ENHANCING SKIP LIST AUTHENTICATION FOR RELATIONAL QUERY OPTIMIZATION

Nguyen Duy Hung*, Nguyen Manh Hung, Do Ngoc Long
Military Technical Academy

ARTICLE INFO

Received: 21/11/2024
Revised: 19/12/2024
Published: 19/12/2024

KEYWORDS

Optimization
Relational query
Authentication
Enhancing Skip list
Database systems

ABSTRACT

This paper proposes several enhancements to improve the performance of Skip list authentication in relational query optimization, addressing the limitations of existing methods such as high computational costs and poor scalability when applied to large databases. The proposed approach optimizes the Improved Authenticated Skip List structure by introducing incremental authentication and employing a novel Enhanced Improved Authenticated Skip List structure. These enhancements significantly reduce authentication time for projection-selection queries while improving overall query processing efficiency. The experimental results, conducted on real-world datasets, demonstrated that the proposed method increases authentication speed by more than five times, compared to earlier methods, particularly for queries involving a small number of attributes. Even for queries with a larger number of attributes, the enhanced method continues to deliver comparable performance while maintaining computational efficiency. This research not only enhances the efficiency and reliability of Skip list authentication but also paves the way for new solutions in relational query optimization within database systems.

CẢI TIẾN XÁC THỰC SKIP LIST TRONG TỐI ƯU HÓA TRUY VẤN QUAN HỆ

Nguyễn Duy Hùng*, Nguyễn Mạnh Hùng, Đỗ Ngọc Long
Học viện Kỹ thuật quân sự

THÔNG TIN BÀI BÁO

Ngày nhận bài: 21/11/2024
Ngày hoàn thiện: 19/12/2024
Ngày đăng: 19/12/2024

TỪ KHÓA

Tối ưu hóa
Truy vấn quan hệ
Xác thực
Cải tiến Skip list
Hệ thống cơ sở dữ liệu

TÓM TẮT

Bài báo này đề xuất một số cải tiến nhằm nâng cao hiệu suất của quá trình xác thực Skip list trong tối ưu hóa truy vấn quan hệ, khắc phục những hạn chế của các phương pháp xác thực hiện tại như chi phí tính toán cao và khả năng mở rộng kém khi áp dụng cho các cơ sở dữ liệu lớn. Phương pháp được đề xuất thông qua việc tối ưu hóa cấu trúc Improved Authenticated Skip List. Phương pháp mới giới thiệu cơ chế xác thực gia tăng và sử dụng cấu trúc Enhanced Improved Authenticated Skip List mới. Điều này giúp giảm đáng kể thời gian xác thực trong các truy vấn lựa chọn và chiếu, đồng thời cải thiện hiệu quả xử lý truy vấn. Kết quả thí nghiệm trên tập dữ liệu thực tế cho thấy, phương pháp được đề xuất tăng tốc độ xác thực lên hơn 5 lần so với phương pháp cũ khi số lượng thuộc tính được chọn thấp. Khi số lượng thuộc tính được lựa chọn lớn hơn, phương pháp cải tiến vẫn cho hiệu suất tương đương. Nghiên cứu này không chỉ cải thiện hiệu suất và độ tin cậy trong xác thực Skip list mà còn mở ra hướng đi mới cho các giải pháp tối ưu hóa truy vấn quan hệ trong các hệ thống cơ sở dữ liệu.

DOI: https://doi.org/10.34238/tnu-jst.11580

* Corresponding author. *Email: duyhung.mta49@gmail.com*

1. Giới thiệu

Trong kỷ nguyên dữ liệu lớn, các cơ sở dữ liệu quan hệ được sử dụng rộng rãi để quản lý lượng lớn thông tin có cấu trúc [1]. Các cấu trúc dữ liệu tích hợp cơ chế xác minh tính xác thực và toàn vẹn của dữ liệu được gọi là cấu trúc dữ liệu xác thực (Authenticated Data Structures - ADS). ADS giảm nhu cầu sử dụng các thủ tục bên ngoài để xác thực bằng cách tích hợp các kỹ thuật mật mã, cho phép xác minh hiệu quả và an toàn tính toàn vẹn của dữ liệu [2], [3].

