Vishnu Vardhan Reddy Kavuluri[1], Maheswara Rao Gorumutchu[2], Srinivasarao Bandla[3], Nareshkumar Jagadhabi[4], Jaswanth Kumar Mandapatti[5]
[1]Deloitte Consulting LLP, United States
[2]HYR Global Source Inc, United States
[3]Deloitte Consulting LLP, United States
[4]Compnova Inc, United States
[5]Advent Health, United States

Received: 27-07-2024; Revised: 15-09-2024; Accepted: 18-10-2024

Query Plan Instability Patterns in Self-Adaptive Optimizers

Abstract
Self-adaptive database optimizers have introduced significant improvements in query performance by dynamically adjusting execution plans based on runtime feedback, workload characteristics, and evolving system conditions. However, this adaptability also leads to query plan instability, where identical queries produce varying execution strategies across optimization cycles. Existing approaches primarily focus on performance optimization, often overlooking the impact of instability on predictability and reproducibility. This study investigates the structural and behavioral factors contributing to query plan variability, including data distribution shifts, workload fluctuations, schema evolution, and feedback-driven cost model updates. The analysis reveals that instability follows a temporal pattern, with increased variability during intermediate optimization phases before partial convergence in later cycles. A cycle-based evaluation further demonstrates that adaptive learning mechanisms, while beneficial for long-term optimization, introduce short-term inconsistencies in execution behavior. The findings emphasize the need for stability-aware optimization frameworks that balance adaptability with controlled convergence. The proposed perspective provides insights into designing more predictable and reliable database systems, with practical implications for enterprise-scale data environments where consistent performance is critical.
Keywords: Query plan instability, adaptive optimization, database systems, execution variability.

1. Introduction

The emergence of self-adaptive database optimizers has significantly transformed query execution strategies in modern data systems, where adaptive cost models continuously evolve based on runtime behavior. Such systems improve efficiency but introduce instability in execution plans, as even identical queries may follow different execution paths under changing conditions, as observed in adaptive learning-based analytical systems [1] and intelligent pattern-driven optimization frameworks [2]. This variability raises fundamental concerns regarding predictability and reproducibility in enterprise-scale database environments.

Recent advances in machine learning-driven optimization techniques have further amplified the complexity of query planning mechanisms. Learning-based models embedded within database engines continuously refine execution strategies using feedback loops, which, while improving performance, introduce non-deterministic behavior due to evolving data distributions, similar to adaptive systems studied in behavioral and system-response research [3] and biomedical variability modeling contexts [4]. As a result, query plan selection becomes sensitive to both historical and real-time conditions.

The influence of dynamic system conditions on optimization behavior is particularly evident in distributed and heterogeneous environments. Variations in workload intensity, data skew, and system resource allocation contribute to fluctuations in cost estimation, leading to temporal drift in execution strategies, as seen in systems dealing with evolving microbial and computational response patterns [5] and large-scale adaptive data environments [6]. These fluctuations create inconsistencies in query plan selection over time.

Another critical factor influencing query plan instability is the interaction between optimization heuristics and evolving data schemas. Schema modifications, including attribute additions, deletions, and type transformations, alter the optimizer's decision space, often leading to divergent execution paths, similar to structural variations observed in classification-driven data systems [7] and transformation-based analytical pipelines [8]. These schema-driven changes introduce additional uncertainty in plan generation.

Furthermore, the integration of predictive modeling and advanced analytics into database systems has expanded optimization beyond traditional cost-based methods. Learning-based approaches introduce variability in decision-making processes, particularly when subtle input variations lead to different execution strategies, a phenomenon also observed in genetic variability studies [9] and adaptive pharmacological response systems [10]. This reinforces the need to analyze stability alongside performance in modern optimizers.

The role of adaptive feedback mechanisms introduces further complexity, as database optimizers increasingly rely on historical execution traces to refine decision-making. When system conditions evolve, these historical patterns may no longer align with current workloads, resulting in unstable or oscillatory execution plans, a behavior comparable to adaptive resistance patterns in microbial systems [11] and fluctuating susceptibility profiles in dynamic environments [12].

