

Auto Admin “What-if” Index

Analysis Utility

Surajit Chaudhuri

Vivek Narasayya

Presented by: Immad Naseer

Discussion by: Pooyan Fazli

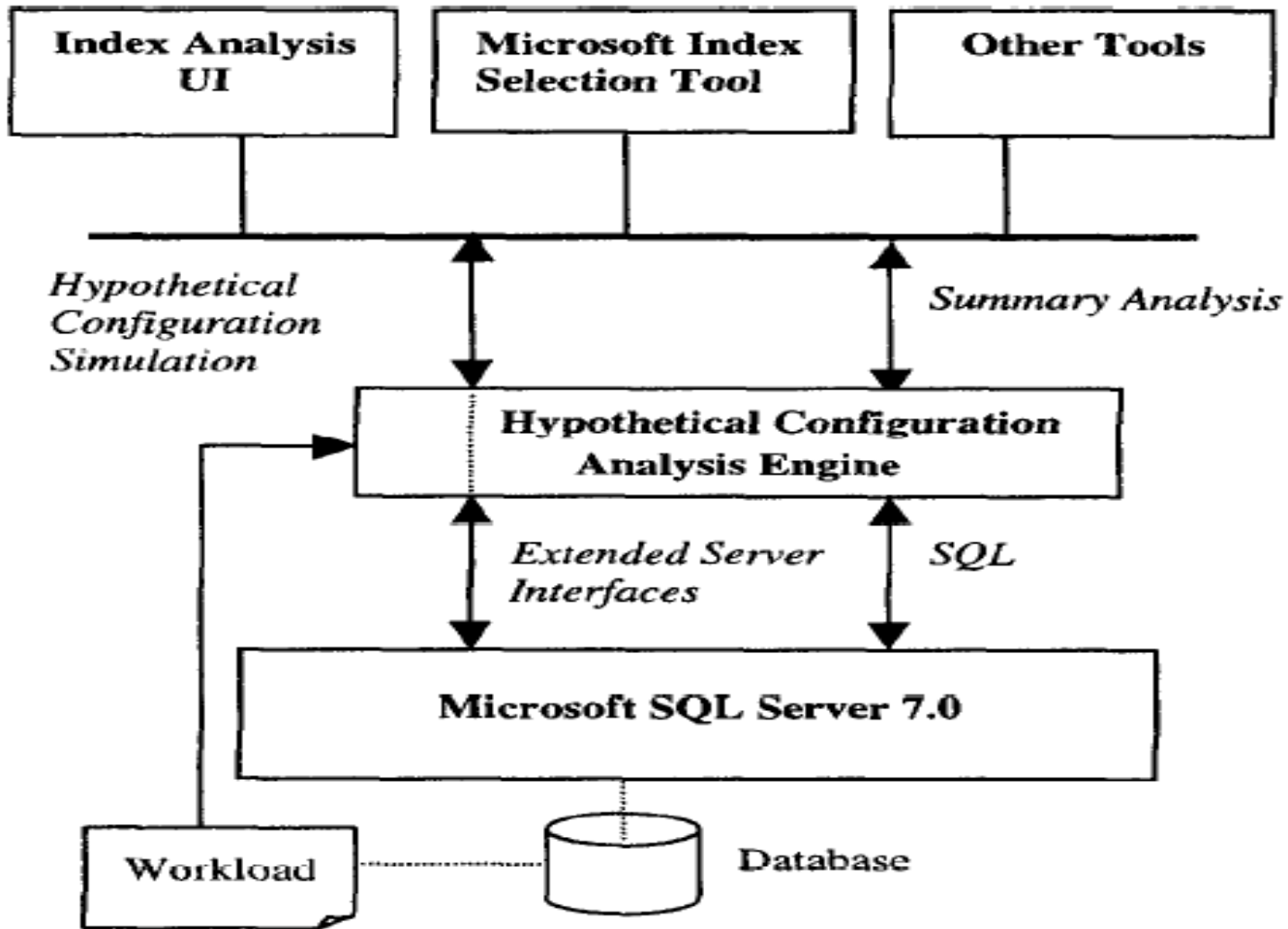
Motivation

- Automatic index selection is a difficult problem
- Enterprise class databases are too complex for the administrator to accept computer recommendations without impact analysis
- An impact analysis can also be used by an automatic index selection tool

Motivation

- Ability to perform quantitative impact analysis using
 - Existing indices
 - Hypothetical indices
 - Different workloads
 - Scaling of tables

Architecture



Some terminology

- Configuration: Set of indices and size of each table in the database.
- Hypothetical configuration: Real and/or hypothetical indices and size of each table in the database.
- Workload: A set of SQL statements

