送出した操作情報とそれに対する結果もこのテーブルにつながって管理される。操作実行のために実行キューと実行待ちキューの 2 つのキューがあり, 操作管理テーブルはいずれかのキューに存在する (図 4)。

5.2 操作分割, 結果合成

List, Search 操作での検索範囲を決定するために, エントリ間の上下関係に関する情報 (DIT 情報) (図 5) を, エントリを識別するオブジェクト識別子 (OID) のルートからの並びをキーとして B-木に保持する。DIT 情報には, 通常のエントリのみならず別名や分散処理知識 (下位参照或は不特定下位参照) も併せて持たせ, OID の最上位 2bit を用いてエントリ, 別名, 知識の区別を行なう。

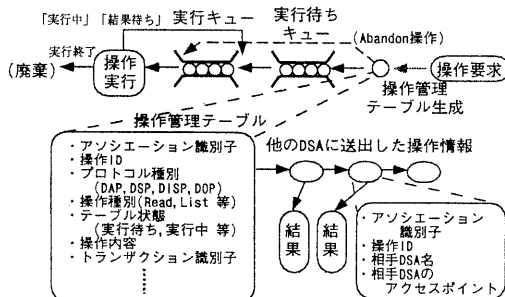


図 4: 操作管理テーブルの構造と実行の流れ

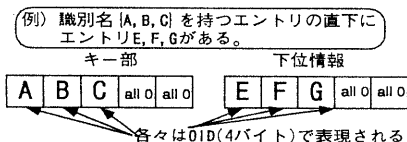


図 5: DIT 情報の表現

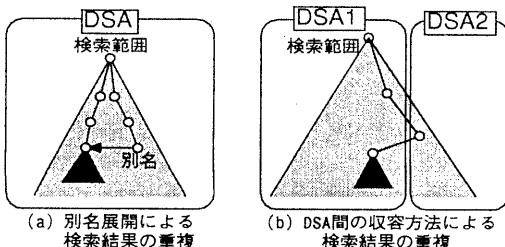


図 6: 検索結果の重複

List 又は Search 操作では検索範囲の頂点エントリの識別名に対する OID の並びをキーとして DIT 情報の一致 (直下のエントリを得る) 又は部分一致 (下位の全エントリを得る) 検索を行ない、検索結果を得る。得られた結果がエントリの場合は通常のフィルタ処理を行ない、別名では別名が示す識別名を基に再度名前解析を行なう。結果が分散処理知識であれば要求された操作の nameResolutionPhase パラメタを completed に変更した操作を、知識の示す DSA に転送する。DIT 情報の検索では DIT を 1 段毎に辿るわけではないため、別名が展開された場合 (図 6(a)) と、部分木の一部分が他の DSA に出て再度戻ってきている場合 (図 6(b)) には元の操作の検索結果と、分割した操作の結果が重複する。このため結果合成では重複したエントリを除く処理も行なう。

5.3 シャドウ機能

標準ではシャドウのモデルとプロトコルのみが規定されており、具体的な実装方法は規定していない。そこで、シャドウ機能を実現するために以下の実装方式とする。

- ・シャドウ情報: 以下の項目をシャドウの対象となる情報とする。

(1) シャドウ範囲のエントリ

(2) シャドウ範囲の分散処理知識

(3) シャドウ範囲の頂点エントリ名からなるコンテキストプレフィックス (DIB 部分木の頂点エントリの識別名)

(4) シャドウ範囲直下のエントリへの下位参照

上記 (3)(4) は本来シャドウ元には存在しない情報であるが、これらの情報を対応するエントリから新たに作成しシャドウ情報として送ることにより、シャドウ先ではシャドウ範囲を必要な分散処理知識が完全に揃った部分木として保持でき、通常の部分木と同様に名前解析処理できるという利点がある。