Tối ưu hóa truy vấn, một thành phần quan trọng của hệ thống cơ sở dữ liệu quan hệ, tập trung vào việc lựa chọn kế hoạch thực thi hiệu quả nhất để giảm thiểu thời gian phản hồi truy vấn. Các cơ chế xác thực, chẳng hạn như Skip list và cây Merkle, được sử dụng để xác minh tính toàn vẹn của dữ liệu trong quá trình đánh giá truy vấn [4] – [6].

Các cấu trúc dữ liệu xác thực đang được nghiên cứu tích cực trên nhiều khía cạnh khác nhau, bao gồm cả thuật toán và cấu trúc [7]. Hash chain là một thành phần quan trọng trong cấu trúc dữ liệu xác thực, cung cấp một phương pháp đơn giản nhưng hiệu quả để đảm bảo tính toàn vẹn dữ liệu trong các kịch bản tuần tự. Bằng cách sử dụng các hàm băm mật mã, hash chain cho phép xác minh hiệu quả và phát hiện sự giả mạo trong các ứng dụng như cấu trúc dữ liệu chỉ cho phép thêm, dấu thời gian mật mã và giao thức truyền thông an toàn. Mặc dù hash chain mang lại khả năng xác minh tính toàn vẹn hiệu quả và đơn giản trong các trường hợp chỉ cho phép thêm, nhưng chúng vẫn có những hạn chế về bảo mật giá trị khởi tạo, truy cập ngẫu nhiên, hiệu quả sửa đổi và các đảm bảo bảo mật mở rộng. Hiểu rõ các thuộc tính và ứng dụng của hash chain có thể đóng góp vào việc thiết kế và triển khai các hệ thống an toàn và đáng tin cậy [8].

Các cấu trúc dữ liệu xác thực thường được sử dụng nhất là cây Merkle và các biến thể của nó [9]. Cây Merkle được xây dựng với cấu trúc cây băm trên danh sách dữ liệu được sắp xếp. Dữ liệu nằm ở các nút lá, và mỗi nút bên trong là giá trị băm của các nút con. Do đó, mỗi khi một nút bị thay đổi, sự thay đổi này sẽ lan truyền lên nút gốc. Hàm băm tại nút gốc đóng vai trò là một biểu diễn cô đọng của tập dữ liệu. Tuy nhiên, do hiệu suất kém và chi phí cao trong việc băm và tuần tự hóa, cây Merkle gây ra chi phí đáng kể [10].

Các phương pháp xác thực Skip list truyền thống gặp phải một số hạn chế. Thứ nhất, khi kích thước cơ sở dữ liệu tăng lên, quá trình xác thực trở nên tốn kém về mặt tính toán, dẫn đến suy giảm hiệu suất. Thứ hai, các cập nhật thường xuyên trong cấu trúc cơ sở dữ liệu yêu cầu tính toán lại các giá trị xác thực, gây ra chi phí bổ sung. Cuối cùng, khả năng mở rộng của phương pháp xác thực Skip list đối với cơ sở dữ liệu quy mô lớn cần được cải thiện để xử lý khối lượng dữ liệu ngày càng tăng [11] – [13].

Bài báo này đề xuất cải tiến kỹ thuật dựa trên Improved Authenticated Skip List (I-ASL), Enhanced Improved Authenticated Skip List (EI-ASL). Thay vì sử dụng các bộ dữ liệu (tuples) để cải thiện xác thực Skip list, phương pháp cải tiến sử dụng cấu trúc lưu trữ thay thế. Qua thực nghiệm cho thấy, phương pháp được đề xuất mặc dù làm tăng không gian lưu trữ nhưng đã cải thiện được hiệu suất tổng thể của xử lý truy vấn quan hệ.

