Self-Tuning Database Systems

ISSDM Summer Mini-Grant Proposal

PI: Neoklis Polyzotis <alkis@cs.ucsc.edu>

1 Project Summary

It is common knowledge that the modern world generates and stores information at an increasing rate. A typical example is the emerging field of e-science, where the adoption of high-performance computing has led to the generation of vast volumes of scientific data. The usefulness of the stored information, of course, depends crucially on the ability to query it effectively, and hence on the use of state of the art data management technology.

Unfortunately, despite the growing trends of data collection, the adoption of database systems in e-science has been less than spectacular and has often lead to a lot of frustration. One important reason can be traced to the complexity of managing and tuning a database system. Essentially, the performance of a database system depends crucially on its physical schema, that is, the set of physical structures, such as indices and materialized views, that can speed up the execution of queries. In order to be effective, of course, the physical schema has to match the traits of the workload, e.g., it must index more heavily the parts of the database that are frequently referenced in queries. At the same time, the schema is typically constrained by a total disk space budget for its storage. Thus, designing an effective physical schema involves a non-trivial optimization problem: maximize the throughput of the query processor, assuming limited resources for storing the materialized structures. Clearly, a scientist (or, for that matter, any non-expert user) is not likely to possess the expertise that is necessary in order to tackle this challenging problem. Hiring a database administrator is not a particularly attractive solution either, since this increases the total cost of ownership.

To address this important issue, we propose to develop a database system that can self-organize its physical configuration without the intervention of a human administrator. The key idea is to augment the system with an on-line tuning module that monitors continuously the incoming workload, gathers statistics on the performance of the system, and reorganizes the physical design periodically in order to maximize query throughput. Of course, realizing this goal involves several challenging problems. More precisely, since tuning happens concurrently with normal database operation, the tuner has to operate with low and controllable overhead so that it does not affect query performance. Moreover, the tuning module has to track continuously the traits of the workload and identify potential physical designs that can improve performance. Even more importantly, the tuner has to determine whether a recent change in the workload constitutes a major shift in the query distribution that justifies the potentially high cost of changing the physical design. These problems stem from the on-line nature of the problem and have not been addressed by previous studies on database tuning. Thus, the proposed research will make significant scientific contributions in the area of self-tuning systems and autonomic computing. Furthermore, it will lower the barrier for the deployment of database systems and hence facilitate the adoption of database technology in several "data-intensive" disciplines.

The proposed project forms part of our more general agenda on the development of database systems that target specifically non-expert users, and in particular scientists. We believe that this user base forms a formidable challenge for database researchers and a chance to bring database systems to the "masses". In this direction, we have been working in parallel on two systems: the Data Ring [1, 2], a peer-to-peer middleware system that supports declarative complex queries over massively distributed data, and Chameleon, a DBMS that allows in-situ querying for data residing in a file system. The notion of self-organization is prevalent in both systems, since the goal is to enable non-expert users to enjoy the benefits of database technology without the inconvenience of system administration.

2 Project Description

As mentioned previously, we propose to do fundamental research in the development of self-organizing database systems. In particular, we propose to investigate techniques that enable a database system to configure automatically its physical schema, in an on-line fashion. In what follows, we briefly discuss the limitations of existing techniques and then outline the salient elements of the proposed research.

Previous work on database tuning. The design of an effective physical schema is one of the most important problems in the deployment of a database system. To assist administrators in this task, earlier studies [3, 5, 24] have introduced techniques that analyze a representative workload and automatically generate a recommended physical configuration. This paradigm is typically referred to as *off-line tuning*, since the workload is gathered and analyzed before the database system goes live. The use of a representative workload implies that off-line tuning is suitable for application domains where the query load is rather stable, that is, there are certain query characteristics that are predominant. This may not be the case in scientific data management, where queries can result from interactive data analysis and their characteristics can thus be highly dependent on the intentions and goals of the analyst.

Previous studies have acknowledged the shortcomings of off-line tuning and have advocated the use of on-line techniques that can adapt to the latest trends of the query workload. More concretely, some past studies have developed rudimentary on-line tools for index selection in relational databases [9, 8]. In the past year, more sophisticated solutions to this problem have been proposed [18, 4], thus indicating the growing need for on-line physical design tuning. These recent techniques provide interesting insights into the problem of selecting indices on-line with limited overhead. However, it is also important to spend the limited overhead computation as wisely as possible. This entails the decision of *when* and *how* to allocate the resources needed to evaluate candidate configurations. Previous work has not addressed this question.

