

オブジェクト指向データベースの性能評価と問題点 (1)

1E-1

安村 義孝 佐伯 剛幸 鶴岡 邦敏

NEC C&C 研究所

1 はじめに

現在我々が開発中であるオブジェクト指向データベース管理システム (OODBMS) Odin の有効性を確かめるために、OODBMS のベンチマークテストとして知られる OO1[2] と OO7[1] を利用して性能評価を行っている。本稿では、オブジェクト識別子の実現手法など Odin の基本アーキテクチャを述べた上で、OO1 を用いた性能測定結果を報告し、データ構造の違い (リンク構造と集合構造) やデータベース (DB) の規模の違いによる影響について考察する。また、現在市販されている商用の OODBMS や RDBMS に対しても同様の性能測定を行い、それらとの比較についても述べる。

2 Odin のアーキテクチャ

Odin の機能を利用するアプリケーションは、DB 演算や集合操作等の機能を備え、ODMG-93 で規定されている仕様の一部を取り込んだ C++ の拡張言語である Odin/C++ を用いて記述される。これを Odin カーネルを構成するライブラリとリンクすることによってアプリケーションの実行形式が得られる。

Odin で扱われるオブジェクトは固定長のページ内に格納され、ページ単位で DB のアクセスが行われる。オブジェクト識別子の実現は大量性・保守性・信頼性のバランスを考慮して、オブジェクトキャッシュ (仮想記憶) 上では間接アドレス (OIA)、DB (二次記憶) 上では論理識別子 (OID) を採用している [3]。OIA はアドレス管理テーブルを介して、間接的にオブジェクトを指す。オブジェクトがキャッシュ上にあるか否かは、ハッシュ管理テーブルにより管理する。オブジェクトがキャッシュ上にあればアドレス管理テーブルを利用でき、そうでなければ DB 内のオブジェクトが存在するページが指定される。このようにして OID はオブジェクトがどこに存在しているようとも一意に特定することができる。

キャッシュ上に存在しないオブジェクトへの参照が起こった場合には、DB からそのオブジェクトが含まれるページがキャッシュに読み込まれ、参照したオブジェクト識別子が OID から OIA へポインタ変換 (swizzling) され、ハッシュ管理テーブルとアドレス管理テーブルのエントリが登録される (図 1)。一旦、キャッシュに読み込まれれば、アドレス管理テーブル上のエントリは捨てられることはなく、その後のアクセスは OIA のままで可能である。また、トランザクションのコミット時には、更新されたページが DB に書き込まれることになるが、その時ページ内の OIA はすべて OID に変換される。

オブジェクトの集合操作では、集合内に存在する要素を B 木の形式で管理し、その B 木を指すオブジェクトを集合オブジェクトとして実現している。現在のところ

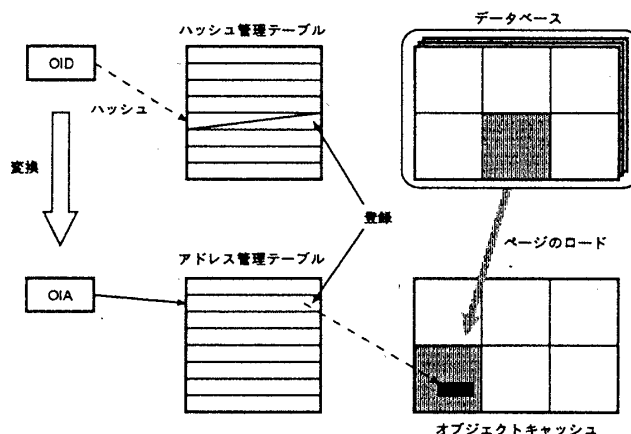


図 1: ポインタ変換

る Set、Bag、List、Varray、Extent の集合タイプがある。集合の要素としてはオブジェクトへのポインタに限らず、int 等の基本型や構造体も格納可能である。また、集合オブジェクトはインスタンス内に埋め込むこともできる。

3 性能評価

Odin と商用 OODBMS のベンチマークテストとして OO1 を用いた基本性能の評価結果について報告する¹。測定に使用したマシンは主記憶 32M バイトを持つ NEC 製 EWS4800/220² で、DB 領域は外付け磁気ディスク装置である (SCSI 接続)。キャッシュサイズはいずれも 8M バイトとし、ローカル DB のみの測定を行った³。

OO1 で用いるデータはパートとコネクションという 2 つのクラスからなる。小規模 DB でパート数は 2 万個存在し、個々のパートは他のパートに対して 3 個ずつある程度局所的に結合されている。この関係をコネクションで実現し、パート間を双方向に辿れる必要がある。また、大規模 DB ではパート数が 20 万個となるだけで、その他の条件は小規模 DB と同じである。

OO1 のデータ構造として、リンク構造と集合構造とを用いて測定を行った (図 2)。リンク構造では、パートにそれから結合しているコネクションとそれに結合されているコネクションの、それぞれ先頭のコネクションへのポインタを持たせ、各コネクションは片方向リンクで結ばれている。このようかなり複雑な構造をしていることがわかる。一方集合構造では、パート内に両方の埋め込み集合を定義し、そこにコネクションへのポインタを格納している。リンク構造に比べると単純でごく自然な

