Profile-Driven Data Management

Mitch Cherniack
Department of Computer Science
Brandeis University

mfc@cs.brandeis.edu

Joint work: Mike Franklin (Berkeley), Stan Zdonik (Brown)
The State of the Web

Large Scale in Many Dimensions

*Users, data, data sources, data volatility, …*

Problems Revealed By Scale

*Competition for Resources (bandwidth, server capacity, …)*

Current Solutions (Driven By Access Patterns)

*Mirroring, proxy caching*
Déjà Vu for the DB Community

Early Years of Data Processing

*Competition for Resources* *(Disk, Buffers, CPU, …)*

Motivated Data Management

*Application-Driven Resource Management:*

– *suite of techniques:* indexing, clustering, …
– *knobs set based on application, data semantics*
– *determined by database administrator* *(DBA)*
The DBA Model

DBA = Human Expert

Assesses Data Needs of User Community

Tweaks Knobs Accordingly

(what to index, what to sort on, where to place data, …)

Model Characteristics

**Manual**: interaction w/ users (discussions, questionnaires, …)

**Holistic**: user needs, data constraints viewed as a whole

**Intrusive**: requires authority to impose sort order, indexes, …
Can This Work for the Web?

Data Management? Yes

DBA Model? No

Web is Large-Scale

Manual: Too Many Users, Too Much Data to Analyze

Holistic: Web-wide policies unrealistic

Web consists of Autonomous Data Sources:

Intrusive: External sites control own data
Outline

1. Motivation: The Web Permits No DBA

2. Profile-Driven Data Management
   - Architecture
   - Profiles: What, How, Where

3. Design of Profile Languages

4. Thoughts on Profile Processing

5. Summary
Profile-Driven Data Management

Alternative to DBA-Model

Key Concepts:

1. Profile
   Formal specification of data requirements

2. WeBMS
   Profile-Driven Data Management Provider
   Serves User/Profile Community
   theme-based? subscription-based?
WeBMS: An Architecture for PDDM
WeBMS: An Architecture for PDDM

The Role of Profiles:

1. *Identify Data User Cares About (Domain)*

2. *Specify Relative Worth of Data (Utility)*
What Does a Profile Look Like?

PROFILE Bookmarks

DOMAIN
  cnn = www.cnn.com
  ny = www.nyse.com
  na = www.nasdaq.com

UTILITY
  U (na) = 3;
  U (ny) = 2;
  U (cnn) = 1

END

Domain, Utility Clauses (Utility distinguishes from P/S profiles)

Integer utility values, \( \uparrow \) = more useful
How Could It Be Used?

Prefetching for a Web Cache [FS00]

Idea: Prestage User’s Proxy with Important Pages

Q: Why the Need for Utility? Why Not Prefetch All Pages?

A: Scalability: Many users may share cache.

Goal: Make the most users the most happy
Why Could PDDM Work for the Web?

**Manual**: Profile processing is automatic

**Holistic**: Only data/users of concern is that in community

**Intrusive**: Middleware – data we can control
Where Do Profiles Come From?

Learned By Data Mining?

Analysis of Clickstreams

Authored By Users?

Augment “Personalization Profiles” (e.g., Bookmarks)

With help of GUI (as with SQL)

Likely Some Combination of Above (Future Work)
Outline

1. Motivation: The Web Permits No DBA

2. Profile-Driven Data Management

3. Design of Profile Languages
   - More Example Profiles and Their Uses
   - A Profile Language Framework

4. Thoughts on Profile Processing

5. Summary
PROFILE Traveler

DOMAIN
R = related: www.hertz.com
S = +Pearson +"Chelsea Hotel" +Shuttle
D = +Pearson +"Chelsea Hotel" +Directions

UTILITY
U (S) = UPTO (1, 2, 0);
U (D) = UPTO (1, 1, 0);
U (R [D > 0]) = 1

Result: Object value is context-dependent

Domain: expressed with search engine inputs (Google)