Proposed research. The proposed research will advance the state-of-the-art in the following ways. First, in contrast with existing approaches, we plan to develop techniques that have controllable overhead. This is particularly important, since the overhead of the on-line tuning process can affect negatively the performance of query evaluation. Second, we plan to develop a general tuning framework that can be applied to a wide range of data models, and not just the relational model as is the case for existing techniques. This feature is essential in the diverse setting of scientific data management, where information can be managed in a variety of native data models, such as, relational, semi-structured, or even plain text files.

The following paragraphs outline some preliminary results on our current work on techniques for on-line tuning. We conclude with our plans for extending our work to the problems described previously.

Completed Work: A framework for on-line index selection. As a first step in our proposed research for on-line tuning, we have developed the COLT [21] framework (short for *Continuous On-Line Tuning*) that supports the on-line selection of index structures based on the current workload. COLT is realized as a separate system module that operates in parallel to the main query processing pipeline. COLT builds a model of the current workload based on the incoming flow of queries, estimates the respective gains of different candidate indices, and selects those that would provide the best performance for the observed workload within the available space constraint. Thus, the system performs continuous profiling and reorganization in order to materialize indices that match the most recent traits of the query workload.

COLT relies heavily on what-if optimization [6] in order to measure accurately the gains of candidate index structures and identify indices that are truly beneficial for query evaluation. To reduce the overhead of what-if optimization, COLT employs two key techniques. First, it reduces the query space by clustering queries based on how they perform with respect to indices, and measuring the performance of each index against a sample of queries from each cluster. Second, it reduces the index space by splitting candidate indices into three sets: (a) the materialized indices, (b) the *hot* indices that are not materialized but are considered promising, and (c) the remaining indices. The intuition is that hot indices have a solid chance to become materialized and hence the system is willing to use what-if optimization to estimate their potential gains. For the less promising indices, on the other hand, COLT uses much coarser (and cheaper) estimates of their performance. COLT carefully moves indices between these classes in order to focus the limited profiling resources on the indices that are most likely to make a difference. To further control the total overhead of self-tuning, COLT determines the amount of profiling resources dynamically, based on the *degree of stability of the workload*. Intuitively speaking, COLT lowers its overhead if the workload is stable and the system is well-tuned, and starts spending more resources when a shift is detected and the system has to adapt to a new configuration. This is a novel element of our framework and an important feature for the application of on-line tuning in practice. We do not discuss further the details of our technique in the interest of space. A more complete description can be found in the corresponding technical report [20].

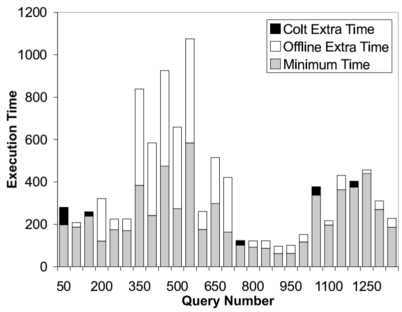

Figure 1: COLT vs. off-line tuning.

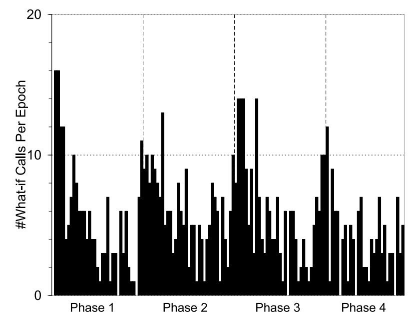

Figure 2: Overhead of on-line tuning in COLT.

We have completed a prototype implementation of COLT inside the Postgres open-source database system and have performed a preliminary experimental study on the effectiveness of our technique. (A demonstration of our prototype was presented in the recent SIGMOD conference [19].) We report here two sets of results that are characteristic of the performance of COLT. Our evaluation is based on a shifting workload that consists of four "phases", each phase having roughly the same number of queries and containing queries from a different distribution. (One can think of each phase as the set of queries that correspond to a different hypothesis tested by an analyst.) We play the workload on a database that is tuned on-line by COLT and measure the total query execution time, which includes the overhead of tuning and index materialization. As a baseline for comparison, we measure the query execution time on a database that is tuned off-line with the optimal set of indices for the complete workload. While this is not a realistic benchmark, it provides useful insights on the performance of on-line tuning. Figure 1 shows the total query execution time in the two systems. The x-axis corresponds to the progression of the workload, while the y-axis shows query execution time. Each bar corresponds to the total execution time of a group of 50 queries and colors are assigned as follows: The gray part always corresponds to the minimum of the two techniques; A white bar shows the extra time taken by the off-line technique; A black bar shows the extra time taken by COLT. Initially, COLT trails behind the off-line technique since the system is completely un-tuned and COLT tries to select an effective set of indices. Past that point, COLT clearly outperforms the off-line technique and offers substantial improvements. The reason is that COLT adapts the index set to the current traits of the queries, and can thus choose a good set of indices for each phase individually. The off-line technique, on the other hand, optimizes for the complete set of queries and thus ignores any temporal variations in the workload. Figure 2 shows another metric of interest for on-line tuning, namely, the overhead of the tuner as the workload progresses. The x-axis is the same as before, but the y-axis now charts the number of what-if optimizer calls per group of 10 queries. Overall, the overhead has four discernible peaks that coincide with the shifts in the workload, and tapers off in between. This demonstrates that COLT works as intended: it intensifies tuning when a change is detected, and reduces it when the system is well tuned. Overall, our preliminary results provide strong evidence on the efficacy of on-line tuning in the implementation of autonomic query processors.

