

Database Tuning

19

- A major problem in making a database run fast is deciding which indexes to create.
- Pro: An index speeds up queries that can use it.
- Con: An index slows down all modifications on its relation because the index must be modified too.

Example: Tuning

20

- Suppose the only things we did with our beers database was:
 1. Insert new facts into a relation (10%).
 2. Find the price of a given beer at a given bar (90%).
- Then SellInd on Sells(bar, beer) would be wonderful, but BeerInd on Beers(manf) would be harmful.

Tuning Advisors

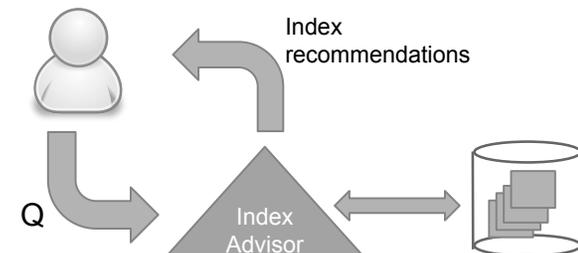
21

- A major research area
 - Because hand tuning is so hard.
- An advisor gets a *query load*, e.g.:
 1. Choose random queries from the history of queries run on the database, or
 2. Designer provides a sample workload.

Tuning Advisors --- (2)

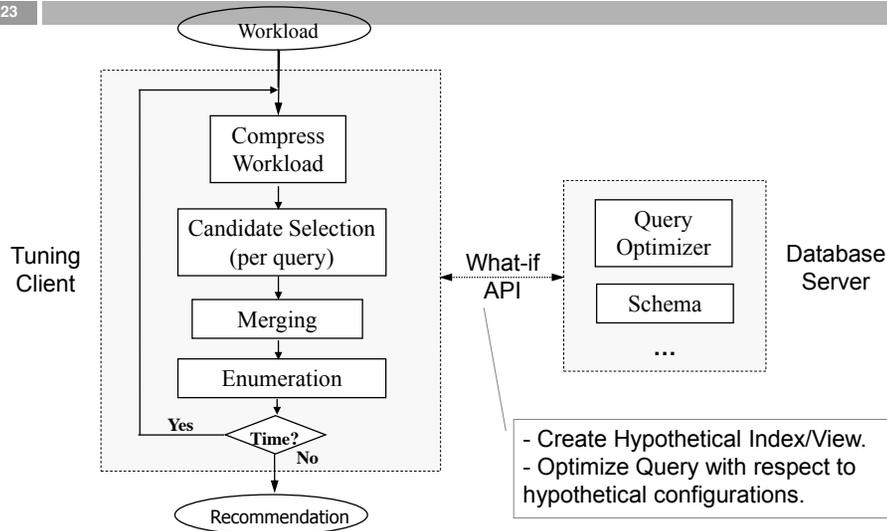
22

- The advisor generates candidate indexes and evaluates each on the workload.
 - Measure the improvement/degradation in the average running time of the queries.



Example: Database Tuning Architecture

23



Credit: Microsoft

Summary

24

- ❑ Many alternative file organizations exist, each appropriate in some situation.
- ❑ If selection queries are frequent, sorting the file or building an *index* is important.
 - Hash-based indexes only good for equality search.
 - Tree-based indexes best for range search; also good for equality search.

Summary (cont'd)

25

- ❑ Can have several indexes on a given file of data records, each with a different search key.
- ❑ Understanding the nature of the *workload* for the application
 - ❑ What are the important queries and updates? What attributes/relations are involved?
- √ Indexes must be chosen to speed up important queries (and perhaps some updates!).
 - Index maintenance overhead on updates to key fields.
 - Choose indexes that can help many queries, if possible.