The Workload You Have, The Workload You Would Like

Matteo Golfarelli, Ettore Saltarelli
DEIS - University of Bologna - Italy

Outline

1. Introduction
2. Our approach
3. Profiling the workload
4. Generating the workload
5. Tests
6. Conclusions and future works

DW optimization

- Performance optimization in DWs is mainly achieved by carrying out view materialization and indexing.

- Most of the approaches in the literature rely on the existence of a reference workload that represent the target for the optimization.

- Real workloads are much larger than those that can be handled by these techniques, thus view materialization and indexing in real projects are tasks carried out “manually” by the designer.
The reference framework

- The gap between academic approaches and real systems could be filled by techniques capable of determining the workload characteristics while maintaining a reduced computational complexity.

The optimization process can be driven by a set of statistical indicators (the profile) that shows the best choices to the designer.

Basics

- Some of the indicators are based on the concept of cardinality of the aggregation pattern \( P \) associated to a given view \( V \).

- The cardinality of an aggregation pattern can be estimated using the Cardenas’ formula

\[
\text{Card}(P) = \Phi\left(|P|_{\text{max}} \cdot |P_0|\right)
\]

where \(|P_0|\) is the cardinality of the base fact table, while \(|P|_{\text{max}}\) is the maximum number of tuples feasible for the pattern

- The aggregation level of a given pattern \( P \) representing a query \( Q \) (or equivalently view) is computed as:

\[
\text{Agg}(P) = 1 - \frac{\text{Card}(P)}{|P_0|}
\]

- \( \text{Agg}(P) \) ranges in \([0..1]\)
  - \( 0 \) unaggregated pattern (i.e. the pattern of the base fact table)
  - \( 0.9999 \) completely aggregated pattern
The average aggregation level

- The aggregation level (AAL) of the full workload is then computed as:

\[ AAL = \frac{1}{n} \sum_{i=1}^{n} \text{Agg}(P_i) \]

where \( n \) is the number of queries in the workload.

- Workloads with high values of AAL will be efficiently optimized using views
  - Views determine a strong reduction of the number of tuples to be read
  - Their limited size allows a higher number of views to be materialized

Skewness

- Workloads with similar values of AAL can behave differently

- Materializing a single view to answer both the queries in the workload is much more useful for \( W_1 \) than for \( W_2 \) since in the first case the ancestor is very close to the queries and still coarse.

Given two queries with patterns \( P_1 \) and \( P_2 \), their ancestor is the most aggregated pattern \( P_1 \oplus P_2 \) on which both queries can be answered.
Skewness I

- Workloads with similar values of AAL can behave differently

\[ W_1 \]

\[ W_2 \]

- Materializing a single view to answer both the queries in the workload is much more useful for \( W_1 \) than for \( W_2 \) since in the first case the ancestor is very close to the queries and still coarse.

- The difference is captured by the distance between two patterns that is calculated as:

\[
\text{Dist}(P_i, P_j) = \text{Agg}(P_i) + \text{Agg}(P_j) - 2 \times \text{Agg}(P_i \oplus P_j)
\]

Skewness II

- The average skewness (ASK) of the full workload is calculated as:

\[
\text{ASK} = \frac{2}{n \cdot (n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{Dist}(P_i, P_j)
\]

- Workload with low values for ASK will be efficiently optimized using materialized views since the similarity of the query patterns makes it possible to materialize few views to optimize several queries
Selectivity Indicators

Profiling of selectivity is harder since the evaluation of the indicators must be based on the values of those of the aggregation ones.

The main indicator is the average selectivity (AS):

\[ AS = \frac{1}{n} \sum_{i=1}^{n} sel(Q_i) \]

Workload with low values for AS will require stronger use of indexes.

AS is not sufficient to characterize the impact of indexing with respect to materialization since it depends on \textit{where} and \textit{how} the statements are formulated.

Selectivity Indicators

\textbf{Where?} On aggregated or un-aggregated patterns?

- \( LSC \approx 0 \) selectivity is equally distributed
- \( LSC < 0 \) selectivity is stronger for queries on aggregated patterns
- \( LSC > 0 \) selectivity is stronger for queries on un-aggregated patterns
Selectivity Indicators III

- **How?** How many predicates are applied in the average on a query?

\[ ACT = \frac{1}{n} \sum_{i=1}^{n} m_i \]

- Where \( m_i \) is the number of constrained tables on query \( i \)

- Given a selectivity value \( AS \), workload with higher values of \( ACT \) will require a higher number of indexes to apply all the conditions profitably.

---

Workload Generation

- An algorithm for generating a GPSJ workload has been devised

  ✓ Testing: easily create large workloads
  ✓ Benchmarking: create workloads with specific characteristics

\[ F, R_{opt}, |W| \rightarrow \text{Generation of Patterns} \rightarrow P, F, R_{opt}, |W| \rightarrow \text{Generation of Selection criteria} \rightarrow \text{Success} \rightarrow W \]

- Given a desired profile \( R_{opt} \):
  - **Generation of pattern:** a taboo-search approach, navigating the multidimensional lattice, have been adopted.
  - **Generation of selection criteria:** selectivity is added exploiting the relationships between the generated patterns and the profile.
Test 20

Tests, carried out generating workloads based on the TPC-H/R benchmark, are aimed at evaluating the correlation between optimization and profile.

