O/R マッピングツールをもちいた データベースアプリケーションに関する研究
- アクセス速度に関して -

2004MT047 小林 史弥
指導教員　野呂 昌満

1　はじめに

近年，リレーショナルデータベース (以下 RDB と呼ぶ) を扱うオブジェクト指向言語で作られたアプリケーションが増えている．RDB をオブジェクト指向アプリケーションが扱う場合，オブジェクトとリレーショナルデータとのデータ構造が異なっているのでその間を補完するための処理が必要となる．そのデータ構造の違いをインピーダンスミスマッチと呼ぶ．データベースの二次元の構造からオブジェクト指向設計の構造へマッピングするプログラムを開発する事が可能であるが，データベースを変更する度に開発し直す事が必要でありコストが高い．そこで O/R マッピングツールを使うことでデータベースのテーブルに対応するオブジェクトを自動生成しインピーダンスミスマッチを解決するプログラムの開発コストを削減することが出来る．しかしデータを取得する際アプリケーションによって必要なデータが異なるので，デフォルトの設定では関連するデータを一度に全て取得するように設計されている．そのため．データ取得時の処理時間が長くなる．本研究ではアプリケーションのソースプログラムからデータベース中の必要な項目を検出し，データ取得時に不要なデータを取得しない事でデータベースに対する処理時間の高速化を図る．

2　O/R マッピングツール

2.1　O/R マッピングツールの利点

Java の SQL ライブラリを利用してデータベースアプリケーションを作成する時に，オブジェクト指向モデルとリレーショナルモデルの設計思想の違いからインピーダンスミスマッチが生じる．またデータベースを変更する場合に多くのクエリを修正する必要があるのでアプリケーション開発の柔軟性が低い．インピーダンスミスマッチを解消しアプリケーション開発の柔軟性を向上させる方法として O/R マッピングツールを利用する事がある．インピーダンスミスマッチを解消させる API を作成する事によって問題を解決する方法もあるが，データベースによって扱うプロパティが変わるのでマッピングの内容が変わってしまう．データベースを変更する度に API の変更が必要となりコストがかかるので O/R マッピングツールを使う事が良い．

2.2　O/R マッピングツールの構造

O/R マッピングツールはオブジェクト指向モデルとリレーショナルモデルとの構造的な違いを緩和するためのツールである．本研究では Java 用のオープンソフトウェアの O/R マッピングツールである Hibernate[1] を使用した．Hibernate は Plain Old Java Object の永続クラスを活用してデータベースのテーブルのデータを格納してテーブルのデータをオブジェクトとして扱うためのクラスを生成する．また HQL という Hibernate 特有のクエリ言語を使いデータベースの操作をすることが可能である．Hibernate を利用するためには以下のファイルを用意する事でデータベース操作が可能となる．

- データベースの接続定義ファイル
  データベースに関する情報やテーブルのマッピングファイルのパスが記述されている．
- テーブルのマッピングファイル
  各テーブルの情報や他のテーブルとの関係について記述されている．(Hibernate によって自動生成可能)
- Database Access Object
  データベースにアクセスし操作するためのメソッドを備えている．

Hibernate は主に Configuration, SessionFactory, Session の三つのクラスによりテーブルと永続クラスのマッピングを行いクエリを生成をしている (**図 1**).

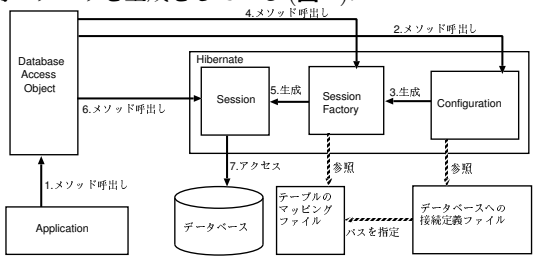

図 1　アプリケーションの構造

2.3　実行速度

データベースからデータを取得する際に O/R マッピングツールを利用時，非利用時で比較しデータベースへのアクセス速度の違いが示す．図 2 のスキーマに対し，Register, Student，Title を表結合してデータを取得する．

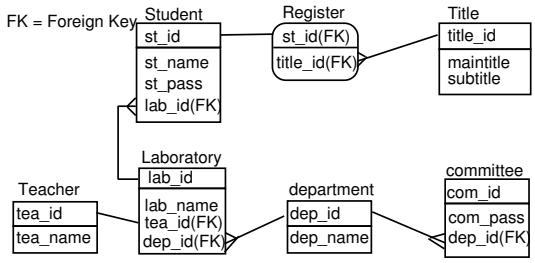

図 2　データベーススキーマ

データベースから取得するデータ数を変化させデータ取得にかかる時間を測定した．各 10 回測定し、その平均時間を表 3.1 に示した．Hibernate を使ったプログラムの結果を A とし，Java の SQL ライブラリを使用し，Hibernate と

似た機能を持たせたプログラムでデータを取得した結果をBとする．またBのアプリケーションでAと同じ表を結合させた結果をCとする．実行環境はOS:VineLinux 3.2, CPU:CeleronM 1.20GHz，メモリ:512MBである．

