Building adaptable cache services

Laurent d’Orazio
Fabrice Jouanot
Cyril Labbé
Claudia Roncancio
(firstname.lastname@imag.fr)

Motivations (1)

• Observations
  – Caching is used to improve performances in many applications
  – Performances result from specific behaviors of a cache
  – Grids may use caches at different levels

<table>
<thead>
<tr>
<th>Persistent object manager</th>
<th>Web searching system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Id01</td>
<td>Barjavel Werber</td>
</tr>
<tr>
<td>Cache</td>
<td>Cache</td>
</tr>
<tr>
<td>Storage</td>
<td>Cache</td>
</tr>
<tr>
<td></td>
<td>Cache</td>
</tr>
<tr>
<td></td>
<td>Proxy</td>
</tr>
<tr>
<td></td>
<td>Server</td>
</tr>
</tbody>
</table>
Motivations (2)

- **Consequence**
  Caches are usually built from scratch for a given application
  ⇒ Time (and money) consuming

- **Solution**
  Static adaptation of cache services

- **Proposal**
  A framework for building adaptable cache services

Adaptable cache service motivations

- **Observation**
  Applications and system environment using a cache may change

- **Consequence**
  Behaviors of a cache may no longer be well suited

- **Solution**
  Dynamic adaptation of cache services

- **Proposal**
  Using a component model to build cache services
ACS a framework for building **Adaptable Cache Services** (1)

ACS

Instantiation

Instantiation

Specific caches services

ACS a framework for building **Adaptable Cache Services** (2)

Web Cache

DBMS Cache

... Grid Cache

Persistent Object manager Cache

Generic Cache

Basic concerns of a cache service
Plan

• Motivations
• Basic concerns of a cache service
  – Addressing
  – Searching
  – Replacement
  – Resolution
• ACS a framework for building adaptable cache services
• Using ACS in a data grid middleware
• Related work
• Conclusions and perspectives

Addressing in the cache

• Role
  Identifying an entry.
• State of the art

<table>
<thead>
<tr>
<th>Identity-based</th>
<th>Semantic</th>
</tr>
</thead>
</table>

• Example
  Semantic caching

Semantic Identity-based Cache

Werber & Barjavel
Barjavel
Searching in the cache

• Role
Finding an entry.

• State of the art

<table>
<thead>
<tr>
<th></th>
<th>Indirect search</th>
<th>Compact table</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Search</td>
<td>Translation</td>
<td>Sorted table</td>
<td>Tree based</td>
</tr>
<tr>
<td></td>
<td>table</td>
<td>Chained table</td>
<td>Hash table</td>
</tr>
<tr>
<td></td>
<td>Unsorted table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Example
R-tree

T

Barjavel

Barjavel

Barjavel

Werber

Pennac

Brown

Replacement of cache entries

• Role
Choosing entries to remove when the cache is full.

• State of the art

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>Entry moment</th>
<th>Use moment</th>
<th>Use frequency</th>
<th>Semantic locality</th>
<th>Cost / Benefit</th>
<th>Hybrid</th>
</tr>
</thead>
</table>

• Example
Cost / benefit
• Retrieval time
• Size
• Frequency use
Cache miss resolution

- Role
  Finding where to look when a cache miss occurs.

- State of the art

<table>
<thead>
<tr>
<th>Direct resolution</th>
<th>Hierarchical resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP</td>
<td>CARP</td>
</tr>
<tr>
<td>CD</td>
<td>Other</td>
</tr>
</tbody>
</table>

- Example
  Internet Cache Protocol (ICP)

Synthesis

- Observation
  Several possibilities for each concern

- Interpretation
  Each concern can be considered as an adaptation point for a cache service
Example 1: Caching in a persistence object manager

<table>
<thead>
<tr>
<th>Addressing</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Search</td>
<td>Indirect Search</td>
</tr>
<tr>
<td>Translation table</td>
<td>Compact table</td>
</tr>
<tr>
<td>Unsorted table</td>
<td>Sorted table</td>
</tr>
<tr>
<td>Replacement</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>Entry moment</td>
</tr>
<tr>
<td>Direct resolution</td>
<td>Hierarchical resolution</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolution</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP</td>
<td>CARP</td>
</tr>
</tbody>
</table>