<table>
<thead>
<tr>
<th>Name</th>
<th>N. Queries</th>
<th>AAL</th>
<th>ASK</th>
<th>AS</th>
<th>LSC</th>
<th>ACT</th>
<th>Cand. Views</th>
<th>Mat. Views (2GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WKL1</td>
<td>20</td>
<td>0.835</td>
<td>0.348</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>97</td>
<td>15</td>
</tr>
<tr>
<td>WKL2</td>
<td>20</td>
<td>0.186</td>
<td>0.327</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>124</td>
<td>2</td>
</tr>
<tr>
<td>WKL3</td>
<td>20</td>
<td>0.790</td>
<td>0.810</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>596</td>
<td>15</td>
</tr>
<tr>
<td>WKL4</td>
<td>20</td>
<td>0.384</td>
<td>0.751</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>868</td>
<td>2</td>
</tr>
<tr>
<td>WKL5</td>
<td>30</td>
<td>0.884</td>
<td>0.316</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>14</td>
</tr>
<tr>
<td>WKL6</td>
<td>30</td>
<td>0.352</td>
<td>0.668</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>&gt;36158</td>
<td>???</td>
</tr>
</tbody>
</table>

![Graphs showing disk space constraint and number of material views vs. disk space constraint](image)

Test 20 II

The second test measures the effect of selectivity.

<table>
<thead>
<tr>
<th>Name</th>
<th>AAL</th>
<th>ASK</th>
<th>AS</th>
<th>LSC</th>
<th>ACT</th>
<th>View save</th>
<th>Index save</th>
<th>V-I space trade-off</th>
</tr>
</thead>
<tbody>
<tr>
<td>WKL1a</td>
<td>0.835</td>
<td>0.348</td>
<td>0.04</td>
<td>0</td>
<td>2</td>
<td>96.7 %</td>
<td>2.1 %</td>
<td>59% - 41%</td>
</tr>
<tr>
<td>WKL1b</td>
<td>0.835</td>
<td>0.348</td>
<td>0.25</td>
<td>0</td>
<td>2</td>
<td>88.2 %</td>
<td>7.8 %</td>
<td>84% - 16%</td>
</tr>
<tr>
<td>WKL1c</td>
<td>0.835</td>
<td>0.348</td>
<td>0.5</td>
<td>0</td>
<td>2</td>
<td>88.9 %</td>
<td>4.9 %</td>
<td>84% - 16%</td>
</tr>
<tr>
<td>WKL4a</td>
<td>0.384</td>
<td>0.751</td>
<td>0.04</td>
<td>0</td>
<td>2</td>
<td>27.3 %</td>
<td>52.6 %</td>
<td>77% - 23%</td>
</tr>
<tr>
<td>WKL4b</td>
<td>0.384</td>
<td>0.751</td>
<td>0.25</td>
<td>0</td>
<td>2</td>
<td>28.1 %</td>
<td>48.8 %</td>
<td>68% - 32%</td>
</tr>
<tr>
<td>WKL4c</td>
<td>0.384</td>
<td>0.751</td>
<td>0.5</td>
<td>0</td>
<td>2</td>
<td>22.0 %</td>
<td>29.9 %</td>
<td>67% - 33%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>AAL</th>
<th>ASK</th>
<th>AS</th>
<th>LSC</th>
<th>ACT</th>
<th>View save</th>
<th>Index save</th>
</tr>
</thead>
<tbody>
<tr>
<td>WKL7a</td>
<td>0.542</td>
<td>0.607</td>
<td>0.349</td>
<td>0.8</td>
<td>1</td>
<td>61.8 %</td>
<td>14.7 %</td>
</tr>
<tr>
<td>WKL7b</td>
<td>0.542</td>
<td>0.607</td>
<td>0.366</td>
<td>-0.8</td>
<td>1.1</td>
<td>54.3 %</td>
<td>0.26 %</td>
</tr>
<tr>
<td>WKL7c</td>
<td>0.542</td>
<td>0.607</td>
<td>0.3</td>
<td>0</td>
<td>1.2</td>
<td>25.9 %</td>
<td>62.9 %</td>
</tr>
<tr>
<td>WKL7d</td>
<td>0.542</td>
<td>0.607</td>
<td>0.29</td>
<td>0.1</td>
<td>2.8</td>
<td>18.0 %</td>
<td>62.2 %</td>
</tr>
</tbody>
</table>
Large workloads must be pre-reduced in order to apply optimization algorithms.

Pre-optimization (clustering) progressively affects the workload profile.

The profile has proved to be representative of the workload characteristics.

The workload generation algorithm is the first attempt in the database field to define large and ad-hoc workloads for benchmarking.

Transforming qualitative information in quantitative ones by means of functions that, based on the workload profile allow the execution cost to be estimated.