2. Cải tiến kỹ thuật xác thực Skip list

2.1. Cấu trúc cơ bản

Trước hết, nghiên cứu này dựa trên cấu trúc I-ASL như nền tảng cho các đề xuất cải tiến. Cấu trúc I-ASLđược J. Xu và cộng sự [14] giới thiệu nhằm cải tiến xác thực Slip list truyền thống, với mục tiêu tối ưu hóa xác thực trong đại số quan hệ. Các hoạt động xác thực này bao gồm các phép toán lựa chọn (Selection), chiếu (Projection), nối (Joint), tổng hợp (Aggregation), và các phép truy vấn khác, trong khi vẫn đảm bảo chi phí tính toán thấp.

Cấu trúc logic của I-ASL có thể được biểu diễn dưới dạng một bộ sáu (S, D, InitHash, Update, Query, Verify), trong đó $S = \{e_1,e_2,...,e_n\}$ là một tập hợp có thứ tự gồm n phần tử. Giả sử bảng dữ liệu truy vấn có các trường $K=\{k_1,k_2,...,k_m\}$ và trường A là khóa chính của bảng. Nghiên cứu trước đây [14] đề xuất xây dựng m cấu trúc I-SALi với đặc điểm rằng bộ $e = (A,k_i)$ chứa cặp giá

trị của khóa chính *A* và trường k_i. Bên cạnh đó, các tác giả cũng xây dựng cấu trúc I-ASL$_{ID}$ với bộ *e = (A,K)* chứa toàn bộ dữ liệu của bảng truy vấn.

Cấu trúc dữ liệu của hai thành phần là node và phần tử trong I-ASL được mô tả như sau [14]:

```
class Node {
    String hashValue;       //giá trị băm của nút
    I-ASLNode next;         //nút tiếp theo
    I-ASLNode pre;          //nút trước
}
class I-ASLNode {
        String attribute;   //giá trị thuộc tính của nút, A
        String key;         //ID của nút, K
        int level;          //thông tin level
        Node [] nodeList;   /mảng các nút
}
```

Hướng dòng băm trong I-ASL được xác định từ dưới lên trên, từ phải sang trái. Do đó, $d_{+\infty,maxlevel}$ được lưu trữ trong nút ở cấp cao nhất của nút bảo vệ bên trái (left sentinel, phần tử đầu tiên của danh sách liên kết) biểu diễn giá trị đặc trưng của toàn bộ I-ASL. Mã giả của quá trình tính toán giá trị băm của một nút trong I-ASL được mô tả như sau [14]:

Input: **T** (e_i, level)

Output: $d_{e_i,level}$

Algorithm:

1. If (**T** is a tail node (key = +∞)), $d_{+\infty,k}$ = *h(0)*, return;
2. If (**T** is a bottom node (level = 0), $d_{e_i,0}$ = *h(h(*e_i*.K,* e_i*.A),* $d_{e_i+1,0}$*)*, return;
3. Else if (ei.next is plateau node, $d_{e_i,k}$ = *h(*$d_{e_{i+1},k}$, $d_{e_i,k-1}$*)*, return;
4. Else $d_{e_i,k}$ = *h(*$d_{e_i,k-1}$*)*

Trong quá trình khởi tạo I-ASL, cấu trúc I-ASL cho mỗi thuộc tính là độc lập. Trong các truy vấn kết hợp giữa phép chiếu (Projection) và lựa chọn (Selection), thuật toán Verify cần xác định và lấy tập khóa chính (ID) của các bản ghi. Sau đó, dựa trên ID, truy vấn tất cả các cặp thuộc tính thoả mãn trong mmm I-ASL tương ứng với mmm thuộc tính chiếu. Các chuỗi bằng chứng (proof sequences) được tạo ra từ việc truy vấn từng I-ASL của mmm thuộc tính sẽ tạo thành *m* tập hợp bằng chứng.

