

# Microsoft Index Tuning Wizard for SQL Server 7.0

Surajit Chaudhuri  
Microsoft Research  
surajitc@microsoft.com

Vivek Narasayya  
Microsoft Research  
viveknar@microsoft.com

## 1. Introduction

Enterprise-class databases require database administrators who are responsible for performance tuning. With large-scale deployment of databases, minimizing database administration function becomes important. One important task of a database administrator is selecting indexes that are appropriate for the workload on the system. In data intensive applications such as decision support and data warehousing picking the right set of indexes becomes crucial for performance. Moreover, the indexes chosen should track changes in the workload. While automating the process of index selection can greatly reduce administration cost, enterprise databases are simply too complex for the administrator to hit the “accept” button on the recommendations of an index selection tool without doing a quantitative impact analysis of the recommendations.

We have developed an index tuning wizard at Microsoft Research that (a) automates the task of picking the right set of indexes for a database and workload, and (b) supports the ability to perform impact analysis of index recommendations. We will demonstrate this tool running on Microsoft SQL Server 7.0 (Sphinx).

## 2. Functionality of the Index Tuning Wizard

### 2.1 Input and Output

The index tuning wizard takes as input a *workload* and recommends a set of indexes *for the given workload*. A workload may consist of one or more SQL statements (i.e. SELECT, UPDATE, INSERT, DELETE). Therefore, when the tool starts up it requests the user to provide a workload file as input. Such a workload file can be generated by logging activity on the database server over a period of time using a utility like SQL Server Profiler. A workload file can also be “hand-generated” by typing in a set of queries (separated by the delimiter GO) into a .sql file. Since the indexes recommended by the tool are based on the workload information, it is important that the workload information be representative of the true or expected workload on the system. The tool reports the expected change in performance of the workload and the expected change in storage cost if the recommended indexes are created. In addition, the user can perform several quantitative impact analyses of the proposed indexes. For example, the user can compare the cost of an individual query in the workload for the current set of indexes with the cost of the query for the proposed set of indexes. Another

example of analysis is computing the *expected index usage* (percentage of queries in which an index is used) with the proposed set of indexes. If the user accepts the recommendation, the index creation/alteration step can be initiated immediately or can be scheduled at a specific date and time.

### 2.2 Customization Options

The index tuning wizard can be customized in several ways depending on the requirements of the user. One way to customize the tool is to specify which tables should be considered for indexing. This allows the user to focus the design on selected tables in the database without altering the indexes for the remaining tables. Another parameter that can be controlled by the user is the total amount of storage consumed by indexes recommended by the tool. The tool picks a set of indexes such that this storage constraint is not violated. Often, when tuning a database, the administrator may not want to change the existing set of indexes, but only consider adding new indexes. The index tuning wizard can be customized to retain all existing indexes.

### 2.3 Launching the Tool

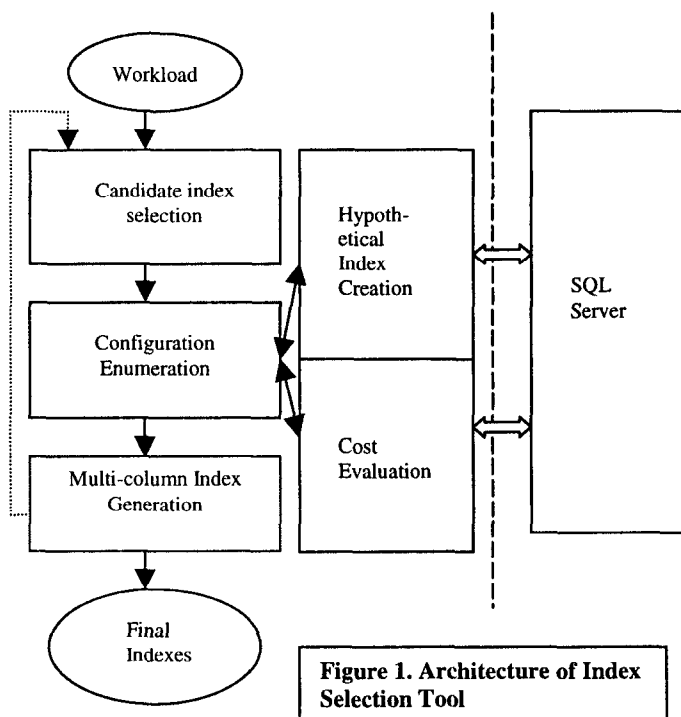
The index tuning wizard is a client tool of Microsoft SQL Server. In SQL Server 7.0, the index tuning wizard can be activated in several ways. It can be launched from the Microsoft Management Console (MMC) or SQL Server Profiler by selecting the *Index Tuning Wizard*. When launched in this mode, the tool requests the user for a workload file for which to tune the database. The tool can also be run from the SQL Server Query Analyzer, by selecting a query on the screen and choosing *Perform Index Analysis* under the Query menu.