- ・シャドウ情報の格納: シャドウ先ではシャドウ情報も通常の情報と同様にエントリ或は知識として格納する。なお格納時には、情報がコピーであることを示すフラグとシャドウ元 DSA の情報を追加してローカルなものとは区別する。
- ・シャドウ情報の利用: シャドウ先の DSA は通常の名前解析処理ではオリジナルであるかシャドウ情報であるか意識する必要がない。コピーが使用できない操作に対して名前解析処理の結果がシャドウ情報によるコピーであった場合、コピー中のシャドウ元 DSA の情報を一種のクロス参照とみなして、直接シャドウ元にチェーンする。
- ・シャドウログ: シャドウをアソシエーションの切断や DSA プロセスの異常終了を超えて有効とするため、シャドウが開始されるとシャドウ元ではシャドウ範囲内の DIB の更新内容をシャドウログと呼ばれる外部ファイルを設けて管理する。シャドウログには更新内容を更新操作の種別に応じて AddEntryArgument, ModifyEntryArgument 等の形式で格納する。またシャドウ範囲内の知識に対する更新もシャドウログに記録する。シャドウログ内容はシャドウ更新操作によりシャドウ先に内容が反映された時点で消去する。

6. 性能評価

性能評価では表 3 に示した測定条件のもとで、図 7 に示す DIB を使用し、DSA でのローカルエントリに対する操作を測定した。また、同様の条件で分散処理性能として DSA での中継操作の処理時間とシャドウによる操作応答時間を測定した。

6.1 ローカル操作

DSA に格納するエントリ件数をパラメータとして、計算機 A 上の DUA から同一計算機 (計算機 A) 上の DSA1 に対して DSA1 がローカルに格納するエントリへの Read 操作を行ない、その操作応答時間を測定した。結果を図 8 に示す。また、図 8 には UNIX の gprof によって取得した DSA 内での主な内部処理が占める割合も併せて示した。図 8 において操作実行時間がグラフに現れないのは、本実装で使用した名前解析処理方式では名前解析時にエントリ自身を読み込む等、ほとんどの Read

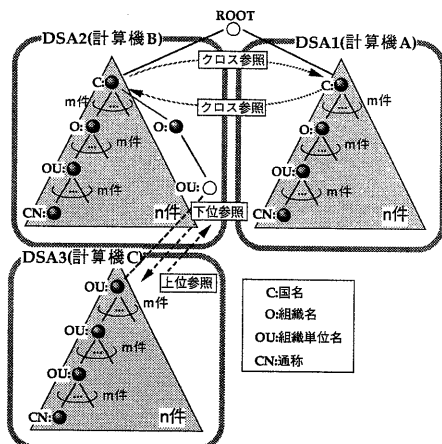


図 7: 測定に用いた DIB

操作に必要な処理を行うため、実質的な操作実行の処理がきわめて小さいことによる。

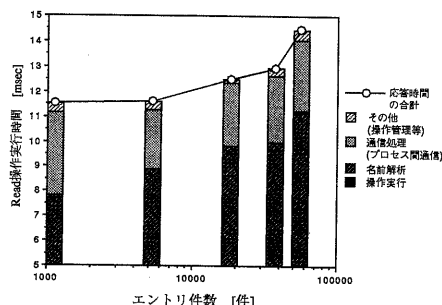


図 8: ローカルな Read 操作応答時間

6.2 分散処理性能

6.2.1 中継操作

図9にDSAに格納するエントリ件数をパラメータとして、計算機A上のDSA1でのRead操作の中継に要する時間を示す。中継時間の測定では計算機A上のDUAから計算機AのDSA1を経由して計算機BのDSA2で操作実行が行なわれた場合の操作応答時間から計算機Bでのローカルな操作応答時間を減じて求めた。また、

表 3: 測定条件

測定環境	計算機 A(DSA1): Sun SPARC2 計算機 B(DSA2): Sun SPARC10 計算機 C(DSA3): Sun SPARC2
下位プロトコル	DUA/DSA 間, DSA/DSA 間ともイーサネット上で TCP/IP(socket インターフェース) を使用。
測定に用いた DIB	図 7に示す。部分木の各段において均等に分岐する。
測定内容	ランダムに選んだエントリの全属性 (5 個, 約 60 バイト) の読み出し (1,000 回)