Do cấu trúc yêu cầu tính toán lại toàn bộ *m* tập bằng chứng, I-ASL sẽ phát sinh chi phí cao khi xử lý các bảng dữ liệu lớn với nhiều trường thuộc tính. Trong trường hợp này, cần có một cơ chế xác thực mới để giải quyết vấn đề với chi phí thấp hơn.

2.2. Phương pháp cải tiến

Để khắc phục các thách thức đã nêu, các kỹ thuật sau được đề xuất:

- Xác thực truy vấn chiếu - lựa chọn (Projection-Selection Queries Authentication):

Một cơ chế xác thực gia tăng (incremental authentication) được giới thiệu nhằm tránh việc tính toán lại các giá trị xác thực cho toàn bộ cơ sở dữ liệu khi thực hiện các phép chiếu - lựa chọn.

- Đầu tiên, bản đồ (map) được sử dụng thay vì bộ giá trị (tuple) trong cấu trúc I-ASL$_{ID}$. Ngoài cặp *(A,K)*, giá trị băm của từng cặp *(A,*k_i*)* được tính toán và lưu trữ dựa trên *m* cấu trúc I-ASL*i* tương ứng.
- Do đó, thay vì chỉ chứa 2 phần tử, bản đồ *E* sẽ chứa *m+2* phần tử. Điều này làm tăng không gian lưu trữ, nhưng giúp giảm thời gian xác thực dữ liệu.

Với một quan hệ *R*, cần thiết lập *k* I-ASL tương ứng với kkk thuộc tính của quan hệ (không bao gồm khóa chính). Sau đó, cấu trúc EI-ASL$_{ID}$ được xây dựng thay thế cho I-ASL$_{ID}$.

Cấu trúc EI-ASL$_{ID}$ được biểu diễn dưới dạng bộ sáu *(S,D,InitHash,Update,Query,Verify)*, tương tự I-ASL$_{ID}$. Tuy nhiên, *S={e,d},* trong đó:

- e: Tập hợp tất cả các giá trị phần tử quan hệ có thứ tự.
- d: Tập hợp giá trị băm của chúng, được tính từ *k* cấu trúc I-ASL tương ứng với khóa chính.

- *Xác thực truy vấn kết nối (Join Queries Authentication):*

Một cơ chế xác thực được phát triển cho các truy vấn kết nối (join queries) sử dụng các phép toán "$>$" và "$\geq$". Các phép tính này chưa được triển khai trong nghiên cứu trước đây [14].

Cơ chế xác thực này đảm bảo việc xác minh hiệu quả các truy vấn kết nối phức tạp trong cơ sở dữ liệu, giúp cải thiện hiệu suất và khả năng mở rộng của hệ thống.

3. Đánh giá cải tiến

Trong các phương pháp xác thực dữ liệu hiện tại, Merkle Hash Tree (MHT) và Signature Chain là những kỹ thuật phổ biến được sử dụng để đảm bảo tính toàn vẹn của dữ liệu trong các hệ thống phân tán. Cả MHT và Signature Chain đều có thể cung cấp chứng minh O(log n) cho các phép toán xác thực, nhưng chúng hoạt động trên các cấu trúc dữ liệu khác nhau, mỗi phương pháp có ưu nhược điểm riêng. MHT sử dụng cấu trúc cây nhị phân hoàn chỉnh, trong đó các băm được tính toán và duy trì trên toàn bộ cấu trúc cây, trong khi Signature Chain sử dụng chuỗi các chữ ký kỹ thuật số để đảm bảo tính toàn vẹn của dữ liệu [15], [16].