Example 2: Caching in a web searching system

<table>
<thead>
<tr>
<th>Addressing</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Search</td>
<td>Indirect Search</td>
</tr>
<tr>
<td>Translation table</td>
<td>Compact table</td>
</tr>
<tr>
<td>Unsorted table</td>
<td>Sorted table</td>
</tr>
<tr>
<td>Replacement</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>Entry moment</td>
</tr>
<tr>
<td>Direct resolution</td>
<td>Hierarchical resolution</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolution</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP</td>
<td>CARP</td>
</tr>
</tbody>
</table>
Plan

- Motivations
- Basic concerns of a cache service
- ACS a Framework for building adaptable cache services
- Using ACS in a data grid middleware
- Related work
- Conclusions and perspectives

Architecture motivation

- Needs
  Dynamic and static adaptation

- A solution: component model
  Based on the separation of concerns principle, enables minimizing parts of the service to modify during an adaptation.
Basic concerns of a cache service

Addressing
Searching
Replacement
Resolution

Cache manager

- Role
  To Bind and find entries in the cache

Cache Manager
Replacement
Resolution
Replacement manager

• Role
  To capture the replacement strategy used to determine items to remove when the cache is full

Resolution manager

• Role
  To capture the resolution protocol used when a cache miss occurs
ACS Architecture

Adaptation of a cache service
Prototype

- Behaviors
  - Java
  - Fractal

- Motivations
  - Reuse of Perseus cache framework
  - Fractal enables static and dynamic adaptation

Plan

- Motivations
- Basic concerns of a cache service
- ACS a Framework for building adaptable cache services
- Using ACS in a data grid middleware
- Related work
- Conclusions and perspectives
Using ACS in a data grid middleware

- **Context**
  Gedeon project: middleware for data management in grids

- **Entities manipulated**
  - Files
    Example: picture.jpg
  - Records: a set of meta data
    Example: \{(Key, k01), (Author, Dupond), (Year, 2005)\}

Caching in Gedeon

- **Record cache behaviors**
  - Semantic addressing
  - R-tree searching
  - Semantic replacement
  - Hierarchical resolution

- **File cache behaviors**
  - Identity based addressing
  - Hash table searching
  - LRU replacement
  - Hierarchical resolution
Plan

• Motivations
• Basic concerns of a cache service
• ACS a framework for building adaptable cache services
• Using ACS in data grid middleware
• Related works
• Conclusions and perspectives

Caching

• Very rich literature on the topic
  Example: some references about semantic caching
  [KB96] [MS05] [RD98] [RD00] [R91] [AK94] [CB98] [CRS99]
  [DRSN98] [F82] [GG97] [XZLL05]

• Context dependant caches
  – DBMS [CFLS91]
  – Web [C]
  – Mediator [AKS98]

• Caching mechanisms
  – Tuple, page and semantic caching [DFJST96]
  – Hierarchical and distributed caching [CDNSW96]

• Framework for building cache services
  – Perseus [GDC03]
  – CaLi [Z04]
Perseus [GDC03]

• Definition
  A persistence object manager framework.

• Advantages
  Separation of concerns, capturing
  – Addressing
  – Searching
  – Replacement

• Difference with ACS
  Resolution not captured in the framework

CaLi [Z04]

• Definition
  A cache framework

• Advantages
  Separation of concerns, capturing
  – Addressing
  – Searching
  – Replacement
  – Distributed caching

• Difference with ACS
  – No dynamic adaptation
  – Distributed entries and communications VS resolution manager
Plan

• Motivations
• Basic concerns of a cache service
• ACS a framework for building adaptable cache services
• Using ACS in data grid middleware
• Related work
  • Conclusions and perspectives

Conclusions

• Basic concerns of a cache service
  – Addressing
  – Searching
  – Replacement
  – Resolution

• Architecture of a framework
  – Static adaptation
  – Dynamic adaptation

• Prototype
Perspectives

• Experimentation
  – Gedeon, middleware for data management in grids
    • Record caches
    • Data caches

• Interactions with other concerns
  – Prefetching
  – Concurrency
  – Consistency

Thank you

Questions ?
References (1)

- [C] Cern server on-line. URL: http://www.w3.org/Daemon/.

References (2)

References (3)


