Caching All Plans with Just One Optimizer Call

Debabrata Dash, Ioannis Alagiannis, Cristina Maier, Anastasia Ailamaki
Goal: A Physical Designer for PostgreSQL

- Automatically suggest indexes, views, partitions, etc.

- Essential in modern DBMSs
  - Provide improved performance
  - Reduce administration costs
  - All major commercial DBMSs provide one

- Not available on open source systems
  - Higher administration cost
Typical Physical Designer

Need a faster cost model to search through the candidates scalably.
INUM: A Fast Cost Model

Once the cache is built, the cost model is very fast!
INUM Cache structure

- Plans are indexed by *Interesting Orders (IntOrd)*
- Columns which help in query performance when sorted [Selinger79]
  - Grouping, joining, ordering columns
- Indexes *provide* IntOrds, plans require them
- Hundreds of these IntOrd combinations per query

Potentially hundreds of optimizer calls → High startup overhead
Requirement: INUM with fast startup

<table>
<thead>
<tr>
<th>INUM Assumptions</th>
<th>Application Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed selectivity</td>
<td>Data updates</td>
</tr>
<tr>
<td>Fixed set of queries</td>
<td>Online workloads</td>
</tr>
<tr>
<td>Fixed underlying structures (tables, partitions)</td>
<td>Automatic partition selection, view selection</td>
</tr>
</tbody>
</table>

On Violation $\rightarrow$ Rebuild cache

Needs fast cache rebuilding

Can we cache INUM’s plans more efficiently?
Yes $\rightarrow$ But need internal access to the optimizer
PINUM $\rightarrow$ Call the optimizer just once!!
Postgres Query Optimizer

- Contains many INUM plans – but not all
- Plans to reduce search space
- If a plan is pruned if it provides less interesting orders and costs more than another plan

PINUM tunes the pruning algorithm to preserve maximum number of optimal plans
PINUM Optimizer

- No pruning at all
  - Search space large $\rightarrow$ slow dynamic program
  - Overhead of transferring the plans
- Prune plans for same interesting orders
  - Set of returned plans is same as INUM
- Prune if smaller interesting order set has lower query cost
  - Plan1 – requires IntOrd (A,B), Costs 100
  - Plan2 – requires IntOrd (A), Costs 10
Preliminary Experiments

- PostgreSQL 8.3.7
- A synthetic star-schema workload
  - 10GB
  - 1 Fact table, 28 dimension tables
- 10 queries with varying number of joins
Experimental Results

- An order of magnitude faster for plan cache construction
- 5 times faster for index access cost construction
- Reduces the INUM cache size by a factor of 6
- 2.7% loss in accuracy on average compared to direct optimizer call
Conclusion

- PINUM reduces the cache building time by a factor of 5
  - By utilizing optimizer discarded plans
- Enables physical designers on dynamic workloads
- Part of a larger project on a physical designer for PostgreSQL
  - Please check out the demos at EDBT and SIGMOD
Related Work

- Configuration Parametric Optimization
  - Builds a “super plan” containing place holder for data accesses
  - Made for a top-down optimizer

- INUM is a better fit for PostgreSQL’s intermediate plans
- Requires minimal changes to the optimizer
PostgreSQL Join Planner