Comparison of three different data mining and non-data mining auto-indexing algorithms for efficient indexing on Oracle’s database system

**Authors:**

Charlie Liu - charlie.liu-1@ou.edu

Dragon Tran - dragon.tran@ou.edu

Emily Black - emily@ou.edu

Charles Johnson - charles.t.johnson@ou.edu

**Objectives:**

For this project, we intend to select, implement and compare auto indexing algorithms on the Oracle SQL database platform, and perform efficiency analysis by comparing metrics such as I/O utilization, CPU usage, response time, etc. In addition, we will compare auto indexing algorithms that use data mining for index selection versus auto indexing algorithms without using data mining approaches to assess their differences in performance.

This project is intended to overall accomplish comparing and finding the most efficient auto-indexing algorithm for Oracle’s platform given three auto-indexing algorithms. Our goal is to measure and find certain metrics that excel in certain areas, which would include things like: decreased disk IO usage, decreased CPU usage, quicker response time, etc. Oracle also has a built-in real time monitoring system that keeps track of certain metrics that we’ll need when comparing each algorithm to each other.

**Significance of the Project:**

The significance of this project positively impacts Database Administrators (DBAs) and their daily work, along with improved performance of the database (DB) itself. The purpose of a DBA is to administer and monitor the DB and its performance. “*DBAs are also required to guess when a tuning session is needed, and when to deploy recommendations. Naturally, this is not a one-time process, but instead DBAs continuously monitor, diagnose, and re-tune database installations*“ [Bruno, 2007]. DBAs can run into issues though, because some tables in a DBS might be thousands upon millions of rows long. This means that “*the number of index sets that must be evaluated to find the optimal configuration is very large.*“ [Chaudhuri, 1997]. Just this alone can quite easily overwhelm any novice or expert DBA. This is where the importance of auto-indexing and significance of our project comes into play.

As a group, we’re looking into online tuning, compared to offline tuning. Online tuning only locks tables when moving to a new index after it has been built, whereas offline tuning completely locks the table of any read/write calls while it rebuilds and moves to a new index. The main idea of onlining tuning is “*to monitor the query load to identify locally dominant patterns (vs. the globally dominant patterns identified by off-line tuning), and automatically adjust the physical configuration to maximize query performance*“ [Schnaitter, 2007]. This is remarkably significant, because we want to have the most optimal configuration of indices, while at the same time having the least amount of down time (read/write) on tables. Using online tuning also reduces DBA interference with their job, which is also very significant.

One more significance is that “*Autonomous index tuning provides a dynamic solution for the well-studied Index Selection Problem (ISP)*“ [Luhring, 2007]. To briefly define it, the Index Selection Problem (ISP) consists of M queries and N index candidates, with associated costs and sizes. The ISP essentially wants to find “*indexes which optimizes the overall execution time by maximizing the profit*“ [Luhring, 2007]. This project carries great significance when finding solutions to problems like ISP and other similar index related problems.

**Research Methodology:**

For this project, we will select and implement two non-clustering algorithms and one clustering algorithm for auto-indexing. To create a controlled testing environment, three copies of a database using the same data and queries will be created in Oracle, and maintained using the different algorithms. Performance metrics of the databases will be collected using Oracle SQL Developer’s built in Real Time SQL Monitor. Other considerations for controlled variable will be the hardware of the machine connecting to SQL, time of access and network stability.

Over the course of the semester, we will increase the amount of data inside our databases, as well as perform/creating additional queries to observe the effect of increasing database volume and complexity on the performance of different algorithms.

**Time Table (tentative):**

Research: 2/6/18 - 3/8/2018

We will be researching different auto-indexing algorithms to determine which ones will be appropriate to use for our comparisons. In addition, we will also be researching the proper protocols for monitoring performance and identifying key metrics. Afterwards, we will decide on 3 auto-indexing algorithms based on factors such as performance overhead, compatibility with Oracle, and etc.

Progress Report 1: 3/8/2018

Implementation: 3/8/2018 - 3/29/2018

We will be implementing the 3 selected auto-indexing algorithms on to three identical Oracle databases. We are considering writing a java program using jdbc that will connect to all 3 databases to populate and query the databases with the same data.

Progress Report 2: 3/29/2018

Performance Monitoring: 3/29/2018 - 4/12/2018

During this phase, we will be executing varying amounts of queries to simulate a typical database application. Throughout the phase, we will be monitoring the performance of the auto-indexing algorithms implementations.

Analysis: 4/12/2018 - 4/26/2018

We should be finalizing our performance tests and comparisons in the first week. Afterwards, we will compare and analyze the metrics gathered throughout the project. Then, we shall summarize and report our findings and conclusions on which auto-indexing algorithm implementation is more optimal.

Final Report: 5/1/2018

Final Presentation: 5/3/2018

**References (APA):**

Bruno, N., & Chaudhuri, S. (2007, April). An online approach to physical design tuning. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (pp. 826-835). IEEE.

Chaudhuri, S., & Narasayya, V. R. (1997, August). An efficient, cost-driven index selection tool for Microsoft SQL server. In *VLDB* (Vol. 97, pp. 146-155).

Luhring, M., Sattler, K. U., Schmidt, K., & Schallehn, E. (2007, April). Autonomous management of soft indexes. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on* (pp. 450-458). IEEE.

Schnaitter, K., Abiteboul, S., Milo, T., & Polyzotis, N. (2007, April). On-line index selection for shifting workloads. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop* (pp. 459-468). IEEE Computer Society.