An Analysis on Automatic Performance Optimization in Database Management Systems

Sachini Samson
Informatics Institute of Technology,
Colombo, Sri Lanka
e-mail: sachini.2016217@iit.ac.lk

Achala Aponso
Informatics Institute of Technology,
Colombo, Sri Lanka
e-mail: achala.a@iit.ac.lk

Abstract—**Maintaining database management systems under optimum condition is crucial for any enterprise application as it affects the overall system performance. However, the optimization process is a non-trivial task even for experienced database administrators due to many underlying challenges. Due to the importance of database optimization, this domain has been researched for the past several years. In this analysis starting off with discussing the importance of an optimum database for the overall performance, then we focus on the reasons behind the poor performance of database management systems. Database optimization is divided under two main categories as Physical design based optimization and Configuration parameter based optimization. While both of these will be discussed, the priority is given for the Configuration parameter optimization. Configuration based optimization comes with its own set of challenges as of 1) Large configuration parameter space, 2) Interdependency of configuration parameters, and 3) Configuration change with different types of workloads. These challenges will be elaborated while carrying out an analysis on the different approaches used in preexisting work to overcome the challenges. Finally, the paper would focus on foundations of optimization techniques which can be utilized for the configuration based database optimization process.**

Keywords-database; optimization; configuration parameters; tuning

I. Introduction

Due to the rapid technological development in the past few decades, there has been an exponential rise in enterprise applications based on different industries. Stakeholders gathered around these enterprises have expanded as a result of this drastic development [1]. Along with the rapid development, the data interaction among the business processes has also increased [2]. Database management system which is a core component of any enterprise application store and manages these critical data while handling the interactions requested by the stakeholders [3]. Systems with sensitive personal data such as banking systems, hospital database systems and identity and access management systems handle millions of data interactions on a daily basis. Maintaining the database management systems under optimal conditions with proper database optimization mechanisms is important to any enterprise application as it would have a direct impact on the company revenue [4].

II. Impact of Performance in Database Management Systems for the Overall System Performance

Maintaining the database systems with optimum conditions has become an overwhelming task for intensive data-driven systems. Often database management systems become the bottleneck to the system due to different performance-related issues. Mike Gualtieri, VP, Principal Analyst from Forrester has also stated that "The Database bottleneck is often nastiest to solve" [5]. Regardless of challenges faced in optimization databases almost every company focus on the optimization process. Maintaining the performance of a system is essential for an enterprise in order to deliver the quality of service to the customers. But often, poor database performance directly affects application slowdowns. TechValidate survey has identified that 71% of the application performance issues are related to databases [6].

III. Poor Performance in Database Management Systems

With the rapid accumulation and process of data within a system increases, there is a probability of performance degradation in database management systems. There are many reasons which would result in poor performance or slower query response time in database management systems.

1) Different hardware factors such as memory usage, CPU usage, IO and network resources [7] would act as a performance burden. Performance degradation of memory usage and I/O could be improved by means the change on database management server configurations. But depending on the system environment there are instances where the resource upgrade is required.

2) Inefficient database design factors such as poor written queries [8], indexing [9], [10], cardinality Estimation [11], partitioning and material views [12] will also have an impact on the performance. Techniques of physical design based optimization could be utilized to improve the performance degradation due to efficient database design factors. These two approaches will be discussed in the next section.

978-1-7281-9738-8/20/$31.00 ©2020 IEEE

IV. PHYSICAL DESIGN BASED OPTIMIZATION

When it comes to database tuning, an extensive number of research has been focused on the physical design of database systems. This approach focuses on the design aspect when tuning the database management systems. [13] of Hasso Plattner Institute, University of Potsdam researched on how multi-attribute index selection can be used in database optimization. Efficient index selection in large commercial database management systems are crucial in terms of database performance. Other than the performance aspect, the indexes consume a considerable amount of main memory as well. They have proposed a novel recursive strategy for index selection which is applicable for large index selection problems. Schlosser et.al[13] have evaluated the proposed solution with Enterprise workload with 4204 relevant attributes in 500 tables and commercial database management systems.