## 3. Architectural Overview

Figure 1 given an architectural overview of the index selection tool. The input to the tool is a workload (a set of SQL queries). As described earlier, a representative workload for the system may be gathered using a utility such as SQL Server Profiler that logs relevant events at the server. The tool recommends a set of indexes appropriate for the given workload. The tool, described in detail in [CN97], interacts with SQL Server in two important ways. First it needs to gather statistics on indexes by creating hypothetical indexes. Second, it needs to tell the optimizer to optimize queries in the workload for a hypothetical configuration (a set of indexes). The index selection tool is robust and “industrial-strength” since it deals with complex index structures

such as multi-column indexes and the fact that modern query optimizers use advanced indexing techniques such as index intersection and index-only access. Candidate index selection, configuration enumeration and multi-column index generation together provide a fast search technique to cut through the complexity of enumerating the space of possible index configurations. The details of the algorithms underlying the tool can be found in [1].

The tool also allows users to perform a quantitative impact analysis by comparing the current and proposed index configurations. The analysis is made possible by the hypothetical configuration analysis (HCA) engine [2]. The HCA engine supports the ability to analyze: (a) a workload, (b) a hypothetical configuration and (c) the cost and index usage of a hypothetical configuration for a given workload. The HCA engine provides a query language using which a variety of “what-if” scenarios can be explored by the user. Some examples of these analyses are (a) the estimated change in cost of the workload for the recommended configuration as compared to the current (existing) database configuration, (b) the index usage of each index in the current configuration, i.e. the fraction of queries where the index was used.



#### 4. What will be demonstrated?

Permission to make digital or hard copies of all or part of this work for 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. SIGMOD '98 Seattle, WA, USA © 1998 ACM 0-89791-995-5/98/006...\$5.00

We will demonstrate the following aspects of the index tuning wizard:

- **How to gather a workload.** We will show how a representative workload can be gathered for database using Microsoft SQL Server Profiler.
- **Adapting to variations in workload.** We will show how the tool can adapt to variations in the workload. We will run the index tuning wizard with two workloads: one query intensive and the other update intensive, and show how the tool picks indexes appropriate to each workload.
- **Customizing the tuning wizard.** We will demonstrate how the index tuning wizard can be customized to focus its analysis on a specified set of tables. We will also show how the tool adapts to a constraint on the available storage space for indexes.
- **Recommending indexes for a single query.** We will show how the SQL Server Query Analyzer uses the index tuning wizard to pick the right set of indexes for a single query. When launched in this mode, the tool always suggests indexes *in addition* to the existing indexes, except on tables that are updated in the query.
- **Performing quantitative impact analysis.** We will demonstrate how the analysis component of the tool can be used to perform a quantitative impact analysis of the proposed index recommendations. We will perform the following analyses: (a) Compare the cost of an individual query for the current set of indexes with the cost of the query for the proposed set of indexes. (b) Show index usage (percentage of queries in which an index is used) for the current and proposed set of indexes. (c) Show a break-down of the cost of the workload over tables in the database for the current and proposed set of indexes.

#### References

[1] Chaudhuri, S., Narasayya V., "An Efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server." Proceedings of the 23rd VLDB Conference Athens, Greece, 1997, pages 146-155.  
 [2] Chaudhuri, S., Narasayya V., "AutoAdmin "What-If" Index Analysis Utility." In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1998. (This Proceedings).

#### Acknowledgments

We would like to acknowledge the help of the SQL Server team, who worked closely with us in developing the index tuning wizard.