Front-end Caching for Dynamic Web Content

MANOLIS VELISKAKIS

Knowledge and Database Systems Laboratory
School of Electrical and Computer Engineering
National Technical University of Athens

Motivation

Most of Nowadays Web Caching approaches concern only Static Web Content

Dynamic Web Content percentage continuously increases

Web Caching Approaches for Dynamic Content are needed

Back-End Approaches address only Web Server-related delays

Front-End Approaches address both Web Server-related and Network-related delays

Front-End Approaches are theoretically preferred

OUR GOAL
Definition of a Front - End Web Caching Algorithm for dynamically generated content
Static Vs Dynamic Content

Static Web pages
- Already cloned to Web Server
- Low update rate
- Presentation and Content are not request-independent
- One request corresponds to one Web Page

Dynamic Web Pages
- Produced on Request
- High update rate. Two-level update.
  - Updates to the Applications/Presentation Level
  - Updates to the Back-end Database (Content Level)
- Low degree of reusability
  - Presentation and Content are request-dependent (i.e. queries on the web)
  - The same request might correspond to several Web Pages (i.e. cookies)
  - A dynamic web page might change on every access (i.e. visitors counter)
- Strong dependency with the back-end site infrastructure
  - The existence of both an Application and Data Content is a prerequisite

Conclusions
- Web Caching approaches concerning Static Content are not efficient for Dynamic Content.
- Front-end caching approaches are difficult to implement for dynamic web pages.
- Following issues must be revaluated
  - What to Cache?
  - Replacement Policy
  - Prefetching Algorithms
  - Cache Consistency

Related Work

Nowadays
Front-end caching for dynamic web content

Approaches

1st Approach: Caching only back-end Database content
- Semantic Caching [DAR VLDB96]
- Form-based Proxy [LUO VLDB01]
- OBProxy [AMIRI VLDB03]

Handling and manipulating the query results used by application programs
- Oracle Web Cache [ANTON SIGMOD02]
- Cache Tables [ALTINEL VLDB03]
- KFCache [LARSON ICDE04]

Full-fledged database caching techniques

2nd Approach: Giving Application Logic to Front-end caching points
- Cache Applets [CAO MIDDLEWARE98]
- DCCP [SMITH USINEX99]

3rd Approach: Partial-Page web caching
- Weave Framework [YAGOUB VLDB00]
- Dynamic Proxy Cache [DATTA SIGMOD02]
- Cache Applets [CAO MIDDLEWARE98]
- DCCP [SMITH USINEX99]

Nowadays
Front-end caching for dynamic web content

Disadvantages
- Handle only database content
- Prerequisite the existence of an edge Application Server
- Server-Controlled
- None consistency or replacement algorithm is proposed
- Handle dynamic web pages without database content
- Handle the static parts of the dynamic web pages
Our Goal

CREATION OF A FRONT-END CACHING FRAMEWORK FOR DYNAMIC WEB PAGES

Main Characteristics
- Handles both the presentation and content level caching of a dynamic web page
- Client-Controlled
  - Transparent
  - Independent

OUR PROPOSAL
Cache the GENERATION PROCESS of a dynamic web page and NOT the web page itself

What to Cache?

![Diagram showing the caching process]

CACHE?
- Result
- Generation Process
- Initial Content
Giving Application and Database Logic to Proxy

General Characteristics

- Attach every dynamically generated web page to its corresponding Application
- Attach every dynamically generated web page to its corresponding back-end Content
- Cache both corresponding Applications and back-end Content in the Proxy
- Give Proxy the ability to produce on request the dynamic web pages

DOMProxy Structure

DOMProxy enables caching for both static and dynamic web pages

It is composed of two main components:
- A standard proxy that cares for the caching of static content
- The Dynamic Object Module (DOM) that provides the functionality for caching dynamic content
**Data Manager**

**Data Manager - Semantic caching**

- Controls the storage and manipulation of the cached data objects.
- The kernel of the Data Manager is an implementation of Semantic Data Caching
  - Each data object represents a single query, the unit with the lowest granularity in our data manipulation approach.
  - Similar queries are grouped in query classes, which are represented by different semantic spaces depending on their FROM clause, meaning that simple queries over the same base table are stored in the same semantic space.
  - Each semantic space is n-dimensional, where dimensions are defined from the number of the attributes that appear in the WHERE clause of the corresponding query class.
  - Tuples of a query result are points in this n-dimensional space; therefore new posed queries are represented by sets of regions.

**Levels of Granularity**

DOMProxy's cached objects are characterized from different levels of granularity.

- **Data Objects**
  - The *semantic regions* and their corresponding cached data define the lowest granularity level.
  - The corresponding data are grouped according to their semantic description and compose the semantic spaces, which comprise the second level of granularity.
  - The different databases that these semantic spaces correspond to, belong to the third level of granularity, databases.

- **Application Objects**
Replacement Policy

**Our Approach**

- Customizable replacement approach with dynamic granularity.
- Three different groups of eviction-criteria, equally to the three different granularity levels that cached data objects present.
- A weight-based algorithm that will take under consideration the spatial and temporal locality of the cached data objects, along with standard ones like size and retrieval cost will be the most suitable solution.

Cache Consistency

- 2 - LAYER Cache Consistency Policy

**Open Issues**

- One type of Cache Consistency Policy for both layers?
- Definition of the relationship between Presentation and Content Level
Prefetching

Prefetching is defined as the fetching of web objects from the back-end web server without first being requested.

**Main Issues**
- Predicting future requests
- Definition of the information that the predictive algorithm uses as an input in order to make efficient predictions.

**DOMProxy Characteristics**
- Information can be gathered from:
  - Application Objects usage
  - Data Object usage
  - Semantic Regions usage
  - Semantic Spaces usage
  - Database tables usage
  - Databases usage

**DOMProxy Characteristics**
- Information can be gathered from:
  - Application Objects usage
  - Data Object usage
  - Semantic Regions usage
  - Semantic Spaces usage
  - Database tables usage
  - Databases usage

New Challenges

Cooperative Proxy Scheme

Cooperative Proxy Scheme is defined as the operation of individual proxies sharing their cached objects with each other’s clients.

**Main Issues**
- Definition of the Scheme (Hierarchical, Directory Base etc)
- Definition of the cached objects’ distribution function among the different proxies

**DOMProxy Characteristics**
- Uses a grouping-oriented caching approach based either to the different nature of the cached objects (application and data objects) or to their different granularity level (semantic regions, semantic sets, databases)
- Has the ability to define the existent dependencies among the cached objects regardless of their granularity level.

New Challenges

New Challenges
Summary

- Nowadays Web caching approaches concerns Static Web Content
- Dynamic Web Content percentage continuously increases
- Front-End caching approaches address both Web server-related and network related delays
- New approach: Cache the generation process of dynamic web pages in a Proxy
  - Give Application and Database logic to Proxy
  - Transparency is critical
  - Web Caching approach must be deployment-independent (Forward Proxy, Edge Proxy, Reverse Proxy, Standard Proxy) → Client-controlled
  - Different granularity levels of the cached objects.
- Separate the presentation level from the content level.
- New issues arise regarding
  - Replacement Policy
  - Consistency Policy
  - Prefetching
  - Cooperative Schemes

QUESTIONS - COMMENTS