Future Work. Currently, COLT supports the tuning of the physical design with single-column indices. We have considered this design choice as an essential initial compromise in order to deal with the complexity of the on-line problem and understand the issues involved. Still, we would like to stress that our work is of practical value, since single-column indices can provide substantial benefits. (This is corroborated by the results in another study [7].) Moreover, one can envision COLT as a general tuning tool that can complement the more powerful (and more expensive) off-line techniques that are used in modern relational systems. In this scenario, the relational database can be tuned off-line based on an initial expected workload, and COLT can be used on-line in order to maintain additional single-column indices based on the deviations of the actual workload from the representative.

As part of our ongoing work on COLT, we plan to extend our framework to multi-column indices and materialized views. We note that the current COLT framework can readily support these extensions in terms of measuring the benefit of candidate physical structures and the allocation of profiling resources. The key issue of course is the increased complexity of the search space, since the tuning framework has essentially many more choices to consider, and the interactions between physical structures become more pronounced.

Another important issue in the efficient materialization of a new design once a shift in the workload has been detected. Currently, COLT employs asynchronous materialization requests (essentially, `CREATE INDEX` commands) that are executed in parallel with user queries. This strategy can impact the performance of normal query evaluation, since index materialization involves several expensive operations, such as, sorting the data and writing to temporary storage. One approach to deal with this issue is to execute these requests at a lower priority than user queries. A more efficient alternative that we plan to consider is the realization of these requests by taking advantage of the byproducts of normal query execution. More specifically, the physical plan for a user query generates intermediate results as part of the processing required to produce the answers. The idea is to reuse these intermediate results in the processing required to materialize an index, thus meshing the installation of a new physical configuration with the execution of the normal query load. This approach has the potential to reduce significantly the overhead of on-line tuning and thus enable its scale-up to large data sets.

Finally, we plan to investigate techniques that can explore the trade-off between the cost of materializing a new physical schema and its potential benefits. The motivation stems again from the need to regulate effectively the overhead of on-line tuning. The main idea is best illustrated with an example. Consider a system that is heavily loaded and that has relatively few spare resources for materializing a new physical schema. Furthermore, assume that the workload has changed significantly and thus it is beneficial to install a new physical design. Among all the candidate designs that can improve the performance of the system, it may be desirable to install a design that represents an incremental change to the current physical schema in an attempt to minimize the overhead of adaptation. When the system load decreases, the tuner can become more aggressive and apply more drastic and thus more expensive modifications that will bring greater benefit. In this fashion, the on-line tuning process can trade-off between the cost of installing a design and the associated benefits, depending on the spare resources of the system. Achieving this goal will require rethinking the on-line tuner, since the "value" of a configuration will need to be determined along two correlated axes, namely, benefit and cost. To the best of our knowledge, this is a novel aspect of the proposed research that has not been considered in previous studies.

3 Personnel and Budget

Personnel. The PI of the project is Neoklis Polyzotis, an assistant professor at UC Santa Cruz. The PI's research has focused on self-tuning database systems [19, 1, 21, 12, 2], approximate query answering for semi-structured and relational data [15, 23, 22, 16, 10, 11, 17, 14], and active data warehousing [13]. The project will also involve Karl Schnaitter, a graduate student at UC Santa Cruz. Mr. Schnaitter has been the leading student behind COLT is thus well prepared to perform the research described in this proposal. Note that Mr. Schnaitter is a US citizen and thus eligible to be funded by LANL.

Budget. The budget comprises one month of summer salary for the PI and funding for one graduate student researcher (GSR) over the summer quarter of 2007. The following table details these components.

References

[1] S. Abiteboul and N. Polyzotis. Data ring: Let us turn the net into a database! In Yannis Manolopoulos, Jaroslav Pokorný, and Timos K. Sellis, editors, *ADBIS*, volume 4152 of *Lecture Notes in Computer Science*, page 1. Springer, 2006.