Tuy nhiên, việc so sánh trực tiếp giữa ASL và các phương pháp này là không khả thi, vì mỗi phương pháp có các yêu cầu khác nhau về bộ nhớ, chi phí tính toán và khả năng cập nhật dữ liệu. ASL, dựa trên cấu trúc danh sách liên kết có nhảy, mang lại lợi thế trong việc tìm kiếm và cập nhật nhanh chóng với độ phức tạp O(log n), nhưng khác với MHT và Signature Chain về cách thức tổ chức và xác thực dữ liệu [17]. Đặc biệt, trong khi MHT và Signature Chain thích hợp cho các tình huống dữ liệu ít thay đổi hoặc không có sự thay đổi liên tục, ASL lại hoạt động tốt hơn trong môi trường yêu cầu cập nhật dữ liệu thường xuyên. Do đó, thay vì so sánh trực tiếp với các phương pháp này, bài báo chỉ tập trung vào việc so sánh các cải tiến mới với thuật toán gốc, I-ASL, nhằm chứng minh tính hiệu quả của các cải tiến trong môi trường thực tế.

Để kiểm tra hiệu quả của các kỹ thuật đề xuất, các thí nghiệm đã được thực hiện trên các bộ dữ liệu thực tế. Bộ dữ liệu được xây dựng từ 10.000 bản ghi và tiến hành 3 thí nghiệm với cấu trúc 5 thuộc tính, 10 thuộc tính và 15 thuộc tính. Các bản ghi dữ liệu được lưu trữ trong hệ quản trị có sở dữ liệu MySQL.

Bảng 1. *Thời gian xác thực trong các thí nghiệm về xác thực truy vấn Projection-Selection*

Thí nghiệm	5 thuộc tính		10 thuộc tính		15 thuộc tính	
	EI-ASL	**I-ASL**	**EI-ASL**	**I-ASL**	**EI-ASL**	**I-ASL**
Thời gian (ms)	180,89	1042,71	249,56	2210,24	279,48	3183,93
Dung lượng (KB)	893	386	1330	422	1762	456

Dựa trên kết quả trong Bảng 1, thời gian xác thực khi lựa chọn 5 thuộc tính bằng phương pháp cũ lớn hơn khoảng 5,79 lần so với phương pháp được đề xuất trên cùng một cơ sở dữ liệu. Thêm vào đó, thời gian xác thực khi thực hiện lựa chọn 10 thuộc tính và 15 thuộc tính lần lượt cao hơn 8,86 và 11,39 lần. Tuy nhiên, kích thước bộ nhớ của EI-ASL cao hơn 2,31 lần so với I-ASL trong trường hợp 5 thuộc tính, và tăng lên 3,15 lần với 10 thuộc tính và 3,86 lần với 15 thuộc tính.

Các kết quả này chỉ ra rằng phương pháp xác thực Skip list được cải tiến có hiệu suất tính toán và thời gian phản hồi truy vấn vượt trội hơn so với các phương pháp trước. Các thí nghiệm cũng làm nổi bật khả năng thích ứng của phương pháp tiếp cận mới với nhiều kích thước cơ sở dữ liệu và khối lượng truy vấn khác nhau. Tuy nhiên, phương pháp trong nghiên cứu này làm tăng kích thước bộ nhớ cho cùng một cơ sở dữ liệu.

4. Kết luận và hướng phát triển

Bài báo này đã trình bày các kỹ thuật mới nhằm cải thiện việc xác thực Skip list trong tối ưu hóa truy vấn quan hệ. Bằng cách giải quyết những hạn chế của các phương pháp hiện có, các kỹ thuật được đề xuất đã nâng cao hiệu quả và khả năng thích ứng của xác thực Skip list. Kết quả thí nghiệm xác nhận tính hiệu quả của nó và tiềm năng đóng góp vào việc tối ưu hóa các hệ thống cơ sở dữ liệu quan hệ.