図9にはUNIXのgprofによって取得したDSA内での主な内部処理が占める割合も併せて示した。

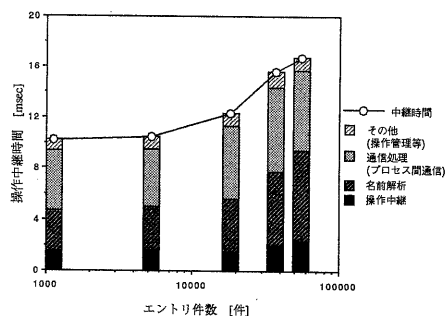


図 9: Read 操作中継時間

6.2.2 シャドウ

シャドウを評価するために、他 DSA からシャドウされたエントリに対して読み出しを行った場合と、シャドウを使わずにチェーンにより読み出しを行った場合の性能を測定した。まず、DSA1に1,000件のエントリを格納し、その葉エントリ26個を同じく1,000件のエントリを格納しているDSA3にシャドウし、DSA3に接続したDUAからDSA3に対してこれらのエントリへのRead操作の操作応答時間を測定した。次にシャドウ機能を使わずに、DSA3, DSA2を経由した場合の同一エントリに対するRead操作の操作応答時間を測定した。また、更新操作についても、シャドウを行う場合とシャドウを行わない場合の両方について、シャドウ範囲のエントリに対するローカルな更新操作応答時間を測定した。これらの結果を表4に示す。

表 4: シャドウ機能の評価

Read 操作	シャドウ有り	11.6 [msec]
	シャドウ無し	30.7 [msec]
更新操作	シャドウ有り	45.7 [msec]
	シャドウ無し	14.8 [msec]

7. 考察

7.1 ローカル操作と他 DSA への中継操作の性能について

図8, 図9より、計算機A上で18,000件のエントリ格納時には、ローカルなエントリに対するRead操作の操作応答時間は12.5ミリ秒、およびRead操作の中継時間は12ミリ秒であった。例えば、UPT等の高度交換通信サービスでの呼処理に必要なデータを提供するネームサーバを構築する場合、ネームサーバには100ミリ秒程度の操作応答時間が要求される^[7]。評価結果より、本 DSA を用いることで1~2段階程度の DSA の中継が入っても十分応答時間の要求を満たせることから、この呼処理用のネームサーバとしても適用できる高速性を持つと言える。内部の処理時間については、測定した格納エントリ件数の範囲では、全体に対する通信処理の割合が

1/3~1/2 程度を占めている。この結果より、ソケットによるプロセス間通信処理がボトルネックになるまで高速化が図れたことが分かる。

7.2 シャドウ機能の性能について

シャドウされたエントリも通常のエントリと同様の形態で格納されるため、検索操作に対しては表4より、ローカルなエントリと同程度の 11.6 ミリ秒という応答時間となった。シャドウ機能を用いない場合の Read 操作応答時間は、DSA3, DSA2 を経由して DSA1 のエントリを読むため、シャドウ機能を用いる場合のおおむね 3 倍になっており、シャドウによる検索操作の応答速度の向上が確認できた。なお、更新操作については、シャドウ機能を用いるとシャドウ元 DSA ではシャドウログへの書き込みやログ内容の送信処理等のオーバーヘッドにより 45.7 ミリ秒の操作応答時間を要している。さらに表4には現れないが、シャドウ先でも通常のシャドウ無しの場合に相当する更新処理が必要となるため、全体ではシャドウ無しの場合と比較して、更新操作は 4 倍程度の処理量になっていると考えられる。

7.3 分散処理による高処理能力化について

本実装では DIB の格納分散とシャドウによる負荷分散の両方を実現している。そこで DSA 数を増加した場合に得られる系全体の処理能力の向上について考察する。まず議論の簡略化のために以下の前提を設ける。