Thresholds: (UPTO (k, a1, a2))

Domain Dependencies: (R [D > 0])

Traveler to Downtown Toronto:
Wants data describing how to get there from Pearson
Will go by rental car or shuttle
Only needs 1 set of directions, shuttle schedule
If takes rental car, needs directions

Each domain a set
Idea: Data Analog of Battery Recharging for Mobile Devices

Similar to Synchronization, but:

1. Location-Independent
   i.e., can plug device into any network jack to recharge

2. Incremental
   i.e., more bandwidth/connection ⇒ “better data”, robust in presence of disconnection

3. Profile-Driven
What is “Better Data”?

**UTILITY**

\[

text{U (S) = UPTO (1, 2, 0);}
\]

\[

text{U (D) = UPTO (1, 1, 0);}
\]

\[

text{U (R [D > 0]) = 1}
\]

<table>
<thead>
<tr>
<th># Objects Sent</th>
<th>Recharge Cache</th>
<th>Total Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>S, D</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>S, D, R</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>S, D, R, R</td>
<td>5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
PROFILE Academic

DOMAIN
S = SELECT School
FROM www.acm.org/JobAnnouncements.db
WHERE Area = “Databases”

P = SELECT Title, Author, Affil, Text
FROM www.vldb.org/Papers.db
WHERE Topic = “Query Optimization”

UTILITY
U (S) = 1;
U (P [#S [S.School = Affil] > 0]) = 10
END

Profile over structured data:
can express Domain with queries
join condition (S.School = Affil) in Utility clause
Used For Data Management

*Automatic Indexing [HC75, FON92, ...]*

Idea: Index domains from common data

Example:

PROFILE Academic1
DOMAIN
P = {papers w/ topic X}

PROFILE Academic2
DOMAIN
P = {papers w/ author Y}
# Lessons from the Examples

1. Data Type ⇒ Domain Definition Language

<table>
<thead>
<tr>
<th>Data Type</th>
<th>DDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any Web Accessible Data</td>
<td>URL-based</td>
</tr>
<tr>
<td>HTML, PDF</td>
<td>Google</td>
</tr>
<tr>
<td>Relations</td>
<td>SQL</td>
</tr>
<tr>
<td>XML</td>
<td>XML-QL, Quilt?</td>
</tr>
</tbody>
</table>

**Implication:** Multiple Profile Languages?
Lessons from the Examples

2. Common Idioms/Uses for Utility Expressions

Common Idioms:

- Conditional values
- Thresholds (UPTO)

Common Uses:

- Knapsack-like Cache Allocation

Implication: Single Profile Language?
Our Approach: A Language Framework

Modular Grammar ([DC90, BB95])

Some nonterminals (parameters) lack production rules
Instantiate each parameter with production rules

Framework

Utility Clause Fully Defined
Domain Clause Defined with Parameter, \textit{DomExp}
Our Approach: A Language Framework

Grammar Excerpt

Profile :: PROFILE IDENT DClause UClause END

DClause :: DOMAIN DomEq ; ... ; DomEq

DomEq :: IDENT = DomExp'

DomExp :: WWW . IDENT . Suffix

Suffix :: COM | EDU | ORG | GOV | ...

Implementation in Progress

Using PreCC [BB95]
Denotational Semantics ([CFZ01b])

Assumes Semantics for $DomExp$ ($[d] : O \rightarrow Bool$)

Semantics Defined for each Nonterminal

Can Use to Assess Values of Data Sets:

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>$[p] : 2^O \rightarrow \text{Int}$</td>
</tr>
<tr>
<td>UTEqn</td>
<td>$[u] : 2^O \rightarrow \text{Int}$</td>
</tr>
<tr>
<td>UTExp</td>
<td>$[e] : 2^O \rightarrow \text{Int}$</td>
</tr>
<tr>
<td>Cond</td>
<td>$[c] : O \times 2^O \rightarrow \text{Bool}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X$</th>
<th>$<a href="X">Traveler</a>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${r_1, r_2}$</td>
<td>0</td>
</tr>
<tr>
<td>${r_1, r_2, d_1}$</td>
<td>3</td>
</tr>
<tr>
<td>${r_1, r_2, d_1, s_1}$</td>
<td>5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Outline