表1　データベースへのアクセス速度

データ数	1000	5000	10000	15000	20000
A	954.9	2773.5	4841.8	6976.0	9286.8
B	134.8	424.5	734.1	1095.9	1660.6
C	360.4	871.9	1480.2	2139.0	2789.6

(単位：ミリ秒)

2.4　アクセス速度の低下

Aは必要なテーブルに関連しているテーブルを全て結合してクエリを発行しているために処理速度が遅くなっている．またAとCで約3倍の違いがある事の原因はSQL文を自動生成している事やマッピングファイルを読み込んでいる事など，様々な種類アプリケーションやデータベースに対してもHibernateを汎用的に使用できる設計がされていることから処理速度が遅くなっている．

3　高速化手法

前章の実験では必要のないテーブルを結合している事が原因で実行速度が遅くなっていた．そこで本研究ではテーブルの結合をより細かく設定しアプリケーションが必要としているテーブルのみを結合させる事でHibernateの実行速度を向上させる．Hibernateはテーブルのマッピングファイルを設定をする事でテーブル毎に結合方法の設定を行う事ができる．その方法として大きく分けてイーガーフェッチと遅延フェッチの二つの方法がある．イーガーフェッチは関連オブジェクトのデータを全て含めて一度のクエリで取得する．遅延フェッチは関連オブジェクトのデータにアクセスした時にデータベースからデータを取得する．

マッピングファイルの設定を遅延フェッチにし，HQLを記述により必要なデータのみを一回で取得する事でデータベースへのアクセス速度を向上させる事ができる．

3.1　実験

図2のスキーマの時に，Register，Title，Student，Department テーブルのデータを取得してRegisterに登録されている学生の研究室名，主題を表示する時間を計測し．第二章と同じ環境で測定する．各10回測定しその平均時間を表3.1に示した．Hibernateをイーガーフェッチに設定してデータを取得した結果をA，遅延フェッチに設定した結果をB，そしてCはHQLを使い結合する範囲を指定してデータを取得した．

表2　データベースへのアクセス速度

データ数	1000	5000	10000	15000	20000
A	1090	3153	5426	7891	10523
B	3482	13529	25569	40517	49532
C	1145	2981	4980	7651	8953

(単位：ミリ秒)

3.2　実行速度について考察

Aは一度のクエリでデータを取得しているが，Department，Teacherテーブルを余分に結合しているためにデータ量が多くなり実行速度が遅かった．Bは扱うデータの数だけクエリを生成しているので実行速度は他の結果と比較して45倍の時間がかかった．遅延フェッチとイーガーフェッチのどちらがそのアプリケーションに適しているかはアプリケーションによって異なる．必要なデータが明確になればHQLを使って結合するテーブルを指定する事で処理を高速化できる．しかし扱うテーブルが一つだけだった場合はAと同じ実行速度になり，扱うテーブルが関連している全てだった場合はBと同じ結果になり提案した手法は効果を発揮しない．また，プログラムコードからデータの依存関係を把握し，それによりアプリケーションが必要としているデータを解析する事で必要なデータを把握する事ができると考えられる．そのデータを把握し取得するコードを生成するAPIを自動生成する事でO/Rマッピングツールをもちいたアプリケーションのデータベースのアクセス速度を向上させる事ができる．

4　関連研究

”アスペクト指向を利用した永続オブジェクト・アクセスの高速化”[2]では次の事が研究されている．

O/Rマッピングツールは RDB からのデータ取得において複雑な指定が困難である．そこでアスペクト指向プログラミングに基づいた O/R 間のデータ変換が可能な Java 向けの永続システムである AspectualStore の提案と開発を行っている．AspectualStore を利用することによってデータベースから取得されるデータをアプリケーションを変更することなく，柔軟にアスペクトとして指定することが出来る．それによってチューニングにかかるコストを削減することができる．この研究の実験によると AspectualStore を使ったときの結合数は減り速度が向上しているが，アスペクトを使っているので全体としての実行時間は既存の O/R マッピングツールとあまり変わらない．一方でアスペクトを使っているので既存のシステムに手を加える必要が無い事が利点となっている．アスペクト指向を利用して本研究とほぼ同じ目的の研究を行っているが，しかしデータ取得の際の必要なデータを把握する方法についての記述が無い点が本研究と異なっている．

5　まとめ

本研究では，O/Rマッピングツールを用いたデータベースアプリケーションのアクセス速度の高速化の手法を提案した．今後の課題は既存のソフトウェアを使ってデータの依存関係をしらべ，それにより必要なデータを解析してその情報から必要なデータを取得するために必要なクラスを自動生成する事である．

参考文献

[1] Christian Bauer, Gavin King 著，倉橋央，勝嶌和彦訳：Hibernate イン アクション，ソフトバンク クリエイティブ株式会社 (2006).

[2] 青木 康博, 千葉 滋, 佐藤 芳樹：“アスペクト指向を利用した永続オブジェクト・アクセスの高速化”, 日本ソフトウェア科学会第 22 回大会 (2005) 論文集, (2005.8).