Trong các nghiên cứu trong tương lai, nhóm nghiên cứu dự định khám phá thêm các phương pháp tối ưu hóa cho xác thực Skip list, chẳng hạn như việc tận dụng các thuật toán học máy để đưa ra quyết định xác thực thông minh. Hơn nữa, các kỹ thuật được đề xuất có thể được tích hợp với các chiến lược tối ưu hóa truy vấn khác, mở ra một hướng đi thú vị cho sự phát triển.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] S. Artzi, K. A. Ross, and R. Secord, "Skip list-based algorithms for distributed databases," *Journal of Parallel and Distributed Computing*, vol. 59, no.1, pp. 111-136, 1999.

[2] X. Feng and M. Wang, "Secure indexing scheme for outsourced databases with skip lists," in *Proceedings of the International Conference on Information Security and Cryptology*, 2007, pp. 67-79.

[3] Y. Li, Y. Zhang, and J. Chen, "An efficient authentication scheme for outsourced databases based on skip list," in *Proceedings of the International Conference on Computational Science and Engineering*, 2012, pp. 355-360.

[4] Y. Wu, D. Li, and S. Wang, "An efficient and secure authentication scheme for outsourced databases based on skip list," *Security and Communication Networks*, vol. 9, no. 10, pp. 1323-1332, 2016.

[5] Y. Zhang, Y. Li, and J. Chen, "An efficient and scalable authentication scheme for outsourced databases based on skip list," *Journal of Network and Computer Applications*, vol. 39, pp. 315-324, 2014.

[6] C. Chen and P. Wang, "Research on the skip list-based indexing method for relational databases," in *Proceedings of the International Conference on Digital Manufacturing and Automation*, 2011, pp. 464-467.

[7] A. Aris, T. G. Michael, and T. Roberto, "Persistent Authenticated Dictionaries and Their Applications," in *Information Security. ISC 2001. Lecture Notes in Computer Science*, vol. 2200, Berlin, Heidelberg: Springer, 2001, pp. 379–393.

[8] Y. Li, Y. Zhang, and J. Chen, "An enhanced authentication scheme for outsourced databases based on skip list," *Security and Communication Networks*, vol. 6, no. 4, pp. 485-494, 2013.

[9] F. Hou, D. Gu, N. Xiao, F. Liu, and H. He, "Performance and Consistency Improvements of Hash Tree Based Disk Storage Protection," in *International Conference on Networking, Architecture, and Storage*, 2009, pp. 51–56.

[10] P. Raju, R. Kadekodi, V. Chidambaram, and I. Abraham, "PebblesDB: Building Key-Value Stores using Fragmented Log-Structured Merge Trees," in *Proceedings of the 26th ACM Symposium on Operating Systems Principles*, 2017, pp. 497-514.

[11] X. Chen, J. Liu, and Z. Cai, "Efficient query processing in distributed databases using skip list," in *Proceedings of the International Conference on Intelligent Computing and Internet of Things*, 2015, pp. 185-192.

[12] F. Duan, J. Li, and J. Lu, "A novel indexing method based on skip list for relational databases," *Journal of Intelligent and Fuzzy Systems*, vol. 36, no. 3, pp. 2011-2021, 2019.

[13] Z. Liu and S. Xu, "An improved skip list indexing method for relational databases," *Journal of Computational Information Systems*, vol.14, no. 1, pp. 285-292, 2018.

[14] J. Xu, Z. Cao, Q. Xiao, and F. Zhou, "An Improved Authenticated Skip List for Relational Query Authentication," *2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications*, 2014, pp. 229-232.

[15] G. Miklau and D. Suciu, "A Formal Approach to Securing Data Provenance," *ACM Transactions on Database Systems (TODS)*, vol. 28, no. 3, pp. 317-360, 2003.

[16] A. Buldas, "Efficient and Flexible Authentication of Data Structures," *Journal of Cryptology*, vol. 18, no. 4, pp. 300-334, 2005.

[17] G. D. Battista and B. Palazzi, "Authenticated Relational Tables and Authenticated Skip Lists," in *Data and Applications Security XXI. DBSec 2007. Lecture Notes in Computer Science*, vol. 4602. Berlin, Heidelberg: Springer, 2007, pp. 31-46.