¹ Odin 新版は実装途中のため、そのデータは一部のみである。

² ただし、OODB2 のみ SparcStation330 で測定した。

³ この場合、Odin はクライアント-サーバの形式をとらない。

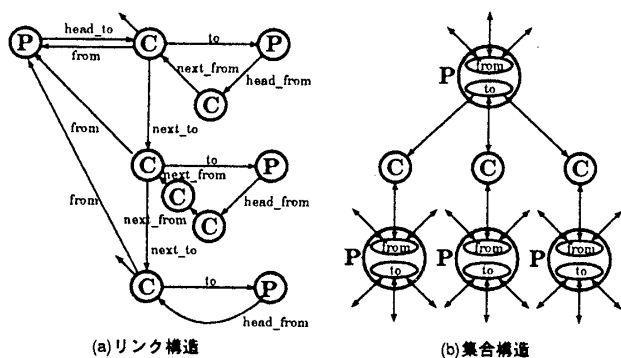


図 2: OO1 のデータ構造

表現方法である。また、どちらの構造もコネクション内に、どのパートからどのパートへ結合されているかの情報(逆ポインタ)を入れている。このリンク構造と集合構造を Odin/C++ で表現したクラス定義は図 3 のようになる。

```
class Part : od_spec {
public:
    int id;
    char type[10];
    int x, y;
    int build;
#ifdef LINK
    Connection *head_from, *head_to;
#elif SET
    Od_Set<Connection*> from, to;
#endif
};

class Connection : od_spec {
public:
    Part *from, *to;
    char type[10];
    int length;
#ifdef LINK
    Connection *next_from, *next_to;
#endif
};
```

図 3: OO1 のクラス定義

リンク構造でのトラバーサル(測定結果を表 1 に示す)で、warm(S) は小規模 DB でのウォームの結果、cold(S) と cold(L) はそれぞれ小規模 DB、大規模 DB でのコールドの結果である⁴。Odin と OODB{1,2} は、再帰的にパートとそれに結合されているコネクションのポインタにアクセスしてパート間を遷移する。それに対して RDB ではパートとコネクションの表を作り、パートとコネクションのそれぞれに対して毎回 select を発行する形で実現している。この結果から、RDBMS に比べて OODBMS が複雑な関係を持つ構造を辿るのに優れていることがわかる。特にウォームの性能は 2 桁以上の違いがある。また、小規模 DB では OODB2 の方が OODB1 よりも優れた性能を示しているが、大規模 DB ではそれらの性能比が縮まっていることがわかる。Odin の性能はウォームで OODB2 に近く、コールドではそれらの中間となる結果が得られた。

次に OODB{1,2} について、集合構造でのトラバーサ

⁴ただし、Odin は小規模 DB での結果のみ。

表 1: リンク構造の測定結果(単位: 秒)

	Odin	RDB	OODB1	OODB2
warm(S)	0.06	44.17	0.67	0.01
cold(S)	11.30	84.06	22.60	6.04
cold(L)	—	245.99	103.68	69.50

ルの測定結果を表 2 に示す。小規模 DB でリンク構造と比較すると、OODB1 でほぼ同等かそれ以上、OODB2 では僅かに性能が低下しているが許容できる範囲の性能差である。大規模 DB では OODB1 と OODB2 の性能は逆転している。これは小規模 DB と大規模 DB の性能比を考えると、リンク構造と集合構造のどちらの場合でも、OODB1 では約 5 倍、OODB2 では約 10 倍ということが影響していると思われる。しかし、RDB ではこの比率は約 3 倍である。

表 2: 集合構造の測定結果(単位: 秒)

	OODB1	OODB2
warm(S)	0.66	0.06
cold(S)	14.83	7.27
cold(L)	65.19	99.27

以上の測定結果から、OODBMS はオブジェクト間を遷移するような状況が多いアプリケーションに有用であり、これらの実現に対して集合構造を用いても性能の劣化があまり見られないことがわかる。しかし、DB が大規模になり、分散配置されたオブジェクトにアクセスする必要がある時のように、クラスタリングがあまり効かない場合に問題がある。それに対して Odin が持つアーキテクチャでは、OODB{1,2} と比べて比較的安定した性能が得られることが予想される。

4 おわりに

本稿では、開発途中である Odin 新版と商用 OODBMS の性能評価を OO1 を用いて行った結果について報告した。OO1 のデータ構造としてリンク構造と集合構造とを定義し、DB 規模の違いにより測定している。これらの結果から Odin の実現方法の有効性が確認できた。今後は、細部に渡るパフォーマンスのチューニングをする必要がある。また、Wisconsin ベンチマークを用いて問合せ処理の性能評価も行う予定である。

参考文献

- [1] Carey, M.J., DeWitt, D.J. and Naughton, J.F., "The OO7 Benchmark," *Proc. ACM SIGMOD*, pp. 12-21, 1993.
- [2] Cattell, R.G.G. and Skeen, J., "Object Operations Benchmark," *ACM Trans. Database Syst.*, pp. 1-31, 1992.
- [3] 鶴岡, 木村, "オブジェクト指向 DBMS のアーキテクチャに関する考察," 情報処理学会データベースシステム研究会 94-14, pp. 109-116, 1993.