Caches et optimisation des accès aux données

Notes de cours

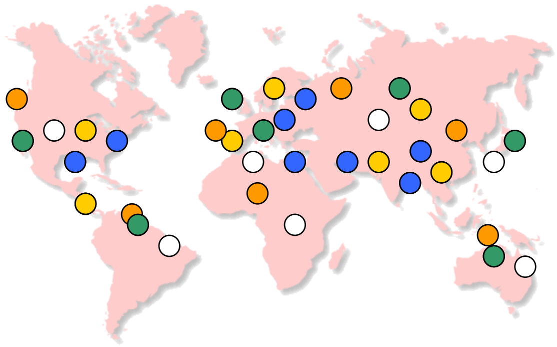
Claudia Lucia Roncancio

Grenoble INP - Ensimag

Caches « everywhere »

Nombreux contextes

- – SGBD++
- – Navigateurs
- – Moteurs de recherche
- – Grilles
- – *Cloud*



•Configuration adaptée !

Notion et utilité des caches

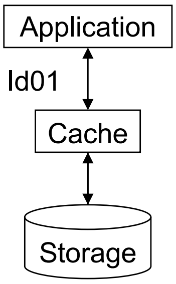
- Support à accès rapide pour un ensemble de données

- Avantages espérés des caches
  - Réduction latence
  - Réduction / répartition charge
  - Augmentation de la disponibilité
- Réduire le temps d'attente perçu par l'utilisateur
  – Réduction du temps de communications
    - Données proches des clients
  – Réduction du temps de calcul
    - Calculs en cache (localité sémantique)

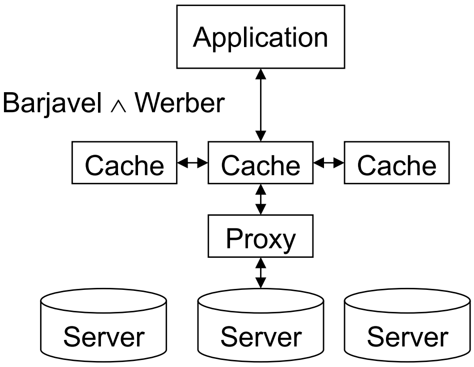
  ⇒ Réduction de la charge des serveurs
- Energie ?

Examples

Persistent object manager



Web searching system



Aspects intervenant dans un cache

- Adressage
  - Identification d’ une entrée dans le cache
- Recherche
  - Trouver une entrée dans le cache
- Résolution
  - Résoudre un défaut *d ’objet*
- Remplacement
  - Choix des données à effacer pour libérer de la place

Query and semantic caching

- Query caching
  - Caching and reuse answers to preceding queries
- Semantic Caching
  - Semantic treatment on cached answers
- One or multi-sources queries
  - Handling sources heterogeneity

Advantages of semantic Caching

Advantages

- – reduce response time / better performance
- – important when accessing paying site
- – minimize network traffic
- – most appropriated large distributed systems

Drawbacks

- – Tuning

Example query Language

- Conjunction of positive or negative terms
- Term of form `Attribute` *Operator* Value
  - – `Attribute` can be `Title`, `Author`, `Full-text`,…
- Result: set of tuples

  a tuple is a list of (Attribute, Value) pairs

For simplicity in the following we use

“Web AND caching”

for query: `Title` contains “Web” AND “caching”

Semantic Cache Architecture

- Cache managed as set of regions

  access information and cache replacement managed at the unit of regions

- Region groups together semantically related data (query, sub-query, merge of queries)

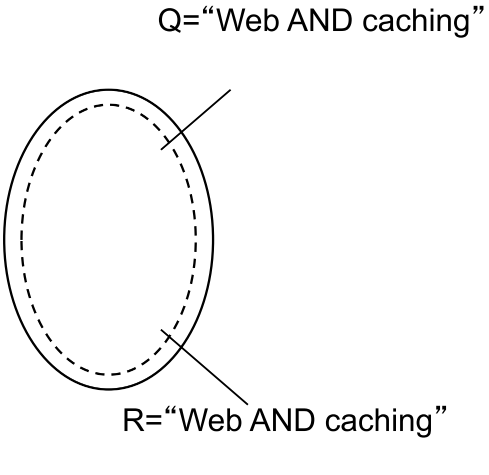
Operational Model

- Query Processing
  - incoming user query is split into probe and remainder query
    - Probe = portion of the answer provided by the cache
    - Remainder = retrieving missing data from the server
- Cache Maintenance
  - coalesce or split regions
  - replace regions to free space for new ones

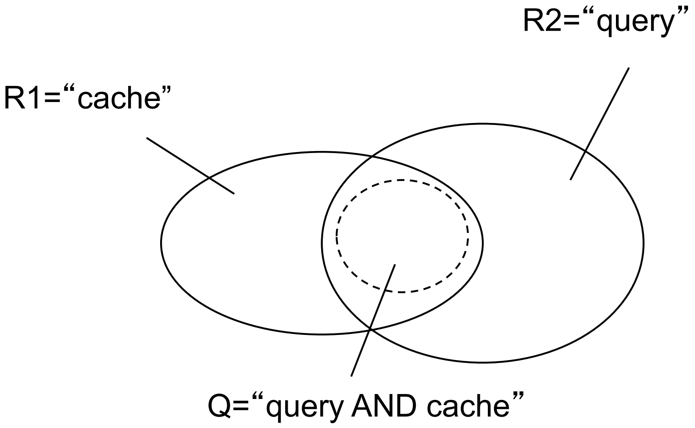
Using the semantic cache: Operational Cases