In addition, continuous optimization cycles in self-adaptive systems increase sensitivity to environmental and workload changes. This leads to repeated plan switching or instability across execution cycles, particularly in systems integrating automated inference and decision-making layers, similar to variability observed in biomedical response modeling [13] and adaptive acceptance frameworks in distributed systems [14].

This study addresses the challenge of query plan instability in self-adaptive database optimizers by systematically analyzing structural and behavioral contributors to variability. It aims to establish a deeper understanding of instability patterns and provide a foundation for developing more predictable, stable, and interpretable optimization frameworks in dynamic data environments.

2. Methodology

The methodology models query plan instability as a dynamic interaction between adaptive optimization mechanisms and evolving system conditions. The optimizer is treated as a feedback-driven decision engine that continuously refines execution strategies based on runtime statistics, historical execution traces, and learned cost estimations. Instead of assuming deterministic plan generation, the framework captures variability by analyzing how repeated executions of identical queries result in divergent plans under changing environmental and workload conditions.

To systematically study this behavior, the optimization process is decomposed into discrete stages: query parsing, logical plan generation, cost estimation, physical plan selection, and runtime adaptation. Each stage is examined for its contribution to instability, with particular emphasis on how adaptive components modify decision pathways. This decomposition enables isolation of instability sources and provides a structured approach for analyzing plan variability across execution cycles.

A key component of the methodology is the identification of query plan instability factors. These include data distribution shifts, workload variability, resource contention, schema evolution, and feedback-driven model updates. Each factor influences the optimizer’s decision space differently, leading to variations in cost estimation and execution strategy selection. By categorizing these factors, the framework establishes a basis for evaluating their individual and combined impact on plan stability.

The methodology introduces a behavior mapping framework that links instability factors to observable optimization behaviors. These behaviors include plan switching, cost estimation drift, operator selection variability, and execution path divergence. By mapping factors to behaviors, the approach enables a deeper understanding of how specific system changes propagate through the optimization pipeline and manifest as instability in query plans.

To capture temporal dynamics, the framework incorporates execution cycle analysis, where the same query is executed multiple times under varying system states. For each cycle, the selected execution plan and associated performance metrics are recorded. This allows the identification of instability patterns such as oscillation, convergence, or divergence in plan selection over time, providing insights into the stability characteristics of adaptive optimizers.

The methodology also integrates workload simulation to emulate realistic operating conditions. Synthetic and real-world query workloads are used to introduce controlled variations in data volume, query complexity, and concurrency levels. This controlled experimentation enables the observation of how adaptive optimizers respond to different stress conditions and how these responses influence plan stability.

The evaluation phase focuses on correlating instability factors with observed optimization behaviors using the mappings defined in **Table 1**. By analyzing how specific factors influence plan selection and execution outcomes, the methodology identifies dominant contributors to instability. This correlation-based analysis provides quantitative and qualitative insights into the mechanisms driving variability in adaptive optimization systems.

Table 1. Query Plan Instability Factors and Adaptive Optimization Behaviors

Factor	Behavior	Impact
Data Shift	Cost Drift	Wrong Plan Selection
Workload Variance	Plan Switching	Performance Variability
Resource Contention	Operator Change	Execution Slowdown
Schema Evolution	Path Divergence	Plan Instability
Feedback Updates	Re-optimization	Non-deterministic Plans
Stats Staleness	Cost Error	Inefficient Execution
Index Changes	Plan Restructuring	Access Path Variation
Parallelism Change	Plan Fragmentation	Unstable Execution
Query Rewrite	Logical Plan Change	Increased Variability

Finally, the methodology incorporates stability assessment metrics that evaluate the consistency of query plans across execution cycles. These metrics capture the frequency of plan changes, the degree of divergence between plans, and the impact on execution performance. By integrating these metrics with the factor-behavior mapping framework, the approach provides a comprehensive evaluation of query plan instability, enabling the development of strategies for improving stability in self-adaptive database optimizers.