Idreos et al. [14] have come up with a hybrid approach for adaptive indexing by considering the database cracking and adaptive merging techniques. They have used the hybrid algorithms to merge the best features of the adaptive merging and database cracking methodologies. The experimental evaluation has been carried on MonetDB while focusing on the concepts of fast convergence to complete index and overhead on first queries.

Optimum query flow is another important concept in optimized database designing. Woltmann et al., [11] have proposed an approach based on deep neural networks in order to obtain the cardinality estimation which is a fundamental task in the query optimization process. Traditionally the cardinality estimation process is measured with statistical models based on assumptions [15]. These assumptions are simplified on factors such as independence and uniformity. But in real-world scenarios, these assumptions are frequently invalid which will result in an adverse impact on query optimization. Woltmann et al overcame these issues using neural network based on local models. The local models have improved the performance in training by a factor of four and accuracy has been improved by two orders of magnitude.

There has been a significant amount of research carried out in optimization database management systems based on database design factors. In this analysis we would focus more on the database optimization using configuration parameters.

V. DATABASE OPTIMIZATION USING CONFIGURATION PARAMETERS

A. Challenges Faced in Configuration Parameter Tuning

There are several challenges faced in configuration parameter optimization.

1) Large number of configuration parameters: Database systems are being handled by a large number of configuration parameters. These parameters control system behaviors such as memory distribution, logging aspects, I/O aspects and much more [16]. Manual optimization of configuration parameters have become overwhelming with the increasing number of parameters.

2) Interdependent Configuration parameters: These configuration parameters are interdependent with each other. That means increasing the value of one parameter may adversely deteriorate the performance of another. One misconfiguration can lead to serious performance issues [17].

3) Optimization defer with the workload: Configured database parameters for one workload may not be the optimum conditions for another workload. Van Aken et al., from Carnegie Mellon University, has identified this as a reason why default parameters won't be ideal for every workload [17].

B. Approaches in Configuration Parameter Optimization

Database configuration tuning focuses on selecting appropriate configuration parameters to optimize the database. Oracle approached the performance issues by developing an internal monitoring system which diagnoses the performance bottlenecks. They defined a measurement called DBtime which was used in understanding the performance impacts with each component. After diagnosing the issues, the system provided a suggestion on how to alleviate performance bottlenecks [18]. Microsoft research along with Carnegie Mellon University worked on proposing a “Resource advisor” to answer “what if” questions for Micro Server database management system [19].

IBM introduced Self-Tuning Memory Manager (STMM) which optimizes the memory heaps and the cumulative memory allocation based on the cost-benefit analysis technology [20]. The cost-modeling approach which has been used by IBM needs a deep understanding of the system internals to build a properly functioning model.

iTuned is the proposed solution for the research carried out by [16] for database configuration parameter tuning. iTuned uses Latin Hypercube Sampling (LHS) sampling techniques to identify the initial set of samples in the initialization process and eventually gather multiple sets of samples and obtain the best out of them. Afterward, iTuned adopts the Gaussian process [21] to check for an optimal database configuration parameter. Van Aken et al., proposed OtterTune which uses a similar approach as iTuned. Machine learning-based pipeline model have been adopted by them to analyze and configure the database management systems [17].

S. F. Rodd and U. P. Kulkarni [22] confronted the problem with a different approach with a novel neural network algorithm. A feed-forward network is used in the control architecture of the literature presented and sigmoid function is used as the activation function. The controlled architecture implementation presented in the paper has only tuned the buffer cache configuration parameter.

VI. CRITICAL ANALYSIS

Different types of optimization methods which have been used over the past years for different domains.

1) Grid search

Grid search for parameter optimization has been used extensively in many machine learning related research. In this approach the developer needs to assign the search space where the grid search would build models and evaluate for all the possible combinations of hyperparameters [23]. Though grid search is considered to be reliable with low dimensional space, this approach would not be suitable for database optimization as there are hundreds of configuration parameters. Building a model for each combination of these parameters would be costly and time consuming.

2) Random search

In literature, J. Bergstra and Y. Bengio [24] have stated that Random search is empirically and theoretically efficient than grid search on the trials selected for parameter optimization. In this approach selected combinations would be evaluated with the objective function within the specified trials. Though this approach is efficient than the grid search method, it cannot guarantee that optimum parameter values could be obtained with the given number of trials as the objective function is computationally expensive to evaluate with the large parameter space.