- Equivalence R=Q
- Query Containment Q⊂ R
- Region Containment R⊂Q
- Other intersection cases

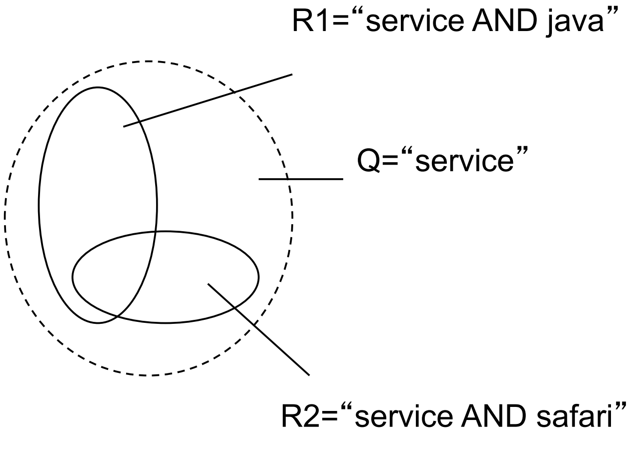
Equivalence



Query Containment



Region Containment



Multidatabase systems

- Systems allowing to query several databases
- Different issues
  - Heterogenity
  - Autonomy
  - Distribution

Large scale systems

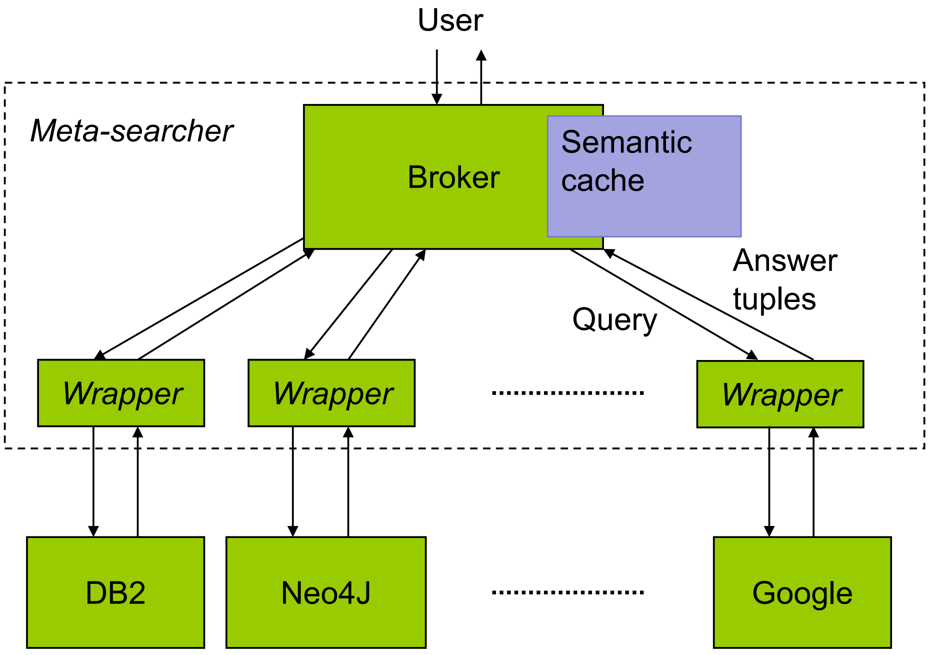
- Highly distributed data sources
- Autonomy + Heterogeneity
- WWW servers as data sources (HTML / XML / JSON), file systems, …
- Information retrieval and Database systems
- Issues on completeness and metadata availability

Databases and Information Retrieval

- The typical paradigm for querying *document collections* is to use information retrieval syle searches based on keyword matching and word position within documents

- The typical paradigm for querying databases is through an expressive, declarative query language that relies on database structures (schema)

Architecture



Other Caching approaches

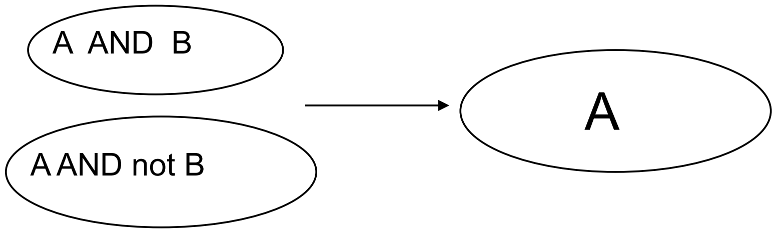
- Page caching
- Tuples caching

Source Heterogeneity

- Complete versus incomplete answers
  - sources send or do not send all relevant answers for a query
- *Checkable* versus non-checkable answers
  - tuples contain or do not contain the whole information used at the source to evaluate the query

Cache maintenance and replacement

- Example
  - LRU adopted to region size
  - avoid storing too large or too small regions
- Mechanisms for Coalescing
  - split and merge regions when necessary



Conclusion about semantic caching and other issues

- More powerful query language (e.g sparql)
  - query containment
  - more semantics by using ontologies

- Semantic caching applies to many distributed querying systems
  - Mobiles, P2P, grids, cloud…

- Prefetching