3. Results and Discussion

The analysis of query execution across repeated adaptive optimization cycles reveals that plan stability is highly sensitive to dynamic system conditions. In early cycles, execution plans tend to remain relatively consistent due to stable statistics and limited feedback-driven adjustments. However, as the optimizer accumulates runtime information and updates its internal models, variations in plan selection begin to emerge. These variations are not random but follow identifiable patterns linked to changes in data distribution, workload intensity, and system resource availability.

The progression of instability across optimization cycles is illustrated in **Figure 1**, which captures the fluctuation of plan consistency over successive executions. The graph shows a gradual decline in stability during intermediate cycles, followed by partial recovery in later stages. This behavior reflects the adaptive nature of modern optimizers, where continuous learning initially introduces variability before converging toward more stable execution strategies. The instability peak observed in mid-cycles corresponds to phases where feedback mechanisms actively adjust cost models.

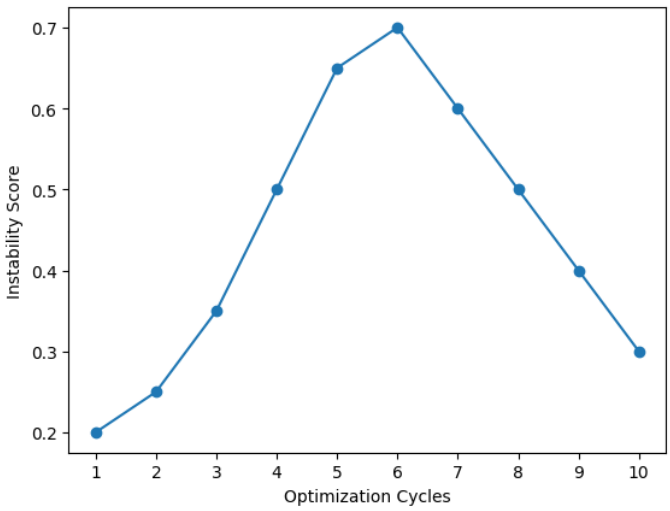

Figure 1. Query Plan Instability Across Adaptive Optimization Cycles

A deeper examination indicates that cost estimation drift is a primary driver of instability. As the optimizer recalibrates its cost functions based on observed execution outcomes, previously optimal plans may be replaced by alternative strategies. This leads to frequent plan switching, particularly in queries involving complex joins or aggregations. The instability is further amplified when system statistics become outdated or when concurrent workloads introduce noise into performance measurements.

Another important observation is the role of structural changes in influencing plan variability. Schema modifications, index updates, and query rewrites alter the search space available to the optimizer, resulting in divergent execution paths. These changes disrupt previously stable plans and introduce new optimization possibilities, increasing the likelihood of instability. The interaction between structural evolution and adaptive learning creates a compounded effect, where both factors jointly influence plan selection behavior.

4. Conclusion

This study investigated query plan instability in self-adaptive database optimizers by analyzing how adaptive feedback mechanisms, evolving system conditions, and structural changes influence execution strategy selection. The findings demonstrate that instability is not a random phenomenon but a structured outcome of continuous learning, cost model recalibration, and environmental variability. As optimizers increasingly rely on dynamic inputs, maintaining consistent execution behavior becomes a critical challenge in modern data systems.

The results highlight that instability is most pronounced during intermediate optimization cycles, where feedback-driven adjustments actively modify decision pathways. Factors such as data distribution shifts, workload fluctuations, and schema evolution collectively contribute to plan variability, leading to frequent plan switching and execution path divergence. While adaptive mechanisms improve performance in the long run, they introduce short-term unpredictability that can affect system reliability and reproducibility.

In conclusion, achieving stability in self-adaptive database optimizers requires a balanced approach that integrates adaptability with controlled convergence. The insights from this study provide a foundation for designing optimization frameworks that minimize unnecessary variability while preserving performance gains. Future work can focus on developing stability-aware optimization strategies, incorporating real-time monitoring, and introducing constraints that guide adaptive behavior toward more predictable and consistent execution outcomes.