3) Bayesian optimization

Bayesian optimization has been proven to be more efficient compared to Grid search and random search methods when it comes to optimization problems with expensive black box function evaluations such as hyperparameter tuning in deep neural networks [25]. Bayesian optimization would use a surrogate model which approximates to the actual functional but is comparatively cheap to evaluate. A popular choice for the surrogate model is the Gaussian process. The prior over functions is defined by a gaussian process which can be incorporated to the prior belief of the black box function [26]. Picking up the hyperparameters from the search space for the next experiment is done by the acquisition function. This balances between exploitation which is the search of regions with high estimated values and exploration where search is carried out on high uncertainty regions. Acquisition function can be evaluated as Expected improvement [27] or by the Upper confidence bound [28]. Unlike Random search and Grid search, Bayesian optimization could be modelled for a database optimization process with an expensive objective function to find optimum parameter values within a considerable amount of time.

4) Evolutionary optimization

Based on evolution theory this algorithm resembles natural selection and evolution. Starting with the initial population of the best individuals are selected after evaluating the fitness function. The selected individuals are subjected to crossover and mutation as defined to create the new generation individuals. The new generation of individuals too would be evaluated and the algorithm iterates until the predefined generation is achieved [29].

In recent times, under evolutionary optimization, the genetic algorithm has been extensively utilized for parameter optimization problems. Ability of crossover and mutation provide more diversity among the selected individuals which gives the opportunity to identify the optimum parameters from the rest. Also, the multi-objective approach on genetic algorithm could be incorporated for the optimization process to tune more than one performance metric.

VII. Conclusion

According to the above stated facts, it is clearly identified that optimization in database management systems will have a huge impact on the system performance and in the end will affect the business revenue as well. But its is not an easy task to optimize the database with the large parameter space even for industry experts. As the configuration parameters are interdependent, optimizing one parameter would negatively affect the configuration of the other dependent parameters. Unlike the Manual tuning which is time and resource consuming, an automated optimization approach would give better results for processes with expensive objective functions.

Random search and grid search methods would not be ideal for database optimization due to the large search space and costly objective function. From the above discussed techniques the Bayesian optimization technique would be most suited to predict configuration parameters with reduced optimization time.

The above analysis will be used to develop an automated configuration parameter optimization framework for database management systems.

Acknowledgment

This analysis paper was prepared for final year research carried out by Sachini Samson, an undergraduate of Informatics Institute of Technology, Sri Lanka affiliated with University of Westminster, UK.

References

[1] M. O. Lewis, B. W. Young, L. Mathiassen, A. Rai, and R. J. Welke, "Business process innovation based on stakeholder perceptions," *Inf. Knowl. Syst. Manag.*, vol. 6, pp. 7–27, 2007.

[2] I. Fernandez, "Database Tuning," in *Beginning Oracle Database 12c Administration: From Novice to Professional, Second Edition [Book]*, Second edition., Apress, 2015.

[3] H. E. Williams and D. Lane, *Web Database Applications with PHP & MySQL, 2Nd Edition*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2004.

[4] I. Forrester Research, "Application Performance Management Is Critical To Business Success," 2014. [Online].Available:https://deploypartners.com/wp-content/uploads/2016/01/Application-Performance-Management-Is-Critical-To-Business-Success.pdf.

[5] M. Gualtieri, "The Nastiest Performance Bottleneck Is Often The Database," *Forrester*, 13-Feb-2011. [Online].Available:https://go.forrester.com/blogs/11-02-13-the_nastiest_performance_bottleneck_is_often_the_database/].

[6] 806-5EA-807 TechValidate, "Fast Applications Start with the Database," *TechValidate*, 25-Nov-2014. [Online].Available:https://www.techvalidate.com/product-research/solarwinds-dpa/facts/806-5EA-807.

[7] F. Charvet and A. Pande, "Database Performance Study," p. 55, 2003.

[8] J. Jung, H. Hu, J. Arulraj, T. Kim, and W.-H. Kang, "APOLLO: Automatic Detection and Diagnosis of Performance Regressions in Database Systems," *PVLDB*, vol. 13, pp. 57–70, 2019.