1. Motivation: The Web Permits No DBA

2. Profile-Driven Data Management

3. Design of Profile Languages

4. Thoughts on Profile Processing
   - Cache Allocation Problem
   - Future Work

5. Summary
Cache Allocation Problem (CAP)

Informally:

*How to “best” fill a cache given:*

\[ O: \text{ A Finite Set of Candidate Objects} \]
\[ S: O \rightarrow \text{Int (object sizes)} \]
\[ P: \text{ A Profile} \]

where “best” is determined by \([p]\)

and where the cache is not filled beyond its capacity, \(C\)

Applications:

*Recharging, Prestaging*
Precedence-Constrained Knapsack [YY00]

Idea: *Knapsack problem* + “precedence constraints”

A allowed in cache only if B in cache

**CAP ≠ PCKP:**

PROFILE Problem ...

\[
\begin{align*}
U (A [\#B > 0]) &= 1; \\
U (B) &= 1; \\
U (C [\#A > 0]) &= 100
\end{align*}
\]

**Misses best solution for cache of 2 objects:** \(\{a, c\}\)
A Greedy Algorithm for CAP

PROFILE Traveler

DOMAIN
R, S, D

UTILITY
U (S) = UPTO (1, 2, 0);
U (D) = UPTO (1, 1, 0);
U (R [#D > 0]) = 1

END

<table>
<thead>
<tr>
<th># Objects Sent</th>
<th>Data In Cache</th>
<th>Total Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>S, D</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>S, D, R</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>S, D, R, R</td>
<td>5</td>
</tr>
</tbody>
</table>
Future Work

Profile Combination

Informally:

*How to reduce n profiles to 1 representative profile*

Applications: Generalizing any processing algorithm to n profiles

Initial Approach: Combine Profiles

 PROFILE P1 ...
   U (R) = 1;
   U (S) = UPTO (1, 2, 1);

 PROFILE P2 ...
   U (R) = 2;
   U (S) = UPTO (2, 3, 1);

 PROFILE P1+P2 ...
   U (R) = 3
   U (S) = UPTO (1, 5, UPTO (1, 4, 2));
Future Work

Profile Combination

How Do We Recognize “Equal Domains”?  

*Easy for URL’s, Undecidable for Queries*

*Don’t Need a Complete Solution*

*Agrawal and Wimmers [AW00]: Combining Preferences*
Future Work
Utility Value Conditions for Profile Languages

1. Context
   e.g., workflow-based utility (schedule, …)
   e.g., location-based utility

2. Resolution
   e.g., low-resolution paper has value (albeit lesser than full)
   Processing strategies for data recharging

3. Data Quality
   e.g., utility based on freshness, corroboration, recommendations, search engine ranking
Future Work

Data Gathering

Work Of Many Communities Will Apply

Finding Data: Information Filtering [FD92]

Preprocessing Data: Data Warehousing [CD97, Wid95, …]

Keeping Data Fresh: Continual Queries [TGNO92, LPT99, …]

Restructuring Data: Web Querying [MMM97, FDL+99, …]
Other Related Work

Profiles

- Publish/Subscribe [OPSS93]
- Personalization (e.g., MyCNN, PointCast [RD98])
- Preferences [AW00]

Automated Data Management

- Data Freshening [CGM00]
- Web Cache Prefetching [FS00]
- Automatic Index Selection [HC76, FON92, …]
- AutoAdmin [CN98, ACN00]
Summary

The Web Needs Data Management

*But DBA’s Can’t Provide It*

Profile-Driven Data Management

1. *Middleware/Overlay WeBMS*
2. *Profiles*

Profile Languages and Processing

1. *A Language Framework*
2. *Cache Allocation*
3. *Profile Combination*...