Discussion

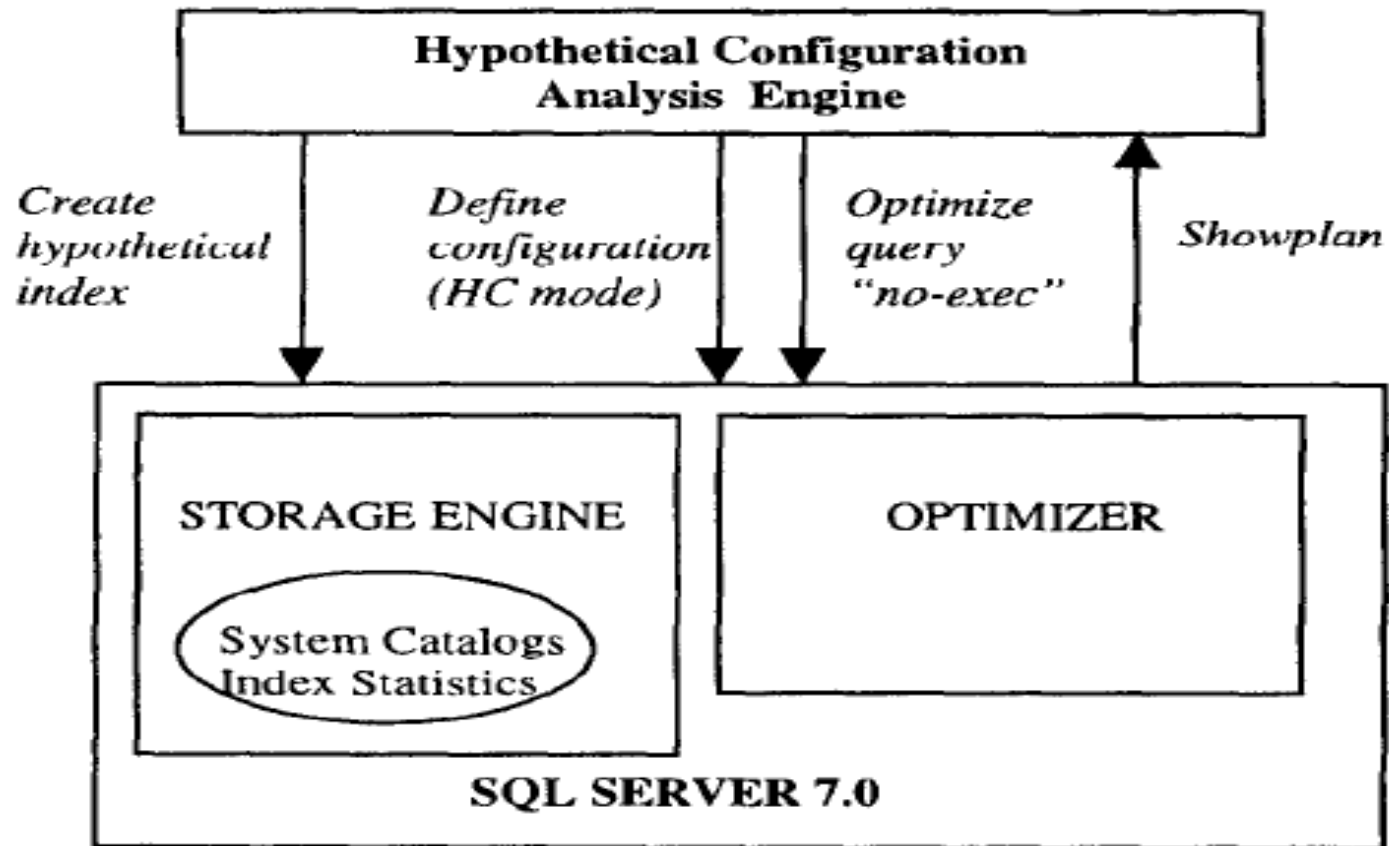
- What other aspects can you imagine being automatically tuned like this?

- The hypothetical configuration analysis (HCA) engine performs two main tasks
 - Simulating a hypothetical configuration
 - Summary analysis on data resulting from simulation

Simulating the hypothetical configuration - Interfaces

- Define a workload
- Define configuration
- Define size
- Estimate configuration
- Remove

Simulating the hypothetical configuration



Simulating the hypothetical configuration

- First of all, we have to define the new indices and gather statistics on them from the tables
- The authors use an adaptive page level sampling for gathering statistics
- It is shown that this reduces the running time and doesn't introduce a large error

Simulating the hypothetical configuration

- The queries are executed in a “no-exec” mode in which they are simply optimized by the optimizer
- This makes sense because ultimately it is optimizer’s estimates which determine whether an index would be used or not

Simulating the hypothetical configuration

- Since normal database operations should not be disturbed, the configuration information is passed to the server in a special “HC” mode

Simulating the hypothetical configuration

- Summary analysis data is optionally saved in user defined tables instead of system catalogs

Summary Analysis

- We have three options
 - Summary statistics can be generated using ad-hoc SQL queries
 - A set of “canned” queries can be provided to the user
 - An interface with some of the flexibility of ad-hoc queries and without the overhead of complex SQL queries

Summary Analysis

- Queries are made over a set of objects with associated structural properties
- The properties may be atomic, such as query type, number of tables, or sets/lists, such as tables referred to in a query etc
- There are 3 distinct kind of objects, namely, workload, configuration and cost_usage objects

Summary Analysis – An example

The administrator starts with a workload of queries run over the last week

He orders the queries by cost

Analyze WORKLOAD WITH

lst_week_wkd TOP 25 BY Cost

Summary Analysis – An example

Then he decides to see the distribution of conditions on queries

```
Analyze WORKLOAD WITH 1st_week_wkd  
SUMMARIZE USING Count BY  
conditions_in_query
```

Summary Analysis – An example

The administrator then sees the indexes defined on the table T2.

```
Analyze CONFIGURATION WITH  
current_conf BY indexes  
WHERE table.name = T2
```

Summary Analysis – An example

The administrator defines a new non-clustered index B and compares costs of the two workloads.

```
Analyze COST-USAGE WITH lst_week_wkd,  
(current_config, proposed_config) SUMMARIZE  
USING Sum BY Cost Where Tables SUPERSET-  
OF T2.
```

Discussion

- Is this automatic enough or does there still need to be more automation to be really useful?

Conclusion

- AutoAdmin allows the administrator to conduct impact analysis on real and/or hypothetical configurations of the database
- It is designed to run efficiently under normal system workload
- It presents a custom query interface for summary analysis of the data

Thank you