

# Web アプリケーション構造解析に基づく データベースアクセス最適化の研究

神奈川大学大学院 工学研究科 経営工学専攻

\*田中 裕介<sup>ろう</sup>, 婁 慶博<sup>けいはく</sup>, 内田 智史<sup>さとし</sup>

## 1 はじめに

Web アプリケーションはデータベースからの情報取得がシステムのいたる部分で行われており、ユーザ数や扱う情報が多い大規模システムであるほど効率が要求される。しかし、大規模システムでは度重なる保守やシステムの複雑化によるデータベースの大規模化・セッション管理の煩雑化が原因となり、パフォーマンスと安定性に優れ、かつチューニングされたシステムを維持することが非常に困難となっている。そのため、今までにも様々な開発支援システム [1] や検証システム [2] が考案されてきた。

本研究では、Web アプリケーションを構成する各プログラムにおけるデータの取り扱いに着目して解析し、既存アプリケーションのセッション管理を自動的に最適化するアルゴリズムを提案する。具体的には、システム内の情報を取得する手段をデータベースアクセスとセッション変数利用の2つに分類し、データベースへのアクセスは時間資源を、セッション変数の使用はメモリ資源を消費すると考え、このトレードオフを考慮して最適化を行う。

## 2 前提知識

### 2.1 OBRDB

OBRDB(Object Based Relational DataBase) は 2003 年に神奈川大学工学部経営工学科内田研究室で開発された技術であり、Java オブジェクトを直列化した後に暗号化を施して、リレーショナルデータベースのカラムに直接格納する仕組みである。キーを暗号化しても高速に検索できることが特徴であり、その場合には SQL 言語とオブジェクトのハッシュ値を利用して検索を行い、格納時とは逆の手順でオブジェクトを復元する。Java オブジェクトを容易に永続化でき、オブジェクトそのものをデータベースから取得する仕組みであるため、本研究室で開発されている Web アプリケーションはすべて OBRDB 技術を用いている。本研究では、OBRDB を利用した開発済み

の Web アプリケーションを研究対象とする。

### 2.2 WebLec

WebLec は、2002 年に神奈川大学工学部経営工学科内田研究室で開発された授業管理支援システムであり、Web 上で出席管理・レポート管理・ミニテスト管理・授業 Web ページ作成・アンケート実施を行うことができる。ユーザビリティ向上のために Ajax などの最新技術も採り入れている。本学以外にも 3 校の大学で実際に利用されており、2007 年 1 月 11 日現在のシステム利用者数は、教員 38 名、学生 3,218 名である。開発者は毎年 5 人程度であるものの、システム規模は Java で約 20 万行、クラス数にして約 900 クラスと比較的大規模であり、改定も頻繁に行われることからセッション管理の徹底が急務であると言える。本研究では、提案する最適化アルゴリズムを WebLec に適用して評価する。

## 3 研究方法

Web アプリケーション内ではいたる部分でデータベースからのデータ取得処理が必要になる。このとき、セッション情報を上手く活用してデータベースの検索回数を最小に抑えることにより効率の良い情報検索が行えるが、システムが大規模化・複雑化するにつれ適切な開発が困難になり冗長な処理を記述してしまう可能性がある。そこで、セッション管理を最適化するために、まず、開発済み JSP/Servlet の構造解析を行うことで、Web アプリケーション構造やセッション変数の生成・破棄状況の可視化を行い構造理解の支援を行う。Web アプリケーションの構造を表すモデルとして本研究ではセッション・マップを提案する (図 1)。

さらに、構築した既存システムのセッション・マップを基にして、Web アプリケーション全体のセッション管理を最適化したものをチューニングされたセッション・マップとして自動的に提案するアルゴリズムを構築する。

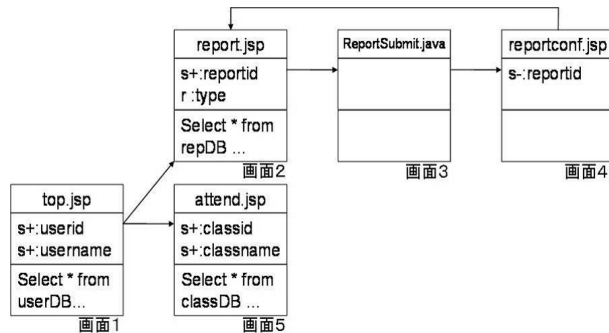


図 1: セッション・マップ

### 3.1 セッション情報の解析

ログイン中のユーザ情報を格納するセッション情報の管理は開発上の重要な課題である。しかし、大規模システムの度重なる改定に伴い、各プログラムに分散するセッション変数への情報格納・破棄の状況を認識し適切な管理を行うことは困難になる。そこで、まず対象となる Web アプリケーション全体の構造を解析し、画面遷移関係を把握した上で各画面ごとのセッション変数への情報の格納・破棄を含めてモデル化することを考える。

この問題を解くため、本研究では Web アプリケーションの構造を有向グラフとして扱い、JSP/Servlet により作られた画面の集合をノード V、各ページを繋ぐリンクの集合をエッジ E として表現する。また、ログインページをグラフの開始ノードとし、JSP/Servlet 内に記述された<form>タグや forward メソッドなどを検出することにより、システム全体の画面遷移の繋がりを探査を行う。

解析後の情報を図 1 のように整理して表現する。これをセッション・マップと呼ぶことにする。

セッション・マップでは、ノードの上段には JSP/Servlet ファイル名を記述し、中段には表 1 の表記法に従ってセッション変数・リクエスト変数の情報を記述する。また、下段にはデータベースアクセスを記述する。なお、条件分岐によって格納されない可能性のある変数については、カッコ付きで表記する。

表 1: セッション・マップの表記法

| 表記  | 意味         |
|-----|------------|
| s + | セッション変数の追加 |
| s - | セッション変数の破棄 |
| s   | セッション変数の利用 |
| r   | リクエスト値の送信  |

## 4 最適化手法

情報取得手段最適化の基本方針として、Web アプリケーション内で多く利用されているセッション変数を割り出し、これを最適化の基準とする。各セッション変数の評価にはコストと利用頻度の概念を導入する。

コストは、あるセッション変数で利用されるデータ量である。本研究では、Web アプリケーション全体で利用するセッション変数のデータ量に上限を設け、この制約の基に最適化する。現段階では、1 セッション変数あたりのコストは一定のものとして扱う。

また、利用頻度が高いほどそのセッション変数の重要性は高いと判断できる。そこで、それぞれのセッション変数について Web アプリケーションにおける利用頻度の計算を行う。単純に利用されている回数をカウントするのではなく、Web アプリケーションの画面遷移ネットワークを接続・選択・反復の部分木の集合として捕らえ、反復の中で利用されている場合は利用頻度を高く設定し、ログインページから分岐の先で利用されている場合は低く設定する。利用頻度の計算は、部分木の内部から再帰的に行っていく。

## 5 今後の研究方針

- セッション・マップ生成システム、セッション・マップ最適化システムの設計および実装を行い、実際のシステムである WebLec に適用して検証する。
- セッション・マップにページ遷移確率の概念を導入し、より正確なセッション変数利用頻度を割り出す。

## 参考文献

- [1] 加藤大樹, 結縁祥治, 阿草清滋, “拡張オートマトンに基づく Web アプリケーションコンポーネントの合成手法”, 日本ソフトウェア科学会第 21 回大会論文集, 2004 年
- [2] 崔 銀恵, 渡邊宏, “Web アプリケーションのクラス設計仕様に対するモデル化と検証”, ソフトウェアテストシンポジウム 2005, 2005 年