A Cache Management Strategy

CS 848: Course Project
Fall, 2006
University of Waterloo
Jeff Pound
Caching trade-off

- Storing too much data can be a waste of cache space, but can yield a high cache hit rate
- Not storing enough data can mean a low cache hit rate, but can be a better use of limited space
- We also have to address the problem of **when** a query can be answered from the cache
Proposed Strategy

- *Weaken* the query sent to the back-end DB to cache a *superset* of the results for the given query
Proposed Strategy

- *Weaken* the query sent to the back-end DB to cache a *superset* of the results for the given query
  
  - Do this in such a way that: we can avoid computing query containment, yet still have an easy way to determine when to use the cache and,
  
  - The extra cached data is semantically related to the query that caused the caching
Proposed Strategy

- Re-write queries (cache misses) with conjunctive predicates in the *where* clause as disjunctions (excluding join-conditions) and cache the results
  - This makes every predicate disjoint from every other predicate that built the cache (a notion of “domain completeness” on every key/value pair)
  - (The results specified by any one predicate are not further restricted by any other predicate)
Proposed Strategy

• Approximating query containment is now a matter of $n$ constant time table look-ups (where $n$ is the number of disjoint predicates in the where clause of the incoming query)
  
  - (we need to satisfy one predicate in each conjunctive clause, and every predicate in each disjunctive clause)

• No need to process probe queries at the cache

• Cache eviction can be done on a per-predicate basis (Evict predicate $p$ by deleting tuples matching $\forall q \ (q \in \text{predicateList}, p \neq q, p \land \neg q)$)
Proposed Project

• Implement and evaluate the specified cache management strategy
  – Compare against baseline strategies of full table caching, and query result caching
  – Measure comparative performance in various caching situations (cache miss, cache hit, eviction, etc...)
  – Cache hit rate is only meaningful full for real workloads, of which I don't have :(
Example

\[ \pi_{a,b}\left(\sigma_{A \land B \land C}(R)\right) \]

\[ \sigma_{A \lor B \lor C}(R) \rightarrow \text{Cache} \]

\[ R:\{A,B,C\} \]
Example

\[ \pi_{a,b}(\sigma_{A \land B \land C}(R)) \]

\[ \sigma_{A \lor B \lor C}(R) \]

\[ \pi_{...}(\sigma_{A}(R)) \quad \text{Hit} \]

Cache

\[ R:\{A,B,C\} \]
Example

\[ \pi_{a,b} (\sigma_{A \land B \land C} (R)) \]

\[ \sigma_{A \lor B \lor C} (R) \quad \text{Cache} \quad R: \{A, B, C\} \]

\[ \pi_{\ldots} (\sigma_{A} (R)) \quad \text{Hit} \]

\[ \pi_{\ldots} (\sigma_{A \land D} (R)) \quad \text{Hit} \]
Example

\[ \pi_{a, b} \left( \sigma_{A \land B \land C}(R) \right) \]

\[ \sigma_{A \lor B \lor C}(R) \]

Cache

R:{A,B,C}

\[ \pi \ldots \left( \sigma_{A}(R) \right) \quad \text{Hit} \]

\[ \pi \ldots \left( \sigma_{A \land D}(R) \right) \quad \text{Hit} \]

\[ \pi \ldots \left( \sigma_{A \land D \land \ldots \land \ldots}(R) \right) \quad \text{Hit} \]
Example

\[ \pi_{a,b} \left( \sigma_{A \land B \land C}(R) \right) \]

\[ \sigma_{A \lor B \lor C}(R) \]

Cache \( R: \{A,B,C\} \)

\[ \pi \ldots \left( \sigma_{A}(R) \right) \]
Hit

\[ \pi \ldots \left( \sigma_{A \land D}(R) \right) \]
Hit

\[ \pi \ldots \left( \sigma_{A \land D \land \ldots \land \ldots}(R) \right) \]
Hit

\[ \pi \ldots \left( \sigma_{A \lor D}(R) \right) \]
Miss

\[ \sigma_{A \lor D}(R) \]

Cache \( R: \{A,B,C,D\} \)
Questions/Comments?