

Materialized View and Index Selection Tool for Microsoft SQL Server 2000

Sanjay Agrawal
Microsoft Research
One Microsoft Way,
Redmond, WA 98052
+1-(425) 705-3507
sagrawal@microsoft.com

Surajit Chaudhuri
Microsoft Research
One Microsoft Way,
Redmond, WA 98052
+1-(425) 703-1938
surajitc@microsoft.com

Vivek Narasayya
Microsoft Research
One Microsoft Way,
Redmond, WA 98052
+1-(425) 703-2616
viveknar@microsoft.com

1. Introduction

Microsoft SQL Server 2000 supports creation and use of materialized views. In data intensive applications such as decision support and data warehousing picking the right set of materialized views and indexes for a given database and workload becomes crucial for performance. In the context of the AutoAdmin project at Microsoft Research [AA-MSR], we have developed a tool called the *Tuning Wizard* for automating this important task. Our tool not only recommends materialized views and indexes, but also considers tradeoffs between indexes and materialized views for the given workload. This tool ships with Microsoft SQL Server 2000.

2. Architecture of the Tuning Wizard

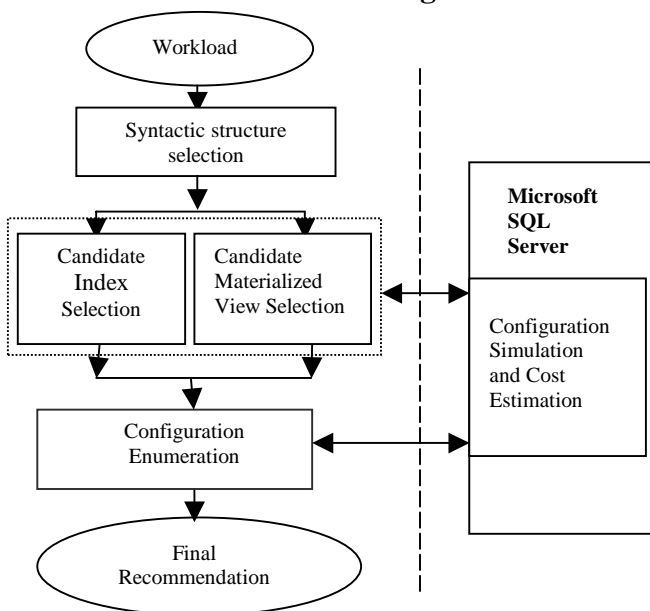


Figure 1. Architecture of Index and Materialized View Selection Tool

An architectural overview of the Tuning Wizard is shown in Figure 1. The input to the tool is a database and a workload consisting of a set of SQL queries. A representative workload can be obtained by using the Microsoft SQL Server Profiler tool that logs events at the server or by using customer or organization

Permission to make digital or hard copies of part or all of this work or personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGMOD 2001 May 21-24, Santa Barbara, California USA
Copyright 2001 ACM 1-58113-332-4/01/05...\$5.00

specific benchmarks. The tool recommends a set of *indexes*, *materialized views* (referred to as *indexed views* in Microsoft SQL Server 2000) and *indexes on materialized views* appropriate for the given workload. The tool, whose algorithms and implementation are described in detail in [ACN00, CN97] interacts with Microsoft SQL Server to (a) simulate indexes and materialized views that do not exist in the current database (referred to as *hypothetical* indexes and materialized views); (b) request the optimizer to optimize a query for a given hypothetical *configuration* (a set of indexes, materialized views and indexes on materialized views). Techniques for efficient implementation of the above functionality in a DBMS are described in detail in [CN98]. The first step in our architecture is to identify indexes and materialized views that are *syntactically relevant* for the given workload in an efficient manner. The goal of the *candidate selection* module is to quickly obtain (in a cost-based manner), a much smaller set of indexes and materialized views compared to the syntactically relevant set, that are worthy of further exploration. Finally, the *configuration enumeration* module is responsible for efficiently searching through the space of candidates to produce the final recommendation while judiciously trading-off the benefits of indexes and materialized views for the workload. The tool also allows users to study the estimated change in cost of the workload and changes in usage of indexes and materialized views for the recommended configuration by quantitatively comparing it with the existing database configuration. Further details about the Tuning Wizard for Microsoft SQL Server 2000 can be found in [ACKN00].

3. What will be demonstrated?

We will show the following aspects of the index selection tool: (a) How to gather a workload using Microsoft SQL Server Profiler. (b) How the tool picks materialized views and indexes that are appropriate for the given workload. (c) How the tool augments its recommendations with a rich set of analysis reports that helps the DBA understand the impact of the proposed recommendations.

4. References

- [AA-MSR] <http://research.microsoft.com/dmx/AutoAdmin>
- [ACKN00] Index Tuning Wizard for Microsoft SQL Server 2000 <http://msdn.microsoft.com/library/techart/itwforsql.htm>
- [ACN00] Agrawal S., Chaudhuri, S., Narasayya V., "Automated Selection of Materialized Views and Indexes for SQL Databases." Proc. of the 26th VLDB Conference Cairo, Egypt, 2000.
- [CN97] Chaudhuri, S., Narasayya V., "An Efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server." Proc. of the 23rd VLDB Conference Athens, Greece, 1997.
- [CN98] Chaudhuri, S., Narasayya V., "AutoAdmin "What-If" Index Analysis Utility". Proc. of ACM SIGMOD, Seattle, 1998.