[9] G. Graefe, "Adaptive merging in database indexes," US9298761B2, 29-Mar-2016.

[10] Y. Wu, J. Yu, Y. Tian, R. Sidle, and R. Barber, "Designing Succinct Secondary Indexing Mechanism by Exploiting Column Correlations," in *Proceedings of the 2019 International Conference on Management*

of Data - SIGMOD '19, Amsterdam, Netherlands, 2019, pp. 1223–1240, doi: 10.1145/3299869.3319861.

[11] L. Woltmann, C. Hartmann, M. Thiele, D. Habich, and W. Lehner, "Cardinality estimation with local deep learning models," 2019, pp. 1–8, doi: 10.1145/3329859.3329875.

[12] I. Ilyas, V. Markl, P. Haas, P. Brown, and A. Aboulnaga, "CORDS: Automatic Discovery of Correlations and Soft Functional Dependencies.," 2004, pp. 647–658, doi: 10.1145/1007568.1007641.

[13] R. Schlosser, J. Kossmann, and M. Boissier, "Efficient Scalable Multi-attribute Index Selection Using Recursive Strategies," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, Macao, Macao, 2019, pp. 1238–1249, doi: 10.1109/ICDE.2019.00113.

[14] S. Idreos, S. Manegold, H. Kuno, and G. Graefe, "Merging What's Cracked, Cracking What's Merged: Adaptive Indexing in Main-memory Column-stores," *Proc VLDB Endow*, vol. 4, no. 9, pp. 586–597, Jun. 2011, doi: 10.14778/2002938.2002944.

[15] V. Leis, A. Gubichev, A. Mirchev, P. Boncz, A. Kemper, and T. Neumann, "How good are query optimizers, really?," *Proc. VLDB Endow.*, vol. 9, no. 3, pp. 204–215, Nov. 2015, doi: 10.14778/2850583.2850594.

[16] S. Duan, V. Thummala, and S. Babu, "Tuning Database Configuration Parameters with iTuned," *PVLDB*, vol. 2, pp. 1246–1257, 2009, doi: 10.14778/1687627.1687767.

[17] D. Van Aken, A. Pavlo, G. J. Gordon, and B. Zhang, "Automatic Database Management System Tuning Through Large-scale Machine Learning," in *Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17*, Chicago, Illinois, USA, 2017, pp. 1009–1024, doi: 10.1145/3035918.3064029.

[18] K. Dias, M. Ramacher, U. Shaft, V. Venkataramani, and G. Wood, "Automatic Performance Diagnosis and Tuning in Oracle," in *CIDR*, 2005.

[19] D. Narayanan, E. Thereska, and A. Ailamaki, "Continuous resource monitoring for self-predicting DBMS," in *13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Atlanta, GA, USA, 2005, pp. 239–248, doi: 10.1109/MASCOTS.2005.21.

[20] A. J. Storm and C. Garcia-arellano, "Adaptive self-tuning memory in DB2," in *In Proc. of the 2006 Intl. Conf. on Very Large Data Bases*, 2006.

[21] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.

[22] S. F. Rodd and U. P. Kulkarni, "Adaptive self-tuning techniques for performance tuning of database systems: a fuzzy-based approach with tuning moderation," *Soft Comput.*, vol. 19, no. 7, pp. 2039–2045, Jul. 2015, doi: 10.1007/s00500-014-1389-3.

[23] M. Claesen, J. Simm, D. Popovic, Y. Moreau, and B. De Moor, "Easy Hyperparameter Search Using Optunity," *ArXiv14121114 Cs*, Dec. 2014.

[24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, Feb. 2012.

[25] E. Brochu, V. Cora, and N. Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning," *CoRR*, vol. abs/1012.2599, Dec. 2010.

[26] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, Lake Tahoe, Nevada, 2012, pp. 2951–2959.

[27] R. Bardenet and B. Kégl, "Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010, 2010.

[28] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 1015–1022.

[29] C. Di Francescomarino *et al.*, "Genetic Algorithms for Hyperparameter Optimization in Predictive Business Process Monitoring," *Inf. Syst.*, Jan. 2018, doi: 10.1016/j.is.2018.01.003.