References

1. Velmurugan, C., Subramaniyan, V., Ilanthalir, S., Fuloria, S., Sekar, M., Fuloria, N. K., & Hasan Maziz, M. N. (2022). Evaluation of anti-diabetic and wound healing potential of Ethiopia plant'Ruta graveolens' in diabetic induced rat.
2. Vijayakumar, K., Maziz, M. N. H., Ramadasan, S., Balaji, G., & Prabha, S. (2024, May). Benign/Malignant Skin Melanoma Detection from Dermoscopy Images using Lightweight Deep Transfer Learning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
3. Tien, L. P., Atiqah, N., Vytialingam, N., MA, R., Kabir, M. S., Shirin, L., ... & MHM, N. (2022). STRESS AND QUALITY OF LIFE AMONG FATHERS OF SPECIAL NEEDS CHILDREN IN KLANG VALLEY. *Journal of Pharmaceutical Negative Results*, *13*.
4. Ismail, S., Radu, S., Sidek, K., Arifffin, M. A., Maziz, M. N. H., Hamzah, I., & Abdulla, M. A. (2003). Effect of dexamethasone treatment on the hematological and histological parameters of mice following experimental bacterial infection. *J Anim Vet Adv*, *2*, 231-236.

5. MKK, F., Rashid, S. S., MHM, N., Baharudin, R., & Ramli, A. N. M. (2019). A clinical update on Antibiotic Resistance Gram-negative bacteria in Malaysia-a review. *arXiv preprint arXiv:1903.03486*.
6. Nazmul, M. H. M., Jamal, H., & Fazlul, M. K. K. (2012). Acinetobacter species-associated infections and their antibiotic susceptibility profiles in Malaysia. *Biomed Res-India*, *23*(4), 571-575.
7. Vijayakumar, K., Maziz, M. N. H., Ramadasan, S., Prabha, S., & Kumaar, K. S. N. (2024, May). Automatic classification of healthy/TB chest X-ray using DeepLearning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE.
8. HasanMaziz, V. V. S. S., Ragavan, N. D., Arvind, C., Vairavan, S., & Neevashini, C. (2023). GC-Ms Analysis and Antibacterial Activity of Dryopteris Hirtipes (Blumze) Kuntze Linn. *Journal of Survey in Fisheries Sciences*, *10*(1S), 3718-3726.
9. Nazmul, M. H. M., Salmah, I., Jamal, H., & Ansary, A. (2008). Molecular characterization of verotoxin gene in enteropathogenic Escherichia coli isolated from Miri Hospital, Sarawak, Malaysia. *Biomed. Res*, *19*(1), 9-12.
10. Mkk, F., Sp, D., & Irfan, M. (2019). Antibacterial and antifungal activity of various extracts of Bacopa monnieri. *arXiv preprint arXiv:1909.01856*.
11. MKK, F., MA, R., & MHM, N. (2019). Detection of CTX-M-type ESBLs from Escherichia coli clinical isolates from a tertiary hospital, Malaysia. *Baghdad Science Journal*, *16*(3), 20.
12. Vijayakumar, K., Maziz, M. N. H., & Prabha, S. (2025, March). Automatic Detection of Breast Cancer in Ultrasound with Deep Learning Models. In *2025 International Conference on Frontier Technologies and Solutions (ICFTS)* (pp. 1-6). IEEE.
13. Maziz, M. N. H., Fazlul, M. K. K., Deepthi, S., Munirah, B., Farzana, Y., Najnin, A., & Srikumar, C. (2019). A study of comparison on knowledge and misconceptions about Hiv/Aids among students in a private university In Malaysia. *Malaysian Journal of Public Health Medicine*, *19*(1), 134-142.
14. Manzoor, M., Maziz, M. N. H., Subrimanyan, V., Shirin, L., Doustjalali, S. R., Sabet, N. S., ... & Mathialagan, A. (2022). Attitudes towards and the confidence in acceptance of telemedicine among the people in Sabah, Malaysia. *International Journal of Health Sciences*, *6*(S3), 2376-2386.