[2] S. Abiteboul and N. Polyzotis. The Data Ring: Community Content Sharing. In *Conference on Innovative Data Systems Research (CIDR)*, page (To Appear), 2007.

[3] N. Bruno and S. Chaudhuri. Automatic Physical Database Tuning: A Relaxation-based Approach. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005.

[4] Nicolas Bruno and Surajit Chaudhuri. An online approach to physical design tuning. In *ICDE '07: Proceedings of the 23rd International Conference on Data Engineering*, 2007.

[5] S. Chaudhuri and V. R. Narasayya. An Efficient Cost-Driven Index Selection Tool for Microsoft SQL Server. In *Proceedings of the 23rd Intl. Conf. on Very Large Data Bases*, pages 146–155, 1997.

[6] S. Chaudhuri and V. R. Narasayya. AutoAdmin 'What-if' Index Analysis Utility. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, pages 367–378, 1998.

[7] Mariano P. Consens, Denilson Barbosa, Adrian Teisanu, and Laurent Mignet. Goals and benchmarks for autonomic configuration recommenders. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 239–250, 2005.

[8] Martin R. Frank, Edward Omiecinski, and Shamkant B. Navathe. Adaptive and automated index selection in rdbms. In *EDBT '92: Proceedings of the 3rd International Conference on Extending Database Technology*, pages 277–292, London, UK, 1992. Springer-Verlag.

[9] Michael Hammer and Arvola Chan. Index selection in a self-adaptive data base management system. In *ACM SIGMOD International Conference on Management of Data*, pages 1–8, 1976.

[10] N. Polyzotis and M. Garofalakis. Statistical Synopses for Graph Structured XML Databases. In *Proceedings of the 2002 ACM SIGMOD Intl. Conf. on Management of Data*, pages 358–369, 2002.

[11] N. Polyzotis and M. Garofalakis. Structure and Value Synopses for XML Data Graphs. In *Proceedings of the 28th Intl. Conf. on Very Large Data Bases*, pages 466–477, 2002.

[12] N. Polyzotis and Y. Ioannidis. Speculative query processing. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research*, 2003.

[13] N. Polyzotis, Spiros Skiadopoulos, Panos Vassiliadis, Alkis Simitsis, and Nils-Erik Frantzell. Supporting streaming updates in an active data warehouse. In *Proceedings of IEEE ICDE 2007*, April 2007.

[14] Neoklis Polyzotis and Minos Garofalakis. Xsketch synopses for xml data graphs. *ACM Trans. Database Syst.*, 31(3):1014–1063, 2006.

[15] Neoklis Polyzotis, Minos Garofalakis, and Yannis Ioannidis. Approximate xml query answers. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 263–274, 2004.

[16] Neoklis Polyzotis, Minos Garofalakis, and Yannis Ioannidis. Selectivity Estimation for XML Twigs. In *Proceedings of the 20th Intl. Conf. on Data Engineering*, pages 264–275, 2004.

[17] Neoklis Polyzotis and Minos N. Garofalakis. XCluster Synopses for Structured XML Content. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *Proceedings of the 22nd Intl. Conf. on Data Engineering*, page 63. IEEE Computer Society, 2006.

[18] Kai-Uwe Sattler, Martin Lühring, Ingolf Geist, and Eike Schallehn. Autonomous management of soft indexes. In *SMDB '07: Proceedings of the 2nd International Workshop on Self-Managing Database Systems*, 2007.

[19] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. Colt: continuous on-line tuning. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 793–795, 2006.

[20] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. On-line Database Tuning. Technical Report UCSC-CRL-06-07, UC Santa Cruz, 2006.

[21] K. Schnaitter, S. Abiteboul, T. Milo, and Neoklis Polyzotis. On-line index selection for shifting workloads. In *2nd International Workshop on Self-Managing Database Systems (SMDB)*, 2007.

[22] Joshua Spiegel and Neoklis Polyzotis. Graph-based synopses for relational selectivity estimation. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 205–216, 2006.

[23] Joshua Spiegel, Emmanuel Pontikakis, Suratna Budalakoti, and Neoklis Polyzotis. Aqax: a system for approximate xml query answers. In *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pages 1159–1162. VLDB Endowment, 2006.

[24] D. C. Zilio, J. Rao, S. Lightstone, G.M. Lohman, A. Storm, C. Garcia-Arellano, and S. Fadden. DB2 Design Advisor: Integrated Automatic Physical Database Design. In *Proceedings of the 30th Intl. Conf. on Very Large Data Bases*, pages 1087–1097, 2004.