OODBMS PERCIO-PC 版における C++-API の実装と性能評価

1Q－2

安村 義孝　鶴岡 邦敏
NEC C&C 研究所

1.はじめに

オブジェクト指向データベース管理システム PERCIO を PC 上に移植するにあたり、新規に提供する C++-API の実装方法について述べる。PERCIO の特徴の 1 つである間接仮想記憶アドレス方式をそのまま受け継ぎ、データベースアクセスのために必要なクラス情報を実行時に参照する構成となっている。また、ベンチマークテストを用いて PERCIO/C++で記述した場合との性能比較を行う。

2.PERCIO-PC 版

現在の PERCIO は ODMG-93[2]の仕様の一部を取り込んだ PERCIO/C++と呼ばれる C++の拡張言語でアプリケーション開発を行うようになっている。PERCIO/C++は可変長クラスや問合せのブロック式、ビューなどを定義することが可能な拡張機能を持つため、専用のトランスレータを用いて C++のコードに変換し、PERCIO カーネルを構成するライブラリとリンクしてアプリケーションの実行形式を得ることになる。この場合、デバッガなどのツール類を専用の開発環境に組み込んで提供する必要があるため、ユーザは今まで慣れ親しんでいる開発ツールを使えなくなるという弊害があった。

PC には Visual C++等の優れた言語開発環境が提供されており、これらの開発環境や既存のライブラリを利用したアプリケーションの作成を可能にすることが望まれる。今回、PERCIO を PC 上に移植するにあたり、既存の開発環境に対してオープンな API を提供するために C++-API を新規に開発した。本 C++-API は ODMG-93 で規定されている C++バインディングに基づいている。ただし、C++-API からは PERCIO/C++が持つ可変長クラスなどの拡張機能を利用できないという欠点がある。

C++-API を利用する場合、あらかじめクラス定義の情報をデータベースに登録するために専用のプリプロセッサを通す必要がある。このプリプロセッサを通すことにより C++-API 用のヘッダファイルが生成される。これは ODMG-93 における ODL プリプロセッサに相当する。また、ブラウザとデータベース保守機能を備えた統合開発環境を利用すると、クラス定義情報を対話的にデータベースに登録したり、登録したクラス定義情報をアプリケーション開発中に参照・更新することが可能となる。

3.C++-API の実装

C++-API は C++言語が持つテンプレート機能を活用したデータベース管理システムとのインタフェースである。データベースにクラス定義情報をあらかじめ登録しておき、C++-API 用のライブラリ内でそれらの情報を利用してデータベースにアクセスする。C++-API で記述したプログラムは通常の C++コンパイラを用いてコンパイルし、PERCIO カーネルとリンクすることで実行ファイルを生成することが可能となる。

PERCIO は大規模データベースでも安定した性能が得られるように間接仮想記憶アドレス方式を採用している[3]。C++-API もその方針にしたがって PERCIO カーネルとのインタフェースを規定している。本方式ではオブジェクト識別子をハッシュテーブルを介した論理的なアドレスとする。このアドレスはデータベース内で OID、メモリ上およびスワップファイル内で OIA となり、オブジェクトキャッシュ上でポインタ変換(Swizzling)を行っている。C++-API はこのオブジェクト識別子をテンプレートクラスの Ref 型で表現し、永続インスタンスと一時インスタンスのどちらにも共通である。Ref 型からメモリアドレス、メモリアドレスから Ref 型への変換演算子を用意することにより、通常の C++ポインタとして扱うことも可能である。

冗長な処理を減らすために、OID から OIA へのポインタ変換のタイミングは、オブジェクト識別子が指すインスタンスのアクセス時に行う。Ref 型の変数が自動変数の場合には、このインスタンスが存在するページにスワップロックが掛けられる。スワップロックを掛けることによりスワップアウトされなくなるので、メモリアドレスに変換された場合でも安全にそのページ内のメモリ領域にアクセスすることができる。Ref 型のデストラクタや代入演算子においてこのスワップロックが外されることになる。

インスタンスの生成やコレクションの操作時にはあらかじめデータベースに格納されているクラス定義情報が利用される。これらのクラス定義情報を持つシステムクラスのインスタンスに対するオブジェクト識別子は、ユーザ領域に確保してあるシステムテーブル内で管

An Implementation of C++-API for OODBMS PERCIO-PC and its Performance Evaluation
Yoshitaka Yasumura and Kunitoshi Tsuruoka
C&C Research Laboratories, NEC Corporation

理する。C++-API のクラス定義をプリプロセッサに通す時にそのためのコードが同時に生成され、アプリケーションコードとリンクされる。システムテーブル内の各項目の値はデータベースのオープン時に設定され、クラス定義情報が必要になるとクラス名のハッシュ値によりシステムテーブルへ参照することになる。これはC++-API 用のライブラリ内で行われる。

