AutoAdmin “What-if”
Index Analysis Utility

Surajit Chaudhuri and Vivek Narasayya
Microsoft Research

Presentation: Atsushi Watase
Discussion: Lee Cuong

Outline
- What’s the goal
- Why hasn’t it been done before
- Overview of Architecture
- Interfaces for Hypothetical Configuration Simulation
- Implementing the Simulation interface
  - Why we can just use what the optimizer would see?
- Options for producing summary statistics on analysis data
- Example of a Session

What’s the goal
- Describe the index analysis utility
- Describe the interfaces for hypothetical configuration simulation
- Discuss the implementation for the interface

Why the problem hasn’t been addressed?
- Index selection is a hard search problem
  - Large number of possible single and multi-column indexes for enterprise class database
  - Hard to manage the search space efficiently

Why the problem hasn’t been addressed? (cont.)
- DBA needs to perform an impact analysis of the suggested indexes
  - Which recent queries and updates will slow down because of the change?
  - Which queries will benefit from the index and to what extent?

Overview of Architecture
- Term definitions:
  - Configuration: set of indexes, size of tables
  - Hypothetical configuration: may consist both existing (real) and hypothetical (what-if) indexes
  - Workload: set of SQL statements
Overview of Architecture (cont.)

- Hypothetical configuration analysis (HCA) engine supports two interfaces:
  - hypothetical configuration simulation
  - summary analysis on the data from the simulation, for example:
    - Workloads
    - Configurations
    - Performance

Interfaces for Hypothetical Configuration Simulation

- Define Workload
  - Frequency value is the number of copies of query
  - Associated with SQL text, set of tables

<table>
<thead>
<tr>
<th>Workload name</th>
<th>Query id</th>
<th>Frequency</th>
<th>Query properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload_A</td>
<td>1</td>
<td>1</td>
<td>=&lt;SQL text&gt;, (T1, T2)</td>
</tr>
</tbody>
</table>

- Define Configuration
  - Associated with set of existing and hypothetical indexes and configuration

- Set Database Size
  - Set the scaling values for each table

<table>
<thead>
<tr>
<th>Tables (T1, T2)</th>
<th>Ind_A</th>
<th>Ind_B</th>
<th>Ind_D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaling value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Num rows</td>
<td>100,000</td>
<td>500,000</td>
<td>600,000</td>
</tr>
<tr>
<td>Num pages</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Cost</td>
<td>0.02</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>Index statistics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Histogram</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Interfaces for Hypothetical Configuration Simulation (cont.)

- Estimate Configuration
  - The unit of cost is relative to the total cost of the workload
  - Indexes_Used attribute represents the indexes expected to be used with the hypothetical configuration

<table>
<thead>
<tr>
<th>Configuration name</th>
<th>Query ID</th>
<th>Cost</th>
<th>Indexes_Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>New_Config</td>
<td>1</td>
<td>0.02</td>
<td>Ind_A, Ind_D</td>
</tr>
<tr>
<td>New_Config</td>
<td>2</td>
<td>0.11</td>
<td>Ind_B</td>
</tr>
</tbody>
</table>
Discussion #1

- Q1: What other aspects can you imagine being automatically tuned like this?
- Q2: Is this 'automatic enough" or does there still need to be more automation to be really useful?

Discussion #2

- Are there still spaces for improvement within in the paper?
  - Are the testing workload & testing settings are well-defined and reasonable?
  - May some historical workload be used?
  - Would it make sense to investigate an efficient way of defining the testing workload using statistical method?
  - Can we make the cost estimation more reliable?
  - A new parsing language is defined to serve as the interfaces for hypothetical configuration simulation. Is this really necessary?
  - What if the database grows much more complex?

Implementing the Simulation interface

- Creation of hypothetical indexes
  - Sampling strategy: adaptive page-level sampling algorithm
- Defining a hypothetical configuration
- Obtaining optimizer estimates

Why we can just use what the optimizer would see?

- An optimizer’s decision on whether or not use an index is solely based on the statistical information
- The statistical measures can be efficiently gathered via sampling
- Optimizer can consider the hypothetical index for plan generation with collected statistics

Options for producing summary statistics on analysis data

- Allow the DBA and other tools to directly use the SQL interface
  - Big performance drawback
- Provide a set of "canned" queries that support a set of predetermined summary analyses
  - Doesn’t provide an extensible framework for new summary statistics
Options for producing summary statistics on analysis data (cont.)

- An interface retains the flexibility of formulating ad-hoc requests, without overhead of manually generating complex SQL queries

Example of a Session

- Analyzing the workload
  - SQL type, most expensive queries, distribution of conditions on tables
- Analyzing the current configuration
  - Request to see all indexes on tables
- Exploring "what-if" scenarios
  - Compare the cost of workload and configuration between current and proposed configuration

Conclusion

- Index analysis utility can help the DBA to select indexes
- Hypothetical configuration analysis can be used to conduct analysis studies
- Implementation of the hypothetical configuration analysis can be possible with the necessary server extensions

Discussion #2

- Are there still spaces for improvement within in the paper?
  - Are the testing workload & testing settings are well-defined and reasonable?
  - May some historical workload be used?
  - Would it make sense to investigate an efficient way of defining the testing workload using statistical method?
  - Can we make the cost estimation more reliable?
  - A new parsing language is defined to serve as the interfaces for hypothetical configuration simulation. Is this really necessary?
  - What if the database grows much more complex?