- 操作は格納するエントリ全てに対して、また存在する DSA に対して、ランダムに行なわれる。
- 格納エントリ件数の変化によって操作に必要な処理量は変化しない。
- DSA が複数存在する場合、全ての DSA 間に通信路が存在し操作は最悪 1 中継で実行される。
- DIB は全ての DSA に同じエントリ件数ずつ分散格納される。
- シャドウを行なう場合、DSA はローカルに格納する全てのエントリを全ての DSA にシャドウする。

評価で得られた操作応答時間を各々必要な処理量とみなした場合について、単体の DSA の処理能力を 1 としたときの DSA 数と系全体の処理能力のグラフを図 10 に示す。図 10 から単純に格納分散を行なった場合、更新比率に関わらず DSA 数にほぼ比例して処理能力の向上が得られることがわかる。またシャドウを行なうことにより、更新比率によっては単独 DSA の方より処理能力が悪くなる場合もあるが、更新比率が小さい場合は、前節から検索操作は常にローカル操作と同等の応答時間が得られ、さらに 2 倍程度の処理能力の向上が得られることがわかる。

8. おわりに

本稿では、シャドウを含む OSI ディレクトリの分散処理機能全てを提供する DSA の実装について報告した。既に開発済みの DIB 専用 DBMS: ASSIST/D をベースに実装し、ハッシュによる高速名前解析処理方式を採用して高速化を図った。シャドウの実現においては、シャ

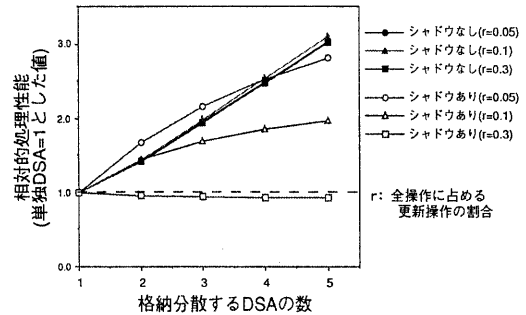


図 10: 格納分散とシャドウによる処理能力の向上
ドウ元が常に名前解析処理に必要な分散処理知識を補完してシャドウ先に転送することで、シャドウ先ではシャドウされた DIB 部分木とローカルな DIB 部分木を名前解析処理で区別する必要がなく、シャドウに対する高速な検索操作を可能とした。評価の結果、DSA がエントリを 18,000 件格納した状態で、Sun SPARC2 上で、ローカルな Read 操作を 12.5 ミリ秒で、また Read 操作の他の DSA への中継を 12 ミリ秒で実行でき、UPT 等の高度交換通信サービスでの呼処理に必要なデータを提供するネームサーバとしても適用できる高速性があることが示された。現在、OSI プロトコルスタックについても実装を進めている。最後に日頃御指導頂く KDD 研究所浦野所長、眞家次長に感謝します。

参考文献

- [1] ITU-T Draft New Recommendation X.500 Series: The Directory (1992).
- [2] 小花 他: ユニバーサルパーソナル通信 (UPT) への OSI ディレクトリの適用と評価, 電子情報通信学会論文誌 Vol. J74-B-1, pp. 959-970 (1991).
- [3] 西山 他: 拡張可能 DBMS 構築技法に基づく高速 OSI ディレクトリ専用 DBMS の設計と評価, 情報処論文誌 Vol.34, No.6, (1993).
- [4] 西山 他: OSI ディレクトリ情報ベース (DIB) のための高速名前解析処理方式, 情報研究会報告 MDP 61-29, (1993).
- [5] 西山 他: ハッシュを用いた OSI ディレクトリの高速名前解析処理方式, 第 47 回情報大全 6F-2, (1993).
- [6] 西山 他: OSI 管理情報ベース (MIB) 用データベースの設計と実装, 情報研究会報告 DBS 95-7, (1993).
- [7] Bellcore: Advanced Intelligent Network (AIN) Release 1: Service Logic Program Framework Generic Requirements, Issue 1, (1991).