メンバ変数の更新やインデックス値の更新には特定のメンバ関数を呼ぶ必要があり、これはユーザ責任である。メンバ変数の更新では、該当するインスタンスが存在するページに更新フラグが立てられ、コミット時にデータベースへの書き込み対象のページとなる。インデックス値の更新では、該当するカーネル関数が呼ばれ、メンバ変数と B+木が更新されることになる。また、データベースに対する問合せはそのための問合せ文を文字列で渡し、その文字列を動的に解釈して PERCIO カーネル内の問合せ処理系を実行させる。ただし、現在のところ単一のコレクションに対する簡単な全件検索や順序付けのみが可能であり、ODMG-93 の OQL に相当する機能は持たない。

4.性能評価

OO1 ベンチマーク[1]を用いて本 C++-API の性能評価を行った。ここでは、C++-API で記述した場合と同様のベンチマークプログラムを PERCIO/C++で記述した場合とで性能を比較する。性能測定のマシンとしてメモリを 80M バイト持つ PC-9821Xt を利用し、OS は Windows NT Workstation 3.5 で、ベンチマーク用のデータベースは NTFS 上に作成した。オブジェクトキャッシュのサイズは OO1 ベンチマークの規定通り 8M バイトである。

データ構造はリンク構造と集合構造を定義した。OO1 ベンチマークに必要なデータは Part クラスと Connection クラスから構成されるが、Part とその間に存在する Connecion や、Connection 同士をポインタで結んだものがリンク構造である。一方、Connection へのすべてのポインタを Part 内のコレクションとして管理するものが集合構造であり、Part と Connection に 1 対多の関係が成立する。

C++-API と PERCIO/C++の OO1 ベンチマークによる測定結果として、小規模 DB の場合を表 1、大規模 DB の場合を表 2 に示す。測定結果の単位はいずれも秒である。ただし、Insert に関してはコミット処理も測定時間に含めるため、hot は測定の対象外とした。

表 1 と表 2 の測定結果により、cold の性能で違いがあるものの、hot の性能は全く同じであることがわかる。cold では OID から OIA へのポインタ変換が発生し、

表 1: 小規模 DB の測定結果

	C++-API				PERCIO/C++			
データ構造	link		set		link		set	
DB サイズ	7.7MB		12.3MB		7.7MB		12.3MB	
測定	cold	hot	cold	hot	cold	hot	cold	hot
Lookup	4.85	0.15	7.39	0.15	4.06	0.14	5.70	0.15
Traversal	5.99	0.04	11.92	0.16	5.91	0.06	8.92	0.17
Insert	14.53		15.93		16.38		12.13	

表 2: 大規模 DB の測定結果

	C++-API				PERCIO/C++			
データ構造	link		set		link		set	
DB サイズ	74.6MB		120.4MB		74.6MB		120.4MB	
測定	cold	hot	cold	hot	cold	hot	cold	hot
Lookup	7.77	0.15	12.04	0.15	8.02	0.15	8.78	0.16
Traversal	19.52	0.05	31.00	1.08	15.04	0.06	24.49	1.07
Insert	39.45		20.29		34.79		17.33	

C++-API ではオブジェクトキャッシュの探索のオーバヘッドがあるため、PERCIO/C++より多少性能が落ちてしまう。また、データベースが大規模になってもPERCIO/C++で記述した場合と同じ比率で測定結果が変化している。これらにより、本 C++-API はデータベースの規模によらず PERCIO/C++とほぼ同等の性能を持つと言える。

5.おわりに

本稿では、PERCIO-PC 版のために作成した C++-API の実装方法を述べた。本 C++-API は ODMG-93 の C++バインディングに基づき、PERCIO の特徴を継承するように設計されている。また、OO1 ベンチマークテストによる性能評価の結果、PERCIO/C++で記述した場合とほぼ同等の性能を持つことを示した。今後は、C++-API から PERCIO の拡張機能を利用できるようにする予定である。

参考文献

[1] Cattel, R.G.G. and Skeen, J., "Object Operations Benchmark," *ACM Trans. Database Syst.*, Vol. 17, No. 1, pp. 1-31, 1992.

[2] Cattell, R.G.G., *The Object Database Standard: ODMG-93, Release 1.1*, Morgan Kaufmann, 1993.

[3] 安村 義孝, 佐伯 剛幸, 鶴岡 邦敏, "オブジェクト指向データベースの性能評価と問題点(1)," 情報処理学会第 48 回全国大会講演論文集(4), 1E-1, 1994.